Exploiting Dense Structures in Parameterized Complexity

William Lochet ✉
Department of Informatics, University of Bergen, Norway

Daniel Lokshtanov ✉
University of California Santa Barbara, CA, USA

Saket Saurabh ✉
Institute of Mathematical Sciences, Chennai, India
Department of Informatics, University of Bergen, Norway

Meirav Zehavi ✉
Ben Gurion University of the Negev, Beer Sheva, Israel

Abstract

Over the past few decades, the study of dense structures from the perspective of approximation algorithms has become a wide area of research. However, from the viewpoint of parameterized algorithm, this area is largely unexplored. In particular, properties of random samples have been successfully deployed to design approximation schemes for a number of fundamental problems on dense structures [Arora et al. FOCS 1995, Goldreich et al. FOCS 1996, Giotis and Guruswami SODA 2006, Karpinksi and Schudy STOC 2009]. In this paper, we fill this gap, and harness the power of random samples as well as structure theory to design kernelization as well as parameterized algorithms on dense structures. In particular, we obtain linear vertex kernels for Edge-Disjoint Paths, Edge Odd Cycle Transversal, Minimum Bisection, d-Way Cut, Multiway Cut and Multicut on everywhere dense graphs. In fact, these kernels are obtained by designing a polynomial-time algorithm when the corresponding parameter is at most $\Omega(n)$. Additionally, we obtain a cubic kernel for Vertex-Disjoint Paths on everywhere dense graphs. In addition to kernelization results, we obtain randomized subexponential-time parameterized algorithms for Edge Odd Cycle Transversal, Minimum Bisection, and d-Way Cut. Finally, we show how all of our results (as well as EPASes for these problems) can be de-randomized.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases Dense graphs, disjoint paths, odd cycle transversal, kernels

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.50

Funding Lochet and Saurabh were supported by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. 819416).
Daniel Lokshtanov: Lokshtanov is supported by BSF grant no. 2018302. and NSF award CCF-2008838.
Saket Saurabh: Saurabh also acknowledges Swarnajayanti Fellowship grant DST/SJF/MSA-01/2017-18.
Meirav Zehavi: Zehavi was supported by Israel Science Foundation (ISF) grant no. 1176/18, and United States – Israel Binational Science Foundation grant no. 2018302.

Acknowledgements We thank one of the referee of a different version of this paper for sketching a proof based on the random walk on expanders for Lemma 10.

1 Introduction

While several interesting optimization problems remain NP-complete even when restricted to sparse graphs or dense graphs, the restriction of a problem to these families of graphs is usually considerably more tractable algorithmically than the problem on general graphs.

With respect to graph classes, sparseness usually refers to families of planar graphs, graphs of bounded genus, graphs excluding some fixed graph H as a minor, graphs of bounded expansion and no-where dense graphs. Here, denseness usually refers to families of graphs with $\Omega(n^2)$ edges. Additionally, sparseness and denseness can be defined for structures beyond graphs – for example, dense 3-SAT instances are those for which the formula has $\Omega(n^3)$ clauses.

In this paper, we focus on designing *deterministic* kernelization algorithms and fixed-parameter tractable (FPT) algorithms for NP-hard problems on dense structures.

We start by defining some basic definitions from Parameterized Complexity, that we make use of. Formally, a *parameterization* of a problem is assigning an integer k to each input instance and we say that a parameterized problem is *fixed-parameter tractable (FPT)* if there is an algorithm that solves the problem in time $f(k) \cdot |I|^{O(1)}$, where $|I|$ is the size of the input and f is an arbitrary computable function depending on the parameter k only. We will also be studying polynomial time preprocessing or kernelization.

A parameterized problem Π is said to admit a *kernel* if there is a polynomial-time algorithm, called a *kernelization algorithm*, that reduces the input instance of Π down to an equivalent instance of Π whose size is bounded by a function $f(k)$ of k. (Here, two instances are equivalent if both of them are either Yes-instances or No-instances.) Such an algorithm is called an $f(k)$-*kernel* for Π. If $f(k)$ is a polynomial function of k, we say that the kernel is a *polynomial kernel*. For more background on Parameterized Complexity and Kernelization, we refer to the following books [21, 15, 23, 46, 25].

1.1 Context of Our Results and Overarching Goals

The algorithmic study of NP-hard problems on dense structures is two decade old and has a rich history. We start by giving definitions of (E)PTAS and denseness that will ease our discussion. A PTAS is an algorithm that takes an instance I of an optimization problem and a parameter $\epsilon > 0$, runs in time $n^{\mathcal{O}(f(1/\epsilon))}$, and produces a solution that is within a factor $1 + \epsilon$ of being optimal. A PTAS with running time $f(1/\epsilon) \cdot n^{\mathcal{O}(1)}$ is called an efficient PTAS (EPTAS).

▶ **Definition 1** ([7, 33]). *A graph on n vertices is δ-*dense *if it has $\delta n^2/2$ edges. It is* everywhere-δ-dense *if the minimum degree is δn. We abbreviate $\Omega(1)$-dense as dense and everywhere-$\Omega(1)$-dense as everywhere-dense.*

Arora, Karger and Karpinski [7] initiated the study of NP-hard problems on dense structures and designed PTASes for several NP-hard optimization problems. Among many other results, they showed that Bisection, k-Way Cut, and Separator admit PTASes on everywhere-dense instances and Max-Cut, Max-d-SAT, and Max-Hypercut(d) admit PTASes on dense instances. The main ingredients of these results are *exhaustive sampling* and its use in approximation of polynomial integer programs. These results lead to a flurry of new ideas and results in this area. Arora, Frieze, and Kaplan [6] used the exhaustive sampling idea to design additive approximation schemes for problems in which feasible solutions are permutations (such as the 0-1 Quadratic Assignment Problem). Frieze and Kannan [27] and, independently, Goldreich, Goldwasser, and Ron [29] showed that exhaustive sampling techniques apply because of certain regularity properties in dense graphs and used this observation to design linear time additive approximation schemes for most of the problems that were considered in [7]. In particular, [27, 29] made PTASes of [7] into EPTASes. Frieze and Kannan [27] also pointed out connections to constructive versions of Szemeredi's Regularity Lemma and Goldreich, Goldwasser, and Ron [29] found its connection in property testing and learning theory based on an idea of degree estimator.

This idea of degree estimator has been extremely useful in further developments in the area. In particular, Giotis and Guruswami [28] used this idea to design a PTAS for correlation clustering in general graphs, when the number of clusters is fixed. That is, they designed a PTAS for d-Correlation Clustering (given an undirected graph G, edit (delete or add) minimum number of edges so that the resulting graph becomes a disjoint union of d cliques) running in time $n^{\mathcal{O}(9^d/\epsilon^2)} \log n$. It is also important to note here that before the paper of Giotis and Guruswami [28], most of the earlier works largely focused on maximization problems. In 2009, Karpinski and Schudy [33] further used the idea of degree estimator and designed linear time EPTASes for several problems, such as d-Correlation Clustering and Fragile Min-d-CSP on everywhere-dense instances. Several other randomized PTASes and EPTAses based on different sets of ideas can be found in [43, 19, 32, 8, 2, 1, 5].

As we established above the algorithmic study of NP-hard problems on dense structures has been extremely rewarding from the perspective of Approximation Algorithms. Could this success be repeated in other algorithmic paradigms meant to cope up with NP-hard problems? In particular, in the field of Parameterized Complexity. This leads to the following question.

