Trivially perfect graphs

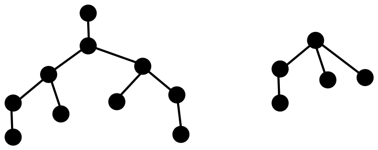
- Based on a joint work with
  Pål Grønås Drange, Fedor V. Fomin, and Yngve Villanger

Trivially perfect graphs

- Based on a joint work with
  Pål Grønås Drange, Fedor V. Fomin, and Yngve Villanger
- A graph H is *trivially perfect* if it is a transitive closure of some rooted forest.

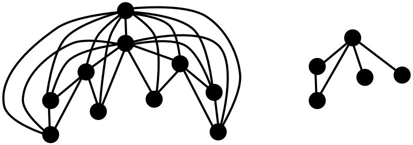
Trivially perfect graphs

- Based on a joint work with
  Pål Grønås Drange, Fedor V. Fomin, and Yngve Villanger
- A graph H is *trivially perfect* if it is a transitive closure of some rooted forest.



Trivially perfect graphs

- Based on a joint work with
  Pål Grønås Drange, Fedor V. Fomin, and Yngve Villanger
- A graph H is *trivially perfect* if it is a transitive closure of some rooted forest.



Trivially perfect graphs: alternative definitions

- **Recursive definition**: A graph H is trivially perfect iff it is K_1 or it can be constructed using the following two operations:

Trivially perfect graphs: alternative definitions

- **Recursive definition**: A graph H is trivially perfect iff it is K_1 or it can be constructed using the following two operations:
  - taking a disjoint union of two trivially perfect graphs, or

Trivially perfect graphs: alternative definitions

- **Recursive definition**: A graph H is trivially perfect iff it is K_1 or it can be constructed using the following two operations:
  - taking a disjoint union of two trivially perfect graphs, or
  - adding a universal vertex to a trivially perfect graph.

Trivially perfect graphs: alternative definitions

- **Recursive definition**: A graph H is trivially perfect iff it is K_1 or it can be constructed using the following two operations:
  - taking a disjoint union of two trivially perfect graphs, or
  - adding a universal vertex to a trivially perfect graph.
- **Forbidden obstacles**: A graph H is trivially perfect iff it does not contain C_4 or P_4 as an induced subgraph.

Trivially perfect graphs: alternative definitions

- **Recursive definition**: A graph H is trivially perfect iff it is K_1 or it can be constructed using the following two operations:
  - taking a disjoint union of two trivially perfect graphs, or
  - adding a universal vertex to a trivially perfect graph.
- **Forbidden obstacles**: A graph H is trivially perfect iff it does not contain C_4 or P_4 as an induced subgraph.
- Threshold $\subseteq$ TriviallyPerfect $\subseteq$ Interval.

Trivially perfect graphs: alternative definitions

- **Recursive definition**: A graph H is trivially perfect iff it is K_1 or it can be constructed using the following two operations:
  - taking a disjoint union of two trivially perfect graphs, or
  - adding a universal vertex to a trivially perfect graph.
- **Forbidden obstacles**: A graph H is trivially perfect iff it does not contain C_4 or P_4 as an induced subgraph.
- $\text{Threshold} \subseteq \text{TriviallyPerfect} \subseteq \text{Interval}$.
- Tight connections with graph parameter *treedepth*.

Trivially perfect graphs: alternative definitions

- **Recursive definition**: A graph H is trivially perfect iff it is K_1 or it can be constructed using the following two operations:
  - taking a disjoint union of two trivially perfect graphs, or
  - adding a universal vertex to a trivially perfect graph.
- **Forbidden obstacles**: A graph H is trivially perfect iff it does not contain C_4 or P_4 as an induced subgraph.
- $\mathrm{Threshold} \subseteq \mathrm{TriviallyPerfect} \subseteq \mathrm{Interval}.$
- Tight connections with graph parameter *treedepth*.
- From now on, all trivially perfect graphs will be connected.

