On the equivalence covering number of splitgraphs

Citation for published version (APA):

Document status and date:
Published: 01/01/1994

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 02. Oct. 2020
On the equivalence covering number of splitgraphs

by

A. Blokhuis and T. Kloks

94/39
On the equivalence covering number of splitgraphs

A. Blokhuis* and T. Kloks †

Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O.Box 513, 5600 MB Eindhoven, The Netherlands

Keywords: Graph algorithms, equivalence covering, splitgraphs.

Abstract

An equivalence graph is a disjoint union of cliques. For a graph $G$ let $eq(G)$ be the minimum number of equivalence subgraphs of $G$ needed to cover all edges of $G$. We call $eq(G)$ the equivalence covering number of $G$. We show that the equivalence covering number for splitgraphs can be approximated within an additive constant $1$. We also show that obtaining the exact value of the equivalence number of a splitgraph is an NP-hard problem. Using a similar method we also show that the computation of the equivalence number remains NP-complete for graphs with maximum degree $6$ and with maximum clique number $4$.

1 Introduction

Definition 1 An equivalence graph is a vertex disjoint union of cliques. An equivalence covering of a graph $G$ is a family of equivalence subgraphs of $G$ such that every edge of $G$ is an edge of at least one member of the family. The equivalence covering number of $G$, denoted by $eq(G)$, is the minimum cardinality of all equivalence coverings of $G$.

The equivalence covering number was studied first in [2]. Interesting bounds for the equivalence covering number in terms of maximal degree of the complement were obtained in [1]. In this note we mainly consider the computation of the equivalence covering number of splitgraphs. We first show an approximation within an additive constant $1$. Then we show that obtaining the exact value is an NP-hard problem.

Definition 2 A graph $G = (V, E)$ is a split graph, if there is a partition $V = S + K$ of its vertex set into a stable set $S$ and a clique $K$.

There is no restriction on edges between vertices of $S$ and vertices of $K$. Notice that in general the partition into $S$ and $K$ need not be unique. Splitgraphs are exactly those graphs which, together with their complements, are chordal. For more general information on splitgraphs we refer to [4].

2 Approximation

In this section we show that the equivalence covering number of a splitgraph can be approximated within an additive constant $1$. Consider a partition $V = S + K$ of the vertex set into an independent set $S$ and a clique $K$. For a vertex $x$ in $K$ let $\delta(x)$ be the number of neighbors of $x$ in $S$. Let $\Delta = \max\{\delta(x) \mid x \in K\}$.

Lemma 1 $eq(G) \geq \Delta$. 

joint. We define one more equivalence graph following theorem.

The approximated index is either done for each equivalence graph in the covering can have only one edge. This proves the lemma. □

**Lemma 2** \( \text{eq}(G) \leq \Delta + 1 \).

**Proof.** Let \( y_1, \ldots, y_t \) be the vertices of \( S \). For each vertex \( x \) in \( K \) consider an arbitrary ordering of its neighbors in \( S \). For \( i = 1, \ldots, \Delta \) define the equivalence graph \( G_i \) as follows. \( G_i \) is the disjoint union of cliques \( W_{i,j} = \{ y_j \} \cup \{ x \in K \mid \text{the } i\text{th neighbor of } x \text{ is } y_j \} \), for \( j = 1, \ldots, t \). It is easy to check that the cliques \( W_{i,j} \) for \( j = 1, \ldots, t \) are all disjoint. We define one more equivalence graph \( G_{\Delta+1} \) consisting of the clique \( K \). Obviously, this gives an equivalence covering with \( \Delta + 1 \) equivalence graphs.

The approximation given in Lemma 2 can be computed in linear time. This proves the following theorem.

**Theorem 1** There exists a linear time algorithm to compute an equivalence covering of a split graph \( G \) with at most \( \text{eq}(G) + 1 \) equivalence graphs.

**Remark 1** Notice that, in case the split graph is a threshold graph (see, e.g., [4]), its equivalence number can easily be computed exactly.

