Kernelization Lower Bounds for Finding Constant-Size Subgraphs

Till Fluschnik[⋆,1], George B. Mertzios[⋆⋆,2], and André Nichterlein[⋆⋆⋆,1]

[1] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany,
{till.fluschnik,andre.nichterlein}@tu-berlin.de
[2] Department of Computer Science, Durham University, UK,
george.mertzios@durham.ac.uk

Abstract. Kernelization is an important tool in parameterized algorithmics. Given an input instance accompanied by a parameter, the goal is to compute in polynomial time an equivalent instance of the same problem such that the size of the reduced instance only depends on the parameter and not on the size of the original instance. In this paper, we provide a first conceptual study on limits of kernelization for several *polynomial-time* solvable problems. For instance, we consider the problem of finding a triangle with negative sum of edge weights parameterized by the maximum degree of the input graph. We prove that a linear-time computable strict kernel of truly subcubic size for this problem violates the popular APSP-conjecture.

1 Introduction

Kernelization is the main mathematical concept for provably efficient preprocessing of computationally hard problems. This concept has been extensively studied (see, e.g., [17,21,26,27]) and it has great potential for delivering practically relevant algorithms [24,31]. In a nutshell, the aim is to significantly and efficiently reduce a given instance of a parameterized problem to its "computationally hard core". Formally, given an instance $(x, k) \in \{0, 1\}^* \times \mathbb{N}$ of a parameterized problem L, a *kernelization* for L is an algorithm that computes in polynomial time an instance (x', k'), called kernel, such that (i) $(x, k) \in L \iff (x', k') \in L$ and (ii) $|x'| + k' \leq f(k)$, for some computable function f. Although studied mostly for NP-hard problems, it is natural to apply this concept also to polynomial-time solvable problems as done e.g. for finding maximum matchings [29]. It is thus also important to know the *limits* of this concept. In this paper we initiate a systematic approach to derive *kernelization lower bounds* for problems in P. We demonstrate our techniques at the example of subgraph isomorphism problems where the sought induced subgraph has constant size and is connected.

When kernelization is studied on NP-hard problems (where polynomial running times are considered computationally "tractable"), the main point of interest

becomes the *size* $f(k)$ of the kernel with respect to the parameter k. In particular, from a theoretical point of view, one typically wishes to minimize the kernel size to an—ideally—polynomial function f of small degree. As every decision problem in P admits a kernelization which simply solves the input instance and produces a kernel of size $O(1)$ (encoding the YES/NO answer), it is crucial to investigate the *trade-off* between (i) the size of the kernel and (ii) the running time of the kernelization algorithm. The following notion captures this trade-off: An (a, b)-*kernelization* for a parameterized problem L is an algorithm that, given any instance $(x, k) \in \{0, 1\}^* \times \mathbb{N}$, computes in $O(a(|x|))$ time an instance (x', k') such that (i) $(x, k) \in L \iff (x', k') \in L$ and (ii) $|x'| + k' \in O(b(k))$.

Kernelization for problems in P is part of the recently introduced framework "FPT in P" [20]. This framework is recently applied to investigate parameterized algorithms and complexity for problems in P [3,14,16,20,29]. Studying lower bounds for kernelization for problems in P is—as it turns out—strongly connected to the active research field of *lower bounds* on the running times of polynomial-time solvable problems (see, e.g, [1,2,3,7]). These running time lower bounds rely on popular conjectures like the Strong Exponential Time Hypothesis (SETH) [22,23] or the 3SUM-conjecture [19], for instance.