> Could we exploit the denseness of structures in designing significantly faster FPT algorithms and polynomial time kernelization algorithm for some of the fundamental problems in the field, the way it has been utilized in the field of approximation algorithms?

Our study shows that the answer is an assertive YES! In particular, we obtain linear kernels for Edge-Disjoint Paths, Edge Odd Cycle Transversal, Minimum Bisection, d-Way Cut, Multiway Cut and Multicut on everywhere dense graphs. In fact, these kernels are obtained by designing a polynomial-time algorithm when the corresponding parameter is $\Omega(n)$. Additionally, we obtain a cubic kernel for Vertex-Disjoint Paths on everywhere dense graphs. In addition to kernelization results, we obtain randomized subexponential-time parameterized algorithms for Edge Odd Cycle Transversal, Minimum Bisection, and d-Way Cut. Finally, we show how all of our results (as well as EPASes for these problems) can be de-randomized.

1.2 Our Results and Methods

In this section we give a brief overview of the problems we address and the results we obtain for these problems. This is complemented with a short discussion on techniques that we apply to design our algorithms.

For maximization problems such as Max Cut on dense graphs, a solution would have size $k = \Omega(n^2)$, which trivially yields solvability in subexponential-time (i.e. $2^{o(k)} \cdot n^{\mathcal{O}(1)}$-time) with respect to k. This is true about several maximization problems. However, this is not the case for well-studied minimization problems such as Edge Odd Cycle Transversal, Minimum Bisection, d-Way Cut, Multiway Cut and Multicut. Thus, a natural class of problems to consider are so called *cut-problems*. The other family of problems for which we do not immediately get an algorithm are *linkage problems*, namely, the Edge-Disjoint Paths and Vertex-Disjoint Paths problems.

We remark that the study of subexponential-time parameterized algorithms of *vertex* (rather than edge) modification problem on everywhere-dense graphs does not make sense for natural problems such as Vertex Cover as such problems become as hard as they are on general graphs (and hence do not admit such algorithms under the ETH). For example, given an instance G of Vertex Cover, create an instance G' of Vertex Cover on everywhere-dense graphs by adding an n-vertex clique whose vertices are all but one adjacent to every vertex of G. Then, the existence of an $2^{o(k)}n^{\mathcal{O}(1)}$-time algorithm for Vertex Cover on everywhere-dense graphs where k is the solution size would imply the existence of a subexponential-time algorithm for Vertex Cover on general graphs with respect to n.

1.2.1 Linkage Problems

The first two problems we address are extremely fundamental in the field of Parameterized Complexity. They are Edge-Disjoint Paths and Vertex-Disjoint Paths. In the Edge-Disjoint Paths problem, we are given a graph G, a set of request pairs $(s_1, t_1), \dots, (s_k, t_k)$, and the objective is to check whether there exist paths $P_1, \dots, P_k$, between s_i and t_i, such that they are pairwise edge disjoint. In the Vertex-Disjoint Paths problem, the input is same as the Edge-Disjoint Paths problem, but the paths $P_1, \dots, P_k$ are suppose to be pairwise vertex disjoint. Both, Edge-Disjoint Paths and Vertex-Disjoint Paths are famously FPT by the graph minor machinery of Robertson and Seymour [48]. However, the $f(k)$ in the running time in the algorithm of Robertson and Seymour [48] and its later improvement is at least triply exponential [36]. Only recently an algorithm with $f(k) = 2^{k^{\mathcal{O}(1)}}$ are designed when the input is restricted to planar graphs [39]. Further, Vertex-Disjoint Paths is not known not to admit a polynomial kernel on general graphs [10]. In this paper we show that both Edge-Disjoint Paths and Vertex-Disjoint Paths admit a polynomial kernel on α-dense graphs. In particular we get the following result about Edge-Disjoint Paths.

▶ **Theorem 2.** *Edge-Disjoint Paths admits an $\mathcal{O}(k)$ vertex kernel on everywhere α-dense graphs.*

Proof of Theorem 2 is obtained by designing a polynomial time algorithm for the Edge-Disjoint Paths problem in α-dense graphs, when the number of demands is small (but still linear) compared to αn. Once this result is proved we know that $k \geq \Omega(n)$, resulting in a linear vertex kernel for the problem.

To design the desired polynomial time algorithm, we use the following strategy. We start by showing that highly edge-connected (linear in n) parts will always contain a solution to an Edge-Disjoint Paths instance. Towards this we first show that if a graph G on n vertices with minimum degree at least cn, then for any pair of vertices x, y of G, if there exists a path between x and y, then there exists a path of length at most $4/c$. We use this result together with high connectivity of G to get the following: Let G be a graph with minimum degree αn, and cn edge-connected for some constant $c \leq \alpha/2$, then any instance of Edge-Disjoint Paths with $k \leq \frac{\alpha n}{8}$ has a solution. Moreover, this solution can be found in polynomial time. Next, we give a lemma that partitions the input graph into small number of parts such that each part has minimum degree and edge-connectivity linear in n.

▶ **Lemma 3.** *For any real α between 0 and 1, there exists a constant $c \leq \alpha/2$ such that, if G is a graph on n vertices and minimum degree αn, then there exists a partition $\mathcal{P}$ of the vertices $V(G)$ into $g \leq \frac{2}{\alpha}$ subsets $V_1, \cdots, V_g$ such that for all $i \in [g]$:*

- *$G[V_i]$ is cn edge-connected.*
- *$G[V_i]$ has minimum degree $\frac{\alpha n}{2}$.*

Moreover, such a partition can be found in polynomial time.

This decomposition is then utilized to complete the proof of Theorem 2.

Our kernelization algorithm for Vertex-Disjoint Paths is more involved, though follows the template outlined for Edge-Disjoint Paths. In particular we obtain the following result.

▶ **Theorem 4** (⋆).[1] *Vertex-Disjoint Paths admits a vertex kernel of size $\mathcal{O}(k^3)$ on everywhere α-dense graphs.*

One of the main technical difficulty in proving Theorem 4 is in adapting the proof of Lemma 3 for Vertex-Disjoint Paths. The main reason being that for Vertex-Disjoint Paths we need to simulate Lemma 3 for vertex connectivity. That is, we need to find *cut-vertices instead of edges*. However, these vertices could have neighbors in many different parts and we cannot say that their relative degree inside a part increases, which is a critical component in the proof of Lemma 3. To mitigate this situation we introduce a vertex set V_0 in the partitioning, that contains all the cut vertices. The whole difficulty lies in carrying this V_0 throughout the process of obtaining the desired partition. However, unlike Edge-Disjoint Paths, getting the desired decomposition in itself does not result in the desired kernel. We need to put in significant technical work to reduce the graph. To achieve this we prove several structural properties of Vertex-Disjoint Paths and its interplay with the parts of $\mathcal{P}$ in order to get the desired kernel.

1.2.2 Cut-Problems

Arguably, a few of the most well-studied cut problems in the realm of Parameterized Complexity are Edge Odd Cycle Transversal, Minimum Bisection, d-Way Cut, Multiway Cut, and Multicut. Input to all these problems are an undirected graph G and an integer k, and the goal is following.

Edge Odd Cycle Transversal: Does there exist a set of at most k edges such that its deletion results in a bipartite graph?

Minimum Bisection: Does there exist a vertex partition (V_1, V_2), such that $||V_1|-|V_2|| \leq 1$, and there are at most k edges with one end-point in V_1 and the other in V_2?

d-Way Cut: Does there exist a set of at most k edges such that its deletion results in at least d connected components?

