Linear-Vertex Kernel for the Problem of Packing r-Stars into a Graph without Long Induced Paths

Florian Barbero[1], Gregory Gutin[2], Mark Jones[2], Bin Sheng[2], and Anders Yeo[3,4]

[1]Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier, 161 rue Ada, 34095 Montpellier cedex 5, France
[2]Department of Computer Science, Royal Holloway, University of London, TW20 0EX, Egham, Surrey, UK
[3]Engineering Systems and Design, Singapore University of Technology and Design, Singapore
[4]Department of Mathematics, University of Johannesburg, Auckland Park, 2006 South Africa

Abstract

Let integers $r \geq 2$ and $d \geq 3$ be fixed. Let $\mathcal{G}_d$ be the set of graphs with no induced path on d vertices. We study the problem of packing k vertex-disjoint copies of $K_{1,r}$ ($k \geq 2$) into a graph G from parameterized preprocessing, i.e., kernelization, point of view. We show that every graph $G \in \mathcal{G}_d$ can be reduced, in polynomial time, to a graph $G' \in \mathcal{G}_d$ with $O(k)$ vertices such that G has at least k vertex-disjoint copies of $K_{1,r}$ if and only if G' has. Such a result is known for arbitrary graphs G when $r = 2$ and we conjecture that it holds for every $r \geq 2$.

1 Introduction

For a fixed graph H, the problem of deciding whether a graph G has k vertex-disjoint copies of H is called H-PACKING. The problem has many applications (see, e.g., [2, 3, 11]), but unfortunately it is almost always intractable. Indeed, Kirkpatrick and Hell [11] proved that if H contains a component with at least three vertices then H-PACKING is NP-complete. Thus, approximation, parameterized, and exponential algorithms have been studied for H-PACKING when H is a fixed graph, see, e.g., [2, 7, 8, 14, 15].

In this note, we will consider H-PACKING when $H = K_{1,r}$ and study $K_{1,r}$-PACKING from a parameterized preprocessing, i.e., kernelization, point

of view.[1] Here k is the parameter. As a parameterized problem, $K_{1,r}$-PACKING was first considered by Prieto and Sloper [14] who obtained an $O(k^2)$-vertex kernel for each $r \geq 2$ and a kernel with at most $15k$ vertices for $r = 2$. (Since the case $r = 1$ is polynomial-time solvable, we may restrict ourselves to $r \geq 2$.) The same result for $r = 2$ was proved by Fellows *et al.* [7] and it was improved to $7k$ by Wang *et al.* [15].

Fellows *et al.* [7] note that, using their approach, the bound of [14] on the number of vertices in a kernel for any $r \geq 3$ can likely be improved to subquadratic. We believe that, in fact, there is a linear-vertex kernel for every $r \geq 3$ and we prove Theorem 1 to support our conjecture. A path P in a graph G, is called *induced* if it is an induced subgraph of G. For an integer $d \geq 3$, let $\mathcal{G}_d$ denote the set of all graphs with no induced path on d vertices.

Theorem 1. *Let integers $r \geq 2$ and $d \geq 3$ be fixed. Then $K_{1,r}$-PACKING restricted to graphs in $\mathcal{G}_d$, has a kernel with $O(k)$ vertices.*

Since d can be an arbitrary integer larger than two, Theorem 1 is on an ever increasing class of graphs which, in the "limit", coincides with all graphs. To show that Theorem 1 is an optimal[2] result, in a sense, we prove that $K_{1,r}$-PACKING restricted to graphs in $\mathcal{G}_d$ is $\mathcal{NP}$-hard already for $d = 5$ and every fixed $r \geq 3$:

Theorem 2. *Let $r \geq 3$. It is $\mathcal{NP}$-hard to decide if the vertex set of a graph in $\mathcal{G}_5$ can be partitioned into vertex-disjoint copies of $K_{1,r}$.*

We cannot replace $\mathcal{G}_5$ by $\mathcal{G}_4$ (unless $\mathcal{NP} = \mathcal{P}$) due to the following assertion, whose proof is provided in Appendix of [1].

