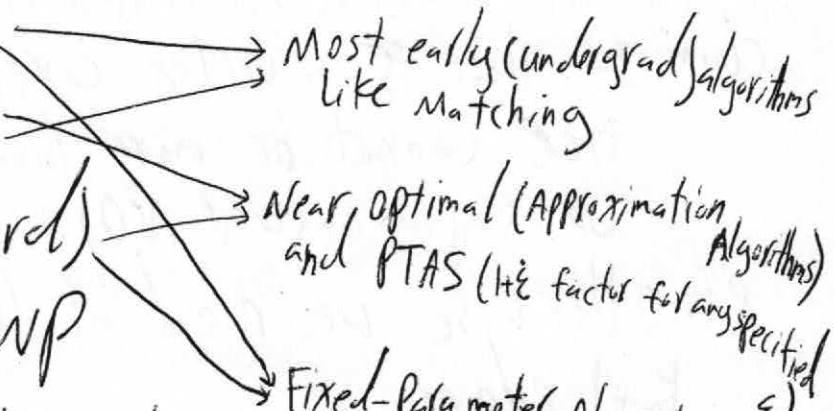
Fix Parameter algorithms and lower bounds for Parameterized Problems

Algorithmic Goals
1) Correct (optimal solution)
2) Fast (polynomial time)
3) Hard problems (NP-hard)
Pick any two (assumming $P \neq NP$)



In FPT algorithms, we confine exponential growth to Parameter k other than (and smaller than) input size n

Bad: n^k (e.g. for clique: the largest set of vertices with a complete induced graph)

Good: $2^k n$ or in general $f(k)n$ (e.g. for vertex cover: minimum # of vertices covering ALL edges)

Great: $2^{O(\sqrt{k})} n^{O(1)}$ - subexponential (e.g. for vertex cover in planar graphs)

Graphs which can be drawn on the plane with no crossing

In FPT, we also consider the concept of Kernelization:
i.e., $n^{O(1)}$ time Preprocessing algorithm reducing a problem from (size-parameter) = (n,k) down to $\leq (f(k), g(k))$

Having a Kernel means $n^{O(1)} + f'(k)$ fixed parameter algorithms.
However it is proved that Kernel exists if and only if the parameter if fixed-parameter tractable.

In general $f(k)$ can be exponential in k but a good kernel should be polynomial (i.e., $f(k) = k^{O(1)}$) or even linear (i.e. $f(k) = O(k)$)

FPT Algorithmic Techniques include for example:
- Kernelization
- Bounded search trees
- Color Coding
- Treewidth
- Iterative compression
- Graph minors
- Bidimensionality
- ...

A simple FPT example for vertex cover: Is there a vertex cover of size k in Graph G?

- Consider one edge and choose one of the endpoints; delete the edge (and its two vertices) and repeat;
- For each selected set check whether it is vertex cover

Claim: If there is a vertex cover of size k the depth of search tree cannot be more than k (i.e., if we go to level k+1 we simply answer NO)

Pf: otherwise we need at least k+1 vertices to cover k+1 edges

The running time of the above algorithm is $O(2^k n)$ ($O(3^k n)$ but can be improved to), though the best vertex cover FPT algorithm can be run in time $O(1.2738^k + kn)$ due to [chen, kanj, Xia'13] in TCS Journal

A simple kernalization for vertex cover:

- First add any vertex of degree more than k+1 to the vertex cover (otherwise we need k+1 vertices to cover its neighboring edges).
- Now find a Maximal (at least) Matching (a set of edges without common vertices which is maximal, i.e., no edges can be added to it)
- Note that the size of the maximal matching cannot be more than k (otherwise we need at least k+1 vertices to cover this edges).
- Also each vertex should be neighbor of at least one of the vertices of the maximal matching (By definition). It means we have a kernel of size (vertex/edge) $O(k^2)$. However the currently best known kernelization algorithm in terms of the number of vertices is due to Lampis (2011) and achieves 2k - clog k vertices for any fixed constant c.

We cannot have a kernel of size O(log k) since then P=NP. Indeed unless (coNP $\subseteq$ NP/poly which is unlikely) we cannot have kernel of size $O(k^{2-\varepsilon})$ edges however vertex size $(2-\varepsilon)k$ for $\varepsilon > 0$ is still OPEN.