In contrast to NP-hard problems, only little is known about kernelization lower bounds for problems in P. To the best of our knowledge all known kernelization lower bounds follow trivially from the corresponding lower bounds of the running time: For instance, assuming SETH, it is known that (i) the hyperbolicity and (ii) the diameter of a graph cannot be computed in $2^{o(k)} \cdot n^{2-\varepsilon}$ time for any $\varepsilon > 0$, where k is (i) the vertex cover number and (ii) the treewidth of the graph [14,3]. This implies that both problems do not admit an $(n^{2-\varepsilon}, 2^{o(k)})$-kernelization—a kernel with $2^{o(k)}$ vertices computable in $O(n^{2-\varepsilon})$ time—since such a kernelization yields an algorithm running in $O(2^{o(k)} + n^{2-\varepsilon})$ time.

In this paper we initiate a systematic approach to derive kernelization lower bounds for problems in P for a—very natural—special type of kernels.

Definition 1 (strict (a, b)-kernelization). *A* strict (a, b)-kernelization *for a parameterized problem L is an algorithm that given any instance $(x, k) \in \{0, 1\}^* \times \mathbb{N}$ computes in $O(a(|x|))$ time an instance (x', k') such that (i) $(x, k) \in L \iff (x', k') \in L$, (ii) $|x'| + k' \in O(b(k))$, and (iii) $k' \leq k$.*

Chen et al. [8] introduced a framework to exclude strict kernels for NP-hard problems, assuming that P $\neq$ NP. Fernau et al. [13] applied the framework to a wide variety of FPT problems and studied it on "less" strict kernelizations. The framework [8,13] is based on the notion of *(strong) diminishers*:

Definition 2 (a-diminisher). *An a-diminisher for a parameterized problem L is an algorithm that given any instance $(x, k) \in \{0, 1\}^* \times \mathbb{N}$ in $O(a(|x|))$ time either decides whether $(x, k) \in L$ or computes an instance (x', k') such that (i) $(x, k) \in L \iff (x', k') \in L$, and (ii) $k' < k$. A strong a-diminisher for L is an a-diminisher for L with $k' < k/c$ for some constant $c > 1$.*

Table 1. Overview of our results. Here, k is interchangeably the order of the largest connected component, the degeneracy, or the maximum degree.

	Negative Weight Triangle (NWT)	Triangle Collection (TC)
lower bounds (Thm. 2)	No strict (n^α, k^β)-kernelization with $\alpha, \beta \geq 1$ and $\alpha \cdot \beta < 3$, assuming:	
	the APSP-conjecture.	the SETH, APSP-, or 3SUM-conjecture.
kernel (Thm. 3)	Strict $(n^{(3+\varepsilon)/(1+\varepsilon)}, k^{1+\varepsilon})$-kernelization for every $\varepsilon > 0$, e.g. strict $(n^{5/3}, k^3)$-kernelization.	

Our Contributions. We adapt the diminisher framework [8,13] to prove kernelization lower bounds for problems in P. Our results concern the H-Subgraph Isomorphism *(H-SI)* problem[3] for *constant-sized connected* graphs H. As a running example, we focus on the fundamental case where H is a triangle and we present diminishers (along with kernelization lower bounds) for the following weighted and colored variants of the problem:

Negative Weight Triangle (NWT)
Input: An undirected graph G with edge weights $w \colon E(G) \to \mathbb{Z}$.
Question: Is there a triangle T in G with $\sum_{e \in E(T)} w(e) < 0$?

Triangle Collection (TC)
Input: An undirected graph G with surjective coloring $\mathrm{col} \colon V(G) \to [f]$.
Question: Does there for all color-triples $C \in \binom{[f]}{3}$ exist a triangle with vertex set $T = \{x, y, z\}$ in G such that $\mathrm{col}(T) = C$?

NWT and TC are conditionally hard: If NWT admits a *truly* subcubic algorithm—that is, with running time $O(n^{3-\varepsilon})$, $\varepsilon > 0$—then APSP also admits a truly subcubic algorithm, breaking the APSP-conjecture [30]. A truly subcubic algorithm for TC breaks the SETH, the 3SUM-, and the APSP-conjecture [4].

