Exploiting Dense Structures in Parameterized Complexity

William Lochet* Daniel Lokshtanov† Saket Saurabh‡ Meirav Zehavi§

October 4, 2020

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 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.

*University of Bergen, Norway. william.lochet@uib.no

†University of California Santa Barbara, USA. daniello@ucsb.edu

‡Institute of Mathematical Science, HBNI, India, and University of Bergen, Norway. saket@imsc.res.in

§Ben-Gurion University, Israel. meiravze@bgu.ac.il

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 [22, 16, 24, 47, 26].

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 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 1.1 ([7, 34])**.** *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 [28] and, independently, Goldreich, Goldwasser, and Ron [30] 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, [28, 30] made PTASes of [7] into EPTASes. Frieze and Kannan [28] also pointed out connections to constructive versions of Szemeredi's Regularity Lemma and Goldreich, Goldwasser, and Ron [30] 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 [29] 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 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 [29], most of the earlier works largely focused on maximization problems. In 2009, Karpinski and Schudy [34] 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 [44, 20, 33, 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 [49]. However, the $f(k)$ in the running time in the algorithm of Robertson and Seymour [49] and its later improvement is at least triply exponential [37]. Only recently an algorithm with $f(k) = 2^{k^{\mathcal{O}(1)}}$ are designed when the input is restricted to planar graphs [40]. 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. In particular we get the following result about EDGE-DISJOINT PATHS.

Theorem 1.1. EDGE-DISJOINT PATHS *admits an $\mathcal{O}(k)$ vertex kernel on everywhere α-dense graphs.*

Proof of Theorem 1.1 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 path. 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 partition the input graph into small number of parts such that each part has minimum degree and edge-connectivity linear in n. In particular we get the following.

Lemma 1.1. *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.

To obtain the desired partition $\mathcal{P}$. We inductively build a sequence of partitions of $V(G)$: $\mathcal{P}_1, \dots, \mathcal{P}_t$. 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 *coarser* partition than $\mathcal{P}_i$. Let each $\mathcal{P}_i$ consists of $V^i_1, \cdots, V^i_{l_i}$ as its parts. Throughout the process, we maintain that each part has the required minimum degree. The only thing we try to fix is the edge-connectivity. If V_j^i satisfies the edge-connectivity property, then we include as a part in $\mathcal{P}_{i+1}$. Else, $G[V_j^i]$ has a cut of size smaller than cn, we delete the edges of this cut and include all the connected components of $G[V_j^i]$ as parts. Indeed, we show that each of these parts have the desired minimum degree. Essentially, what happens is that the way we select c compared to α, at each iteration "the graph induced on parts" become denser than before. Eventually, each part has the property that for every pair of vertices there are more than cn common neighbors, leading to high edge-connectivity. Finally, we show that this process stops after constantly many steps, resulting in the desired partition.

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 $G[V_j]$. Using the properties of the parts we can prove the following claim.

Claim 1.1.1. $(G, (s_1, t_1), \ldots, (s_k, t_k))$ *is an yes-instance of* Edge-Disjoint Paths *if and only if* $(G', (s'_1, t'_1), \ldots, (s'_k, t'_k))$ *is an yes-instance of* Edge-Disjoint Paths*.*

Claim 1.1.1 reduces our problem to 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 $\rho = 2^{\frac{2}{\alpha}} \frac{2}{\alpha}!$ different paths in G', where a path may appear multiple times. Here, we see a path as a sequence of vertices. 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 "distribution" 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 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.

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 1.2. 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 1.2 is in adapting the proof of Lemma 1.1 for Vertex-Disjoint Paths. The main reason being that for Vertex-Disjoint Paths we need to simulate Lemma 1.1 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 1.1. 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. In particular, we prove the following decomposition lemma.

Lemma 1.2. *For any two reals α_1 and α_2, between 0 and 1, there exists a constant $c \leq \alpha_2/6$ such that, if G is a graph on n vertices and minimum degree $\alpha_1 n$, then there exists a partition $\mathcal{P}$ of the vertices $V(G)$ into $g \leq \frac{2}{\alpha_1}$ subsets $V_0, \cdots, V_g$ with the following properties:*

- *For all $1 \leq i < j \leq g$, $E(V_i, V_j) = \emptyset$.*
- *For all $1 \leq i \leq g$, $G[V_i]$ is cn vertex-connected.*
- *$|V_0| \leq \alpha_2 n$.*

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

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. We leave the details to the corresponding section.

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 [19, 17, 15, 48, 42, 43, 13, 14, 36, 12] 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. Further, Edge Odd Cycle Transversal admits a randomized polynomial kernel on general graphs [38, 39]; on the other hand Minimum Bisection and Multicut are known not to admit a polynomial kernel [18, 50]. The kernelization complexity of Multiway Cut is still open. In this paper we obtain the following results about these problems on everywhere dense graphs.

Theorem 1.3. 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 1.4. 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 1.3 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 1.3 and 1.4) are similar to each other. Thus, to illustrate our methods we focus on giving intuition for the proof of d-Way Cut. The main ingredient of Theorems 1.3 and 1.4 is the following sampling primitive, a form of which has been extensively used in designing PTASes and EPTASes in everywhere α-dense graphs.

Lemma 1.3 (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 1.3 in order to guess the degree of the vertices in $V(G)$ 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 1.3 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 word, 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 1.3, 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 $|X| \leq \alpha/200$. 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[1]. Let us make our first guess of A_i: for every $i \in [d]$, let A'_i be the set 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 1.4.1. *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 $x \notin X_j$.

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

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 1.4.1 implies that $d_{A'_i}(x) \geq d_{A_i}(x) - |X|$, *even if* x *belongs to* X. However, since $d_{A_i}(x) \geq d(x) - \alpha/100n$, we have that $d_{A'_i}(x) \geq d(x) - \alpha/50n$. 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 1.4.2. *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, expect for a set $R \subseteq L$. To conclude, the algorithm then tries all the partition 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 1.3 that have been used in several applications in [7, 28, 30, 29, 34] 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, 28, 30, 29, 34]. 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 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 1.4. *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* $\mathcal{O}(2^{100/(\epsilon_1^2\epsilon_2)})$ *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 1.3, we can replace the random sampling by simply trying all the elements of the family $\mathcal{T}$ provided by the Lemma 1.4. 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 [31]).

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 [16]. Subsequently, there appeared several other works on the design of parameterized subexponential-time algorithms for problems on tournaments, see e.g. [27, 23, 35]. Afterwards, dense classes of digraphs that are not tournaments have also been considered in the same context [46, 41]. Also, d-CORRELATION CLUSTERING is known to admit a subexponential-time parameterized algorithm [25]. When d is not fixed, the problem is known not to admit a parameterized subexponential-time algorithm under the Exponential Time Hypothesis (ETH) [25].

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 [16].

We follow the standard graph theory notations from [21]. 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* (reps. *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[45].

Theorem 2.1. *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 $\{1,\ldots,k\}$. For any real numbers a,b and c we write $a = b \pm c$ if $b - c \le a \le b + c$. The following easy observation will be used throughout the paper.

Observation 2.1. *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 2.1. *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 $(x_1,\ldots,x_k)$, the distribution $(f(x_1), f(x_2),\ldots,f(x_k))$ where $f \in \mathcal{S}$ is chosen uniformly at random and the distribution $(f'(x_1), f'(x_2),\ldots,f'(x_k))$ where $f' \in \mathcal{S}_n$ is chosen uniformly at random, are such that*

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

Efficient construction of a k-wise independent family of permutation 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 2.1. *Let X be a random variable with expected value μ and variance σ^2. Then for any real number $k > 0$, $Pr[|X - \mu| \ge k\sigma] \le \frac{1}{k^2}$.*

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 3.1. *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 $|P|cn \le 4n$ which implies $|P| \le \frac{c}{4}$. ☐

Lemma 3.2. *Let G be a graph with minimum degree αn, and cn edge-connected for some constant $c \le \alpha/2$. Any instance of* Edge-Disjoint Paths *with $k \le \frac{\alpha cn}{8}$ 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 3.1 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 $k \leq \frac{\alpha n}{8}$, there is always one path available between s_i and t_i. □

For the proof of Lemma 3.2, we could have used a previously known result [32]. 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 3.3. *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 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 *coarser* 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 3.4. *The* Edge-Disjoint Paths *problem can be solved in time $k^\rho n^{\mathcal{O}(1)}$ on everywhere α-dense graphs, when $k \leq \frac{\alpha cn}{16}$. Here, c is the constant defined in Lemma 3.3 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 $k \leq \frac{\alpha cn}{8}$. Let $\mathcal{P} = V_1,\ldots,V_g$, $g \leq \frac{2}{\alpha}$, be the partition of $V(G)$ obtained by applying Lemma 3.3.

Claim 3.0.1. *If $(G,(s_1,t_1),\ldots,(s_k,t_k))$ is an 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 $k \leq \frac{\alpha cn}{16}$. By Lemma 3.2, 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 3.0.2. $(G, (s_1, t_1), \ldots, (s_k, t_k))$ *is an yes-instance of* Edge-Disjoint Paths *if and only if* $(G', (s'_1, t'_1), \ldots, (s'_k, t'_k))$ *is an yes-instance of* Edge-Disjoint Paths.

Proof. Forward direction follows from Claim 3.0.1. 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 3.2, 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. $\square$

Claim 3.0.2 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[2]. (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 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. $\square$

Lemma 3.4 implies the following result.

Theorem 3.1. Edge-Disjoint Paths *admits a linear 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 3.3. If $k \leq \frac{\alpha cn}{16}$, then we apply Lemma 3.4 and solve the problem in time $k^{\mathcal{O}(\frac{2}{\alpha}!)} n^{\mathcal{O}(1)}$. Based on the answer of Lemma 3.4, we either return a solution or a trivial

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

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. $\square$

4 Vertex-disjoint paths on everywhere dense graphs

In this section we give a polynomial kernel for Vertex-Disjoint Paths on everywhere α-dense graphs. It would have been nice to adapt the arguments for Edge-Disjoint Paths for Vertex-Disjoint Paths. Unfortunately, we are not able to design a linear kernel for the Vertex-Disjoint Paths problems, using the tools developed for Edge-Disjoint Paths. We are however able to design a cubic kernel for the Vertex-Disjoint Paths problem, with a proof structure similar to the one used for the Edge-Disjoint Paths problem.

