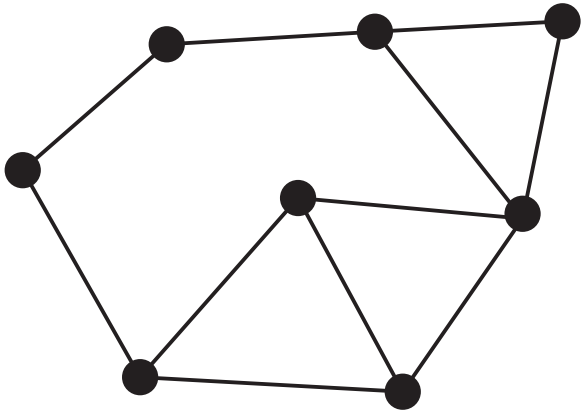
Minimum weight clique cover in claw-free perfect graphs

Flavia Bonomo [1]

[1]Departamento de Computación, FCEN, Universidad de Buenos Aires

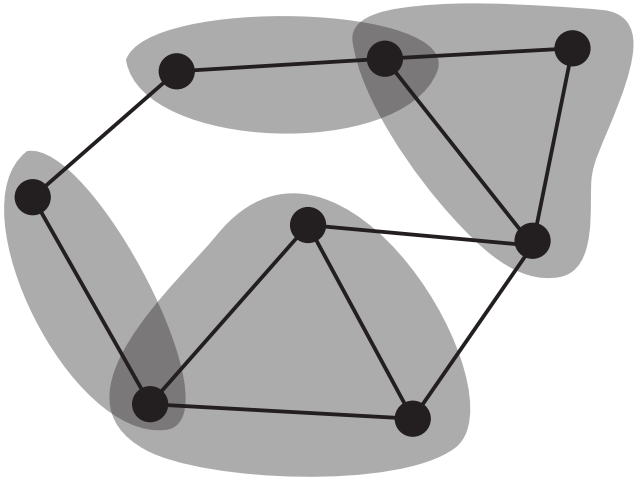
Mathematical research seminar, University of Primorska, Koper, Slovenia, October 6th 2014

Clique cover



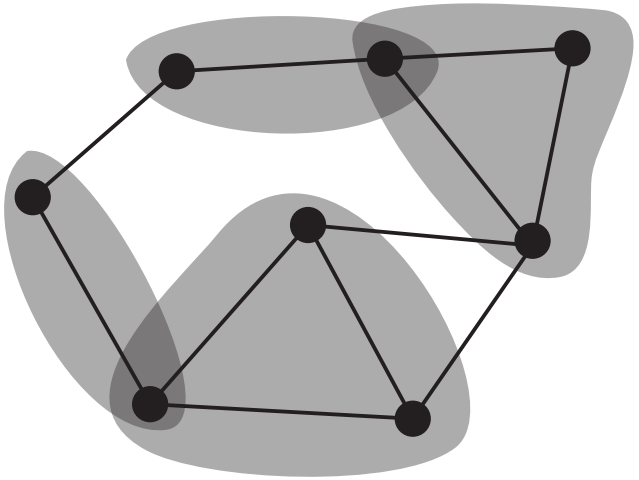
Task: Cover all the vertices of a graph by cliques.

Clique cover



Task: Cover all the vertices of a graph by cliques.

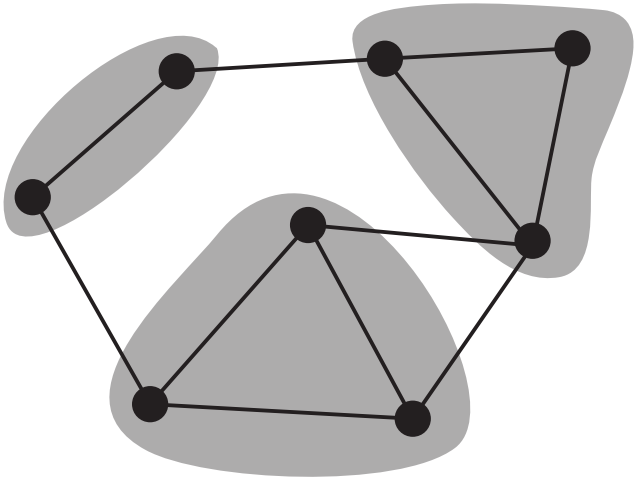
Clique cover



Task: Cover all the vertices of a graph by cliques.

Goal: Minimize the number of cliques (MCC).

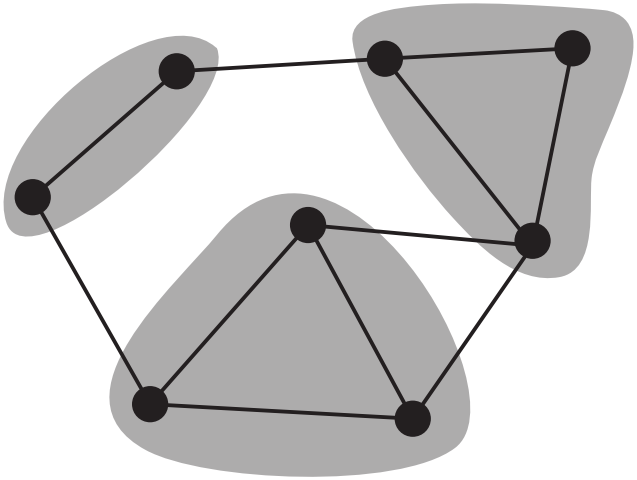
Clique cover



Task: Cover all the vertices of a graph by cliques.

Goal: Minimize the number of cliques (MCC).

Clique cover

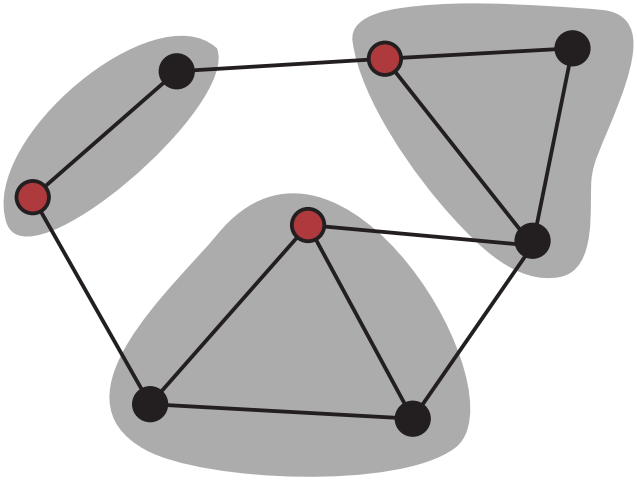


Task: Cover all the vertices of a graph by cliques.

Goal: Minimize the number of cliques (MCC).

Complexity: NP-hard in general.

Clique cover



Task: Cover all the vertices of a graph by cliques.

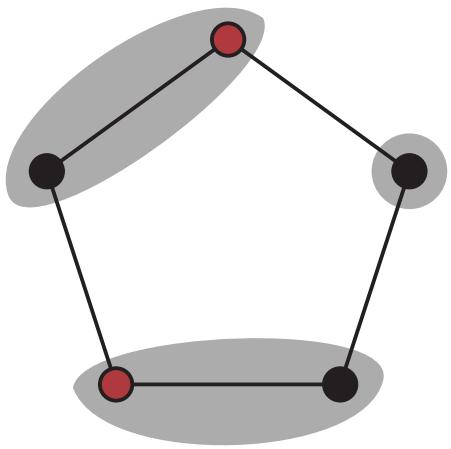
Goal: Minimize the number of cliques (MCC).

Complexity: NP-hard in general.

Lower bound: Maximum stable set (MSS).

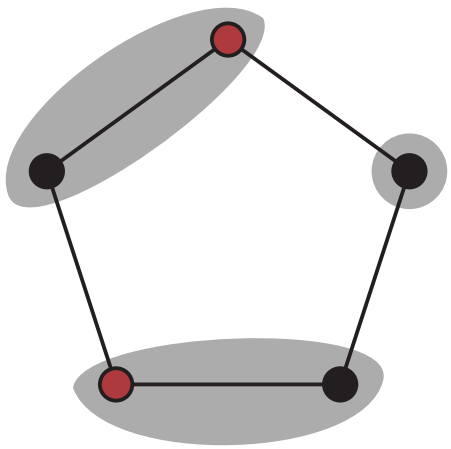
Perfect graphs

The values of a MCC and a MSS are not always the same....



Perfect graphs

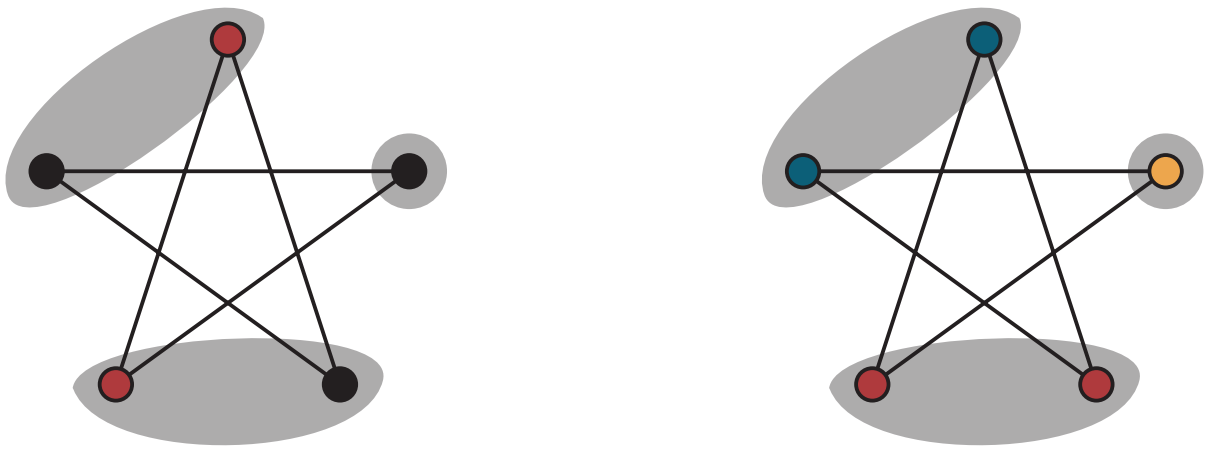
The values of a MCC and a MSS are not always the same....



A graph is perfect if and only if MCC = MSS for every induced subgraph (Perfect Graph Theorem, Lóvasz 1972).

Perfect graphs

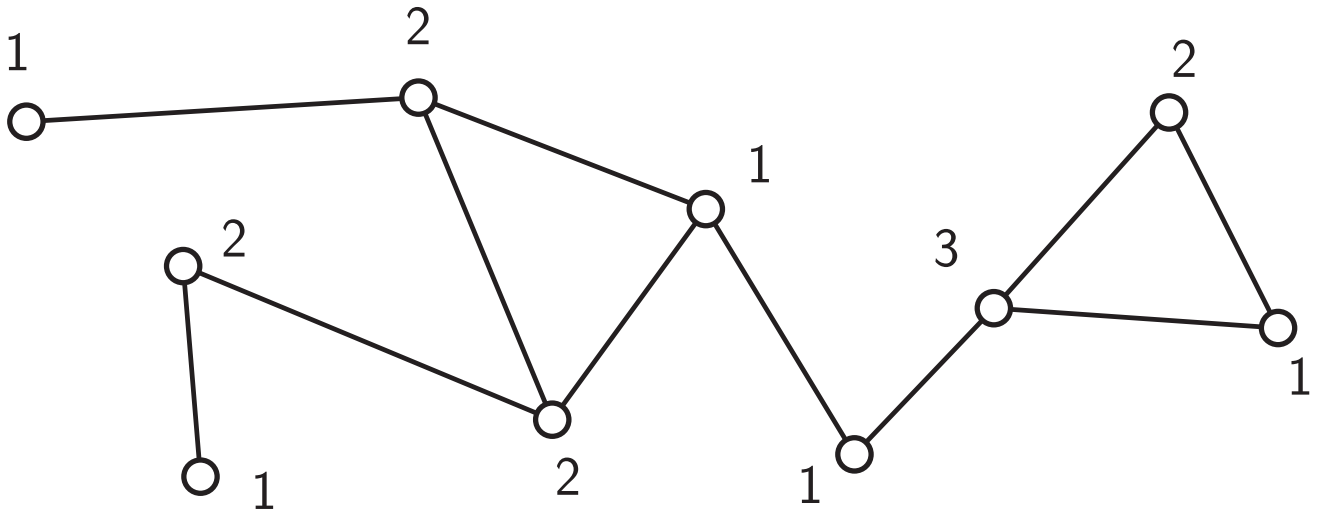
The values of a MCC and a MSS are not always the same....



A graph is perfect if and only if MCC = MSS for every induced subgraph (Perfect Graph Theorem, Lóvasz 1972). Perfect graphs were defined by Berge in 1960 as the graphs such that the clique number and chromatic number are equal for every induced subgraph.

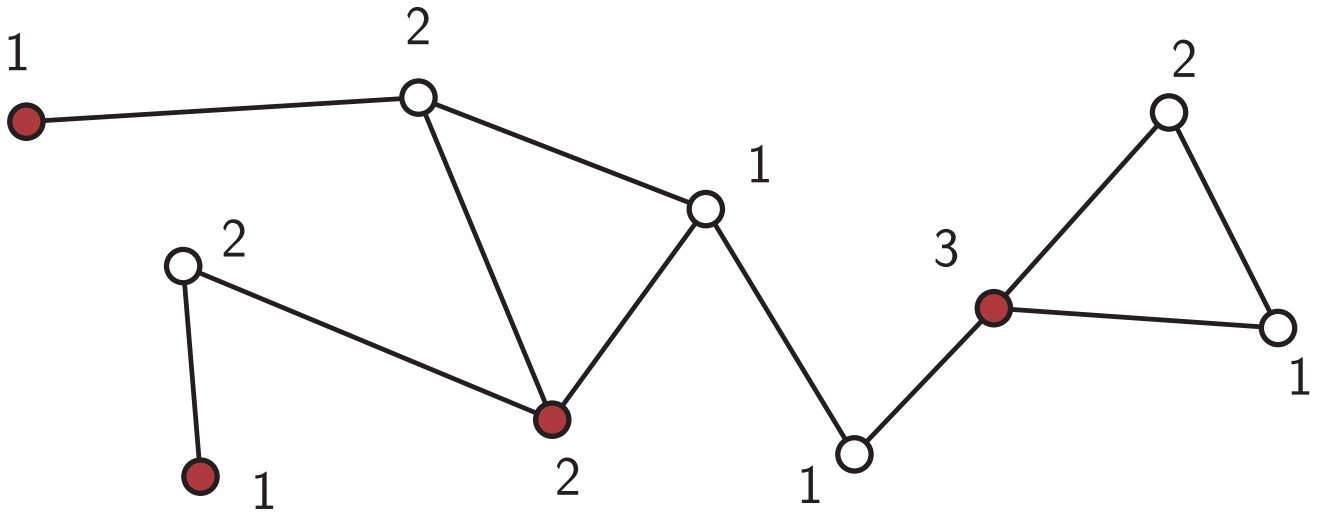
The weighted version

Maximum weighted stable set (MWSS): Given a graph $G(V, E)$ with a nonnegative weight function on the vertices w, find a set of pairwise nonadjacent vertices maximizing the sum of their weight.



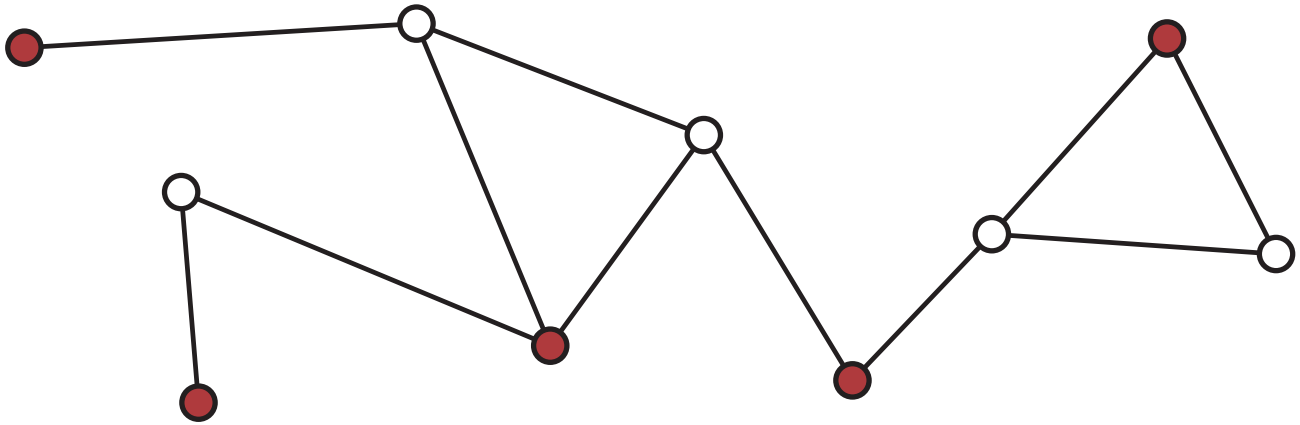
The weighted version

Maximum weighted stable set (MWSS): Given a graph $G(V, E)$ with a nonnegative weight function on the vertices w, find a set of pairwise nonadjacent vertices maximizing the sum of their weight.