FPT hardness or Parameterized Complexity

To build a complexity theory for parameterized problems, we need two things:

- An appropriate notion of reduction (polynomial-time reductions are not good for us, See Example 1)
- An appropriate hypothesis

Example 1: Graph G has an independent set of size k (a set k of vertices with no edges in between) if and only if it has a vertex cover of size $n-k$. This transforms an Independent set of instance (G,k) into a vertex cover instance $(G,n-k)$ by a polynomial-time reduction. However vertex cover is FPT but independent set is not known to be FPT.

Parameterized reduction from problem P to problem Q:

a function ϕ with the following properties

- $\phi(x)$ can be computed in time $f(k).|x|^{O(1)}$, where k is the parameter of x, the instance.
- $\phi(x)$ is a yes-instance of $Q \iff x$ is a yes-instance of P.
- If k is the parameter of instance x and k' is the parameter of instance $\phi(x)$, then $k' \leq g(k)$ for some function g.

Note that by above definition, if there is a parameterized reduction from problem P to problem Q and Q is FPT, then P is also FPT.

Note that transforming Indepenset set (G,k) into vertex cover instance $(G,n-k)$ is NOT a parameterized reduction though transforming an Independent set instance (G,k) into a clique instance $(\bar{G},k)$ IS a parameterized reduction ($\bar{G}$ is the complement of G).

Multicolored clique:

A very useful variant of clique which helps in simplifying details in FPT reductions by allowing an almost systematic gadget-construction.

Def: The vertices of the input graph G are colored with k colors and we have to find a clique containing one vertex from each color.

Note that indeed multicolored versions exists for almost all natural subset problems (e.g. independent set) and one can show they are as hard as their uncolored counterparts [see Fellows, Hermelin, Rosamond, Vialette '09]

Thm: There is a parameterized reduction from clique to multicolored clique.

Pf: Via a reduction from k-clique. Given an instance (G,k) for k-clique, we construct a graph G' by taking k copies $v_1, \ldots v_k$ of each vertex G and color v_i with color i. We then add an edge in G' between two vertices u_i and $v_j, i \neq j$, iff u and v are connected in G. It is straightforward to verify that G has a k-clique iff G' has a k-multicolored clique.

Now lets see an important reduction:

Thm: There is a parameterized reduction from (multicolored) Independent Set (same as multicolored clique) to Dominating Set (a set of vertices such that each vertex not in the set has a neighbor in the set).

Pf: Let G be a graph with n vertices, m edges and k be an integer. We construct a graph H such that G has a multicolored independent set of size k iff H has a dominating set of size k. (MIS)

Let $[G, k, (V_1, \ldots V_k)]$ be an instance of MIS. we construct graph G' as follows for dominating set

i) for every vertex $v \in V(G)$, we introduce v in G'

ii) For every $1 \leq i \leq k$, we make the set V_i a clique in G'.

iii) for every $1 \leq i \leq k$, we introduce two new vertices x_i, y_i into G' and make them adjacent to every vertex of V_i

iv) for every edge $e \in E(G)$ with endpoints $u \in V_i$ & $v \in V_j$, we introduce a vertex w_e into G' and make it adjacent to every vertex of $(V_i \cup V_j) \setminus \{u, v\}$.

we claim G has an MIS with exactly one vertex from each V_i iff G' has a dominating set of size k. Intuitively vertices x_i and y_i ensure dominating set of size k selects exactly one vertex from each V_i. The vertices we ensure the selected set is MIS in G: if both endpoints of an edge e is selected then vertex w_e is not dominated in G'.

Indeed lots of other parameterized problems are known to be at least as hard as clique such as set cover, Hitting set, connected dominating set, Independent dominating set, Partial vertex cover, etc

Now let's consider basic hypotheses

Exercise: Prove there is a reduction from Dominating set to Connected dominating set & set cover

So far we cannot build parameterized complexity based on $P \neq NP$ and thus we need to have stronger assumption.

Let's consider assumptions below:

[clique Hardness] k-clique (or k-independent set) cannot be solved in time $f(k)\, n^{o(1)}$.

