23rd International Symposium on Theoretical Aspects of Computer Science, STACS 2005, Lecture Notes in Computer Science, Springer Verlag, Vol. 3884, pp. 361–371, 2005.

Fast FPT-algorithms for cleaning grids*

Josep Diaz Dimitrios M. Thilikos

Abstract

We consider the problem that given a graph G and a parameter k asks whether the edit distance of G and a rectangular grid is at most k. We examine the general case where the edit operation are vertex/edge removals and additions. If the dimensions of the grid are given in advance, then we give a parameterized algorithm that runs in $2^{O(\log k \cdot k)} + O(n^3)$ steps. In the case where the dimensions of the grid are not given we give a parameterized algorithm that runs in $2^{O(\log k \cdot k)} + O(k \log k \cdot n^3)$ steps. We insist on parameterized algorithms with running times where the relation between the polynomial and the non-polynomial part is additive. For this, our algorithm design is based on the technique of kernelization. In particular we prove that for each version of the above problem there exists a kernel of size $O(k^4)$.

1 Introduction

An interesting problem with many applications in Computer Science is the problem of measuring the *degree of similarity* of two graphs G and H, where by degree of similarity is consider to be the minimum number of *edit operations* (insert, delete, and modify) needed to transform one graph into the other. This measure of similarity between graphs is also denoted as the *edit distance* by analogy with the related problem on strings. A particularly important application has been the recognition of objects and shapes, where the objects are represented by special type of labelled trees and use the operations, deletion, insertion and relabelling of nodes. The problem is NP-Complete in general graphs and can be computed by dynamic programming in polynomial time (see for ex. [10]). Further work has been done of computing the edit distance of more involve weighted trees, called shock trees, using and extended sets of operations: slice of edges, contract of edges, deform of edges, (see for ex. [8])

In this paper, we consider decision problems associated with the editing distance between G and H when H is a grid of size $p \times q$ (we call such grid (p, q)*-grid and we denote it as* $H_{p,q}$). An *edit operation* on a graph G can be either the removal or the insertion of either an edge or a vertex. We represent these operations by the members

*Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Campus Nord – Edifici Ω, c/Jordi Girona Salgado 1-3, E-08034, Barcelona, Spain. This research was supported by the spanish CICYT project TIN-2004-07925 (GRAMMARS). The first author was partially supported by the *Distinció per a la Promoció de la Recerca de la GC, 2002.* Emails: {diaz,sedthilk}@lsi.upc.edu of the set $\mathcal{U} = \{\mathsf{e}\text{-out}, \mathsf{e}\text{-in}, \mathsf{v}\text{-out}, \mathsf{v}\text{-in}\}$. Given $\mathcal{E} \subseteq \mathcal{U}$, we denote $\mathcal{E}$-**dist**(G, H) as the *edit distance of G and H with respect to $\mathcal{E}$*, which is the minimum number of operations in the set $\mathcal{E}$ that when applied to G can transform it to H.

Our study adopts the point of view of parameterized complexity introduced by Downey and Fellows (see [2]). We consider parameterizations of hard problems, i.e. problems whose input contains some (in general small) parameter k and a some main part. A parameterized problem belongs in the class FPT if it can be solved by an algorithm of time complexity $g(k) \cdot n^{O(1)}$ where g is a super-polynomial function of k and n is the size of the problem (we call such an algorithm *FPT-algorithm*). A popular technique on the design of parameterized algorithms is *kernelization*. Briefly, this technique, consists in finding a polynomial time reduction of a parameterized problem to itself in a way that the sizes of the instances of the new problem, we call it *kernel* depend *only* on the parameter k. The function that bounds the size of the main part of the reduced problem determines the *size* of the kernel and is usually of polynomial (on k) size. Clearly, a parameterized problem admits a reduction to a problem kernel, then it is in FPT because any brute force algorithm can solve the reduced problem in time that does not depend on the main part of the original problem. Notice also that this technique provides FPT-algorithms of time complexity $g(k) + n^{O(1)}$ where the non-polynomial part $g(k)$ is just additive to the overall complexity.

The first result of this paper is the classification in FPT of the following problem by giving a kernel of size $O(k^4)$ or $O(k^3)$ depending on the size of the grid we are looking for (see Section 3)

k-ALMOST GRID
Input: A graph G, two positive integers p, q, a non-negative integer k, and a set $\mathcal{E} \subseteq \mathcal{U}$.
Parameter: A non-negative integer k.
Question: Can G be transformed to a (p, q)-grid after at most k edit operations from $\mathcal{E}$?

Notice that the non-parameterized version of k-ALMOST GRID is NP-complete, as the problem with $q = 1$ and with $\mathcal{E}$ consisting only in the *erasing a vertex* operation, is equivalent to the LONGEST PATH problem, i.e. the problem of given a graph G and a constant $k \geq 0$ decide if G contains a simple path of length at least k (we just set $k \leftarrow |V(G)| - k$). Therefore, our parameterization also includes the "dual" parameterization of the LONGEST PATH, in the sense that now the parameter is not the length of the path but the number of vertices in G that are absent in such a path. The "primal" parameterization of the LONGEST PATH problem was known to be in FTP[1] [6, 1].

[1]In an abuse of notation, we indistinctly refer to FPT problem or to problem in the FPT class

However, no result was known, so far, for he dual parameteriztion of LONGEST PATH.

