# The effect of girth on the kernelization complexity of Connected Dominating Set

N Misra, G Philip, V Raman, and S Saurabh

*The Institute of Mathematical Sciences, Chennai, India*

FSTTCS 2010
Chennai, India
December 15, 2010

# Outline

# Outline

# Dominating Set

## Definition

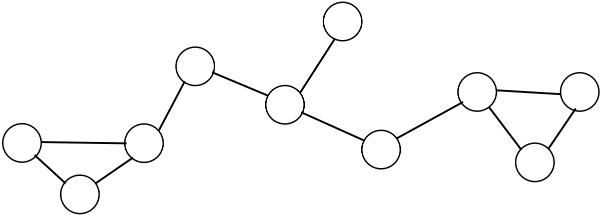
Let $G = (V, E)$ be a graph. A set $S \subseteq V$ of vertices of $G$ is said to be a *dominating set* of $G$ if every vertex which is not in $S$ has at least one neighbour in $S$.

# Dominating Set

## Definition

Let $G = (V, E)$ be a graph. A set $S \subseteq V$ of vertices of $G$ is said to be a *dominating set* of $G$ if every vertex which is not in $S$ has at least one neighbour in $S$.

## Example

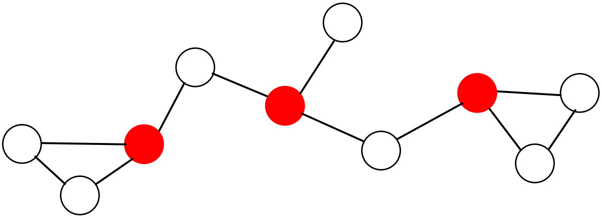


# Dominating Set

## Definition

Let $G = (V, E)$ be a graph. A set $S \subseteq V$ of vertices of $G$ is said to be a *dominating set* of $G$ if every vertex which is not in $S$ has at least one neighbour in $S$.

## Example



# Connected Dominating Set

## Definition

Let $G = (V, E)$ be a graph. A set $S \subseteq V$ of vertices of $G$ is said to be a *connected* dominating set of $G$ if

- $S$ is a dominating set of $G$, and,
- the subgraph $G[S]$ induced by $S$ is connected.

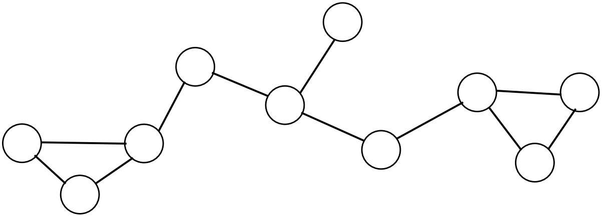
# Connected Dominating Set

## Definition

Let $G = (V, E)$ be a graph. A set $S \subseteq V$ of vertices of $G$ is said to be a *connected* dominating set of $G$ if

- $S$ is a dominating set of $G$, and,
- the subgraph $G[S]$ induced by $S$ is connected.

## Example



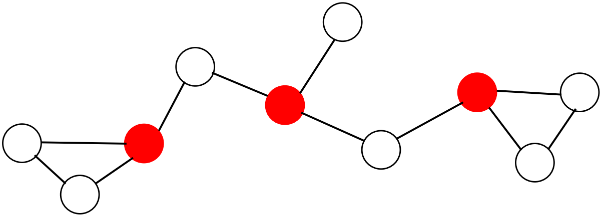
# Connected Dominating Set

## Definition

Let $G = (V, E)$ be a graph. A set $S \subseteq V$ of vertices of $G$ is said to be a *connected* dominating set of $G$ if

- $S$ is a dominating set of $G$, and,
- the subgraph $G[S]$ induced by $S$ is connected.

## Example



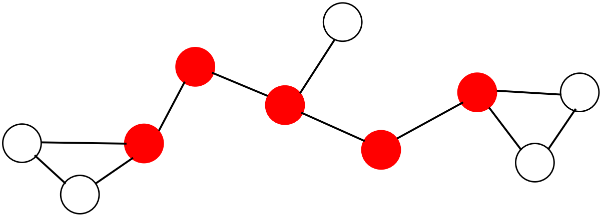
# Connected Dominating Set

## Definition

Let $G = (V, E)$ be a graph. A set $S \subseteq V$ of vertices of $G$ is said to be a *connected* dominating set of $G$ if

- $S$ is a dominating set of $G$, and,
- the subgraph $G[S]$ induced by $S$ is connected.