Multiway Cut: Here, we are also given a vertex subset $T \subseteq V(G)$ (called terminals) and the objective is to test if there exists a set of at most k edges such that after its deletion no two terminals belong to the same connected component.

Multicut: Here, we are also given a set of request $(s_1, t_1), \ldots, (s_\ell, t_\ell)$ and the objective is to test if there exists a set of at most k edges such that after its deletion no request belong to the same connected component.

All the aforementioned problems are extremely well studied [18, 16, 14, 47, 41, 42, 12, 13, 35, 11] and are known to be FPT. However, for most of these problems we know that there can not exist an algorithm with running time $2^{o(k)}n^{\mathcal{O}(1)}$ on general graphs assuming ETH. Further, Edge Odd Cycle Transversal admits a randomized polynomial kernel on general graphs [37, 38]; on the other hand Minimum Bisection and Multicut are known not to admit a polynomial kernel [17, 49]. The kernelization complexity of Multiway Cut is still open. In this paper we obtain the following results about these problems on everywhere dense graphs.

[1] Results marked with (⋆) could be found in the extended version.

▶ **Theorem 5** ($\star$). Edge Odd Cycle Transversal, Minimum Bisection, d-Way Cut, Multiway Cut, *and* Multicut *admit* $\mathcal{O}(k)$ *vertex kernel on everywhere* α*-dense graphs.*

▶ **Theorem 6** ($\star$). Edge Odd Cycle Transversal, *and* Minimum Bisection *admit an algorithm with running time* $2^{\mathcal{O}(\sqrt{k})}n^{\mathcal{O}(1)}$ *on everywhere* α*-dense graphs. Further,* d-Way Cut *admits an algorithm with running time* $2^{\mathcal{O}(\sqrt{k}\log k)}n^{\mathcal{O}(1)}$.

These are the first subexponential time parameterized algorithms for Edge Odd Cycle Transversal, Minimum Bisection, and d-Way Cut on everywhere α-dense graphs. The proof of Theorem 5 is obtained by designing a polynomial time algorithm when the solution size for these problems is smaller than $\alpha \cdot n$ (for some α). This is similar to our kernelization strategy for the Edge-Disjoint Paths problem. For example, if the solution for Edge Odd Cycle Transversal is of size $k \leq \alpha \cdot n$ (for some α), then the problem can be solved in polynomial time, and otherwise $n < k/\alpha$ and hence we already have a kernel at hand.

The proof of these results (Theorems 5 and 6) are similar to each other. Thus, to illustrate our methods we focus on giving intuition for the proof of d-Way Cut. A more formal presentation is left to the extended version. The main ingredient of Theorems 5 and 6 is the following sampling primitive, a simple consequence of Chebyshev's inequality which has been extensively used in designing PTASes and EPTASes in everywhere α-dense graphs.

▶ **Lemma 7** (Degree Estimator Lemma). *For any constants* ϵ_1 *and* ϵ_2*, if* U *is a universe on* n *elements,* $\mathcal{K}$ *is a set of subsets of* U *and* S *is a multi-set obtained by doing* $t(\epsilon_1, \epsilon_2) = \frac{1}{\epsilon_1^2\epsilon_2}$ *independent and uniform random draws in* U*, then with probability at least* $1/2$*, the number of sets* $X \in \mathcal{K}$ *such that* $\left|\frac{|S\cap X|n}{t} - |X|\right| \geq \epsilon_1 n$ *is smaller than* $\epsilon_2|\mathcal{K}|$.

We next show how we use Degree Estimator Lemma for our purpose. Suppose that G is a graph on n vertices and A is a set of linear size $\Omega(n)$. We use Lemma 7 in order to guess the degree of the vertices of $V(G)$ in A *without knowing* the set. That is, to estimate the number of neighbors of a vertex that belong to the set A. Indeed, let us fix some constants ϵ_1 and ϵ_2 and pick uniformly at random a set S of $t = t(\epsilon_1, \epsilon_2) = \frac{1}{\epsilon_1^2\epsilon_2}$ vertices from $V(G)$. Since A is of linear size, with constant probability, all the elements of S belong to A. If this event is satisfied, then by applying Lemma 7 with $U = A$ and $\mathcal{K}$ being the set of neighborhood inside A, we have that with probability at least $1/2$, the number of vertices x such that $\left|\frac{|S\cap N(x)||A|}{t} - d_A(x)\right| \geq \epsilon_1 n$ is smaller than $\epsilon_2 n$. In other words, without knowing A, the value $\frac{|S\cap N(x)||A|}{t}$ provides a good estimation of the degree in A for a large fraction of the vertices in $V(G)$.

Let us now see how we use the aforementioned argument for d-Way Cut. Let (G, k) be an instance of d-Way Cut, where G is a everywhere α-dense graph. Further assume that we are looking for a solution, S, where k is small, say $k \leq \frac{\alpha n}{200}$. Let $(A_1, \ldots, A_d)$ be the connected components after removing the edges in S. Since, $k \leq \frac{\alpha n}{200}$ and every vertex has degree at least αn, this implies that every vertex $x \in A_i$ has degree at least $\alpha n - \frac{\alpha n}{200} \geq \frac{\alpha n}{2}$ in A_i, and degree less than $\frac{\alpha n}{200}$ in the other A_j, for $j \neq i$. It means that $|A_i| \geq \frac{\alpha n}{2}$ for every i, and thus $d \leq \frac{2}{\alpha}$.

The idea now is to estimate the degree of every vertex inside each A_i in two rounds. For the first round we sample d sets $M_1, \ldots, M_d$ of $t = t(\alpha/200, \alpha^2/400)$ vertices each. By applying Lemma 7, with constant probability (because each A_i is linear), each M_i will be a subset of A_i such that the set X_i of vertices x for which $\left|\frac{|M_i\cap N(x)||A_i|}{t} - d_{A_i}(x)\right| \geq n\alpha/200$ is smaller than $n\alpha^2/400$. Assume that this is the case for every i, and let us denote $X = \cup_{i\in[d]} X_i$. Since $d \leq 2/\alpha$, we have that $|X| \leq \alpha n/200$. This means that apart from this small set X, all the other vertices x of G are such that $\frac{|M_i \cap N(x)||A_i|}{t}$ is a good estimate of its degree inside A_i[2]. Let us make our first guess of A_i: for every $i \in [d]$, let A_i' be the set x of vertices of G such that $\frac{|M_i \cap N(x)||A_i|}{t} \geq d(x) - \frac{\alpha n}{25}$. We can then show the following.

▷ Claim 8. For every $i \in [d]$, $(A_i \setminus X) \subseteq A_i'$.

Indeed, for every $x \in (A_i \setminus X)$, we have that $\frac{|M_i \cap N(x)||A_i|}{t} \geq d_{A_i}(x) - n\alpha/200 \geq (d(x) - k) - n\alpha/200 \geq d(x) - \alpha/n$ because $x \notin X_i$. Moreover, for every $j \neq i$, $\frac{|M_j \cap N(x)||A_j|}{t} \leq d_{A_j}(x) + n\alpha/200 \leq n\alpha/50$ because $x \in A_i$ and $x \notin X_j$.

