Parameterized Complexity of the Clique Partition Problem

Egbert Mujuni[*1] **Frances Rosamond**[2]

[1] Department of Mathematics
University of Dar-es-Salaam
Box 35062, Dar es Salaam
Tanzania
Email: emujuni@maths.udsm.ac.tz

[2] Parameterized Complexity Research Unit
University of Newcastle
Callaghan, Australia
Email: frances.rosamond@newcastle.edu.au

Abstract

The problem of deciding whether the edge-set of a given graph can be partitioned into at most k cliques is well known to be NP-complete. In this paper we investigate this problem from the point of view of parameterized complexity. We show that this problem is fixed parameter tractable if we choose the number of cliques as parameter. In particular, we show that in polynomial time, a kernel bounded by k^2 can be obtained, where k is the number of cliques. We also give an $\mathcal{O}(2^{((k+3)\log k)/2}n)$ algorithm for this problem in K_4-free graphs.

1 Introduction

The problem of finding a minimum set that covers or partitions the edge-set of a given graph arises in many applications (e.g., see (15)). The problem is defined as follows: Let G be a graph. A set $\mathcal{S} = \{G_1, G_2, \ldots, G_k\}$, $k \geq 1$ of subgraphs of G is called a covering of G if $E(G) = \cup_{i=1}^{k} E(G_i)$. If each element of $\mathcal{S}$ is a clique, then $\mathcal{S}$ is called a *clique cover* of G. A *clique partition* is a clique cover $\mathcal{S}$ in which each edge belongs to exactly one member of $\mathcal{S}$; that is, for two distinct $C, C' \in \mathcal{S}$ it follows that $E(C) \cap E(C') = \emptyset$. The clique partition problem asks whether a given graph G has a clique partition of size at most k.

The clique partition problem is known to be NP-complete in general graphs (14). The problem remains NP-complete even for K_4-free graphs (16).

In this paper we investigate the parameterized complexity of this problem using the framework developed by Downey and Fellows (5). Here we give a quick review of parameterized complexity theory. For a detailed discussion we refer the reader to (5) or (13). In parameterized complexity theory, we consider the input of an instance of a parameterized problem as consisting of two parts; that is, a pair (I, k), where I is the main input and k (usually an integer) is a parameter. We say a problem of size n and parameter k is *fixed parameter tractable* if the problem can be solved in time $\mathcal{O}(f(k)n^c)$, where f denotes a computable function and c denotes a constant which is independent of the parameter k. Therefore, a parameterized algorithm may provide an efficient solution to a problem whose parameter is reasonably small.

Clustering problems have wide applicability (See for example, (2; 7; 8; 11)). The problem (EDGE) CLIQUE COVER in general graphs is an important NP-complete problem that has received considerable attention. Clique Cover in general graphs is hard to approximate in polynomial time and nothing better than a polynomial approximation factor is known (1). However Gramm *et al.* (See (9) and also (10)) show that CLIQUE COVER is fixed-parameter tractable with respect to the parameter k, the number of cliques, and has a kernel of size 2^k.

CLIQUE COVER
Instance : A graph $G = (V, E)$
Parameter : An integer k
Question : Is there a set of at most k cliques in G such that each edge in E has both its endpoints in at least one of the selected cliques?

Gramm *et al.* (9; 10) describe an exact algorithm based on search tree techniques. Combining their kernelization rules with a sophisticated search tree algorithm, they were able to obtain an FPT algorithm for CLIQUE COVER that can solve problem instances on graphs of several hundred vertices efficiently.

The key difference between CLIQUE COVER and our problem, CLIQUE PARTITION, is whether the cliques share edges or not. Although we have not implemented our kernelization rules, they are polynomial-time data reduction techniques similar to those of Gramm *et al.* that significantly shrink the input, and then for the reduced instances one can use search tree, exhaustive search or other algorithms to efficiently find optimal solutions in reasonable time.

More formally, in this paper we study following parameterized problem:

CLIQUE PARTITION
Instance : A graph $G = (V, E)$
Parameter : An integer k
Question : Is there a set of at most k cliques in G such that each edge in E has both its endpoints in exactly one of the selected cliques?