## Example



# The CDS problem

## Definition (The CDS problem)

- Input: A graph $G$ and a positive integer $k$.
- Question: Does $G$ have a connected dominating set of size at most $k$?

## Classical Complexity

- The problem is NP-complete, even in planar graphs of maximum degree $\Delta \leq 4$ [Garey and Johnson, SIAM J. Appl. Math 1977].
- Can be approximated in polynomial time to within a factor of $\ln(\Delta) + 3$ [Guha and Khuller, Algorithmica 1998].

The Institute of Mathematical Sciences

# The CDS problem

## Definition (The CDS problem)

- Input: A graph $G$ and a positive integer $k$.
- Question: Does $G$ have a connected dominating set of size at most $k$?

## Classical Complexity

- The problem is NP-complete, even in planar graphs of maximum degree $\Delta \leq 4$ [Garey and Johnson, SIAM J. Appl. Math 1977].
- Can be approximated in polynomial time to within a factor of $\ln(\Delta) + 3$ [Guha and Khuller, Algorithmica 1998].

# Outline

# PC: Brief Overview 1/3

- Goal: Solve NP-hard problems in polynomial time ...
  - ... when some aspect (the *parameter*) of the input is bounded.
- A parameterized problem instance has two parts:
  - The input itself, and,
  - A *parameter*, usually a number denoted by $k$.

# PC: Brief Overview 1/3

- Goal: Solve NP-hard problems in polynomial time ...
  - ... when some aspect (the *parameter*) of the input is bounded.
- A parameterized problem instance has two parts:
  - The input itself, and,
  - A *parameter*, usually a number denoted by $k$.

The Institute of Mathematical Sciences

# PC: Brief Overview 2/3

- A parameterized problem is *fixed-parameter tractable* (**FPT**) if
  - It can be solved in time $f(k) \cdot n^c$ time, where
    - $c$ is a constant,
    - $f()$ is computable, and,
    - $f(k)$ depends only on $k$.
- A *kernelization algorithm* for a parameterized problem
  - Runs in time polynomial in the input size, and
  - Outputs an **equivalent** instance — a *kernel* — of size bounded by some $g(k)$.
- Fact (folklore): A parameterized problem is FPT if and only if it has a kernelization algorithm.

# PC: Brief Overview 2/3

- A parameterized problem is *fixed-parameter tractable* (**FPT**) if
  - It can be solved in time $f(k) \cdot n^c$ time, where
    - $c$ is a constant,
    - $f()$ is computable, and,
    - $f(k)$ depends only on $k$.
- A *kernelization algorithm* for a parameterized problem
  - Runs in time polynomial in the input size, and
  - Outputs an **equivalent** instance — a *kernel* — of size bounded by some $g(k)$.
- Fact (folklore): A parameterized problem is FPT if and only if it has a kernelization algorithm.

# PC: Brief Overview 3/3

Given a parameterization of an NP-hard problem, we ask:

- Is the problem FPT?
  - Does it have an algorithm that runs in $f(k) \cdot n^c$ time?
  - There is a hardness theory to show that certain problems are *unlikely* to be FPT.
- If YES, does it have a *small* kernel?
  - Polynomial-size kernels are good, linear-size even better.
  - There is a hardness theory to show that certain problems are *unlikely* to have polynomial-size kernels.

# PC: Brief Overview 3/3

Given a parameterization of an NP-hard problem, we ask:

- Is the problem FPT?
  - Does it have an algorithm that runs in $f(k) \cdot n^c$ time?
  - There is a hardness theory to show that certain problems are *unlikely* to be FPT.
- If YES, does it have a *small* kernel?
  - Polynomial-size kernels are good, linear-size even better.
  - There is a hardness theory to show that certain problems are *unlikely* to have polynomial-size kernels.

# The CDS problem
## Parameterized Version

### Definition

- Input: A graph $G$ and a positive integer $k$.
- Parameter: $k$
- Question: Does $G$ have a connected dominating set of size at most $k$?