For our second round, we use $d_{A_i'}(x)$ as an estimate for $d_{A_i}(x)$. Indeed, if $x \in A_i$, then Claim 8 implies that $d_{A_i'}(x) \geq d_{A_i}(x) - |X|$, *even if x belongs to X*. However, since $d_{A_i}(x) \geq d(x) - \alpha n/100$, we have that $d_{A_i'}(x) \geq d(x) - \alpha n/50$. Similarly, $d_{A_j'}(x) \leq d_{A_j}(x) + |X| \leq \alpha n/50$. Because $d(x) \geq \alpha n$ for every $x \in G$, we have the following claim.

▷ Claim 9. For every i, A_i is exactly the set of vertices x of G such that $d_{A_i'}(x) \geq d(x) - \alpha n/50$.

This ends the proof of a polynomial algorithm in the case $k \leq \alpha n/100$, which implies the proof of a linear kernel. The proofs for Edge Odd Cycle Transversal, Minimum Bisection, Multiway Cut, and Multicut are almost identical.

When $k \geq \alpha n/100$, we have to be more careful with respect to vertices that are incident to many edges of the solution, say more than $\alpha n/200$. Let us note that all of these problems admit an exact algorithm, by doing a dynamic programming algorithm over subset and applying fast subset-convolution, running in time $2^n n^{\mathcal{O}(1)}$ [9]. Thus, if $k \geq (\alpha n/200)^2$, then $2^n = 2^{\mathcal{O}(\sqrt{k})}$ and this algorithm is a subexponential time algorithm. If $k \leq (\alpha n/200)^2$, then we can show that the set L of vertices of G that are adjacent to more than $\alpha n/200$ edge of the *solution* is such that $|L| \leq \sqrt{k}$. Now by doing essentially the same argument as in the case $k \leq \alpha n/100$ we will be able to recover the position of every vertex x, except for a set $R \subseteq L$. To conclude, the algorithm then tries all the partitions of R. This part takes $|L|^{|L|} = 2^{\mathcal{O}(\sqrt{k}\log k)}$, resulting in the desired algorithm.

1.2.3 Derandomization

We first abstract out the main properties of Degree Estimator Lemma 7 that have been used in several applications in [7, 27, 29, 28, 33] and several other articles.

> Let U be a universe of size n and t be a constant. A random sample S of t elements of U has the following properties:
> **Property A.** For every subset A of the universe of $\Omega(n)$ elements, the probability that the sample S is a subset of A is constant;
> **Property B.** Conditioned on the sample S being a subset of A, we have that for every subset B of A of size $\Omega(n)$, $\frac{|S \cap B||A|}{t}$ is a good estimator of $|B|$ with probability close to 1.

These two properties of random samples have been successfully deployed to design randomized approximation schemes for a number of fundamental problems on dense structures [7, 27, 29, 28, 33]. Typically, algorithms based on this approach can be de-randomized by going over all possible subsets S of size t, and observing that at least one of them has the

[2] We assume here that $|A_i|$ is known. In fact, an approximation to the size will be enough for our purpose.

desired property. Unfortunately, this leads to an overhead of roughly n^t in the running time (which typically yields deterministic PTASes in place of randomized EPTASes). We present an efficient way to derandomize most of the algorithms based on the procedure. Our main derandomization tool is the following lemma.

▶ **Lemma 10** (⋆). *For any constants ϵ_1, ϵ_2 and ϵ_3 smaller than 1, and U a universe on n elements, there exists a set $\mathcal{T}$ of $\mathcal{O}(2^{100/(\epsilon_1^2\epsilon_2)}n)$ subsets of U, such that if A is a subset of at least $\epsilon_3 n$ elements of U and $\mathcal{K}$ a collection of subsets of A, then there exists a set $T \in \mathcal{T}$ such that the number of sets X of $\mathcal{K}$ such that $||T \cap X| - \frac{|T||X|}{|A|}| \geq \epsilon_1|T|$ is smaller than $\epsilon_2|\mathcal{K}|$. Moreover, the set $\mathcal{T}$ can be computed deterministically in $n^{\mathcal{O}(1)}$ time.*

Therefore, in all the proof using Lemma 7, we can replace the random sampling by trying all the elements of the family $\mathcal{T}$ provided by the Lemma 10. The proof involves using the known construction of pairwise (2-wise) independent permutations (see [4] for more details). The proof can also be done via expander random walk method (see Section 3.2 of [30]).

1.3 Related Works

Over the last two decade, the design of parameterized subexponential-time algorithms for problems on sparse graphs has been extremely fruitful. However, the same could not be said about research on dense graphs. The first problem on dense graphs shown to admit a parameterized subexponential-time algorithm is the Feedback Arc Set on Tournaments (FAST) problem [3]. The design of this algorithm exhibited a new method to develop parameterized algorithms called chromatic coding, which is now textbook material [15]. Subsequently, there appeared several other works on the design of parameterized subexponential-time algorithms for problems on tournaments, see e.g. [26, 22, 34]. Afterwards, dense classes of digraphs that are not tournaments have also been considered in the same context [45, 40]. Also, d-Correlation Clustering is known to admit a subexponential-time parameterized algorithm [24]. When d is not fixed, the problem is known not to admit a parameterized subexponential-time algorithm under the Exponential Time Hypothesis (ETH) [24].

2 Preliminaries

A *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed, finite alphabet. Let L be a parameterized problem. For an instance (x, k) of L, k is called the *parameter*. A *polynomial kernel* on L is an algorithm which, for any given instance (x, k) of L outputs, in polynomial time in the size of (x, k), an instance (x', k') of L with the following properties:

- (x', k') is a yes-instance $\iff$ (x, k) is a yes-instance.
- $|x'|, k' \leq h(k)$, where h is a polynomial function.

For further notions related to parameterized algorithm, we refer the reader to [15].

We follow the standard graph theory notations from [20]. Let $G = (V(G), E(G))$ be a graph and $x \in V(G)$. Then, $N(x)$ denotes the neighborhood of x, and $d(x) = |N(x)|$ its degree. If A is a subset of $V(G)$, then $d_A(x) = |N(x) \cap A|$ denotes the degree of x inside A. If A and B are two subsets of vertices in $V(G)$, then $E(A, B)$ denotes the set of edges with exactly one endpoint in A and one endpoint in B. A set of edges S is said to be a *d-cut* if $G - S$ has exactly d connected components.

A graph G is said to be *k-edge connected* (resp. *k-vertex connected*) if for any pair of vertices x and y in G, there exists k edge-disjoint (resp. vertex-disjoint) paths between x and y. For a graph G and two vertices x and y, a set of edges A is said to be an (x, y)-edge cut if $G - A$ does not contain any path between x and y. Likewise, a set of vertices S is said to be a (x, y)-vertex cut if $G - S$ does not contain any path between x and y. Let us cite the celebrated Menger's Theorem [44].

▶ **Theorem 11.** *Let G be a graph and x, y two vertices of G. The maximum number of vertex-disjoint (resp. edge-disjoint) paths between x and y is equal to the minimum size of a (x, y)-vertex cut (resp. (x, y)-edge cut).*

Let G be a graph and X a set of vertices, the graph obtained by *contracting X and keeping multiedges*, is the graph G' obtained from G by removing X, adding a new vertex x, and for every $v \in G$ such that v is adjacent to k vertices in X adding k multi-edges between x and v. Let U be a universe. Then, 2^U denotes all subsets of U and $\binom{U}{t}$ denotes all the subsets of size t of U. For an integer k, $[k]$ denotes the set $\{1, \dots, k\}$. For any real numbers a, b and c we write $a = b \pm c$ if $b - c \leq a \leq b + c$. The following easy observation will be used throughout the paper.