In Section 4 we consider the following more general problem:

k-ALMOST SOME GRID
Input: A graph G and a set $\mathcal{E} \subseteq \mathcal{U}$.
Parameter: A non-negative integer k.
Question: Decide if there exist some pair p, q such that $\mathcal{G}\text{-}\mathbf{dist}(G, H_{p,q}) \leq k$.

Clearly, the above problem can be solved applying the algorithm for k-ALMOST GRID for all possible values of p and q. This implies an algorithm of time complexity $O(\log n(g(k) + n^{O(1)}))$. In Section 4, we explain how to avoid this $O(\log n)$ overhead and prove that there exists also a time $O(g(k) + n^{O(1)})$ algorithm for the k-ALMOST SOME GRID problem. That way, both of our algorithms can be seen as pre-processing algorithms that reduce the size of the problem input to a function depending *only* on the parameter k and not on the main part of the problem.

A different but somehow related sets of problems, which have received plenty of attention is the following: Given a graph G and a property Π, decide what is the minimum number of edges (nodes) that must be removed to obtain a subgraph of G with property Π. In general all these problems are NP-complete [9, 4, 7]. Some of those problems have been studied from the parameterized point of view [5], however, all these problems have the characteristic that the property Π must be an hereditary property. We stress that the k-ALMOST GRID and k-ALMOST ANY GRID problems completely different nature, as the property of *containing a grid* is not hereditary.

2 Definitions and basic results

All graphs we consider are undirected, loop-less and without multiple edges. Define the *neighbourhood* of a vertex $v \in V(G)$ as $N_G(v) = \{u \in V(G) \,|\, (u, v) \in E(G)\}$. Let $\Delta(G) = \max\{|N_G(v)| \mid v \in V(G)\}$ (i.e. $\Delta(G)$ is the maximum degree of a vertex in a graph).

We call (p, q)-grid any graph $H_{p,q}$ that is isomorphic to a graph with vertex set $\{0, \ldots, p-1\} \times \{0, \ldots, q-1\}$ and edge set

$$\{\{(i, j), (k, l)\} \mid (i, j), (k, l) \in V(H) \text{ and } |i-k| + |j-l| = 1\}.$$

The r-border of $H_{s,r}$ is defined as

$$B^{(r)}(H_{s,r}) = ((i, 0), (i, r-1) \mid i = 1, \ldots, r-1).$$

We call a vertex (edge) of $H_{s,r}$ *internal* if it is not in (none of its endpoints is in) $B^{(r)}(H_{s,r})$.

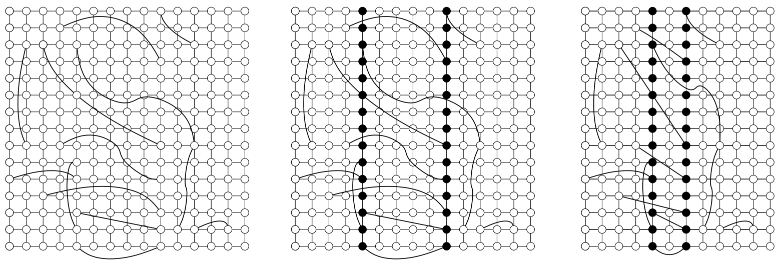
Figure 1: A graph G where $\mathbf{dist}(G, H_{15,15}) \leq 13$ with a $(6, 15)$-band J in it and the result of a 3-contraction of J in G.

All the algorithms and results in this paper will be for the general case where $\mathcal{E} = \mathcal{U}$. The other cases are straightforward simplifications of our results[2]. Subsequently, we will also drop the $\mathcal{E}$ part in the $\mathcal{E}$-$\mathbf{dist}(G, H)$ notation. If $\mathbf{dist}(G, H) \leq k$ we will denote as $V_{\text{dead}}/E_{\text{dead}}$ the vertices/edge that should be removed and $V_{\text{add}}/E_{\text{add}}$ the vertices/edges that should be added towards transforming G to H. Without loss of generality, we assume that the transformation procedure gives priority first to vertex removals then to edge removals, then to vertex insertions, and finally to edge insertions. We also use the notation $k_1 = |V_{\text{dead}}|$, $k_2 = |E_{\text{dead}}|$, $K_3 = |V_{\text{add}}|$ and $k_4 = |E_{\text{add}}|$. Finally, we observe that $k_1 + k_2 + k_3 + k_4 \leq k$.

We say that a graph G contains an *r-band* J of *width* s if the following three conditions are satisfied:

a) G contains $H_{s,r}$ as induced subgraph.

b) There is no edge $\{x, y\} \in E(G)$ such that $x \in V(G) - V(H)$ and $y \in V(H) - B^{(r)}(J)$.

If G is a graph and $H_{p,q}$ is a (p, q)-band in G where $p \geq 3$ and $q \geq 1$, the result of the *c-contraction of $H_{p,q}$ in G* where $0 \leq c \leq p - 3$, is the graph obtained if we contract in G all the edges of $H_{p,q}$ that are in the set $\{\{(i, m), (i + 1, m)\} \mid 2 \leq i \leq c + 1, 1 \leq m \leq q\}$. To denote the result of the operation we just described, we use the notation $\mathsf{contract}(G, H_{p,q}, c)$. Notice that routine $\mathsf{contract}(G, H_{p,q}, c)$ runs $O(cq)$ in steps. For an example of a c-contraction, see Figure 1.