# The CDS problem

## Parameterized Complexity: Known Results

- W[2]-hard on graphs in general:
  - $O(f(k) \cdot poly(n))$-time algorithms are unlikely to exist.
- FPT on graphs of bounded degeneracy (Golovach, Villanger : WG 2008).
  - Planar graphs, graphs of bounded treewidth, ...
- Has a *linear* kernel in apex-minor-free graphs (Fomin, Lokshtanov, Saurabh, Thilikos : SODA 2010).
- *Unlikely* to have polynomial kernels in graphs of bounded degeneracy (Cygan, Pilipczuk, Pilipczuk, Wojtaszczyk : WG 2010).

# Outline

The Connected Dominating Set problem

- Introduction
- Parameterized Complexity
- Our Results

# The CDS problem

## Parameterized Complexity: Our Results

- We explore what happens when the *girth* of the input graph is bounded.
  - The girth of a graph $G$ is the size (number of edges) of a smallest cycle in $G$.
  - "Orthogonal" to degeneracy:
    - There are graphs of fixed girth and arbitrarily large degeneracy, and vice versa.

# The CDS problem

## Parameterized Complexity: Our Results

- We completely characterize the parameterized complexity of CDS for different values of girth:
    - W[2]-hard on graphs of girth 3 or 4.
    - FPT on graphs of girth at least 5.
    - Polynomial kernels unlikely in graphs of girth 5 or 6.
    - Cubic ($O(k^3)$) kernel in graphs of girth at least 7.

# The CDS problem

## Parameterized Complexity: Our Results

- We completely characterize the parameterized complexity of CDS for different values of girth:
    - W[2]-hard on graphs of girth 3 or 4.
    - FPT on graphs of girth at least 5.
    - Polynomial kernels unlikely in graphs of girth 5 or 6.
    - Cubic ($O(k^3)$) kernel in graphs of girth at least 7.

# CDS in Graphs of Girth 3 or 4

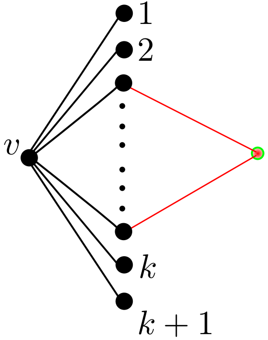
## W[2]-hardness Reduction

- Reduction from Dominating Set.
- Appears in earlier work of Raman and Saurabh.
  - To show that *Dominating Set* is $W[2]$-hard on bipartite graphs.
- We observe that it works for CDS as well.

# CDS in Graphs of Girth at least 5

## FPT Algorithm: The Degree Rule

- If a vertex $v$ has degree more than $k$, then $v$ must be in every dominating set of size at most $k$.



# CDS in Graphs of Girth at least 5

## FPT Algorithm: In Brief

- The Degree Rule : every vertex of ''high'' degree must be in *any* small CDS.
- At most $k + k(k+1) + k^2(k+1)$ "candidate" vertices from which to pick any minimal dominating set of size at most $k$:
  - Guess a minimal DS $S$ from among these,
  - Find a minimum-size Steiner tree $T$ for this DS, and,
  - Check if $|V(T)| \leq k$.
- Total running time: $O^{\star}(k^{O(k)})$.

# Graphs of Girth 5 or 6 : No Polynomial Kernels

## Kernel Lower Bound Machinery

- We make use of kernel lower bound techniques developed by
  - Bodlaender, Downey, Fellows, Hermelin : *JCSS* 2009
  - Bodlaender, Thomassé, Yeo : *ESA* 2009
- These techniques allow us to conclude that a problem does not have polynomial kernels unless $NP \subseteq CoNP/Poly$ (and the Polynomial-Time Hierarchy collapses to the third level).

## Composition Algorithm

A composition algorithm for a parameterized problem

- takes as input a *sequence* $\langle (x_1, k), (x_2, k), \ldots, (x_t, k) \rangle$
- runs in time polynomial in $\sum_{i=1}^{t} |x_i| + k$
- outputs an instance $(y, k')$ :
  - $(y, k')$ is YES iff **at least one** $(x_i, k)$ is YES,
  - $k'$ is polynomial in $k$.