The weighted version

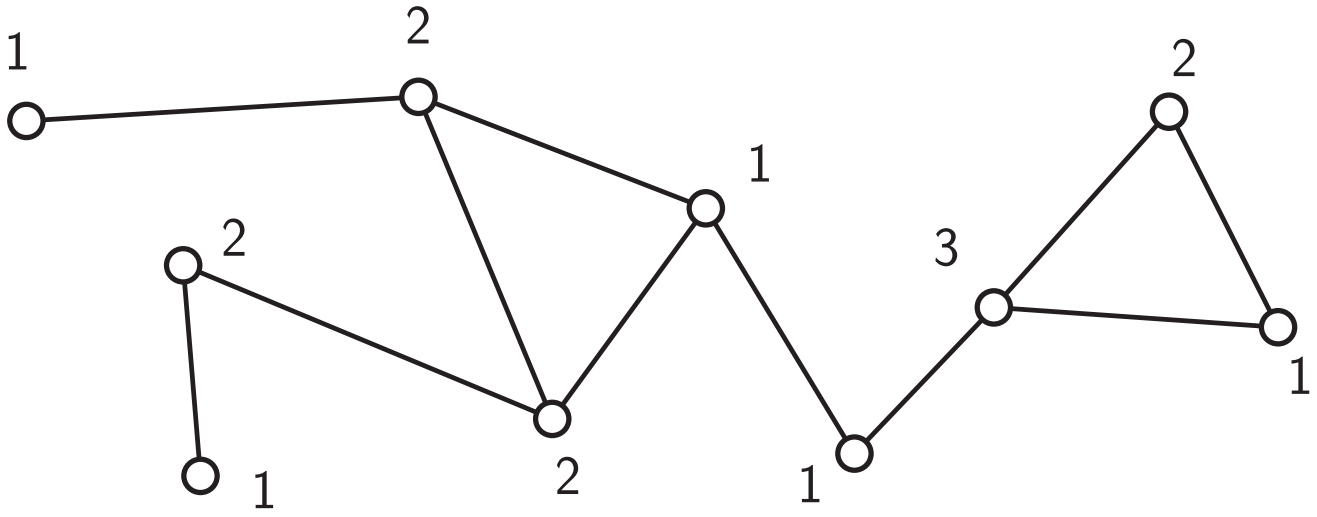
Maximum weighted stable set (MWSS): Given a graph $G(V, E)$ with a nonnegative weight function on the vertices w, find a set of pairwise nonadjacent vertices maximizing the sum of their weight.



Not always it is of maximum cardinality.

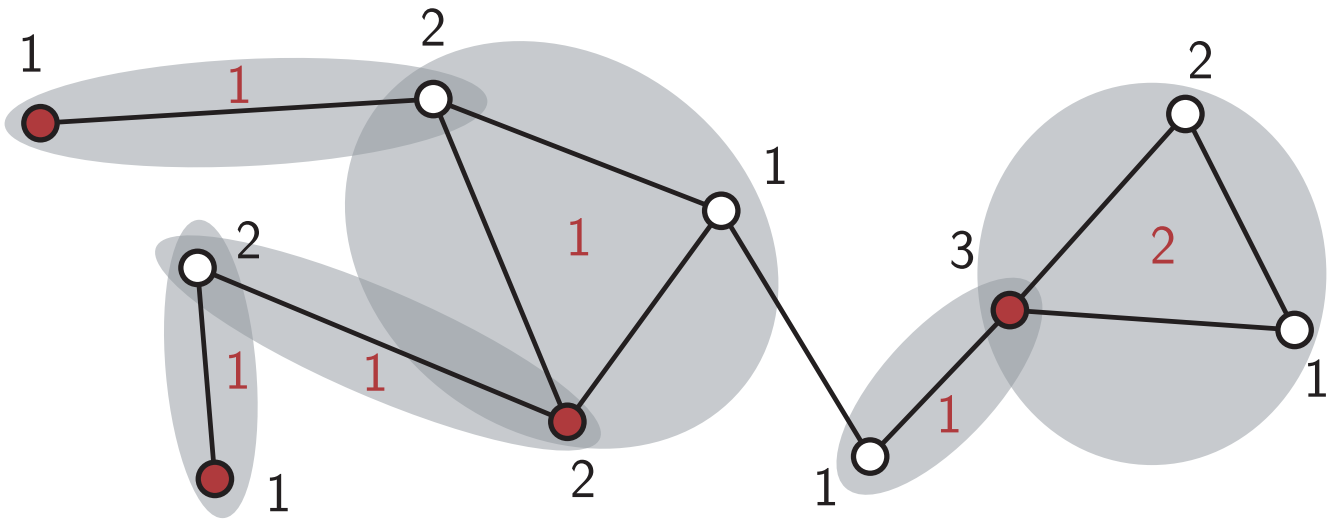
Minimum weight clique cover (MWCC)

Given a graph $G(V, E)$ with a nonnegative weight function on the vertices w, find a collection of cliques $\mathcal{K}$ and a non negative value y_K for each clique $K \in \mathcal{K}$ such that $\sum_{K:v \in K} y_K \geq w(v)$ for every vertex $v \in V$ and $\sum_{K \in \mathcal{K}} y_K$ is minimum.



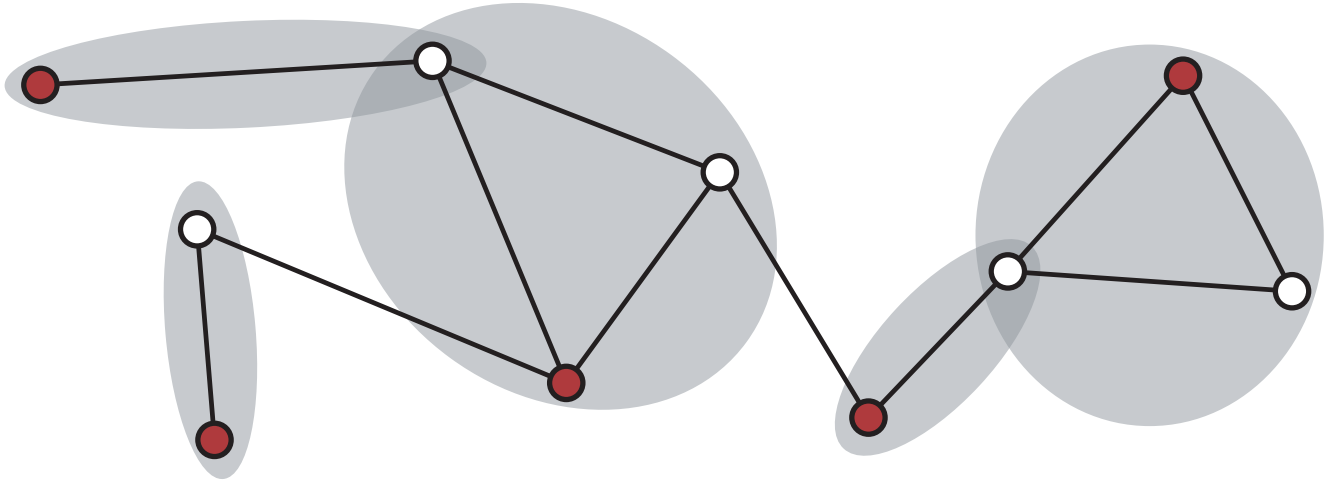
Minimum weight clique cover (MWCC)

Given a graph $G(V, E)$ with a nonnegative weight function on the vertices w, find a collection of cliques $\mathcal{K}$ and a non negative value y_K for each clique $K \in \mathcal{K}$ such that $\sum_{K:v \in K} y_K \geq w(v)$ for every vertex $v \in V$ and $\sum_{K \in \mathcal{K}} y_K$ is minimum.



Minimum weight clique cover (MWCC)

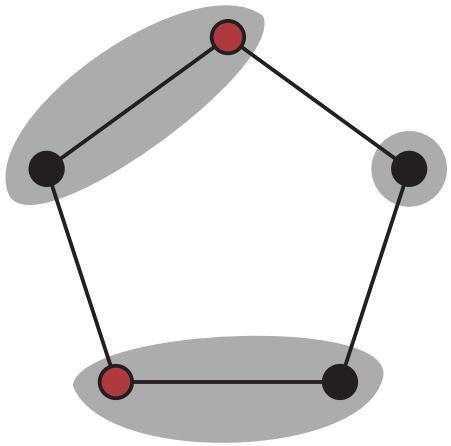
Given a graph $G(V, E)$ with a nonnegative weight function on the vertices w, find a collection of cliques $\mathcal{K}$ and a non negative value y_K for each clique $K \in \mathcal{K}$ such that $\sum_{K:v \in K} y_K \geq w(v)$ for every vertex $v \in V$ and $\sum_{K \in \mathcal{K}} y_K$ is minimum.

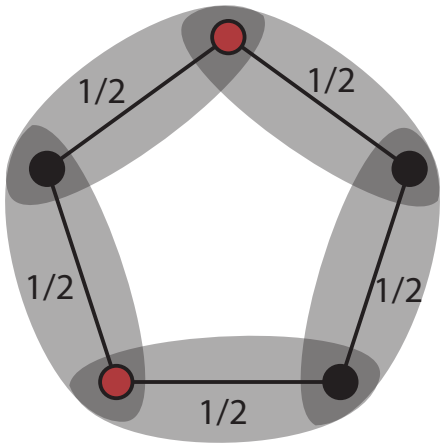


Not always it is of minimum cardinality.

Integrality

Not always the integral optimum is the fractional optimum...





Stable set and clique cover in perfect graphs

- The MWCC is the dual of the linear relaxation of the clique formulation for the MWSS.

$$\max \sum_{v \in V} w(v) x_v \qquad \sum_{v \in K} x_v \leq 1 \quad \forall K \in \mathcal{K}(G) \qquad x_v \geq 0 \quad \forall v \in V$$

$$\min \sum_{K \in \mathcal{K}(G)} y_K \qquad \sum_{K \in \mathcal{K}(G): v \in K} y_K \geq w(v) \quad \forall v \in V \qquad y_K \geq 0 \quad \forall K \in \mathcal{K}(G)$$

- For perfect graphs, the weights of a MWSS and a MWCC are equal and, moreover, for an integer weight function w, there is a MWCC where the weight of each clique is integer (Fulkerson, 1973).
- Both problems can be solved in polytime in perfect graphs through the ellipsoid method and using the Lovász ϑ-function (Grötschel, Lovász, and Schrijver, 1981–1988). What about combinatorial algorithms?

Stable set and clique cover in perfect graphs

- The MWCC is the dual of the linear relaxation of the clique formulation for the MWSS.

$$\max \sum_{v \in V} w(v) x_v \qquad \sum_{v \in K} x_v \leq 1 \quad \forall K \in \mathcal{K}(G) \qquad x_v \geq 0 \quad \forall v \in V$$

$$\min \sum_{K \in \mathcal{K}(G)} y_K \qquad \sum_{K \in \mathcal{K}(G): v \in K} y_K \geq w(v) \quad \forall v \in V \qquad y_K \geq 0 \quad \forall K \in \mathcal{K}(G)$$

- For perfect graphs, the weights of a MWSS and a MWCC are equal and, moreover, for an integer weight function w, there is a MWCC where the weight of each clique is integer (Fulkerson, 1973).
- Both problems can be solved in polytime in perfect graphs through the ellipsoid method and using the Lovász ϑ-function (Grötschel, Lovász, and Schrijver, 1981–1988). What about combinatorial algorithms?

Stable set and clique cover in perfect graphs

- The MWCC is the dual of the linear relaxation of the clique formulation for the MWSS.

$$\max \sum_{v \in V} w(v) x_v \qquad \sum_{v \in K} x_v \leq 1 \quad \forall K \in \mathcal{K}(G) \qquad x_v \geq 0 \quad \forall v \in V$$

$$\min \sum_{K \in \mathcal{K}(G)} y_K \qquad \sum_{K \in \mathcal{K}(G): v \in K} y_K \geq w(v) \quad \forall v \in V \qquad y_K \geq 0 \quad \forall K \in \mathcal{K}(G)$$

- For perfect graphs, the weights of a MWSS and a MWCC are equal and, moreover, for an integer weight function w, there is a MWCC where the weight of each clique is integer (Fulkerson, 1973).
- Both problems can be solved in polytime in perfect graphs through the ellipsoid method and using the Lovász ϑ-function (Grötschel, Lovász, and Schrijver, 1981–1988). What about combinatorial algorithms?

Crucial clique

- A clique that intersects every M(W)SS of the graph.
- G perfect $\Rightarrow$ has a crucial clique
- every clique in a M(W)CC of a perfect graph is crucial (otherwise we've used a clique and still have a stable set of maximum weight to cover!)
- Algorithm:
  - find a crucial clique K
  - assign as weight $\alpha_w(G) - \alpha_w(G - K)$.
- Problem: We don't know in general how to find crucial cliques of perfect graphs. :)

Crucial clique

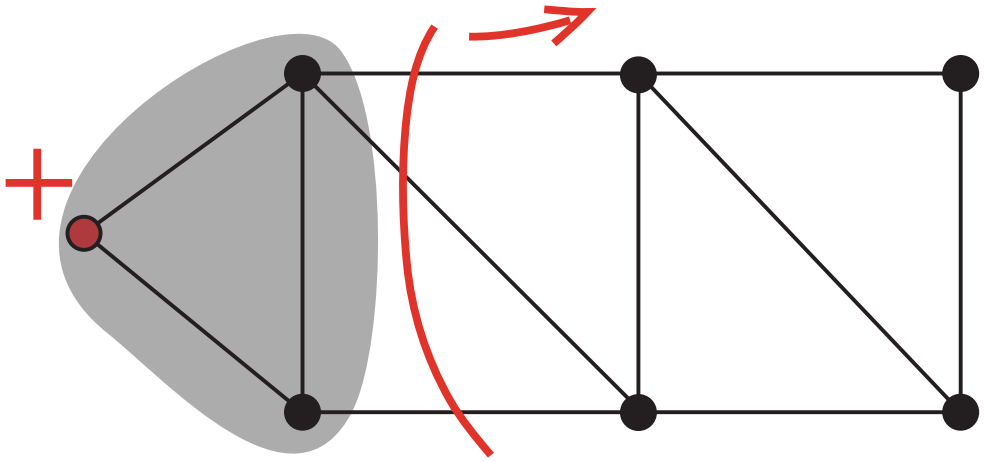
- A clique that intersects every M(W)SS of the graph.
- G perfect $\Rightarrow$ has a crucial clique
- every clique in a M(W)CC of a perfect graph is crucial (otherwise we've used a clique and still have a stable set of maximum weight to cover!)
- Algorithm:
  - find a crucial clique K
  - assign as weight $\alpha_w(G) - \alpha_w(G - K)$.
- Problem: We don't know in general how to find crucial cliques of perfect graphs. :)

Crucial clique

- A clique that intersects every M(W)SS of the graph.
- G perfect $\Rightarrow$ has a crucial clique
- every clique in a M(W)CC of a perfect graph is crucial (otherwise we've used a clique and still have a stable set of maximum weight to cover!)
- Algorithm:
  - find a crucial clique K
  - assign as weight $\alpha_w(G) - \alpha_w(G - K)$.
- Problem: We don't know in general how to find crucial cliques of perfect graphs. :)

Crucial cliques of chordal graphs

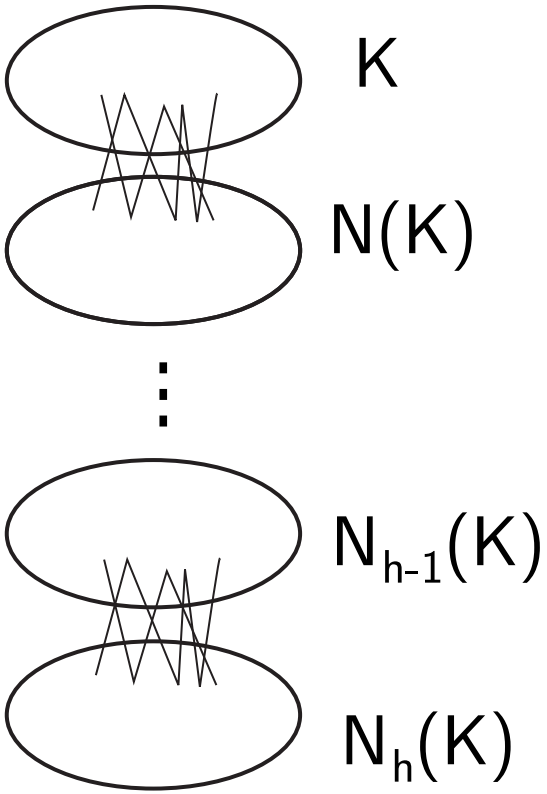
v simplicial vertex with $w(v) > 0 \Rightarrow N[v]$ crucial clique



Weight: we can assign $w(v)$ (we will possibly give further weight to $N(v)$)

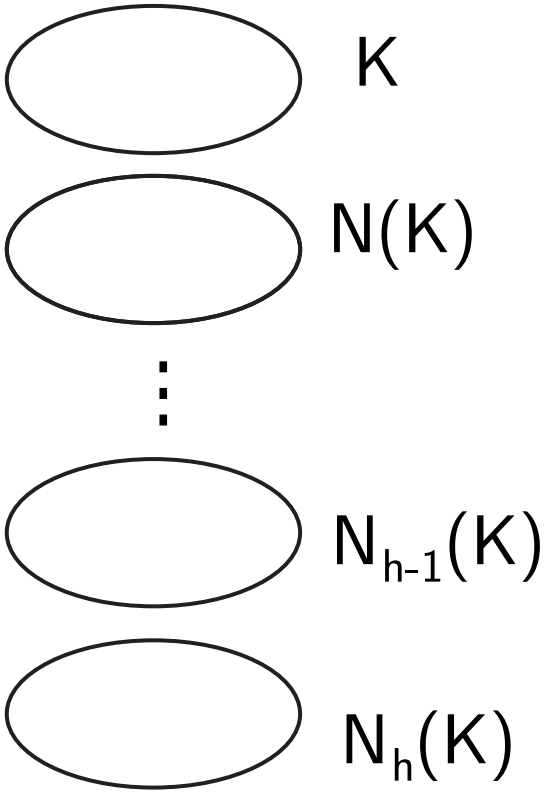
Crucial cliques in distance simplicial graphs

A graph is distance simplicial w.r.t. a clique K if $N(K), N^2(K), \ldots, N^j(K), \ldots$ are cliques.



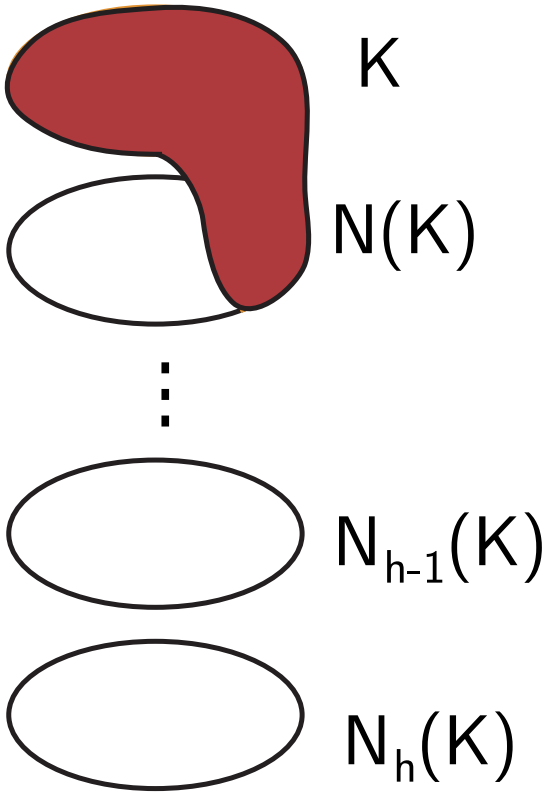
Crucial cliques in distance simplicial graphs

A graph is distance simplicial w.r.t. a clique K if $N(K), N^2(K), \ldots, N^j(K), \ldots$ are cliques.