▶ **Observation 12.** *If c is a real in $[0, 1/2]$ and $x = 1 \pm c$, then $\frac{1}{x} = (1 \pm 2c)$.*

To construct estimators deterministically, we rely on the well known notion of k-wise independence, in the particular setting of permutations.

▶ **Definition 13.** *Let $n, k \in \mathbb{N}$. A family $\mathcal{S}$ of permutations of $\mathcal{S}_n$ is k*-wise independent *if, for any k-tuple of distinct elements $(x_1, \dots, x_k)$, the distribution $(f(x_1), f(x_2), \dots, f(x_k))$ where $f \in \mathcal{S}$ is chosen uniformly at random and the distribution $(f'(x_1), f'(x_2), \dots, f'(x_k))$ where $f' \in \mathcal{S}_n$ is chosen uniformly at random, are such that*

$$\sum_{(a_1,\dots,a_k)\in[n]^k} |Pr(f(x_1), \dots, f(x_k) = (a_1, \dots, a_k)) - Pr(f'(x_1), \dots, f'(x_k) = (a_1, \dots, a_k))| = 0.$$

Efficient construction of a k-wise independent family of permutations are known for $k = 2$ and $k = 3$ but open for $k > 4$ (see [4] for more details). In particular, there exists for every n, a family $\mathcal{S}(n)$ of $\mathcal{O}(n)$ pairwise (2-wise) independent permutations. This family will be sufficient for our derandomization purposes.

Throughout this paper, we will make an extensive use of Chebyshev's inequality:

▶ **Proposition 14.** *Let X be a random variable with expected value μ and variance σ^2. Then for any real number $k > 0$, $Pr[|X - \mu| \geq k\sigma] \leq \frac{1}{k^2}$.*

3 Edge-disjoint paths in everywhere dense graphs

In this section we design a linear vertex kernel for Edge-Disjoint Paths on everywhere α-dense graphs. We first present a polynomial time algorithm for the Edge-Disjoint Paths problem in α-dense graphs, when the number of demands is small (but still linear) compared to αn. Towards this, we start-by showing that highly edge-connected parts will always contain a solution to an Edge-Disjoint Paths instance.

▶ **Lemma 15.** *Let c be a constant between 0 and 1, and G be a graph on n vertices with minimum degree at least cn. For any pair of vertices x, y of G, if there exists a path between x and y, then there exists a path of length at most $4/c$.*

Proof. Let P be a shortest path between x and y. If there exists a vertex $u \in G$ such that u is adjacent to 4 vertices of P, then two of these vertices will be at distance at least 3 in the path. Denoting x_1 and x_2 these vertices, replacing the subpath of P between x_1 and x_2

by the path x_1ux_2 gives a path between x and y shorter than P, which is a contradiction. Therefore, the sum of the degree of the vertices of P is smaller than $4n$ and thus $|P|cn \leq 4n$ which implies $|P| \leq \frac{4}{c}$. ◀

▶ **Lemma 16.** *Let G be a graph with minimum degree αn, and cn edge-connected for some constant $c \leq \alpha/2$. Any instance of* Edge-Disjoint Paths *with $k \leq \frac{\alpha c n}{8}$ has a solution. Moreover, this solution can be found in polynomial time.*

Proof. Let $(G, (s_1, t_1), \cdots, (s_k, t_k))$ be an instance of the Edge-Disjoint Paths problem. For every pair (s_i, t_i), since G is cn-edge connected, there exists cn edge-disjoint paths $P_1, \ldots, P_{cn}$ between s_i and t_i. Moreover, we can assume that all these paths are shorter than $\frac{8}{\alpha}$. Indeed, removing the edges of all but one path P_j leaves G with minimum degree at least $\alpha n - cn \geq \frac{\alpha n}{2}$ and Lemma 15 implies that P_j can actually be taken shorter than $\frac{8}{\alpha}$. This means that we can select a solution for the Edge-Disjoint Paths problem greedily using these paths. Indeed, each path is of length smaller than $\frac{8}{\alpha}$, so the path selected between s_i and t_i intersects at most $\frac{8}{\alpha}$ of the paths between s_j and t_j. Since $k \leq \frac{\alpha c n}{8}$, there is always one path available between s_i and t_i. ◀

For the proof of Lemma 16, we could have used a previously known result [31]. However, we still give the proof here, as it is simple on dense graphs, and helps in a complete understanding of the algorithm. The next lemma is an essential part of the proof. The goal is to find a partition of the vertices of $V(G)$ into a bounded number of parts, such that each part induces a graph with large edge-connectivity.

▶ **Lemma 3.** *For any real α between 0 and 1, there exists a constant $c \leq \alpha/2$ such that, if G is a graph on n vertices and minimum degree αn, then there exists a partition $\mathcal{P}$ of the vertices $V(G)$ into $g \leq \frac{2}{\alpha}$ subsets $V_1, \cdots, V_g$ such that for all $i \in [g]$:*
- *$G[V_i]$ is cn edge-connected.*
- *$G[V_i]$ has minimum degree $\frac{\alpha n}{2}$.*

Moreover, such a partition can be found in polynomial time.

Proof. Let t be an integer such that $\frac{\alpha}{(1-\alpha/3)^t} > 2/3$, and c be a sufficiently small constant such that $tc < \alpha/6$, $\alpha/2 \geq c$ and for all $i < t$:

$$cn < \frac{\alpha^2 n}{(1-\alpha/3)^{i-1}} \left(\frac{1}{1-\alpha/2} - \frac{1}{1-\alpha/3} \right)$$

We inductively build a sequence of partitions of $V(G)$: $\mathcal{P}_1, \ldots, \mathcal{P}_t$. Each $\mathcal{P}_{i+1}$ is obtained from $\mathcal{P}_i$ by applying a set of operations. Further, either a part of $\mathcal{P}_i$ remains a part in $\mathcal{P}_{i+1}$ or breaks into several parts in $\mathcal{P}_{i+1}$. In particular, $\mathcal{P}_{i+1}$ is a *finer* partition than $\mathcal{P}_i$. Let each $\mathcal{P}_i$ consists of $V_1^i, \cdots, V_{l_i}^i$ as its parts. Throughout the proof these parts satisfy the following invariants. That is, for all $j \in [l_i]$:

Invariant 1: $G[V_j^i]$ has minimum degree $(\alpha - ci)n$.

Invariant 2: Either $G[V_j^i]$ is cn edge-connected; or every vertex of $v \in V_j^i$ has more than $\frac{\alpha}{(1-\alpha/3)^{i-1}}|V_j^i|$ neighbours in $G[V_j^i]$ (note that, $\frac{\alpha}{(1-\alpha/3)^{i-1}} \geq \alpha$ and thus, $G[V_j^i]$ is denser than G).

Note that, as we chose t such that $\frac{\alpha}{(1-\alpha/3)^t} > 2/3$, and c such that $tc < \alpha/2$, if the previous properties are satisfied, then $\mathcal{P}_t$ is the partition that we are looking for. Indeed, the second condition tells us that, if $G[V_j^t]$ is not cn-edge connected, then every vertex of V_j^t has more than $2/3|V_j^t|$ neighbors in $G[V_j^t]$. Since $|V_j^t| \geq (\alpha - ct)n \geq \alpha n/2$, it means that any pair of vertices in V_j^t have more than $\alpha n/6$ common neighbors in V_j^t, which implies that $G[V_j^t]$ cn-edge connected. Moreover, since $|V_j^t| \geq \alpha n/2$, this partition has less than $\frac{2}{\alpha}$ parts.

