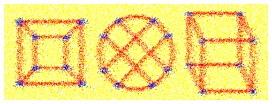

Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 17, no. 2, pp. 129–146 (2013)
DOI: 10.7155/jgaa.00288

Parameterized Algorithms for (r, l)-Partization

R. Krithika[1] *N.S. Narayanaswamy*[1]

[1]Department of Computer Science and Engineering,
Indian Institute of Technology Madras, India

Abstract

We consider the (r, l)-*Partization* problem of finding a set of at most k vertices whose deletion results in a graph that can be partitioned into r independent sets and l cliques. Restricted to perfect graphs and split graphs, we describe sequacious fixed-parameter tractability results for $(r, 0)$-*Partization*, parameterized by k and r. For (r, l)-*Partization* where $r + l = 2$, we describe an $O^*(2^k)$ algorithm for perfect graphs. We then study the parameterized complexity hardness of a generalization of the *Above Guarantee Vertex Cover* by a reduction from (r, l)-*Partization*.

Keywords: Parameterized complexity, Odd cycle transversal, r-Partization, Bicochromatization, Perfect graphs, Split graphs, Generalized above guarantee vertex cover.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
April 2012	July 2012	August 2012	December 2012	December 2012

Published: January 2013

Article type: Regular Paper

Communicated by: S.-i. Nakano

A preliminary version of this paper appeared as *Generalized Above Guarantee Vertex Cover and r-Partization* in the proceedings of WALCOM 2012

E-mail addresses: krithika@cse.iitm.ac.in (R. Krithika) swamy@cse.iitm.ac.in (N.S. Narayanaswamy)

1 Introduction

We consider only simple, finite, connected and undirected graphs. Given an undirected graph G and integers k, r and l, we define the (r, l)-*Partization* problem as follows.

(r, l)-*Partization*
Input: $\langle G, k, r, l\rangle$ where k, r and l are non-negative integers
Parameter: k, r, l
Question: Does there exist $S \subseteq V(G)$, $|S| \leq k$ such that $V(G \setminus S)$ can be partitioned into l cliques and r independent sets?

The (r, l)-*Partization* problem generalizes many classical problems. For instance, *Vertex Cover* is $(1, 0)$-*Partization*, *Odd Cycle Transversal (OCT)* is $(2, 0)$-*Partization* and r-*Coloring*, *Clique Cover* are $(r, 0)$-*Partization*, $(0, l)$-Partization, respectively, with $k = 0$ [GT1, GT21, GT4, GT15 in [12]]. *Vertex Cover* and *OCT* have been widely studied in the parameterized complexity framework. *Vertex Cover* was one of the first problems to be shown as fixed-parameter tractable (FPT). Research on this problem has led to new research directions in parameterized complexity [18]. An OCT of a graph is a subset of vertices whose deletion makes the graph bipartite. The fixed-parameter tractability of *OCT* was shown only quite recently [32]. Interestingly, none of the already known techniques for designing FPT algorithms were able to ascertain the fixed-parameter tractability of *OCT*. A technique referred to as *iterative compression* was introduced for this purpose. This was a breakthrough in parameterized algorithmics and led to the design of FPT algorithms for many other problems [17]. The algorithm for *OCT* reported in [32] runs in $O^*(3^k)$ time. After nearly a decade, this bound was improved to $O^*(2.3146^k)$ [24].

The r-*Coloring* problem can be thought of as partitioning the graph into r color classes with each class being an independent set. Clearly, if a graph is r-colorable, then its complementary graph can be partitioned into r cliques. As a natural generalization, the r-*Cocoloring* problem is to determine if a graph can be partitioned into r sets each of which is a clique or an independent set. The cochromatic number $z(G)$ of G is the minimum number r of sets (each of which is either a clique or an independent set) that $V(G)$ can be partitioned into. Brandstädt showed that determining if $V(G)$ can be partitioned into $r \leq 2$ independent sets and $l \leq 2$ cliques is polynomial-time solvable [3]. However, the problem is NP-complete if either $r \geq 3$ or $l \geq 3$ as it includes 3-*Coloring* as a special case. Thus, (r, l)-*Partization* is trivially para-NP-hard (i.e. NP-hard even for constant k). Further, $(r, 0)$-Partization, which we subsequently refer to as r-*Partization*, is at least as hard as r-*Coloring* which is W-hard when parameterized by the number of colors r [28]. Thus, on general graphs, r-*Partization* is W-hard when parameterized by k and r.

Perfect graphs is a well-studied special graph class that subsumes many interesting graph classes like chordal graphs, split graphs, bipartite graphs, comparability graphs and parity graphs [14]. Perfect graphs can be recognized in

polynomial time [5] and problems such as *Maximum Independent Set*, *Minimum Coloring* and *Maximum Clique* are polynomial-time solvable using the classical algorithm of Grötschel, Lovász and Schrijver [15]. None of these polynomial-time solutions are purely combinatorial. They employ polyhedral combinatorics and semidefinite programming. Recently, on perfect graphs, (r,l)-*Partization* with $k = 0$ was shown to be FPT with $r + l$ as the parameter [20]. Also, r-*Partization* is known to be NP-complete on split graphs in which *Vertex Cover* and *OCT* are polynomial-time solvable. Further, for each fixed r, r-*Partization* is polynomial-time solvable on chordal graphs using dynamic programming in $O(n^r)$ time [6, 35]. For perfect graphs, *Vertex Cover* is polynomial-time solvable using polyhedral combinatorics [16] and *OCT* is NP-complete by a recent result [2]. Thus, r-*Partization* on perfect graphs is W-hard when parameterized by r.

Our Results: On perfect graphs and split graphs, we show that r-*Partization* is FPT with respect to k and r as parameters. The algorithms have run-times $O^*((r+1)^k)$ on perfect graphs and $O^*(2^{k+r})$ on split graphs. For perfect graphs, on which *OCT* is known to be NP-complete, we describe an $O^*(2^k)$ algorithm using iterative compression. The property of perfectness of the graph enables an algorithm better than the $O^*(2.3146^k)$ time algorithm reported in [24] for general graphs. Also, we study (r,l)-*Partization* in perfect graphs when $r + l = 2$. We refer to this problem as *Bicochromatization*.
Roadmap: In Section 2, we study r-*Partization* in perfect graphs and split graphs. In Section 3, we show that *Bicochromatization* on perfect graphs admits an $O^*(2^k)$ algorithm. Finally, in Section 4, we explore the relationship between (r,l)-*Partization* and *Vertex Cover*.