4.1 Decomposing the graph: A vertex partitioning

In this section we prove an analogous of Lemma 3.3 for the Vertex-Disjoint Paths problem. The main technical difficulty in adapting the proof of Lemma 3.3 for Vertex-Disjoint Paths lies in the fact that we need to simulate Lemma 3.3 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 was a critical component in the proof of Lemma 3.3. 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. In particular, we prove the decomposition lemma.

Lemma 4.1. *For any two reals α_1 and α_2, between 0 and 1, there exists a constant $c \leq \alpha_2/6$ such that, if G is a graph on n vertices and minimum degree $\alpha_1 n$, then there exists a partition of the vertices $V(G)$ into $g \leq \frac{2}{\alpha_1}$ subsets $V_0, \cdots, V_g$ with the following properties:*

- *For all $i, j \in [g]$, with $i \neq j$, $E(V_i, V_j) = \emptyset$.*
- *For all $i \in [g]$, $G[V_i]$ is cn vertex-connected.*
- *$|V_0| \leq \alpha_2 n$.*

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

Proof. Let t be an integer such that $\frac{\alpha_1}{(1-\alpha_1/3)^t} > 2/3$, and c be a sufficiently small constant such that $tc < \alpha_1/2$, $\alpha_1/6 \geq c$, $t\frac{2c}{\alpha_1} \leq \alpha_2$ and for all $i < t$:

$$cn < \frac{\alpha_1^2 n}{(1-\alpha_1/3)^{i-1}} \left(\frac{1}{1-\alpha_1/2} - \frac{1}{1-\alpha_1/3} \right).$$

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 *coarser* partition than $\mathcal{P}_i$. Let each $\mathcal{P}_i$ consists of $V_0^i, \cdots, V_{l_i}^i$ as its parts. Throughout the proof these parts satisfy the following invariants. That is, for all $j, j' \in [l_i]$:

Invariant 1: $E(V_j^i, V_{j'}^i)$ is empty.

Invariant 2: $G[V_j^i]$ has minimum degree $(\alpha_1 - ci)n$.

Invariant 3: Either $G[V_j^i]$ is cn vertex-connected; or every vertex of $v \in V_j^i$ has more than $\frac{\alpha_1}{(1-\alpha_1/3)^{i-1}}|V_j^i|$ neighbours in $G[V_j^i]$ (note that, $\frac{\alpha_1}{(1-\alpha_1/3)^{i-1}} \geq \alpha_1$ and thus, $G[V_j^i]$ is denser than G).

Invariant 4: $|V_0^j| \le j\frac{2cn}{\alpha_1}$.

Note that, as we chose t such that $\frac{\alpha_1}{(1-\alpha_1/3)^t} > 2/3$, c such that $tc < \alpha_1/2$ and $t\frac{2c}{\alpha_1} \le \alpha_2$, if the previous properties are satisfied, then $\mathcal{P}_t$ is the partition that we are looking for. Indeed, the second condition tells that, if $G[V_j^t]$ is not cn vertex-connected, then every vertex of V_j^t has more than $2|V_j^t|/3$ neighbors in $G[V_j^t]$. Since $|V_j^t| \ge (\alpha_1 - ct)n \ge \frac{\alpha_1 n}{2}$, it means that any pair of vertices have more than $\frac{\alpha_1 n}{6}$ neighbors in common in $G[V_j^t]$, which means the graph $G[V_j^t]$ is cn vertex-connected.

We will show the existence of the partition $\mathcal{P}_i$ by induction, setting $\mathcal{P}_1 = V(G)$ which trivially satisfies all the properties. Suppose now that we have constructed the partition $\mathcal{P}_i = V_0^i, \cdots, V_{l_i}^i$ for some $i < t$. For each $j \in l_i$, we define a partition of V_j^i into $H_j^0, \dots H_j^{x_j}$ for some $x_j < (2/\alpha_1)$ as follows: If $G[V_j^i]$ is cn connected, then $x_j = 1$ and $H_j^1 = V_j^i$. If not, let H_j^0 be a vertex cut of size smaller than cn and $H_j^1, \dots, H_j^{x_j}$ be the connected components of $G[V_j^i]$ after removing H_j^0. Note that since every vertex has degree at least $(\alpha_1 - ci)n - cn \ge \frac{\alpha_1 n}{2}$ after removing the cut, it means that the size of each component is at least $\frac{\alpha_1 n}{2}$. This means in particular that the number of components is smaller than $(2/\alpha_1)$. Moreover, let x be a vertex of one component H_j^r, we know that the degree of x in $G[V_j^i]$ is greater than $\frac{\alpha_1}{(1-\alpha_1/3)^{i-1}}|V_j^i|$. Since the cut is of size cn, it means that the degree of x in $G[H_j^r]$ is greater than $\frac{\alpha_1}{(1-\alpha_1/3)^{i-1}}|V_j^i| - cn$. Moreover, because there is at least one other component,

$$|H_j^r| < |V_j^i| - \frac{\alpha_1 n}{2} < \left(1 - \frac{\alpha_1}{2}\right)|V_j^i|.$$

This means that the degree of x in $G[H_j^r]$ is greater than $\frac{\alpha_1}{(1-\alpha_1/3)^{i-1}}(\frac{1}{1-\alpha_1/2}|H_j^r|) - cn$, which by the choice of c is greater than $\frac{\alpha_1}{(1-\alpha_1/3)^{i}}|H_j^r|$. Let V_0^{i+1} denote the union of V_i^0 as well as all the H_0^j for $j \in [l_i]$. Because each H_0^j is smaller than cn, we have that $|V_0^{i+1}| \le |V_0^{i+1}| + x_j cn \le (i+1)\frac{2cn}{\alpha_1}$. Let $\mathcal{P}_{i+1}$ be the partition of $V(G)$ consisting of V_0^{i+1} and all the H_j^r for all $j \in [l_i]$ and $r \in [x_j]$, it satisfies all the required properties. □

Let us now mention the following theorem due to Bollobs and Thomason [11] which will be useful in finding solutions to instances of Vertex-Disjoint Paths on graphs with high vertex-connectivity (for example $G[V_i]$, for $i \ge 1$, arising in Lemma 4.1).

Theorem 4.1 ([11]). *For every k, if G is a graph with vertex connectivity at least $22k$, then any instance of the* Vertex-Disjoint Paths *problem admits a solution in G.*

4.2 The cubic vertex kernel

Let $(G, (s_1, t_1), \dots, (s_k, t_k))$ be an instance of the Vertex-Disjoint Paths problem, such that G is an everywhere α-dense graph on n vertices. Let $H = \cup_{i\in[k]}\{s_i, t_i\}$. Let c be the constant obtained from applying Lemma 4.1 with constants $\alpha_1 = \alpha$ and $\alpha_2 = \alpha/3$ and let $\alpha' = \min\{c/66, \alpha^2/12\}$. As in the proof of Theorem 3.1, if $k \ge \alpha' n$, then $n \le k/\alpha'$ and G itself is a linear vertex kernel. Therefore, from now on we assume that $k \le \alpha' n$.

Consider $V_0, \dots, V_g$, the partition obtained by applying Lemma 4.1 to G with constants α_1 and α_2. Since $k \le n\alpha^2/22$, $|V_0| \le \alpha n/6$ and $g \le 2/\alpha$, every vertex x of V_0 has a component V_j such that $|N(x) \cap V_j| \ge (\alpha n - |V_0|) \cdot \frac{\alpha}{2} \ge 6k$. For every $i \in [g]$, let H_i denote the set of elements x of V_0 such that V_i is the set of vertices such that $N(x) \cap V_i$ is maximum (breaking ties arbitrarily). The next lemma can be seen as an analogous of Claim 3.0.1, with a similar proof.

Lemma 4.2. *Let $(G, (s_1, t_1), \dots, (s_k, t_k))$ be a yes-instance. Then, there exists a path system, $P_1, \dots, P_k$, between s_i and t_i, that satisfy the following properties:*

- *For every $j \in [g]$ and $i \in [k]$, the intersection of V_j and P_i is a subpath of P_i.*
- *For every $i \in [k]$ and $j \in [g]$, $|P_i \cap H_j| \leq 2$.*

Proof. Let $(P_1, \dots, P_k)$ be a solution. For every $j \in [g]$, we say that $(P_1, \dots, P_k)$ satisfies the *property $\mathcal{F}_j$*, if $P_i \cap V_j$ has following features.

- For every $i \in [k]$, the intersection of V_j and P_i is a subpath of P_i.
- For every $i \in [k]$, $|P_i \cap H_j| \leq 2$.

Suppose $(P_1, \dots, P_k)$ does not satisfy the property $\mathcal{F}_j$, for some $j \in [g]$. For every $i \in [k]$, let a^i_j be the first vertex of P_i that belongs to either H_j or V_j. Likewise, let b^i_j be the last vertex of P_i that belongs to either H_j or V_j. We will define h^i_j and l^i_j as follows:

- If $a^i_j \in V_j$, then $h^i_j = a^i_j$, if not h^i_j is an element of $N(a^i_j) \cap V_j$;
- If $b^i_j \in V_j$, then $l^i_j = b^i_j$, if not l^i_j is an element of $N(b^i_j) \cap V_j$;

Note that by the definition of H_j, $|N(b^i_j) \cap V_j| \geq 6k$ when $b^i_j \in H_j$, which means that we can chose h^i_j, l^i_j, for $i \in [k]$, so that these vertices are pairwise disjoint. The set of pairs (h^j_i, l^j_i), for $i \in [k]$, together with $G[V_j]$ defines an instance of Vertex-Disjoint Paths with $k \leq \alpha' n$. Since, $G[V_j]$ is cn vertex-connected and $\alpha' \leq c/66$, Theorem 4.1 shows the existence of a solution to this problem. Let P^i_j denote the path between h^j_i and l^j_i in this solution. Now for each $i \in [k]$, let P'_i be the path obtained by replacing the subpath of P_i from a^j_i to b^j_i by P^i_j and possibly the edges $a^i_j h^i_j$ and $b^i_j l^i_j$, if they exist. Because all the P^i_j are disjoint, the P'_i are also disjoint, and by definition of the a^i_j, b^i_j, $(P'_1, \dots, P'_k)$ is a solution satisfying property $\mathcal{F}_i$.