We develop a set of reduction rules that in polynomial time replace a given CLIQUE PARTITION instance (G, k) consisting of a graph G and a nonnegative integer k by a "simpler" instance (G', k') such

[*] The reseach has been supported by International Science Programme (ISP) of Sweden, under the project titled "The Eastern African Universities Mathematics Programme (EAUMP)".

that (G,k) has a solution iff (G',k') has a solution. An instance to which none of the reduction rules applies is called "reduced" with respect to these rules. A parameterized problem such as CLIQUE PARTITION (the parameter is k) is said to have a problem kernel if, after the application of the reduction rules, the reduced instance has size $f(k)$ for a function f depending only on k. It is a well-known result from parameterized complexity theory that the existence of a problem kernel implies fixed-parameter tractability for a parameterized problem (5; 13) and (10).

Main Results: In this paper we show that CLIQUE PARTITION has a kernel bounded by k^2, hence it is fixed parameter tractable. We also give an $\mathcal{O}(2^{((k+3)\log k)/2}n)$ algorithm for K_4-free graphs.

Notations: All graphs considered in this paper are undirected finite graphs without loops and multiple edges. Let $G=(V,E)$ be a graph. A *clique* is complete subgraph of G that is not necessarily maximal. The set of neighbours of a vertex v is denoted by $N(v)$, and we set $N[v]=N(v)\cup\{v\}$. For $T\subset V$, we set $N(T):=\bigcup_{v\in T}N(v)$. If $V'\subseteq V$, we denote by $G[V']$ the subgraph of G induced by V'. We refer the reader to (3) for graph theoretic terminology not defined in this paper.

2 Kernelization

In this section we present a set of reduction rules which leads to a problem kernel consisting of at most k^2 vertices. We show that if these rules are not applicable to an instance (G,k) of CLIQUE PARTITION and G has more than k^2 vertices then we conclude that G does not have a clique partition of size at most k.

Definition 1 *A kernelization for a parameterized problem $\mathcal{L}$ is a transformation which maps an instance (I,k) onto (I',k') (which is called a problem kernel) such that:*

1. *$k'\le k$ and $|I'|\le g(k)$ for some computable function g*
2. *The transformation from (I,k) onto (I',k') is computable in polynomial time.*
3. *(I,k) is a yes-instance of $\mathcal{L}$ if and only if (I',k') a yes-instance of $\mathcal{L}$.*

The function $g(k)$ is called the *size of a kernel* for $\mathcal{L}$. The following result is well known.

Lemma 2 *(6) A parameterized problem is fixed-parameter tractable if and only if it has a kernelization.*

We first present simple reduction rules that can be easily applied to simplify an instance of the CLIQUE PARTITION problem; trivially we may assume that $k>1$.

- **Rule 1:** Given an instance (G,k) of CLIQUE PARTITION and a vertex $v\in V(G)$ of degree 0, then the answer to (G,k) is yes if and only if $(G-v,k)$ is yes.
- **Rule 2:** Given an instance (G,k) of CLIQUE PARTITION and a vertex $v\in V(G)$ of degree 1, then the answer to (G,k) is yes if and only if $(G-v,k-1)$ is yes.
- **Rule 3:** Given an instance (G,k) of CLIQUE PARTITION and an edge $e=uv\in E(G)$ such that $N(u)\cap N(v)=\emptyset$, then the answer to (G,k) is yes if and only if $(G-\{e\},k-1)$ is yes.

Clearly, the following is true.

Lemma 3 *Rules 1-3 are correct and they can be executed in $\mathcal{O}(n^2)$ time, where n is the number of vertices of the input graph.*

Definition 4
- *A clique partition $\mathcal{S}$ of a complete graph G is said to be trivial $\mathcal{S}$ if it consists of a single clique.*
- *Let G be a complete graph. Denote by $\rho(G)$ the cardinality of a minimum non-trivial clique partition of G.*

Lemma 5 *Suppose G is a complete graph on n vertices. Then $\rho(G)=n$.*

Lemma 5 is just a corollary of the following result of de Bruijn and Erdos (1948), which was stated in terms of set theory.

Theorem 6 *(4) Suppose $A_1,\dots,A_m$ are subsets of the set $A=\{a_1,\dots,a_n\}$, and that $A_i\ne A$, $1\le i\le m$. If each pair $\{a_r,a_s\}$ occurs in one and only one A_i, then $m\ge n$, and equality holds if and only if either (1) $A_1=\{a_1,\dots,a_{n-1}\}$, $A_2=\{a_1,a_n\},\dots,A_n=\{a_{n-1},a_n\}$ or (2) n is of the form $n=k(k+1)+1$ and all the A_i's have precisely $k+1$ elements, and each a_j occurs in exactly $k+1$ of the A_i's, $1\le i\le m$, $1\le j\le n$.*

In the terminology of graph theory, Theorem 6 says that if $\mathcal{S}$ is a non-trivial clique partition of the edges of K_n, then $|\mathcal{S}|\ge n$ and equality holds if and only if $\mathcal{S}$ consists of one clique on $n-1$ vertices and $n-1$ copies of K_2 incident with a single vertex of K_n, or n is of the form $n=k^2+k+1$ and $\mathcal{S}$ consists of n copies of K_{k+1}, where each vertex of K_n belongs to exactly $k+1$ cliques of $\mathcal{S}$.