3 NP-completeness

We use a reduction from *EDGE-COLORING*.

The chromatic index of a graph \( G \), denoted by \( \chi'(G) \), is the number of colors required to color the edges of the graph in such a way that no two adjacent edges have the same color. By Vizing's theorem (see, e.g., [3]) the chromatic index is either \( d \) or \( d + 1 \), where \( d \) is the maximum vertex degree.

Notice that, in general, the chromatic index is an upperbound for the equivalence covering number. Also, these parameters coincide for triangle-free graphs. It follows that, for bipartite graphs, the equivalence covering number equals the maximum degree. Unfortunately, for split graphs the bound is not of much use, which is illustrated by a clique.

It is by now well-known that it is NP-complete to determine the chromatic index of an arbitrary graph [5, 6]. Holyer [5] obtained the following result.

**Theorem 2** It is NP-complete to determine whether the chromatic index of a cubic graph is 3 or 4.

Consider a cubic graph \( G \) and construct a graph \( H \) as follows. For each edge \( e \) of \( G \) introduce a new vertex \( x_e \) and make this adjacent to the two end vertices of \( e \). We call \( x_e \) the *special vertex* at \( e \).

**Lemma 3** \( \chi'(G) = 3 \Leftrightarrow \text{eq}(H) = 3 \).

**Proof.** First assume \( \chi'(G) = 3 \). Notice that \( \text{eq}(H) \geq 3 \) since \( H \) has an induced \( K_{1,3} \) subgraph. (If \( p \) is a vertex of \( G \) incident with edges \( e, f \) and \( g \) in \( G \), then \( \{ p, x_e, x_f, x_g \} \) induces a \( K_{1,3} \) in \( H \).) Consider an edge coloring of \( G \) with three colors. For each color class define an equivalence graph as follows. For each edge in that color class, the triangle consisting of the edge and the special vertex at that edge is a clique of the equivalence graph. It is easy to check that this defines an equivalence covering with three equivalence graphs.

Now assume \( H \) has an equivalence covering with three equivalence graphs \( H_1, H_2 \) and \( H_3 \). We claim that no triangle of \( G \) is contained in a clique of one of the equivalence graphs. Assume, by way of contradiction, that \( \{ a, b, c \} \) is a triangle of \( G \) which is contained in a clique of \( H_1 \). Vertex \( a \) is adjacent to three special vertices, say \( x_1, x_2 \) and \( x_3 \). Then each of the edges \( (a, x_i) \) is contained in a clique of an equivalence graph, and no two are in a clique of the same equivalence graph. Without loss
of generality we may assume that \((a, x_i)\) is contained in a clique of \(H_i\). But then \(H_i\) cannot contain the triangle \(\{a, b, c\}\) since \(x_1\) has degree two and hence the clique containing \(a\) and \(x_1\) can have at most two vertices of \(G\).

We can color the edges of \(G\) as follows. If the edge \(e\) is contained in a clique of \(H_i\) then we give it color \(i\). (If \(e\) is contained in cliques of more than one equivalence graph, we can choose one arbitrarily). By the remark above this gives a correct edge-coloring with three colors. 

\[\square\]

**Corollary 1** It is NP-complete to determine whether the equivalence covering number of a graph with maximum degree \(\leq 6\) and without induced \(K_4\) is 3 or 4.

Given a cubic graph \(G\) we construct a splitgraph \(G^*\) as follows. The vertex set of \(G^*\) is split into a clique \(K\) and an independent set \(S\). The vertices of \(K\) are the vertices of \(G\). For each edge \(e\) of \(G\) introduce two new vertices \(x_{e,1}\) and \(x_{e,2}\) which are both made adjacent to the endvertices of \(e\). For each nonedge \(f\) of \(G\), we introduce one new vertex \(y_f\) which is made adjacent to the endvertices of \(f\). We again call the new vertices, which are the vertices of \(S\), special vertices.