Theorem 3. *Let $r \geq 3$ and $G \in \mathcal{G}_4$. We can find the maximal number of vertex-disjoint copies of $K_{1,r}$ in G in polynomial time.*

2   Terminology and Notation

For a graph G, $V(G)$ ($E(G)$, respectively) denotes the vertex set (edge set, respectively) of G, $\Delta(G)$ denotes the maximum degree of G and n its number of vertices. For a vertex u and a vertex set X in G, $N(u) = \{v : uv \in E(G)\}$, $N[u] = N(u) \cup \{u\}$, $d(u) = |N(u)|$, $N_X(u) = N(u) \cap X$, $d_X(u) = |N_X(u)|$ and $G[X]$ is the subgraph of G induced by X. We call $K_{1,r}$ an *r-star*. We say a star *intersects* a vertex set if the star uses a vertex in the set. We use

---

[1]We provide basic definitions on parameterized algorithms and kernelization in the next section, for recent monographs, see [5, 6]; [12, 13] are recent survey papers on kernelization.

[2]If $K_{1,r}$-PARKING was polynomial time solvable, then it would have a kernel with $O(1)$ vertices.

(G, k, r) to denote an instance of the r-star packing problem. If there are k vertex-disjoint r-stars in G, we say (G, k, r) is a YES-instance, and we write $G \in \star(k, r)$. Given disjoint vertex sets S, T and integers s, r, we say that S has s r-stars in T if there are s vertex-disjoint r-stars with centers in S and leaves in T.

A *parameterized problem* is a subset $L \subseteq \Sigma^* \times \mathbb{N}$ over a finite alphabet Σ. A parameterized problem L is *fixed-parameter tractable* if the membership of an instance (I, k) in $\Sigma^* \times \mathbb{N}$ can be decided in time $f(k)|I|^{O(1)}$ where f is a computable function of the *parameter k* only. Given a parameterized problem L, a *kernelization of L* is a polynomial-time algorithm that maps an instance (x, k) to an instance (x', k') (the *kernel*) such that $(x, k) \in L$ if and only if $(x', k') \in L$ and $k' + |x'| \leq g(k)$ for some function g. It is well-known that a decidable parameterized problem L is fixed-parameter tractable if and only if it has a kernel. Kernels of small size are of main interest, due to applications.

3 Proof of Theorem 1

Note that the 1-star packing problem is the classic maximum matching problem and if $k = 1$, the r-star packing problem is equivalent to deciding whether $\Delta(G) \geq r$. Both of these problems can be solved in polynomial time. Henceforth, we assume $r, k > 1$.

A vertex u is called a *small vertex* if $\max\{d(v) : v \in N[u]\} < r$. A graph without a small vertex is a *simplified graph*.

We now give two reduction rules for an instance (G, k, r) of $K_{1,r}$-PACKING.

Reduction Rule 1. *If graph G contains a small vertex v, then return the instance $(G - v, k, r)$.*

It is easy to observe that Reduction Rule 1 can be applied in polynomial time.

Reduction Rule 2. *Let $G = (V, E)$ be a graph and let C, L be two vertex-disjoint subsets of V. The pair (C, L) is called a constellation if $G[C \cup L] \in \star(|C|, r)$ and there is no star $K_{1,r}$ intersecting L in the graph $G[V \setminus C]$. If (C, L) is a constellation, return the instance $(G[V \setminus (C \cup L)], k - |C|)$.*

It is easy to observe that Reduction Rule 2 can be applied in polynomial time, provided we are given a suitable constellation.

Lemma 1. *Reduction Rules 1 and 2 are safe.*

Proof. Clearly, a small vertex v can not appear in any r-star. Therefore Reduction Rule 1 is safe as G and $G - v$ will contain the same number of r-stars.

To see that Reduction Rule 2 is safe, it is sufficient to show that $G \in \star(k, r)$ if and only if $G[V \setminus (C \cup L)] \in \star(k - |C|, r)$. On the one hand, if $G[V \setminus (C \cup L)] \in \star(k - |C|, r)$, the hypothesis $G[C \cup L] \in \star(|C|, r)$ implies $G \in \star(k, r)$. On the other hand, there are at most $|C|$ vertex-disjoint stars intersecting C. But by hypothesis, every star intersecting L also intersects C. We deduce that there are at most $|C|$ stars intersecting $C \cup L$, and so if $G \in \star(k, r)$, there are at least $k - |C|$ stars in $G[V - (C \cup L)]$: $G[V \setminus (C \cup L)] \in \star(k - |C|, r)$. □

Note that as both rules modify a graph by deleting vertices, any graph G' that is derived from a graph $G \in \mathcal{G}_d$ by an application of Rules 1 or 2 is also in $\mathcal{G}_d$.

Recall the Expansion Lemma, which is a generalization of the well-known Hall's theorem.

