Data Reduction, Exact, and Heuristic Algorithms for Clique Cover

Jens Gramm     Jiong Guo     Falk Hüffner     Rolf Niedermeier

Friedrich-Schiller-Universität Jena
Institut für Informatik
Clique Cover

**Definition**

*(Edge) Clique Cover*

**Input:** An undirected graph $G = (V, E)$.

**Task:** Find a minimum number $k$ of cliques such that each edge is contained in at least one clique.
Definition

**EDGE CLIQUE COVER**

**Input:** An undirected graph $G = (V, E)$.  

**Task:** Find a minimum number $k$ of cliques such that each edge is contained in at least one clique.
Clique Cover

Also known as

- **Keyword Conflict** [Kellerman, IBM 1973]
- **Intersection Graph Basis** [Garey & Johnson 1979]
Clique Cover

Also known as

- **Keyword Conflict** [Kellerman, IBM 1973]
- **Intersection Graph Basis** [Garey&Johnson 1979]

Applications

- compiler optimization,
- computational geometry,
- statistics visualization, ...
Clique Cover

Also known as
- **Keyword Conflict** [Kellerman, IBM 1973]
- **Intersection Graph Basis** [Garey&Johnson 1979]

Applications
- compiler optimization,
- computational geometry,
- statistics visualization, ...

Properties
- **NP-complete** [Garey&Johnson 1979]
- **NP-hard to approximate to constant factor** [Ausiello et al. 1999]
Definition

A data reduction rule replaces a Clique Cover instance by a simpler instance, such that the solution to the original instance can be reconstructed from the solution of the simpler instance.
Data Reduction Rules for Clique Cover

Definition

A data reduction rule replaces a Clique Cover instance by a simpler instance, such that the solution to the original instance can be reconstructed from the solution of the simpler instance.

Annotated Clique Cover

- Edges can be marked as covered
- Only uncovered edges have to be covered by cliques
Rule 1

Remove isolated vertices and vertices that are only adjacent to covered edges.
**Simple Data Reduction Rules for Clique Cover**

**Rule 1**
Remove isolated vertices and vertices that are only adjacent to covered edges.

**Rule 2**
If an edge \(\{u, v\}\) is contained only in exactly one maximal clique \(C\), then add \(C\) to the solution, mark its edges as covered, and decrease \(k\) by one.
Simple Data Reduction Rules for Clique Cover

**Rule 1**
Remove isolated vertices and vertices that are only adjacent to covered edges.

**Rule 2**
If an edge \( \{u, v\} \) is contained only in exactly one maximal clique \( C \), then add \( C \) to the solution, mark its edges as covered, and decrease \( k \) by one.

![Diagram of a clique graph with a highlighted clique and a vertex labeled \( k - 1 \)]
Prisoner/Exits Reduction Rules for Clique Cover

Partition the neighborhood of a vertex $v$ into:
- prisoners $p$ with $N(p) \subseteq N(v)$ and
- exits $x$ with $N(x) \setminus N(v) \neq \emptyset$.

**Rule 4**
If all exits have at least one prisoner as neighbor, then delete $v$. 

![Graph example](image-url)
Consider a Clique Cover instance with $n$ vertices and $k$ cliques allowed.

**Theorem**

After applying all reduction rules exhaustively, a Clique Cover instance has at most $2^k$ vertices, that is, Clique Cover has a problem kernel of size $2^k$. 

**Corollary**

Clique Cover is fixed-parameter tractable with respect to the parameter $k$, that is, it can be solved in time $f(k) \cdot n^{O(1)}$ for some function $f$ depending only on $k$. 

Gramm et al. (FSU Jena)
Consider a Clique Cover instance with $n$ vertices and $k$ cliques allowed.

**Theorem**

After applying all reduction rules exhaustively, a Clique Cover instance has at most $2^k$ vertices, that is, Clique Cover has a problem kernel of size $2^k$.

**Corollary**

Clique Cover is fixed-parameter tractable with respect to the parameter $k$, that is, it can be solved in time $f(k) \cdot n^{O(1)}$ for some function $f$ depending only on $k$. 
Exact Algorithm for Clique Cover

Search-tree algorithm for **Clique Cover**:

- Choose some uncovered edge $e$
- For each maximal clique $C$ that contains $e$, mark all edges in $C$ as covered, decrease $k$ by one, and call the algorithm recursively

Results:
- Horrible worst-case complexity...
- ...but:
  - Works nicely in practice when combined with data reduction rules.
  - Can solve all instances in a benchmark from applied statistics within a second (up to 124 vertices and 4847 edges).
  - Can solve sparse instances with hundreds of vertices and tens of thousands of edges within minutes.

Gramm et al. (FSU Jena)
Exact Algorithm for Clique Cover

Search-tree algorithm for Clique Cover:
- Choose some uncovered edge $e$
- For each maximal clique $C$ that contains $e$, mark all edges in $C$ as covered, decrease $k$ by one, and call the algorithm recursively

Results:
- Horrible worst-case complexity...
Exact Algorithm for Clique Cover

Search-tree algorithm for **Clique Cover**:
- Choose some uncovered edge $e$
- For each maximal clique $C$ that contains $e$, mark all edges in $C$ as covered, decrease $k$ by one, and call the algorithm recursively

Results:
- Horrible worst-case complexity...

...but:
- Works nicely in practice when combined with data reduction rules.
- Can solve all instances in a benchmark from applied statistics within a second (up to 124 vertices and 4847 edges).
- Can solve sparse instances with hundreds of vertices and tens of thousands of edges within minutes.
Exact Algorithm for Clique Cover

![Graph showing runtime in seconds vs. number of vertices for different densities]

- Curve 1: Sparse graph
- Curve 2: Density 0.1
- Curve 3: Density 0.15

Gramm et al. (FSU Jena)
Summary

- Data reduction rules can be successfully applied to Clique Cover.
- An exact algorithm based on the data reduction rules and a search tree can solve many practically relevant instances.
- Further results in the paper: runtime improvement for a heuristic.
Open question

In the statistics application, it is also desirable to minimize the sum of clique sizes.

Question

Is there a solution that minimizes the sum of clique sizes, but not the number of cliques?