[Exponential-Time Hypothesis (ETH)]: n-variable 3-SAT cannot be solved in time $2^{o(n)}$.

Indeed one can show the second conjecture implies the first one as well. [Chen et al. 2004]

Note that as aforementioned there is a reduction from Independent set to dominating set what about the reverse?

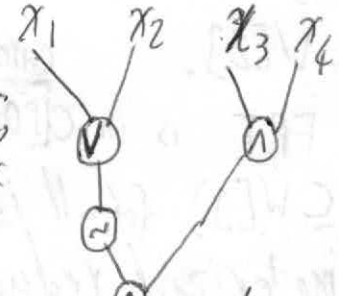
The answer is NO most probably. Indeed unlike NP-completeness where most problems are equivalent, here we have a hierarchy of hardness.

Independent set is W[1]-complete.

but Dominating set is W[2]-complete.

Let's see definition of W[1] and W[2].

Boolean circuit consists of input gates, negation gates, AND gates, OR gates and a simple output gate.

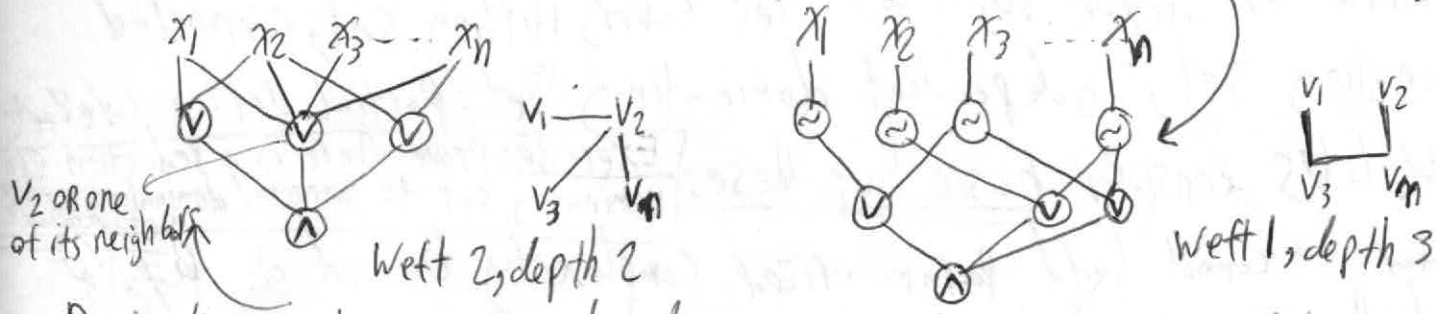


Circuit Satisfiability: Given a Boolean circuit C, decide if there is an assignment on the inputs of C such that the output is true.

Weighted Circuit Satisfiability: Given a Boolean circuit C and an integer k, decide if there is an assignment of weight k such that the output is true, where the weight is the number of true values.

Weighted circuit Satisfiability:

Independent Set can be reduced to Weighted Circuit Satisfiability:



Dominating Set can be reduced to weighted circuit Satisfiability

The <u>depth</u> of a circuit is the maximum length of a path from an input to the output.

A gate is <u>large</u> if it has more than 2 inputs. The <u>weft</u> of a circuit is the maximum number of large gates on a path from an input to the output.

<u>The W-hierarchy:</u> Let $C[t,d]$ be the set of all circuits having weft at most t and depth at most d.

A problem P is in the class $W[t]$ if there is a constant d (that holds for all instances of the problem) and a parameterized reduction from P to Weighted Circuit Satisfiability of $C[d,t]$.

As we saw independent set is in $W[1]$ and Dominating set is in $W[2]$

Indeed we can prove Independent set is $W[1]$-complete and dominating set is $W[2]$-complete as well. (↳ i.e. is in $W[1]$ and as hard as any problem in $W[1]$)

Thus if ang $W[1]$-complete problem is FPT, then $FPT = W[1]$ and every problem in $W[1]$ is FPT.

Similarly if any $W[2]$-complete problem is in $W[1]$, then $W[1] = W[2]$ and thus if there is a parameterized reduction from Dominating set to Independent set, then $W[1] = W[2]$.