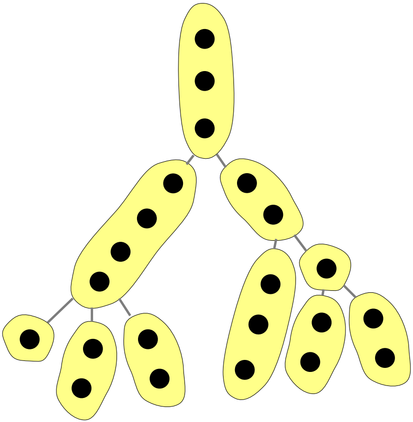
Universal Clique Decomposition

Universal Clique Decomposition

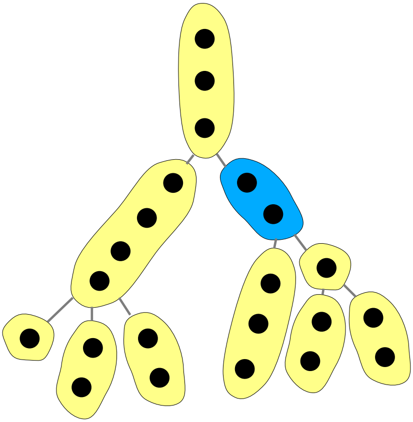
A *universal clique decomposition* of a graph H is a rooted tree $\mathcal{T}$, where every node t is assigned a bag B_t such that:

- $\{B_t \colon t \in V(\mathcal{T})\}$ form a partition of $V(H)$;
- non-leaf nodes of $\mathcal{T}$ have at least two sons;
- $uv \in E(H)$ if and only if u and v share a bag, or their bags are in ancestor-descendant relation.

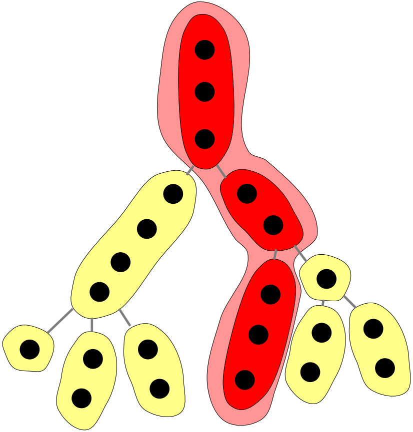
UC-decomposition on a picture



UC-decomposition on a picture



UC-decomposition on a picture



UC-decomposition and TP graphs

Universal Clique Decomposition

A *universal clique decomposition* of a graph H is a rooted tree $\mathcal{T}$, where every node t is assigned a bag B_t such that:

- $\{B_t \colon t \in V(\mathcal{T})\}$ form a partition of $V(H)$;
- non-leaf nodes of $\mathcal{T}$ have at least two sons;
- $uv \in E(H)$ if and only if u and v share a bag, or their bags are in ancestor-descendant relation.

Lemma

A connected graph is trivially perfect if and only if it admits a universal clique decomposition. Moreover, the universal clique decomposition of a trivially perfect graph is unique.

Trivially perfect completion

- F is a *completion set* for G if $G + F = (V(G), E(G) \cup F)$ is a TP-graph.

Trivially perfect completion

- F is a *completion set* for G if $G + F = (V(G), E(G) \cup F)$ is a TP-graph.
- Trivially Perfect Completion: given G and k, does there exists a completion F for G such that $|F| \leq k$?

Trivially perfect completion

- F is a *completion set* for G if $G + F = (V(G), E(G) \cup F)$ is a TP-graph.
- Trivially Perfect Completion: given G and k, does there exists a completion F for G such that $|F| \leq k$?
- **WRONG**: Is there a completion set of size $\leq k$ that kills all induced C_4-s and P_4-s?

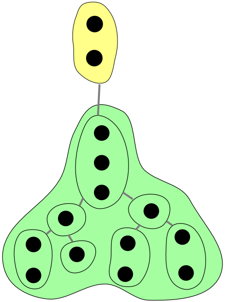
Trivially perfect completion

- F is a *completion set* for G if $G + F = (V(G), E(G) \cup F)$ is a TP-graph.
- Trivially Perfect Completion: given G and k, does there exists a completion F for G such that $|F| \leq k$?
- **WRONG**: Is there a completion set of size $\leq k$ that kills all induced C_4-s and P_4-s?
- **RIGHT**: Can one find a rooted tree of bags that needs at most k fill edges to be turned into a universal clique decomposition?