For both NWT and TC we consider three parameters (in decreasing order): (i) order (that is, the number of vertices) of the largest connected component, (ii) maximum degree, and (iii) degeneracy. We prove that both NWT and TC admit a strong linear-time diminisher for all these three parameters. Together with the conditional hardness, we then obtain lower bounds on strict kernelization. Our results are summarized in Table 1.

Complementing our lower bounds, we prove a strict $(n^{5/3}, k^3)$-kernelization for NWT and TC (k being any of the three aforementioned parameters) and a strict $(n \cdot \Delta^{\lfloor c/2 \rfloor + 1}, \Delta^{\lfloor c/2 \rfloor + 1})$-Turing kernelization for H-Subgraph Isomorphism when parameterized by the maximum degree Δ, where $c = |V(H)|$.

Notation and Preliminaries. We use standard notation from parameterized complexity [10] and graph theory [11]. For an integer j, we define $[j] := \{1, \dots, j\}$.

---

[3] The H-Subgraph Isomorphism asks, given an undirected graph $G = (V, E)$, whether G contains H as a subgraph.

2   Frameworks to Exclude Polynomial Kernelizations

We briefly recall the existing frameworks to exclude (strict) polynomial-size kernels for NP-hard problems. We further discuss the difficulties that appear when transferring these approaches to polynomial-time solvable problems.

Composition Framework. The frequently used (cross-)composition frameworks [5,18,6] are the tools to exclude polynomial-size problem kernels under the assumption NP $\subseteq$ coNP/poly. There are some issues when adapting these frameworks for problems in P. We discuss the issues using the H-SUBGRAPH ISOMORPHISM (H-SI) problem for constant-sized connected H.

Adapting the proofs of Bodlaender et al. [5] and Fortnow and Santhanam [18] for H-SI leads to the following: "If H-SI parameterized by the order k of the largest connected component admits an $(n^c, k^{c'})$-kernelization, then $\overline{H\text{-SI}} \in$ NTIME$(n^{c'(c+1)})/n^{c+1}$." Since there exists a trivial $O(n^{|H|+1})$-time brute-force algorithm for H-SI, there also exist trivial polynomial-time computable kernels for H-SI. Hence, we have to stick with specifically chosen c and c' (with $c \cdot c' < |H|$). Furthermore, we cannot transfer these results easily to other problems in P due to the lack of a suitable completeness theory (H-SI belongs to P).

One drawback of the composition approach for any problem L in P is the lack of clarity on the assumption's ($\overline{L} \notin$ NTIME$(n^{c'(c+1)})/n^{c+1}$) reasonability. Moreover, due to a missing equivalent to the NP-completeness theory, the assumption bases on specific problems and not on complexity classes.

Strict Kernelization and Diminishers. Chen et al. [8] introduced a framework to exclude *strict* kernelization, that is, kernelization that do not allow an increase in the value of the parameter in the obtained kernel instance. This framework builds on the assumption P $\neq$ NP and can be easily adapted to exclude strict kernels for polynomial-time solvable problems. Recall that for problems in P, both the size of the kernel *and* the kernelization running time are important.

Theorem 1 ($\star$[4]). *Let L be a parameterized problem with parameter k such that each instance with parameter $k \leq c$ for some constant $c > 0$ is a trivial instance of L. If L with parameter k admits a strict (a, b)-kernelization and an a'-diminisher (a strong a'-diminisher), then any instance (x, k) is solvable in $O(k \cdot (a(a'(b(k))) + a(|x|))$ time (in $O(\log k \cdot (a(a'(b(k))) + a(|x|))$ time).*

We point out that—in contrast to "classic" kernelization for NP-hard problems—for two parameters k and k' for a problem L such that k' is stronger [25] than k, a strict kernelization regarding k does not imply a strict kernelization regarding k'.