What remains to show is that indeed there exists a sequence of partitions of $V(G)$: $\mathcal{P}_1, \ldots, \mathcal{P}_t$. We show the existence of the partition $\mathcal{P}_i$ by induction on i, setting $\mathcal{P}_1 = V(G)$ which trivially satisfies all the properties. Suppose now that we have constructed the partition $\mathcal{P}_i = V_1^1, \cdots, V_{l_i}^i$ for some $i < t$. For each $j \in l_i$, we define a partition of V_j^i into $H_j^1, \ldots, H_j^{x_j}$ for some $x_j < (2/\alpha)$ as follows: If $G[V_j^i]$ is cn-edge connected, then $x_j = 1$ and $H_j^1 = V_j^i$. If not, let $H_j^1, \ldots, H_j^{x_j}$ be the connected components of $G[V_j^i]$ after removing the edges of a cut of size smaller than cn. Note that every vertex has degree at least $(\alpha - ci)n - cn \geq \frac{\alpha n}{2}$ after removing the cut edges, which implies Invariant 1. This means that the size of each component is at least $\frac{\alpha n}{2}$. This means in particular that the number of components is smaller than $(2/\alpha)$. Moreover, let w be a vertex in one of the connected components, H_j^r, we know that the degree of w in $G[V_j^i]$ is greater than $\frac{\alpha}{(1-\alpha/3)^{i-1}}|V_j^i|$. Since the cut is of size cn, it means that the degree of w in $G[H_j^r]$ is greater than $\frac{\alpha}{(1-\alpha/3)^{i-1}}|V_j^i| - cn$. Since, there is at least *one other component*, we have that $|H_j^r| < |V_j^i| - \frac{\alpha n}{2} < (1 - \frac{\alpha}{2})|V_j^i|$. This means that the degree of w in $G[H_j^r]$ is greater than $\frac{\alpha}{(1-\alpha/3)^{i-1}}(\frac{1}{1-\alpha/2}|H_j^r|) - cn$, which by the choice of c is greater than $\frac{\alpha}{(1-\alpha/3)^{i}}|H_j^r|$. Finally, we take $\mathcal{P}_{i+1}$ as the union of all the H_j^r for all $j \in [l_i]$ and $r \in [x_j]$. That is, $\mathcal{P}_{i+1}$ consists of either a part from $\mathcal{P}_i$, or connected components of a part that has a cut of size smaller than cn. By the above description, it follows that $\mathcal{P}_i$ satisfies both the invariants. This completes the proof. ◀

▶ **Lemma 17.** *The* Edge-Disjoint Paths *problem can be solved in time $k^\rho n^{\mathcal{O}(1)}$ on everywhere α-dense graphs, when $k \leq \frac{\alpha c n}{16}$. Here, c is the constant defined in Lemma 3 and $\rho = 2^{\frac{2}{\alpha}}\frac{2}{\alpha}!$.*

Proof. Let $(G, (s_1, t_1), \ldots, (s_k, t_k))$ be an instance of the Edge-Disjoint Paths problem in an everywhere α-dense graph G of size n, where $k \leq \frac{\alpha c n}{8}$. Let $\mathcal{P} = V_1, \ldots, V_g$, $g \leq \frac{2}{\alpha}$, be the partition of $V(G)$ obtained by applying Lemma 3.

▷ Claim 18. If $(G, (s_1, t_1), \ldots, (s_k, t_k))$ is a yes-instance of Edge-Disjoint Paths, then there exists a path system $\tilde{P}_1, \ldots, \tilde{P}_k$, connecting s_i to t_i such that the intersection of any path $\tilde{P}_j$ with any V_i for $i \in [g]$ is a subpath (possibly empty) of $\tilde{P}_j$.

Proof. Let $(P_1, \ldots, P_k)$ be a solution. For every $j \in [g]$, we say that $(P_1, \ldots, P_k)$ satisfies the property $\mathcal{H}_j$ if $P_i \cap V_j$ is a subpath of P_i for every $i \in [k]$.

Suppose that the solution $(P_1, \ldots, P_k)$ does not satisfy property $\mathcal{H}_j$. For every $i \in [k]$ denote by h_i and l_i, the first and the last vertex of P_i in V_j, respectively. If P_i does not intersect V_j, then we assign h_i and l_i to $\emptyset$. Furthermore, h_i could be equal to l_i. Observe that $(G[V_j], (h_1, l_1), \ldots, (h_k, l_k))$ is an instance of Edge-Disjoint Paths with $k \leq \frac{\alpha c n}{16}$. By Lemma 16, there is a solution $(P_1', \ldots, P_k')$ to this problem in $G[V_j]$. Let $(P_1^1, \ldots, P_k^1)$ denote the solution obtained from $(P_1, \ldots, P_k)$ by replacing each subpath of P_i from h_i to l_i by P_i'.

Clearly the solution $(P_1^1, \ldots, P_k^1)$ satisfies property $\mathcal{H}_j$. Moreover, let us show that if $(P_1, \ldots, P_k)$ satisfies property $\mathcal{H}_{j'}$ for some $j' \in [g]$ $j \neq j'$, then so does $(P_1^1, \ldots, P_k^1)$. This would conclude our proof of the lemma, as it means we can apply the previous procedure for every $j \in [g]$, iteratively.

Let i be an index of $[k]$ and suppose that $P_i \cap V_{j'}$ is a subpath of P_i. We want to show that $P_i^1 \cap V_{j'}$ is also a subpath of P_i^1. If $P_i \cap V_{j'}$ is empty, then so is $P_i^1 \cap V_{j'}$ as the vertices of $P_i^1 \setminus P_i$ belong to V_j and $j \neq j'$. Suppose now that $P_i^1 \cap V_{j'}$ is a subpath and denote by a_i and b_i the first and the last vertex of this path. Remember that h_i and l_i denote the first and the last vertex of $P_i \cap V_j$. If $P_i \cap V_j$ is empty, then $P_i^1 = P_i$ and there is nothing to prove, so let us assume it is not. Since the subpath of P_i between a_i and b_i is in $V_{j'}$ it means that h_i and l_i do not belong to this subpath. Therefore we are in one of the following three cases.

- h_i and l_i appear before a_i on P_i
- h_i and l_i appear after b_i on P_i
- h_i appears before a_i on P_i and l_i after b_i

In the first two cases, $P_i^1 \cap V_{j'} = P_i \cap V_{j'}$, which is still a subpath of P_i^1. In the last case, $P_i^1 \cap V_{j'}$ becomes empty. This concludes the proof. ◁

Consider the graph G' obtained from G by contracting every part V_j of the partition $\mathcal{P}$ into one vertex v_j (keeping multi-edges). That is, although the number of vertices in G' is g, the number of parallel edges between v_i and v_j is same as the number of edges between V_i and V_j. Thus, there is a one-to-one correspondence between edges in G' and the edges between a pair of vertices $w_1 \in V_i$ and $w_2 \in V_j$ such that $i \neq j$. For every $i \in [k]$, let s'_i (resp. t'_i) denote the vertex of G' corresponding to the part containing s_i (resp. t_i) in G. Notice that same pair of v_i and v_j could be assigned to several pairs of s_i and t_i. In fact, if both s_i and t_i belong to the same part, say V_j, then $s'_i = v_j$ and $t'_i = v_j$. In this case it just means that the path must be completely contained inside the graph $G[V_j]$.