Minimal completions: properties

Lemma 1

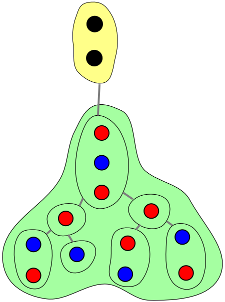
Suppose F is a minimal completion for G, and let C be a subtree of the UC-decomposition of $G + F$. Then $G[V(C)] := G[\bigcup_{t \in V(C)} B_t]$ is connected.



Minimal completions: properties

Lemma 1

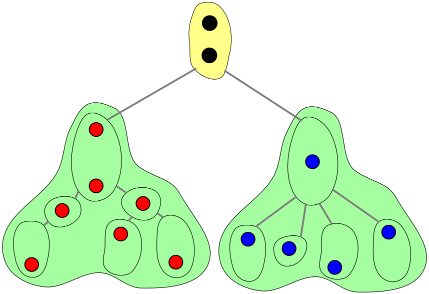
Suppose F is a minimal completion for G, and let C be a subtree of the UC-decomposition of $G + F$. Then $G[V(C)] := G[\bigcup_{t \in V(C)} B_t]$ is connected.



Minimal completions: properties

Lemma 1

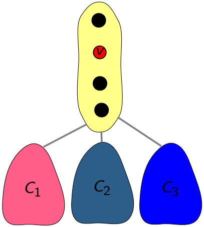
Suppose F is a minimal completion for G, and let C be a subtree of the UC-decomposition of $G + F$. Then $G[V(C)] := G[\bigcup_{t \in V(C)} B_t]$ is connected.



Minimal completions: properties

Lemma 2

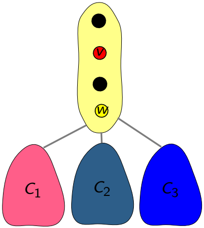
Assume F is a minimal completion for G and let v be any vertex. Let $C_1, C_2, \ldots, C_p$ be the subtrees of the UC-decomposition of $G + F$ below the bag of v. Then v has at least one neighbour in each of $V(C_1), V(C_2), \ldots, V(C_p)$ in G.



Minimal completions: properties

Lemma 2

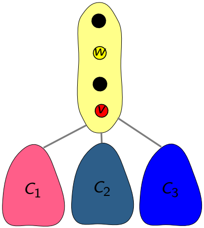
Assume F is a minimal completion for G and let v be any vertex. Let $C_1, C_2, \ldots, C_p$ be the subtrees of the UC-decomposition of $G + F$ below the bag of v. Then v has at least one neighbour in each of $V(C_1), V(C_2), \ldots, V(C_p)$ in G.



Minimal completions: properties

Lemma 2

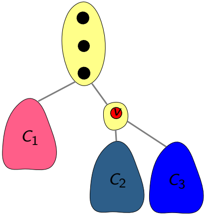
Assume F is a minimal completion for G and let v be any vertex. Let $C_1, C_2, \ldots, C_p$ be the subtrees of the UC-decomposition of $G + F$ below the bag of v. Then v has at least one neighbour in each of $V(C_1), V(C_2), \ldots, V(C_p)$ in G.



Minimal completions: properties

Lemma 2

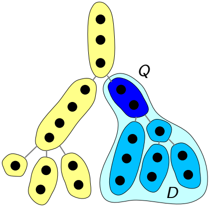
Assume F is a minimal completion for G and let v be any vertex. Let $C_1, C_2, \ldots, C_p$ be the subtrees of the UC-decomposition of $G + F$ below the bag of v. Then v has at least one neighbour in each of $V(C_1), V(C_2), \ldots, V(C_p)$ in G.



Blocks

Block

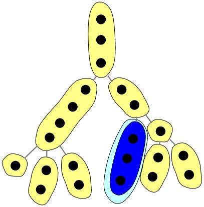
For the UC-decomposition $\mathcal{T}$ of a TP graph H and its node t, a *block* at t is the pair (Q, D), where $Q = B_t$ and D is the set of vertices in the subtree of $\mathcal{T}$ rooted in t.



