TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 04/08 pp346-357
Volume 19, Number 4, August 2014

An Overview of Kernelization Algorithms for Graph Modification Problems

Yunlong Liu, Jianxin Wang*, and Jiong Guo

Abstract: Kernelization algorithms for graph modification problems are important ingredients in parameterized computation theory. In this paper, we survey the kernelization algorithms for four types of graph modification problems, which include vertex deletion problems, edge editing problems, edge deletion problems, and edge completion problems. For each type of problem, we outline typical examples together with recent results, analyze the main techniques, and provide some suggestions for future research in this field.

Key words: graph modification problem; fixed-parameter tractable; kernelization algorithm

1 Introduction

Graph modification problems call for making a minimum number of modifications to the vertex/edge set of a given graph such that the resulting graph has some desired property. These problems can model a large number of practical applications including image processing, numerical algebra, relational databases, and computational biology[1-3].

Graph modification problems also constitute a broad range of NP-complete problems in computer science[1]. From a practical perspective, the number of optimal modifications is rather small in most instances of many engineering applications. This fact makes parameterized computation an effective approach to deal with these NP-hard problems. In recent years, a series of graph modification problems have been shown to be Fixed-Parameter Tractable (FPT) with respect to a parameter k. That is, each of these problems can be solved by an algorithm A with running time $O(f(k)n^{O(1)})$, where f is a recursive function, k is the number of modifications, and n is the size of the input graph. Correspondingly, the algorithm A is called an FPT algorithm for the problem considered[4,5].

Kernelization is an effective preprocessing procedure in many efficient FPT algorithms for graph modification problems. Let Q be a parameterized graph modification problem and (G,k) be an instance of Q. An algorithm K is called a kernelization algorithm for Q if K satisfies the following conditions: (1) K transforms (G,k) into the reduced instance (G',k') in polynomial time; (2) (G,k) is a yes-instance of Q if and only if (G',k') is a yes-instance of Q; and (3) $|G'| \leqslant g(k)$ and $k' \leqslant k$, where $g(k)$ is a computable function. Correspondingly, the problem Q is called *kernelizable* and the reduced instance (G',k') is called a *kernel*. In particular, Q is said to admit a *polynomial kernel* if $g(k)$ is a polynomial function on k. A parameterized problem is FPT if and only if it is kernelizable[4,5].

A graph modification problem with a polynomial kernel often admits an efficient FPT algorithm. More precisely, the time complexity of this kind of algorithm can be expressed by $O(f(k')|G'|^{O(1)} + \text{poly}(|G|))$, where $\text{poly}(|G|)$ denotes a polynomial

• Yunlong Liu and Jianxin Wang are with School of Information Science and Engineering, Central South University, Changsha 410083, China. E-mail: jxwang@mail.csu.edu.cn.
• Yunlong Liu is also with College of Mathematics and Computer Science, Key Laboratory of High Performance Computing and Stochastic Information Processing (Ministry of Education of China), Hunan Normal University, Changsha 410081, China. E-mail: hnsdlyl@163.com.
• Jiong Guo is with Universität des Saarlandes, Campus E 1.7, D-66123 Saarbrücken, Germany. E-mail: jguo@mmci.uni-saarland.de.
∗ To whom correspondence should be addressed.
Manuscript received: 2014-06-09; accepted: 2014-07-17

function on $|G|$. Thus, kernelization algorithms for graph modification problems have been the subject of increasing attention in recent years. In particular, some well-known examples including vertex cover, feedback vertex set, and cluster editing were successively shown to admit polynomial kernels[6-8].

In this paper, we provide an overall survey of kernelization algorithms for various graph modification problems admitting polynomial kernels. This survey has some novel aspects. First, we mainly focus on the regularities of kernelization algorithms for the same type of problems, and attempt to generalize some techniques that were used in several concrete problems. We believe that such general techniques could not only be applied to other unexplored problems, but also could help in developing new techniques. Second, in analyzing algorithms, we highlight the strategy of bounding the size of the reduced instance by a polynomial function $g(k)$. In other words, we do not describe the reduction rules step by step, but attempt to analyze the method for the purpose of deriving reduction rules.

With respect to the content arrangement, this survey is described along the lines of four types of graph modification problems, including vertex deletion problems, edge editing problems, edge deletion problems, and edge completion problems. For each type of problem, we first list some examples together with recent results, analyze the typical techniques used in some representative algorithms, and then suggest potential research directions and open avenues for further study.

Before going on to the next section, we briefly mention another line of inquiry in this field. Recent results have shown that not every FPT graph modification problem admits a polynomial kernel under certain assumptions. For example, Kratsch and Wahlström[9] presented a special graph class H on seven vertices for which H-free edge deletion and H-free edge editing do not admit polynomial kernels unless NP $\subseteq$ coNP/poly. Let H be a path or cycle and assume coNP $\nsubseteq$ NP/poly. Cai and Cai[10] proved that H-free edge deletion, completion, and editing have no polynomial kernel if H has at least 4 edges. A consideration of this line of inquiry is not contained in the present survey, and concerning the details of this direction, we refer to Refs. [11, 12] for excellent overviews.

2 Terminology and Notations

All graphs considered in this paper are simple, loopless, and undirected. For a given graph $G = (V, E)$, $V(G)$ represents its vertex set and $E(G)$ its edge set. Moreover, let $n = |V|$ and $m = |E|$. The set of *neighbors* of a vertex $v \in V$ is denoted by $N_G(v) = \{u|uv \in E\}$. For a subset K of vertices, $N_G(K)$ denotes the set of vertices that are not in K but adjacent to some vertices in K, i.e., $N_G(K) = \bigcup_{v \in K}(N_G(v)) \setminus K$. The *degree* of a vertex v is denoted by $d_G(v) = |N_G(v)|$. The *closed neighborhood* of v is denoted by $N_G[v] = N_G(v) \cup \{u\}$. Two vertices $u, v \in V$ are *true twins* if $N_G[u] = N_G[v]$. Note that the subscript will be omitted when there is no risk of ambiguity.

Given a subset $S \subseteq V$, $G[S]$ denotes the subgraph induced by S, i.e., the graph $G[S] = (S, E'), E' = \{uv|uv \in E$ and $u, v \in S\}$. In particular, we denote $G[V']$ by $G \setminus \{x\}$ if $V'=V \setminus \{x\}$. An induced cycle of length at least 4 is called a *hole*. A hole with l vertices is denoted by H_l.

A graph G is F-free for a given subgraph F if G does not contain any induced subgraph isomorphic to F. For a family $\mathcal{F}$ of subgraphs, G is called $\mathcal{F}$-free if G is F-free for every $F \in \mathcal{F}$. If a graph class $\mathcal{G}$ is $\mathcal{F}$-free, then the graphs in $\mathcal{F}$ are called the *forbidden induced subgraphs* of $\mathcal{G}$. Note that in some following tables, the term *forbidden induced subgraphs* is represented by FISGs.