The composition algorithm acts like an **OR** gate.

# Kernel Lower Bound Machinery

## Tool 1 : Composition

### Theorem (BDFH 2009)

*If a parameterized NP-complete problem has a composition algorithm, then it does not have a polynomial kernel, unless $NP \subseteq CoNP/Poly$.*

### Example

- $k$-path: Does graph $G$ have a path of length at least $k$?
- NP-hard (Hamiltonian Path)
- Is compositional: take the disjoint union of all input graphs.
- This implies: No $poly(k)$ kernel unless ...

# Kernel Lower Bound Machinery

Tool 1 : Composition

## Theorem (BDFH 2009)

*If a parameterized NP-complete problem has a composition algorithm, then it does not have a polynomial kernel, unless* $NP \subseteq CoNP/Poly$.

## Example

- $k$-path: Does graph $G$ have a path of length at least $k$?
- NP-hard (Hamiltonian Path)
- Is compositional: take the disjoint union of all input graphs.
- This implies: No $poly(k)$ kernel unless ...

# Kernel Lower Bound Machinery

## Tool 2 : Reduction

- Let $\mathcal{A}, \mathcal{B}$ be parameterized problems.

### Definition (Polynomial Parameter Transformation)

A *polynomial parameter transformation* (**PPT**) from $\mathcal{A}$ to $\mathcal{B}$ is a function $f$ such that:

- $f$ is polynomial-time computable.
- If $f(x, k) = (x', k')$, then
  - $(x, k) \in \mathcal{A} \iff (x', k') \in \mathcal{B}$.
  - $k' \leq p(k)$ for some polynomial $p$.

# Kernel Lower Bound Machinery

## Tool 2 : Reduction

### Theorem (BTY 2009)

*Let $\mathcal{A}, \mathcal{B}$ be parameterized problems where $\mathcal{A}$ is NP-complete and $\mathcal{B}$ is in NP. Suppose there is a PPT from $\mathcal{A}$ to $\mathcal{B}$. Then, if $\mathcal{B}$ has a polynomial kernel, then $\mathcal{A}$ also has a polynomial kernel.*

- If $\mathcal{A}$ has no polynomial kernel, and there is a PPT from $\mathcal{A}$ to $\mathcal{B}$, then $\mathcal{B}$ has no polynomial kernel.

# Kernel Lower Bound Machinery

## Tools : Summary

- Composition: **OR**-like algorithm, implies no poly-kernel unless $NP \subseteq CoNP/Poly$.
- PPT Reduction: An NP-completeness reduction with at most a polynomial increase in the parameter. Helps propagate no-poly-kernel bounds.

# CDS in Graphs of Girth 5 or 6

## Kernel Lower Bound

- Our approach:
  - Introduce an intermediate problem, Fair Connected Colours (FCC).
    - Show that FCC is NP-hard, and compositional.
  - Show that there is a polynomial parameter transformation (PPT) from FCC to CDS in graphs of girth 5 or 6.

- Input: A graph $G$ whose vertices are *properly* coloured with $k$ colours such that all neighbours of each vertex have distinct colours (a *fair* colouring).
- Parameter: $k$
- Question: Does $G$ contain a tree $T$ on $k$ vertices where all vertices in $T$ have different colours (a *colourful* tree)?

# CDS in Graphs of Girth 5 or 6

## Fair Connected Colours (FCC)

- FCC is NP-complete
  - Reduction from SAT
- FCC is compositional
  - Just take the disjoint union!
- So from the earlier theorem, we have:
  - FCC has no polynomial kernel unless $NP \subseteq CoNP/Poly$.

# CDS in Graphs of Girth 5 or 6

## Fair Connected Colours (FCC)

- FCC is NP-complete
    - Reduction from SAT
- FCC is compositional
    - Just take the disjoint union!
- So from the earlier theorem, we have:
    - FCC has no polynomial kernel unless $NP \subseteq CoNP/Poly$.

- FCC is NP-complete
  - Reduction from SAT
- FCC is compositional
  - Just take the disjoint union!
- So from the earlier theorem, we have:
  - FCC has no polynomial kernel unless $NP \subseteq CoNP/Poly$.