Lemma 7 *Let G be a graph. Let $\mathcal{S}$ be a clique partition of G of size k. If $G'\subseteq G$ is a complete subgraph of G on more than k vertices, then there is an element $C\in\mathcal{S}$ such that $G'\subseteq C$.*

Proof: If the edges of G' are covered by more than one clique of $\mathcal{S}$, then by Lemma 5, they must be covered by more than k cliques. This implies in turn that $|\mathcal{S}|>k$, which is a contradition to the hypothesis. $\diamond$

With this lemma at hand, we can state the following reduction rule.

- **Rule 4:** Let (G,k) be an instance of CLIQUE PARTITION. Suppose that v is a vertex such that $|N[v]|>k$ and the graph G^* induced by $N[v]$ is a clique, then the answer to (G,k) is yes if and only if (G',k') is yes, where $G'=G-v-E(G^*)$ and $k'=k-1$.

Lemma 8 *Rule 4 is correct.*

Proof: First suppose that (G,k) is true. Let $\mathcal{S}=\{C_1,\dots,C_m\}$, $m\le k$, be a clique partition of G, and let G^* be as defined in Rule 4. It follows that G^* is a maximal clique. Since G^* is a clique on more than k vertices, Lemma 7 implies that there is an index i such that $G^*\subseteq C_i$. However, since G^* is a maximal clique, it follows that $G^*=C_i$. Therefore, $\mathcal{S}-\{C_i\}$ is a clique partition of G' with $m-1\le k-1$ elements; i.e., (G',k') is true.

Now suppose that $(G',k-1)$ is true. Then (G,k) is true because G is the edge-disjoint union of G' and G^*, and G^* is a complete graph. $\diamond$

We say that an instance (G,k) of CLIQUE PARTITION is reduced (with respect to Rules 1-4) if none of the reduction rules can be applied.

Theorem 9 *Suppose an instance (G, k) of CLIQUE PARTITION is reduced and that it does have a solution of size at most k. Then G has at most k^2 vertices.*

Proof: Suppose a reduced instance (G, k) of CLIQUE PARTITION has the answer yes. Let $\mathcal{S}$ be a clique partition of G of size at most k. We claim that each element of $\mathcal{S}$ has at most k vertices.

Suppose to the contrary that there is a clique $C \in \mathcal{S}$ with more than k vertices. Since Rule 4 is not applicable, each vertex of C has a neighbour in G which does not belong to C. This implies that each vertex of C belongs to another unique clique in $\mathcal{S}$. However, by the definition of the CLIQUE PARTITION problem, if C_x and C_y are cliques in $\mathcal{S} - \{C\}$ containing $x, y \in C$, respectively, $x \neq y$, then $C_x \neq C_y$. Thus $|\mathcal{S}| > k$, which is a contradition. We may now conclude that each element of $\mathcal{S}$ has at most k vertices.

Note that $\mathcal{S}$ covers the vertices of G, since Rule 1 is not applicable. Therefore

$$|V(G)| \leq \sum_{C \in \mathcal{S}} |V(C)| \leq k \times \max\{|C| : C \in \mathcal{S}\} \leq k^2$$

This completes the proof the theorem.

$\diamond$

Remark 10 As can be seen from the proof of Theorem 9, Rule 2 and Rule 3 have no impact there (i.e., Theorem 9 remains true if we restrict the reductions to applying Rules 1 and 4 only). However, we included Rule 2 and Rule 3 because they may be used to reduce the size of the input graph, and hence speedup the computations. For example, Gramm *et al.*(9) experimented with disabling one of their more complicated and expensive rules. They found that for larger cover sizes over 80, the rule nearly doubles the range of instances that can be solved smoothly and is clearly worthwhile.