A graph admitting a property Π means that it belongs to a graph class $\mathcal{G}$, namely the set of graphs that satisfy the property Π. Furthermore, a graph property Π being characterized by the forbidden induced subgraphs in $\mathcal{F}$ indicates that the corresponding graph class $\mathcal{G}$ is $\mathcal{F}$-free.

Let G be a resulting graph modified by some edges. A vertex v in G is *affected* if v is touched by at least one modified edge, otherwise, it is *unaffected*.

3 Vertex Deletion Problems

Vertex deletion problems constitute a fundamental class of graph modification problems.

For a graph property Π, the parameterized Π-vertex deletion problem is defined as follows.

Input: an undirected graph $G = (V, E)$ and an integer k.

Parameter: k.

Task: find a set $S \subseteq V$ of size at most k such that the graph $G[V \setminus S]$ satisfies the property Π or answer "NO".

3.1 Known results for typical problems

In recent years, the studies of kernelization for vertex deletion problems have made considerable progress. One remarkable conclusion is that, whenever a graph property Π can be characterized by a family of finite forbidden induced subgraphs, the Π-vertex deletion problem admits a polynomial kernel[9, 13]. Some problems in which the graph property is characterized by a family of infinite forbidden induced subgraphs have also been shown to admit polynomial kernels[7, 14]. In addition, some problems involving the non-hereditary graph property have also been vigorously pursued[15, 16]. Typical problems admitting non-trivial kernels are listed in Table 1.

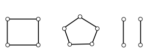
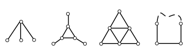
3.2 The main technical points

We mainly discuss the problems for which the graph property Π is hereditary.

For problems with finite forbidden induced subgraphs, there exists a general method of kernelization. Given an instance, we first remove the vertices that do not occur in any forbidden subgraph since they can never be included in any optimal solution. Since each remaining vertex must be in at least one forbidden induced subgraph, we can further reduce the instance by the technique used in kernelization for d-hitting set[13, 20].

More precisely, we can directly translate the considered problem to d-hitting set, where the constant d is the number of vertices in the largest forbidden induced subgraph. For d-hitting set, Flum and Grohe[20] proved a kernel with $O(k^d)$ elements by employing the sunflower Lemma due to Erdös and Rado[21]. Thus, each problem in this class must admit a polynomial kernel. Additionally, Kratsch[22] presented a claim that the standard parameterization p-$\mathcal{Q}$ of $\mathcal{Q}$ admits a polynomial kernelization in which $\mathcal{Q} \in \text{MIN F}^+\Pi_1$. This also indicates that every graph modification problem in MIN $\text{F}^+\Pi_1$ admits a polynomial kernel.

Table 1 Typical vertex deletion problems admitting non-trivial polynomial kernels.

Property/class	FISGs	Result	Ref.
Independent set		$2k$	[6]
Split		$O(k^3)$	[17]
s-plex cluster	$\mathcal{F}(s, n-s-1)$*	$O(k^2s^3)$	[19]
Proper interval		$O(k^{53})$	[14]
Cycle-free	No definition	$O(k^2)$	[7]
(p,q)-grid	Non-hereditary	$O(k^4)$**	[15]
r-regular	Non-hereditary	$O(kr(r+k)^2)$	[16]

* We refer to Ref. [18] for its formal definition.

** The edit operations include vertex/edge removals and additions.

To obtain smaller kernels for these problems, extensive research has been conducted in recent years. In particular, Abu-Khzam[13] presented an improved kernel for d-hitting set. Several methods for vertex deletion problems have been proposed. A typical method that has been employed[13, 23, 24] can be described as follows[25], where Triangle Vertex Deletion (TVD) is taken as an example.

Construct a witness. The first step is to greedily construct a maximal witness W for the considered problem. Note that the maximality of the witness is crucial since it plays a key role in subsequent steps. The size of W also provides a lower bound on the number of deletions required to obtain an instance satisfying the property Π. In many cases, a witness is a family of forbidden induced subgraphs restricted by certain conditions. In particular, the witness is a maximal set of edge-disjoint triangles in TVD.

Bound the size of witness. Let W be a constructed witness of a yes-instance. In the next step, we mainly show that the number of vertices in W can be bounded by a polynomial function $p(k)$. To obtain such a result, one often needs to use the sunflower reduction rule. For the TVD problem, there are at most $2k^2 + k$ vertices in W.

Reduction. The remaining step is to reduce the size of $V \setminus V(W)$ by a polynomial function $h(k)$. Its procedure depends on the specific property of a given problem. Let (G, k) be a yes-instance of TVD and assume that the vertices that do not occur in any triangle have been removed. It can be proven that $I = V(G) \setminus V(W)$ is an independent set and every triangle containing one vertex in I shares one edge with at least one triangle in W. Based on these facts, we can construct an auxiliary bipartite graph and derive a reduction rule. By this rule, we reduce the size of I to $3k^2$ and obtain a problem kernel with $5k^2 + k$ vertices[13].

This method has also been denoted as "conflict packing"[24, 25] since the witness was specially referred to as conflict packing in the pertinent literature. In particular, conflict packing provides a kernelization tool for problems that are not known to admit constant-factor approximation algorithms[25].

A similar method is the "kernelization through tidying" technique proposed by Bevern et al.[19] Let (G, k) be an instance of a given problem. The first step is to greedily compute an approximation set S for G. The next step is to compute a tidying set $T \subseteq (V(G) \backslash S)$ by employing some reduction rules. Finally, the third step is to shrink the graph $G[V \setminus (S \cup T)]$ using some specific reduction rules. Applying this method, Bevern et al.[19] developed an $O(k^2 s^3)$-vertex problem kernel for s-plex cluster vertex deletion that can be computed in $O(ksn^2)$ time, which improved the $O(k^{s+1+T_s})$-vertex problem kernel obtained from Kratsch's general technique, where T_s is the maximum integer satisfying $T_s \cdot (T_s + 1) \leqslant s$[19].

For the problems with infinite forbidden induced subgraphs, the kernelization appears to be more difficult. The main difficulty lies in finding a bounded witness since the size of some forbidden subgraphs may be large or infinite. In studying proper interval vertex deletion, Fomin et al.[14] proposed a novel strategy in which the small constant size forbidden subgraphs were "insulated" from the large or infinite forbidden subgraphs. Toward this goal, Fomin et al.[14] obtained a polynomial kernel for a variant of d-hitting set that preserves all minimal solutions of size at most k along with a witness for the minimality. Using this strategy, one can first reduce the family of all small constant size forbidden induced subgraphs (i.e., *claws, nets, tents*, and *holes* $H_l (4 \leqslant l \leqslant 8)$) in the input graph. The next step is to shrink "clique and clique paths" in proper interval graphs by some other reduction rules. Hence a problem kernel with $O(k^{53})$ vertices was obtained[14]. This strategy seems promising for obtaining polynomial kernels for other vertex deletion problems.