**Lemma 4** \(\chi'(G) = 3 \iff eq(G^*) = n + 2\), where \(n\) is the number of vertices of \(G\).

**Proof.** The proof goes along the same lines as the proof of Lemma 3. Assume \(G\) can be edge-colored with three colors. Notice that \(eq(G^*) \geq n + 2\) since \(K_{1,n+2}\) is an induced subgraph. Since \(G\) is cubic, \(n\) is even. We can construct an equivalence covering for \(G^*\) as follows. First, consider an edge-coloring of \(K\) with \(n - 1\) colors (see [3]). For each color class, define an equivalence graph as follows. For each edge in \(K\) in that color class, add one special vertex at that edge and let that triangle be a clique of the equivalence graph.

Next consider an edge-coloring of \(G\) with three colors. For each color class define an equivalence graph as follows. For each edge in that color class add the other special vertex and let that triangle be a clique of the equivalence graph.

Clearly, this defines an equivalence covering of \(G^*\) with \(n + 2\) equivalence graphs.

Assume that \(G^*\) has an equivalence covering with \(n + 2\) equivalence graphs. Consider a vertex \(a \in K\). This vertex \(a\) is adjacent to \(n+2\) special vertices, and each of the edges between \(a\) and a special vertex defines a unique equivalence graph. It follows that no triangle of \(G\) can be contained in a clique of an equivalence graph. We thus obtain a correct edge-coloring of \(G\) in the same manner as in the proof of Lemma 3. 

\[\square\]

**Corollary 2** It is NP-complete to determine whether the equivalence covering number of a splitgraph, in which every vertex of the independent set has degree two, is \(\Delta\) or \(\Delta + 1\), where \(\Delta = \max\{\delta(x) \mid x \in K\}\) for a given partition of the vertex set into a clique \(K\) and an independent set \(S\).

4 Concluding remarks

In this note we considered the equivalence covering number for splitgraphs. Related problems are the clique covering number, and the clique partition number. The clique covering number is the minimum number of cliques which cover all the edges of the graph. It was shown in [8] that the clique covering number can be computed in linear time for chordal graphs. The clique partition number is the minimum number of cliques such that every edge is contained in exactly one clique. Determining the clique partition number is NP-hard for chordal graphs [9]. It would be interesting to determine the complexity of the computation of the clique partition number for split graphs. It should be remarked however that it is unlikely that a polynomial time algorithm exists, due to the following [10, 7]. Consider the following splitgraph \(G\). Take a
clique with \( m^2 + m + 1 - r \) vertices and an independent set with \( r \) vertices. Make every vertex of the independent set adjacent to every vertex of the clique. (\( G \) is sometimes denoted as \( K_{m^2+m+1} \setminus K_r \).) If \( 2 < r < m^2 + m + 1 \) then the clique partition number of \( G \) is at least \( m^2 + m \) with equality holding if and only if a projective plane of order \( m \) exists and \( r = m + 1 \).

References


Computing Science Reports

In this series appeared:

91/01  D. Alstein

91/02  R.P. Nederpelt
        H.C.M. de Swart

91/03  J.P. Katoen
        L.A.M. Schoenmakers

91/04  E. v.d. Sluis
        A.F. v.d. Stappen

91/05  D. de Reus

91/06  K.M. van Hee

91/07  E. Poll

91/08  H. Schepers

91/09  W.M.P.v.d.Aals
t

91/10  R.C. Backhouse
        P.J. de Bruin
        P. Hoogendijk
        G. Malcolm
        E. Voermans
        J. v.d. Woude

91/11  R.C. Backhouse
        P.J. de Bruin
        G. Malcolm
        E. Voermans
        J. van der Woude

91/12  E. van der Sluis

91/13  F. Rietman

91/14  P. Lemmens

91/15  A.T.M. Aerts
        K.M. van Hee

91/16  A.J.J.M. Marcelis

Department of Mathematics and Computing Science
Eindhoven University of Technology


Implication. A survey of the different logical analyses "if...,then...", p. 26.