Lemma 1 *Let G be a graph containing a $(3 + c, q)$-band $H_{3+c,q}$ for some $c > \frac{k}{q}$, and let $G' = \mathsf{contract}(G, H_{p,q}, c)$. Then $\mathbf{dist}(G, H_{p,q}) \leq k$ iff $\mathbf{dist}(G', H_{p-c}) \leq k$.*

Proof: $\square$

[2]Our results also hold for more general sets of operations given that they locally change the structure of the input graph.

We call an edge of a graph q-*extreme* if $q > 1$ and both its endpoints have degree 3 or $p = 1$ and both its endpoints have degree 2. For the set of q-extreme edges of a graph G we use the notation $E^{(q)}_{\text{ext}}(G)$.

```
find-edge-max-band($G, q, e$)
**Input:** A graph $G$, a positive integer $q$ and a $q$-extreme edge $e$ of $G$.
**Output:** Return a maximal length $(3+c, q)$-band $H_{3+c,q}$ where $c \geq 0$ and $e$ is also an
extreme edge of $H$. If such a $(3+c, q)$-band does not exists, then return "NO".

1 let $e = \{y_{1,0}, z_{1,0}\}$ and set $L \leftarrow \{y_{1,0}\}$, $R \leftarrow \{z_{1,0}\}$, $i \leftarrow 2$
2 while $i \leq q$ and there are vertices $y_{i,0}$ and $z_{i,0}$
        such that $\{y_{i,0}, y_{i,0}\}, \{z_{i,0}, z_{i,0}\}, \{y_{i,0}, z_{i,0}\} \subseteq E(G)$
        $(d(y_{i,,0}) = d(z_{i,0}) = 4$ and $i < q)$ or $(d(y_{i,0}) = d(z_{i,0}) = 3$ and $i = q)$
                                         or $(d(y_{i,0}) = d(z_{i,0}) = 1$ and $q = 1)$,
        and $N(y_{i,0}) \cap (L \cup R - \{y_{i,0}\} = N(z_{i,0}) \cap (L \cup R - \{z_{i,0}\} = \emptyset$, do
        set $L \leftarrow L \cup \{y_{i,0}\}$, $R \leftarrow R \cup \{z_{i,0}\}$, and $i \leftarrow i + 1$,
3 if $i = q + 1$, then do
        begin
        set $r \leftarrow 1$,
        for $i = 1, \ldots, q$, let $\{z_{i,r}\} = N(z_{i,r-1}) - R$
        while $N(z_{i,r}) \cap (L \cup R) = \{z_{i,r-1}\}$, $\forall_{1 \leq i < j \leq q}\ \{z_{i,r}, z_{j,r}\} \in E(G) \Leftrightarrow j = i + 1$,
            and $\forall_{1 \leq i \leq q}\ (d(z_{i,r}) = 4$ and $1 < i < q)$ or $((d(z_{i,r}) = 3$ and $i \in \{1, q\})$
                                              or $((d(z_{i,r}) = 2$ and $q = 1)$, do
            $R \leftarrow R \cup \{z_{1,r}, \ldots, z_{q,r}\}$; $r \leftarrow r + 1$; for $i = 1, \ldots, q$, $\{z_{i,r}\} \leftarrow N(z_{i,r-1}) - R$
        if $N(z_{i,r}) \cap R = \{z_{i,r-1}\}$ and $\forall_{1 \leq i < j \leq q}\ \{z_{i,r}, z_{j,z}\} \in E(G) \Leftrightarrow j = i + 1$, then
          $R \leftarrow R \cup \{z_{1,r}, \ldots, z_{q,r}\}$, otherwise $r \leftarrow r - 1$
        set $l \leftarrow 1$,
        for $i = 1, \ldots, q$, let $\{y_{i,l}\} = N(y_{i,l-1}) - L$
        while $N(y_{i,l}) \cap (L \cup R) = \{y_{i,l-1}\}$, $\forall_{1 \leq i < j \leq q}\ \{y_{i,l}, y_{j,l}\} \in E(G) \Leftrightarrow j = i + 1$,
            and $\forall_{1 \leq i \leq q}\ (d(y_{i,l}) = 4$ and $1 < i < q)$ or $((d(y_{i,l}) = 3$ and $i \in \{1, q\})$, do
            $L \leftarrow L \cup \{y_{1,l}, \ldots, y_{q,l}\}$; $l \leftarrow l + 1$; for $i = 1, \ldots, q$, $\{y_{i,l}\} \leftarrow N(y_{i,l-1}) - L$
        if $N(y_{i,l}) \cap L = \{y_{i,l-1}\}$ and $\forall_{1 \leq i < j \leq q}\ \{y_{i,l}, y_{j,l}\} \in E(G) \Leftrightarrow j = i + 1$, then
          $L \leftarrow L \cup \{y_{1,l}, \ldots, y_{q,l}\}$, otherwise $l \leftarrow l - 1$
        if $r + l > 0$ then return $G[R \cup L]$
        end
4 return "NO".
```