3.3 Challenges and further research

Although the study of vertex deletion problems has greatly progressed, substantial potential for further development remains as follows.

(1) For the problems involving finite forbidden induced subgraphs, improving the kernel obtained from the general methods is an important research direction. Along this line, vertex cover admits an improved kernel with $2k$ vertices[6] and split vertex deletion admits an improved kernel with $O(k^3)$ vertices[17]. The question to be considered is whether we can obtain improved kernels for other problems such as cluster vertex deletion, chain vertex deletion, threshold vertex deletion, and co-trivially perfect vertex deletion.

(2) For the problems with infinite forbidden induced subgraphs, some well-known problems remain to be explored.

- Proper interval vertex deletion admits an $O(k^{53})$-vertex kernel[14]. The question to be considered is whether we can obtain an improved kernel with at most $O(k^{10})$ vertices[14,26].
- Solving the open problem whether directed feedback vertex set admits a polynomial kernel. This has been open since the FPT algorithm was proposed by Chen et al.[27] in 2008.
- Recently, interval vertex deletion has been shown to be FPT[28,29]. The question to be considered is whether the problem admits a polynomial kernel[14,26].

(3) An interesting direction is to provide randomized polynomial kernels for some problems. Recently, Kratsch and Wahlström[30] have presented a randomized polynomial kernel for odd cycle transversal. The existence of randomized polynomial kernels is an exciting question for several related problems such as directed feedback vertex set, multiway cut, and vertex multicut[30].

(4) Another active research area involves development of problem kernels for special graphs such as planar graphs and graphs of bounded genus[31-37]. In this area, further research can be accomplished in many ways. For example, it can be investigated whether some of the open problems mentioned above admit polynomial kernels on planar graphs.

4 Edge Editing Problems

Edge editing problems constitute the most common type of edge modification problem.

For a graph property Π, the parameterized Π-edge editing problem is defined as follows.

Input: an undirected graph $G = (V, E)$ and an integer k.

Parameter: k.

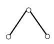
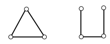
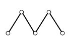
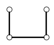
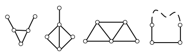
Task: find a set $F \subseteq V \times V$ of size at most k such that $G' = (V, (E \setminus F) \cup (F \setminus E))$ satisfies the property Π or answer "NO".

4.1 Known results for typical problems

At present, a series of edge editing problems have been shown to admit polynomial kernels.

Typical problems with their recent results are listed in Table 2.

Table 2 Typical edge editing problems admitting polynomial kernels.

Property/class	FISGs	Result	Ref.
Cluster		$2k$	[38]
Bicluster		$6k$	[39]
s-plex cluster	$\mathcal{F}(s, n-s-1)$	$(4s^2-2)k+$ $(4s^2-2)$	[18]
Flip consensus tree		$O(k^3)$*	[40]
Cograph		$O(k^3)$	[41]
3-leaf power		$O(k^3)$	[42]

* The input graph is a bipartite graph.

4.2 The main technical points

To obtain a kernel for a given edge editing problem, an adverse deduce method is usually employed to derive reduction rules.

Let (G, k) be an instance of a given problem and G_{opt} be the graph resulting from applying a solution S with $|S| \leqslant k$ to G. The set $V(G_{\text{opt}})$ can be partitioned into a set of affected vertices and a set of unaffected vertices. In this case, we also say that $V(G)$ can be partitioned into two subsets, and use $X(G)$ to denote the set of affected vertices and use $Y(G)$ to denote the set of those unaffected in G.

In searching for a reduced graph G', one can begin with the graph G_{opt}, analyze which vertices in G_{opt} should be removed for a problem kernel, and then derive the corresponding reduction rules. In practice, there are three typical strategies to bound the size of G'.

(1) Apply a function linear in $|X(G')|$ to bound $|Y(G')|$.

The central task of this strategy is to discover the structural relationship between two kinds of vertices. If the vertices in $Y(G)$ are "surrounded" by the vertices in $X(G)$ and $|Y(G)| \leqslant c|X(G)|$, then we can obtain a problem kernel with $2(c+1)k$ vertices. This strategy has been successfully used in cluster editing, bicluster editing, and s-plex editing[8, 18, 39]. Generally speaking, the structural relationship is based on some deeper insights into the structural property of unaffected vertices in G_{opt}. For cluster editing, the unaffected vertices contained in each cluster in G_{opt} must form a critical clique in the original graph G[8]. Similarly, for bicluster editing, the unaffected vertices contained in each biclique in G_{opt} must form at most two critical independent sets in the original graph G[39].

Once the structural relationship is discovered, some reduction rules can be consequently derived from it. We continue using cluster editing in Guo[8] as an example. According to the previous observation, it becomes more effective to reduce the critical clique graph than to directly reduce the input graph. Along this line, Guo[8] derived a reduction rule to reduce the critical cliques having a large number of unaffected vertices. Hence, in the reduced graph G', $|Y(G')| \leqslant 2|X(G')|$ results in a linear kernel for this problem. For bicluster editing and s-plex editing, some crucial reduction rules were derived in a similar way[18, 39].

(2) Apply the sum of the editing degree of affected vertices to bound $|V(G')|$.

In recent research on cluster editing, Chen and Meng[38] observed that, in many cases, an optimal solution to a graph G will make $K \cup N(K)$ a disjoint clique for a critical clique K in G. Based on this observation, they introduced a novel notion, i.e, *the editing degree* of affected vertices, to study the kernel of this problem.

As described by Fig. 1, let K be a critical clique in a graph G and let $v \in N(K)$. The *editing degree* $p_K(v)$ of v with respect to K is defined to be the number of vertex pairs $\{v, w_1\}$, where $w_1 \in N(K) \setminus \{v\}$ and $w_1 \notin N(v)$, plus the number of edges (v, w_2), where $w_2 \notin K \cup N(K)$[38].

The editing degree of the affected vertices has been used as a key tool to obtain an improved kernel for cluster editing[38]. If all vertices in the reduced graph can be bounded by the sum of the editing degree on the affected vertices, then a kernel with $2k$ vertices can be obtained.

For an instance of cluster editing (G, k), and K representing a critical clique in G, Chen and Meng[38] obtained an important observation: if K satisfies one of the following conditions (1)–(3), then $K \cup N(K)$ can be made a disjoint clique and be safely removed from G since the corresponding edge operations are contained in an optimal solution of G. (1) $|K| > k$; (2) $|K| \geqslant |N(K)|$ and $|K| + |N(K)| > \sum_{v \in N(K)} p_K(v)$; and (3)

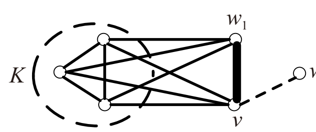


Fig. 1 An illustration of the definition of editing degree. The dashed line denotes a deleted edge and the thick line denotes an inserted edge.