Lemma 2. (**Expansion Lemma**)[9] *Let r be a positive integer, and let m be the size of the maximum matching in a bipartite graph G with vertex bipartition $X \cup Y$. If $|Y| > rm$, and there are no isolated vertices in Y, then there exist nonempty vertex sets $S \subseteq X, T \subseteq Y$ such that S has $|S|$ r-stars in T and no vertex in T has a neighbor outside S. Furthermore, the sets S, T can be found in polynomial time in the size of G.*

Henceforth, we will use the following modified version of the expansion lemma.

Lemma 3. (**Modified Expansion Lemma**) *Let r be a positive integer, and let m be the size of the maximum matching in a bipartite graph G with vertex bipartition $X \cup Y$. If $|Y| > rm$, and there are no isolated vertices in Y, then there exists a polynomial algorithm(in the size of G) which returns a partition $X = A_1 \cup B_1, Y = A_2 \cup B_2$, such that B_1 has $|B_1|$ r-stars in B_2, $E(A_1, B_2) = \emptyset$, and $|A_2| \leq r|A_1|$.*

Proof. Apply the Expansion Lemma on graph G to get nonempty vertex sets $S \subseteq X, T \subseteq Y$ such that S has $|S|$ r-stars in T and no vertex in T has a neighbor in X outside S. Let $X' = X \setminus S$ and $Y' = Y \setminus T$. If $G[X' \cup Y']$ has isolated vertices in Y', move all of them from Y' to T. If $|Y'| \leq r|X'|$, we may return $A_1 = X'$, $A_2 = Y'$, $B_1 = S$, and $B_2 = T$.

So now assume $|Y'| > r|X'|$. In this case, apply the algorithm recursively on $G[X' \cup Y']$ to get a partition $X' = A_1' \cup B_1', Y' = A_2' \cup B_2'$, such that B_1' has $|B_1'|$ stars in B_2', $E(A_1', B_2') = \emptyset$, and $|A_2'| \leq r|A_1'|$. Then return $A_1 = A_1', B_1 = B_1' \cup S, A_2 = A_2', B_2 = B_2' \cup T$. Observe that B_1 has $|B_1'| + |S| = |B_1|$ stars in B_2, $E(A_1, B_2) \subseteq E(A_1', B_2') \cup E(X \setminus S, T) = \emptyset$, and $|A_2| = |A_2'| \leq r|A_1'| = r|A_1|$, as required. As each iteration reduces $|X|$ by at least 1, we will have to apply less than $|X| + |Y|$ iterations, each of which uses at most one application of the Expansion Lemma, and so the algorithm runs in polynomial time. □

Proof of Theorem 1. By exhaustively applying Reduction Rule 1, we may assume we have a simplified graph. Let G be a simplified graph in $\mathcal{G}_d$. Now find a maximal r-star packing of the graph G with q stars. We may assume $q < k$ as otherwise we have a trivial YES-instance. Let S be the set of vertices in this packing, and let $D = V(G) \setminus S$.

For any $u \in D$, let $D[u]$ be the set of vertices $v \in D$ for which there is a path from v to u using only vertices in D - that is, $D[u]$ is the the set of vertices in the component of $G[D]$ containing u. As our star-packing is maximal, $d_D(v) < r$ for every $v \in D$. As $G \in \mathcal{G}_d$, every $v \in D[u]$ has a path to u in $G[D]$ with at most $d-1$ vertices (as otherwise the shortest path in $G[D]$ from v to u is an induced path on at least d vertices). It follows that $|D[u]| \le 1 + r + r^2 + \cdots + r^{d-1} \le r^d$.

We claim that there is a partition of S into $Big(S) \cup Small(S)$, and D into $B(D) \cup U(D)$, such that $U(D)$ has no neighbors in $Small(S) \cup B(D)$ and $|B(D)| \le r^{d+1}|Small(S)|$. Moreover either $(Big(S), U(D))$ is a constellation or $|U(D) \cap N(Big(S))| \le r|Big(S)|$. As $|Small(S)| \le |S| \le (r+1)k$, it follows that either we can apply Reduction Rule 2 on $(Big(S), U(D))$, or $|V(G)| = |S| + |U(D)| + |B(D)| \le |S| + r^{d+1}|Big(S)| + r^{d+1}|Small(S)| \le (r^{d+1} + 1)|S| \le (k-1)(r+1)(r^{d+1} + 1)$.