Graph Theoretic Preliminaries: Graph theoretic terminologies are as defined in [14, 34]. For a graph G, $V(G)$ and $E(G)$ denote the vertex set and edge set, respectively. Let $|V(G)| = n$, $|E(G)| = m$ and $N_G(v) = \{u \mid \{u,v\} \in E(G)\}$. The graph G is referred to as an r-partite graph if $V(G)$ can be partitioned into r independent sets. The clique number $\omega(G)$ and the independence number $\alpha(G)$ are the cardinalities of a largest clique and independent set, respectively, in G. The size of a maximum matching in G is denoted as $\mu(G)$ and the chromatic number of G is denoted by $\chi(G)$. Also, G is said to be r-colorable if $\chi(G) \leq r$.

Clearly, $\chi(G) \geq \omega(G)$. A graph G in which for every induced subgraph H, $\chi(H) = \omega(H)$ holds, is called perfect. It is also known that G is perfect if and only if $\overline{G}$ is perfect. One of the first classes of graphs to be recognized as being perfect is the class of chordal (triangulated) graphs [14]. Chordal graphs are graphs in which every induced cycle is a triangle and co-chordal graphs are the corresponding complementary graphs. A graph G is a spilt graph if $V(G)$ can be partitioned into a clique and an independent set. Also, G is split if and only if G and $\overline{G}$ are both chordal. Since an independent set in G is a clique in $\overline{G}$ and vice-versa, it follows that G is a split graph if and only if $\overline{G}$ is also a split graph. Further, a split graph is a $2K_2$-free chordal graph.

Parameterized Complexity Preliminaries: The goal in parameterized complexity is to identify parameters that cause the inherent hardness of NP-complete problems and design algorithms with run-times bounded by $f(k)|x|^{O(1)}$, where x is the input instance and f is any arbitrary computable function dependent only on the problem-specific parameter k. The running time $f(k)|x|^{O(1)}$ of a parameterized algorithm is generally denoted as $O^*(f(k))$ by suppressing the polynomial terms. Parameterized problems for which such algorithms exist are referred to as fixed-parameter tractable (FPT). For a parameterized problem, a kernelization algorithm is a polynomial-time pre-processing procedure that transforms an instance of the problem into an equivalent instance, called a problem kernel, whose size depends only on the input parameter(s). By obtaining a kernel, the problem is clearly FPT. An active area of research in parameterized algorithmics is to obtain as small kernels as possible or establish the infeasibility of such kernels under standard complexity theoretic assumptions.

A central tool in the analysis of parameterized and exact exponential algorithms is the Exponential-Time Hypothesis: 3-SAT cannot be solved in $2^{o(n)}$ time [21]. By reductions preserving subexponential time complexities, problems like *3-Coloring* and *Maximum Independent Set* do not have subexponential time algorithms under Exponential-Time Hypothesis [22]. In order to classify parameterized problems as fixed-parameter tractable (denoted as FPT) or intractable (referred to as W-hard), Downey and Fellows [9] defined the W-hierarchy $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{W}[P] \subseteq \mathsf{XP}$ using Boolean circuits and the notion of parameterized reductions. A parameterized problem P is parameterized reducible to a parameterized problem Q if there is a FPT algorithm that transforms an instance $\langle x, k\rangle$ of P into an instance $\langle x', k'\rangle$ of Q such that $\langle x, k\rangle \in P \Leftrightarrow \langle x', k'\rangle \in Q$ and $k' = g(k)$ for some function g. A parameterized problem P to which all problems in W[1] can be reduced by a parameterized reduction is referred to as W[1]-hard and believed not to be FPT. W[1]-hard problems in W[1] are called W[1]-complete. For the other classes in the W-hierarchy, hardness and completeness are analogously defined.

Design Techniques for FPT Algorithms: The parameterized algorithms described in this work largely use the following two popular design paradigms.
1. Depth-bounded Search Trees: This technique [28] involves a systematic exploration of the solution space of a parameterized problem in a tree-like fashion. The depths of such search trees are usually upper-bounded by values depending on the parameter. The idea is to find a *small subset* of the input instance in polynomial time such that at least one element of this subset is part of an optimal solution to the problem. A search tree is now built with nodes corresponding to recursive calls on smaller problems accompanied by smaller parameters.
2. Iterative Compression: This technique, introduced in [32], typically works for minimization problems parameterized by the solution size. The idea is to begin with a solution of size $(k + 1)$ and attempt to compress it (in FPT time) to a solution of size k. The compression version of a parameterized problem is: Given a graph G, a non negative integer k, a $(k + 1)$ size solution S' of

G, does there exist a solution, S of G such that $|S| \leq k$? The compression algorithm takes as input a triple $(G[v_1, \cdots, v_{i=k+1}], \{v_1, \cdots, v_{i=k+1}\}, k)$ and either reports that $G[v_1, \cdots, v_i]$ has no solution of size k or returns a k-size solution S. In the former case, we declare that G has no k sized solution. In the latter case, we call the compression algorithm with input $(G[v_1, \cdots, v_i, v_{i+1}], S \cup \{v_{i+1}\}, k)$ and proceed iteratively. The iterative run of the compression routine terminates when either a solution of size k is obtained for the entire graph, or if an intermediate instance turns out to be incompressible. If a k-size solution exists for G, we are guaranteed to find it after $n - k$ compressions. The template for iterative compression is as described in Algorithm 1.

Algorithm 1 IC$(G(V, E), k)$

Consider an arbitrary order $v_1, v_2, \cdots, v_n$ of vertices of G
Define $V' = \{v_1, v_2, \cdots, v_{k+1}\}$
$S \leftarrow \emptyset$, $i \leftarrow 1$
$S' \leftarrow V'$ `/*Initialize current` $(k+1)$`-size solution` S' `of` $G[V']$`*/`
while $i \leq n - k$ **do**
  $S \leftarrow COMPRESS(G[V'], S', k)$ `/* k-size solution of` $G[V']$ `*/`
  $i \leftarrow i + 1$, $S' \leftarrow S \cup \{v_{k+i}\}$ and $V' \leftarrow V' \cup \{v_{k+i}\}$
end while
return S

For more details on parameterized complexity, we refer the reader to [9, 11, 28].

2 r-Partization in Special Graph Classes

As *r-Partization* in general graphs is trivially para-NP-hard and W-hard when paramaterized by k and r, we focus our study to special graph classes. In particular, we consider perfect graphs, split graphs and bounded tree-width graphs.

2.1 r-Partization in Perfect Graphs

Any r-colorable graph cannot have K_{r+1} as a subgraph and this condition is sufficient for perfect graphs. Also, for a perfect graph G, there is a polynomial-time algorithm to determine whether $\chi(G)$ is at most r [16]. The following characterization of r-colorable perfect graphs was provided in [6, 35] which we rephrase and present below for completeness.