$|K| < |N(K)|$ and $|K| + |N(K)| > \sum_{v \in N(K)} p_K(v)$, and there is no vertex $u \in N_2(K)$ with $|N(u) \cap N(K)| > (|K| + |N(K)|)/2$, where $N_2(S)$ denotes the neighbors of $N(S)$ that are not in $S \cup N(S)$.

With two additional reduction rules, we can obtain a sufficiently reduced instance $(G' = (V', E'), k')$.

Let (G', k') be a yes-instance and S be a solution. For all vertices v in G', denote by $p_0(v)$ the number of edges incident to v that are inserted/deleted by S. The clusters in $G'' = (V', (E' \backslash S) \cup (S \backslash E'))$ can be divided into two sub-collections: $\mathcal{P}_1$ consisting of the clusters in which all vertices are touched, and $\mathcal{P}_2$ consisting of the clusters that are not in $\mathcal{P}_1$. The number of vertices in $\mathcal{P}_1$ can be bounded by the formula: $\sum_{C \in \mathcal{P}_1} |C| \leqslant \sum_{C \in \mathcal{P}_1} \sum_{v \in C} p_0(v)$. For each cluster C in $\mathcal{P}_2$, $|C| = |K| + |N(K)| \leqslant \sum_{v \in N(K)} p_K(v) = \sum_{v \in C} p_0(v)$. The number of vertices in $\mathcal{P}_2$ can be bounded by the formula: $\sum_{C \in \mathcal{P}_2} |C| \leqslant \sum_{C \in \mathcal{P}_2} \sum_{v \in C} p_0(v)$. Hence, $|V'| \leqslant \sum_{C \in \mathcal{P}_1} \sum_{v \in C} p_0(v) + \sum_{C \in \mathcal{P}_2} \sum_{v \in C} p_0(v) = \sum_{v \in V'} p_0(v) \leqslant 2k$[38].

As can be shown, the editing degree of affected vertices plays an important role in deriving reduction rules and bounding the size of a kernel. The notion of the editing degree originates from the special structure of a clique. The presented example demonstrates that the discovery of some novel structural properties of the problem considered is a decisive step in studying kernels for graph modification problems.

(3) Apply a special tree to bound the structure of G'.

For some problems, one G_{opt} corresponds to a special tree $\mathcal{T}$. For example, in flip consensus tree editing, each M-free bipartite graph corresponds to one critical independent set tree[40]. In cograph editing, each cograph corresponds to a totally decomposable modular decomposition tree[41]. Further, in 3-leaf power editing, each critical clique graph of a 3-leaf power graph is exactly a forest[42]. Applying the special properties of a tree, we can derive some reduction rules working on $\mathcal{T}$ and obtain a reduced graph G'.

Note that every *node* in $\mathcal{T}$ has a one-to-one correspondence to a subset of vertices in G_{opt}. For a considered tree $\mathcal{T}$, if the number of vertices in each node is at most $k + 1$ and the number of nodes in $\mathcal{T}$ is at most $h(k)$, then $|V(G_{\text{opt}})| = |\mathcal{T}| \leqslant (k + 1) \cdot h(k)$.

Generally speaking, reducing each node in $\mathcal{T}$ is based on discovering its special properties associated with the optimal solutions. Typical nodes include the modular node, critical clique node, and critical independent set node. For reducing the critical clique node, Bessy et al.[42] drew a general conclusion. Let (G, k) be an instance of a given Π-edge editing problem. If the graph property Π is hereditary and closed under true twin addition, then there always exists an optimal edition F of G that preserves the critical cliques of G[42]. Hence, the reduction rule that removes $|C| - (k + 1)$ arbitrary vertices from each critical clique C with $|C| > (k+1)$ is safe[42]. For the modular node and critical independent set node, similar reduction rules are safe[40,41].

In addition, we have to reduce the considered paths in $\mathcal{T}$. Applying the properties of a tree, we can bound the number of considered paths in a yes-instance. The remaining task is to bound the number of non-affected nodes in each considered path. Along this line, some reduction rules have been derived. For example, in cograph editing, a sunflower reduction rule was derived to bound the number of non-affected nodes in each considered path by $2k + 3$[41]. In 3-leaf power editing, a reduction rule was derived to bound the number of critical cliques in each 2-branch-path by 8[42]. Further, in flip consensus tree editing, a reduction rule was derived to bound the length of each degree-two-path by $2k + 1$[40].

4.3 Challenges and future research

For edge editing problems, the study of kernelization algorithms is a new subject with substantial scope for development, listed as follows.

(1) In the study of cluster editing, editing degree has proven to be a novel notion. Recently, Cao and Chen[43] proposed another new technique based on edge cuts, and obtained a kernel with $2k$ vertices for weighted cluster editing. These techniques seem promising for obtaining efficient kernels for other problems. This provokes a question of whether, for example, the kernel for bicluster editing can be improved to $2k$ in the manner used in cluster editing.

(2) Several of the problems listed in Table 2 admit kernels with $O(k^3)$ vertices. Improving these to quadratic or linear kernels (if possible) is a challenging task.

(3) Some problems open to further study are listed as follows.

- Trivially perfect editing has been recently shown to be NP-hard[44]. Since trivially perfect graphs are (P_4, C_4)-free graphs, this problem is obviously fixed-parameter tractable. It remains open whether it admits a polynomial kernel.

- Chordal editing has been recently shown to be FPT, where the parameter is the number of three types of operations, i.e., vertex deletions, edge deletions, and edge completions[45]. Again, the question remains whether this problem admits a polynomial kernel.

5 Edge Deletion Problems

In some edge modification problems, the insertion operation makes no sense. For some edge editing problems, their computational complexity is still open. Thus, edge deletion problems are often studied as a separate branch.

For a graph property Π, the parameterized Π-edge deletion problem is defined as follows.

Input: an undirected graph $G = (V, E)$ and an integer k.

Parameter: k.

Task: find a set $F \subseteq E$ of at most k such that $G' = (V, E \setminus F)$ satisfies the property Π or answer "NO".

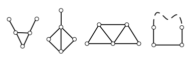
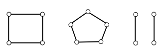
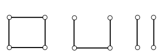
5.1 Known results for typical problems

Edge deletion problems with known kernelization results are listed in Table 3. These problems can be classified into three types. The first type includes some problems in which the insertion operation makes no sense. The second type includes some variants of edge editing problems. The third type is the $2K_2$-free family, including chain edge deletion and related problems.

5.2 The main technical points

The kernelization algorithms of edge deletion problems can be classified into different groups according to

Table 3 Typical edge deletion problems admitting polynomial kernels.

Property/class	FISGs	Result	Ref.
Triangle-free		$6k$	[46]
s-cycle transversal	No definition	$O(k^{s-1})$*	[47]
Cograph		$O(k^3)$	[41]
3-leaf power		$O(k^3)$	[42]
Chain		$O(k^2)$**	[48]
Split		$O(k^2)$	[17]
Co-trivially perfect		$O(k^3)$	[48]
Threshold		$O(k^3)$	[48]

* This result is for $s > 4$, and it is $6k^2$ when $s = 4$.