▷ Claim 19. $(G, (s_1, t_1), \ldots, (s_k, t_k))$ is a yes-instance of Edge-Disjoint Paths if and only if $(G', (s'_1, t'_1), \ldots, (s'_k, t'_k))$ is a yes-instance of Edge-Disjoint Paths.

Proof. Forward direction follows from Claim 18. Indeed, as explained before, if there is a solution in G, then we can assume that this solution is such that the intersection of any path with any part V_j is a subpath. Therefore, contracting the V_i along these paths create paths in G' and these paths are a solution to the problem in G'. Suppose now that we have a solution $P'_1, \ldots, P'_k$ to the Edge-Disjoint Paths problem in G'. For every i, let $u^i_1, \ldots, u^i_{r_i}$ denote the sequence of edge in P'_i. Note that each of these edge corresponds to a specific edge in G. For every $j \in [r_1]$ such that v_j is an inner vertex of P'_i, let us define $a^i_j \in V(G)$ and $b^i_j \in V(G)$ as the extremities of the two edges among $u^i_1, \ldots, u^i_{r_i}$ which are incident to v_j. For the first vertex v_s of P'_i, we define similarly a^i_s as s_i and b^i_s is the extremity of the only edge of P'_i adjacent to v_j. Likewise, we can define $a^i_t \in V(G)$ and $b^i_t \in V(G)$ for the last vertex v_t of the path. Overall, replacing each v_j by a path from a^i_j to b^i_j gives a path from s_i to t_i in G. However, for every $j \in [g]$, $(G[V_j], (a^i_1, b^i_1), \ldots, (a^i_k, b^i_k))$ defines an instance of Edge-Disjoint Paths. Since $G[V_j]$ satisfies the properties of Lemma 16, in polynomial time we can find a solution to our instance. For every $i \in [k]$ and $j \in [g]$, let Q^i_j denote the path from a^i_j to b^i_j in this solution. Finally, for each $i \in [k]$, let P_i denote the path obtained from P'_1 by replacing each v_j by Q^i_j. Thus, $P_1, \ldots, P_k$ forms a solution to the instance $(G, (s_1, t_1), \ldots, (s_k, t_k))$, which in particular implies that such a solution exists. ◁

Claim 19 shows that it is enough to solve our problem on the instance $(G', (s'_1, t'_1), \ldots, (s'_k, t'_k))$. Let us now explain how to solve this problem in G'. Recall that G' is a graph on a finite (at most $\frac{2}{\alpha}$) number of vertices. In particular it means that there is at most $2^{\frac{2}{\alpha}} \frac{2}{\alpha}!$ different paths in G', where a path may appear multiple times[3]. (First, choose the subset of vertices that appear in the path and then guess the permutation of the chosen vertices). Thus, the number of paths is upper bounded by $\rho = 2^{\frac{2}{\alpha}} \frac{2}{\alpha}!$. Therefore, a solution to this problem consists of assigning to each of these paths an integer of value at most k, which denotes the number of requests that will be resolved using this path. It means that the number of possible "distributions" of the requests among these paths is upper bounded by k^ρ. Moreover, once we have chosen the distribution of the requests among these paths, then testing whether this

[3] Here we see a path as a sequence of vertices.

distribution is indeed a solution requires only to count the number of times each multi-edge is used. So in total, to find a solution to the problem in G', we only need to check the $\mathcal{O}(k^\rho)$ possible distributions. Since, we can test each distribution in $n^{\mathcal{O}(1)}$ time, the running time of the algorithm follows. ◀

Lemma 17 implies the following result.

▶ **Theorem 2.** *Edge-Disjoint Paths admits an $\mathcal{O}(k)$ vertex kernel on everywhere α-dense graphs.*

Proof. Let $(G,(s_1,t_1),\ldots,(s_k,t_k))$ be an instance of Edge-Disjoint Paths. Further, let c be the constant defined in Lemma 3. If $k \leq \frac{\alpha c n}{16}$, then we apply Lemma 17 and solve the problem in time $k^{\mathcal{O}(\frac{2}{\alpha}!)}n^{\mathcal{O}(1)}$. Based on the answer of Lemma 17, we either return a solution or a trivial no-instance of the problem. However, now we have that $k \geq \frac{\alpha c n}{16}$, and hence $n \leq \frac{16k}{\alpha c} = \mathcal{O}(k)$. This concludes the proof. ◀

4 Conclusion

Inspired by the success of designing of PTASes and EPTASes for computationally intractable problems on everywhere dense graphs (every vertex has minimum degree at least αn, for some fixed constant $\alpha > 0$), in this paper we undertook a study for computationally intractable problems on dense graphs in the realm of Parameterized Complexity on dense graphs. We obtained linear kernels for Edge-Disjoint Paths, Edge Odd Cycle Transversal, Minimum Bisection, d-Way Cut, Multiway Cut and Multicut on everywhere dense graphs. Additionally, we obtained a cubic kernel for Vertex-Disjoint Paths on everywhere dense graphs. In addition to kernelization results, we obtained subexponential-time parameterized algorithms for Edge Odd Cycle Transversal, Minimum Bisection, and d-Way Cut. Finally, we showed how all of our results (as well as EPASes for these problems) can be de-randomized. Studying other NP-hard problems on dense graphs is an interesting research avenue. We conclude our paper with some concrete open problems.

1. Does Vertex-Disjoint Paths admit a linear vertex kernel on everywhere α-dense graphs?
2. Does Edge-Disjoint Paths and Vertex-Disjoint Paths admit an algorithm with running time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ on everywhere α-dense graphs?

References

1 Noga Alon, Wenceslas Fernandez de la Vega, Ravi Kannan, and Marek Karpinski. Random sampling and approximation of max-csps. *J. Comput. Syst. Sci.*, 67(2):212–243, 2003. doi:10.1016/S0022-0000(03)00008-4.

2 Noga Alon, Alan M. Frieze, and Dominic Welsh. Polynomial time randomized approximation schemes for tutte-gröthendieck invariants: The dense case. *Random Struct. Algorithms*, 6(4):459–478, 1995. doi:10.1002/rsa.3240060409.

3 Noga Alon, Daniel Lokshtanov, and Saket Saurabh. Fast FAST. In *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, pages 49–58, 2009.

4 Noga Alon and Shachar Lovett. Almost k-wise vs. k-wise independent permutations, and uniformity for general group actions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 350–361, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

5 Gunnar Andersson and Lars Engebretsen. Property testers for dense constraint satisfaction programs on finite domains. *Random Struct. Algorithms*, 21(1):14–32, 2002. doi:10.1002/rsa.10041.

6 Sanjeev Arora, Alan M. Frieze, and Haim Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. *Math. Program.*, 92(1):1–36, 2002. doi:10.1007/s101070100271.

7 Sanjeev Arora, David R. Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. *J. Comput. Syst. Sci.*, 58(1):193–210, 1999. doi:10.1006/jcss.1998.1605.

8 Cristina Bazgan, Wenceslas Fernandez de la Vega, and Marek Karpinski. Polynomial time approximation schemes for dense instances of minimum constraint satisfaction. *Random Struct. Algorithms*, 23(1):73–91, 2003. doi:10.1002/rsa.10072.

9 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009.