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Outline

- Given an instance $(G, k)$ of FCC, we construct an instance $(H, k^2 + k)$ of CDS such that:
  - $H$ has girth 6.
  - A colourful tree in $G$ becomes a connected dominating set in $H$:
    - The one vertex in each colour class dominates (well, sort of) all of that class.
    - The underlying tree provides connectivity.
  - And vice versa.
    - The $+k^2$ is spent for increasing the girth, and for domination.
  - The modification for girth 5 is not difficult.

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Outline

- Given an instance $(G, k)$ of FCC, we construct an instance $(H, k^2 + k)$ of CDS such that:
  - $H$ has girth 6.
  - A colourful tree in $G$ becomes a connected dominating set in $H$:
    - The one vertex in each colour class dominates (well, sort of) all of that class.
    - The underlying tree provides connectivity.
  - And vice versa.
    - The $+k^2$ is spent for increasing the girth, and for domination.
  - The modification for girth 5 is not difficult.

# CDS in Graphs of Girth 5 or 6

## Summary

- Fair Connected Colours is NP-complete and compositional.
- Fair Connected Colours reduces to Connected Dominating Set
  - In polynomial time, and with polynomial increase in the parameter.
- These together imply that Connected Dominating Set is not likely to have polynomial kernels.

# CDS in Graphs of Girth at Least 7

## Cubic Kernel: Outline

- The main reduction rule is the same degree rule as before:
  - Any vertex with degree more than $k$ must be part of the solution.
- We use colours to keep track of the state:
  - Red vertices are in the dominating set that we construct.
  - Vertices dominated by the red ones are green.
  - Undominated vertices are blue.
- The larger girth (7) allows us to bound the number of green vertices as well.
  - In girth-5 graphs, we could only bound the number of green vertices *which were adjacent to* blue vertices.

# CDS in Graphs of Girth at Least 7

## Cubic Kernel: Outline

- At most $k$ red vertices.
- At most $k(k+1)$ blue vertices.
- At most $k(k^2+k)+3\cdot\binom{k}{2}$ green vertices.
- A coloured kernel on at most $k^3+\frac{7}{2}k^2+\frac{k}{2}$ vertices.
- To get a “plain” kernel:
  - Attach a new pendant vertex to each red vertex.
  - Remove all colours.
  - A kernel for CDS on at most $k^3+\frac{7}{2}k^2+\frac{3}{2}k$ vertices.

# Recapitulating . . .

- Connected Dominating Set parameterized by solution size $k$ is:
  - W[2]-hard on graphs in general.
  - W[2]-hard on graphs of girth at most 4.
  - FPT on graphs of girth at least 5.
  - No polynomial kernel (. . . ) in graphs of girth 5 or 6.
  - A cubic ($O(k^3)$) kernel in graphs of girth at least 7.

Thank You!

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Outline

- Given an instance $(G, k)$ of FCC, we construct an instance $(H, k^2 + k)$ of CDS such that:
    - $H$ has girth 6.
    - A colourful tree in $G$ becomes a connected dominating set in $H$:
        - The one vertex in each colour class dominates (well, sort of) all of that class.
        - The underlying tree provides connectivity.
    - And vice versa.
        - The $+k^2$ is spent for increasing the girth, and for domination.
    - The modification for girth 5 is not difficult.

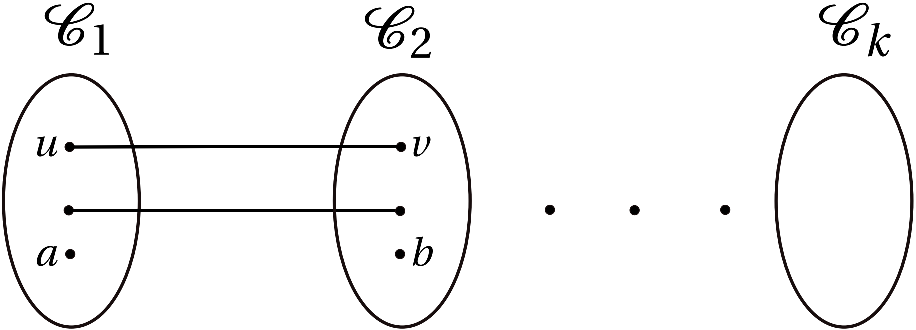
# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Outline

- Given an instance $(G, k)$ of FCC, we construct an instance $(H, k^2 + k)$ of CDS such that:
  - $H$ has girth 6.
  - A colourful tree in $G$ becomes a connected dominating set in $H$:
    - The one vertex in each colour class dominates (well, sort of) all of that class.
    - The underlying tree provides connectivity.
  - And vice versa.
    - The $+k^2$ is spent for increasing the girth, and for domination.
  - The modification for girth 5 is not difficult.