** The input graph is a bipartite graph.

the three types of problems defined above. For the first type of problem, such as triangle-free and 4-cycle-transversal, the basic strategy for kernelization is similar to the conflict packing method used in vertex deletion problems[46,47]. For s-cycle-transversal ($s > 4$), Xia and Zhang[47] presented a nontrivial generalization of the conflict packing method by exploring an extended reduction rule. After computing the witness W for G, the algorithm checks repeatedly whether there are edges in $E(W)$ that can be safely deleted. Once such edges are found, they will be deleted and this procedure begins again. For the variants of edge editing problems, such as cograph deletion and 3-leaf power deletion, the techniques are similar to those used in corresponding edge editing problems[41,42].

In the following, we mainly discuss the technique used in some problems with $2K_2$ as one of the forbidden induced subgraphs[48].

For the edge deletion problems with finite forbidden induced subgraphs, a possible strategy for kernelization is that used in vertex deletion problems with finite forbidden induced subgraphs. More precisely, we delete the vertices that do not occur in any forbidden induced subgraph, and then bound the number of each kind of forbidden subgraph by some sunflower reduction rules. Let l be the number of vertices in the largest forbidden subgraph and $h(k)$ be the upper bound on the number of the largest forbidden subgraph in the reduced graph. A problem kernel with $O(l \cdot h(k))$ vertices can consequently be obtained. However, for an edge deletion problem, it is possible that all optimal solutions include an edge not involved in any forbidden subgraph in the original graph. Thus, how to deal with the vertices that do not occur in any of the forbidden subgraphs is a difficult point in reducing a given instance of an edge deletion problem.

For the $2K_2$-free graph classes, such as chain, split, threshold, and co-trivially perfect, the vertices that do not occur in any forbidden subgraphs can be safely removed. Hence, the strategy mentioned above can be applied in the kernelization algorithms for these problems. We take split edge deletion as an example to analyze this technique.

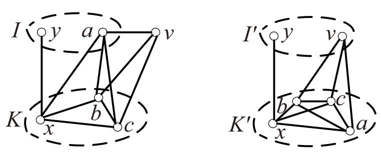
A graph $G = (V, E)$ is called a *split graph* if there exists a partition (K, I) of V such that K is a clique and I is an independent set. Equivalently, a graph G is a split graph if and only if G contains no induced C_4, C_5, and $2K_2$ (see Table 3). Let (G, k) be an instance of split edge deletion, and v a vertex in G that does not occur in any induced C_4, C_5, and $2K_2$, as described in Fig. 2a. It can be proven that $(G = (V, E), k)$ is a yes-instance if and only if $(G \setminus \{v\}, k)$ is a yes-instance[48]. By Fig. 2, we illustrate the main idea in its sufficiency proof.

Assume that $(G \setminus \{v\}, k)$ is a yes-instance. Let S be a solution for it and let $V(G \setminus \{v\})$ be partitioned into (K, I) (see Fig. 2a). It can be proven that the set $N_G(v)$ is a clique. On this basis, we can obtain a solution S' from S for the graph G by constructing another split graph. Let K' denote the vertex set $N_G(v) \cup X$ in which $X = \{v \in K | v \in (\bigcap_{u \in N_G(v)} N_G(u))\}$ and I' denote the vertex set $\{v\} \cup (I \setminus N_G(v)) \cup (K \setminus X)$, and let E' denote the edge set containing all possible edges between the vertices in K' and all edges in E between K' and I'. Then, $H = ((K', I'), E')$ is exactly a split graph. For example, the split graph illustrated in Fig. 2b is constructed from the graph in Fig. 2a in this manner. It can be further proven that $|S'| \leqslant |S|$. Thus, S' is a solution for (G, k).

Note that the forbidden subgraph $2K_2$ plays a crucial role in the reduction rule above. First, the construction of the new split graph is heavily based on the fact that the set $N_G(v)$ is a clique, which relies on the condition that v does not occur in any induced $2K_2$, C_4, and C_5. Next the proof of the relation formula $|S'| \leqslant |S|$ also depends on the condition that v does not occur in any induced $2K_2$.

Applying this strategy, Guo[48] also obtained polynomial kernels for several other problems, such as chain edge deletion, co-trivially perfect edge deletion, and threshold edge deletion.

The method used in this example is a typical method to prove the equivalence between the original instance (G, k) and the reduced instance (G', k'). In many cases, some solutions for (G', k') may not be solutions for (G, k). To prove that (G, k) is a yes-instance, one general approach is presented in this example. Let S be a solution for (G', k'). If S is also a solution for (G, k), then we are done. Otherwise, the main task is to seek another special solution S' from S for (G', k') such that S' is also a solution for (G, k).



(a) (b)

Fig. 2 An illustration of the transformation between two distinct solutions for split edge deletion.

5.3 Challenges and future research

Kernelization of edge deletion problems has been the subject of much attention in recent years. However, there are still many problems to be studied, such as the following.

(1) For some problems with special forbidden induced subgraphs, the vertices that do not occur in any forbidden subgraphs can be safely deleted. It is an interesting open question which graph classes, other than the $2K_2$-free graph class, also admit this property.

(2) The known results in Table 3 may be further improved. For example, the kernel for split edge deletion was improved from $O(k^4)$ to $O(k^2)$ vertices[17]. It is of interest to consider if the kernel of threshold (/co-trivially perfect) edge deletion can be improved in a similar way.

(3) Some well-known problems, as follows, remain open to study.

- There is a question of whether edge multiway cut and edge multicut, which have been the subject of considerable attention in recent years, admit polynomial kernels[26].
- Whether pseudosplit edge deletion admits a quadratic or linear kernel is also of interest. This problem has recently been proven to admit a subexponential parameterized algorithm[49].
- Line graph edge deletion seeks to delete at most k edges from the input graph to obtain a line graph. While this problem admits an $O^*(11^k)$ FPT algorithm, it is of interest to determine if it admits a polynomial kernel[26].
- Claw-free edge deletion seeks to delete at most k edges from the input graph to obtain a $K_{1,3}$-free graph. Obviously, this problem admits an $O^*(3^k)$ FPT algorithm, but it remains open whether it admits a polynomial kernel[26].

6 Edge Completion Problems

Edge completion problems are also an important type of edge modification problem.

For a graph property Π, the parameterized Π-edge completion problem is defined as follows.

Input: an undirected graph $G = (V, E)$ and an integer k.

Parameter: k.

Task: find a set F of at most k edges such that $G + F = (V, E \cup F)$ satisfies the property Π or answer "NO".

6.1 Known results for typical problems

The study of kernelization complexity for edge completion problems originated from that for chordal edge completion (also denoted as the minimum fill in problem)[50]. Recently, some related problems have been shown to admit polynomial kernels[42,51].