We will find the required partition of S and D by constructing $Big(S)$, $Small(S)$, $B(D)$, $U(D)$ algorithmically as described below. Throughout, we will preserve the properties that

1. $|B(D)| \le |Small(S)|r^{d+1}$,

2. $U(D)$ has no neighbors in $Small(S) \cup B(D)$.

Initially, set $Big(S) = S, U(D) = D, Small(S) = B(D) = \emptyset$.

While $|U(D) \cap N(Big(S))| > r|Big(S)|$, do the following.

If there is a vertex $u \in Big(S)$ such that $|N(u) \cap U(D)| < r$, let $X = \bigcup\{D[v] : v \in N(u) \cap U(D)\}$. Observe that as $|D[v]| \le r^d$ for all $v \in D$, $|X| < r^{d+1}$. Now set $Small(S) = Small(S) \cup \{u\}$, $Big(S) = Big(S) \setminus \{u\}$, $B(D) = B(D) \cup X$, $U(D) = U(D) \setminus X$. It follows that Property 1 is preserved. Note that no vertex in the new $U(D)$ has a neighbor in X (as all neighbors of X in D lie in X). Similarly no vertex in the new $U(D)$ is adjacent to u (as such a vertex would be in the old $U(D)$ and so would have been added to X). Therefore there are still no edges between the new $U(D)$ and the new $Small(S) \cup B(D)$, and so Property 2 is preserved.

Otherwise (if every vertex $u \in Big(S)$ has $|N(u) \cap U(D)| \ge r$), let H denote the maximal bipartite subgraph of G with vertex partition $Big(S) \cup (U(D) \cap N(Big(S)))$, and apply the Modified Expansion Lemma to H. We will get a partition $Big(S) = A_1 \cup B_1$ and $U(D) \cap N(Big(S)) = A_2 \cup B_2$ such that $E(A_1, B_2) = \emptyset, |A_2| \le r|A_1|$ and B_1 has $|B_1|$ r-stars in B_2.

If the Modified Expansion Lemma returns $B_1 = Big(S)$, then we claim that $(Big(S), U(D))$ is a constellation. To see this, firstly note that $|Big(S)|$ has $|Big(S)|$ r-stars in $U(D)$. Secondly, note that since we chose the vertices of a maximal star packing for S, there is no r-star contained in $G[U(D)]$. As $U(D)$ has no neighbors in $Small(S) \cup B(D)$, it follows that there is no r-star intersecting $U(D)$ in $G \setminus Big(S)$. Thus $(Big(S), U(D))$ is a constellation, and the claim is proved. In this case the algorithm stops.

So now assume that the Modified Expansion Lemma returns $Big(S) = A_1 \cup B_1$ with $A_1 \neq \emptyset$. Let $X = \bigcup\{D[v] : v \in N(A_1) \cap U(D)\}$. Note that as $E(A_1, B_2) = \emptyset$ and $|A_2| \leq r|A_1|$, we have $|X| \leq |\bigcup\{D[v] : v \in A_2\}| \leq |A_2|r^d \leq |A_1|r^{d+1}$. Then let $Small(S) = Small(S) \cup A_1$, $Big(S) = Big(S) \setminus A_1$, $B(D) = B(D) \cup X$, $U(D) = U(D) \setminus X$. Note that after this move, we still have that $|B(D)| \leq |Small(S)|r^{d+1}$, and $U(D)$ has no neighbors in $Small(S) \cup B(D)$.

Note that in either case, $|Big(S)|$ strictly decreases, so the algorithm must eventually terminate, either because $(Big(S), U(D))$ is a constellation, or because $|U(D) \cap N(Big(S))| \leq r|Big(S)|$. If $(Big(S), U(D))$ is a constellation, apply Reduction Rule 2 using $(Big(S), U(D))$. This gives us a partition in which $Big(S) = U(D) = \emptyset$. Thus in either case, we have that $|U(D) \cap N(Big(S))| \leq r|Big(S)|$. Note that every vertex $u \in U(D)$ is in $D[v]$ for some $v \in N(S)$ (as otherwise, either $\max\{d(v) : v \in N[u]\} < r$ or $G[D]$ contains an r-star, a contradiction in either case). Moreover such a v must be in $U(D) \cap N(Big(S))$, as there are no edges between $U(D)$ and $Small(S) \cup B(D)$. Thus $|U(D)| \leq r^d|U(D) \cap N(Big(S))| \leq r^{d+1}|Big(S)|$. Then we have $|V(G)| = |S| + |U(D)| + |B(D)| \leq |S| + r^{d+1}|Big(S)| + r^{d+1}|Small(S)| \leq (r^{d+1}+1)|S| \leq (k-1)(r+1)(r^{d+1}+1) = O(k)$. $\square$

4 Proof of Theorem 2

A *split graph* is a graph where the vertex set can be partitioned into a clique and an independent set.