Reductions for Transferring Kernels. There are two issues when using the strategy of *polynomial parameter transformations* to transfer results of Theorem 1 along polynomial-time solvable problems: First, we need to require the transformation to be computable "fast" enough and that the parameter does not

---

[4] Results marked with ($\star$) are deferred to a long version [15] of the paper.

increase ($k' \leq k$). Second, in order to transfer a strict kernel we need to show a reverse transformation from L' to L which again is computable "quick" enough and does not increase the parameter. Hence, we essentially need to show that the two problems L and L' are *equivalent* under these restrictive transformations.

3 Kernelization Lower Bounds via Diminishers

In this section, we present diminishers for H-SUBGRAPH ISOMORPHISM (H-SI) for connected H with respect to the structural parameters (i) order ℓ of the largest connected component, (ii) maximum degree Δ, and (iii) degeneracy d. Observe that $d \leq \Delta \leq \ell$ in every graph. These lead to our following main result.

Theorem 2. *If NWT (TC) parameterized by k being the (i) order ℓ of the largest connected component, (ii) maximum degree Δ, or (iii) degeneracy d admits a strict (n^α, k^β)-kernel for constants $\alpha, \beta \geq 1$ with $\alpha \cdot \beta < 3$, then the APSP-conjecture (the SETH, the 3SUM-, and the APSP-conjecture) breaks.*

Parameter Order of the Largest Connected Component. In the following, we prove a linear-time strong diminisher regarding the parameter order of the largest connected component for problems of finding constant-size subgraphs (with some specific property). The idea behind our diminisher is depicted as follows: for each connected component, partition the connected component into small parts and then take the union of not too many parts to construct new (connected) components (see Figure 1 for an illustration of the idea with H being a triangle).

Construction 1. Let H be an arbitrary but fixed connected constant-size graph of order $c > 1$. Let $G = (V, E)$ be a graph with the largest connected component being of order ℓ. First, compute in $O(n + m)$ time the connected components $G_1, \ldots, G_r$ of G. Then, construct a graph G' as follows.

Let G' be initially the empty graph. If $\ell \leq 4c$, then set $G' = G$. Otherwise, if $\ell > 4c$, then construct G' as follows. For each connected component $G_i = (V_i, E_i)$, do the following. If the connected component $G_i = (V_i, E_i)$ is of order at most $\ell/2$, then add G_i to G'. Otherwise, if $n_i := |V_i| > \ell/2$, then we partition V_i as follows. Without loss of generality let V_i be enumerated as $V_i = \{v_i^1, \ldots, v_i^{n_i}\}$. For every $p \in \{1, \ldots, 4c\}$, define $V_i^p := \{v_i^q \in V_i \mid q \bmod 4c = p-1\}$. This defines the partition $V_i = V_i^1 \uplus \cdots \uplus V_i^{4c}$. Then, for each $\{a_1, \ldots, a_c\} \in \binom{[4c]}{c}$, add the graph $G[V_i^{a_1} \cup \ldots \cup V_i^{a_c}]$ to G'. This completes the construction.   ◇

Employing Construction 1, we obtain the following.

Proposition 1 ($\star$). *NWT and TC parameterized by the order ℓ of the largest connected component admit a strong $(n + m)$-diminisher.*

There is a straight-forward $O(k^2 \cdot n)$-time algorithm for NWT and TC: Check for each vertex all pairs of other vertices in the same connected component. However, under the APSP-conjecture (and SETH for TC) there are no $O(n^{3-\varepsilon})$-time algorithms for any $\varepsilon > 0$ [4,30]. Combining this with our diminisher in Proposition 1 we can exclude certain strict kernels as shown below.
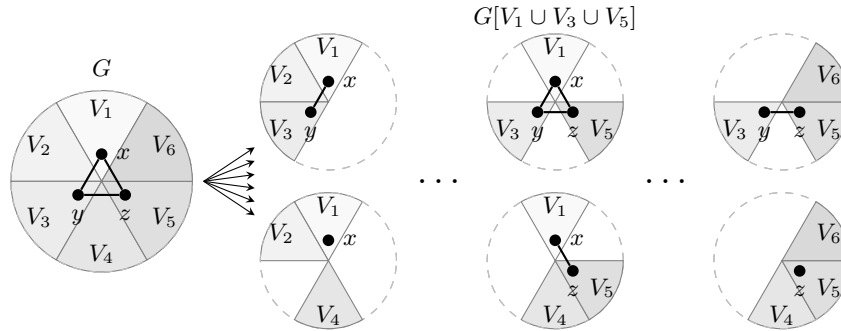
Fig. 1. Schematic illustration of the idea behind our diminisher for the parameter order of the largest connected component.