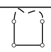
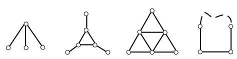
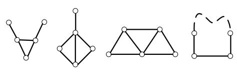
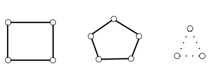
Typical problems and their recent results are listed in Table 4. We also mention a family of problems, including cograph edge completion, chain edge completion, split edge completion, and threshold edge completion, each of which is exactly the complement problem of the corresponding edge deletion problem. Since the related edge deletion problems were described in the previous section, we do not list them in Table 4.

6.2 The main technical points

As shown in Table 4, problems involving the *hole*-free property constitute the main body of edge completion problems admitting polynomial kernels. Hole admits an important property that any chordless hole of length l needs at least $l-3$ inserted edges to be triangulated. Thus, for a yes-instance, the number of vertices in any induced hole will not exceed $k+3$.

Chordal edge completion is one of the most notable edge completion problems. We first introduce its kernelization algorithm presented by Kaplan et al.[50] Let (G,k) be a yes-instance. The scheme employed in that algorithm is to partition the vertex set $V(G)$ into two subsets A and B such that: The vertices of every chordless hole in G are contained in A; A set of edges F is a minimal triangulation of G if and only if F is a minimal triangulation of $G[A]$; and $|A| = O(k^3)$. The partition algorithm starts with $B = V(G)$, $A = \varnothing$, and applies sequentially the following three procedures, denoted by P_1, P_2, and P_3. (P_1) Search repeatedly for independent chordless holes in $G[B]$ and move their vertices from B to A. After performing P_1, $|A| = O(k)$ since $\sum_{C\in G[A]}(|C|-3) \leqslant k$. ($P_2$) Search repeatedly for chordless holes in G containing at least two consecutive vertices from B. Let C be such a hole and $\mathcal{Q}$ be the family of disjoint maximal sub-paths of C containing only vertices from B. For each $R \in \mathcal{Q}$ with $l(R) \geqslant 1$, where $l(R)$ denotes the length of R, move the vertices of R from B to A. After performing P_2, the size of A remains $O(k)$ since the number of vertices added to A by the procedure P_2 is at most $2k$ (we refer to Ref. [50] for this analysis). (P_3) For every nonadjacent pair of vertices $x, y \in A$, compute the set $A_{x,y}$ of all vertices $b \in B$ such that x, b, and y appear consecutively on some chordless hole in G. If $|A_{x,y}| > 2k$, then add the edge (x, y) to G. Otherwise, move all vertices in $A_{x,y}$ from B to A. Since there are $O(k^2)$ non-edges in $G[A]$ prior to operation by P_3 and P_3 may add at most $2k$ vertices to A for each non-edges in $G[A]$, the size of A is $O(k^3)$ after performing procedure P_3. Moreover, the overall complexity of the partitioning algorithm is $O(k^2nm)$. Thus, a kernel with $O(k^3)$ vertices is obtained[50].

Table 4 Typical edge completion problems admitting polynomial kernels.

Property/class	Forbidden induced subgraphs	Result	Ref.
Chordal		$2k^2+4k$	[52]
Proper interval		$O(k^3)$	[51]
3-leaf power		$O(k^3)$	[42]
Bi-clique chain		$O(k^2)$	[51]

Based on the algorithm described above, Natanzon et al.[52] conducted further study and obtained an $O(k^2)$ kernel for this problem. The researchers performed a refined analysis on the number of vertices that were moved in procedure P_3. By denoting by A^i and B^i the partition obtained after procedure P_i is completed, for $i = 1, 2$, and 3, and letting $x, y \in A^2$, $(x, y) \notin E$, an important observation was obtained. If $A_{x,y} \neq \varnothing$, then, for any triangulation F of G, either $(x, y) \in F$, or, for every $b \in A_{x,y}$, F contains an edge incident on b. Based on this observation, the number of vertices that were moved to A^2 in procedure P_3 can be obviously decreased. Assume that (G,k) is a yes-instance, and that, in procedure P_3, all sets $A_{x,y}$ moved into A are of size at most d. Then $|A^3 - A^2| \leqslant M\cdot k$, where $M = \max\{d, 2\}$. Since $|A_{x,y}| \leqslant 2k$ and $|A^2| \leqslant 4k$, the partition algorithm terminates with $|A| \leqslant 2k(k+2)$.

Next, we discuss the kernelization algorithm for proper interval edge completion presented by Bessy and Perez[51]. Except for some generic reduction rules, the branches reduction rule is the main reduction rule used in this algorithm.

Proper interval graphs admit an important property as follows. A graph is a proper interval graph if and only if its vertex set admits an ordering respecting the "umbrella" property[53]. Based on this property, a proper interval graph can be decomposed into a path of cliques in which every consecutive clique is connected by a join[25]. Thus, the notion of branches can be employed for its kernelization.

Let (G, k) be a yes-instance of proper interval edge completion reduced under some generic reduction rules including the connected component rule, the sunflower rule, and the critical clique rule. The following task is to reduce the size of the branches. Due to the presence of *claws* or 4-*cycles* intersecting the branches, the vertices in branches are partitioned into two subsets to be discussed respectively. Let A be the set of vertices belonging to a claw or a 4-cycle in G and B be the set of other vertices. It can be proven that there are at most $4k^3 + 15k^2 + 16k$ vertices in A. The remaining task is to reduce the size of B. According to the umbrella property, there are four types of branches between two consecutive affected vertices in G, i.e., K-join, 1-branch, two disjoint 1-branch, and 2-branch, some of them being described in Fig. 3. Hence, reducing the size of B can be accomplished by reducing the size of each type of branch.

Along this line, Bessy and Perez[51] presented the corresponding reduction rules including the K-join rule, 1-branch rule, and 2-branch rule. Let (G', k') be a yes-instance reduced under these rules, K be an arbitrary K-join, R be an arbitrary 1-branch, and P be an arbitrary 2-branch in G'. These reduction rules ensure that the following relation formulas are sound: $|B' \cap K| \leqslant 2k + 2$, $|B' \cap R| \leqslant 4k + 3$, and $|B' \cap P| \leqslant 12k + 10$, where B' denotes the set of unaffected vertices that are not contained in any claw or 4-cycle of G'. Moreover, since there are at most $2k$ affected vertices in G', the total number of branches is $2k + 1$. On this basis, Bessy and Perez[51] obtained a problem kernel with $O(k^3)$ vertices.

The branches reduction rule has also been applied to bi-clique chain edge completion and 3-leaf power edge completion[42,51]. Generally, this rule was considered to be well-suited for graph modification problems in which the target graph admits an adjacency decomposition[25].

6.3 Challenges and future research

In recent years, kernelization algorithms for edge completion problems have been the subject of increasing attention. We list some interesting problems as follows.