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction



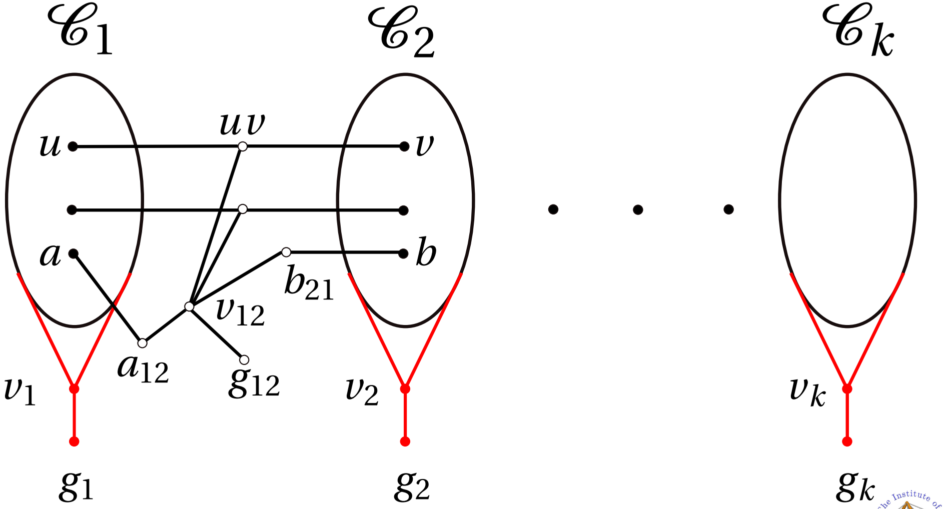
# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction

- Add a global vertex $v_i$ to each colour class $\mathcal{C}_i$.
- Attach a pendant (guard) vertex $g_i$ to each $v_i$.

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction



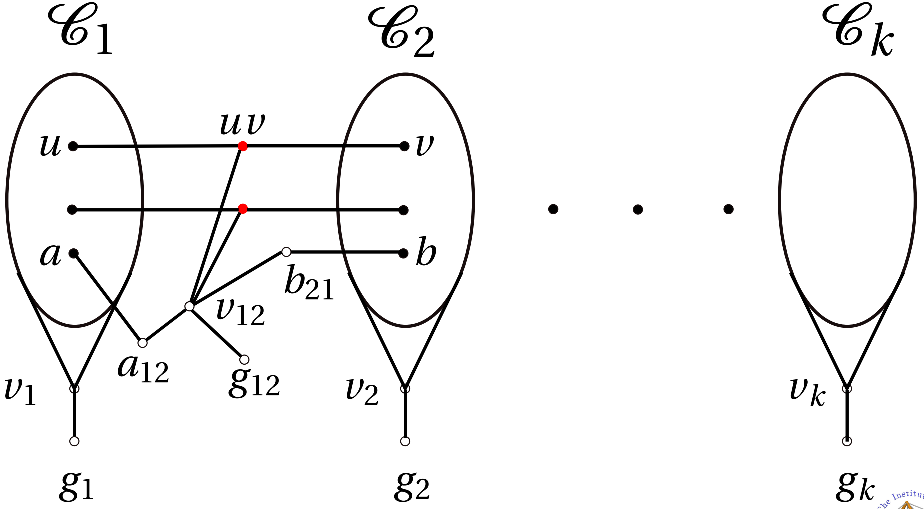
# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction

- Add a global vertex $v_i$ to each colour class $\mathcal{C}_i$.
- Attach a pendant (guard) vertex $g_i$ to each $v_i$.
- Subdivide each edge $\{u, v\}$ of $G$ with a new vertex $uv$.