Finally, let us show that if $(P_1, \dots, P_k)$ satisfies property $\mathcal{F}_{j'}$, for $j' \in [g]$ different from j, then so does $(P'_1, \dots, P'_k)$, which would conclude our proof. Indeed, fix $i \in [k]$, and let x_i and y_i be the first and last vertex of the subpath $P_i \cap V_{j'}$. Because $j \neq j'$, $|P'_i \cap V_{j'}| \leq |P_i \cap V_{j'}|$. Moreover, since the the subpath of P_i between x_i and y_i is a path of $V_{j'}$, it means that either a^i_j and b^i_j appear before x_i, or after y_i on the path, or a^i_j appears before x_i and b^i_j appears after y_i. In the former case, replacing the subpath of P_i between a^i_j and b^i_j does not change the intersection with $V_{j'}$, and in the latter case it means that $P'_i \cap V_{j'}$ is empty. Overall $(P'_1, \dots, P'_k)$ still satisfies the property $\mathcal{F}_{j'}$, which concludes our proof. □

We are now ready to describe the cubic vertex kernel. Because of Lemma 4.2, we know that each path intersects each V_j at most once, so there will be at most k disjoint paths to find inside each sets. Moreover, Theorem 4.1 implies that we will always be able to find such paths and it is clear that we don't need to keep too many vertices inside these sets, just enough to understand the boundary with V_0 and to maintain the high connectivity. Likewise, inside each set H_j, for $j \in [g]$, we know that the number of vertices used is bounded by $2k$. Moreover, because two vertices of H_j have a lot of neighbors in V_j, it is always possible to link these vertices using paths inside $G[V_j]$. For these reasons, again we don't need to keep too many vertices inside each H_j, just enough to understand the adjacencies with H, the other $H_{j'}$ and the other $V_{j'}$. Formalizing these are the goal of the following definitions.

- For any $j \in [g]$, let $A_j \subseteq H_j$ be a set of at most $8k^2$ vertices such that for every vertex $x \in H$, either $x \in A_j$, $|N(x) \cap A_j| \geq 4k$ or $N(x) \cap H_j \subseteq A_j$.
- For any $j \in [g]$, let $B_j \subseteq H_j$ be a set of at most $g32k^3$ vertices such that, for every vertex $x \in A_{j'}$ for some $j' \in [g]$ (including j), then either $|N(x) \cap B_j| \geq 4k$ or $N(x) \cap H_j \subseteq B_j$.

- For any pair $i, j \in [g]$, let $H^j_{i,j}$ be a set of $4k$ vertices of $H_j - (A_j \cup B_j)$ with some neighbors in $H_i - (A_i \cup B_i)$, or $N(H_i - (A_i \cup B_i)) \cap (H_j - (A_j \cup B_j))$ if this set is smaller than $4k$. Let $C^j_{i,j}$ be a minimal subset of $H_j - (A_j \cup B_j)$ containing $H^j_{i,j}$ such that for every vertex $x \in H^i_{i,j}$, either $|N(x) \cap C^j_{i,j}| \geq 4k$ or $N(x) \cap H_j \subseteq C^j_{i,j}$. Finally, let $C^i_{i,j}$ be a minimal subset of $H_i - (A_i \cup B_i)$ containing $H^i_{i,j}$ such that for every vertex $x \in H^j_{i,j}$, either $|N(x) \cap C^i_{i,j}| \geq 4k$ or $N(x) \cap H_i \subseteq C^i_{i,j}$. Note that $|C^j_{i,j}| \leq 20k^2$.
- For every $i, j \in [g]$, we define D^i_j as a set of $4k$ vertices of $H_j - (A_j \cup B_j \bigcup_{i' \in [g]} C^j_{i',j})$ adjacent to some vertices of V_i, or $N(V_i) \cap \left(H_j - (A_j \cup B_j \bigcup_{i' \in [g]} C^j_{i',j})\right)$ is this set is smaller than $4k$. We also define V^j_i as a set of $4k$ vertices of $V_i - H$ adjacent to some vertices of H_j, or $N(H_j) \cap (V_i - H)$ if this set is smaller than $4k$. Let E^i_j be a minimal set of vertices in $H_j - (A_j \cup B_j \bigcup_{i' \in [g]} C^j_{i',j})$ containing D^i_j and such that for every vertex $x \in V^j_i$, either $|N(x) \cap E^i_j| \geq 4k$ or $N(x) \cap H_j \subseteq E^i_j$. Likewise we define $V_{i,j}$ as a minimal set of vertices of V_i containing V^j_i such that for any $x \in D^i_j$, either $N(x) \cap V_i \subseteq V_{i,j}$ or $|N(x) \cap V_i| \geq 4k$.

For every $j \in [g]$, let H'_j denote the union of A_j, B_j, all the $C^j_{i,j}$ and all the E^i_j. Note that $|H'_j| = \mathcal{O}(k^3)$. Consider the graph G' obtained from G as follows:

- Remove for every $j \in [g]$ all the vertices of H_j not in H'_j.
- Replace V_j by V'_j, obtained by removing all the vertices of V_j not in H or some $V_{j,r}$ for $r \in [g]$ and add a clique W_j of $6k$ vertices adjacent to all the vertices of $V'_j \cup H_j$.

Note that for any $j \in [g]$, because every vertex of V'_j is adjacent to every vertex of W_j, $G'[V'_j]$ is such that any instance of the Vertex-Disjoint Paths problem has a solution for $k' \leq 6k$ (at most k' request pairs). Moreover, all the vertices of H belong to G'. We will show that G' is the desired kernel. For this purpose let us show the following lemmas.

Lemma 4.3. *If $(G, (s_1, t_1), \ldots, (s_k, t_k))$ is a yes-instance, then $(G', (s_1, t_1), \ldots, (s_k, t_k))$ is a yes-instance.*

Proof. Let $P_1, \ldots, P_k$ be a solution to $(G, (s_1, t_1), \ldots, (s_k, t_k))$. Because of Lemma 4.2, we can assume that for all $j \in [g]$ and $i \in [k]$, the intersection of V_j and P_i is always a subpath of P_i and moreover $|P_i \cap H_j| \leq 2$. For any $i \in [k]$ and $j \in [g]$, let (h^j_i, l^j_i) denote the extremities of $P_i \cap H_j$. For any $i \in [k]$, denote $P_i = x^i_1, \ldots, x^i_{\ell_i}$ for some integer ℓ_i. For *some* $r \in [\ell_i - 1]$, we will define some pair of vertices (u^i_r, v^i_{r+1}) of G', such that all the u^i_r and v^i_r for $i \in [k]$ and $r \in [\ell_i]$ are disjoint, apart from possibly $u^i_r = v^i_r$. The intuition here is that if $x^i_r \notin G'$, we will try to replace it by two vertices v^i_r and u^i_r of G', where v^i_r will play the same "role" as x^i_r regarding the previous vertex on the path, and u^i_r regarding the next. For example if the vertex before x^i_r belongs to $H_j \setminus G'$, and the previous vertex x^i_{r-1} on the path P_i is in H, then we will pick a vertex of A_j adjacent to x^i_{r-1} as v^i_r. Note that by definition of A_j, if $x^i_r \notin A_j$, it means that $|N(x) \cap A_j| \geq 4k$, so there is always enough "space" to pick one vertex in $N(x) \cap A_j$ different from all the other $u^{i'}_{r'}$ and $v^{i'}_{r'}$. Formally:

- If x^i_r and x^i_{r+1} belong to G', then $u^i_r = x^i_r$ and $v^i_{r+1} = x^i_{r+1}$.
- If $x^i_r \in A_j$ and $x^i_{r+1} \in H_{j'}$, then $u^i_r = x^i_r$ and v^i_{r+1} is a vertex of $B_{j'}$ adjacent to x^i_r (and symmetrically if $x^i_{r+1} \in A_j$ and $x^i_r \in H_{j'}$).
- If $x^i_r \in H$ and $x^i_{r+1} \in H_{j'}$, then $u^i_r = x^i_r$ and v^i_{r+1} is a vertex of $A_{j'}$ adjacent to x^i_r (and symmetrically).

- If $x_r^i \in (H_j \setminus A_j)$ and $x_{r+1}^i \in V_{j'}$, then u_r^i is a vertex of $E_{j'}^j$ and v_{r+1}^i is a vertex of $V'_{j'}$ adjacent to u_r^i (and symmetrically).
- If $x_r^i \in (H_j \setminus A_j)$ and $x_{r+1}^i \in (H_{j'} \setminus A_{j'})$, then let u_r^i be a vertex of $C_{j,j'}^j$ and v_{r+1}^i be a vertex of $C_{j,j'}^{j'}$ such that u_r^i and v_{r+1}^i are adjacent.

It is important to note that for every $i \in [k]$ and $r \in [\ell_i - 1]$, u_r^i and v_{r+1}^i are *adjacent* in G'. Also, it is important to remember that, if x_r^i is the first vertex of $P_i \cap V_j$, then v_r^i is a vertex of V'_j. Likewise, if x_r^i is the last vertex of $P_i \cap V_j$, then u_r^i is a vertex of V'_j. Moreover, the elements u_r^i and v_r^i are not defined when x_r^i is an internal vertex of the path $P_i \cap V_j$. This means that the number of vertices u_r^i or v_r^i defined in each V'_j is bounded by $2k$. Likewise, since $|P_i \cap H_j| \leq 2$ for all $i \in [k]$ and $j \in [g]$, it means that for each $j \in [g]$, the number of u_j^i and v_j^i selected in H'_j is at most $4k$. Because of the choices of the sizes of the sets $A_i, B_i, C_{j,i}^i, E_j^i$ and W_j, this implies that it is possible to make the choices of all the u_i^j and v_i^j such that these vertices are pair-wise disjoint. Indeed suppose for example that $x_r^i \in A_j$ and $x_{r+1}^i \in H_{j'}$ for some $i \in [k]$, $j \in [g]$ and $r \in [\ell_i - 1]$. Since $x_r^i \in G'$, it means that x_{r+1}^i is not in G'. However, by definition of $B_{j'}$, it means that $|N(x_r^i) \cap B_{j'}| \geq 4k$, so we can chose one among them different from all of the other selected vertices $u_{r'}^{i'}$ and $v_{r'}^{i'}$. The other cases are identical.