Lemma 1 *A perfect graph G is r-colorable if and only if G forbids K_{r+1} as a subgraph. Further, an r-partization solution for G is to find a set $S \subset V(G)$ such that for each clique C in G, $|C \cap S| \geq |C| - r$.*

As an r-partization solution for G is to find a minimum set of vertices that intersects all cliques of size $(r + 1)$ in G, we obtain a parameterized algorithm using a depth-bounded search tree technique.

Theorem 1 *Given a perfect graph G and integers k, r, there is an algorithm to decide whether G has an r-partization solution of size at most k in time $O^*((r+1)^k)$.*

Proof: Choose a K_{r+1} in G. Since at least one of the $(r+1)$ vertices are in any solution, the size of the graph and the parameter drop by 1 in each of the $(r+1)$ branches. Recurse on the remaining graph till no K_{r+1} is found or the parameter budget is exhausted. Note that since G is perfect, $\omega(G)$ and a clique on $(r+1)$ vertices (if one exists) can be obtained in polynomial time using the Lovász theta function [16]. As the depth of the search tree is upper-bounded by k, the algorithm runs in $O^*((r+1)^k)$ time. □

Theorem 2 *For every fixed r, r-*Partization *in perfect graphs, parameterized by k admits an $O(k^r)$ size kernel.*

Proof: For each fixed r, in $O(n^{r+1})$ time, an *r-Partization* instance $\langle G, k\rangle$ can be transformed into an $(r+1)$-uniform *Hitting Set* instance $\langle U = V(G), \mathcal{C} = \mathcal{K}, k' = k\rangle$, where $\mathcal{K}$ denotes the set of cliques of size $(r+1)$ in G. Thus, we obtain an $O(k^r)$ size kernel using the kernelization techniques employed in [1]. □

Theorem 3 *The Maximum Induced r-Colorable Subgraph problem in perfect graphs admits a polynomial-time r-approximation algorithm.*

Proof: Observe that if $\langle G, k, r\rangle$ is an YES instance of *r-Partization*, we have $\alpha(G) \geq \frac{n-k}{r}$. Thus, a maximum independent set I of G, which can be obtained in polynomial time for perfect graphs [16], is a maximum induced r-colorable subgraph of G with at least $\frac{n-k}{r}$ vertices. Thus, for perfect graphs, this is an r-approximate solution for the dual of *r-Partization*, referred to as the *Maximum Induced r-Colorable Subgraph* problem [35]. □

2.2 r-Partization in Split Graphs

Recall that a graph G is a split graph if $V(G)$ can be partitioned into a clique and an independent set. A split graph $G'(K' \cup I', E)$ can be transformed into an equivalent split graph $G(K \cup I, E)$ such that $|K| = \omega(G') = \omega(G)$ in linear time. For rest of the discussion, we will assume that this property is satisfied by any split graph.

Lemma 2 *For a split graph $G(K \cup I, E)$, there exists an optimum r-*Partization *solution S such that $S \subseteq K$.*

Proof: Consider a minimal solution S and let $x \in S \cap I$. As $G \setminus S$ is r-colorable and S is minimal, x is adjacent to at least one vertex in every color class of $G \setminus S$. Consider a color class C of $G \setminus S$. Since $N(x) \subseteq K$, x is adjacent to exactly one vertex (say v) in C. Replace x by v in S. By iteratively processing such vertices in S, it follows that the resultant solution S obtained is a subset of K. □

Using Lemma 2, we describe Algorithms 1 and 2 that employ *structured iterative compression*. The idea is to build an iterative algorithm, and at each iteration, compress the current solution, which is a clique, into a smaller one unless it is optimal.

Algorithm 2 COMPRESS(G', S', k)

Transform $G'(K' \cup I', E')$ to $G(K \cup I, E)$ such that $|K| = \omega(G') = \omega(G)$
Preprocess S' such that $S' \subseteq K$ `/* Linear time pre-processing */`
`/*` $K' \cup I'$ `denotes` $V(G) \setminus S'$ `*/`
for all $Y \subseteq S'$ {Iterate over sets Y to be retained in solution S} **do**
  `/*` $G[S' \setminus Y]$ `is` r`-colorable */`
  If $\exists x \in I'$ that is in a K_{r+1} in $G[V \setminus Y]$ **then** skip to the next choice of Y
  for all $X \subseteq K'$ with $|X| \leq k - |Y|$ **do**
    If $G[V \setminus (X \cup Y)]$ is r-colorable **then** return $X \cup Y$
  end for
end for
Declare $\langle G, k, r \rangle$ as a NO instance

Lemma 3 *Given a split graph G and integers k, r, Algorithms 1 and 2 decide whether G has an r-partization solution of size at most k in time $O^*(2^{k+r})$.*

Proof: Delete isolated vertices from I as these vertices cannot be a part of any optimal solution. Consider a $(k+1)$-size solution S'. Denote $V(G) \setminus S'$ as $K' \cup I'$, where K' is a maximum clique and I' is an independent set. Let S be the required k-size solution such that $S \cap S' = Y$. Since $G[V \setminus S']$ is r-colorable, $|K'| \leq r$. Observe that S' is a clique. As $G[S' \setminus Y]$ is r-colorable, $G[S' \setminus Y]$ is a clique on at most r vertices. If $|Y| = i$ then $|S' \setminus Y| = k + 1 - i$. Since $G[S' \setminus Y]$ is an r-colorable clique, $k+1-i \leq r$. Therefore, the number of ways of choosing Y from S is $\binom{k+1}{k+1-r} + \cdots + \binom{k+1}{k} = \sum_{i=k+1-r}^{k} \binom{k+1}{i}$. After choosing Y, a subset $X \subseteq K'$ such that $|X| \leq k - |Y|$ and $G[V \setminus (X \cup Y)]$ is r-colorable can be found by an exhaustive search among the $\binom{r}{k-|Y|}$ choices. The run-time of the algorithm is bounded by $\sum_{i=0}^{k} \binom{k+1}{i}\binom{r}{k-i} \leq \binom{k+r+1}{k} \leq 2^{k+r+1}$. Thus, the required solution, if one exists, is obtained in $O^*(2^{k+r})$ time. □

Remark: *r-Partization* on chordal graphs (and hence for split graphs) is known to be polynomial-time solvable for each fixed r, using dynamic programming in $O(n^r)$ time [6, 35]. We observe that the run-time $\sum_{i=0}^{k} \binom{k+1}{i}\binom{r}{k-i}$ of our algorithm can also be bounded by $\binom{k+r+1}{r+1} \leq (k+r+1)^{r+1}$. Thus, we obtain a new polynomial-time algorithm for each fixed r on split graphs.