Parallel Programs for the Recognition of P-invariant Segments, p. 16.

Performance Analysis of VLSI Programs, p. 31.

An Implementation Model for GOOD, p. 18.

SPECIFICATIEMETHODEN, een overzicht, p. 20.

CPO-models for second order lambda calculus with recursive types and subtyping, p. 49.

Terminology and Paradigms for Fault Tolerance, p. 25.

Interval Timed Petri Nets and their analysis, p.53.

POLYNOMIAL RELATORS, p. 52.

Relational Catamorphism, p. 31.


A note on Extensionality, p. 21.

The PDB Hypermedia Package. Why and how it was built, p. 63.


An example of proving attribute grammars correct: the representation of arithmetical expressions by DAGs, p. 25.
Transforming Functional Database Schemes to Relational Representations, p. 21.

Transformational Query Solving, p. 35.

Some categorical properties for a model for second order lambda calculus with subtyping, p. 21.


assertional Data Reification Proofs: Survey and Perspective, p. 18.


Z and high level Petri nets, p. 16.

Formal semantics for BRM with examples, p. 25.

A compositional proof system for real-time systems based on explicit clock temporal logic: soundness and completeness, p. 52.

The GOOD based hypertext reference model, p. 12.

Embedding as a tool for language comparison: On the CSP hierarchy, p. 17.

A compositional proof system for dynamic process creation, p. 24.

Correctness of acceptor schemes for regular languages, p. 31.

An algebra for process creation, p. 29.

Some algorithms to decide the equivalence of recursive types, p. 26.


The modelling and analysis of queueing systems with QNM-ExSpect, p. 23.

Specifying fault tolerant programs in deontic logic, p. 15.
Asynchronous communication in process algebra, p. 20.

A note on compositional refinement, p. 27.

A compositional semantics for fault tolerant real-time systems, p. 18.

Real space process algebra, p. 42.

Program derivation in acyclic graphs and related problems, p. 90.

Conservative fixpoint functions on a graph, p. 25.

Discrete time process algebra, p. 45.

The fine-structure of lambda calculus, p. 110.

On stepwise explicit substitution, p. 30.


Composition and decomposition in a CPN model, p. 55.

Demonic operators and monotype factors, p. 29.


Set theory and nominalisation, Part II, p. 22.

The total order assumption, p. 10.

A system at the cross-roads of functional and logic programming, p. 36.

Integrity checking in deductive databases; an exposition, p. 32.

Interval timed coloured Petri nets and their analysis, p. 20.

A unified approach to Type Theory through a refined lambda-calculus, p. 30.

Axiomatizing Probabilistic Processes: ACP with Generative Probabilities, p. 36.

Are Types for Natural Language? P. 32.
92/21 F.Kamareddine  
Non well-foundedness and type freeness can unify the interpretation of functional application, p. 16.

92/22 R. Nederpelt  
F.Kamareddine  
A useful lambda notation, p. 17.

92/23 F.Kamareddine  
E.Klein  
Nominalization, Predication and Type Containment, p. 40.

92/24 M.Codish  
D.Dams  
Eyal Yardeni  
Bottom-up Abstract Interpretation of Logic Programs, p. 33.

92/25 E.Poll  
A Programming Logic for Foa, p. 15.

92/26 T.H.W.Beelen  
W.J.J.Stut  
P.A.C.Verkoulen  
A modelling method using MOVIE and SimCon/ExSpect, p. 15.

92/27 B. Watson  
G. Zwaan  
A taxonomy of keyword pattern matching algorithms, p. 50.

93/01 R. van Geldrop  
Deriving the Aho-Corasick algorithms: a case study into the synergy of programming methods, p. 36.

93/02 T. Verhoeff  
A continuous version of the Prisoner's Dilemma, p. 17

93/03 T. Verhoeff  
Quicksort for linked lists, p. 8.

93/04 E.H.L. Aarts  
J.H.M. Korst  
P.J. Zwietering  
Deterministic and randomized local search, p. 78.