Proof (of Theorem 2(i)). By Proposition 1, we know that NWT admits a strong $(n + m)$-diminisher. Suppose that NWT admits a strict (n^α, k^β)-kernel for $\alpha \geq 1, \beta \geq 1$ with $\alpha \cdot \beta = 3 - \varepsilon_0$, $\varepsilon_0 > 0$. It follows by Theorem 1 that NWT is solvable in $t(n, k) \in O(k^{\beta \cdot \alpha} \log(k) + n^\alpha)$ time. Observe that $\log(k) \in O(k^{\varepsilon_1})$ for $0 < \varepsilon_1 < \varepsilon_0$. Together with $k \leq n$ and $\alpha \cdot \beta = 3 - \varepsilon_0$ we get $t(n, k) \in O(n^{3-\varepsilon})$ with $\varepsilon = \varepsilon_0 - \varepsilon_1 > 0$. Hence, the APSP-conjecture breaks [30]. The proof for TC works analogously. □

Parameter Maximum Degree. The diminisher described in Construction 1 does not necessarily decrease the maximum degree of the graph. We thus adapt the diminisher to partition the edges of the given graph (using an (improper) edge-coloring) instead of its vertices. Furthermore, if H is of order c, then H can have up to $\binom{c}{2} \leq c^2$ edges. Thus, our diminisher considers all possibilities to choose c^2 (instead of c) parts of the partition. For the partitioning step, we need the following.

Lemma 1 ($\star$). *Let $G = (V, E)$ be a graph with maximum degree Δ and let $b \in \mathbb{N}$. One can compute in $O(b(n + m))$ time an (improper) edge-coloring* col: $E \rightarrow \mathbb{N}$ *with less than $2b$ colors such that each vertex is incident to at most $\lceil \Delta/b \rceil$ edges of the same color.*

Construction 2. Let H be an arbitrary but fixed connected constant-size graph of order $c > 1$. Let $G = (V, E)$ be a graph with maximum degree Δ. First, employ Lemma 1 to compute an (improper) edge-coloring col: $E \rightarrow \mathbb{N}$ with $4c^2 \leq f < 8c^2$ many colors (without loss of generality we assume $\Im(\text{col}) = \{1, \ldots, f\}$) such that each vertex is incident to at most $\lceil \Delta/(4c^2) \rceil$ edges of the same color.

Now, construct a graph G' as follows. Let G' be initially the empty graph. If $\Delta \leq 4c^2$, then set $G' = G$. Otherwise, if $\Delta > 4c^2$, then construct G' as follows. We first partition E: Let E^p be the edges of color p for every $p \in \{1, \ldots, f\}$. Clearly, $E = E^1 \uplus \cdots \uplus E^f$. Then, for each $\{a_1, \ldots, a_{c^2}\} \in \binom{[f]}{c^2}$, add the graph $(V, E^{a_1} \cup \ldots \cup E^{a_{c^2}})$ to G'. This completes the construction. ◇

Proposition 2 ($\star$). *NWT and TC parameterized by maximum degree Δ admit a strong $(n + m)$-diminisher.*

Parameter Degeneracy. The degeneracy of a graph is the smallest number d such that every induced subgraph contains a vertex of degree at most d. For parameter degeneracy, the diminisher follows the same idea as the diminisher for the parameter maximum degree (see Construction 2). The only difference between the two diminishers is how the partition of edge set is obtained.