An instance of the well-known $\mathcal{NP}$-hard problem 3-DIMENSIONAL MATCHING contains a vertex set that can be partitioned into three equally large sets V_1, V_2, V_3 (also called partite sets). Let k denote the size of each of V_1, V_2, V_3. It furthermore contains a number of 3-sets containing exactly one vertex from each V_i, $i = 1, 2, 3$. The problem is to decide if there exists a set of k vertex disjoint 3-sets (which would then cover all vertices). Such a set of k vertex disjoint 3-sets is called a perfect matching. The 3-sets are also called *edges* (or *hyperedges*).

Theorem 4. *Let $r \geq 3$. It is $\mathcal{NP}$-hard to decide if the vertex set of a split graph can be partitioned into vertex disjoint copies of $K_{1,r}$.*

Proof. We will reduce from 3-DIMENSIONAL MATCHING. Let $\mathcal{I}$ be an instance of 3-dimensional matching. Let V_1, V_2, V_3 denote the three partite sets of $\mathcal{I}$ and let E denote the set of edges in $\mathcal{I}$. Let $m = |E|$ and $k = |V_1| = |V_2| = |V_3|$. We will build a split graph $G_{\mathcal{I}}$ as follows. Let $V = V_1 \cup V_2 \cup V_3$ be the vertices of $\mathcal{I}$. Let X_1 be a set of m vertices and X_2 be a set of $m - k$ vertices and let $X = X_1 \cup X_2$. Let Y be a set of $(m - k)(r - 1)$ vertices and let W be a set of $k(r - 3)$ vertices (if $r = 3$ then W is empty). Let the vertex set of $G_{\mathcal{I}}$ be $V \cup X \cup Y \cup W$.

Add edges such that X becomes a clique in $G_{\mathcal{I}}$. Let each vertex in X_1 correspond to a distinct edge in E and connect that vertex with the 3 vertices in V which belongs to the corresponding edge in E. Furthermore add all edges from X_1 to W. Finally, for each vertex in X_2 add $r - 1$ edges to Y in such a way that each vertex in Y ends up with degree one in $G_{\mathcal{I}}$. This completes the construction of $G_{\mathcal{I}}$.

Clearly $G_{\mathcal{I}}$ is a split graph as X is a clique and $V \cup Y \cup W$ is an independent set. We will now show that the vertex set of $G_{\mathcal{I}}$ can be partitioned into vertex disjoint copies of $K_{1,r}$ if and only if $\mathcal{I}$ has a perfect matching.

First assume that $\mathcal{I}$ has a perfect matching. Let $E' \subseteq E$ denote the edges of the perfect matching. For the vertices in X_1 that correspond to the edges in E' we include the three edges from each such vertex to V as well as $r - 3$ edges to W. This can be done such that we obtain k vertex disjoint copies of $K_{1,r}$ covering all of V and W as well as k vertices from X_1. Now for each vertex in X_2 include the $r - 1$ edges to Y as well as one edge to an unused vertex in X_1. This can be done such that we obtain an additional $m - k$ vertex disjoint copies of $K_{1,r}$. We have now constructed m vertex disjoint copies of $K_{1,r}$ which covers all the vertices in $G_{\mathcal{I}}$, as required.

Now assume that the vertex set of $G_{\mathcal{I}}$ can be partitioned into vertex disjoint copies of $K_{1,r}$. As $|V \cup W \cup Y \cup X| = m(r + 1)$ we note that we have m vertex disjoint copies of $K_{1,r}$, which we will denote by $\mathcal{K}$. As all vertices in Y need to be included in such copies we note that every vertex of X_2 is the center vertex of a $K_{1,r}$. Let $\mathcal{K}'$ denote these $m - k$ copies of $K_{1,r}$. Each $K_{1,r}$ in $\mathcal{K}'$ must include 1 edge from X_2 to X_1. These $m - k$ edges form a matching, implying that $m - k$ vertices of X_1 also belong to the copies of $K_{1,r}$ in $\mathcal{K}'$. This leaves k vertices in X_1 that are uncovered and rk vertices in $V \cup W$ that are uncovered. Furthermore, as $V \cup W$ is an independent set, each copy of $K_{1,r}$ in $\mathcal{K} \setminus \mathcal{K}'$ must contain a vertex of X_1. As $|\mathcal{K} \setminus \mathcal{K}'| = k$ we note that the k copies of $K_{1,r}$ in $\mathcal{K} \setminus \mathcal{K}'$ must include exactly one vertex from X_1. Also as each vertex in X_1 has exactly three neighbours in V, each such $K_{1,r}$ also contains 3 vertices from V (as V needs to be covered) and therefore $r - 3$ vertices form W. Therefore the k vertices in X_1 that belong to copies of $K_{1,r}$ in $\mathcal{K} \setminus \mathcal{K}'$ correspond to k edges in E which form a perfect matching in $G_{\mathcal{I}}$.