As a consequence of Theorem 9, we have

Corollary 11 *CLIQUE PARTITION is fixed parameter tractable.*

3 Algorithm

In the previous section it was shown that CLIQUE PARTITION has a kernel of size k^2. We now describe an algorithm that decides CLIQUE PARTITION. The algorithm is based on bounded search tree. We proceed as follows. Let instance (G, k) be a reduced instance of CLIQUE PARTITION. We choose an edge uv such that $|N(u) \cap N(v)|$ is minimum, and then enumerate a set $\mathcal{S}$ of all cliques in the graph induced by $N(u) \cap N(v)$. Branch according to the elements of $K \in \mathcal{S}$ by adding the clique K' induced by $\{u, v\} \cup V(K)$ to the clique partition, and we set $G := G - E(K')$. The recursion stops as soon as a solution is found or k cliques are generated without finding a solution. This algorithm is presented Figure 1.

We analyze the algorithm in Figure 1 as follows. Let n and Δ be the number of vertices and maximum degree of G, respectively. $|N(u) \cap N(v)| < \Delta$. So $|\mathcal{S}| \in \mathcal{O}(2^{\Delta})$. Thus, each non-leaf node in the searching tree has at most $\mathcal{O}(2^{\Delta})$ children. Since the depth of tree is bounded by k and we can test each leaf in linear time, the algorithm computes the solution in $\mathcal{O}(2^{\Delta k} n)$. Note that, since G is reduced, $\Delta \leq k^2$.

```
C_Partition(Graph G, Set 𝒞, Integer k)
begin
  Reduce(G,k).
  if E(G) = ∅ Return TRUE.
  else if k = 0 Return FALSE.
  else
    choose uv ∈ E(G) such that
       |N(u) ∩ N(v)| is minimum.
    find a set 𝒮 of all clique in
      G[N(u) ∩ N(v)].
    for each K ∈ 𝒮
         K' := V(K') ∪ {u,v}.
         𝒞' := 𝒞 ∪ {G[K']}.
         k' := k − 1.
         H := G − E(K').
         if C_Partition (H, 𝒞', k')
            Return TRUE.
    end for
  Return FALSE.
end.
```

Figure 1: Algorithm for CLIQUE PARTITION in general graphs.

4 K_4-Free Graphs

As mentioned in the introduction, the classical decision version of the CLIQUE PARTITION problem remains NP-hard even for K_4-free graphs (16). We now present a fixed-parameter algorithm for CLIQUE PARTITION in this class of graphs. First note that any non-trivial clique in this class of graphs must be K_2 or K_3.

Observation 12 *Let (G, k) be an instance of CLIQUE PARTITION, where G is K_4-free. If G contains an edge uv such that $|N(u) \cap N(v)| > \frac{k+1}{2}$, then G does not have a clique partition of G of size at most k.*

Theorem 13 *Let G be a K_4-free graph. Then the CLIQUE PARTITION problem can be solved in $\mathcal{O}(2^{((k+3)\log k)/2} n)$ time, where n is the number of vertices of G.*

Proof: We construct a bounded search tree T of height k and each node of T has at most $k+1$ children. Each node is associated with a set $\mathcal{C}$ of edge-disjoint cliques, a subgraph $H = G - \bigcup_{C \in \mathcal{C}} E(C)$ and $k' = k - |\mathcal{C}|$, where $\mathcal{C}$ is a partial clique cover constructed at each step of the algorithm. For the root, we have $H := G$, $\mathcal{C} := \emptyset$ and $k' := k$.

We recursively proceed as follows: At a node i, we choose an edge $uv \in E(H)$. If $|N_H(u) \cap N_H(v)| > \frac{k'+1}{2}$, we stop searching in this branch as we know H does not have a clique partition of size at most k', by Observation 12. Otherwise, we create a child for uv and one child for each vertex $w \in N_H(u) \cap N_H(v)$. Thus, this node has at most $\frac{k'+1}{2} + 1 = \frac{k'+3}{2}$ children.

Repeat this expansion for each child node, using the depth-searching strategy. Note that since we add one clique to the partial solution $\mathcal{C}$ at each expansion, the size of $\mathcal{C}$ at level l is also l. The recursion stops as soon as a solution is found or k cliques are generated without finding a solution. T has at most $k^{(k+3)/2} = 2^{(k+3)\log k/2}$ nodes. At each node we need $\mathcal{O}(n)$ time to compute the set $N(u) \cap N(v)$. Hence, the total running time is $\mathcal{O}(2^{((k+3)\log k)/2}n)$. $\diamond$

Proof of the above theorem yields a fixed parameter algorithm for the CLIQUE PARTITION problem in the class of K_4-free graphs. The algorithm is given in Figure 2.

```
C_Partition(K4-free G, Set C, Integer k)
begin
  if E(G) - E(C) = ∅ Return TRUE.
  else if k = 0 Return FALSE.
  else
    choose an edge uv ∈ E(G)
    if |N(u) ∩ N(v)| > (k+1)/2
       Return FALSE.
    else
       C' := C ∪ {{u,v}}.
       k' := k - 1.
       H := G - {uv}.
       if Clique_Partition (H, C', k')
          Return TRUE.
       else
         for each w ∈ N(u) ∩ N(v)
            C' := C ∪ {{u,v,w}}.
            k' := k - 1.
            H := G - {uv,uw,vw}.
            if C_Partition (H, C', k')
               Return TRUE.
          end for
  Return FALSE.
end.
```

Figure 2: Algorithm for CLIQUE PARTITION in K_4-free graphs.