Construction 3. Let H be an arbitrary but fixed constant-size graph of order $c > 1$. Let $G = (V, E)$ be a graph with degeneracy d. First, compute a degeneracy ordering[5] σ in $O(n + m)$ time [28]. Construct a graph G' as follows.

Let G' be initially the empty graph. If $d \leq 4c^2$, then set $G' = G$. Otherwise, if $d > 4c^2$, then construct G' as follows. First, for each vertex $v \in V$, we partition the edge set $E_v := \{\{v, w\} \in E \mid \sigma(v) < \sigma(w)\}$ going to the right of v with respect to σ into $4c^2$ parts. Let E_v be enumerated as $\{e_1, \ldots, e_{|E_v|}\}$. For each v, we define $E_v^p := \{e_i \in E_v \mid i \bmod 4c^2 = p - 1\}$ for every $p \in [4c^2]$. Clearly, $E_v = E_v^1 \uplus \cdots \uplus E_v^{4c^2}$. Next, we define $E^p := \bigcup_{v \in V} E_v^p$ for every $p \in [4c^2]$. Clearly, $E = \biguplus_{1 \leq p \leq 4c^2} E^p = \biguplus_{1 \leq p \leq 4c^2} \biguplus_{v \in V} E_v^p$. Then, for each $\{a_1, \ldots, a_{c^2}\} \in \binom{[4c^2]}{c^2}$, add the graph $(V, E^{a_1} \cup \ldots \cup E^{a_{c^2}})$ to G'. This completes the construction. $\diamond$

Proposition 3 ($\star$). *NWT and TC parameterized by degeneracy admit a strong $(n + m)$-diminisher.*

4 (Turing) Kernelization Upper Bounds

We complement our results on kernelization lower bounds by showing straightforward strict kernel results for H-Subgraph Isomorphism for connected constant-size H to show the limits of any approach showing kernel lower bounds.

Strict Turing Kernelization. For the parameters order of the largest connected component and maximum degree, we present strict (a, b)-Turing kernels:

Definition 3. *A* strict (a, b)-Turing kernelization *for a parameterized problem L is an algorithm that decides every input instance (x, k) in time $O(a(|x|))$ given access to an oracle that decides whether $(x', k') \in L$ for every instance (x', k') with $|x'| + k' \leq b(k)$ in constant time.*

Note that the diminisher framework in its current form cannot be applied to exclude (strict) (a, b)-Turing kernelizations. In fact, it is easy to see that H-Subgraph Isomorphism for connected constant-size H parameterized by the order ℓ of the largest connected component admits an $(n + m, \ell^2)$-Turing kernel, as each oracle call is on a connected component (which is of size at most $O(\ell^2)$) of the input graph. We present a strict Turing kernelization for H-SI for connected constant-size H parameterized by maximum degree Δ.

---

[5] This is an ordering of the vertices such that each vertex v has at most d neighbors ordered after v.

Proposition 4 (⋆). *H-*SUBGRAPH ISOMORPHISM *for connected H with $c = |V(H)|$ parameterized by maximum degree Δ admits a strict $(n \cdot \Delta \cdot (\Delta-1)^{\lfloor c/2 \rfloor}, \Delta \cdot (\Delta-1)^{\lfloor c/2 \rfloor})$-Turing kernel.*

Running-time Related Strict Kernelization. For NP-hard problems, it is well-known that a decidable problem is fixed-parameter tractable if and only if it admits a kernel [12]. In the proof of the *only if*-statement, one derives a kernel of size only depending on the running time of a fixed-parameter algorithm solving the problem in question. We adapt this idea to derive a strict kernel where the running time and size admit such running time dependencies.