Blocks

Block

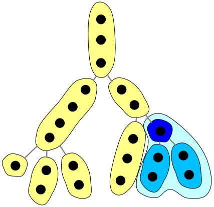
For the UC-decomposition $\mathcal{T}$ of a TP graph H and its node t, a *block* at t is the pair (Q, D), where $Q = B_t$ and D is the set of vertices in the subtree of $\mathcal{T}$ rooted in t.



Blocks

Block

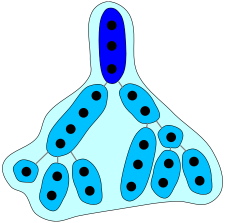
For the UC-decomposition $\mathcal{T}$ of a TP graph H and its node t, a *block* at t is the pair (Q, D), where $Q = B_t$ and D is the set of vertices in the subtree of $\mathcal{T}$ rooted in t.



Blocks

Block

For the UC-decomposition $\mathcal{T}$ of a TP graph H and its node t, a *block* at t is the pair (Q, D), where $Q = B_t$ and D is the set of vertices in the subtree of $\mathcal{T}$ rooted in t.



Dynamic programming on potential blocks

- Let $\mathbb{S}_0$ be the set of all pairs (Q, D) such that $Q \subseteq D \subseteq V(G)$ and $G[D]$ is connected. Of course $|\mathbb{S}_0| \leq 3^n$.

Dynamic programming on potential blocks

- Let $\mathbb{S}_0$ be the set of all pairs (Q, D) such that $Q \subseteq D \subseteq V(G)$ and $G[D]$ is connected. Of course $|\mathbb{S}_0| \leq 3^n$.
- Define the following value:

$$T[Q, D] \quad := \quad \text{Minimum \#edges needed to turn } G[D] \text{ into} \\ \text{a TP-graph with } Q \text{ being the universal clique}$$

Dynamic programming on potential blocks

- Let $\mathbb{S}_0$ be the set of all pairs (Q, D) such that $Q \subseteq D \subseteq V(G)$ and $G[D]$ is connected. Of course $|\mathbb{S}_0| \leq 3^n$.
- Define the following value:

$$T[Q, D] \;:=\; \text{Minimum \#edges needed to turn } G[D] \text{ into a TP-graph with } Q \text{ being the universal clique}$$

- **Lemma 1** gives the following recurrence:

$$T[Q, D] \;=\; \text{\#edges needed to make } Q \text{ a UC} + \sum_{C \in cc(G[D \setminus Q])} \left(\min_{Q_C \subseteq C} T[Q_C, C] \right)$$

Trimmed dynamic programming

- Exact $\mathcal{O}^\star(4^n)$ algorithm: compute $T[\cdot,\cdot]$ for the whole family $\mathbb{S}_0$, and output $\min_{Q\subseteq V(G)} T[Q, V(G)]$.

Trimmed dynamic programming

- Exact $\mathcal{O}^\star(4^n)$ algorithm: compute $T[\cdot,\cdot]$ for the whole family $\mathbb{S}_0$, and output $\min_{Q \subseteq V(G)} T[Q, V(G)]$.
- **Idea**: Consider some $\mathbb{S} \subseteq \mathbb{S}_0$ and trim the DP to $\mathbb{S}$:

Trimmed dynamic programming

- Exact $\mathcal{O}^\star(4^n)$ algorithm: compute $T[\cdot,\cdot]$ for the whole family $\mathbb{S}_0$, and output $\min_{Q \subseteq V(G)} T[Q, V(G)]$.
- **Idea**: Consider some $\mathbb{S} \subseteq \mathbb{S}_0$ and trim the DP to $\mathbb{S}$:
  - We define $T_{\mathbb{S}}[Q, D]$ **only** for $(Q, D) \in \mathbb{S}$.

Trimmed dynamic programming

- Exact $\mathcal{O}^\star(4^n)$ algorithm: compute $T[\cdot,\cdot]$ for the whole family $\mathbb{S}_0$, and output $\min_{Q \subseteq V(G)} T[Q, V(G)]$.
- **Idea**: Consider some $\mathbb{S} \subseteq \mathbb{S}_0$ and trim the DP to $\mathbb{S}$:
  - We define $T_{\mathbb{S}}[Q, D]$ **only** for $(Q, D) \in \mathbb{S}$.
  - In the recurrence for $T_{\mathbb{S}}[\cdot,\cdot]$ we take the minimum of $T_{\mathbb{S}}[Q_C, C]$ over $Q_C \subseteq C$ such that $(Q_C, C) \in \mathbb{S}$.