This completes the proof as we have shown that $G_{\mathcal{I}}$ can be partitioned

into vertex disjoint copies of $K_{1,r}$ if and only if $\mathcal{I}$ has a perfect matching. $\quad\square$

The following lemma is known. We give the simple proof for completeness.

Lemma 4. *No split graph contains an induced path on 5 vertices.*

Proof. Assume G is a split graph where $V(G)$ is partitioned into an independent set I and a clique C. For the sake of contradiction assume that $P = p_0p_1p_2p_3p_4$ is an induced P_5 in G. As I is independent we note that $\{p_0, p_1\} \cap C \neq \emptyset$ and $\{p_3, p_4\} \cap C \neq \emptyset$. As C is a clique there is therefore an edge from a vertex in $\{p_0, p_1\}$ to a vertex in $\{p_3, p_4\}$. This edge implies that P is not an induced P_5 in G, a contradiction. $\quad\square$

Proof of Theorem 2. By Lemma 4, $\mathcal{G}_5$ contains all split graphs. The result now follows immediately from Theorem 4. $\quad\square$

References

[1] F. Barbero, G. Gutin, M. Jones, B. Sheng, and A. Yeo, Linear-Vertex Kernel for the Problem of Packing r-Stars into a Graph without Long Induced Paths, arXiv:1510.03564, Oct 2015.

[2] R. Bar-Yehuda, M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira, Scheduling split intervals, in 30th Annu. ACM-SIAM Symp. on Discrete Algorithms, 2002, pp. 732–741.

[3] R. Bejar, B. Krishnamachari, C. Gomes, and B. Selman, Distributed constraint satisfaction in a wireless sensor tracking system, Workshop on Distributed Constraint Reasoning, Internat. Joint Conf. on Artificial Intelligence, 2001.

[4] A. Brandstädt, V.B. Le, and J.P. Spinrad. Graph Classes: A Survey, SIAM, 1999.

[5] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, Parameterized Algorithms, Springer, 2015.

[6] R.G. Downey and M.R. Fellows, Foundations of Parameterized Complexity, Springer, 2013.

[7] M. Fellows, J. Guo, H. Moser, and R. Niedermeier, A Generalization of Nemhauser and Trotter's Local Optimization Theorem, J. Comput. Syst. Sci. 77:1141–1158, 2011.

[8] M. Fellows, P. Heggernes, F. Rosamond, C. Sloper, and J.A. Telle, Finding k Disjoint Triangles in an Arbitrary Graph. In WG'05, Lect. Notes Comput. Sci. 3353:235–244, 2005

[9] F.V. Fomin, D. Lokshtanov, N. Misra, G. Philip, and S. Saurabh, Hitting forbidden minors: Approximation and Kernelization. In STACS 2011, LIPIcs 9:189–200, 2011.

[10] J. Guo and R. Niedermeier. Linear problem kernels for NP-hard problems on planar graphs. In ICALP 2007, Lect. Notes Comput. Sci. 4596:375-386, 2007.

[11] D.G. Kirkpatrick and P. Hell, On the completeness of a generalized matching problem. In 10th STOC, ACM Symposium on Theory of Computing, 240–245, 1978.

[12] S. Kratsch, Recent developments in kernelization: A survey. Bulletin EATCS, no. 113, 2014.

[13] D. Lokshtanov, N. Misra, and S. Saurabh, Kernelization - preprocessing with a guarantee. Lect. Notes Comput. Sci. 7370:129-161, 2012.

[14] E. Prieto and C. Sloper, Looking at the stars, Theor. Comput. Sci. 351:437–445, 2006.

[15] J. Wang, D. Ning, Q. Feng, and J. Chen, An improved parameterized algorithm for a generalized matching problem, In TAMC'08, Lect. Notes Comput. Sci., 4978:212–222, 2008.