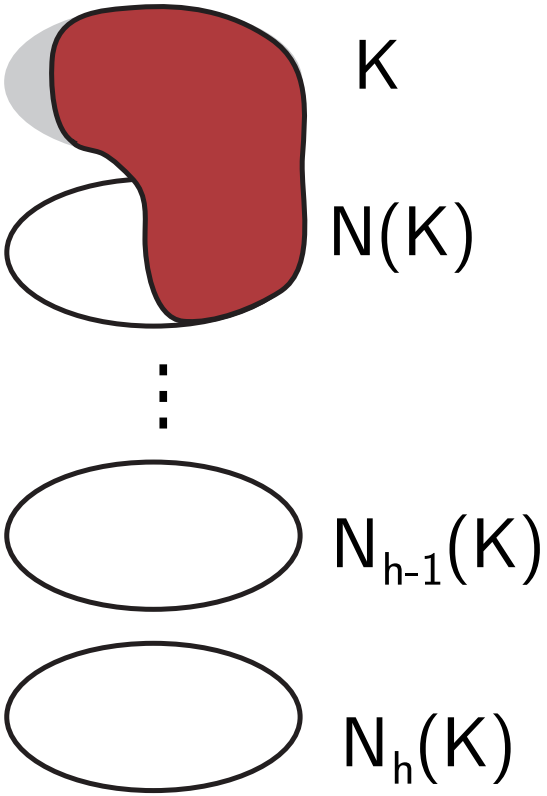
Crucial cliques in distance simplicial graphs

A graph is distance simplicial w.r.t. a clique K if $N(K), N^2(K), \ldots, N^j(K), \ldots$ are cliques.



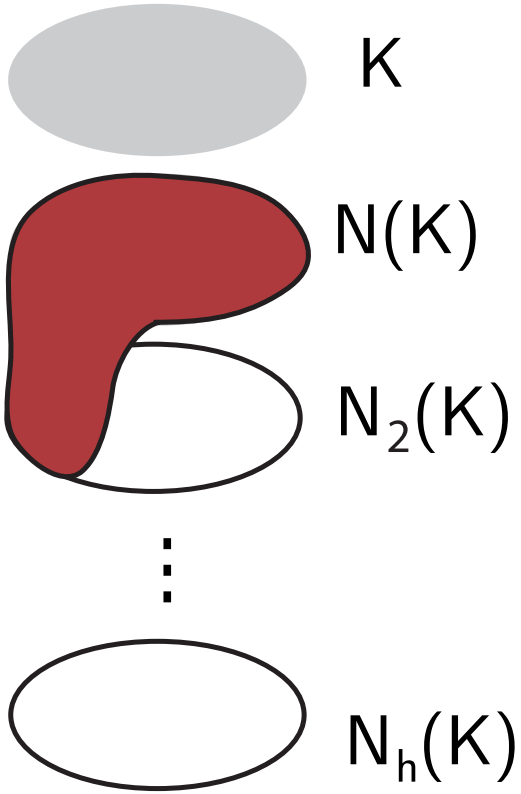
Crucial cliques in distance simplicial graphs

A graph is distance simplicial w.r.t. a clique K if $N(K), N^2(K), \ldots, N^j(K), \ldots$ are cliques.



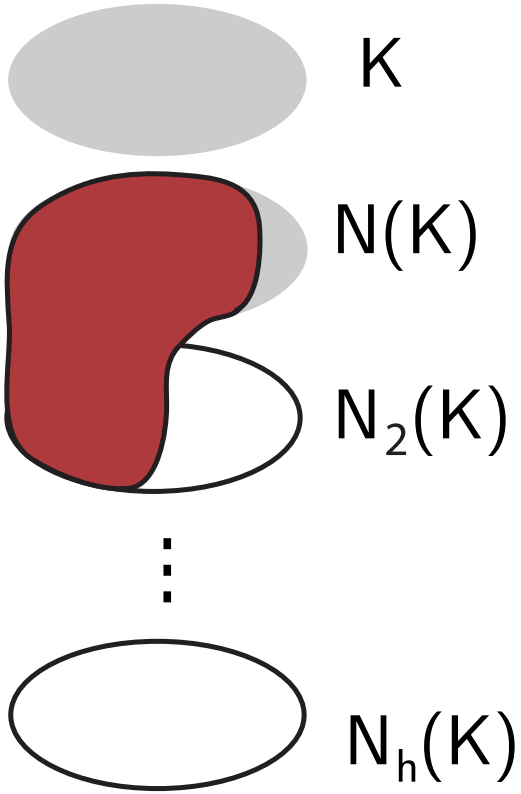
Crucial cliques in distance simplicial graphs

A graph is distance simplicial w.r.t. a clique K if $N(K), N^2(K), \ldots, N^j(K), \ldots$ are cliques.



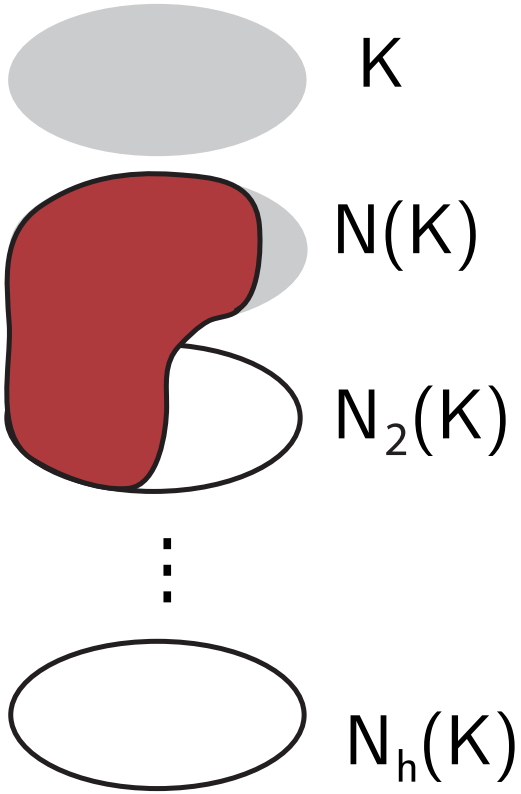
Crucial cliques in distance simplicial graphs

A graph is distance simplicial w.r.t. a clique K if $N(K), N^2(K), \ldots, N^j(K), \ldots$ are cliques.



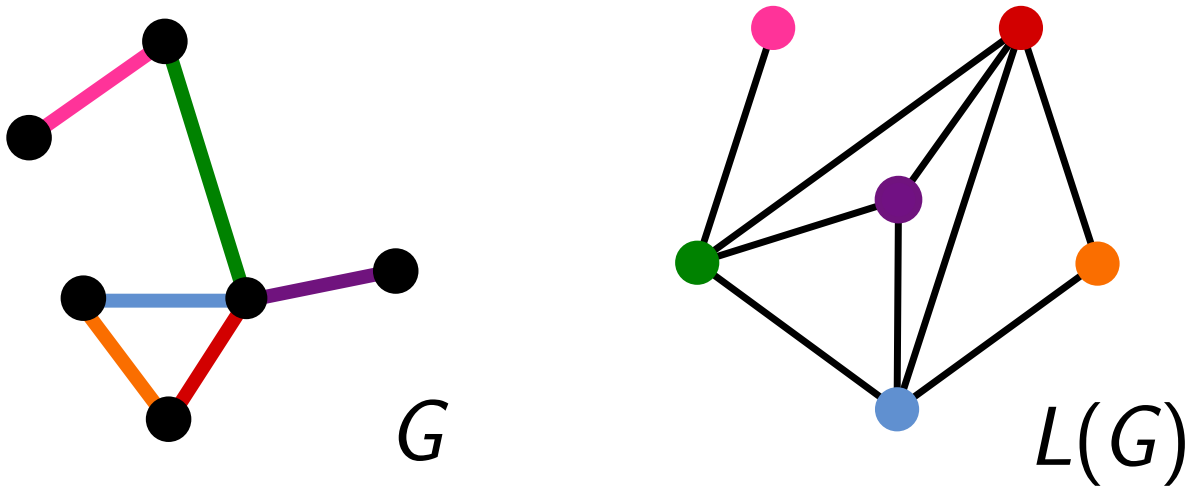
Crucial cliques in distance simplicial graphs

A graph is distance simplicial w.r.t. a clique K if $N(K), N^2(K), \ldots, N^j(K), \ldots$ are cliques.

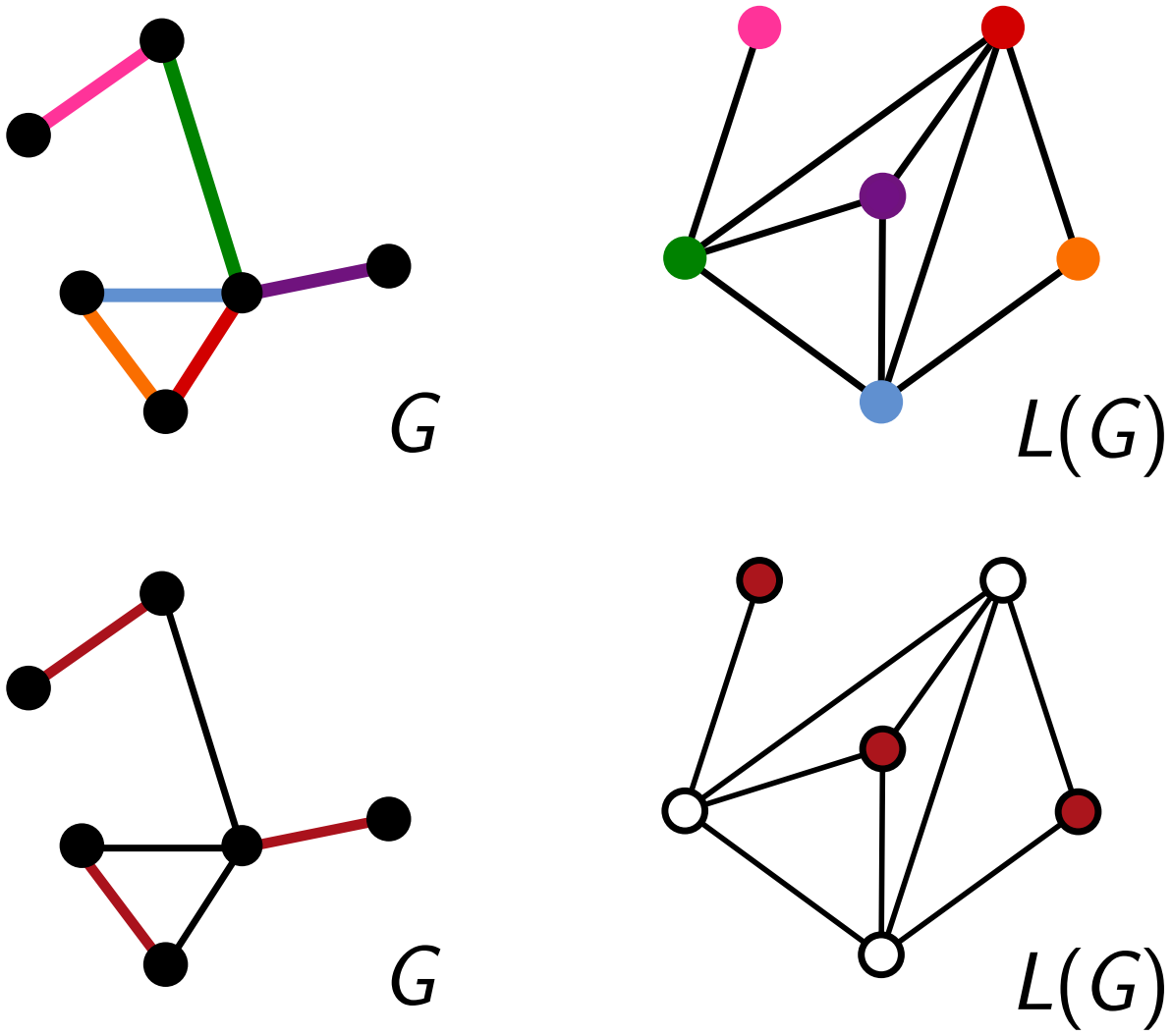


Which other classes admit combinatorial algorithms?

Line graphs



Line graphs

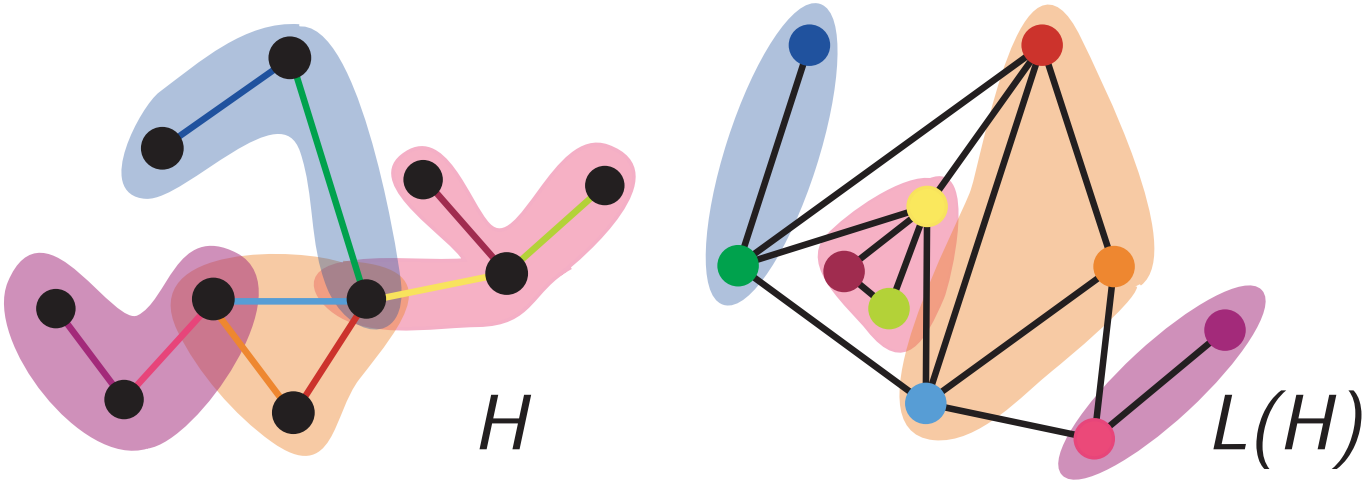


Maximum weighted stable set on line graphs

This problem is equivalent to maximum weighted matching in the root graph. Edmonds' algorithm (1965) can run in $O(\sqrt{n}m)$ time, and there is an algorithm by Mucha and Sankowski based on matrix multiplication that is $O(n^{2.376})$.

On line graphs....

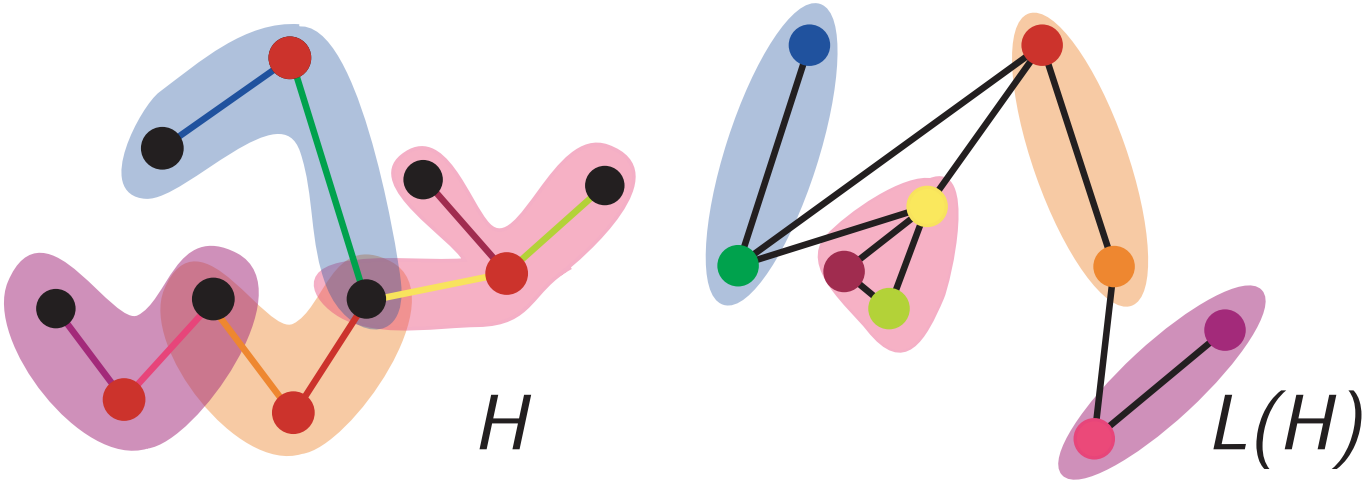
If $G = L(H)$ then a MWCC of G is composed by stars and triangles of H.



Moreover, if H is bipartite, just stars appear and it is equivalent to minimum weighted vertex cover. So, the result by Fulkerson generalizes the Kőnig-Egerváry property for line graphs of bipartite graphs.

On line graphs....

If $G = L(H)$ then a MWCC of G is composed by stars and triangles of H.



Moreover, if H is bipartite, just stars appear and it is equivalent to minimum weighted vertex cover. So, the result by Fulkerson generalizes the Kőnig-Egerváry property for line graphs of bipartite graphs.