93/05 J.C.M. Baeten  
C. Verhoef  
A congruence theorem for structured operational semantics with predicates, p. 18.

93/06 J.P. Veltkamp  
On the unavoidability of metastable behaviour, p. 29

93/07 P.D. Moerland  
Exercises in Multiprogramming, p. 97

93/08 J. Verhoosel  
A Formal Deterministic Scheduling Model for Hard Real-Time Executions in DEDOS, p. 32.

93/09 K.M. van Hee  

93/10 K.M. van Hee  
Systems Engineering: a Formal Approach Part II: Frameworks, p. 44.

93/11 K.M. van Hee  

93/12 K.M. van Hee  

93/13 K.M. van Hee  
Systems Engineering: a Formal Approach
93/14 J.C.M. Baeten  
J.A. Bergstra  

93/15 J.C.M. Baeten  
J.A. Bergstra  
R.N. Bol  

93/16 H. Schepers  
J. Hooman  
A Trace-Based Compositional Proof Theory for Fault Tolerant Distributed Systems, p. 27.

93/17 D. Alstein  
P. van der Stok  
Hard Real-Time Reliable Multicast in the DEDOS system, p. 19.

93/18 C. Verhoef  
A congruence theorem for structured operational semantics with predicates and negative premises, p. 22.

93/19 G-J. Houben  
The Design of an Online Help Facility for ExSpect, p.21.

93/20 F.S. de Boer  

93/21 M. Codish  
D. Dams  
G. Filité  
M. Bruynooghe  

93/22 E. Poll  
A Typechecker for Bijective Pure Type Systems, p. 28.

93/23 E. de Kogel  
Relational Algebra and Equational Proofs, p. 23.

93/24 E. Poll and Paula Severi  
Pure Type Systems with Definitions, p. 38.

93/25 H. Schepers and R. Gerth  

93/26 W.M.P. van der Aalst  
Multi-dimensional Petri nets, p. 25.

93/27 T. Kloks and D. Kratsch  
Finding all minimal separators of a graph, p. 11.

93/28 F. Kamareddine and R. Nederpelt  
A Semantics for a fine λ-calculus with de Bruijn indices, p. 49.

93/29 R. Post and P. De Bra  
GOLD, a Graph Oriented Language for Databases, p. 42.

93/30 J. Deogun  
T. Kloks  
D. Kratsch  
H. Müller  
On Vertex Ranking for Permutation and Other Graphs, p. 11.

93/31 W. Körver  
Derivation of delay insensitive and speed independent CMOS circuits, using directed commands and production rule sets, p. 40.

93/32 H. ten Eikelder and H. van Geldrop  
93/33 L. Loyens and J. Moonen  ILIAS, a sequential language for parallel matrix computations, p. 20.

93/34 J.C.M. Baeten and J.A. Bergstra  Real Time Process Algebra with Infinitesimals, p.39.


93/36 J.C.M. Baeten and J.A. Bergstra  Non Interleaving Process Algebra, p. 17.


93/38 C. Verhoef  A general conservative extension theorem in process algebra, p. 17.


93/41 A. Bijlsma  Temporal operators viewed as predicate transformers, p. 11.

93/42 P.M.P. Rambags  Automatic Verification of Regular Protocols in P/T Nets, p. 23.

93/43 B.W. Watson  A taxonomy of finite automata construction algorithms, p. 87.

93/44 B.W. Watson  A taxonomy of finite automata minimization algorithms, p. 23.