Next note that, if x_r^i is a vertex of $P_i \cap H_j$, then u_r^i and v_r^i are elements of H'_j. As explained, we want to use u_r^i and v_r^i to replace x_r^i in P_i. For that purpose, we need to find a path between these two vertices, which we will do by using the fact that they both belong to H'_j and thus have a lot of neighbors in W_j. Let us assign, for every $i \in [k]$ and $r \in [\ell_i]$ such that $x_r^i \in H'_j$, two vertices $s(x_r^i)$ and $l(x_r^i)$ of W_j such that $s(x_r^i)$ is adjacent to u_r^i and $l(x_r^i)$ is adjacent to v_r^i. Once again, since there are at most $2k$ vertices of P_i in each H_j and $|W_j| = 6k$, we can chose all these vertices to be disjoint and avoid all the vertices of u_r^i and v_r^i. For any fixed $j \in [g]$, consider an instance of the Vertex-Disjoint Paths problem defined by the following requests.

- All the $(s(x_r^i), l(x_r^i))$ for $x_r^i \in H_j$ for some $i \in [k]$ and $r \in [\ell_i]$.
- All the $(v_r^i, u_{r'}^i)$ for some $i \in [k]$ such that x_r^i is the first vertex of $P_i \cap V_j$ and $x_{r'}^i$ its last.

This define in an instance of Vertex-Disjoint Paths, with with $k_i \leq 5k$. By definition of $G'[V'_j]$, it is possible to find a solution to this problem. Let Q_r^i denote the path of such a solution between $s(x_r^i)$ and $l(x_r^i)$ and P_j^i the one between v_r^i and $u_{r'}^i$ where x_r^i is the first vertex of $P_i \cap V_j$ and $x_{r'}^i$ its last. For every $i \in [k]$, consider the path P'_i of G' obtained from P_i by:

- Replacing each $x_r^i \in H_j$ by $H_r^i := u_r^i.Q_r^i.v_r^i$.
- Replacing $P_i \cap V_j$ by the path P_i^j.

Claim 4.1.1. *For every $i \in [k]$, P'_i is a path from s_i to t_i.*

Proof. Let r be the first index such that $x_r^i \in V_j$ for some $j \in [g]$. Because each H_q^i is a path from v_q^i to u_q^i and each u_q^i is adjacent to v_{q+1}^i, we can see that the union of the H_q^i, for $q \leq r$, forms a path from s_i to u_r^i.

Similarly, for any maximal sequence $x_{r_1}^i, \ldots, x_{r_2}^i$ of P_i such that none of the vertices belong to any V_j for $j \in [g]$, then the union of all the Q_r^i, $r \in \{r_1, \ldots, r_2\}$ forms a path from $u_{r_1}^i$ to $v_{r_2}^i$.

We conclude the proof by noting that for all $i \in [k]$ and $j \in [g]$, P_j^i is a path between v_r^i and $u_{r'}^i$ where x_r^i is the first vertex of $P_i \cap V_j$ and $x_{r'}^i$ its last. Moreover, u_{r-1}^i is adjacent to u_r^i and $v_{r'+1}^i$ is adjacent $u_{r'}^i$. □

Moreover, by our construction, it is clear that the P'_i are disjoint, as all the (u_r^i, v_r^i), H_q^i and Q_q^i are. This concludes the proof. □

Let us now prove the other direction.

Lemma 4.4. *If $(G', (s_1, t_1), \ldots, (s_k, t_k))$ is a yes-instance, then $(G, (s_1, t_1), \ldots, (s_k, t_k))$ is a yes-instance.*

Proof. Let $(P_1, \ldots, P_k)$ be a solution to the instance $(G', (s_1, t_1), \ldots, (s_k, t_k))$. The same proof as in Lemma 4.2 would show that the intersection of every P_i with V'_j is a subpath of P_i. Moreover, note that the only vertices of G' that do not exists in G are the vertices of W_j. For every $i \in [k]$ and $j \in [g]$, let a^i_j denote the first vertex of $P_i \cap V'_j$ and b^i_j the last. Let us define p^i_j and q^i_j as follows:

- If a^i_j (reps. b^i_j) belongs to $V'_j - W_j$, then $p^i_j = a^i_j$ (resp. $q^i_j = b^i_j$).
- If a^i_j (resp. b^i_j) belongs to W_j, then let a (resp. b) be the vertex appearing just before (resp. just after) on P'_i. We define p^i_j (reps. q^i_j) as any vertex of V_j adjacent to a (reps. b).

Note that in the second case, it means that a or b belong to H_j, and thus we can chose all the p^i_j and q^i_j for $i \in [k]$ and $j \in [g]$ so that these vertices are disjoint. Once again, we can solve in every V_j the instance of the Vertex-Disjoint Paths problem defined by the (p^i_j, q^i_j) for $i \in [k]$ and $j \in [g]$. Moreover, if we note Q^i_j the path of this solution between p^i_j and q^i_j and define P'_i the path obtained from P_i by replacing each subpath $P_i \cap V'_j$ by Q^i_j, then P'_i is a path between s_i and t_i in G. Finally, because we chose all the p^i_j and q^i_j to be disjoint, and because the Q^i_j are disjoint as well, the $P'_1, \ldots, P'_k$ are pairwise disjoint and thus form a solution to the original disjoint path problem in G, which concludes the proof. ☐

Lemmas 4.3 and 4.4 imply the following theorem

Theorem 4.2. *The* Vertex-Disjoint Paths *problem on everywhere α-dense graphs admits a vertex kernel of size $\mathcal{O}(k^3)$.*

5 Subexponential Algorithms and Linear Vertex Kernel for Cut Problems

In this section we give parameterized subexponential time algorithms and linear vertex kernel for several cut problems, such as Edge Odd Cycle Transversal, Minimum Bisection, d-Way Cut, Multiway Cut, and Multicut, on everywhere dense graphs. Our algorithm uses random sampling akin to the one used in designing PTASes for problems on everywhere dense graphs. We first give the sampling primitive and then use this to design our kernel and subexponential time algorithms. Later, we show how to derandomize these algorithms.

5.1 A sampling Primitive

By abusing notation, if S is a multi-set and X a set, then $|S \cap X|$ counts the number of elements of S (with duplicates) belonging to X.

Lemma 5.1. *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 $||S \cap X| - \frac{t(\epsilon_1, \epsilon_2)|X|}{n}| \geq \epsilon_1 t(\epsilon_1, \epsilon_2)$ is smaller than $\epsilon_2 |\mathcal{K}|$.*

Proof. For any set $X \in \mathcal{K}$, $|S \cap X|$ follows a binomial distribution with parameters $t(\epsilon_1, \epsilon_2)$ and $p = \frac{|X|}{n}$: Using Chebyshev's inequality, we can show that:

$$\Pr[||S \cap X| - pt| \geq \epsilon_1 t] \leq \frac{p(1-p)}{\epsilon_1^2 t} \leq \frac{\epsilon_2}{2}.$$

Let B be the set of elements $X \in \mathcal{K}$ such that $||S \cap X| - pt| \geq \epsilon_1 t$. We have,

$$E[|B|] = \sum_{X \in \mathcal{K}} \Pr[||S \cap X| - pt| \geq \epsilon_1 t] \leq \frac{\epsilon_2 |\mathcal{K}|}{2}$$

and thus, by Markov's inequality, we have that $\Pr[|B| \geq \epsilon_2 |\mathcal{K}|] \leq 1/2$, which concludes the proof. □

5.2 Edge odd cycle transversal

In this subsection, we develop a subexponential-time parameterized algorithm for Edge Odd Cycle Transversal on everywhere dense graphs. Then, we extract an ingredient of this proof and use it to design a vertex kernel for the problem.

Theorem 5.1. *For any fixed $\alpha > 0$, there exists a $2^{\mathcal{O}(\sqrt{k/\alpha})} n^{\mathcal{O}(1)}$-time algorithm for* Edge Odd Cycle Transversal *on everywhere α-dense graphs.*

Proof. Let G denote the input graph. If $k \geq (\alpha n/200)^2$, then the algorithm tries all bipartitions of $V(G)$ and counts the number of non-crossing edges and gives an answer in time $2^n n^{\mathcal{O}(1)} \leq 2^{\sqrt{200k/\alpha}} n^{\mathcal{O}(1)}$.

Suppose now $k \leq (\alpha n/200)^2$ and G admits a solution S of size at most k. Let (A, B) be the corresponding bipartition and denote by L the set of vertices adjacent to more than $\alpha n/200 \geq \sqrt{k}$ edges of S. As $k \geq \frac{|L|\sqrt{k}}{2}$, we have that $|L| \leq 2\sqrt{k} \leq \alpha n/100$. Let $A_1 = A \setminus L$ and $B_1 = B \setminus L$. Without loss of generality, we can assume that $n_1 = |A_1| \geq |B_1|$. By trying n numbers, we also assume that the algorithm knows the value of n_1.

The algorithm then picks a (multi)set M of $t = t(\alpha/200, \alpha/200) = (\frac{200}{\alpha})^3$ vertices in $V(G)$, uniformly at random. With probability at least $(\frac{n_1}{n})^t$, all these vertices belong to A_1 and thus, by Lemma 5.1, the set X of vertices x such $\left|\frac{|N(x) \cap M| n_1}{t} - d_{A_1}(x)\right| \geq \frac{\alpha n_1}{200}$ is smaller than $\alpha n_1/200$ with probability at least $(\frac{n_1}{n})^t \cdot \frac{1}{2} \geq (\frac{1}{4})^{t+1}$. From now on suppose that it is the case and let B' be the set of vertices of G such that $\frac{|N(x) \cap M| n_1}{t} \geq d(x)/2 + \frac{\alpha n}{25}$ and A' be the set of vertices of G such that $\frac{|N(x) \cap M| n_1}{t} \leq d(x)/2 - \frac{\alpha n}{25}$.