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction



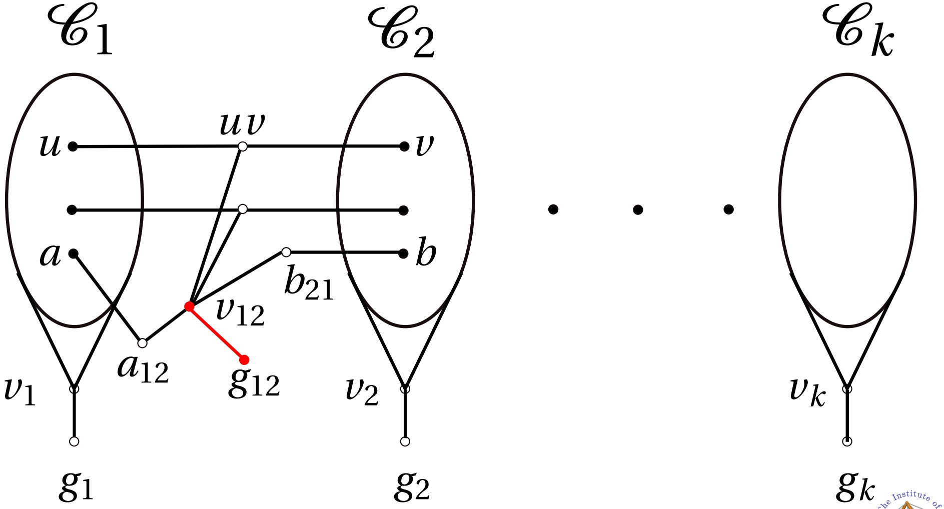
# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction

- Add a global vertex $v_i$ to each colour class $\mathcal{C}_i$.
- Attach a pendant (guard) vertex $g_i$ to each $v_i$.
- Subdivide each edge $\{u, v\}$ of $G$ with a new vertex $uv$.
- For every two colour classes $\mathcal{C}_i, \mathcal{C}_j$,
  - Add two new vertices $v_{ij}, g_{ij}$ and the edge $\{v_{ij}, g_{ij}\}$

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction



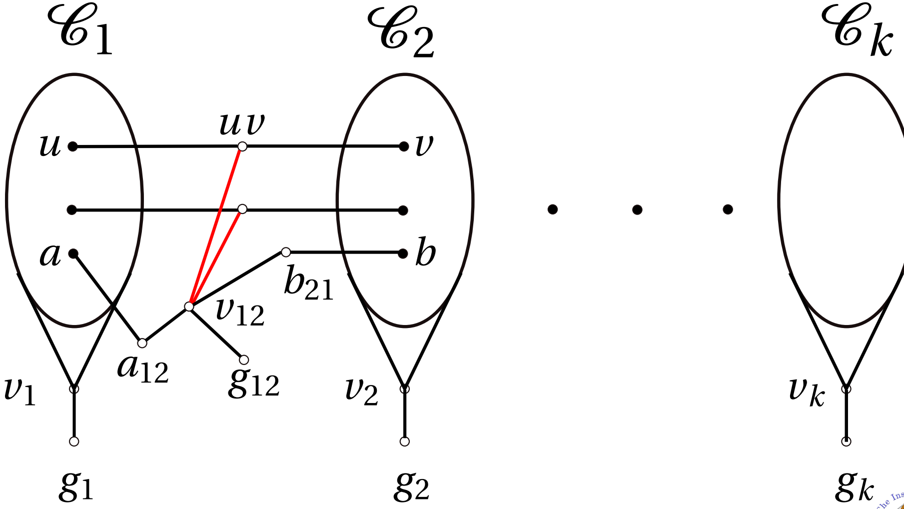
# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction

- Add a global vertex $v_i$ to each colour class $\mathcal{C}_i$.
- Attach a pendant (guard) vertex $g_i$ to each $v_i$.
- Subdivide each edge $\{u, v\}$ of $G$ with a new vertex $uv$.
- For every two colour classes $\mathcal{C}_i, \mathcal{C}_j$,
  - Add two new vertices $v_{ij}, g_{ij}$ and the edge $\{v_{ij}, g_{ij}\}$
  - For each edge $\{u, v\}; u \in \mathcal{C}_i, v \in \mathcal{C}_j$ in $G$, add the edge $\{uv, v_{ij}\}$.