2.3 r-Partization in Treewidth-Bounded Graphs

It is known that any problem expressible in monadic second order logic is FPT when parameterized by the treewidth [7]. Atomic predicates $V(x)$ and $S(v)$ denote $x \in V(G)$ and $v \in S$, $S \subseteq V(G)$, respectively. $Inc(x, e)$ represents the incidence relation between $x \in V(G)$ and $e \in E(G)$.

$$\phi = \exists S(AtMost_k(S) \wedge Sets_r \wedge Coloring_r)$$

$AtMost_k(S) : \forall s_1, \cdots, \forall s_{k+1} \bigvee_{1 \leq i \neq j \leq k+1}(s_i = s_j)$ is true if and only if $|S| \leq k$.

$Sets_r : \exists X_1 \exists X_2 \cdots \exists X_r(\forall x(V(x) \wedge \neg S(x)) \rightarrow ((X_1(x) \vee \cdots \vee X_r(x)) \wedge \forall_{1 \leq i \neq j \leq r} \neg \exists x(X_i(x) \wedge X_j(x)))$ is true if and only if $V(G) \setminus S$ can be partitioned into r sets.

$Coloring_r : \forall x \forall y((V(x) \wedge V(y) \wedge \neg S(x) \wedge \neg S(y) \wedge (x \neq y)) \wedge \exists z(Inc(x, z) \wedge Inc(y, z)) \rightarrow \bigwedge_{1 \leq i \leq r} \neg(X_i(x) \wedge X_i(y)))$ is true if and only if $G[V \setminus S]$ can be properly colored using r colors.

By expressing r-*Partization* in monadic second order logic, we conclude that the problem parameterized by treewidth and r is FPT.

3 Bicochromatization in Perfect Graphs

Recall that a graph G is perfect if for every induced subgraph H of G, $\chi(H) = \omega(H)$. The *Bicochromatization* problem in a perfect graph G is to determine the existence of a set S of at most k vertices satisfying one of the following properties:

1. $G \setminus S$ is bipartite. This is to solve for OCT on G.
2. $V(G \setminus S)$ can be partitioned into 2 cliques. In other words, $\overline{G} \setminus S$ is bipartite. Thus, it suffices to solve OCT on $\overline{G}$.
3. $G \setminus S$ is a split graph. Here, $V(G \setminus S)$ can be partitioned into a clique and an independent set. This is to solve *Split Deletion* on G.

By the weak perfect graph theorem, G is perfect if and only if $\overline{G}$ is perfect [14]. Thus, on perfect graphs, algorithms for OCT and *Split Deletion* solve *Bicochromatization*. We describe the algorithms for OCT and *Split Deletion* in Sections 3.1 and 3.2. These algorithms employ the iterative compression technique with different COMPRESS routines.

3.1 OCT in Perfect Graphs

The problem of finding a maximum independent set in a perfect graph is solvable in polynomial time using polyhedral combinatorics [16]. However, similar complexity results do not hold for OCT which was recently shown to be NP-hard [2]. Observe that a minimum OCT in a perfect graph is a minimum cardinality set of vertices that intersects every triangle. Exploiting this fact and the structure of perfect graphs, we describe an $O^*(2^k)$ parameterized algorithm (Algorithms 1 and 3) using iterative compression technique. The key idea is that the COMPRESS subroutine first pre-processes the instance as long as possible: let S' denote the current $(k+1)$-size solution and Y be a subset of S'. If there exists

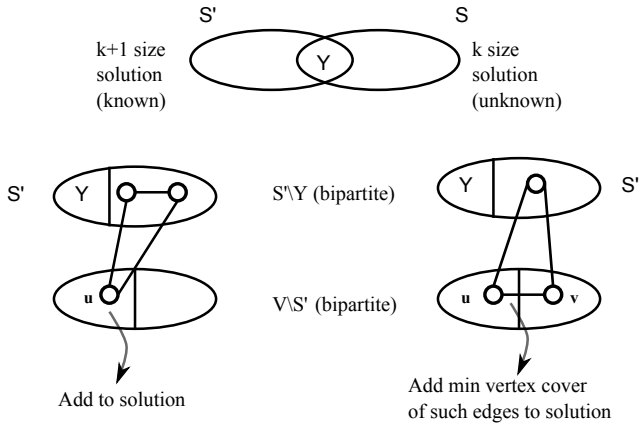
a K_3 in $G[V \setminus Y]$ with exactly one vertex v in $V(G) \setminus S'$, add v to the solution and decrease k by 1. After processing such triangles, we reduce the problem to *Vertex Cover* in a bipartite graph.

Algorithm 3 COMPRESS(G, S', k)

```
/* Iterate over each set Y to be retained in solution S */
for all Y ⊆ S' do
  If G[S' \ Y] is not bipartite then skip to next subset of S'
  X ← ∅
  for all K_3 = {u, v ∈ S' \ Y, x ∈ V \ S'} in G[V \ Y]
      /* Add x to solution and decrease k by 1 */
      X ← X ∪ {x}, k ← k − 1
  T = set of triangles in G[V \ (X ∪ Y)] and E' = E(G[V \ S']) ∩ E(T)
  X' ← min vertex cover of G(V \ S', E')
  If |X| + |X'| ≤ k − |Y| then return X ∪ X' ∪ Y
end for
Declare ⟨G, k⟩ as NO instance
```

Lemma 4 *Given a perfect graph G and an integer k, Algorithms 1 and 3 decide whether G has an OCT of size at most k in time $O^*(2^k)$. Further, these algorithms decide if G has an at most k-size set S such that $V(G \setminus S)$ can be partitioned into 2 cliques.*

Proof: Consider a $(k+1)$-size solution S'. If a smaller solution S exists, it has an intersection Y (possibly empty) with S'. The COMPRESS subroutine searches for such a solution S by iterating over all $Y \subseteq S'$ to be retained in S and choosing $X \subseteq V(G) \setminus S'$ to replace $S' \setminus Y$. We consider only subsets Y for which $G[S' \setminus Y]$ is bipartite. Since $G[V \setminus S']$ and $G[S' \setminus Y]$ are bipartite, $G[V \setminus Y]$ is 4-colorable. Thus, $\omega(G[V \setminus Y]) \leq 4$.