The following claim is the crux behind the correctness of our classification, namely, we will be able to know (by just having M) for a large number of vertices to which side they should belong.

Claim 5.1. *The following statements are true:*

- *For every $x \in A_1 \setminus X$, $x \in A'$.*
- *For every $x \in B_1 \setminus X$, $x \in B'$.*
- *For every $x \in A \setminus X$, $x \notin B'$*
- *For every $x \in B \setminus X$, $x \notin A'$*

Proof of claim. Let $x \in B_1 \setminus X$. Since $x \in B_1$, it means that $d_{A_1}(x) \geq (d(x) - |A \cap L| - \alpha n/200) \geq (d(x) - \alpha n/50)$. Since $x \notin X$, it means that $|\frac{|N(x) \cap M| n_1}{t} - d_{A_1}(x)| \leq \alpha n_1/200$, and thus $\frac{|N(x) \cap M| n_1}{t} \geq (d_{A_1}(x) - \alpha n/200) \geq (d(x) - \alpha n(1/50 + 1/200))$. Moreover, since $d(x) \geq \alpha n$, we get that $x \in B'$. The argument for $x \in A_1 \setminus X$ is identical.

Suppose now that $x \in B \setminus X$. Because (A, B) is an optimal partition, $d_A(x) \geq d(x)/2$, which implies that $d_{A_1}(x) \geq d(x)/2 - |A \cap L| \geq d(x)/2 - \alpha n/50$. Thus, since $x \notin X$, we have $\frac{|N(x) \cap M| n_1}{t} \geq d(x)/2 - (1/50 + 1/200)\alpha n$, and therefore $x \notin A'$. The argument for $x \in A \setminus X$ is identical. ♦

Now define S_A as the set of vertices x such that $|N(x) \cap B'| \geq d(x)/2 + \alpha n/25$ and S_B as the set of vertices x such that $|N(x) \cap A'| \geq d(x)/2 + \alpha n/25$.

Claim 5.2. *$A_1 \subseteq S_A \subseteq A$ and $B_1 \subseteq S_B \subseteq B$.*

Proof of claim. Let $x \in B_1$, which means that $|N(x) \cap A_1| \geq d(x) - |A_1 \cap L| - \alpha n/200 \geq (d(x) - \alpha n/50)$. However, by Claim 5.3, $(A_1 \setminus X) \subseteq A'$. This imply that $|N(x) \cap A'| \geq |N(x) \cap A_1| - |X| \geq (d(x) - (1/50 + 1/200)\alpha n)$ and thus $x \in S_B$. Now let $x \in A$. We know that $|N(x) \cap A| \leq d(x)/2$, and by Claim 5.3, $(A' \setminus A) \subseteq X$. These imply that $|N(x) \cap A'| \leq |N(x) \cap A| + |X| \leq d(x)/2 + \alpha n/200$ and thus $x \notin S_B$. This proves that $B_1 \subseteq S_B \subseteq B$. The proof of $A_1 \subseteq S_A \subseteq A$ is identical. ♦

Let $R = V(G) \setminus (S_A \cup S_B)$. The previous claim shows that $R \subseteq L$, $S_A \subseteq A$ and $S_B \subseteq B$. To find the partition (A, B), we only need to figure out where do the vertices in R belong. However, since $|R| \leq |L| \leq 2\sqrt{k}$, we can try all the bipartitions of R in time $2^{2\sqrt{k}}$, which concludes the proof. □

When $k \leq \alpha n/200$, the set L, using the notations of the previous proof, is empty. This means that the algorithm runs in polynomial time, when $k \leq \alpha n/200$. We summarize this in the next lemma.

Lemma 5.2. *There exists a randomized polynomial time algorithm for the* Edge Odd Cycle Transversal *problem on everywhere α-dense graphs, when $k \leq \alpha n/200$.*

Lemma 5.2 implies the following result.

Theorem 5.2. Edge Odd Cycle Transversal *admits a linear vertex kernel on everywhere α-dense graphs.*

Proof. Let (G, k) be an instance of Edge Odd Cycle Transversal. If $k \leq \alpha n/200$, then we apply Lemma 5.2 and solve the problem in polynomial time. Based on the answer of Lemma 5.2, we either return a solution or a trivial no-instance of the problem. However, now we have that $k \geq \alpha n/200$, and hence $n \leq \frac{200k}{\alpha} = \mathcal{O}(k)$. This concludes the proof. □

5.3 Minimum Bisection

In this subsection, we develop a subexponential-time parameterized algorithm for Minimum Bisection on everywhere dense graphs. Then, we extract an ingredient of this proof and use it to design a vertex kernel for the problem. The proof is analogous to Theorem 5.1.

Theorem 5.3. *For any fixed $\alpha > 0$, there exists a $2^{\mathcal{O}(\sqrt{k/\alpha})} n^{\mathcal{O}(1)}$-time algorithm for* Bisection *on everywhere α-dense graphs.*

Proof. Let G denote the input graph. If $k \geq (\alpha n/200)^2$, then the algorithm tries all bipartitions of $V(G)$ into equal parts and counts the number of crossing edges and gives an answer in time $2^n n^{\mathcal{O}(1)} \leq 2^{\sqrt{200k/\alpha}} n^{\mathcal{O}(1)}$.

Suppose now $k \leq (\alpha n/200)^2$ and G admits a solution S of size at most k. Let (A, B) be the corresponding bipartition and denote by L the set of vertices adjacent to more than $\alpha n/200 \geq \sqrt{k}$ edges of S. As $k \geq \frac{|L|\sqrt{k}}{2}$, we have that $|L| \leq 2\sqrt{k} \leq \alpha n/100$. Let $A_1 = A \setminus L$

and $B_1 = B \setminus L$. Without loss of generality, we can assume that $n_1 = |A_1| \geq |B_1|$. By trying n numbers, we also assume that the algorithm knows the value of n_1.

The algorithm then picks a (multi)set M of $t = t(\alpha/200, \alpha/200) = (\frac{200}{\alpha})^3$ vertices in $V(G)$, uniformly at random. With probability at least $(\frac{n_1}{n})^t$, all these vertices belong to A_1 and thus, by Lemma 5.1, the set X of vertices x such $\left|\frac{|N(x)\cap M|n_1}{t} - d_{A_1}(x)\right| \geq \frac{\alpha n_1}{200}$ is smaller than $\alpha n_1/200$ with probability at least $(\frac{n_1}{n})^t \cdot \frac{1}{2} \geq (\frac{1}{4})^{t+1}$. From now on suppose that it is the case and let B' be the set of vertices of G such that $\frac{|N(x)\cap M|n_1}{t} \geq d(x)/2 + \frac{\alpha n_1}{25}$ and A' be the set of vertices of G such that $\frac{|N(x)\cap M|n_1}{t} \leq d(x)/2 - \frac{\alpha n}{25}$.

The following claim is the crux behind the correctness of our classification, namely, we will be able to know (by just having M) for a large number of vertices to which side they should belong.

Claim 5.3. *The following statements are true:*

- *For every $x \in A_1 \setminus X$, $x \in A'$.*
- *For every $x \in B_1 \setminus X$, $x \in B'$.*
- *For every $x \in A \setminus X$, $x \notin B'$*
- *For every $x \in B \setminus X$, $x \notin A'$*

Proof of claim. Let $x \in B_1 \setminus X$. Since $x \in B_1$, it means that $d_{A_1}(x) \geq (d(x) - |A \cap L| - \alpha n/200) \geq (d(x) - \alpha n/50)$. Since $x \notin X$, it means that $|\frac{|N(x)\cap M|n_1}{t} - d_{A_1}(x)| \leq \alpha n_1/200$, and thus $\frac{|N(x)\cap M|n_1}{t} \geq (d_{A_1}(x) - \alpha n/200) \geq (d(x) - \alpha n(1/50 + 1/200))$. Moreover, since $d(x) \geq \alpha n$, we get that $x \in B'$. The argument for $x \in A_1 \setminus X$ is identical.

Suppose now that $x \in B \setminus X$. Because (A, B) is an optimal partition, $d_A(x) \geq d(x)/2$, which implies that $d_{A_1}(x) \geq d(x)/2 - |A \cap L| \geq d(x)/2 - \alpha n/50$. Thus, since $x \notin X$, we have $\frac{|N(x)\cap M|n_1}{t} \geq d(x)/2 - (1/50 + 1/200)\alpha n$, and therefore $x \notin A'$. The argument for $x \in A \setminus X$ is identical. ♦

Now define S_A as the set of vertices x such that $|N(x) \cap B'| \geq d(x)/2 + \alpha n/25$ and S_B as the set of vertices x such that $|N(x) \cap A'| \geq d(x)/2 + \alpha n/25$.

Claim 5.4. *$A_1 \subseteq S_A \subseteq A$ and $B_1 \subseteq S_B \subseteq B$.*

Proof of claim. Let $x \in B_1$, which means that $|N(x) \cap A_1| \geq d(x) - |A_1 \cap L| - \alpha n/200 \geq (d(x) - \alpha n/50)$. However, by Claim 5.3, $(A_1 \setminus X) \subseteq A'$. This imply that $|N(x) \cap A'| \geq |N(x) \cap A_1| - |X| \geq (d(x) - (1/50 + 1/200)\alpha n)$ and thus $x \in S_B$. Now let $x \in A$. We know that $|N(x) \cap A| \leq d(x)/2$, and by Claim 5.3, $(A' \setminus A) \subseteq X$. These imply that $|N(x) \cap A'| \leq |N(x) \cap A| + |X| \leq d(x)/2 + \alpha n/200$ and thus $x \notin S_B$. This proves that $B_1 \subseteq S_B \subseteq B$. The proof of $A_1 \subseteq S_A \subseteq A$ is identical. ♦

Let $R = V(G) \setminus (S_A \cup S_B)$. The previous claim shows that $R \subseteq L$, $S_A \subseteq A$ and $S_B \subseteq B$. To find the partition (A, B), we only need to figure out where do the vertices in R belong. However, since $|R| \leq |L| \leq 2\sqrt{k}$, we can try all the bipartitions of R in time $2^{2\sqrt{k}}$, which concludes the proof. □

When $k \leq \alpha n/200$, the set L, using the notations of the previous proof, is empty. This means that the algorithm runs in polynomial time, when $k \leq \alpha n/200$. We summarize this in the next lemma.