Lemma 2 *The algorithm* find-edge-max-band(G, q, e) *is correct. If the answer is "*NO*" the algorithm finishes in $O(q)$ steps. If the answer is a vertex set with $(3 + c)q$ vertices, then the algorithm finishes in $O(qc)$ steps.*

Proof: □

Lemma 3 *If G is a graph where $\Delta(G) \leq d$ and* $\mathbf{dist}(G, H_{p,q}) \leq k$ *where $p \geq q$, then G contains at most $(p-2)+(q-2)+2dk$ q-extremal edges.*

Proof: Suppose that G contains more than $(p-2)+(q-2)+2dk$ q-extremal edges. Notice first that the removal of the vertices in V_{dead} can decrease the size of this set by at most $2dk_1$. Then the removal of the edges of E_{dead} can further decrease it by at most $5k_2$. The addition of V_{add} does not decrease this number and finally, the addition of E_{add} can decrease it by $2k_4$. As $k_1+k_2+k_4 \leq k$, we have that H will have more than $(p-2)+(q+2)$ extremal edges, a contradiction as H contains exactly $(p-2)+(q-2)$ q-extremal edges. □

```
find-max-band($G, q, c$)
**Input:** A graph $G$, and two positive integers $q, c$
**Output:** Return a maximal length $(3+c', q)$-band $H_{3+c,q}$ where $c' \geq c$. If such a $(3+c', q)$-band does not exists, then return “NO”.

1 Let $E' \leftarrow E^{(q)}_{\text{ext}}(G)$
2 for while $E' \neq \emptyset$ do
    begin
    let $e$ be an edge in $E'$.
    if find-edge-max-band($G, q, e$) $= H_{3+c',q}$ then
      if $c' \geq c$, then return $H_{3+c',q}$, otherwise,
        set $E' \leftarrow E' - E^*$ where $E^*$ are the $q$-internal extreme edges of$H_{3+c',q}$,
      otherwise set $E' \leftarrow E' - \{e\}$
    end
3 return “NO”
```

Lemma 4 *Algorithm* find-edge-max-band(G, q) *is correct. In case of negative answer it runs in $O(q|E^q_{\text{ext}}(G)|)$ steps. In case of a positive answer it runs in $O(qc|E^q_{\text{ext}}|)$ steps.*

Proof: □

Suppose that G can be transformed to H after a sequence of edit operations. We call a vertex $v \in G$ *safe* if is not removed by a vertex removal operation in this sequence. If a vertex of G is not live then we call it *dead*. We also say that an safe vertex v is a *dirty* vertex if some of the edit operations alters the set of edges incident to v in G (i.e. either adds or removes some edge incident to v). If a safe vertex is not dirty, when we call it *clean*. Finally, we call a vertex of H that is not a vertex of G (i.e. was introduced during some vertex insertion operation) *new*. Clearly, if $\mathbf{dist}(G,H) \leq k$, then there are at most k new vertices in H and at most k dead vertices in G and this implies the following lemma.

Lemma 5 *Let G and H be two graphs where $\mathbf{dist}(G,H) \leq k$ and such that the transformation of G to H involves k_1 vertex removals and k_2 vertex additions. Then, $|V(H)| - k_2 \leq |V(G)| \leq |V(H)| + k_1$.*

Proof: □

Lemma 6 *Let G be a graph such that $\mathbf{dist}(G, H_{p,q}) \leq k$. Supose also that G does not contain any $(3+c,q)$-band where $q \cdot c > k$ and $H_{p,q}$ contains $\leq d$ dirty vertices. Then $p \leq (\lceil \frac{k}{q} \rceil + 3)(d+k+1)$.*

Proof: Assume that $H_{p,q}$ contains d dirty vertices and at most $k_3 = |V_{\text{in}}|$ new vertices. Let $l = \lceil \frac{k}{q} \rceil$. Notice that $H_{p,q}$ contains $p-(l+3)-1$ distinct (but possibly overlapping) $(l+3,q)$-bands. We denote this collection as $\mathcal{B}$. A new vertex can be in columns at most $l+3$ members of $\mathcal{B}$ and a dirty vertex can be in columns of at most $l+1$ members of $\mathcal{B}$. If H contains a member of $\mathcal{B}$ that does not contain columns with new vertices or interior columns with dirty vertices, then $p-(l+3)+1-k_3(l+3)-d(l+1) > 0$. Therefore $p \leq l+2+k_3(l+3)+d(l+1) \leq l+2+k(l+3)+d(l+1)$. □

Lemma 7 *Let G be a graph with a vertex v of degree more than $k+4$ in G and let $H_{p,q}$ be a grid. Then any way to transform G to $H_{p,q}$ should involve the operation of the removal of v. In particular, $\mathbf{dist}(G,H_{p,q}) \leq k$ iff $\mathbf{dist}(G[V(G)-v],H) \leq k-1$.*

Proof: □

Lemma 8 *Let G be a graph where $\Delta(G) \leq b$ and let H be some grid. If $\mathbf{dist}(G,H) \leq k$ then H will contain at most $\max\{b,2\} \cdot k$ dirty vertices.*

Proof: If v is a dirty vertex of H this means that it is either adjacent to a vertex in V_{dead} or is incident to an edge in $E_{\text{add}} \cup E_{\text{dead}}$. Notice that each vertex in X_{dead} is can be adjacent to at most b dirty vertices of H. Also an edge in $E_{\text{dead}} \cup E_{\text{dead}}$ can be incident to at most two dirty vertices of H. Therefore, each edit operation creates at most $\max\{2,b\}$ dirty vertices and the result follows. □

3 Looking for a given grid $H_{p,q}$

In this section, we show that the k-Almost grid problem is in FPT. The next lemma introduces the function f to bound the size of the kernel.