For any triangle in $G[V \setminus Y]$ that has exactly one vertex $v \in V(G) \setminus S'$, we add v to X. After processing such triangles as long as possible, any triangle in $G[V \setminus Y]$, if exists, has two vertices in $V(G) \setminus S'$ and one vertex in $S' \setminus Y$. Let

E' be the set of edges of such triangles having both endpoints in $V(G) \setminus S'$. A minimum vertex cover X' of $G(V \setminus S', E')$ essentially intersects every triangle in $G[V \setminus Y]$. Since $G[V \setminus S']$ is bipartite, a minimum vertex cover can be found in polynomial time [34]. If $|X \cup X' \cup Y| \leq k$ then $S = X \cup X' \cup Y$ is the required solution. Such a set S, if one exists, is found by considering all possible 2^{k+1} partitions of S' in $O^*(2^k)$ time. Further, as $\overline{G}$ is perfect (since G is perfect), it suffices to determine the existence of an at most k-size OCT for $\overline{G}$ to decide if G has an at most k-size set S such that $V(G) \setminus S$ can be partitioned into 2 cliques. □

3.2 Split Deletion

The *Split Deletion* problem of determining the existence of $S \subseteq V(G)$ such that $G[V \setminus S]$ is a split graph is known to admit an $O^*(2^k)$ algorithm in general by a recent result [13]. We describe the algorithm in this section. For a split graph $G(K \cup I, E)$, exactly one of the following conditions holds [14]:

1. $|I| = \alpha(G)$ and $|K| = \omega(G)$. Here, the partition $V(G) = K \cup I$ is unique.
2. $|I| = \alpha(G)$ and $|K| = \omega(G) - 1$. Here, there exists a vertex $v \in I$ such that $K \cup \{v\}$ is a clique.
3. $|I| = \alpha(G) - 1$ and $|K| = \omega(G)$. Here, there exists a vertex $v \in K$ such that $I \cup \{v\}$ is independent.

Therefore, for a split graph G on n vertices, there are at most $n + 1$ split partitions which can be enumerated in polynomial time. The algorithm in [13] uses the iterative compression technique. The working of the COMPRESS routine is described as Algorithm 4.

Algorithm 4 COMPRESS(G, S', k)

```
/* Iterate over sets Y to be retained in solution S */
for all Y ⊆ S' do
  If G[S' \ Y] is not a split graph then skip to next subset of S'
  X ← ∅
  for each split partition K_{S'} ∪ I_{S'} of S' \ Y
      for each candidate split partition K' ∪ I' of G[V \ S']
          for all u ∈ K' non-adjacent to a vertex v ∈ K_{S'}
              X ← X ∪ {u}, k ← k − 1
          for all u ∈ I' adjacent to a vertex v ∈ I_{S'}
              X ← X ∪ {u}, k ← k − 1
  If |X| ≤ k − |Y| then return X ∪ Y
end for
Declare ⟨G, k⟩ as NO instance
```

Lemma 5 *Given a graph G and an integer k, Algorithms 1 and 4 decide whether G has a set S of size at most k such that $G \setminus S$ is a split graph in time $O^*(2^k)$.*

Proof: Consider a $(k+1)$-size solution S' and let $K' \cup I'$ denote a split partition of $G \setminus S'$. The COMPRESS subroutine searches for a smaller solution S by iterating over all $Y \subseteq S'$ to be retained in S and choosing $X \subseteq V(G) \setminus S'$ to replace $S' \setminus Y$. We consider only subsets Y for which $G[S' \setminus Y]$ is a split graph. For a split partition $K_{S'} \cup I_{S'}$ of $S' \setminus Y$, we search for a $X \subseteq V(G) \setminus S'$ such that in a split partition $K \cup I$ of $G \setminus S$, $K_{S'} \subseteq K$ and $I_{S'} \subseteq I$. For each split partition $K_{S'} \cup I_{S'}$ of $S' \setminus Y$, we consider all possible split partitions of $G[V \setminus S']$. Let $K \cup I$ be one such partition. We now describe the structure of vertices in X. If a vertex $u \in K'$ is non-adjacent to a vertex $v \in K_{S'}$, then u must be in X. Similarly, if a vertex $u \in I'$ is adjacent to a vertex $v \in I_{S'}$, then u must be in X. Hence, X can be obtained by identifying such vertices in $V(G) \setminus S'$. Thus, we can determine if the set S of at most k vertices exists in $O^*(2^k)$ time. □

From Lemmas 4 and 5, we have the following result.

Theorem 4 *Given a perfect graph G and an integer k, there is an algorithm running in time $O^*(2^k)$ that decides if G has a bicochromatization solution of size at most k.*

Bicochromatization in Co-chordal Graphs

To solve *Bicochromatization* in a co-chordal graph G, we have to determine the existence of a set S of at most k vertices satisfying one of the following properties: (1) $G \setminus S$ is bipartite. (2) $\overline{G} \setminus S$ is bipartite. (3) $\overline{G} \setminus S$ is a split graph. These can be achieved by solving for *OCT* on G (co-chordal), *OCT* on $\overline{G}$ (chordal) and *Split Deletion* on $\overline{G}$ (chordal), respectively. We show that *Bicochromatization* restricted to co-chordal graphs is polynomial-time solvable. *OCT* and *Split Deletion* on chordal graphs are known to be polynomial-time solvable [6, 10, 35]. We now describe a polynomial-time algorithm for *OCT* in co-chordal graphs. In [30], Raman et al. showed that if S is a minimum OCT of G, then $V(G) \setminus S$ can be partitioned into a maximal independent set V_1 of G and a maximum independent set V_2 of $G \setminus V_1$. Thus, it follows that *OCT* is polynomial time solvable in any graph class in which the maximal independent sets can be enumerated in polynomial time. Co-chordal graphs are one such graph class. The number of maximal independent sets in a co-chordal graph can be enumerated in linear time as the number of maximal cliques in a chordal graph is at most n [14]. Thus, we obtain an $O(n^2)$ time algorithm for *OCT* in co-chordal graphs by a simple enumeration technique: Iterate over all K, a maximal independent set of G. If $|K| + |K'| \geq n - k$, where K' is a maximum independent set in $G \setminus K$ then $V(G) \setminus (K \cup K')$ is the required OCT. If no maximal independent set K yields such an OCT, we declare $\langle G, k \rangle$ as a NO instance.

4 Generalized Above Guarantee Vertex Cover

In Sections 2 and 3, for perfect graphs, we have described parameterized algorithms for $(r, 0)$-*Partization* and (r, l)-*Partization* with $r + l = 2$. We know that

(r,l)-*Partization* is W-hard in general. We use this hardness to understand the parameterized complexity of a generalization of an above guarantee parameterization of *Vertex Cover*. We define *Generalized Above Guarantee Vertex Cover* as follows.