Lemma 5.3. *There exists a randomized polynomial time algorithm for the* MINIMUM BISECTION *problem on everywhere α-dense graphs, when $k \leq \alpha n/200$.*

Lemma 5.3 implies the following result.

Theorem 5.4. Minimum Bisection *admits a linear vertex kernel on everywhere α-dense graphs.*

Proof. Let (G, k) be an instance of Minimum Bisection. If $k \leq \alpha n/200$, then we apply Lemma 5.3 and solve the problem in polynomial time. Based on the answer of Lemma 5.3, we either return a solution or a trivial no-instance of the problem. However, now we have that $k \geq \alpha n/200$, and hence $n \leq \frac{200k}{\alpha} = \mathcal{O}(k)$. This concludes the proof. $\square$

5.4 Cut-problems in everywhere α-dense graphs

The arguments used for Edge Odd Cycle Transversal and Minimum Bisection can be generalized to give a $(\frac{1}{\alpha})^{\mathcal{O}((1/\alpha)^3)} \cdot n^2 \cdot 2^{\mathcal{O}\left(\sqrt{\frac{k}{\alpha}}\log(\sqrt{\frac{k}{\alpha}})\right)}$ time algorithm for d-Way Cut on everywhere α-dense graphs. Here, the parameter k is the size of the solution. A partition of the vertex set $(A_1, \ldots, A_d)$ of the vertices is called an *optimal d-cut partition*, if the total number of edges crossing two parts is minimized.

Theorem 5.5. *The d-*Way Cut *problem, parameterized by the size of the solution admits an algorithm with running time $(\frac{1}{\alpha})^{\mathcal{O}((1/\alpha)^3)} \cdot n^2 \cdot 2^{\mathcal{O}\left(\sqrt{\frac{k}{\alpha}}\log(\sqrt{\frac{k}{\alpha}})\right)}$ on everywhere α-dense graphs*

Proof. Let (G, k) be an input instance to the d-Way Cut problem and let $(A_1, \ldots, A_d)$ be an optimal d-cut partition such that $|A_1| \geq \cdots \geq |A_d|$. There exists an exact algorithm with running time $2^n n^{\mathcal{O}(1)}$ algorithm for d-Way Cut 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 this algorithm runs in time $2^{\mathcal{O}(\sqrt{k/\alpha})}$, which ends our proof.

Suppose now $k \leq (\alpha n/200)^2$ and G admits a solution S of size at most k. Denote by L the set of vertices adjacent to more than $\alpha n/200 \geq \sqrt{k}$ edges of S. As $k \geq \frac{|L|\sqrt{k}}{2}$, we have that $|L| \leq 2\sqrt{k} \leq \alpha n/100$. Note that every part A_i of the optimal d-cut such that $|A_i| \leq \frac{\alpha n}{2}$ is entirely contained in L. Indeed, for every vertex $x \in A_i$, x has to be adjacent to at least $d(x) - |A_i| \geq \frac{\alpha n}{2}$ vertices of the solution. Let d' denote the maximum index such that $|A_i| \leq \frac{\alpha n}{2}$ for every $i \in [d']$ (recall, that we assumed $|A_1| \geq \cdots \geq |A_d|$). Note, and this is quite important for our argument, that $d' \leq \frac{2}{\alpha}$. By trying at most $\lceil \frac{2}{\alpha} \rceil$ possibilities, we can assume that the algorithm knows the value of d'.

The goal now will be to do sampling inside each A_i for $i \in [d']$, in order to guess the degree of the vertices of G inside this set. For every $i \in [d']$, let $n_i = |A_i|$ and define $A_i^b = A_i \setminus L$. Note that if $x \in A_i^b$, then $|N(x) \setminus A_i| \leq \alpha n/200$. By trying at most $\frac{400}{\alpha}$ possibilities for each $i \in [d']$, so $(\frac{400}{\alpha})^{d'} \leq (\frac{400}{\alpha})^{2/\alpha}$ in total, we can assume that the algorithm is given an integer n'_i such that $|n'_i - n_i| \leq \frac{\alpha n}{400}$ for every $i \in [d']$.

For each $i \in [d']$, the algorithm will then pick a (multi)set M_i of $t = t(\alpha/400, \alpha^2/(400)) = (\frac{400}{\alpha})^3$ random vertices obtained by doing independent and uniform random sample in $V(G)$. With probability greater than $(\frac{n_i}{n})^t \geq (\frac{\alpha}{2})^t$, we have that $M_i \subseteq A_i$. This means that, by Lemma 5.1, the set X_i of vertices x such $|\frac{|N(x) \cap M_i| n_i}{t} - d_{A_i}(x)| \geq \frac{\alpha n_i}{400}$ is smaller than $\alpha^2 n/400$, with probability at least $(\frac{n_i}{n})^t \cdot \frac{1}{2} \geq (\frac{\alpha}{4})^t$. By doing the previous sampling at most $(\frac{4}{\alpha})^t = (\frac{4}{\alpha})^{(\alpha/400)^3}$ times, we know this is the case for all $i \in [d']$ with constant probability, and let $X = \bigcup_{i \in [d']} X_i$.

Note that $|X| \leq \alpha n/200$ and for every vertex $x \notin X$ and $i \in [d']$:

$$\begin{aligned}\left|\frac{|N(x)\cap M_i|n_i'}{t} - d_{A_i}(x)\right| &\leq \left|\frac{|N(x)\cap M_i|n_i}{t} - d_{A_i}(x) + \frac{|N(x)\cap M_i|n_i'}{t} - \frac{|N(x)\cap M_i|n_i}{t}\right| \\ &\leq \left|\frac{|N(x)\cap M_i|n_i'}{t} - d_{A_i}(x)| + |\frac{|N(x)\cap M_i|n_1'}{t} - \frac{|N(x)\cap M_i|n_i}{t}\right| \\ &\leq \frac{\alpha n}{200}.\end{aligned}$$

For any $i \in [d']$, let A_i' denote the set of vertices of G such that $\frac{|N(x)\cap M_i|n_i'}{t} \geq d(x) - \frac{\alpha n}{25}$.

Claim 5.5.1. *For every $i, j \in [d']$, $i \neq j$:*

- *For every $x \in A_i^b \setminus X$, $x \in A_i'$.*
- *For every $x \in A_j^b \setminus X$, $x \notin A_i'$*

Proof of claim. Let $x \in A_i^b \setminus X$. Since $x \in A_i^b$, it means that $d_{A_i}(x) \geq (d(x) - \alpha n/200)$. Since $x \notin X$, it means that $|\frac{|N(x)\cap M_i|n_i'}{t} - d_{A_i}(x)| \leq \alpha n/200$, and thus $\frac{|N(x)\cap M|n_i}{t} \geq (d_{A_i}(x) - \alpha n/100)$.

Suppose now that $x \in A_j^b \setminus X$. Since $x \in A_j^b$, it means that $d_{A_i}(x) \leq \alpha n/200$. However, since $x \notin X$, we have that $|\frac{|N(x)\cap M_i|n_i'}{t} - d_{A_i}(x)| \leq \alpha n/200$, and thus $\frac{|N(x)\cap M|n_i'}{t} \leq \alpha n/100$. This ends the proof as $d(x) \geq \alpha n$. ♦

The previous claim implies that $A_i' \subseteq (A_i \cup X)$. Now for every $i \in [d']$, define S_i to be the set of vertices $x \in V(G)$ such that $|N(X) \cap A_i'| \geq d(x) - \alpha n/50$.

Claim 5.5.2. *For every $i \in [d']$, $A_i^b \subseteq S_i \subseteq A_i$.*

Proof of claim. Suppose $x \in A_i^b$. By definition of A_i^b, it means that $|N(x) \cap A_i^b| \geq |N(x) \cap A_i| - |L| \geq (d(x) - \alpha n/200) - \alpha n/100$. Moreover, Claim 5.5.1 shows that $(A_i^b \setminus X) \subseteq A_i'$. This means that for $x \in A_i^b$, $|N(x) \cap A_i'| \geq |N(x) \cap A_i^b| - |X| \geq d(x) - \alpha n/50$ and thus $x \in S_i$.

Suppose now that $x \in A_j$ for $j \neq i$. By optimality of the partition $(A_1, \ldots, A_d)$, we have that $|N(x) \cap A_i| \leq d(x)/2$. This implies that $|N(x) \cap A_i'| \leq d(x)/2 + |X| \leq d(x)/2 + \alpha n/200$. Since $d(x) \geq \alpha n$, it implies that $x \notin S_i$. ♦

Let $R = V(G) \setminus (\bigcup_{i\in[d']} S_i)$. Claim 5.5.2 shows that $R \subseteq L$, and $S_i \subseteq A_i$ for all $i \in [d']$. Moreover, the algorithm finds R in times $(\frac{1}{\alpha})^{\mathcal{O}((1/\alpha)^3)} \cdot n^2$ with constant probability, as it only needs to compute the degree of every vertex $x \in G$ inside the M_i and then inside the A_i'. This means that, in order to find the partition $(A_1, \ldots, A_d)$, the algorithm only needs to figure out where do the vertices in R belong. Remember that $|R| \leq |L| \leq 2\sqrt{k}$ and $d' \leq \frac{2}{\alpha}$, so the algorithm will try all the $(\frac{2}{\alpha}+1)^{|L|}$ assignment of the vertices of L to the parts of $A_1, \ldots, A_l$ and all the $|L|^{|L|}$ possible partitions for the remaining vertices. Overall this gives an algorithm in $(\frac{1}{\alpha})^{\mathcal{O}((1/\alpha)^3)} \cdot n^2 \cdot 2^{\mathcal{O}\left(\sqrt{\frac{k}{\alpha}}\log(\sqrt{\frac{k}{\alpha}})\right)}$. This concludes the proof. □

Again, when $k \leq \alpha n/200$, the set L, using the notations of the previous proof, is empty. This means that the algorithm runs in polynomial time, when $k \leq \alpha n/200$. We summarize this in the next lemma.