(1) The problems with positive results are mainly concentrated on some problems involving the *hole*-free property. Searching for other graph properties covering a series of problems admitting polynomial kernels is an interesting direction.

(2) It would be of interest to develop improved kernels for the problems listed in Table 4.

(3) Recently, considerable progress has been made on exploring subexponential parameterized complexity for edge completion problems. A series of problems, such as chordal edge completion, chain edge completion, split edge completion, trivially perfect edge completion, pseudosplit edge completion, and threshold edge completion, have been proven to admit subexponential parameterized algorithms[49,54]. An interesting problem is to study the existence of polynomial kernels for the edge completion problems admitting subexponential parameterized algorithms. In other words, we aim to determine whether every edge completion problem admitting a subexponential parameterized algorithm has a polynomial kernel.

(4) Some additional problems open to further study are listed as follows.

- It is of interest to know whether a problem in which the graph property is defined by *holes* and finite subgraphs admits a polynomial kernel. This problem is FPT, and several special cases of the problem have been shown to admit polynomial kernels[25].
- The question of whether Interval Edge Completion admits a polynomial kernel is also interesting[25]. Its FPT algorithm has been known since 2009[55].

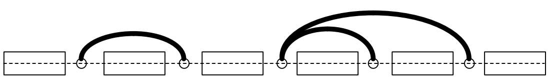


Fig. 3 An illustration of the kernel for proper interval edge completion. The bold edges denote the inserted edges.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 61070224, 61232001, and 61173051) and the China Postdoctoral Science Foundation (No. 2012M521551).

References

[1] R. Sharan, Graph modification problems and their applications to genomic research, PhD degree dissertation, Tel-Aviv University, 2002.

[2] S. L. Lauritzen and D. J. Spiegelhalter, Local computations with probabilities on graphical structures and their application to expert systems, *Readings in Uncertain Reasing*, pp. 415-448, 1990.

[3] F. Mancini, Graph modification problems related to graph classes, PhD degree dissertation, University of Bergen Norway, 2008.

[4] R. Downey and M. Fellows, *Parameterized Complexity*. New York: Springer, 1999.

[5] J. Chen, Parameterized computation and complexity: A new approach dealing with NP-Hardness, *Journal of Computer Science and Technology*, vol. 20, no. 1, pp. 18-37, 2005.

[6] J. Chen, I. A. Kanj, and W. Jia, Vertex cover: Further observations and further improvements, *Journal of Algorithms*, vol. 41, pp. 280-301, 2001.

[7] S. Thomassé, A $4k^2$ kernel for feedback vertex set, *ACM Transactions on Algorithms*, vol. 6, no. 2, pp. 1-6, 2010.

[8] J. Guo, A more effective linear kernelization for cluster editing, *Theoretical Computer Science*, vol. 410, pp. 718-726, 2009.

[9] S. Kratsch and M. Wahlström, Two edge modification problems without polynomial kernels, *Discrete Optimization*, vol. 10, no. 3, pp. 193-199, 2013.

[10] L. Cai and Y. Cai, Incompressibility of H-free edge modification problems, in *Proc. 8th International Symposium on Parameterized and Exact Computation (IPEC)*, Lecture Notes in Computer Science Volume 8246, Sophia Antipolis, France, 2013, pp. 84-96.

[11] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin, On problems without polynomial kernels, *Journal of Computer and System Sciences*, vol. 75, no. 8, pp. 423-434, 2009.

[12] A. Rai, Kernel lower bounds: A survey, *A thesis submitted to the board of studies in Theoretical Computer Science*, 2012.

[13] F. N. Abu-Khzam, Kernelization algorithms for d-hitting set problems, in *Proc. 10th International Workshop on Algorithms and Data Structures (WADS)*, Lecture Notes in Computer Science 4619, Halifax, Canada, 2007, pp. 434-445.

[14] F. V. Fomin, S. Saurabh, and Y. Villanger, A polynomial kernel for proper interval deletion, *SIAM J. Discrete Math.*, vol. 27, no. 4, pp. 1964-1976, 2013.

[15] J. Díaz and D. M. Thilikos, Fast FPT-Algorithms for cleaning grids, in *Proc. 23rd Annual Conference on Theoretical Aspects of Computer Science (STACS)*, Lecture Notes in Computer Science Volume 3884, Marseille, 2006, pp. 361-371.

[16] H. Moser, D. M. Thilikos, Parameterized complexity of finding regular induced subgraphs, *Journal of Discrete Algorithms*, vol. 7, pp. 181-190, 2009.

[17] E. Ghosh, S. Kolay, M. Kumar, P. Misra, F. Panolan, A. Rai, and M. S. Ramanujan, Faster parameterized algorithms for deletion to split graphs, in *Proc. 13th Scandinavian Symposium and Workshop on Algorithm Theory (SWAT)*, Lecture Notes in Computer Science Volume 7357, Helsinki, Finland, 2012, pp. 107-118.

[18] J. Guo, C. Komusiewicz, R. Niedermeier, and J. Uhlmann, A more relaxed model for graph-based data cluster: s-plex editing, *SIAM J. Discrete Math.*, vol. 24, no. 4, pp. 1662-1683, 2010.

[19] R. V. Bevern, H. Moser, and R. Niedermeier, Kernelization through tidying: A case study based on s-plex cluster vertex deletion, in *Proc. 9th Latin American Theoretical Infornatics Symposium (LATIN)*, Lecture Notes in Computer Science 6034, México, 2010, pp. 528-539.

[20] J. Flum and M. Grohe, Parameterized complexity theory, *Texts in Theoretical Computer Scicence, An EATCS Series*. Springer, 2006.

[21] P. Erdös and R. Rado, Intersection theorems for systems of sets, *Journal of London Mathematical Society*, vol. 35, pp. 85-90, 1960.

[22] S. Kratsch, Polynomial kernelization for MIN $\mathrm{F}^{+}\Pi_1$ and MAX NP, *Algorithmica*, vol. 63, nos. 1-2, pp. 532-550, 2012.

[23] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier, Fixed-parameter algorithms for cluster vertex deletion, *Theory of Computing Systems*, vol. 47, no. 1, pp. 196-217, 2010.

[24] C. Paul, A. Perez, and S. Thomassé, Conflict packing yields linear vertex-kernels for k-FAST, k-dense RTI and a related problem, in *Proc. 36th Conference on Mathematical Foundations of Computer Science (MFCS)*, Lecture Notes in Computer Science Volume 6907, Warsaw, Poland, 2011, pp. 497-507.

[25] A. Perez, Kernelization algorithms for graph and other structure modification problems, *Extented Abstract of PhD Thesis*, 2012.

[26] M. Cygan, L. Kowalik, and M. Pilipczuk, Open problems from workshop on kernels, http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf, Warsaw, 2013.

[27] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon, A fixed-parameter algorithm for the directed feedback vertex set problem, *Journal of The ACM*, vol. 55, no. 5, pp. 1-19, 2008.