5 Concluding Remarks

We have obtained the first fixed-parameter tractability result for the clique partition problem, when the number of cliques is the parameter. It would be interesting to improve our algorithm for clique partition. The parameterized complexity hierarchy:

$$P \subseteq Lin(\mathrm{k}) \subseteq Poly(\mathrm{k}) \subseteq FPT \subseteq W[1]\ldots$$

leads to the natural question of whether CLIQUE PARTITION is in Lin(k), or to show that a kernel of linear size in k is not possible. It is interesting that k-CLIQUE COVER is probably not in Poly(k) even though it is in FPT.

Acknowledgments

We are gratefully acknowledge the assistance of Mike Fellows, who originally suggested the problem and provided valuable discussions. We thank Hebert Fleischner for stimulating discussions and providing various references. This paper was done while the first author was enjoying the hospitality of the Vienna Technical University, Austria, which is gratefully acknowledged.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties. Springer, 1999.

[2] M. Benson, L. Carlsson, G. Guillot, M. Jernis, M. A. Langston, M. Rudemo, and B. Andersson. A network-based analysis of allergen-challenged CD4+ T cells from patients with allergic rhinitis. *Genes and Immunity* 7 (2006) 514–521.

[3] G. Chartrand and L. Lesniak. *Graphs & Digraphs.* Chapman&Hall, third edition, 1991.

[4] N.G. de Bruijn and P. Erdos. On a combinatorial problem. *Indag. Math. 10*, pages 421–423, 1948.

[5] R. Downey and M. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.

[6] R. Downey, M. Fellows, and U. Stege. Parameterized complexity: A framework for systematically confronting computational intractability. *AMS-DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 49*, pages 49–99, 1999.

[7] J.D. Eblen, I.C. Gerling, A.M. Saxton, J. Wu, J.R. Snoddy, and M.A. Langston. Graph Algorithms for Integrated Biological Analysis, with Applications to Type 1 Diabetes Data. *Clustering Challenges in Biological Networks*, W. A. Chaovalitwongse, ed., World Scientific (2007).

[8] M.R. Fellows, M.A. Langston, F.A. Rosamond and P. Shaw. Efficient parameterized preprocessing for cluster editing. *Proc. FCT 2007*, Springer Verlag, *Lecture Notes in Computer Science* 4598 (2007) 312–321.

[9] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Data reduction, exact, and heuristic algorithms for Clique Cover. *Proc. 8th ACM-SIAM ALENEX*, ACM-SIAM (2006) 86–94. Long version to appear in *The ACM Journal of Experimental Algorithmics.*

[10] J. Guo and R. Niedermeier. Guest column: Invitation to data reduction and kernelization. *ACM SIGACT NEWS* 38 (March 2007) 31–45.

[11] M.A. Langston, A.D. Perkins, D.J. Beare, R.W. Gauldie, P.J. Kershaw, J.B. Reid, K. Winpenny, and A.J. Kenny. Combinatorial Algorithms and High Performance Implementations for Elucidating Complex Ecosystem Relationships from North Sea Historical Data. *Proceedings of the International Council for the Exploration of the Sea Annual Science Conference* (2006).

[12] W. Moon and L. Moser. On cliques in graphs. *Israel. J. Math. 3*, pages 23–28, 1965.

[13] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.

[14] J. Orlin. Contentment in graph theory : Covering graphs with cliques. *Indagationes Math. 39*, pages 406–424, 1977.

[15] F.S. Roberts. Applications of edge coverings by cliques. *Discrete Appl. Math. 10*, pages 93–109, 1985.

[16] Ma Shaohan, W.D. Wallis, and Wu Ju Lin. The complexity of the clique partition number problem. *Nineteenth Southeastern Conference on Combinatorics, Graph Theory, and Computing (Baton Rouge, LA, 1988). Congr. Numer. 67*, pages 59–66, 1988.