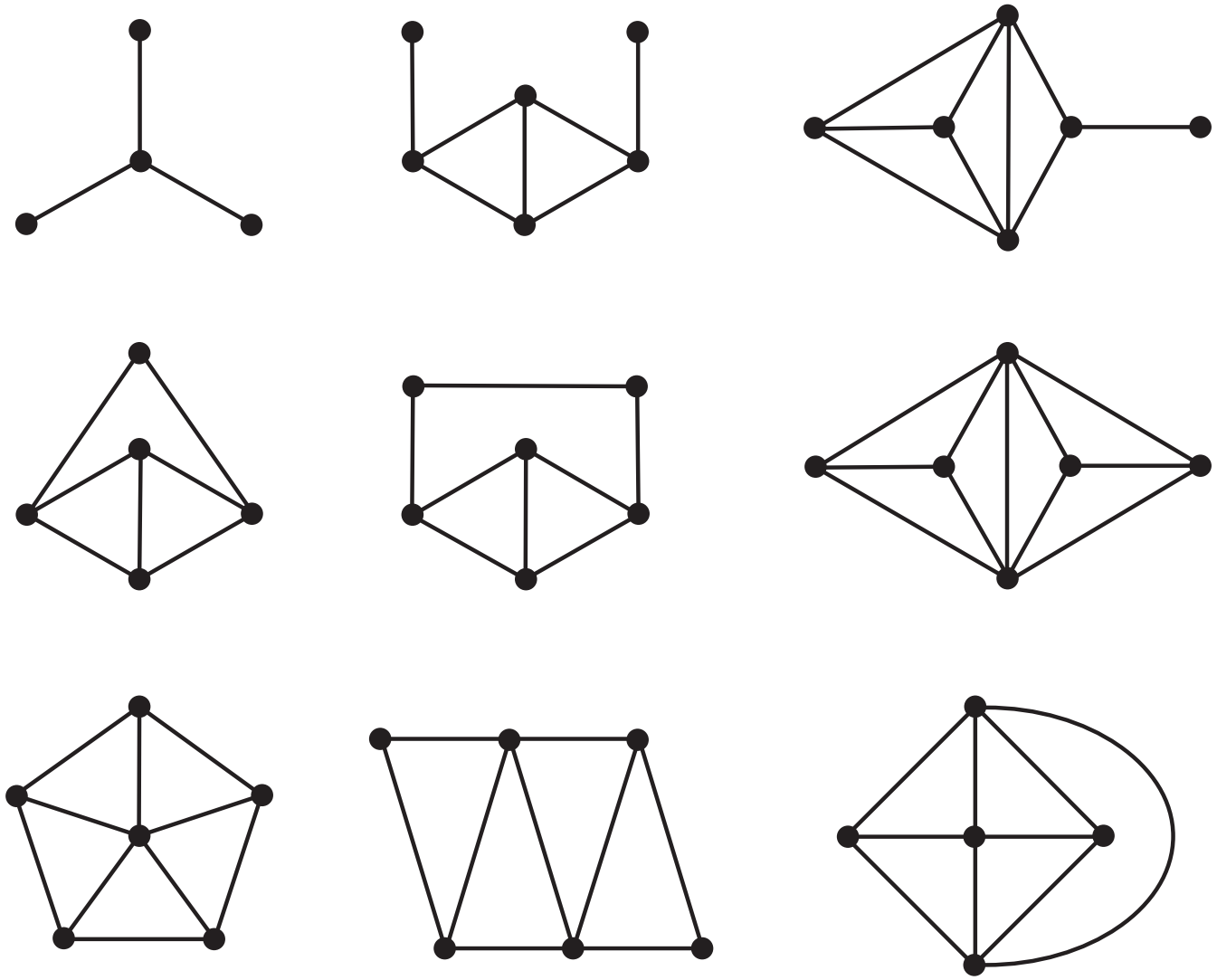
MCC and MWCC on line graphs

To solve the MWCC on the line graph of a bipartite graph the Hungarian method (Kuhn '55) gives a $O(n^3)$-time primal-dual algorithm.

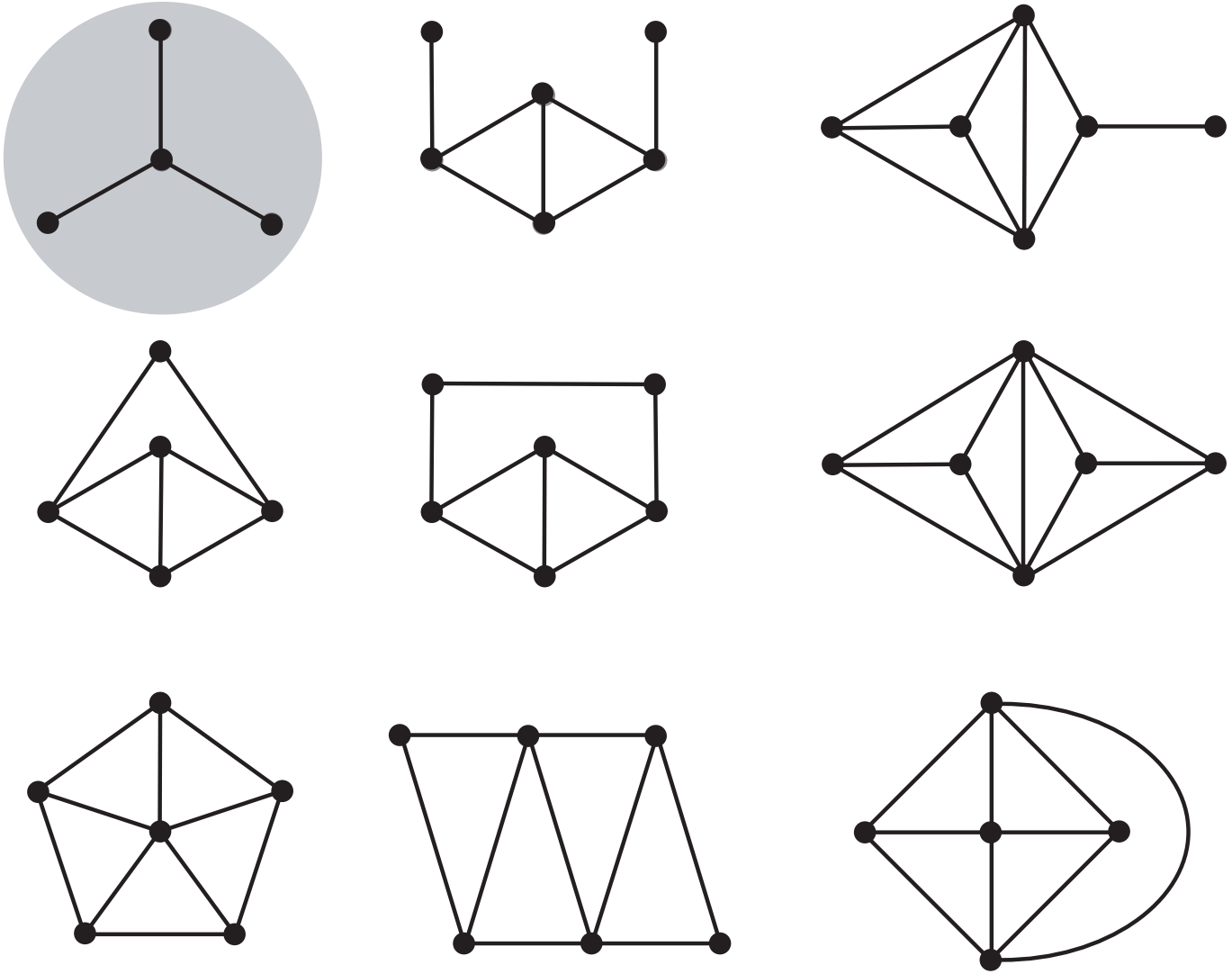
To solve the MWCC on a perfect line graph G, there is a primal-dual algorithm by Gabow (1990) that solves concurrently a maximum weighted matching on the root graph of G and a covering of it by stars and triangles (i.e., cliques of G) in $O(n^2 \log(n))$.

For the unweighted case, a previous (similar) algorithm by Trotter (1977) was known.

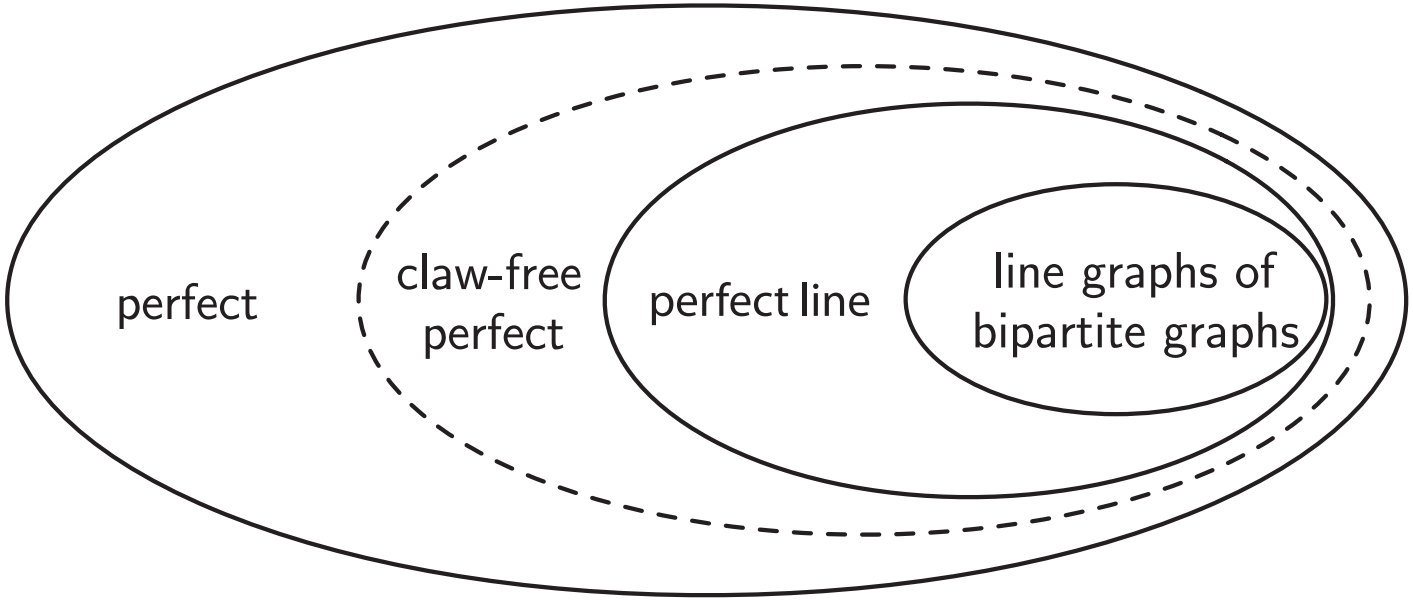
Minimal forbidden induced subgraphs for line graphs (Beineke, 1970)

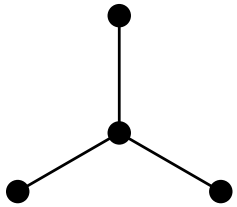


Minimal forbidden induced subgraphs for line graphs (Beineke, 1970)



Claw-free graphs





claw

MSS and MWSS on claw-free graphs (not necessarily perfect)

Combinatorial algorithms:

- Minty 1980, Nakamura and Tamura 2001: based on augmenting paths and reductions ($O(n^6)$).
- Oriolo, Pietropaoli and Stauffer 2008: based on graph decomposition ($O(n^4)$).
- Nobili and Sassano 2011: based on Lovász-Plummer clique reduction and augmenting paths in line graphs ($O(n^4 \log(n))$).
- Faenza, Oriolo and Stauffer 2010: based on a strip decomposition of claw-free graphs ($O(n^3)$).

MCC and MWCC on claw-free perfect graphs

Combinatorial algorithms:

- Hsu and Nemhauser 1981, 1982: building upon a solution of several instances of M(W)SS in order to find crucial cliques ($O(n^5)$).
- Combination of results by Whitesides 1982, Chvátal and Sbihi 1988 and Maffray and Reed 1999 on clique cutsets and claw-free perfect graphs ($O(n^4 \log(n))$, only for the unweighted case).
- B., Oriolo and Snels 2012: based on strip decomposition combined with clique cutsets decomposition ($O(n^3)$).
- B, Oriolo, Snels and Stauffer 2013: building upon a reformulation and 'nice' polyhedra, solved by 2-sat and shortest paths ($O(n^3)$).

MCC and MWCC on claw-free perfect graphs

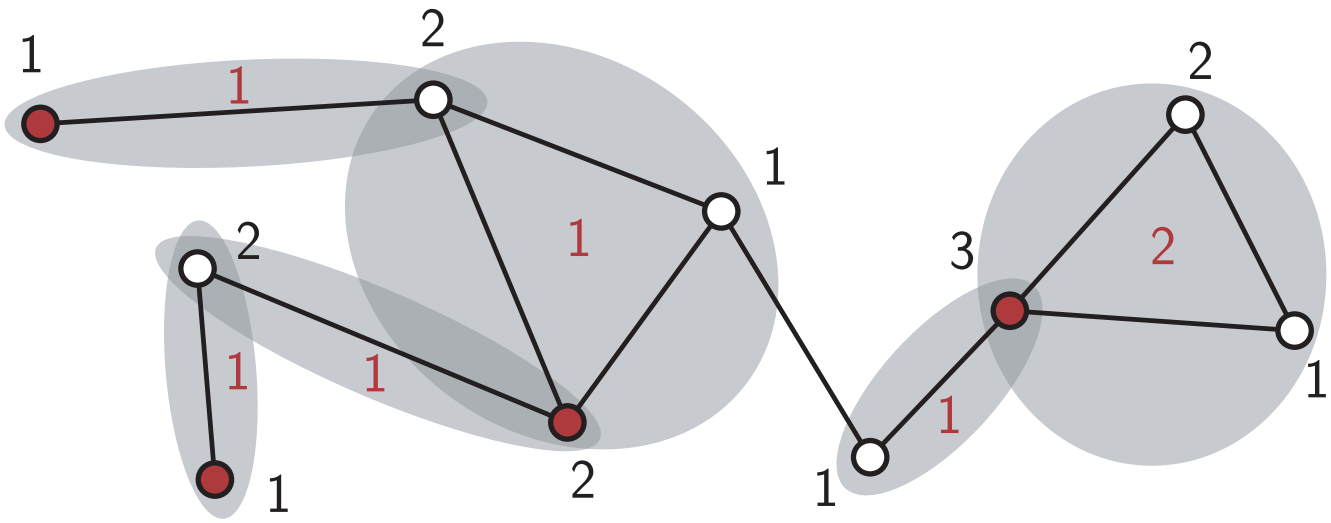
Combinatorial algorithms:

- Hsu and Nemhauser 1981, 1982: building upon a solution of several instances of M(W)SS in order to find crucial cliques ($O(n^5)$).
- Combination of results by Whitesides 1982, Chvátal and Sbihi 1988 and Maffray and Reed 1999 on clique cutsets and claw-free perfect graphs ($O(n^4 \log(n))$, only for the unweighted case).
- B., Oriolo and Snels 2012: based on strip decomposition combined with clique cutsets decomposition ($O(n^3)$).
- B, Oriolo, Snels and Stauffer 2013: building upon a reformulation and 'nice' polyhedra, solved by 2-sat and shortest paths ($O(n^3)$).

All of them are strongly based on perfection: the MCC and MWCC problems are NP-complete on claw-free graphs, e.g. from vertex cover in triangle-free graphs (Garey, Johnson and Stockmeyer, 1976), or coloring in triangle-free graphs (Maffray and Preissmann, 1996).

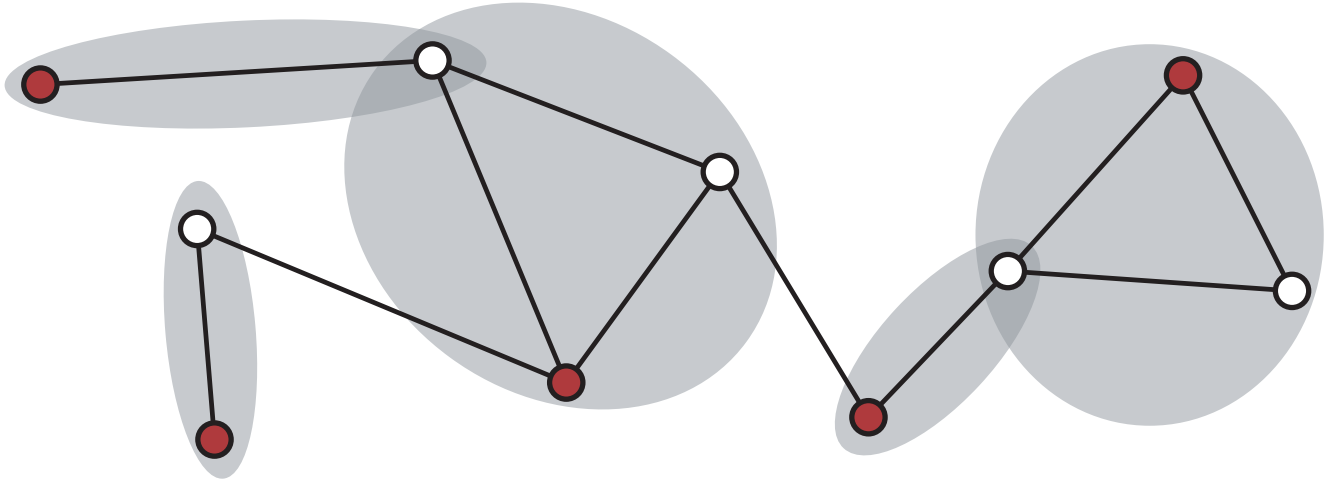
Stable set and clique cover in perfect graphs

Given a perfect graph and a MWSS S and a MWCC y of it, every clique with strictly positive weight intersects S and $\sum_{K:v\in K} y_K = w(v)$ for every vertex $v \in S$.



Stable set and clique cover in perfect graphs

Given a perfect graph and a MWSS S and a MWCC y of it, every clique with strictly positive weight intersects S and $\sum_{K:v\in K} y_K = w(v)$ for every vertex $v \in S$.



In the unweighted case, there is exactly one clique containing each vertex of S.

Outline of Hsu and Nemhauser's algorithm

- Compute a MWSS S of G.
- Fix a vertex $v \in S$
  - For each vertex $z \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z\}$; if $w(S') = w(S)$, mark the vertex z.
  - For each nonedge $\{z, z'\} \subseteq N(v)$, compute a MWSS S' such that $S' \cap N[v] = \{z, z'\}$; if $w(S') = w(S)$, mark the nonedge.
- Compute a clique K containing v, all the marked vertices in $N(v)$, and one endpoint of each marked nonedge in $N(v)$: this will be a crucial clique.
- Compute the weight $y_K = w(S) - \alpha_w(G - K)$ and redefine $w(z) := w(z) - y_K$ for every vertex in K.
- Repeat until $w(v) = 0$ and then proceed with another vertex of S.