Trimmed dynamic programming

- Exact $\mathcal{O}^\star(4^n)$ algorithm: compute $T[\cdot,\cdot]$ for the whole family $\mathbb{S}_0$, and output $\min_{Q \subseteq V(G)} T[Q, V(G)]$.
- **Idea**: Consider some $\mathbb{S} \subseteq \mathbb{S}_0$ and trim the DP to $\mathbb{S}$:
  - We define $T_{\mathbb{S}}[Q, D]$ **only** for $(Q, D) \in \mathbb{S}$.
  - In the recurrence for $T_{\mathbb{S}}[\cdot,\cdot]$ we take the minimum of $T_{\mathbb{S}}[Q_C, C]$ over $Q_C \subseteq C$ such that $(Q_C, C) \in \mathbb{S}$.
- **Easy**: $T_{\mathbb{S}}[Q, D] \geq T[Q, D]$ for every $(Q, D) \in \mathbb{S}$.

Trimmed dynamic programming

- Exact $\mathcal{O}^\star(4^n)$ algorithm: compute $T[\cdot,\cdot]$ for the whole family $\mathbb{S}_0$, and output $\min_{Q \subseteq V(G)} T[Q, V(G)]$.
- **Idea**: Consider some $\mathbb{S} \subseteq \mathbb{S}_0$ and trim the DP to $\mathbb{S}$:
  - We define $T_{\mathbb{S}}[Q, D]$ **only** for $(Q, D) \in \mathbb{S}$.
  - In the recurrence for $T_{\mathbb{S}}[\cdot,\cdot]$ we take the minimum of $T_{\mathbb{S}}[Q_C, C]$ over $Q_C \subseteq C$ such that $(Q_C, C) \in \mathbb{S}$.
- **Easy**: $T_{\mathbb{S}}[Q, D] \geq T[Q, D]$ for every $(Q, D) \in \mathbb{S}$.
- **Crucial**: Suppose there exists a minimum completion F such that every block of the UC-decomposition of $G + F$ belongs to $\mathbb{S}$. Then

$$\min_{(Q,V(G)) \in \mathbb{S}} T_{\mathbb{S}}[Q, V(G)] = \min_{Q \subseteq V(G)} T[Q, V(G)].$$

Trimmed dynamic programming

- Running time of the trimmed DP: $\mathcal{O}^\star(|\mathbb{S}|^2)$.

Trimmed dynamic programming

- Running time of the trimmed DP: $\mathcal{O}^\star(|\mathbb{S}|^2)$.
- **Goal**: Enumerate a family $\mathbb{S}$ of potential blocks that is:

Trimmed dynamic programming

- Running time of the trimmed DP: $\mathcal{O}^\star(|\mathbb{S}|^2)$.
- **Goal**: Enumerate a family $\mathbb{S}$ of potential blocks that is:
  - **rich** enough so that every block of $G + F$ is captured for some optimum F, providing that $|F| \leq k$, and

Trimmed dynamic programming

- Running time of the trimmed DP: $\mathcal{O}^\star(|\mathbb{S}|^2)$.
- **Goal**: Enumerate a family $\mathbb{S}$ of potential blocks that is:
  - **rich** enough so that every block of $G + F$ is captured for some optimum F, providing that $|F| \leq k$, and
  - **small** enough so that the DP will be efficient.

Trimmed dynamic programming

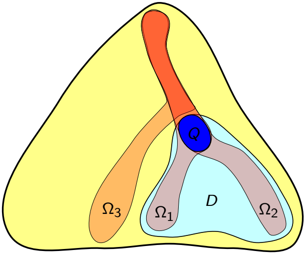
- Running time of the trimmed DP: $\mathcal{O}^\star(|\mathbb{S}|^2)$.
- **Goal**: Enumerate a family $\mathbb{S}$ of potential blocks that is:
  - **rich** enough so that every block of $G + F$ is captured for some optimum F, providing that $|F| \leq k$, and
  - **small** enough so that the DP will be efficient.
  - Our family $\mathbb{S}$ will be of size $k^{\mathcal{O}(\sqrt{k})}$ (starting from a polykernel).