93/48 R. Gerth  Verifying Sequentially Consistent Memory using Interface Refinement, p. 20.
<table>
<thead>
<tr>
<th>Paper No.</th>
<th>Authors</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>94/01</td>
<td>P. America M. van der Kammen R.P. Nederpelt O.S. van Roosmalen H.C.M. de Swart</td>
<td>The object-oriented paradigm, p. 28.</td>
</tr>
<tr>
<td>94/02</td>
<td>F. Kamareddine R.P. Nederpelt</td>
<td>Canonical typing and Π-conversion, p. 51.</td>
</tr>
<tr>
<td>94/04</td>
<td>J.C.M. Baeten J.A. Bergstra</td>
<td>Graph isomorphism Models for Non Interleaving Process Algebra, p. 18.</td>
</tr>
<tr>
<td>94/06</td>
<td>T. Basten T. Kunz J. Black M. Coffin D. Taylor</td>
<td>Time and the Order of Abstract Events in Distributed Computations, p. 29.</td>
</tr>
<tr>
<td>94/08</td>
<td>O.S. van Roosmalen</td>
<td>A Hierarchical Diagrammatic Representation of Class Structure, p. 22.</td>
</tr>
<tr>
<td>94/09</td>
<td>J.C.M. Baeten J.A. Bergstra</td>
<td>Process Algebra with Partial Choice, p. 16.</td>
</tr>
<tr>
<td>94/10</td>
<td>T. verhoeff</td>
<td>The testing Paradigm Applied to Network Structure. p. 31.</td>
</tr>
<tr>
<td>94/13</td>
<td>R. Seljée</td>
<td>A New Method for Integrity Constraint checking in Deductive Databases, p. 34.</td>
</tr>
<tr>
<td>94/14</td>
<td>W. Peremans</td>
<td>Ups and Downs of Type Theory, p. 9.</td>
</tr>
<tr>
<td>94/16</td>
<td>R.C. Backhouse H. Doombos</td>
<td>Mathematical Induction Made Calculational, p. 36.</td>
</tr>
<tr>
<td>Paper Number</td>
<td>Authors</td>
<td>Title</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>94/18</td>
<td>F. Kamareddine R. Nederpelt</td>
<td>Refining Reduction in the Lambda Calculus, p. 15.</td>
</tr>
<tr>
<td>94/19</td>
<td>B.W. Watson</td>
<td>The performance of single-keyword and multiple-keyword pattern matching algorithms, p. 46.</td>
</tr>
<tr>
<td>94/20</td>
<td>R. Bloo F. Kamareddine R. Nederpelt</td>
<td>Beyond β-Reduction in Church's λ→, p. 22.</td>
</tr>
<tr>
<td>94/22</td>
<td>B.W. Watson</td>
<td>The design and implementation of the FIRE engine: A C++ toolkit for Finite automata and regular Expressions.</td>
</tr>
<tr>
<td>94/23</td>
<td>S. Mauw and M.A. Reniers</td>
<td>An algebraic semantics of Message Sequence Charts, p. 43.</td>
</tr>
<tr>
<td>94/25</td>
<td>T. Kloks</td>
<td>K₁₅-free and W₁⁴-free graphs, p. 10.</td>
</tr>
<tr>
<td>94/29</td>
<td>J. Hooman</td>
<td>Correctness of Real Time Systems by Construction, p. 22.</td>
</tr>
<tr>
<td>94/30</td>
<td>J.C.M. Baeten J.A. Bergstra Gh. Ştefănescu</td>
<td>Process Algebra with Feedback, p. 22.</td>
</tr>
<tr>
<td>94/33</td>
<td>T. Laan</td>
<td>A formalization of the Ramified Type Theory, p.40.</td>
</tr>
<tr>
<td>94/35</td>
<td>J.C.M. Baeten S. Mauw</td>
<td>Delayed choice: an operator for joining Message Sequence Charts, p. 15.</td>
</tr>
<tr>
<td>94/36</td>
<td>F. Kamareddine R. Nederpelt</td>
<td>Canonical typing and Π-conversion in the Barendregt Cube, p. 19.</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>R. Bol</td>
<td></td>
</tr>
<tr>
<td></td>
<td>M. Voorhoeve</td>
<td></td>
</tr>
<tr>
<td>94/38</td>
<td>A. Bijlsma</td>
<td>Point-free substitution, p. 10.</td>
</tr>
<tr>
<td></td>
<td>C.S. Scholten</td>
<td></td>
</tr>
</tbody>
</table>