Lemma 5.4. *There exists a randomized polynomial time algorithm for the d-Way Cut problem on everywhere α-dense graphs, when $k \leq \alpha n/200$.*

Lemma 5.4 implies the following result.

Theorem 5.6. *d-Way Cut admits a linear vertex kernel on everywhere α-dense graphs.*

In fact Lemma 5.4 can also be proved for Multicut and Multiway Cut. Indeed, for both of these problems if S is a solution of size $k \leq \frac{\alpha n}{200}$, then **every vertex** belongs to the component of $G - S$ containing the vast majority of its neigbhors, and the exact same proof as Lemma 5.4 implies the following result.

Lemma 5.5. *There exists a randomized polynomial time algorithm for the for* Edge Multicut *problem and the* Multiway Cut *problem on everywhere α-dense graphs, when $k \leq \alpha n/200$.*

Lemma 5.5 yields the following result.

Theorem 5.7. Multicut *and* Multiway Cut *admit a linear kernel on everywhere α-dense graphs.*

6 De-randomization

In this section, we will show how to de-randomize all our applications of Lemma 5.1.

Lemma 6.1. *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 $2^{100/(\epsilon_1^2\epsilon_2)}|\mathcal{S}(n)|$ 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 $T \in \mathcal{T}$, $T \subseteq A$ 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.*

Proof. Throughout the proof, we assume that the elements of U are ordered $u_1, \dots, u_n$ and abusing notations, we will associate sets of integers of $[n]$ with sets of elements of U.

Let t be a constant that we will fix later on and consider a random set S of elements of U obtained by picking uniformly an element of $\mathcal{S}(n)$ and keeping the first t elements. Because $\mathcal{S}(n)$ is a family of pairwise independent permutations, we have that for any $x \in U$, $\Pr[x \in U] = \frac{t}{n}$ and for any $x, y \in U$, $\Pr[x \in S \cap y \in S] = \frac{t^2}{n(n-1)}$. For any element $v \in U$, let X_v be the random variable equal to 1 if $v \in S$ and 0 otherwise.

Claim 6.0.1. *For any constants c_1, c_2 in $[0,1]$ and set $X \subseteq U$, if $t \geq \frac{2}{c_1^2 c_2}$, then $\Pr[||S \cap X| - \frac{t|X|}{n}| \geq c_1 t] \leq c_2$*

Proof. We want to apply Chebyshev's inequality, and therefore need an upper bound on the variance of $|S \cap X| = \sum_{v \in S} X_v$. For every $v \in X$, $var(X_v) = E[(X_v)^2] - E[X_v]^2 \leq \frac{t}{n}$ and for every $v, z \in X$, $cov(X_z, X_v) = E[X_v X_z] - E[X_v]E[X_z] = Pr[x, v \in S] - Pr[x \in S]Pr[v \in S]$, which implies that

$$cov(X_z, X_v) = \frac{t^2}{n(n-1)} - \frac{t^2}{n^2} = \frac{t^2}{n^2(n-1)}.$$

Overall this implies that $var(|S \cap X|) \leq \frac{|X|t}{n} + \frac{(t|X|)^2}{n^2(n-1)} \leq 2t$, and by Chebyshev's inequality, we have for any constant δ:

$$\Pr[||S \cap X| - E[|S \cap X|]| \geq \delta t] \leq \frac{var(|S \cap X|)}{(\delta t)^2}$$

and therefore:

$$\Pr[||S \cap X| - \frac{t|X|}{n}| \geq t\delta] \leq \frac{2t}{(\delta t)^2}.$$

Which ends the proof as $t \geq \frac{2}{c_1^2 c_2}$. □

Let us now fix A a set of size at least $\epsilon_3 n$ and $X \subseteq A$. Suppose that $t \geq \frac{100}{(\epsilon_1\epsilon_3)^2\epsilon_2}$. The previous claim implies that with probability at least $1-\epsilon_2$, $|S \cap X| = \frac{t|X|}{n} \pm (\epsilon_1\epsilon_3)t/5$ and $|S \cap A| = \frac{t|A|}{n} \pm (\epsilon_1\epsilon_3)t/5$. This means that $|S\cap A| = \frac{t|A|}{n}(1 \pm \frac{(\epsilon_1\epsilon_3)n}{5|A|})$ and thus, using Observation 2.1 and the fact that $|A| \geq \epsilon_3 n$, we get:

$$\frac{t}{n} = \frac{|S \cap A|}{|A|}(1 \pm \frac{2\epsilon_1}{5}).$$

From this, we can deduce that:

$$\begin{aligned}\frac{\epsilon_1\epsilon_3 t}{5} &= \frac{\epsilon_1\epsilon_3}{5} \cdot \frac{n|S \cap A|}{|A|}(1 \pm \frac{2\epsilon_1}{5})\\ &\leq \frac{\epsilon_1|S \cap A|}{5}(1 \pm \frac{2\epsilon_1}{5})\\ &\leq \frac{\epsilon_1|S \cap A|}{2}\end{aligned}$$

and overall:

$$|S \cap X| = \frac{|X||S \cap A|}{|A|} \pm \epsilon_1|S \cap A|.$$

This means that for any $X \in \mathcal{K}$, the probability that $||S \cap X| - \frac{|X||S\cap A|}{|A|}| \geq |S \cap A|\epsilon_1$ is smaller that ϵ_2. By linearity of expectation, we get that the expected number of sets $X \in \mathcal{K}$ such that $||S \cap X| - \frac{|X||S\cap A|}{|A|}| \geq |S \cap A|\epsilon_1$ is smaller than $\epsilon_2|\mathcal{K}|$.

Therefore, there exists a permutation $S_1 \in \mathcal{S}(n)$ such that noting S the first t elements of S_1, the number of sets $X \in \mathcal{K}$ such that $||S \cap X| - \frac{|X||S\cap A|}{|A|}| \geq |S \cap A|\epsilon_1$ is smaller than $\epsilon_2|\mathcal{K}|$. For such a set, if we denote $T = S \cap A$, then T is a subset of A 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 $\epsilon_2|\mathcal{K}|$.

Finally, this means that the set $\mathcal{T}$ consisting of all the subsets of the first t elements of permutations of $\mathcal{S}(n)$ satisfies the properties of the lemma. □

Therefore, in all the proof using Lemma 5.1, we can replace the random sampling by simply trying all the elements of the family $\mathcal{T}$ provided by the Lemma 6.1.

6.1 De-randomization of EPTAS

Let us now explain how to adapt the previous argument in order to provide deterministic EPTAS for all the problems in [34].

6.1.1 Max-Cut

First, let us explain how to derandomize the Max-Cut algorithm of Goldreich et al. [30] to a deterministic EPTAS. Their algorithm uses $\mathcal{O}(1/\epsilon)$ calls of the sampling argument, which means that we cannot simply use Lemma 6.1, $\mathcal{O}(1/\epsilon)$ times, as it would yield an algorithm with factor $n^{\mathcal{O}(1/\epsilon)}$ in the running time.

Let us briefly describe (in high level) the randomized algorithm of Goldreich at al. [30]. Let G be an α-dense graph and (A, B) be an optimal partition for Max-Cut. Let $l = \mathcal{O}(1/\epsilon)$ and partition $V(G)$ arbitrarily into sets $V_1, \dots V_l$ of size n/l. The goal of the algorithm is to decide the bipartition of each V_i one after the other. The algorithm starts by picking a sample of t vertices in $V \setminus V_1$. With constant probability, this sample is a good estimator for the degree inside A for almost all the vertices in V_1. Next, Goldreich et al. [30] show, and this is the core of the proof, is that modifying the partition (A, B) by placing the vertices of V_1 according to the estimated degree in A provided by T only increases the value of the partition by $\mathcal{O}(\epsilon^2 n^2)$. Let (A_1, B_1) be the partition obtained from (A, B) by placing the vertices of V_1 as we described. Now the algorithm will sample a new set of t vertices from $V \setminus V_2$. With constant probability, this sample is a good estimator for the degree inside A_1 for almost all the vertices in V_2. Let (A_2, B_2) be the partition obtained from (A_1, B_1) by placing the vertices of V_2 according to this estimate. By repeating this procedure l times, we obtain a partition (A_l, B_l) of the vertices of G whose value differs from the optimal by at most $\mathcal{O}(\epsilon^2 n^2) \cdot l = \mathcal{O}(\epsilon n^2)$ and this gives the desired approximation.

The problem here is that the l samples corresponds to a different partition, and the partition at the step j depends on all the previous samples. To avoid this problem, note that if (A_r, B_r) denotes the partition that the algorithm is trying to approximate after r steps, then $A_r \cap (\cup_{i \leq r} V_i)$ corresponds to the vertices $V_1, \ldots, V_r$ that the algorithm placed in A, which the algorithm knows, and $A_r \cap (\cup_{i>r} V_i)$ corresponds to the vertices of $A \cap (\cup_{i>r} V_i)$, which correspond to the *initial partition*. So, if we are able to find estimators for the degree in every $A \cap V_i$, we are able to get an estimation of the degree of the vertices inside A_r at any step of the algorithm. Note that for the randomised version, this does not change much, because the algorithm still has to sample l times. However, for us it is crucial that all these samplings correspond to the same partition, because then we can use the following lemma, whose proof is almost identical to the one of Lemma 6.1 to obtain a set of $n^{\mathcal{O}(1)}$ l-tuples, such that for any partition (A, B), there is one of the l-tuples $(S_1, \ldots, S_l)$ where each S_i gives a good estimation of the degree inside $V_i \cap A$.