Generalized Above Guarantee Vertex Cover
Input: $\langle G, \mathcal{K}, k\rangle$, where $\mathcal{K}$ is a clique cover of G and $k \in \mathbb{N}$
Parameter: k
Question: Does G have a vertex cover of size at most $k + \sum_{C\in\mathcal{K}}(|C| - 1)$?

As the size of any vertex cover of G is at least the cardinality $\mu(G)$ of a maximum matching in G, a parameterization of *Vertex Cover* where the parameter is the difference between the optimum solution size and the guaranteed lower bound is appropriate to study. Such above guarantee parameterized problems were first considered by Mahajan and Raman in [25]. *Vertex Cover Above Maximum Matching* (also known as *Above Guarantee Vertex Cover*) is one of the most popular such parameterizations of *Vertex Cover* that have been studied in the literature [8, 19, 24, 26, 27, 29, 31]. The problem is to decide if G has a vertex cover of size at most $\mu(G) + k$, where k is the above guarantee parameter and was shown to be FPT by a parameter preserving reduction to *Almost 2-SAT* and an $O^*(15^k)$ algorithm for the same [27, 31]. The $O^*(2.3146^k)$ algorithm in [24] is the current fastest algorithm for *Vertex Cover Above Maximum Matching*. *Generalized Above Guarantee Vertex Cover* is a generalization of *Vertex Cover Above Maximum Matching* in which the lower bound on the size of a minimum vertex cover is obtained from the contribution of a clique cover. Note that this bound is a stronger bound as compared to the size of a maximum matching used in *Vertex Cover Above Maximum Matching*. While a maximum matching of a graph can be obtained in polynomial time, this is quite unlikely for a minimum clique cover [12]. However, the cliques in an arbitrary clique cover $\mathcal{K}$, not necessarily minimum, can be obtained in polynomial time by finding a maximal clique C in G and recursing on $G\backslash V(C)$ until every vertex of G is in some clique $C \in \mathcal{K}$.

Generalized Above Guarantee Vertex Cover is W-hard: We describe a parameterized reduction from (r,l)-Partization to Generalized Above Guarantee Vertex Cover by defining the following transformation: given a graph $G(V,E)$ with $V(G) = \{v_1, v_2, \cdots, v_n\}$ and integers r and l, the graph $G^{r,l}$ is constructed by taking r copies $G_1(\{v_{11}, \cdots, v_{1n}\}, E_1), \cdots G_r(\{v_{r1}, \cdots, v_{rn}\}, E_r)$ of G and l copies $\overline{G}_1(\{u_{11}, \cdots, u_{1n}\}, E'_1), \cdots \overline{G}_l(\{u_{l1}, \cdots, u_{ln}\}, E'_l)$ of $\overline{G}$. Also, for each $v_i \in V(G)$, the vertices $\{v_{1i}, \cdots, v_{ri}, u_{1i}, \cdots, u_{li}\}$ form a clique. A clique cover $\mathcal{K}$ of $G^{r,l}$ is $\{\{v_{ij}, u_{hj} \mid 1 \leq i \leq r, 1 \leq h \leq l\} \mid v_j \in V(G)\}$.

Lemma 6 *For any $k' \geq 0$, G has an (r,l)-partization solution of size k' if and only if $G^{r,l}$ has a vertex cover of size $(r + l - 1)n + k'$.*

Proof: Consider a subgraph H of G with $V(H) = V(G) \setminus S$, where $S \subseteq V(G)$ is an (r,l)-partization solution of size k' in G. Consider a partition of $V(H)$ into r independent sets $V_1, \cdots, V_r$ and l cliques $U_1, \cdots, U_l$. Define a subset

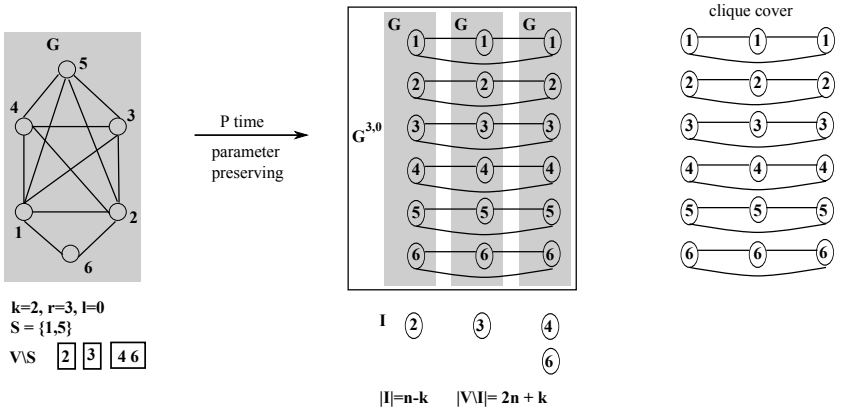


Figure 1: (r,l)-Partization $\leq_{fpt}$ Generalized Above Guarantee Vertex Cover

$I \subseteq V(G^{r,l})$ as follows: for each vertex $v_i \in V_j$, $v_{ji} \in I$ for $1 \leq j \leq r$ and for each vertex $v_i \in U_j$, $u_{ji} \in I$ for $1 \leq j \leq l$. By the construction of $G^{r,l}$, for each $\{v_i, v_j\} \in E(G)$, $\{v_{hi}, v_{pj}\}, \{u_{h'i}, u_{p'j}\} \notin E(G^{r,l})$ for all $1 \leq h \neq p \leq r$, $1 \leq h' \neq p' \leq l$. Also, for every $v_i \in V(H)$, exactly one of $v_{1i}, \cdots, v_{ri}, u_{1i}, \cdots, u_{li}$ is in I. Thus, we have $|I| = n - k'$ and I is an independent set in $G^{r,l}$. Hence, $V(G^{r,l}) \setminus I$ is a vertex cover of size $(r+l-1)n + k'$ in $G^{r,l}$. Conversely, consider a vertex cover X of size $(r+l-1)n+k'$ in $G^{r,l}$. Let $I = V(G^{r,l}) \setminus X$ be the corresponding independent set of size $n - k'$ in $G^{r,l}$. Clearly, for any $v_i \in G$, $1 \leq i \leq n$, at most one of $v_{j'i}$ and $u_{j'i}$, $1 \leq j \leq r$, $1 \leq j' \leq l$ can be in I. Partition I into $V'_1 \cup \cdots \cup V'_r \cup U'_1 \cup \cdots \cup U'_l$ such that each $v_{ij} \in I \cap V'_i$ and each $u_{ij} \in I \cap U'_i$. Define $V_j = \{v_i \mid v_{ji} \in V'_j\}$, $1 \leq j \leq r$ and $U_j = \{u_i \mid u_{ji} \in U'_j\}$, $1 \leq j \leq l$. We claim that $G[V_1 \cup \cdots \cup V_r]$ is r-colorable and the complement of $G[U_1 \cup \cdots \cup U_l]$ is l-colorable. On the contrary, if $G[V_1 \cup \cdots \cup V_r]$ is not r-colorable, without loss of generality, let $v_j, v_h \in V_i$ be adjacent. By the definition of V_i, it follows that v_{ij} and v_{ih} are adjacent in V'_i contradicting that I is an independent set in G. Similarly, if the complement of $G[U_1 \cup \cdots \cup U_l]$ is not l-colorable, then there exists vertices $u_j, u_h \in U_i$ which are non-adjacent in G. Thus, there exist an i' with $u_{i'j}, u_{i'h} \in U'_i \cap I$ such that $u_{i'j}$ and $u_{i'h}$ are adjacent contradicting the fact that I is an independent set in $G^{r,l}$. Hence, $V(G) \setminus (V_1 \cup \cdots \cup V_r \cup U_1 \cup \cdots \cup U_l)$ is an (r,l)-partization solution of k' vertices in G. □