Note that FPT is in $C[O(f(k)), 0]$ (corresponding to time) and thus in $W[0]$ (essentially by definition). Indeed $FPT = W[0]$

Also $W[i] \subseteq W[j]$ for all $i \leq j$ and the classes in W hierarchy are also closed under parameterized reduction. We believe $\subseteq$ is indeed C.

Many natural computational problems occupy the lower levels, $W[1]$, $W[2]$

XP is the class of parameterized problems that can be solved in $n^{f(k)}$.

Finally we have $FPT = W[0] \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq XP$

In summary by parameterized reductions, we can show lots of problems are at least as hard as clique or dominating set and thus in W[1] or W[2].

Exponential Time Hypothesis (ETH) introduced by Impagliazzo, Paturi and Zane:

ETH: there is no $2^{o(n)}$-time algorithm for n-variable 3-SAT (the current best bound is 1.30704^n [Hertli 2011].) Note that if SETH holds, ETH holds as well.

→ Note that an n-variable 3SAT can have $\Omega(n^3)$ clauses but Impagliazzo et al show that there is a $2^{o(n)}$-time algorithm for n-variable 3-SAT iff there is a $2^{o(m)}$-time algorithm for m-clause 3-SAT. Thus ETH also says: there is no $2^{o(m)}$-time algorithm for m-clause 3-SAT.

The standard textbook reduction from 3SAT to 3-coloring construct a graph of $O(n+m)$ edges and $O(n+m)$ vertices to solve 3SAT instance of n variables and m-clauses. Thus:

($m = O(n)$, NP-hardness $O(n+m) = O(m)$)

Assuming ETH, there is no $2^{o(n)}$ algorithm for 3-coloring on an n-vertex graph G.

→ since there are standard polynomial-time reduction say from 3-coloring to many other problems such that the reduction increases the number of vertices by at most a constant factor, we have

corollary: Assuming ETH, there is no $2^{o(n)}$ time algorithm on n-vertex graphs for Independent set, clique, dominating set, vertex cover, Hamiltonian Path, feedback vertex set, etc.

similarly since in the problems above $k \leq n$

we have no $2^{o(k)} n^{O(1)}$ algorithm for k-version of the problems above.

(note that $2^{o(n)} n^{O(1)} = 2^{o(n)}$)

Tighter bounds:

As aforementioned ETH implies clique Harness; indeed we can prove a much stronger and more intersting theorem:

Thm [Chen et al. 2004]: Assuming ETH, there is no $f(k).n^{o(k)}$ algorithm for k-clique for any computable function f.

Pf: Assume by contradiction k-clique can be solved in time $f(k).n^{\frac{k}{s(k)}}$, where $s(k)$ is a monotone increasing unbounded function. We use this algorithm to solve 3-coloring on an n-vertex graph G in time $2^{o(n)}$.

Let k be the largest integer such that $f(k) \leq n$ and $k^{\frac{k}{s(k)}} \leq n$. Thus function $k := k(n)$ is monotone increasing and unbounded. Now we split the vertices of G into k groups. Let us build a graph H, where each vertex corresponds to a proper 3-coloring of one of the groups and connect two vertices if they are not conflicting (i.e., having adjacent vertices of same color). Thus every k-clique of H corresponds to a proper 3-coloring of G and thus a 3-coloring of G can be found in time

$$f(k)\,|V(H)|^{\frac{k}{s(k)}} \leq n.\left(k\,3^{\frac{n}{k}}\right)^{\frac{k}{s(k)}} = n.k^{\frac{k}{s(k)}}.3^{\frac{n}{s(k)}} \leq n^2.3^{\frac{n}{s(k)}} = 2^{o(n)}$$

since $f(k) \leq n$ & partition into k groups

since $k := k(n)$ and $s(k)$ are monotone increasing & unbounded

It is easy to see that if we have a reduction from k-clique to Problem A instance $(x', g(k))$ then $f(k)\,n^{o(g^{-1}(k))}$ algorithm for $A_{(x,k)}$ implies $f(k)n^{o(k)}$ algorithm for k-clique. Thus

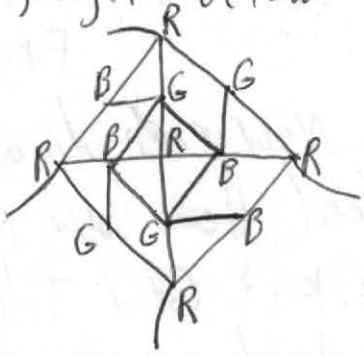
→ To rule out $f(k)n^{o(k)}$ algorithms for A, we need a parameterized reduction that blows up the parameter at most linearly.