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction



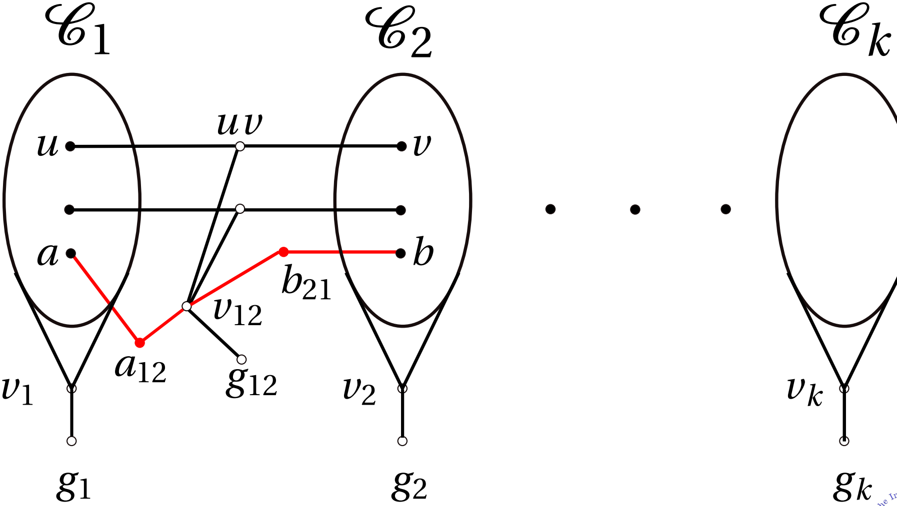
# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction

- Add a global vertex $v_i$ to each colour class $\mathcal{C}_i$.
- Attach a pendant (guard) vertex $g_i$ to each $v_i$.
- Subdivide each edge $\{u, v\}$ of $G$ with a new vertex $uv$.
- For every two colour classes $\mathcal{C}_i, \mathcal{C}_j$,
    - Add two new vertices $v_{ij}, g_{ij}$ and the edge $\{v_{ij}, g_{ij}\}$
    - For each edge $\{u, v\}; u \in \mathcal{C}_i, v \in \mathcal{C}_j$ in $G$, add the edge $\{uv, v_{ij}\}$.
    - If $u \in \mathcal{C}_i$ has no neighbour in $\mathcal{C}_j$, add a new vertex $u_{ij}$ and the edges $\{u, u_{ij}\}, \{u_{ij}, v_{ij}\}$

# CDS in Graphs of Girth 5 or 6

## Reduction from FCC : Construction



# CDS in Graphs of Girth 5 or 6

Reduction from FCC : Correctness

## Forward direction

- To a colourful tree on $k$ vertices in $G$ we add:
  - The $k-1$ "edge" vertices;
  - The $k$ global vertices for the colour classes;
  - The $\binom{k}{2}$ vertices $v_{ij}$, and,
  - $\binom{k}{2}-(k-1)$ vertices to dominate the vertices $v_{ij}$
- ... to get a CDS of $H$ on $k^2+k$ vertices.

# CDS in Graphs of Girth 5 or 6

Reduction from FCC : Correctness

## Reverse direction

- Any CDS of $G$ must contain:
  - The $k$ vertices $v_i$, and the $\binom{k}{2}$ vertices $v_{ij}$
    - Domination constraint
  - And at least one (distinct) neighbour of each of these vertices
    - Connectivity constraint
- This uses up the budget of $k^2 + k$.
  - The vertices $v_i$ and $v_{ij}$ are all leaves in the CDS.

## Reverse direction

- Prune the leaves $v_i$ and $v_{ij}$ of the CDS
- The remaining part $T'$
  - is connected,
  - contains exactly one vertex in each colour class, and,
  - contains exactly $\binom{k}{2} + k$ vertices
- For each $v_{ij}$,
  - If its neighbour in $T'$ is a leaf, delete it.
  - Otherwise, it is one of the "edge" vertices: contract it
- This leaves a connected graph $T$ on exactly $k$ vertices, one from each colour class.