Outline of Hsu and Nemhauser's algorithm

- Compute a MWSS S of G.
- Fix a vertex $v \in S$
  - For each vertex $z \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z\}$; if $w(S') = w(S)$, mark the vertex z.
  - for each nonedge $zt \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z, t\}$; if $w(S') = w(S)$, mark the nonedge zt.
- Compute a clique K containing v, all the marked vertices in $N(v)$, and one endpoint of each marked nonedge in $N(v)$: this will be a crucial clique.
- Compute the weight $y_K = w(S) - \alpha_w(G - K)$ and redefine $w(z) := w(z) - y_K$ for every vertex in K.
- Repeat until $w(v) = 0$ and then proceed with another vertex of S.

Outline of Hsu and Nemhauser's algorithm

- Compute a MWSS S of G.
- Fix a vertex $v \in S$
  - For each vertex $z \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z\}$; if $w(S') = w(S)$, mark the vertex z.
  - for each nonedge $zt \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z, t\}$; if $w(S') = w(S)$, mark the nonedge zt.
- Compute a clique K containing v, all the marked vertices in $N(v)$, and one endpoint of each marked nonedge in $N(v)$: this will be a crucial clique.
- Compute the weight $y_K = w(S) - \alpha_w(G - K)$ and redefine $w(z) := w(z) - y_K$ for every vertex in K.
- Repeat until $w(v) = 0$ and then proceed with another vertex of S.

Outline of Hsu and Nemhauser's algorithm

- Compute a MWSS S of G.
- Fix a vertex $v \in S$
  - For each vertex $z \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z\}$; if $w(S') = w(S)$, mark the vertex z.
  - for each nonedge $zt \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z, t\}$; if $w(S') = w(S)$, mark the nonedge zt.
- Compute a clique K containing v, all the marked vertices in $N(v)$, and one endpoint of each marked nonedge in $N(v)$: this will be a crucial clique.
- Compute the weight $y_K = w(S) - \alpha_w(G - K)$ and redefine $w(z) := w(z) - y_K$ for every vertex in K.
- Repeat until $w(v) = 0$ and then proceed with another vertex of S.

Outline of Hsu and Nemhauser's algorithm

- Compute a MWSS S of G.
- Fix a vertex $v \in S$
  - For each vertex $z \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z\}$; if $w(S') = w(S)$, mark the vertex z.
  - for each nonedge $zt \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z, t\}$; if $w(S') = w(S)$, mark the nonedge zt.
- Compute a clique K containing v, all the marked vertices in $N(v)$, and one endpoint of each marked nonedge in $N(v)$: this will be a crucial clique.
- Compute the weight $y_K = w(S) - \alpha_w(G - K)$ and redefine $w(z) := w(z) - y_K$ for every vertex in K.
- Repeat until $w(v) = 0$ and then proceed with another vertex of S.

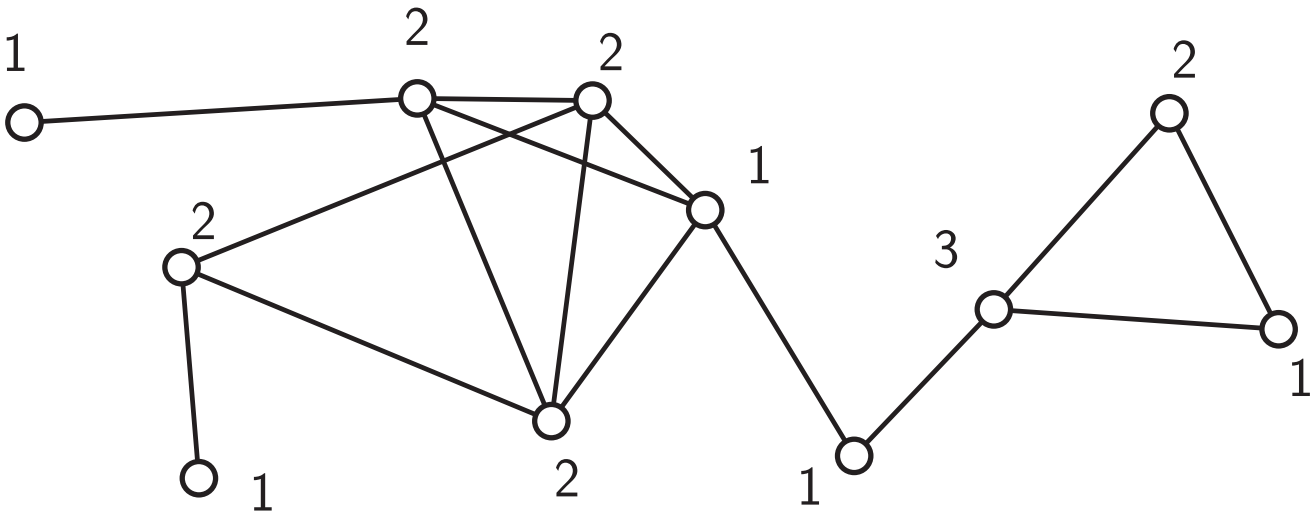
Outline of Hsu and Nemhauser's algorithm

- Compute a MWSS S of G.
- Fix a vertex $v \in S$
  - For each vertex $z \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z\}$; if $w(S') = w(S)$, mark the vertex z.
  - for each nonedge $zt \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z, t\}$; if $w(S') = w(S)$, mark the nonedge zt.
- Compute a clique K containing v, all the marked vertices in $N(v)$, and one endpoint of each marked nonedge in $N(v)$: this will be a crucial clique.
- Compute the weight $y_K = w(S) - \alpha_w(G - K)$ and redefine $w(z) := w(z) - y_K$ for every vertex in K.
- Repeat until $w(v) = 0$ and then proceed with another vertex of S.

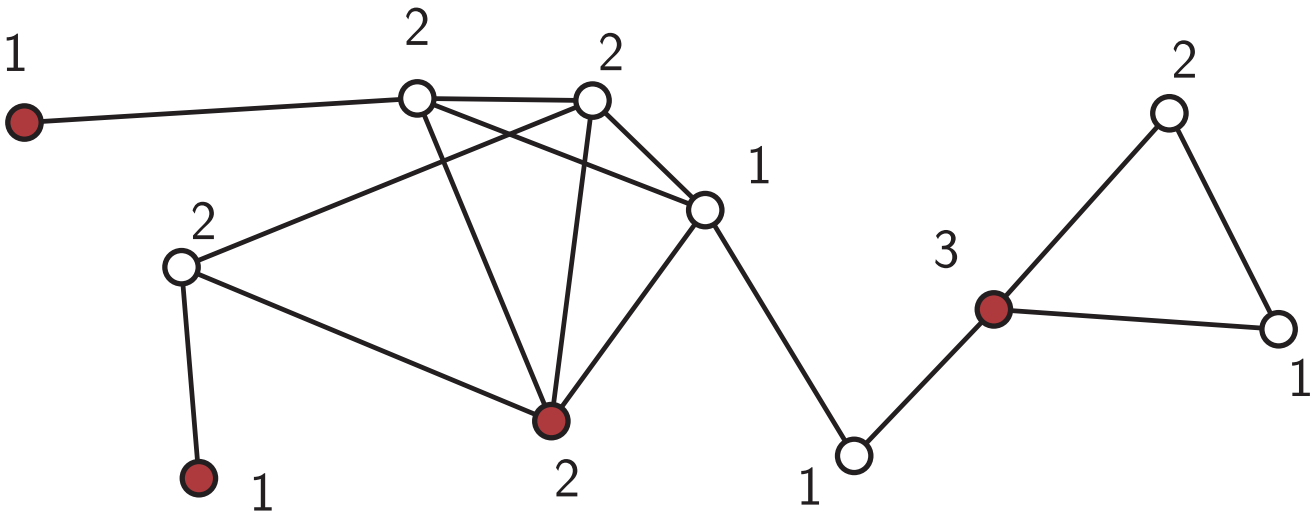
Outline of Hsu and Nemhauser's algorithm

- Compute a MWSS S of G.
- Fix a vertex $v \in S$
  - For each vertex $z \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z\}$; if $w(S') = w(S)$, mark the vertex z.
  - for each nonedge $zt \in N(v)$, compute the MWSS S' such that $S' \cap N[v] = \{z, t\}$; if $w(S') = w(S)$, mark the nonedge zt.
- Compute a clique K containing v, all the marked vertices in $N(v)$, and one endpoint of each marked nonedge in $N(v)$: this will be a crucial clique.
- Compute the weight $y_K = w(S) - \alpha_w(G - K)$ and redefine $w(z) := w(z) - y_K$ for every vertex in K.
- Repeat until $w(v) = 0$ and then proceed with another vertex of S.

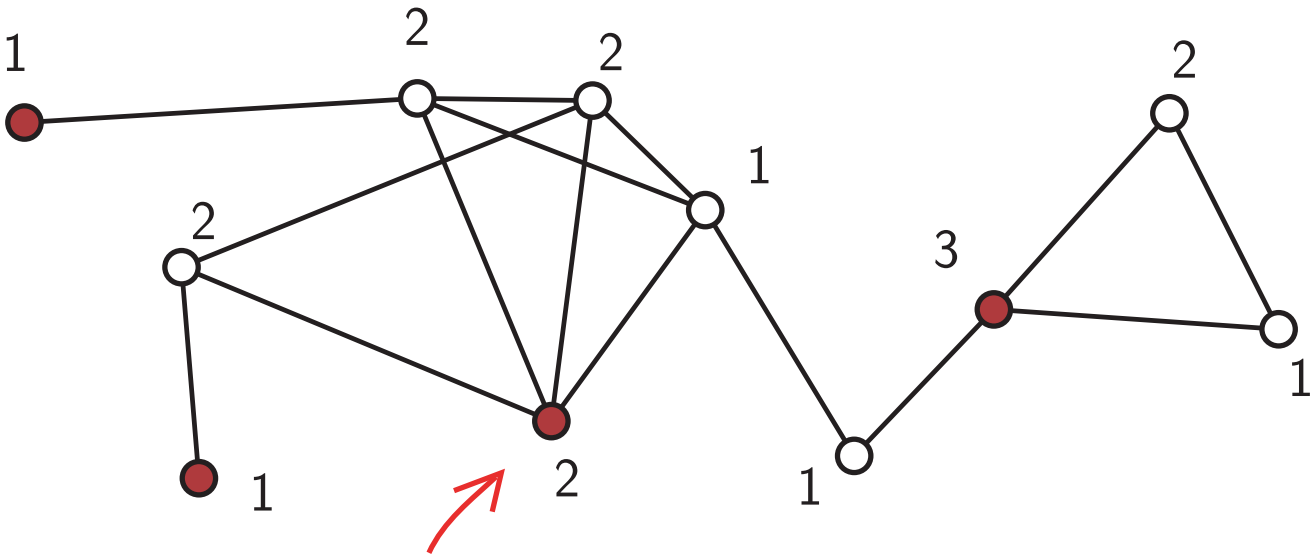
Outline of Hsu and Nemhauser's algorithm



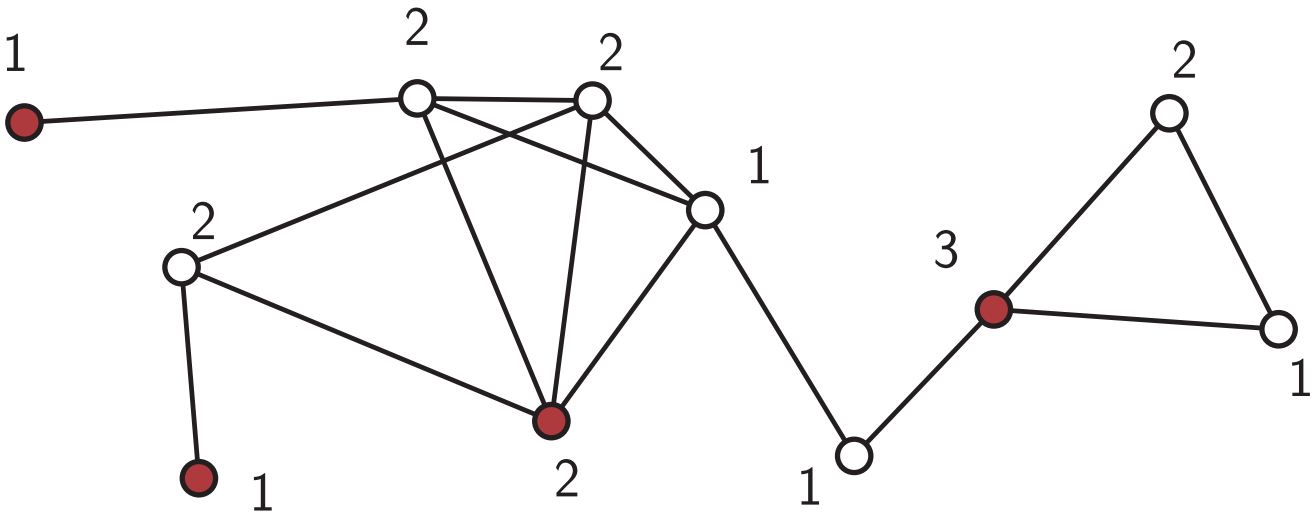
Outline of Hsu and Nemhauser's algorithm



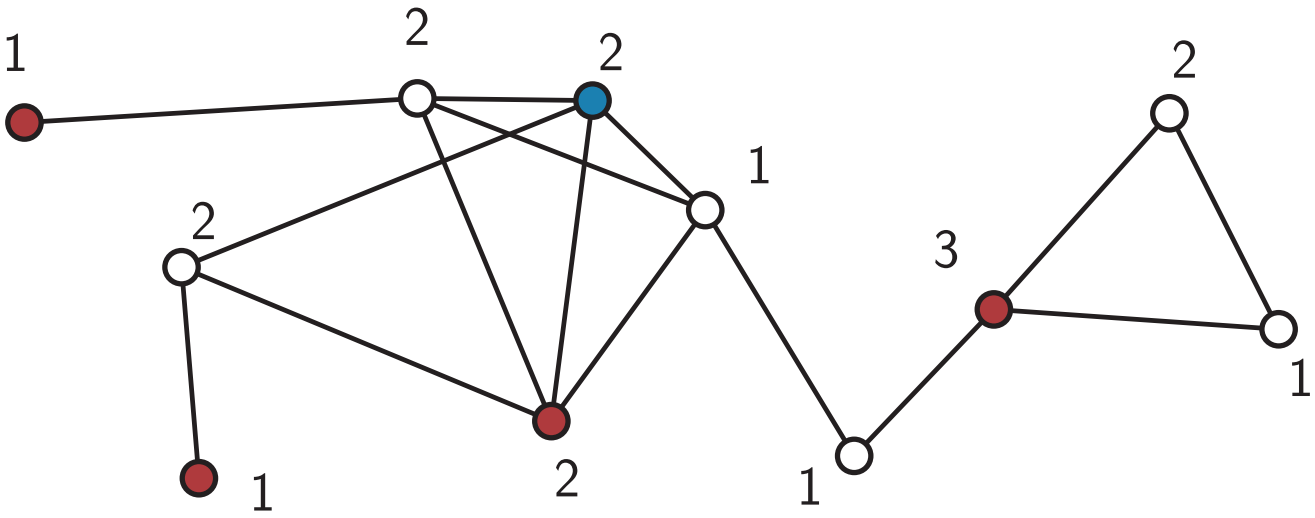
Outline of Hsu and Nemhauser's algorithm



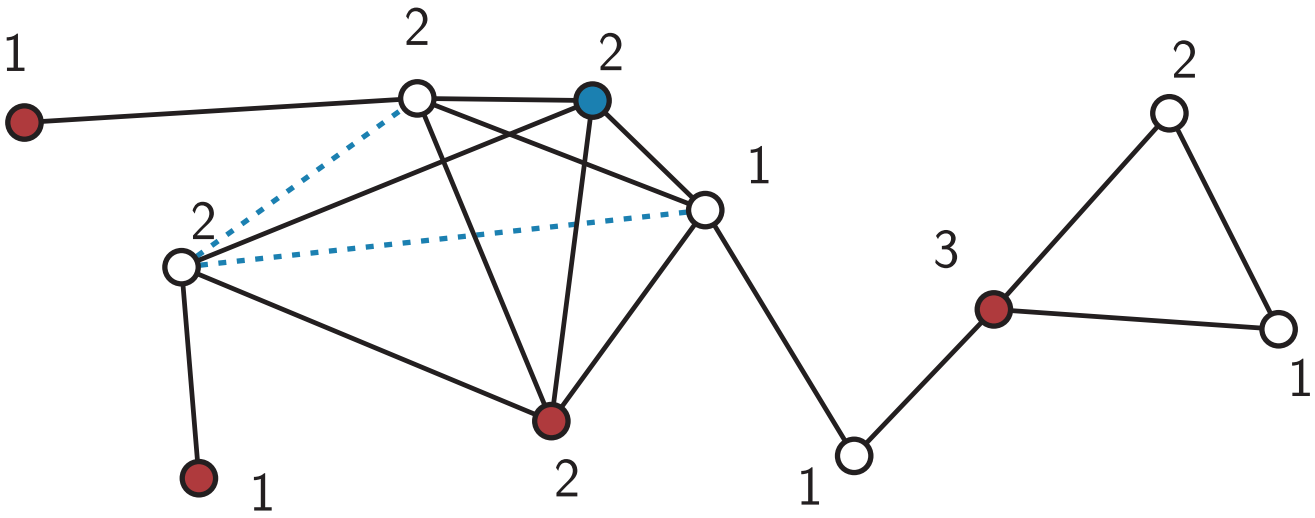
Outline of Hsu and Nemhauser's algorithm