→ To rule out $f(k)n^{o(\sqrt{k})}$ algorithms, we need a parameterized reduction that blows up the parameter at most quadratically.

→ Thus assuming ETH, there is no $f(k)n^{o(k)}$ algorithms for Set Cover, Hitting set, Connected dominating set, Independent dominating set, Partial vertex cover & dominating set in bipartite graphs.

What about planar Graphs:

There is a standard (NP-complete) reduction from 3-coloring in general graphs to planar 3-coloring uses a "crossover gadget" below



claim 1: In every 3-coloring of the gadget, opposite external connectors have the same color

claim 2: Every coloring of the external connectors where the opposite vertices have the same color can be extended to the whole gadget.

Thus if two edges cross, we replace them with a crossover gadget and claim 1 & 2 guarantees that two end-points of an edge have different colors.

The reduction from 3-coloring to planar 3-coloring introduces $O(1)$ new edges/vertices. Thus a graph with m edges can be drawn with $O(m^2)$ crossings.

Corollary: Assuming ETH, there is no $2^{o(\sqrt{n})}$ algorithm for 3-coloring an n-vertex planar graph G (observed by [Cai and Juedes '03]).

Consequence: Assuming ETH, there is no $2^{o(\sqrt{m})}$ time and no $2^{o(\sqrt{k})} n^{O(1)}$ time algorithm on planar graphs for: (K-) Independent set, (K-) Dominating set, (K-) vertex cover, (K-) path and (K-) Feedback vertex set, etc.

Note that above bounds are indeed tight due to Bidimensionality Algorithms.

To get stronger and more bounds, Grid Tiling is the key problem (introduced by [Marx '07])

Grid Tiling:

Input: A $k \times k$ matrix and a set of pairs $S_{i,j} \subseteq [D] \times [D]$ for each cell

Output: find a pair $s_{i,j} \in S_{i,j}$ for each cell such that
- Vertical neighbors agree in the first coordinate,
- Horizontal neighbors agree in the second coordinate.

Thm: Grid Tiling is W[1]-hard

Pf: Reduction from K-clique

definition of sets: for $i=j$ $(x,y) \in S_{i,j} \Leftrightarrow x=y$

For $i \neq j$: $(x,y) \in S_{ij} \Leftrightarrow x$ and y are adjacent in G.

Now each diagonal cell defines a vertex in clique.

(1,1) (3,1) (3,4)	(5,1) (1,4) (5,3)	(1,1) (2,4)
(2,2) (1,4)	(3,1) (1,2)	(2,2)
(2,3) (1,3)	(1,1) (1,3)	(2,3)

example

Indeed the above reduction gives even stronger lower bound:

Thm: $k \times k$ Grid Tiling is W[1]-hard and assuming ETH, cannot be solved in time $f(k)n^{o(k)}$ for any function f.

This lower bound is the key for proving hardness results for planar graphs or even general graphs: (the matrix is like a grid which is planar)

Examples: - List Coloring on planar graphs

- multiway cut on planar graphs with k terminals

- Independent Set for unit disks

(✗) planar directed steiner Forest with k terminals

Let's see an overview of the proof of (✗).

We want to show DSF (Directed steiner forest, which connects k terminal pairs (s_i, t_i) in a directed graph), under ETH does not have any algorithm $f(k)n^{o(k)}$ even on planar DAGs.