[28] A. Rafiey, Single exponential FPT algorithm for interval vertex deletion and interval completion problem, *CoRR*, abs/1211.4629, 2012.

[29] Y. Cao and D. Marx, Interval deletion is fixed-parameter tractable, in *Proc. 25rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Portland, Oregon, 2014, pp.122-141.

[30] S. Kratsch and M. Wahlström, Compression via matroids: A randomized polynomial kernel for odd cycle transversal, in *Proc. 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Kyoto, Japan, 2012, pp.94-103.

[31] J. Guo and R. Niedermeier, Linear problem kernels for NP-hard problems on planar graphs, in *Proc. 34th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science Volume 4596, Wroclaw, Poland, 2007, pp. 375-386.

[32] J. Wang, Y. Yang, J. Guo, and J. Chen, Planar graph vertex partition for linear problem kernels, *Journal of Computer and System Sciences*, vol. 79, no. 5, pp. 609-621, 2013.

[33] J. Wang, W. Luo, Q. Feng, J. Guo, and J. Chen, Improved linear problem kernel for planar connected dominating set, *Theoretical Computer Science*, vol. 511, pp. 2-12, 2013.

[34] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos, (Meta) Kernelization, in *Proc. 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, Atlanta, Georgia, 2009, pp. 629-638.

[35] Q. Feng, J. Wang, and J. Chen, Matching and weighted P_2-packing: Algorithms and kernels, *Theoretical Computer Science*, vol. 522, pp. 85-94, 2014.

[36] J. Wang, P. Tan, J. Yao, Q. Feng, and J. Chen, On the minimum link-length rectilinear spanning path problem: Complexity and Algorithms, *IEEE Transactions on Computers*, doi.ieeecomputersociety.org/10.1109/TC.2013.163, 2014.

[37] Q. Feng, J. Wang, S. Li, and J. Chen, Randomized parameterized algorithms for P_2-packing and co-path packing problems, *Journal of Combinatorial Optimization*, 2014, DOI 10.1007/s10878-013-9691-z.

[38] J. Chen and J. Meng, A $2k$ kernel for the cluster editing problem, *Journal of Computer and System Sciences*, vol. 78, no.1, pp. 211-220, 2012.

[39] J. Guo, F. Hüffner, C. Komusiewicz, and Y. Zhang, Improved algorithms for bicluster editing, in *Proc. 5th Theory and Applications of Models of Computation (TAMC)*, Lecture Notes in Computer Science Volume 4978, Xi'an, China, 2008, pp. 445-456.

[40] C. Komusiewicz and J. Uhlmann, A cubic-vertex kernel for flip consensus tree, *Algorithmica*, vol. 68, no. 1, pp. 81-108, 2014.

[41] S. Guillemot, C. Paul, and A. Perez, On the (non-)existence of polynomial kernels for P_l-free edge modification problems, *Algorithmica*, vol. 65, no. 4, pp. 900-926, 2013.

[42] S. Bessy, C. Paul, and A. Perez, Polynomial kernels for 3-leaf power graph modification problems, *Discrete Applied Mathmatics*, vol. 158, pp. 1732-1744, 2010.

[43] Y. Cao and J. Chen, Cluster editing: Kernelization based on edge cuts, *Algorithmica*, vol. 64, pp. 152-169, 2012.

[44] J. Nastos and Y. Gao, Familial groups in social networks, *Social Networks*, vol. 35, no. 3, pp. 439-450, 2013.

[45] Y. Cao and D. Marx, Chordal editing is fixed-parameter tractable, in *Proc. 31st International Symposium on Theoretical Aspects of Computer Science (STACS)*, Lyon, France, 2014, pp. 214-225.

[46] D. Brügmann, C. Komusiewicz, and H. Moser, On generating triangle-free graphs, *Electronic Notes in Discrete Mathematics*, vol. 32, pp. 51-58, 2009.

[47] G. Xia and Y. Zhang, Kernelization for cycle transversal problems, *Discrete Applied Mathmatics*, vol. 160, nos. 7-8, pp. 1224-1231, 2012.

[48] J. Guo, Problem kernels for NP-Complete edge deletion problems: split and related graphs, in *Proc. 18th International Symposium on Algorithms and Computation, (ISAAC)*, Lecture Notes in Computer Science Volume 4835, Sendai, Japan, 2007, pp. 915-926.

[49] P. G. Drange, F. V. Fomin, M. Pilipczuk, and Y. Villanger, Exploring subexponential parameterized complexity of completion problems, in *Proc. 31st International Symposium on Theoretical Aspects of Computer Science (STACS)*, Lyon, France, 2014, pp. 288-299.

[50] H. Kaplan, R. Shamir, and R. E. Tarjan, Tractability of parameterized completion problems on chordal, strongly chordal, and proper intervalgraphs, *SIAM J. Comput.*, vol. 28, no. 5, pp. 1906-1922, 1999.

[51] S. Bessy and A. Perez, Polynomial kernels for proper interval completion and related problems, *Information and Computation*, vol. 231, pp. 89-108, 2013.

[52] A. Natanzon, R. Shamir, and R. Sharan, A polynomial approximation algorithm for the minimum fill-in problem, *SIAM J. Comput.*, vol. 30, no. 4, pp. 1067-1079, 2000.

[53] P. J. Looges and S. Olariu, Optimal greedy algorithms for indifference graphs, *Computer and Mathematics with Applications*, vol. 25, no. 7, pp. 15-25, 1993.

[54] F. V. Fomin and Y. Villanger, Subexponential parameterized algorithm for minimum fill-in, in *Proc. 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Kyoto, Japan, 2012, pp. 1737-1746.

[55] Y. Villanger, P. Heggernes, C. Paul, and J. A. Telle, Interval completion is fixed-parameter tractable, *SIAM Journal on Computing*, vol. 38, no. 5, pp. 2007-2020, 2009.

Yunlong Liu received the PhD degree in computer science from Central South University, China, in 2009. Currently, he is an associate professor at College of Mathematics and Computer Science, Hunan Normal University, China. His current research interests include computer algorithms and parameterized computation.

Jianxin Wang received the PhD degree in computer science from Central South University, China, in 2001. Currently, he is a professor at School of Information Science and Engineering, Central South University, China. His current research interests include algorithm analysis and optimization, computer network, and bioinformatics. He has published more than 100 papers in various international journals and refereed conferences. Dr. Wang is serving as the program committee chair or member of several international conferences. He is a senior member of Institute of Electrical and Electronics Engineers (IEEE).

Jiong Guo received the PhD degree in computer science from the University of Jena, Germany, in 2006. In August 2009, he joined the Cluster of Excellence "Multimodal Computing and Interaction" at the University of Saarbrücken and is currently the leader of a research group whose main goal is to design efficient algorithms for NP-hard problems. His current research interests include computational complexity theory, kernelization algorithms, parameterized computation, and bioinformatics.