Lemma 9 *Let G be a graph where* $\mathbf{dist}(G, H_{p,q}) \leq k$. *suppose also that* $\Delta(G) \leq b$ *and that G does not contain any $(3+c, q)$-band where $c > k/q$ or any $(3+c, p)$-band where $c > k/p$. Then $|V(G)| \leq f(k, b, p, q)$ where*

$$f(k,b,p,q) = \begin{cases} k + k^2 & \text{if } p \leq k \text{ and } q \leq k \\ k + 5k(b \cdot k + k + 1) & \text{if } \min\{p,q\} \leq k < \max\{p,q\} \\ k + 16(b \cdot k + k + 1)^2 & \text{if } p > k \text{ and } q > k \end{cases}$$

Proof: From Lemma 5 we get that $|V(G)| \leq p \cdot q + k$. From Lemma 8, the dirty vertices of H will be at most $b \cdot k$. From Lemma 6 we get that $p \leq (\lceil \frac{k}{q} \rceil + 3)(b \cdot k + k + 1)$ and $q \leq (\lceil \frac{k}{q} \rceil + 3)(b \cdot k + k + 1)$. If $p \leq k$ and $q \leq k$ then clearly $|V(G) \leq k + k^2$. If $q \leq k$ then $p \leq (\lceil \frac{k}{q} \rceil + 3)(b \cdot k + k + 1) \leq (\frac{k}{q} + 4)(b \cdot k + k + 1)$ and therefore $|V(G)| \leq k + p \cdot q \leq k + q(\frac{k}{q} + 4)(b \cdot k + k + 1) \leq k + 5k(b \cdot k + k + 1)$. The symmetric analysis works when $p \leq k$. If now $p, q > k$ then $\lceil \frac{k}{q} \rceil = \lceil \frac{k}{p} \rceil = 1$ and therefore $p, q \leq 4(b \cdot k + k + 1)$. We conclude that $|V(G)| \leq k + 16(b \cdot k + k + 1)^2$. □

```
Kernel-for-Check-Grid$(G, p, q, k)$
**Input:** A graph $G$, two positive integers $p, q$, and a non-negative integer $k$
**Output:** Either returns "NO", meaning that that $\mathbf{dist}(G, H_{p,q}) > k$, or returns a graph $G'$ and a triple $p', q', k'$ such that $\mathbf{dist}(G, H_{p,q}) \leq k$ iff $\mathbf{dist}(G', H_{p',q'}) \leq k'$.

0. Set $k' = k$, $G' = G$, $p' = p$ and $q' = q$.
1. As long as $G'$ has a vertex of degree $\geq k' + 4$ remove it from $G'$ and set $k' \leftarrow k' - 1$.
   If after the end of this proccess $k' < 0$ then return "NO".
2. Apply any of the following procedures as long as this is possible:
   • As long as find-max-band$(G', q', \lceil \frac{k'}{q'} \rceil + 1) = H_{3+c,q'}$, then $G' =$ contract$(G', H_{3+c,q'}, c)$
     and set $p' \leftarrow p' - c$.
   • As long as find-max-band$(G', p', \lceil \frac{k'}{p'} \rceil + 1) = H_{3+c,p'}$, then $G' =$ contract$(G', H_{3+c,p'}, c)$
     and set $q' \leftarrow q' - c$.
3 If $|V(G')| > f(k', k + 4, p', q')$ then return "NO".
4. return $G', p', q', k'$.
```

Theorem 1 *Algorithm* Kernel-for-Check-Grid(G, p, q, k) *is correct and if it outputs the graph G' then $|V(G')| \leq f(k, k+4, p, q)$. Moreover, it runs in $O(pq(p + q + k^2))$ steps.*

Proof: Step **1** is justified by Lemma 7. Notice that before the algorithm enters Step **2**, $\Delta(G) \leq k + 4$. Step **2** creates equivalent instances of the problem because of Lemma 1. From Lemma 4, Each time a contraction of step **2** is applied, the corresponding q-band (p-band) requires $O(q'c(p' + q' + k'^2))$ $(O(p'c(p' + q' + k'^2)))$ steps. As the the sum of the lengths of the bands cannot exceed $q(p)$ we obtain that step **2** requires, in total, $O(pq(p+q+k'^2))$ steps. Moreover, before the check of Step **3**, all the conditions of