Here we have $2k$ terminals (a_i, b_i) and (c_i, d_i) as shown in Fig 4 (see page 11), we construct the graph in Fig 4, which has the same size as Grid tiling and thus $f(k)n^{o(k)}$ hardness of Grid Tiling gives the same hardness for DSF even on planar DAGs. (combing Thm 9.1 & 9.2 with hardness of Grid Tiling. (see more details in Chitnis, H, Marx '14)

If $(x,y) = s_{i,j} \in S_{i,j}$ then we color green the vertex in the gadget $G_{i,j}$. (See A in Fig 4)

each gadget corresponds to one set in Grid Tiling

P_i^x and Q_j^y are Canonical Paths

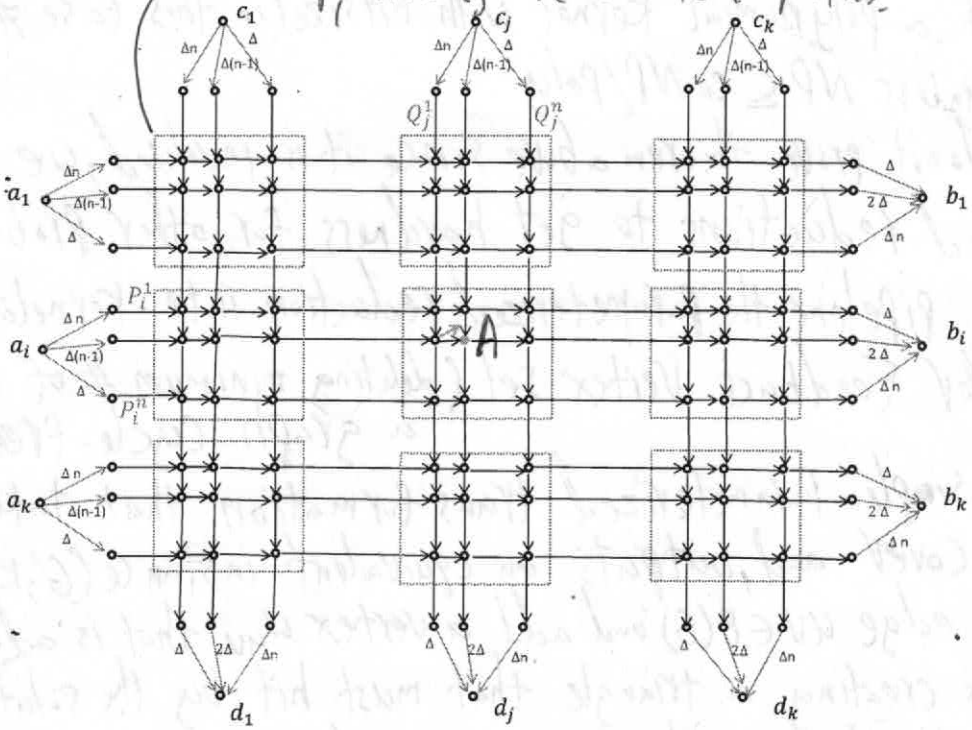


Figure 4: The instance of DSF created from an instance of Grid Tiling.

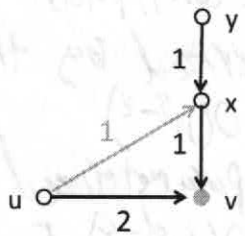


Figure 5: Let u, v be two consecutive vertices on the canonical path say P_i^ℓ. Let v be on the canonical path $Q_j^{\ell'}$ and let y be the vertex preceding it on this path. If v is a green vertex then we subdivide the edge (y,v) by introducing a new vertex x and adding two edges (y,x) and (x,v) of weight 1. We also add an edge (u,x) of weight 1. The idea is if both the edges (y,v) and (u,v) were being used initially then now we can save a weight of 1 by making the horizontal path choose (u,x) and then we get (x,v) for free, as it is already being used by the vertical canonical path.

We need a small technical modification: we add one dummy row and column to the GRID TILING instance. Essentially we now have a dummy index 1. So neither the first row nor the first column of any $S_{i,j}$ has any elements in the GRID TILING instance. That is, no green vertex can be in the first row or first column of any gadget. Combining this fact with the orientation of the edges we get the only gadgets which can intersect any $a_i \rightsquigarrow b_i$ path are $G_{i,1}, G_{i,2}, \ldots, G_{i,k}$. Similarly the only gadgets which can intersect any $c_j \rightsquigarrow d_j$ path are $G_{1,j}, G_{2,j}, \ldots, G_{k,j}$.