Theorem 3 (⋆). *Let L be a parameterized problem admitting an algorithm solving each instance (x, k) in $k^c \cdot |x|$ time for some constant $c > 0$. Then for every $\varepsilon > 0$, each instance (x, k) admits a strict $(|x|^{1+c/(1+\varepsilon)}, k^{1+\varepsilon})$-kernel.*

NWT and TC are both solvable in $O(k^2 \cdot n)$ time (k being the order ℓ of the largest connected component, the maximum degree Δ, or the degeneracy d [9]). Together with Theorem 3 gives several kernelization results for NWT and TC, for instance, with $\varepsilon = 2$:

Corollary 1. *NWT admits a strict $(n^{5/3}, d^3)$-kernel when parameterized by the degeneracy d of the input graph.*

Note that the presented kernel is a strict (n^α, d^β)-kernel with $\alpha = 5/3$ and $\beta = 3$. As $\alpha \cdot \beta = 5$ in this case, there is a gap between the above kernel and the lower bound of $\alpha \cdot \beta \geq 3$ in Theorem 2(iii). Future work could be to close this gap.

5   Conclusion

We provided the first conceptual analysis of strict kernelization lower bounds for problems solvable in polynomial time. To this end, we used and (slightly) enhanced the parameter diminisher framework [8,13]. Our results for NEGATIVE WEIGHT TRIANGLE and TRIANGLE COLLECTION rely on the APSP-conjecture and SETH, but these assumptions can be replaced with any running-time lower bound known for the problem at hand. Indeed the framework is not difficult to apply and we believe that developing special techniques to design diminishers is a fruitful line of further research.

We point out that the framework excludes certain *trade-offs* between kernel size and running time: the smaller the running time of the diminisher, the larger the size of the strict kernel that can be excluded. However, the framework in its current form cannot be used to exclude the existence of *any* strict kernel of polynomial size in even linear time.

In this work, we only considered parameters that we call *dispersed* parameters, defined as follows. Let G be an instance of a graph problem L, and let $G_1, G_2, \ldots, G_p$ be its connected components, where $p \geq 1$. A parameter k of G is *dispersed* if $k(G)$ (i.e. the value of the parameter k in the graph G) is

equal to $k(G_i)$ for *at least one* connected subgraph G_i of G. Otherwise, if $k(G)$ is *larger* than $k(G_i)$ for *every* connected subgraph G_i of G, then we call k an *aggregated* parameter. In our opinion, it is of independent interest to apply the (strong) diminisher framework to graph problems with aggregated parameters. Note that such a classification into dispersed and aggregated parameters has not been studied previously.

We close with one concrete challenge: Is there a (strong) diminisher for NWT or TC with respect to the (aggregated) parameter feedback vertex number? Note that the disjoint union operation that we use in all our diminishers in Section 3 can increase this parameter.

References

1. Abboud, A., Grandoni, F., Vassilevska Williams, V.: Subcubic equivalences between graph centrality problems, APSP and diameter. In: Proc. of 26th SODA. pp. 1681–1697. SIAM (2015)
2. Abboud, A., Vassilevska Williams, V.: Popular conjectures imply strong lower bounds for dynamic problems. In: Proc. of 55th FOCS. pp. 434–443. IEEE Computer Society (2014)
3. Abboud, A., Vassilevska Williams, V., Wang, J.R.: Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In: Proc. of 27th SODA. pp. 377–391. SIAM (2016)
4. Abboud, A., Vassilevska Williams, V., Yu, H.: Matching triangles and basing hardness on an extremely popular conjecture. In: Proc. of 47th STOC. pp. 41–50. ACM (2015)
5. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comput. System Sci. 75(8), 423–434 (2009)
6. Bodlaender, H.L., Jansen, B.M.P., Kratsch, S.: Kernelization lower bounds by cross-composition. SIAM J. Discrete Math. 28(1), 277–305 (2014)
7. Bringmann, K.: Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. In: Proc. of 55th FOCS. pp. 661–670 (2014)
8. Chen, Y., Flum, J., Müller, M.: Lower bounds for kernelizations and other preprocessing procedures. Theory Comput. Syst. 48(4), 803–839 (2011)
9. Chiba, N., Nishizeki, T.: Arboricity and subgraph listing algorithms. SIAM J. Comput. 14(1), 210–223 (1985)
10. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015)
11. Diestel, R.: Graph Theory, Graduate Texts in Mathematics, vol. 173. Springer, 4th edn. (2010)
12. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Springer (2013)
13. Fernau, H., Fluschnik, T., Hermelin, D., Krebs, A., Molter, H., Niedermeier, R.: Diminishable parameterized problems and strict polynomial kernelization. CoRR abs/1611.03739 (2016)