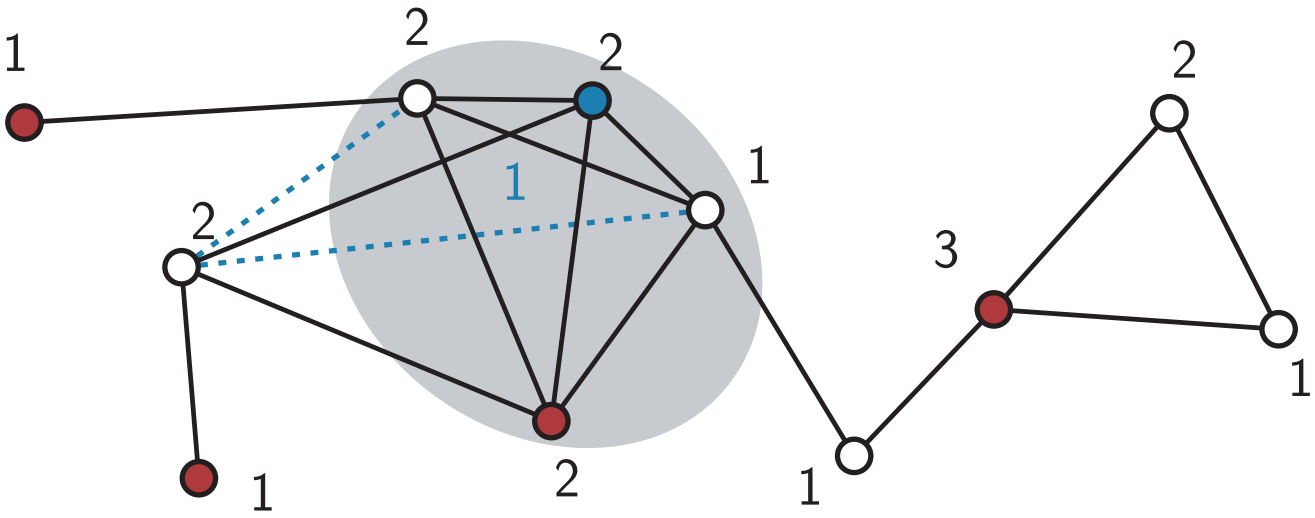
Outline of Hsu and Nemhauser's algorithm



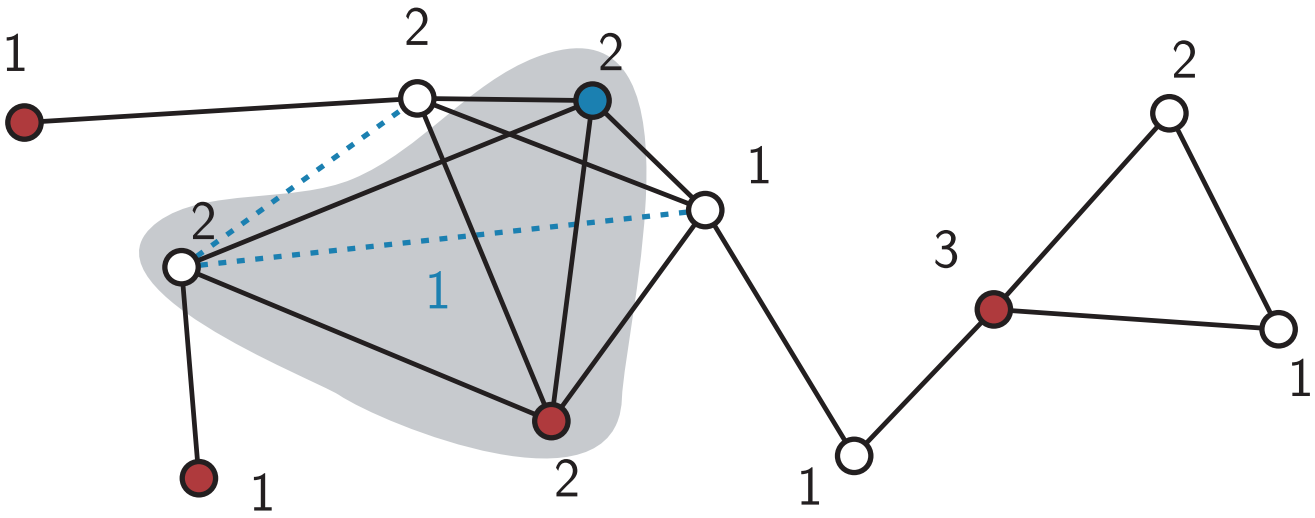
Outline of Hsu and Nemhauser's algorithm



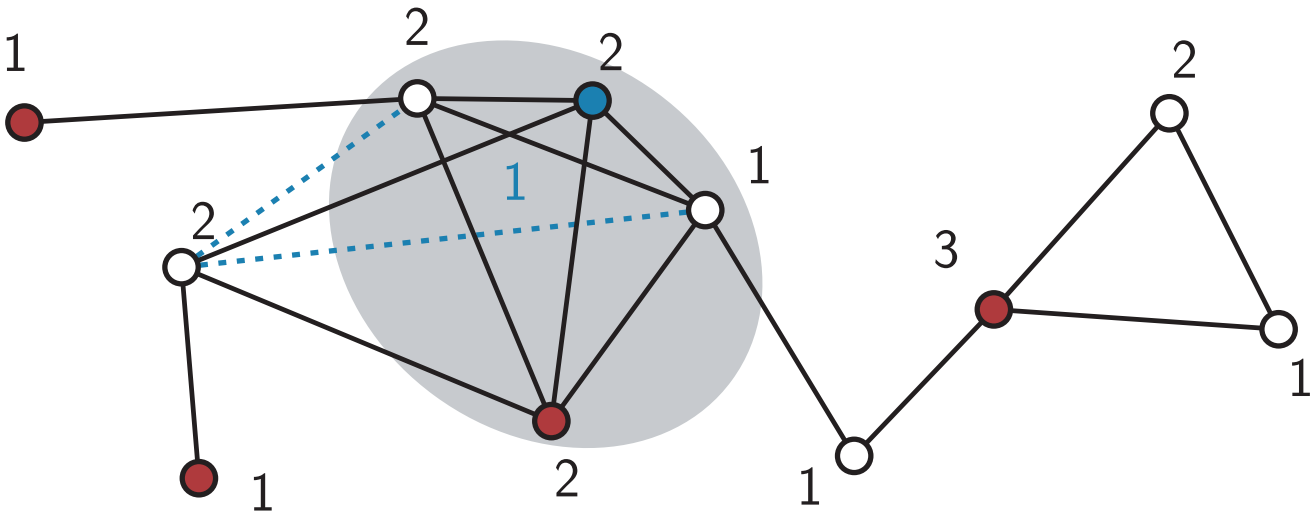
Outline of Hsu and Nemhauser's algorithm



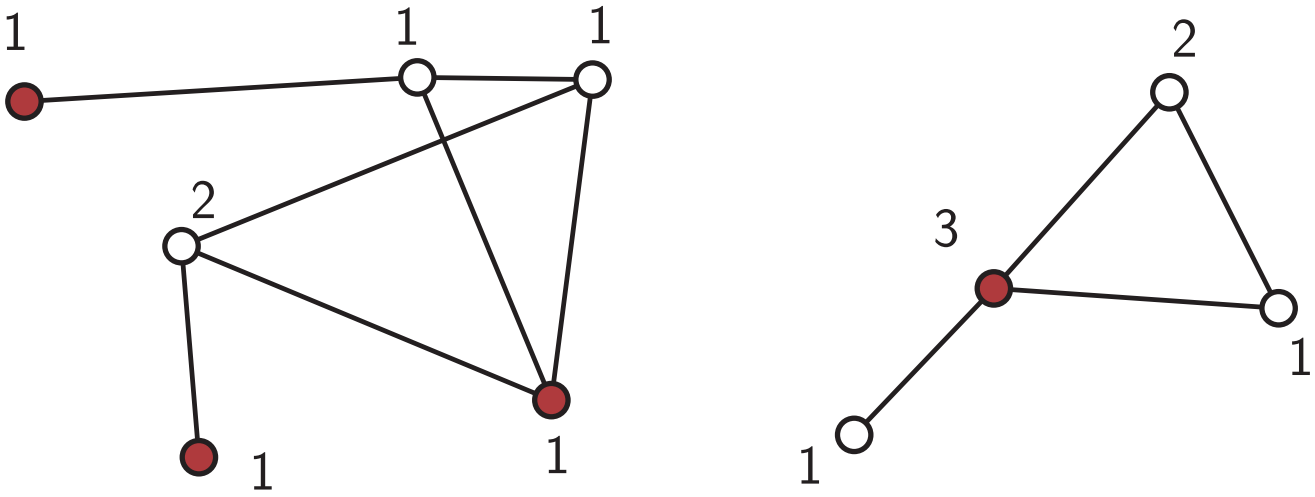
Outline of Hsu and Nemhauser's algorithm



Outline of Hsu and Nemhauser's algorithm



Outline of Hsu and Nemhauser's algorithm



Algorithms based on decompositions

In 1982, Whitesides describes a combinatorial algorithm for MWSS on graphs that can be decomposed by clique cutsets into pieces in which one can solve MWSS in a combinatorial way.

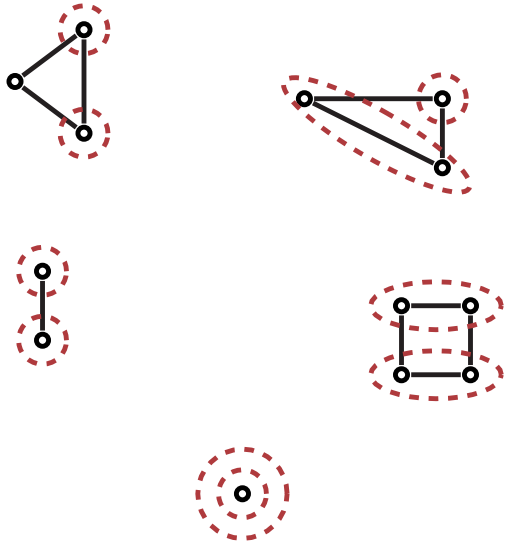
She also describes such an algorithm for MWCC on perfect graphs that can be decomposed by clique cutsets into pieces in which one can solve MWCC in a combinatorial way.

The ideas are not far from those used by Faenza et. al for MWSS on strip composed graphs and those used by B., Oriolo and Snels for MWCC on strip composed perfect graphs.

Nevertheless, the case of clique cutsets is simpler because of its tree-like structure.

Strip composition

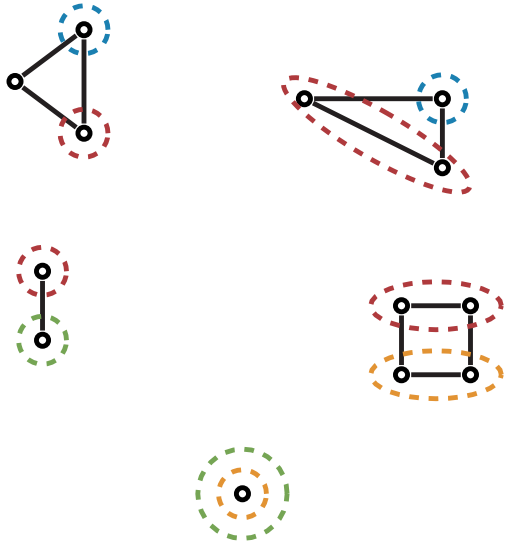
- A graph G is strip composed if G is a composition of some set of strips w.r.t. some partition $\mathcal{P}$.
- A strip $H = (G, \mathcal{A})$ is a graph G (not necessarily connected) with a multi-family $\mathcal{A}$ of either one or two designated non-empty cliques of G. The cliques in $\mathcal{A}$ are called the extremities of H.



- This generalizes line graphs, where each strip is a single vertex.

Strip composition

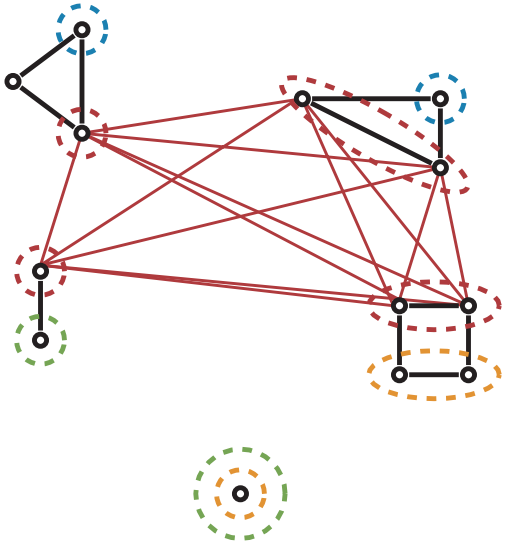
- A graph G is strip composed if G is a composition of some set of strips w.r.t. some partition $\mathcal{P}$.
- A strip $H = (G, \mathcal{A})$ is a graph G (not necessarily connected) with a multi-family $\mathcal{A}$ of either one or two designated non-empty cliques of G. The cliques in $\mathcal{A}$ are called the extremities of H.



- This generalizes line graphs, where each strip is a single vertex.

Strip composition

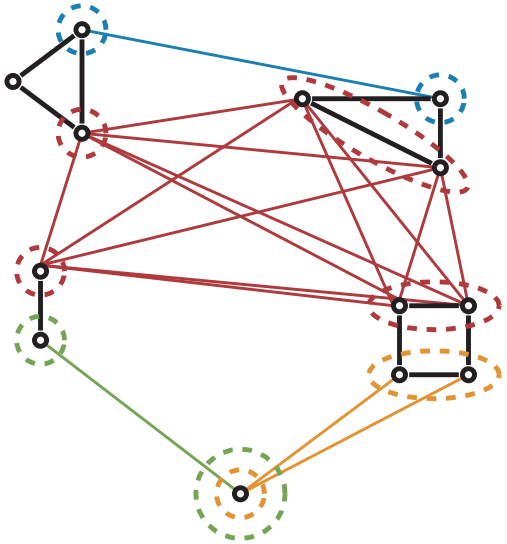
- A graph G is strip composed if G is a composition of some set of strips w.r.t. some partition $\mathcal{P}$.
- A strip $H = (G, \mathcal{A})$ is a graph G (not necessarily connected) with a multi-family $\mathcal{A}$ of either one or two designated non-empty cliques of G. The cliques in $\mathcal{A}$ are called the extremities of H.



- This generalizes line graphs, where each strip is a single vertex.

Strip composition

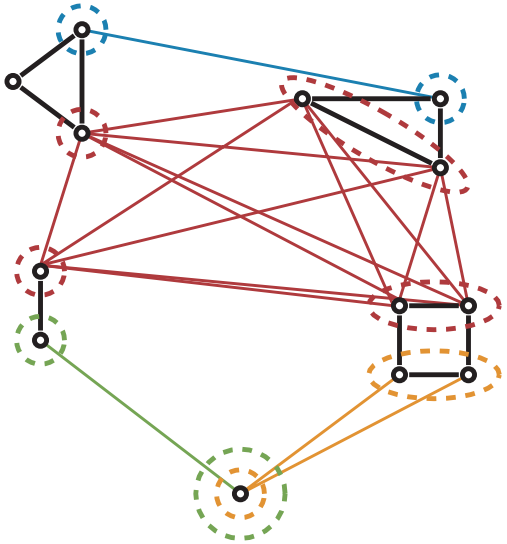
- A graph G is strip composed if G is a composition of some set of strips w.r.t. some partition $\mathcal{P}$.
- A strip $H = (G, \mathcal{A})$ is a graph G (not necessarily connected) with a multi-family $\mathcal{A}$ of either one or two designated non-empty cliques of G. The cliques in $\mathcal{A}$ are called the extremities of H.



- This generalizes line graphs, where each strip is a single vertex.

Strip composition

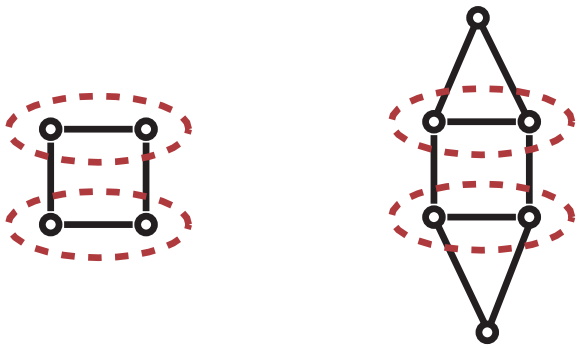
- A graph G is strip composed if G is a composition of some set of strips w.r.t. some partition $\mathcal{P}$.
- A strip $H = (G, \mathcal{A})$ is a graph G (not necessarily connected) with a multi-family $\mathcal{A}$ of either one or two designated non-empty cliques of G. The cliques in $\mathcal{A}$ are called the extremities of H.



- This generalizes line graphs, where each strip is a single vertex.

Strip composition

- Each class of the partition of the extremities defines a clique of the composed graph, and is called a partition-clique.
- A strip satisfies property Π when the graph obtained by adding, for each extremity, a vertex complete to it, satisfies property Π.



- The composition of line (claw-free, quasi-line) strips is a line (claw-free, quasi-line) graph.
- The composition of perfect strips is not necessary a perfect graph.

Strip decomposition theorems for claw-free graphs

Existence Theorem (Chudnovsky and Seymour, 2005)

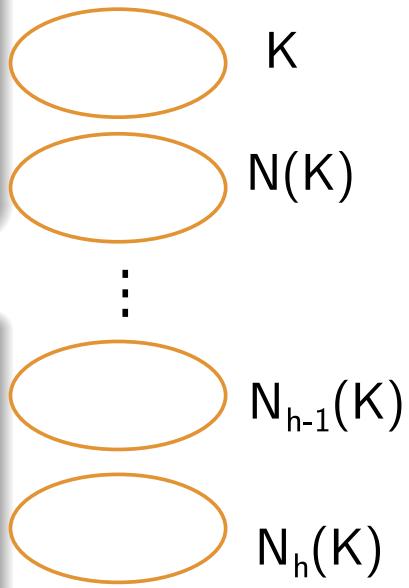
G claw-free, $\alpha(G) \geq 4$:

1. either G is the composition of fuzzy linear interval strips
2. or G is a fuzzy circular interval graph.

Algorithmic Theorem (Faenza, Oriolo and Stauffer, 2010)

G claw-free, $\alpha(G) \geq 4$:

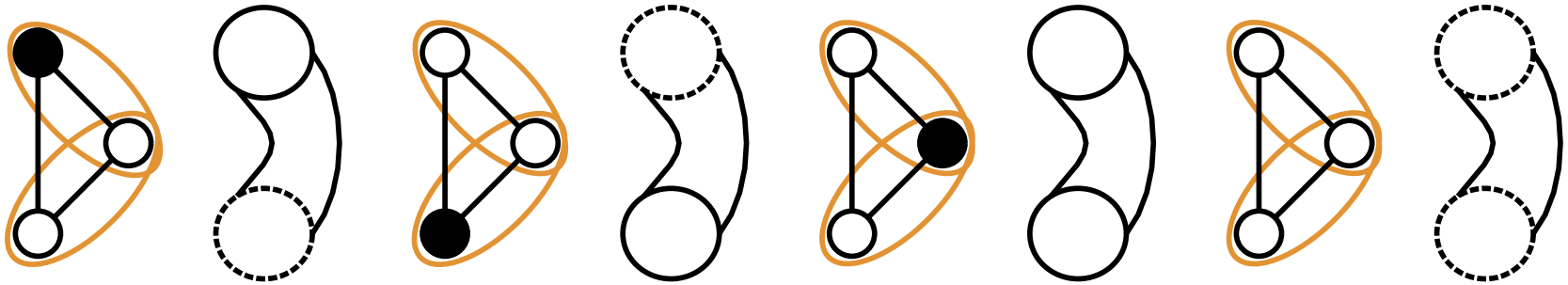
1. either G is the composition of distance simplicial strips
2. or G is distance claw-free and net-free.