Theorem 5 *For input instances of Generalized Above Guarantee Vertex Cover with cliques sizes at least 3 in the clique cover, the problem is not FPT unless* $\mathsf{P} = \mathsf{NP}$.

Proof: Consider an (r,l)-*Partization* instance $\langle G(\{v_1, v_2, \cdots, v_n\}, E), k, r, l\rangle$. The corresponding *Generalized Above Guarantee Vertex Cover* instance $\langle G^{r,l}, \mathcal{K}, k\rangle$ where $\mathcal{K}$ is a clique cover of $G^{r,l}$ defined as $\{\{v_{ij}, u_{i'j} \mid 1 \leq i \leq r, 1 \leq i' \leq l\} \mid v_j \in V(G)\}$. By Lemma 6, G has an (r,l)-partization solution of size at most k if and only if $G^{r,l}$ has a vertex cover of size at most $(r+l-1)n + k$.

For $k = l = 0$ and $r = 3$, it follows that G is 3-colorable if and only if $G^{r,l}$ has a vertex cover of size $2n$. An $O^*(f(k))$ algorithm for *Generalized Vertex Cover Above Guarantee* results in a polynomial-time algorithm for 3-coloring which is unlikely unless $\mathsf{P} = \mathsf{NP}$. Thus, the problem of determining if G has a vertex cover whose size is at most k more than the contribution from the clique cover is unlikely to be FPT unless $\mathsf{P} = \mathsf{NP}$. □

As (r,l)-*Partization* is trivially para-NP-hard for $r \geq 3$ or $l \geq 3$, this polynomial-time parameter-preserving reduction shows that *Generalized Above Guarantee Vertex Cover* is also para-NP-hard. It is known that the standard parameterized *Vertex Cover* cannot have subexponential parameterized algorithms under *Exponential-Time Hypothesis* [4]. Therefore, it follows that algorithms subexponential in the parameter are unlikely for *Vertex Cover Above Maximum Matching*. This result can also be obtained from Theorem 5.

Theorem 6 *The problem of deciding if G has a vertex cover of size at most $\mu(G) + k$, where k is the above guarantee parameter, does not have $2^{o(k)}$ time parameterized algorithm under Exponential-Time Hypothesis.*

Proof: By the construction of $G^{3,0}$, we have $\mu(G^{3,0}) \geq n + \mu(G)$. Let $\mu(G^{3,0}) = n + \mu(G) + c$ where $c \geq 0$. Note that as $\mu(G^{3,0})$ can be obtained in polynomial time, c can be determined in polynomial time. If there is an $2^{o(k)}$ algorithm for *Vertex Cover Above Maximum Matching*, then a vertex cover of size $2n = \mu(G^{3,0}) + (n - \mu(G) - c)$ can be obtained in $2^{o(n-\mu(G)-c)}$ time for $G^{3,0}$. Thus, in time subexponential in n, we can determine if G is 3-colorable or not. This is unlikely under *Exponential-Time Hypothesis*. □

As a consequence of this study, we observe the inter-relationship among the problems addressed in this paper as presented in Figure 2.

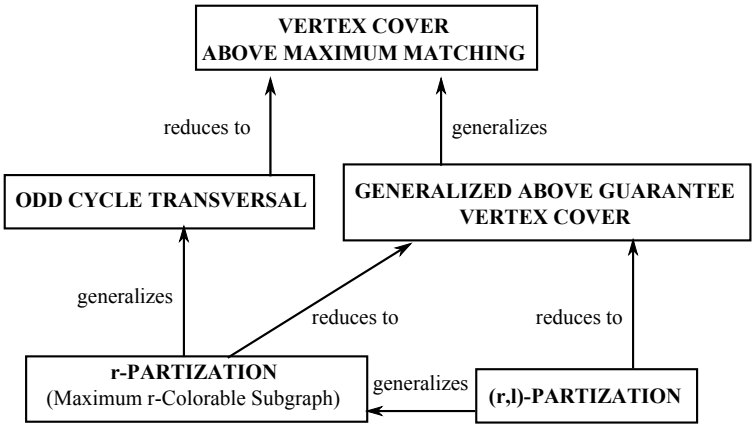


Figure 2: Zoo of Problems Addressed

5 Concluding Remarks

Though a randomized polynomial kernelization algorithm is known for OCT by a recent result [23], the existence of a deterministic kernelization algorithm is still open. It is interesting to note that OCT on perfect graphs, being a restricted *3-Hitting Set*, has a quadratic vertex kernel [1]. Determining if linear kernels exists is an interesting direction of research. Also, our $O^*(2^k)$ algorithm for OCT on perfect graphs is a faster algorithm for restricted *3-Hitting Set* instances than the best known $O^*(2.0755^k)$ algorithm for *3-Hitting Set* in general [33]. Though OCT on perfect graphs is NP-complete, from Lemma 6, it follows that OCT is polynomial solvable on perfect graphs $\mathcal{G}$ if the transformed graphs $\mathcal{G}^{2,0}$ are perfect since *Vertex Cover* is in polynomial time for perfect graphs using semidefinite programming [16]. A structural characterization of such graphs is an exciting study.

Acknowledgements

We thank Venkatesh Raman for pointing out that our OCT algorithm for perfect graphs can be extended to *Bicochromatization* and bringing the *Split Deletion* algorithm in [13] to our knowledge. We also acknowledge the comments of the referees which have increased the readability of this paper.