Lemma 9 are satisfied and therefore, if the algorithm returns G', p', q', k', then $|V(G')| \leq f(k', k+4, p', q') \leq f(k, k+4, p, q)$. □

```
Check-Grid$(G, p, q, k)$
**Input:** A graph $G$, two positive integers $p, q$, and a non-negative integer $k$
**Output:** Either returns "NO", meaning that that $\mathbf{dist}(G, H_{p,q}) > k$, or
returns a sequence of at most $k$ operations that transforms $G$ to $H_{p,q}$.

1. for any set $V_{\text{dead}} \subseteq V(G)$ of at most $k$ vertices do
      if $|E(G[V(G) - V_{\text{dead}}])| \leq 2pq - p - q + k$ then
        for any set $E_{\text{dead}}$ of at most $k - |V_{\text{dead}}|$ edges of $E(G[V_{\text{dead}}])$ do
            for any $i = 0, \ldots, k - |V_{\text{dead}}| - |E_{\text{dead}}|$ do
                begin
                let $G^*$ be the graph obtained from $G$ after the
                    removal of the vertices in $V_{\text{dead}}$ and the
                    edges in $E_{\text{dead}}$ and the addition of a set $V_{\text{add}}$ of $i$ new vertices.
                let $S$ be the set of vertices in $G^*$ with degree less than 4.
                if $|S| \leq 2p + 2q - 4 + 2k$ then do
                  for any $E_{\text{add}} \in S \times S$ where $|E_{\text{add}}| \leq k - |V_{\text{dead}}| - |E_{\text{dead}}| - |V_{\text{add}}|$ do
                      if $(V(G^*), E(G^*) \cup E_{\text{add}})$ is isomorphic to $H_{p,q}$ then return
                          a sequence containing the sequence of operations
                          defined by the sets $V_{\text{dead}}$, $E_{\text{dead}}$, $V_{\text{add}}$, and $E_{\text{add}}$
                end
2. return "NO"
```

Theorem 2 *Algorithm* Check-Grid(G, p, q, k) *is correct and runs in* $O(16^k \cdot (2n + 3k)^{2k})$ *steps.*

Proof: we recall the notation $k_1 = |V_{\text{dead}}$, $k_2 = |E_{\text{dead}}|$, $K_3 = |V_{\text{add}}|$ and $k_4 = |E_{\text{add}}|$. The first "for" of the algorithm guesses the set of vertices V_{dead} that should be removed while transforming G to $H_{p,q}$. Clearly, there are $\binom{n}{k} = O(n^{k_1})$ ways to make this guess. From Lemma 5 these guesses are at most $(pq + k_1)^{k_1}$. If this is a correct guess, then the remaining graph will be a subgraph of $H_{p,q}$ with at most k edges more. As $H_{p,q}$ has $(p-1)q + (q-1)p$ edges, then $G[V(G) - V_{\text{dead}}]$ will have at most $2pq - p - q + k_2$ edges and the filter of the algorithm before the second "for" is correct. The second "for" guesses the set E_{dead} of edges that should be removed and there are at most $\binom{2pq-p-q+k_2}{k_2} \leq (2pq + k_2)^{k_2}$ ways to make such a guess. Then comes the turn to guess the set of vertices V_{add} that should be added towards constructing $H_{p,q}$. As we cannot add more than k_3 vertices, the third "for" makes at most k_3 calls one for each possible quantity of vertices in V_{add}. Notice that if all the guesses done so far are correct, G^* is a subgraph of $H_{p,q}$ with $\leq k_4$ edges less than $H_{p,q}$. As $H_{p,q}$ has $2(p-1)+2(q-1)$ vertices of degree less than 4 $V(G^*)$ should contain a subset S of at most $2p+2q-4+2k_4$ vertices of degree less than 4 (an missing edge can harm the degree of at most 2 vertices). Clearly, the edges to add in G^* towards constructing $H_{p,q}$ will have endpoints in S and therefore each choice of an edge to add will be done by a set of $\binom{2p+2q-4+2k_4}{2} \leq 4(p+q+k_4)^2$ candidate edges. As there are at most k_4 guesses of edges to be done in the fourth "for" we finally have at most $\binom{4(p+q+k_4)^2}{k_4} \leq (4(p+q+k_4))^{2k_4}$ edge sets to guess. After than, we can verify whether the resulting graph is isomorphic to $H_{p,q}$, which can be done in $O(pq)$ steps. Resuming, the total running time is $O((pq+k_1)^{k_1} \cdot (2pq+k_2)^{k_2} \cdot k_3 \cdot (4(p+q+k_4))^{2k_4} \cdot pq) = O(16^k \cdot (p+q+k)^{2k})$. As $p+q \leq 2pq$, using Lemma 5 we observe that $(p+q+k) \leq 2pq+k \leq 2(pq-k)+3k \leq 2n+3k$ and we have the required time bound. □

```
Almost-grid$(G, p, q, k)$
**Input:** A graph $G$, two positive integers $p, q$, and a non-negative integer $k$
**Output:** Either returns "NO", meaning that that $\mathbf{dist}(G, H_{p,q}) > k$, or
returns a sequence of at most $k$ operations that transforms $G$ to $H_{p,q}$.

1. If Kernel-for-Check-Grid$(G, p, q, k) = (G', p', q', k')$ then return Check-Grid$(G, p, q, k)$
2. return "NO"
```

Theorem 3 *Algortihm* Almost-grid(G, p, q, k) *solves the* k-Almost-Grid *problem in* $2^{O(k \log k)} + O(n^3)$ *steps.*

Proof: The algorithm we first calls the algorithm Kernel-for-Check-Grid(G, p, q, k) that gives some answer in $O(pq(p+q+k^2)) = O(n^3)$ steps. If the answer is "NO" then returns that $\mathbf{dist}(G, H_{p,q}) > k$. If the algorithm returns (G', p', q', k') then solve k-Almost-Grid with input G', p', q', k' using brute force and return the corresponding answer. From Theorem 1 we have that $|V(G')| \leq k + 16(k^2+5k+1)^2 = O(k^4)$. From Theorem 2, Check-Grid(G, p, q, k) requires $(O(k))^{8k}$ steps. □

4 Looking for any grid

For $\alpha, \beta, \gamma \in \mathbb{N}$, we define $\mathbf{prods}(\alpha, \beta, \gamma) = \{(p, q) \subseteq \mathbb{N}^2 \mid p \leq \alpha \text{ and } \gamma - \beta \leq p \cdot q \leq \gamma + \beta\}$.