10 Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.

11 Nicolas Bousquet, Jean Daligault, and Stéphan Thomassé. Multicut is FPT. *SIAM J. Comput.*, 47(1):166–207, 2018.

12 Yixin Cao, Jianer Chen, and Jia-Hao Fan. An $O(1.84^k)$ parameterized algorithm for the multiterminal cut problem. *Inf. Process. Lett.*, 114(4):167–173, 2014.

13 Jianer Chen, Yang Liu, and Songjian Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.

14 Rajesh Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michal Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. *SIAM J. Comput.*, 45(4):1171–1229, 2016.

15 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms.* Springer, 2015.

16 Marek Cygan, Pawel Komosa, Daniel Lokshtanov, Michal Pilipczuk, Marcin Pilipczuk, and Saket Saurabh. Randomized contractions meet lean decompositions. *CoRR*, abs/1810.06864, 2018. arXiv:1810.06864.

17 Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Magnus Wahlström. Clique cover and graph separation: New incompressibility results. *ACM Trans. Comput. Theory*, 6(2):6:1–6:19, 2014.

18 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Minimum bisection is fixed-parameter tractable. *SIAM J. Comput.*, 48(2):417–450, 2019.

19 Wenceslas Fernandez de la Vega and Marek Karpinski. Polynomial time approximation of dense weighted instances of MAX-CUT. *Random Struct. Algorithms*, 16(4):314–332, 2000.

20 Reinhard Diestel. *Graph theory.* Springer Publishing Company, Incorporated, 2018.

21 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

22 Uriel Feige. Faster fast(feedback arc set in tournaments). *CoRR*, abs/0911.5094, 2009. arXiv:0911.5094.

23 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

24 Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Yngve Villanger. Tight bounds for parameterized complexity of cluster editing with a small number of clusters. *J. Comput. Syst. Sci.*, 80(7):1430–1447, 2014. doi:10.1016/j.jcss.2014.04.015.

25 Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization. Theory of parameterized preprocessing.* Cambridge University Press, Cambridge, 2019.

26 Fedor V. Fomin and Michal Pilipczuk. Subexponential parameterized algorithm for computing the cutwidth of a semi-complete digraph. In *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, pages 505–516, 2013.

27 Alan M. Frieze and Ravi Kannan. The regularity lemma and approximation schemes for dense problems. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 12–20. IEEE Computer Society, 1996. doi:10.1109/SFCS.1996.548459.

28 Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(13):249–266, 2006. doi:10.4086/toc.2006.v002a013.

29 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

30 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

31 Andreas Huck. A sufficient condition for graphs to be weaklyk-linked. *Graphs and Combinatorics*, 7(4):323–351, 1991.

32 Marek Karpinski. Polynomial time approximation schemes for some dense instances of np-hard optimization problems. *Algorithmica*, 30(3):386–397, 2001. doi:10.1007/s00453-001-0012-z.

33 Marek Karpinski and Warren Schudy. Linear time approximation schemes for the gale-berlekamp game and related minimization problems. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 313–322. ACM, 2009. doi:10.1145/1536414.1536458.

34 Marek Karpinski and Warren Schudy. Faster algorithms for feedback arc set tournament, kemeny rank aggregation and betweenness tournament. In *Algorithms and Computation - 21st International Symposium, ISAAC2010, Jeju Island, Korea, December 15-17, 2010, Proceedings, Part I*, pages 3–14, 2010.

35 Ken-ichi Kawarabayashi and Mikkel Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 160–169. IEEE Computer Society, 2011.

36 Ken-ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: the unique linkage theorem. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 687–694. ACM, 2010.

37 Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Trans. Algorithms*, 10(4):20:1–20:15, 2014.

38 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020.

39 Daniel Lokshtanov, Pranabendu Misra, Michal Pilipczuk, Saket Saurabh, and Meirav Zehavi. An exponential time parameterized algorithm for planar disjoint paths. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1307–1316. ACM, 2020.

40 Jayakrishnan Madathil, Roohani Sharma, and Meirav Zehavi. A sub-exponential fpt algorithm and a polynomial kernel for minimum directed bisection on semicomplete digraphs. In *44th International Symposium on Mathematical Foundations of Computer Science MFCS 2009, Aachen, Germany, August 26-30, 2019*, 2019.

41 Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.

42 Dániel Marx and Igor Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. *SIAM J. Comput.*, 43(2):355–388, 2014.

43 Claire Mathieu and Warren Schudy. Yet another algorithm for dense max cut: go greedy. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 176–182. SIAM, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347102.

44 Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

45 Pranabendu Misra, Saket Saurabh, Roohani Sharma, and Meirav Zehavi. Sub-exponential time parameterized algorithms for graph layout problems on digraphs with bounded independence number. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, pages 35:1–35:19, 2018.

46 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.

47 Marcin Pilipczuk, Michal Pilipczuk, and Marcin Wrochna. Edge bipartization faster than 2^k. *Algorithmica*, 81(3):917–966, 2019.

48 Neil Robertson and Paul D. Seymour. Graph minors XIII. the disjoint paths problem. *J. Comb. Theory, Ser. B*, 63(1):65–110, 1995.

49 René van Bevern, Andreas Emil Feldmann, Manuel Sorge, and Ondrej Suchý. On the parameterized complexity of computing graph bisections. In Andreas Brandstädt, Klaus Jansen, and Rüdiger Reischuk, editors, *Graph-Theoretic Concepts in Computer Science - 39th International Workshop, WG 2013, Lübeck, Germany, June 19-21, 2013, Revised Papers*, volume 8165 of *Lecture Notes in Computer Science*, pages 76–87. Springer, 2013.

A Definition of the studied problems

We now define all the problems mentioned in the paper.

EDGE-DISJOINT PATHS **Parameter:** k
Input: A graph G and a set of request pairs $(s_1, t_1), \ldots, (s_k, t_k)$.
Question: Does there exist a set of paths $P_1, \ldots, P_k$, between s_i and t_i, such that they are pairwise edge disjoint?

VERTEX-DISJOINT PATHS **Parameter:** k
Input: A graph G and a set of request pairs $(s_1, t_1), \ldots, (s_k, t_k)$.
Question: Does there exist a set of paths $P_1, \ldots, P_k$, between s_i and t_i, such that they are pairwise vertex disjoint?

EDGE ODD CYCLE TRANSVERSAL **Parameter:** k
Input: A graph G and an integer k.
Question: Does there exists $S \subseteq E(G)$ of size at most k such that $G - S$ is bipartite?

MINIMUM BISECTION **Parameter:** k
Input: A graph G and an integer k.
Question: Does there exists a partition (A, B) of $V(G)$ such that $||A| - |B|| \leq 1$ and $E(A, B) \leq k$?

MULTIWAY CUT **Parameter:** k
Input: A graph G, a set $T \subseteq V(G)$ and an integer k.
Question: Does there exists a set $S \subseteq E(G)$ of size at most k such that every vertex of T lies in a different connected component of $G - S$?

Multicut **Parameter:** k
Input: A graph G, a set of pairs $(s_i, t_i)_{i=1}^{\ell}$ and an integer k.
Question: Does there exists $S \subseteq E(G)$ of size at most k such that for every $i \in [\ell]$, vertices s_i and t_i lie in different connected components of $G - S$?

d-Way Cut **Parameter:** k
Input: A graph G and an integer k.
Question: Does there exists a set $S \subseteq E(G)$ of size at most k such that $G - S$ has at least d connected components?