References

[1] F. N. Abu-Khzama. A kernelization algorithm for d-hitting set. *Journal of Computer and System Sciences*, 76(7):524–531, 2010. doi:10.1016/j.jcss.2009.09.002.

[2] L. A. Berry, W. S. Kennedy, A. D. King, Z. Li, and B. A. Reed. Finding a maximum-weight induced k-partite subgraph of an i-triangulated graph. *Discrete Applied Mathematics*, 158(7):765–770, 2010. doi:10.1016/j.dam.2008.08.020.

[3] A. Brandstädt. Partitions of graphs into one or two independent sets and cliques. *Discrete Mathematics*, 152(1-3):47–54, 1996. doi:10.1016/0012-365X(94)00296-U.

[4] L. Cai and D. Juedes. On the existence of subexponential parameterized algorithms. *Journal of Computer and System Sciences*, 67(4):789–807, 2003. doi:10.1016/S0022-0000(03)00074-6.

[5] M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, and K. Vusković. Recognizing berge graphs. *Combinatorica*, 25(2):143–186, 2005.

[6] D. G. Corneil and J. Fonlupt. The complexity of generalized clique covering. *Discrete Applied Mathematics*, 22(2):109–118, 1988-89. doi:10.1016/0166-218X(88)90086-8.

[7] B. Courcelle. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

[8] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. On multiway cut parameterized above lower bounds. *Parameterized and Exact Computation, Lecture Notes in Computer Science*, 7112:1–12, 2011. doi:10.1007/978-3-642-28050-4_1.

[9] R. G. Downey and M. R. Fellows. Parameterized complexity. *Springer-Verlag*, 1999.

[10] T. Ekim and D. de Werra. On split-coloring problems. *Journal of Combinatorial Optimization*, 10(3):211–225, 2005. doi:10.1007/s10878-005-4103-7.

[11] J. Flum and M. Grohe. Parameterized complexity theory. *Springer-Verlag*, 2006.

[12] M. R. Garey and D. S. Johnson. Computers and intractability: A guide to the theory of NP-completeness. *W.H.Freeman and Company*, 1979.

[13] E. Ghosh, S. Kolay, M. Kumar, P. Misra, F. Panolan, A. Rai, and M. S. Ramanujan. Faster parameterized algorithms for deletion to split graphs. *Algorithm Theory - SWAT, Lecture Notes in Computer Science*, 7357:107–118, 2012. doi:10.1007/978-3-642-31155-0_10.

[14] M. C. Golumbic. Algorithmic graph theory and perfect graphs. *Academic Press*, 1980.

[15] M. Grötschel, L. Lovász, and A. Schrijver. Polynomial algorithms for perfect graphs. *Annals of Discrete Mathematics*, 21:325–356, 1984.

[16] M. Grötschel, L. Lovász, and A. Schrijver. Geometric algorithms and combinatorial optimization. *Springer-Verlag*, 1988.

[17] J. Guo, H. Moser, and R. Niedermeier. Iterative compression for exactly solving NP-hard minimization problems. *Algorithmics of Large and Complex Networks, Lecture Notes in Computer Science*, 5515:65–80, 2009. doi:10.1007/978-3-642-02094-0_4.

[18] J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of generalized vertex cover problems. *Algorithms and Data Structures, Lecture Notes in Computer Science*, 3608:36–48, 2005. doi:10.1007/11534273_5.

[19] G. Gutin, E. J. Kim, M. Lampis, and V. Mitsou. Vertex cover problem parameterized above and below tight bounds. *Theory of Computing Systems*.

[20] P. Heggernes, D. Kratsch, D. Lokshtanov, V. Raman, and S. Saurabh. Fixed-parameter algorithms for cochromatic number and disjoint rectangle stabbing. *Algorithm Theory - SWAT, Lecture Notes in Computer Science*, 6139:334–345, 2010. doi:10.1007/978-3-642-13731-0_32.

[21] R. Impagliazzo and R. Paturi. Complexity of k-sat. *Proceedings of IEEE Conference on Computational Complexity*, pages 237–240, 1999. doi:10.1109/CCC.1999.766282.

[22] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity. *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

[23] S. Kratsch and M. Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *Proceedings of SODA*, pages 94–103, 2012.

[24] D. Lokshtanov, N. S. Narayanaswamy, V. Raman, M. S. Ramanujan, and S. Saurabh. Faster parameterized algorithms using linear programming. *CoRR abs/1203.0833*, 2012.

[25] M. Mahajan and V. Raman. Parameterizing above guaranteed values: MaxSat and MaxCut. *Journal of Algorithms*, 31(2):335–354, 1999. doi:10.1006/jagm.1998.0996.

[26] M. Mahajan, V. Raman, and S. Sikdar. Parameterizing above or below guaranteed values. *Journal of Computer and System Sciences*, 75(2):137–153, 2009. doi:10.1016/j.jcss.2008.08.004.

[27] S. Mishra, V. Raman, S. Saurabh, S. Sikdar, and C. R. Subramanian. The complexity of könig subgraph problems and above-guarantee vertex cover. *Algorithmica*, 61(4):857–881, 2011. doi:10.1007/s00453-010-9412-2.

[28] R. Niedermeier. Invitation to fixed-parameter algorithms. *Oxford Lecture Series in Mathematics and Its Applications, Oxford University Press*, 2006.

[29] V. Raman, M. S. Ramanujan, and S. Saurabh. Paths, flowers and vertex cover. *Algorithms - ESA, Lecture Notes in Computer Science*, 6942:382–393, 2011. doi:10.1007/978-3-642-23719-5_33.

[30] V. Raman, S. Saurabh, and S. Sikdar. Efficient exact algorithms through enumerating maximal independent sets and other techniques. *Theory of Computing Systems*, 41(3):563–587, 2007. doi:10.1007/s00224-007-1334-2.

[31] I. Razgon and B. O'Sullivan. Almost 2-sat is fixed-parameter tractable. *Journal of Computer and System Sciences*, 75(8):435–450, 2009. doi:10.1016/j.jcss.2009.04.002.

[32] B. A. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004. doi:10.1016/j.orl.2003.10.009.

[33] M. Wahlström. Algorithms, measures and upper bounds for satisfiability and related problems. *PhD thesis, Department of Computer and Information Science, Linköpings universitet, Sweden*, 2007.

[34] D. B. West. Introduction to graph theory. *Prentice Hall of India*, 2003.

[35] M. Yannakakis and F. Gavril. The maximum k-colorable subgraph problem for chordal graphs. *Information Processing Letters*, 24(2):133–137, 1987. doi:10.1016/0020-0190(87)90107-4.