Clearly, to solve the k-Almost some Grid problem it is enough to apply Check-Grid(G,p,q,k) for all $(p,q) \in \mathbf{A}(n,k) = \bigcup_{n-k \le i \le n+k} \mathbf{prods}(i,n,k)$. Clearly, $|\mathbf{A}(n,k) \le 2k^2 \log(k+n)$. Therefore, k-Almost some Grid can be solved after $O(k^2 \log(n+k))$ calls of Almost-grid(G,p,q,k) which gives a running time of $O(\log n)(2^{O(k \log k)}) + O(k^2 n^3 \log n)$ steps. We call this algorithm check-all-cases(G,k). We stress that there are cases where $|\mathbf{A}(n,k)| \ge \frac{1}{2}\log n$ and therefore, that way, we may not avoid the "$\log n$"-overhead. In what remains we will explain an alternative approach that gives running times of the same type as the case of k-Almost Grid.

We call *r-band collection* $\mathcal{C}$ of a graph G a collection of $|\mathcal{C}|$ r-bands in G of widths $s_1, \ldots, s_{|\mathcal{C}|}$ where no pair of them have common interior vertices. The *width* of such a collection is equal to $\sum_{i=1,\ldots,|\mathcal{C}|}(s_i - 2)$.

```
find-max-band-collection$(G,r,w)$
**Input:** A graph $G$ and two positive integers $r$ and $w$
**Output:** Return "YES" if $G$ contains an $r$-band collection of width $\ge w$; otherwise, return "NO".

1 Let $E_{\text{ext}}$ be the set of extremal edges of $G$, $t \leftarrow 0$
2 while $E_{\text{ext}} \ne \emptyset$ and $t < w$ do
        begin
        $t \leftarrow 0$
        Let $e$ be an edge in $E_{\text{ext}}$
        set $E' \leftarrow E_{\text{ext}}$
        while find-edge-max-band$(G,r,e) = H_{3+c,r}$ do
            begin
            set $E' \leftarrow E' - E^*$ where $E^*$ are the extreme edges of $H_{3+c,r}$
            set $t \leftarrow t + c + 1$
            let $e$ be an edge in $E'$
            end
        set $E_{\text{ext}} \leftarrow E_{\text{ext}} - \{e\}$
        end
3 if $t \ge w$ then return "'YES", otherwise return "NO".
```

Lemma 10 *The algorithm* find-max-band-collection(G,r,w) *is correct and runs in* $O(rw|E_{\text{ext}}(G)| + n)$ *steps.*

Proof: □

We first need some conditions on when it is possible to make an estimation of the dimensions of a grid.

Lemma 11 *Let G be a graph where $\Delta(G) \leq b$ and let $H_{p,q}$ be a grid where $\mathbf{dist}(G, H) \leq k$. Then If G does not contain any q-band collection of width $> k$ then $|V(H)| \leq (b \cdot k + 2)^2$.*

Proof: $H_{p,q}$ contains a single element q-band collection $\mathcal{C}_H$ of width $p - 2$. We apply back the $\leq k$ operations that transformed G to H. If we remove an edge in Y_{in}, in the worst case, its endpoints may belong in neighboring columns of $\mathcal{C}_H$ and this can remove at most 2 units from the width of $\mathcal{C}_H$. If we add back an edge in Y_{out}, in the worst case, its endpoints may belong in different columns of $\mathcal{C}_H$ and this can remove at most 4 units from the width of $\mathcal{C}_H$. If finally we add back a vertex in X_{out}, in the worst case, all its neighbours will belong into different columns and this may reduce the width of $\mathcal{C}_H$ by $\leq \Delta(G) \leq b$. Resuming, we have that G contains a q band collection of width at least $p - 2 - e_{\text{in}} - e_{\text{out}} - b \cdot x_{\text{out}} \geq p - 2 - b \cdot k$ which implies that $p \leq b \cdot k + 2$. Working symmetrically, we also conclude that $q \leq b \cdot k + 2$ and the result follows. $\square$

Lemma 12 *Let G be a graph where $\Delta(G) \leq b$ and let H be a grid where $\mathbf{dist}(G, H) \leq k$. Then if G contains an r-band collection of width $> 3k$ then $H = H_{s,r}$ for some $s \in [\frac{n-k}{r}, \frac{n+k}{r}]$. Moreover at most two such r-band collections exist and for each r, s will be unique when $r > 2k$.*

Proof: Let $\mathcal{C}$ be the interior columns in such a collection of bands. We will apply the edit operations that transform G to H. The removal of a vertex in X_{out} can reduce the width of $\mathcal{C}$ by at most 3. The removal of an edge in Y_{out} can reduce the width of $\mathcal{C}$ by at most 3. The addition of an edge in Y_{in} can reduce the width of $\mathcal{C}$ by at most 2. Therefore, in the worst case, H still contains a collection of bands of width $3k + 1 - 3k > 1$. So, there exist at least one r-band of width at least 1 in H and this means that $H = H_{s,r}$ for some $s \geq 1$.