MWSS of strip composed graphs (Faenza, Oriolo and Stauffer)

- Replace every strip with a weighted simple line strip (a gadget), and obtain a line graph G' as the composition of the gadgets, such that $\alpha_w(G) = f(\alpha_w(G'))$ and we know f.
- In order to find a MWSS of G', find a maximum weighted matching in the root graph of G'.
- The vertices in the MWSS S of G' represent a guideline to choose a suitable stable set in each strip that forms a MWSS of G.

They use it for the strip decomposable claw-free graphs and solve the case of distance claw-free graphs separately.

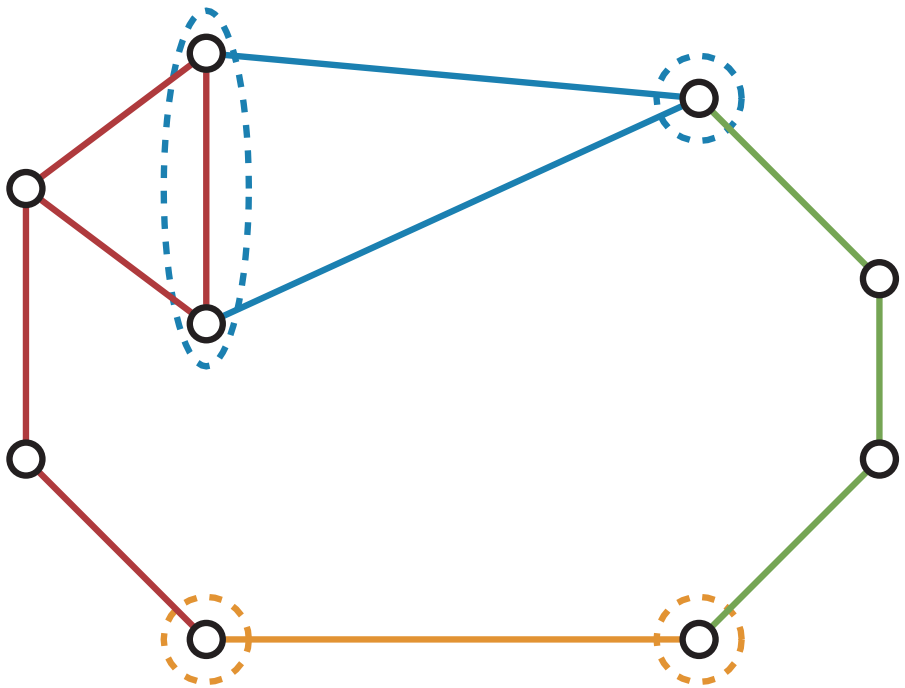


Extending it to the MWCC on strip composed perfect graphs

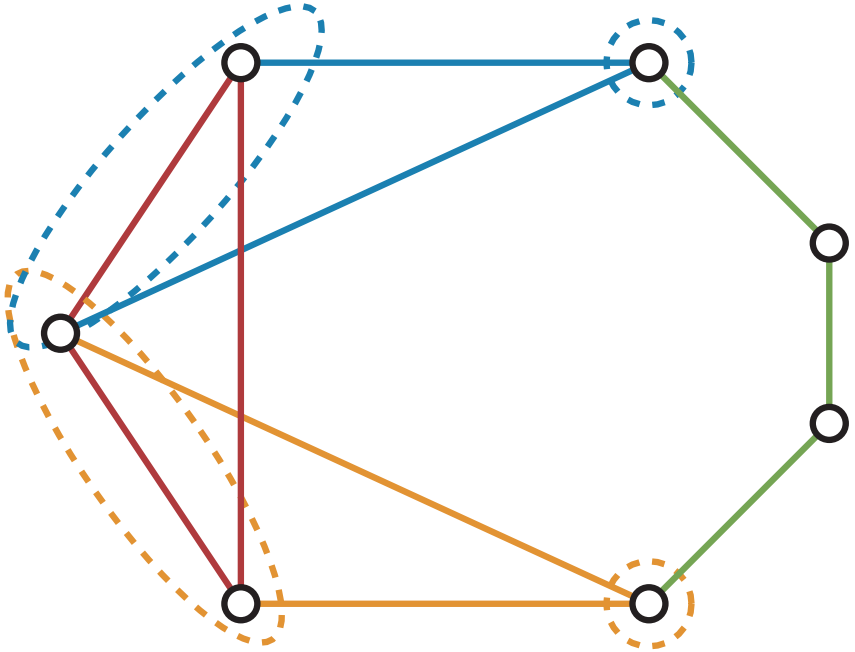
We borrow the MWSS idea, but...

- The original gadgets do not preserve perfection. We introduced four different gadgets depending on the parity of the strips (non-trivial theorems to prove the reduction and the perfection).
- Even knowing how to solve MWCC on strips (in the claw-free case they are distance simplicial) and on the line graph G', sometimes it is not trivial to deduce a MWCC of G from one of G'. Some cliques of G' do not translate straightforward into a clique of G. We have to deal with seven different cases.

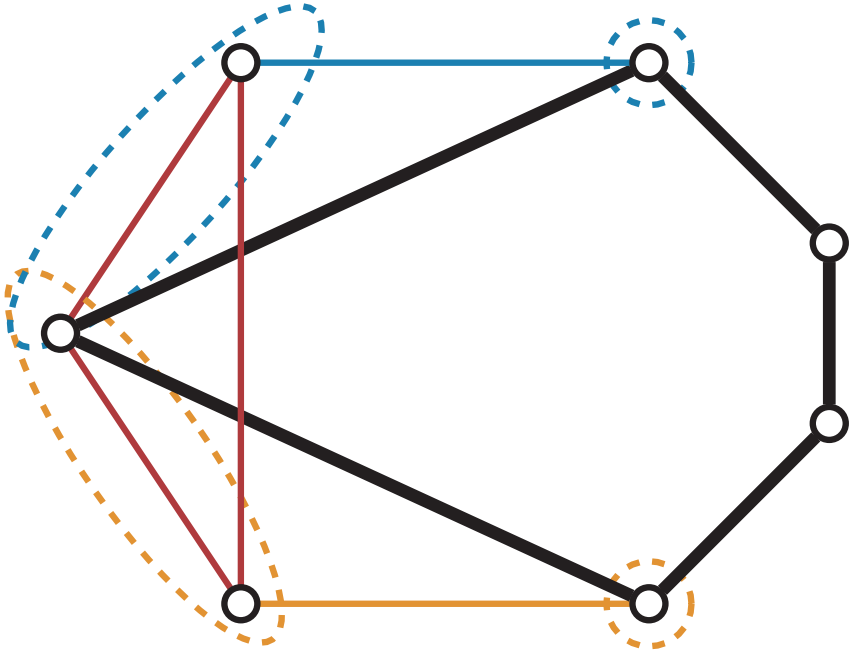
The perfection problem



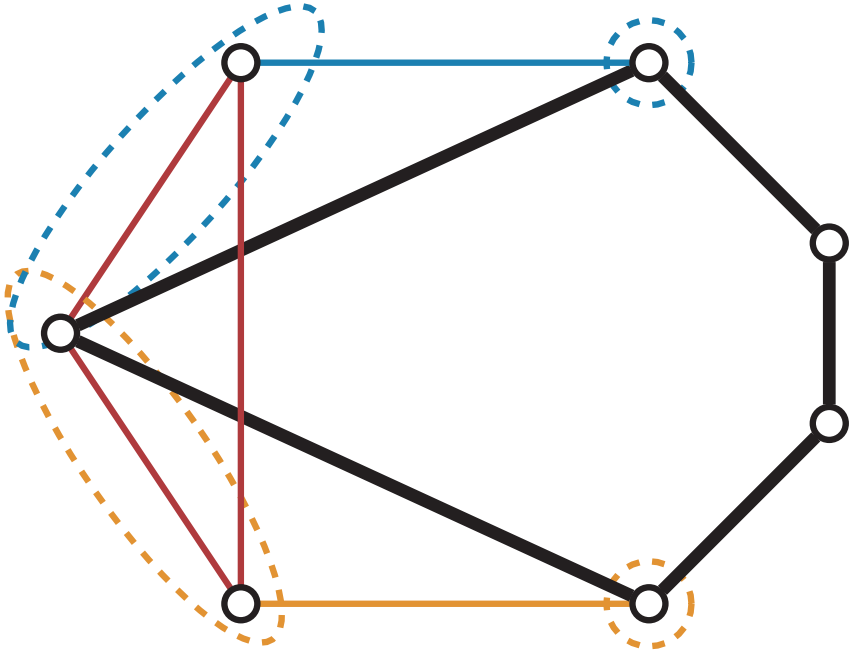
The perfection problem



The perfection problem



The perfection problem



New gadget strips (still line strips)

MWCC on strip composed perfect graphs

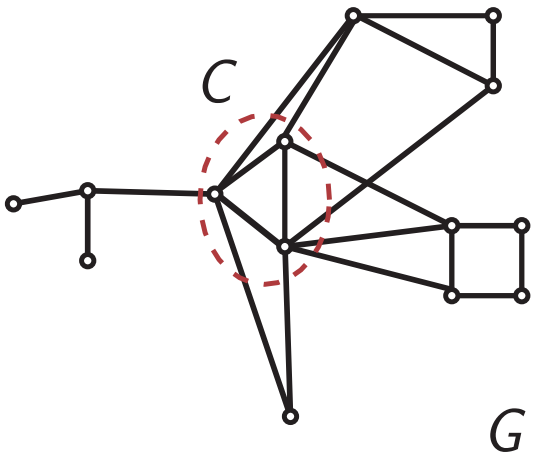
The outline of the algorithm is the following:

- Find a strip decomposition of the graph (Faenza et al.)
- Compute the values of an MWCC on four suitable induced subgraphs of each strip.
- Replace in the composition each strip by a weighted gadget (the weight of the vertices will be a function of the values computed in 2., and the gadgets will depend on some parity issues in order to preserve perfection)
- Obtain a weighted perfect line graph, and solve the MWCC using, for instance, the primal-dual algorithm for maximum weight matching by Gabow (1990).
- Using this clique cover and the ability of computing a MWCC on a strip, reconstruct a MWCC of the original graph (analyzing the different cases).

What if the graph is not strip composed? Get back to clique cutsets...

A set C is a clique cutset of a graph $G[C]$ is complete and $G[V(G) \setminus C]$ has more connected components than G.

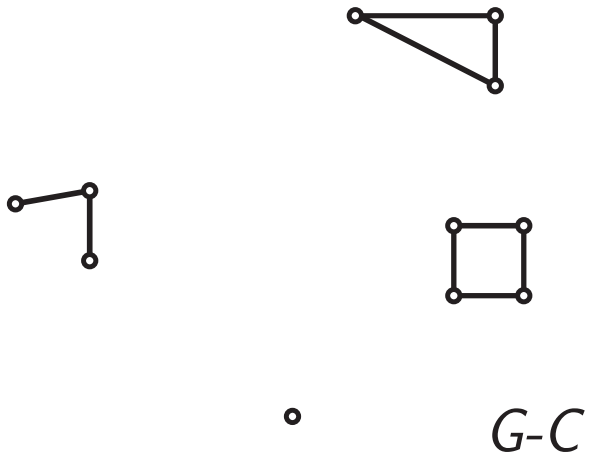
If $G_1, \ldots, G_k$ are the connected components of $G[V(G) \setminus C]$, then C decomposes G into the graphs $G[G_1 \cup C], \ldots, G[G_k \cup C]$ (not disjoint, they all have C in common).



What if the graph is not strip composed? Get back to clique cutsets...

A set C is a clique cutset of a graph $G[C]$ is complete and $G[V(G) \setminus C]$ has more connected components than G.

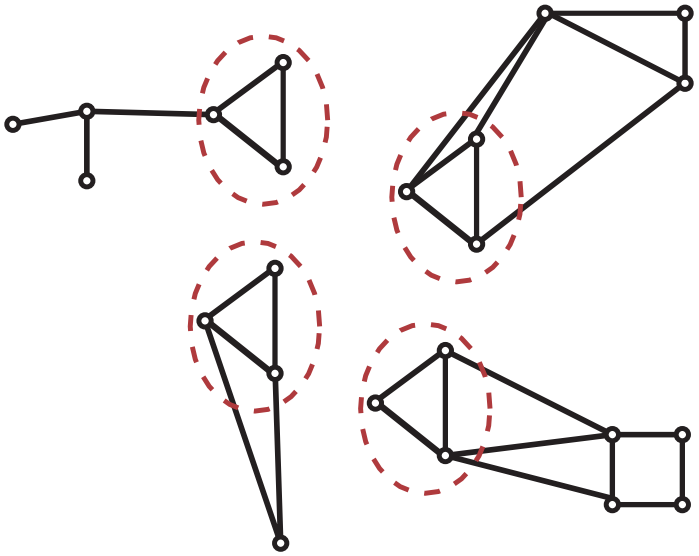
If $G_1, \ldots, G_k$ are the connected components of $G[V(G) \setminus C]$, then C decomposes G into the graphs $G[G_1 \cup C], \ldots, G[G_k \cup C]$ (not disjoint, they all have C in common).



What if the graph is not strip composed? Get back to clique cutsets...

A set C is a clique cutset of a graph $G[C]$ is complete and $G[V(G) \setminus C]$ has more connected components than G.

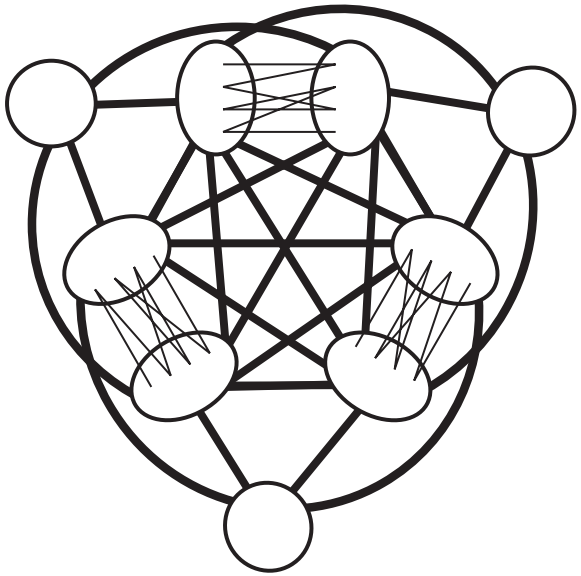
If $G_1, \ldots, G_k$ are the connected components of $G[V(G) \setminus C]$, then C decomposes G into the graphs $G[G_1 \cup C], \ldots, G[G_k \cup C]$ (not disjoint, they all have C in common).



Claw-free perfect graphs decomposition by clique cutsets

Chvátal and Sbihi in 1988 proved that a claw-free perfect graphs can be decomposed via clique cutsets into peculiar and elementary graphs.

Peculiar graphs have a very simple structure, and the MWCC problem can be solved on them in a similar fashion than on distance simplicial graphs (i.e., iteratively computing crucial cliques).



Claw-free perfect graphs decomposition by clique cutsets

Maffray and Reed in 1999 showed that elementary graphs are indeed strip composed: they arise by replacing some particular edges of the line graph of a bipartite graph by the complement of a bipartite graph.

They in fact show how to solve the unweighted MCC on that class (the reduction is similar to ours, in the very particular case in which all the cliques involved in a MCC are partition cliques).

So, using the algorithm by Whitesides together with this results and our approach for the weighted case on elementary graphs, leads to an $O(n^4 \log(n))$ algorithm for MWCC on claw-free perfect graphs.

But... why not to apply it just to the non-strip-composed case?

Claw-free perfect graphs decomposition by clique cutsets

Maffray and Reed in 1999 showed that elementary graphs are indeed strip composed: they arise by replacing some particular edges of the line graph of a bipartite graph by the complement of a bipartite graph.

They in fact show how to solve the unweighted MCC on that class (the reduction is similar to ours, in the very particular case in which all the cliques involved in a MCC are partition cliques).

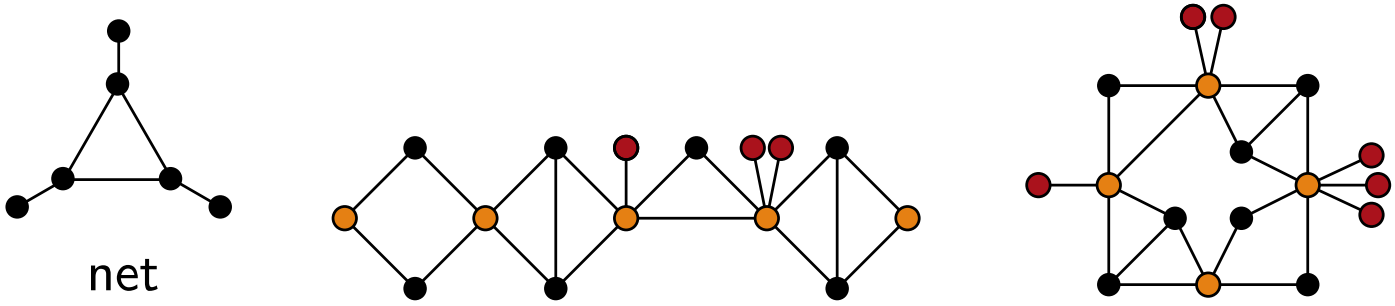
So, using the algorithm by Whitesides together with this results and our approach for the weighted case on elementary graphs, leads to an $O(n^4 \log(n))$ algorithm for MWCC on claw-free perfect graphs.

But... why not to apply it just to the non-strip-composed case?

{claw,net}-free line graphs

Brandstädt and Dragan in 2003 characterized {claw,net}-free graphs, and this structure was used by Faenza et al. to deal in the MWSS problem with the case of non-strip composed claw-free graphs, but we could not adapt these ideas to the MWCC.