14. Fluschnik, T., Komusiewicz, C., Mertzios, G.B., Nichterlein, A., Niedermeier, R., Talmon, N.: When can graph hyperbolicity be computed in linear time? In: Proc. 15th WADS. LNCS, vol. 10389, pp. 397–408. Springer (2017)
15. Fluschnik, T., Mertzios, G.B., Nichterlein, A.: Kernelization lower bounds for finding constant size subgraphs. CoRR abs/1710.07601 (2017), `http://arxiv.org/abs/1710.07601`
16. Fomin, F.V., Lokshtanov, D., Pilipczuk, M., Saurabh, S., Wrochna, M.: Fully polynomial-time parameterized computations for graphs and matrices of low treewidth. In: Proc. of 28th SODA. pp. 1419–1432 (2017)
17. Fomin, F.V., Saurabh, S.: Kernelization methods for fixed-parameter tractability. In: Tractability: Practical Approaches to Hard Problems, pp. 260–282. Cambridge University Press (2014)
18. Fortnow, L., Santhanam, R.: Infeasibility of instance compression and succinct PCPs for NP. J. Comput. System Sci. 77(1), 91–106 (2011)
19. Gajentaan, A., Overmars, M.H.: On a class of $O(n^2)$ problems in computational geometry. Comput. Geom. 5, 165–185 (1995)
20. Giannopoulou, A.C., Mertzios, G.B., Niedermeier, R.: Polynomial fixed-parameter algorithms: A case study for longest path on interval graphs. Theor. Comput. Sci. 689, 67–95 (2017)
21. Guo, J., Niedermeier, R.: Invitation to data reduction and problem kernelization. ACM SIGACT News 38(1), 31–45 (2007)
22. Impagliazzo, R., Paturi, R.: On the complexity of k-SAT. J. Comput. System Sci. 62(2), 367–375 (2001)
23. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. System Sci. 63(4), 512–530 (2001)
24. Iwata, Y.: Linear-time kernelization for feedback vertex set. In: Proc. 44th ICALP. LIPIcs, vol. 80, pp. 68:1–68:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017)
25. Komusiewicz, C., Niedermeier, R.: New races in parameterized algorithmics. In: Proc. 37nd MFCS. LNCS, vol. 7464, pp. 19–30. Springer (2012)
26. Kratsch, S.: Recent developments in kernelization: A survey. Bulletin of EATCS 113, 58–97 (2014)
27. Lokshtanov, D., Misra, N., Saurabh, S.: Kernelization - preprocessing with a guarantee. In: The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, vol. 7370, pp. 129–161. Springer (2012)
28. Matula, D.W., Beck, L.L.: Smallest-last ordering and clustering and graph coloring algorithms. J. ACM 30(3), 417–427 (1983)
29. Mertzios, G.B., Nichterlein, A., Niedermeier, R.: The power of data reduction for matching. In: Proc. of 42nd MFCS. LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017), to appear.
30. Vassilevska Williams, V., Williams, R.: Subcubic equivalences between path, matrix and triangle problems. In: Proc. of 51st FOCS. pp. 645–654. IEEE Computer Society (2010)
31. Weihe, K.: Covering trains by stations or the power of data reduction. In: Proc. of 1st ALEX pp. 1–8 (1998)