Trimmed dynamic programming

- Running time of the trimmed DP: $\mathcal{O}^\star(|\mathbb{S}|^2)$.
- **Goal**: Enumerate a family $\mathbb{S}$ of potential blocks that is:
  - **rich** enough so that every block of $G + F$ is captured for some optimum F, providing that $|F| \leq k$, and
  - **small** enough so that the DP will be efficient.
  - Our family $\mathbb{S}$ will be of size $k^{\mathcal{O}(\sqrt{k})}$ (starting from a polykernel).
- **Note**: In family $\mathbb{S}$ we just need to hit all the blocks of OPT. We may put into $\mathbb{S}$ a lot of unnecessary states as well, and they do not make any harm.

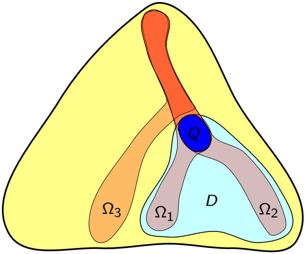
Potential Maximal Cliques → Potential Blocks

- Let (Q, D) be a block and $\Omega_1, \Omega_2, \Omega_3$ be as on the figure.



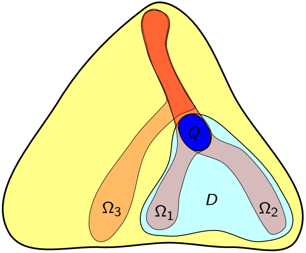
Potential Maximal Cliques → Potential Blocks

- Let (Q, D) be a block and $\Omega_1, \Omega_2, \Omega_3$ be as on the figure.
- Then $Q = (\Omega_1 \cap \Omega_2) \setminus \Omega_3$,



Potential Maximal Cliques → Potential Blocks

- Let (Q, D) be a block and $\Omega_1, \Omega_2, \Omega_3$ be as on the figure.
- Then $Q = (\Omega_1 \cap \Omega_2) \setminus \Omega_3$,
- and D is the vertex set of the connected component of $G \setminus \Omega_3$ that contains Q.



Potential Maximal Cliques → Potential Blocks

- Every block in $G + F$ can be described by a triple of maximal cliques in $G + F$.

Potential Maximal Cliques $\rightarrow$ Potential Blocks

- Every block in $G + F$ can be described by a triple of maximal cliques in $G + F$.
- If we enumerate a family $\mathbb{P}$ of potential maximal cliques, then we have a family of $|\mathbb{P}|^3$ potential blocks.

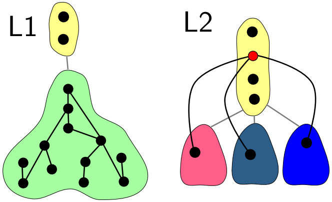
Potential Maximal Cliques $\rightarrow$ Potential Blocks

- Every block in $G + F$ can be described by a triple of maximal cliques in $G + F$.
- If we enumerate a family $\mathbb{P}$ of potential maximal cliques, then we have a family of $|\mathbb{P}|^3$ potential blocks.
  - Family $\mathbb{P}$ has to capture every maximal clique in $G + F$ for some optimum completion F.

Potential Maximal Cliques $\rightarrow$ Potential Blocks

- Every block in $G + F$ can be described by a triple of maximal cliques in $G + F$.
- If we enumerate a family $\mathbb{P}$ of potential maximal cliques, then we have a family of $|\mathbb{P}|^3$ potential blocks.
  - Family $\mathbb{P}$ has to capture every maximal clique in $G + F$ for some optimum completion F.
  - We are going to find such a family of size $k^{\mathcal{O}(\sqrt{k})}$. Let us fix some optimum completion F with $|F| \leq k$.

Turn to blackboard



Polynomial kernel: There exists a cubic kernel of Guo for TP-Completion. Hence we can assume that $n = \mathcal{O}(k^3)$.

Goal: A family $\mathbb{P}$ of $k^{\mathcal{O}(\sqrt{k})}$ subsets of $V(G)$ such that every maximal clique in $G + F$ belongs to $\mathbb{P}$.