Suppose now that there are three integers $r_1 < r_2 < r_3$ and that G contains an r_i-band collection of width $> 3k$, for $i = 1, 2, 3$. Let also $I_i, = 1, 2, 3$ be the interior columns of each of these band collections. Clearly, for any $i < j, I_i \cap I_j \neq \emptyset$. According to the previous analysis, after the edit operations, three, different, size r_i-bands of width al least 1 will survive which is impossible to exist in a rectangular grid.

Notice now that $s \cdot r = n - x_{\text{out}} + x_{\text{in}}$. Let $\delta = |x_{\text{out}} - x_{\text{in}}|$. We obtain that $s \cdot r \in [n - \delta, n + \delta]$ which implies that $s \in [\frac{n-k}{r}, \frac{n+k}{r}]$, as $\delta \leq k$. If now the interval $[\frac{n-k}{r}, \frac{n+k}{r}]$ contains two integers, then $\frac{n+k}{r} - \frac{n-k}{r} \geq 1$ which implies that $r \leq 2k$. $\square$

```
Check-some-Grid$(G, k)$

Input: A graph $G$and a non-negative integer $k$
Output: The answer to the $k$-ALMOST-SOME-GRID problem with instance $G$ and parameter $k$.
1. As long as $G$ has a vertex of degree $\geq k+4$ remove it from $G$ and
        set $k \leftarrow k-1$. If after the end of this proccess $k<0$ then return "NO".
2. $R \leftarrow \emptyset$
3. for $i=1,\ldots,n$, if find-max-band-collection$(G, i, 3k+1) =$"YES", then set $R \leftarrow R \cup \{i\}$
4. If $|R| = 0$ then if $V(G) > k + ((k+4)k+2)^2$ then return "NO", otherwise, goto Step 7
5. For any $r \in R$,
   If $r > 2k$, then
     begin
     If $\mathbb{N} \cap [\frac{n-k}{r}, \frac{n+k}{r}] = \emptyset$, then return "NO", otherwise,
       return Almost-Grid$(G, r, s, k)$ where $\{s\} = \mathbb{N} \cap [\frac{n-k}{r}, \frac{n+k}{r}]$.
     end
   otherwise, for $(i, j) \in$ prods$(2k, n, k)$,
     begin
     If Almost-Grid$(G, i, j, k) =$ YES, then return Almost-Grid$(G, i, j, k)$.
     end
6. Return "NO".
7. check-all-cases$(G, k)$.
```

Theorem 4 *Algorithm* Check-some-Grid(G, k) *is correct and the* k-ALMOST-ANY-GRID *problem is in* FPT *and can be solved in time* $2^{O(k \log k)} + O(k \log k \cdot n^3)$.

Proof: Step **1** is justified by Lemma 7 and we may assume that before the algorithm enters Step **2**, $\Delta(G) \leq k+4$. Steps **2**–**4** are based on Lemma 11. Step **3** involves $O(n)$ calls of find-max-band-collection$(G, i, 3k+1)$ which requires $O(kn^2)$ steps because of Lemma 10. Theorefore, Step **3** requires $O(kn^3)$ steps. Finally the loop in Step **5** is correct because of Lemma 12. Note that the first loop in step **5** is applied at most 2 times and that $|\mathbf{prods}(2k, n, k)| = O(k \cdot \log k)$. Therefore, step **5** runs in $2^{O(k \log k)} + O(k \log k \cdot n^3)$ steps. As we noticed in the begining of this section check-all-cases(G, k) requires $O(\log n')(2^{O(k \log k)} + O(\log n' \cdot n'^3 k^2))$ steps where n' is the number of vertices of G in step **7** where $n' \leq O(k^4)$ and this means that step **7** requires $2^{O(k \log k)}$ steps. □

References

[1] N, Alon, R. Yuster and U. Zwick. Color-Coding. *Journal of the ACM*, 42(4), 844–856, 1995.

[2] G. Downey and M. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.

[3] H. Kaplan, R. Shamir and R. Tarjan.. Tractability of parameterized completion problems on chordal, strongly chordal and proper interval graphs. *SIAM Journal of Computing*, 28, 1906–1922, 1999.

[4] J. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-Complete. *Journal Comput. and Systems Sci.* 20(2), 219–230, 1980.

[5] Leizhen Cai. Fixed-Parameter Tractability of Graph Modification Problems for Hereditary Properties. *Information Processing Letters*, 58(4)171–176, 1996

[6] B. Monien. How to find paths efficiently. *Annals of Discrete Mathematics*, 25, 239–254, 1985.

[7] A. Natanzon, R. Shamir and R. Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113(1), 109–128, 2001.

[8] T. Sebastian, P. Klein, and B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Matching and Machine Intelligence*, 26, 550–571, 2004.

[9] M. Yannakakis. Node and Edge Deletion NP-complete problems. *ACM Symposium on Theory of Computing (STOC)*, 253–264, 1978.

[10] K. Zhang and D. Sasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing* 18, 1245–1262, 1989.