Lemma 6.2. *For any constants ϵ_1 and ϵ_2 and ℓ, there exists a constant $t(\epsilon_1, \epsilon_2, \ell)$ such that if U is a universe on n elements, then there exists a set $\mathcal{T}$ of $|\mathcal{S}(n)| \cdot 2^{t(\epsilon_1, \epsilon_2, \ell)}$ ℓ-tuples of subsets of U, such that if $(A_1, \ldots, A_\ell)$ is a collection of disjoint subsets of U and for every i, $\mathcal{K}_i$ a collection of subsets of A_i, then there exists a ℓ-tuple $(T_1, \ldots, T_\ell) \in \mathcal{T}$ such that for each $i \in [\ell]$, the number of sets X of some $\mathcal{K}_i$ such that $||T_i \cap X| - \frac{|T_i||X|}{|A_i|}| \geq \epsilon_1 \frac{|X||T_i|}{|A_i|}$ is smaller than $\epsilon_2 \sum_{i \in [\ell]} |\mathcal{K}_i|$. Moreover, the set $\mathcal{T}$ can be computed deterministically in $n^{\mathcal{O}(1)}$ time.*

Lemma 6.2 directly derandomizes the algorithm of Goldreich at al. [30].

6.1.2 Other problems

In [30], they show how to adapt the algorithm to solve Min-Bisection as well as Max-Bisection on α-dense graphs. Our derandomization result can be used there directly as well to provide deterministic EPTASes for these problems. We can also derandomize the EPTASes for d-Way Cut and d-Multiway Cut on α-dense graphs and d-Correlation Clustering on general graphs given in [34], as they either rely on a generalization of the Max-Cut algorithm, or use only a constant number of times the sampling argument (independent of ϵ). We also note that, by first running the algorithm from Max-Cut and then using the ideas of [34] in order to solve the case where the solution is small, one can show this existence of an EPTAS for Edge Odd Cycle Transversal on α-dense graphs. This algorithm can be made deterministic using our method.

7 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?

Acknowledgments. 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 6.1.

References

[1] N. ALON, W. F. DE LA VEGA, R. KANNAN, AND M. KARPINSKI, *Random sampling and approximation of max-csps*, J. Comput. Syst. Sci., 67 (2003), pp. 212–243.

[2] N. ALON, A. M. FRIEZE, AND D. WELSH, *Polynomial time randomized approximation schemes for tutte-gröthendieck invariants: The dense case*, Random Struct. Algorithms, 6 (1995), pp. 459–478.

[3] N. ALON, D. LOKSHTANOV, AND S. SAURABH, *Fast FAST*, in Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I, 2009, pp. 49–58.

[4] N. ALON AND S. LOVETT, *Almost k-wise vs. k-wise independent permutations, and uniformity for general group actions*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Berlin, Heidelberg, 2012, Springer Berlin Heidelberg, pp. 350–361.

[5] G. ANDERSSON AND L. ENGEBRETSEN, *Property testers for dense constraint satisfaction programs on finite domains*, Random Struct. Algorithms, 21 (2002), pp. 14–32.

[6] S. ARORA, A. M. FRIEZE, AND H. KAPLAN, *A new rounding procedure for the assignment problem with applications to dense graph arrangement problems*, Math. Program., 92 (2002), pp. 1–36.

[7] S. ARORA, D. R. KARGER, AND M. KARPINSKI, *Polynomial time approximation schemes for dense instances of np-hard problems*, J. Comput. Syst. Sci., 58 (1999), pp. 193–210.

[8] C. BAZGAN, W. F. DE LA VEGA, AND M. KARPINSKI, *Polynomial time approximation schemes for dense instances of minimum constraint satisfaction*, Random Struct. Algorithms, 23 (2003), pp. 73–91.

[9] A. BJÖRKLUND, T. HUSFELDT, AND M. KOIVISTO, *Set partitioning via inclusion-exclusion*, SIAM J. Comput., 39 (2009), pp. 546–563.

[10] H. L. BODLAENDER, S. THOMASSÉ, AND A. YEO, *Kernel bounds for disjoint cycles and disjoint paths*, Theor. Comput. Sci., 412 (2011), pp. 4570–4578.

[11] B. Bollobás and A. Thomason, *Highly linked graphs*, Combinatorica, 16 (1996), pp. 313–320.

[12] N. Bousquet, J. Daligault, and S. Thomassé, *Multicut is FPT*, SIAM J. Comput., 47 (2018), pp. 166–207.

[13] Y. Cao, J. Chen, and J. Fan, *An $O(1.84^k)$ parameterized algorithm for the multiterminal cut problem*, Inf. Process. Lett., 114 (2014), pp. 167–173.

[14] J. Chen, Y. Liu, and S. Lu, *An improved parameterized algorithm for the minimum node multiway cut problem*, Algorithmica, 55 (2009), pp. 1–13.

[15] R. Chitnis, M. Cygan, M. Hajiaghayi, M. Pilipczuk, and M. Pilipczuk, *Designing FPT algorithms for cut problems using randomized contractions*, SIAM J. Comput., 45 (2016), pp. 1171–1229.

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

[17] M. Cygan, P. Komosa, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Randomized contractions meet lean decompositions*, CoRR, abs/1810.06864 (2018).

[18] M. Cygan, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and M. Wahlström, *Clique cover and graph separation: New incompressibility results*, ACM Trans. Comput. Theory, 6 (2014), pp. 6:1–6:19.

[19] M. Cygan, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Minimum bisection is fixed-parameter tractable*, SIAM J. Comput., 48 (2019), pp. 417–450.

[20] W. F. de la Vega and M. Karpinski, *Polynomial time approximation of dense weighted instances of MAX-CUT*, Random Struct. Algorithms, 16 (2000), pp. 314–332.

[21] R. Diestel, *Graph theory*, Springer Publishing Company, Incorporated, 2018.

[22] R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013.

[23] U. Feige, *Faster fast(feedback arc set in tournaments)*, CoRR, abs/0911.5094 (2009).

[24] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2006.

[25] F. V. Fomin, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and Y. Villanger, *Tight bounds for parameterized complexity of cluster editing with a small number of clusters*, J. Comput. Syst. Sci., 80 (2014), pp. 1430–1447.

[26] F. V. Fomin, D. Lokshtanov, S. Saurabh, and M. Zehavi, *Kernelization. Theory of parameterized preprocessing*, Cambridge University Press, Cambridge, 2019.

[27] F. V. Fomin and M. 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, 2013, pp. 505–516.

[28] A. M. Frieze and R. 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, IEEE Computer Society, 1996, pp. 12–20.

[29] I. Giotis and V. Guruswami, *Correlation clustering with a fixed number of clusters*, Theory of Computing, 2 (2006), pp. 249–266.

[30] O. Goldreich, S. Goldwasser, and D. Ron, *Property testing and its connection to learning and approximation*, J. ACM, 45 (1998), pp. 653–750.

[31] S. Hoory, N. Linial, and A. Wigderson, *Expander graphs and their applications*, Bulletin of the American Mathematical Society, 43 (2006), pp. 439–561.

[32] A. Huck, *A sufficient condition for graphs to be weaklyk-linked*, Graphs and Combinatorics, 7 (1991), pp. 323–351.

[33] M. Karpinski, *Polynomial time approximation schemes for some dense instances of np-hard optimization problems*, Algorithmica, 30 (2001), pp. 386–397.

[34] M. Karpinski and W. 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, ACM, 2009, pp. 313–322.

[35] ——, *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, 2010, pp. 3–14.

[36] K. Kawarabayashi and M. Thorup, *The minimum k-way cut of bounded size is fixed-parameter tractable*, in IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011, R. Ostrovsky, ed., IEEE Computer Society, 2011, pp. 160–169.

[37] K. Kawarabayashi and P. Wollan, *A shorter proof of the graph minor algorithm: the unique linkage theorem*, in Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010, L. J. Schulman, ed., ACM, 2010, pp. 687–694.

[38] S. Kratsch and M. Wahlström, *Compression via matroids: A randomized polynomial kernel for odd cycle transversal*, ACM Trans. Algorithms, 10 (2014), pp. 20:1–20:15.

[39] ——, *Representative sets and irrelevant vertices: New tools for kernelization*, J. ACM, 67 (2020), pp. 16:1–16:50.

[40] D. Lokshtanov, P. Misra, M. Pilipczuk, S. Saurabh, and M. Zehavi, *An exponential time parameterized algorithm for planar disjoint paths*, in Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020, K. Makarychev, Y. Makarychev, M. Tulsiani, G. Kamath, and J. Chuzhoy, eds., ACM, 2020, pp. 1307–1316.

[41] J. Madathil, R. Sharma, and M. 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.

[42] D. Marx, *Parameterized graph separation problems*, Theor. Comput. Sci., 351 (2006), pp. 394–406.

[43] D. Marx and I. Razgon, *Fixed-parameter tractability of multicut parameterized by the size of the cutset*, SIAM J. Comput., 43 (2014), pp. 355–388.

[44] C. Mathieu and W. 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, SIAM, 2008, pp. 176–182.

[45] K. Menger, *Zur allgemeinen kurventheorie*, Fundamenta Mathematicae, 10 (1927), pp. 96–115.

[46] P. Misra, S. Saurabh, R. Sharma, and M. 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, 2018, pp. 35:1–35:19.

[47] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.

[48] M. Pilipczuk, M. Pilipczuk, and M. Wrochna, *Edge bipartization faster than* 2^k, Algorithmica, 81 (2019), pp. 917–966.

[49] N. Robertson and P. D. Seymour, *Graph minors XIII. the disjoint paths problem*, J. Comb. Theory, Ser. B, 63 (1995), pp. 65–110.

[50] R. van Bevern, A. E. Feldmann, M. Sorge, and O. Suchý, *On the parameterized complexity of computing graph bisections*, in Graph-Theoretic Concepts in Computer Science - 39th International Workshop, WG 2013, Lübeck, Germany, June 19-21, 2013, Revised Papers, A. Brandstädt, K. Jansen, and R. Reischuk, eds., vol. 8165 of Lecture Notes in Computer Science, Springer, 2013, pp. 76–87.

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 $(s_1, t_1), \dots, (s_k, t_k)$.
Question: Does there exist a set of paths $P_1, \dots, P_k$, 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 $(s_1, t_1), \dots, (s_k, t_k)$.
Question: Does there exist a set of paths $P_1, \dots, P_k$, 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?