M. Safe in 2011 gave a more detailed characterization (but with the same flavour) for the case of {claw,net}-free line graphs. Namely, they are mainly line graphs of linear or circular concatenations of certain small graphs.

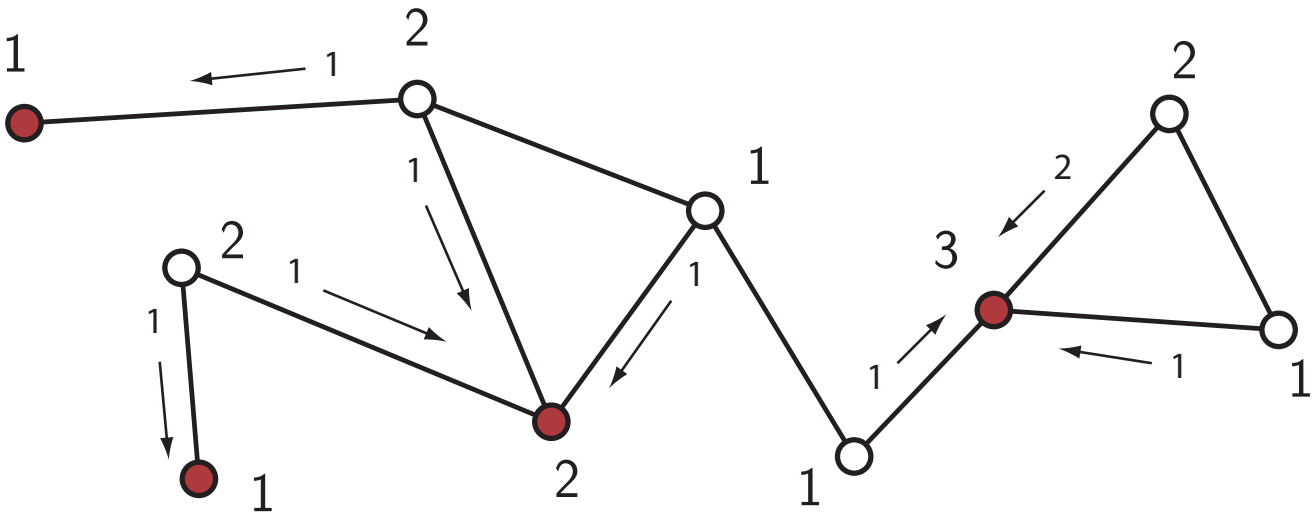


{claw,net}-free line graphs

By using this last characterization restricted to the extra-conditions given by the elementary graphs description, we can improve the complexity of the clique-cutset based algorithm to $O(n^3)$ for the case of {claw,net}-free perfect graphs, so we cover our remaining case.

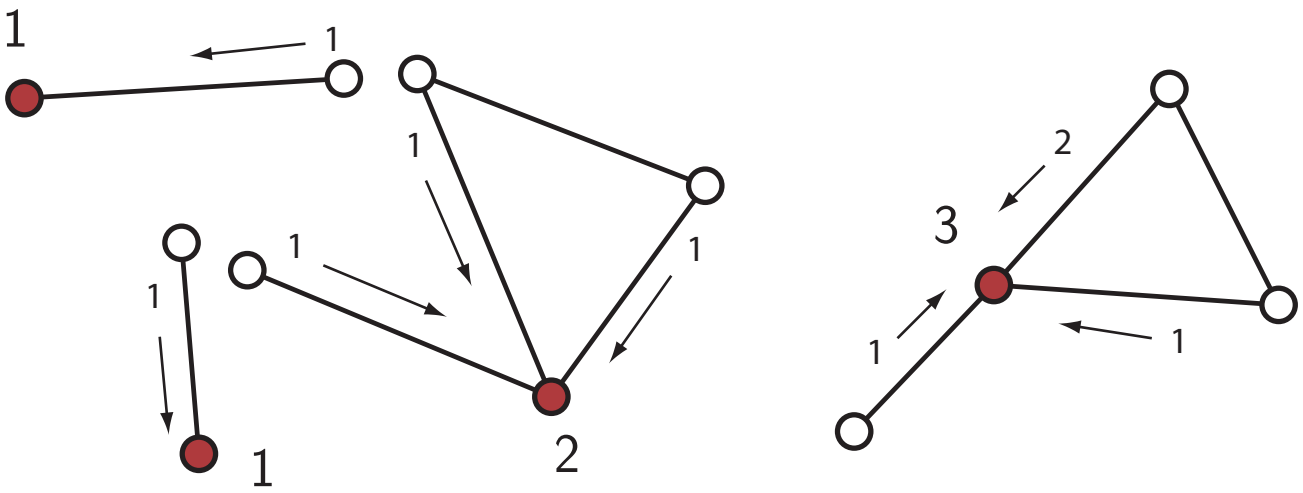
Outline of our last algorithm

First compute a MWSS S of G. Then compute all at once, for each vertex $z \in V \setminus S$ and each neighbor $v \in S$ of z, how much of the weight of z will be covered with cliques containing v.



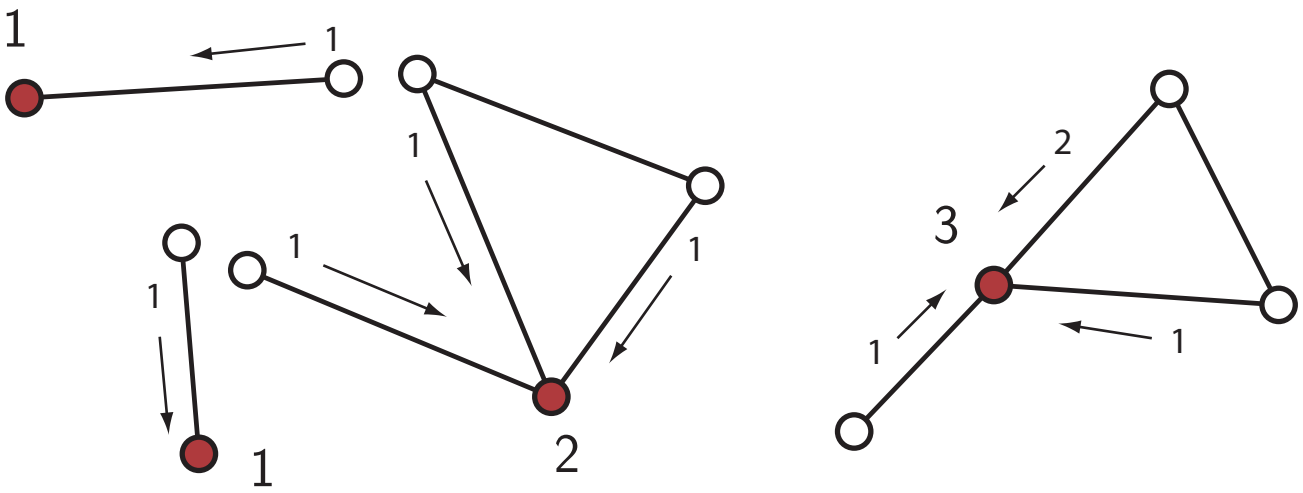
Outline of our last algorithm

First compute a MWSS S of G. Then compute all at once, for each vertex $z \in V \setminus S$ and each neighbor $v \in S$ of z, how much of the weight of z will be covered with cliques containing v. Then we solve a MWCC problem on $N[v]$ for each $v \in S$, using the weights defined above.



Outline of our last algorithm

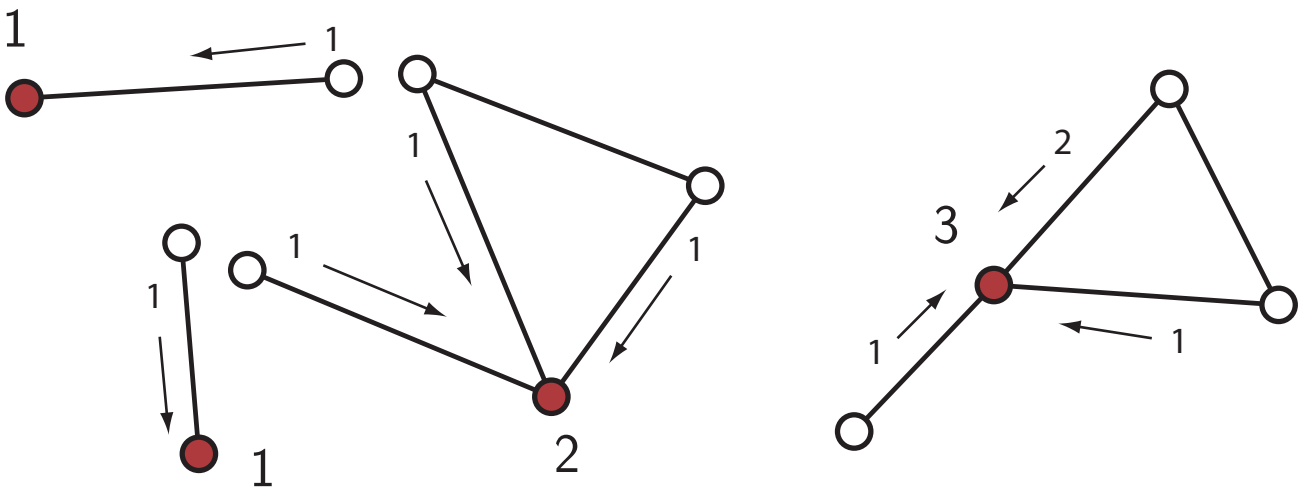
First compute a MWSS S of G. Then compute all at once, for each vertex $z \in V \setminus S$ and each neighbor $v \in S$ of z, how much of the weight of z will be covered with cliques containing v. Then we solve a MWCC problem on $N[v]$ for each $v \in S$, using the weights defined above.



Note that if G is claw-free, each vertex in $V \setminus S$ has at most two neighbors in S.

Outline of our last algorithm

First compute a MWSS S of G. Then compute all at once, for each vertex $z \in V \setminus S$ and each neighbor $v \in S$ of z, how much of the weight of z will be covered with cliques containing v. Then we solve a MWCC problem on $N[v]$ for each $v \in S$, using the weights defined above.



Note that if G is claw-free, each vertex in $V \setminus S$ has at most two neighbors in S. Also, if G is claw-free perfect, the graph induced by $N[v]$ is the complement of a bipartite graph.

Clique cover in perfect graphs: Reformulation

- All cliques intersect S so we can aggregate the cliques taking $s \in S$
  $\rightarrow y_{vs}$: covering of v by cliques picking s

$$\sum_{s \in N(v) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- It's not obvious how to recover y_K efficiently in general.

Clique cover in perfect graphs: Reformulation

- All cliques intersect S so we can aggregate the cliques taking $s \in S$
  $\rightarrow y_{vs}$: covering of v by cliques picking s

$$\sum_{s \in N(v) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad \textit{integer}$$

  $\Rightarrow$ a feasible clique cover y_K, $K \in \mathcal{K}(G)$ yields a feasible solution to the new reformulation with $y_{vs} := \sum_{K \in \mathcal{K}(G): v, s \in K} y_K$

- It's not obvious how to recover y_K efficiently in general.

Clique cover in perfect graphs: Reformulation

- All cliques intersect S so we can aggregate the cliques taking $s \in S$
  $\rightarrow y_{vs}$: covering of v by cliques picking s

$$\sum_{s \in N(v) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

  $\Rightarrow$ a feasible clique cover y_K, $K \in \mathcal{K}(G)$ yields a feasible solution to the new reformulation with $y_{vs} := \sum_{K \in \mathcal{K}(G): v, s \in K} y_K$
  $\Leftarrow$ vice-versa, the second conditions imply that $\{s\}$ is a maximum weighted stable set of $G[s \cup N(s)]$ with weight $w'_v = y_{vs}$ for all $v \in N(s)$. But because the graph $G[s \cup N(s)]$ is perfect, there is a clique cover of same weight $w(s)$. We take the union of all those 'local' clique covers.

- It's not obvious how to recover y_K efficiently in general.

Clique cover in perfect graphs: Reformulation

- All cliques intersect S so we can aggregate the cliques taking $s \in S$
  $\rightarrow y_{vs}$: covering of v by cliques picking s

$$\sum_{s \in N(v) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad \textit{integer}$$

  $\Rightarrow$ a feasible clique cover y_K, $K \in \mathcal{K}(G)$ yields a feasible solution to the new reformulation with $y_{vs} := \sum_{K \in \mathcal{K}(G): v, s \in K} y_K$
  $\Leftarrow$ vice-versa, the second conditions imply that $\{s\}$ is a maximum weighted stable set of $G[s \cup N(s)]$ with weight $w'_v = y_{vs}$ for all $v \in N(s)$. But because the graph $G[s \cup N(s)]$ is perfect, there is a clique cover of same weight $w(s)$. We take the union of all those 'local' clique covers.

- It's not obvious how to recover y_K efficiently in general.

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - Non-zero coefficients can be thought as edges
  - Each inequality is a clique
  - Each variable is a node
  - Shortest path algorithm
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - In an arbitrary graph in $O(n^2 + m)$ time.
  - In bipartite graphs in $O(n + m)$ time.
  - In co-bipartite graphs in $O(n + m)$ time.
  - In any distance hereditary graph in $O(n + m)$ time.
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S \qquad \bigvee_{s \in N(u) \cap S} y_{vs}$$

$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s) \qquad \bigvee_{v \in T} \neg y_{vs}$$

$$y \quad \textit{integer} \quad y_{vs} \in \{0, 1\}$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - Schrijver 1991: adapting Fourier-Motzkin ($O(n^3)$)
  - Lauer, Sherman, Sweedyk, Ullman, Lee 1995: $O(n^2 \log n)$
  - Hochbaum, Naor 1994: $O(n^2)$
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - Schrijver 1991: adapting Fourier-Motzkin ($O(n^3)$)
  - Jaffar, Maher, Stuckey and Yap 1994: shortest path and rounding ($O(n^2 \log n + nm)$)
  - Peis 2007: shortest path + 2-SAT
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - Schrijver 1991: adapting Fourier-Motzkin ($O(n^3)$)
  - Jaffar, Maher, Stuckey and Yap 1994: shortest path and rounding ($O(n^2 \log n + nm)$)
  - Peis 2007: shortest path + 2-SAT
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - Schrijver 1991: adapting Fourier-Motzkin ($O(n^3)$)
  - Jaffar, Maher, Stuckey and Yap 1994: shortest path and rounding ($O(n^2 \log n + nm)$)
  - Peis 2007: shortest path + 2-SAT
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - Schrijver 1991: adapting Fourier-Motzkin ($O(n^3)$)
  - Jaffar, Maher, Stuckey and Yap 1994: shortest path and rounding ($O(n^2 \log n + nm)$)
  - Peis 2007: shortest path + 2-SAT
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - Schrijver 1991: adapting Fourier-Motzkin ($O(n^3)$)
  - Jaffar, Maher, Stuckey and Yap 1994: shortest path and rounding ($O(n^2 \log n + nm)$)
  - Peis 2007: shortest path + 2-SAT
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Clique cover in claw-free perfect graphs: Reformulation

- Reformulation for claw-free perfect graphs.
- Each inequality has at most two variables.

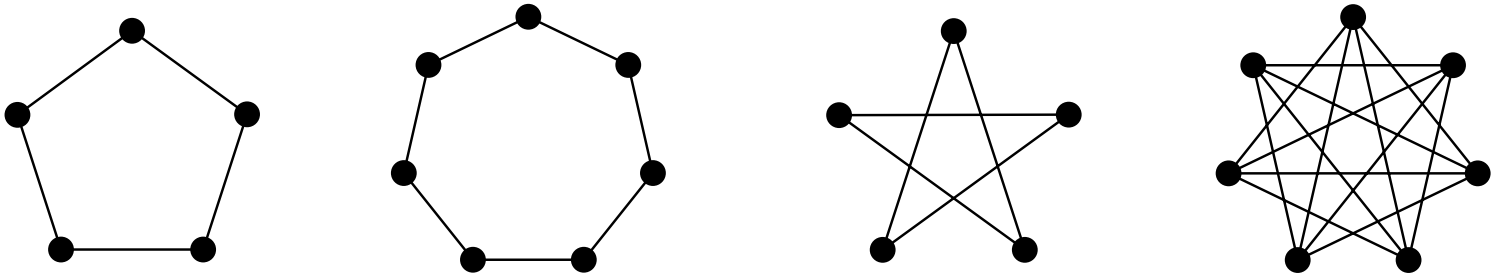
$$\sum_{s \in N(u) \cap S} y_{vs} \geq w(v), \quad \forall v \in V \setminus S$$
$$\sum_{v \in T} y_{vs} \leq w(s), \quad \forall s \in S, \quad \forall T \text{ stable set of } N(s)$$
$$y \quad integer$$

- For the unweighted case is just 2-sat and one gets directly the cliques!
- Feasible integer solutions of systems where each inequality has at most two variables can be solved combinatorially in polynomial time:
  - Schrijver 1991: adapting Fourier-Motzkin ($O(n^3)$)
  - Jaffar, Maher, Stuckey and Yap 1994: shortest path and rounding ($O(n^2 \log n + nm)$)
  - Peis 2007: shortest path + 2-SAT
- For this particular case, shortest paths give upper and lower bounds and one can choose upper or lower by a 2-SAT instance equivalent to the unweighted MCC formulation in the graph where free vertices where removed.
- Recovering the y_K variables is easy (locally the graph is co-bipartite, so distance simplicial)

Further results

For the unweighted case, if one starts in an arbitrary claw-free graph G with a maximal (not necessarily maximum) stable set S, from the digraph associated with the 2-SAT instance (Aspvall, Plass and Tarjan algorithm, 1979), one can obtain either:

- A clique covering of G with the same cardinality of S (thus S was maximum), or
- an augmenting path for S, so we can iterate, or
- an odd hole or an odd antihole (so G is not perfect).



Combinatorial algorithms for MCC and MWCC on claw-free perfect graphs

- Hsu and Nemhauser 1981, 1982: building upon a solution of several instances of M(W)SS in order to find crucial cliques ($O(n^5)$).
- Combination of results by Whitesides 1982, Chvátal and Sbihi 1988 and Maffray and Reed 1999 on clique cutsets and claw-free perfect graphs ($O(n^4 \log(n))$, only for the unweighted case).
- B., Oriolo and Snels 2012: based on strip decomposition for combined with clique cutsets decomposition ($O(n^3)$).
- B, Oriolo, Snels and Stauffer 2013: building upon a reformulation and 'nice' polyhedra, solved by 2-sat and shortest paths ($O(n^3)$).