We now prove two theorems which together give a reduction from GRID TILING to DSF.

THEOREM 9.1. GRID TILING *has a solution implies OPT for DSF is at most* $\beta - k^2$.

Proof. For each $1 \le i, j \le k$ let $s_{i,j} \in S_{i,j}$ be the vertex in the solution of the GRID TILING instance. Therefore for every $i \in k$ we know that each of the k vertices $s_{i,1}, s_{i,2}, \ldots, s_{i,k}$ have the same x-coordinate, say α_i. Similarly for every $j \in [k]$ each of the k vertices $s_{1,j}, s_{2,j}, \ldots, s_{k,j}$ has the same x-coordinate, say γ_j. We now use the canonical path $P_i^{\alpha_i}$ for (a_i, b_i) and the canonical path $Q_j^{\gamma_j}$ for (c_j, d_j). Each of the $c_j \rightsquigarrow d_j$ paths will pay the full weight of a canonical path, which is $2(n+1) + 2(k+1) + 2k(n-1)$. However each $a_i \to b_i$ path will encounter a green vertex in each of the k gadgets along the way and save k in each path. Hence overall terminals we save a weight of k^2 and we have a solution to DSF of weight $\beta - k^2$.

Thm 9.2. optimum for DSF is at most $\beta - k^2$ implies Grid Tiling has a solution.

Pf: The proof is more involved and use concept of Canonical Path.

Finally note that one probably most important result in lower bound for (12)
kernalization is that:

Thm(*) For any $\epsilon > 0$, the vertex cover problem parameterized by solution size does not admit a polynomial kernel with bitsize (in this case # of edges in $O(k^{2-\epsilon})$ unless $NP \subseteq coNP/poly$.

→ Though we dont prove theorem above since it is involved, we can use Parameterized reductions to get hardness for other problems (since we can always pipeline the parameterized reduction into kernelization Algorithm).
e.g. Consider Feedback Vertex set (deleting minimum # of vertices to make a graph cycle-free).

There is a simple Parameterized transformation that takes an instance (G,k) of vertex cover and outputs an equivalent instance (G',k) of Feedback V.S. Take every edge $uv \in E(G)$ and add a vertex w_{uv} that is adjacent only to u & v (in G'); thus creating a triangle that must hit by the solution. Thus we get a similar Thm to Thm(*) for Feedback vertex set as well.
we have more involved theorems as well.

Thm: let $\epsilon > 0$ be any constant. Unless $NP \subseteq coNP/poly$;
- for any $q \geq 3$, the q-SAT problem parameterized by the number of variables n does not have a kernel of bitsize $O(n^{q-\epsilon})$
- For any $d \geq 2$, the d-Hitting Set problem parameterized by solution size k does not have a kernel with bitsize $O(k^{d-\epsilon})$. [in d-Hitting set, we [generalization vertex cover] want to find a subset of min. size that intersects every set of collection C which has size $\leq d$]

Note that Thm above says trivial kernelization which just removes duplicates of clauses or sets is the best that we can do.

Another important theorem is as follows for Steiner Tree [which asks for minimum # of edges which connects a terminal set $T \subseteq V(G)$.]

Thm: Steiner tree parameterized by the size of the tree does not admit a polynomial kernel unless $NP \subseteq coNP/poly$.

Cor: The same holds is steiner tree is parameterized by $|T|$ instead of solution size (since $|T| \leq$ solution size $+1$)

FPT & Approximation: An FPT optimum approximation algorithm for a problem Q with approximation ratio ρ is an algorithm A that, given an input x output a $y \in Sol(x)$ such that

$$\begin{cases} cost(x,y) \leq opt(x) \cdot \rho(opt(x)) & \text{if goal is min} \\ cost(x,y) \geq opt(x)/\rho(opt(x)) & \text{if goal is max} \end{cases}$$

we require that on input x the algorithm A runs in $f(opt(x)) \cdot |x|^{O(1)}$ time for some computable function f.

Chitnis, Hajiaghayi and Kortsarz [IPEC'13] prove

thm 1: Under the ETH and another conjecture Projection Games Conjecture (PGC), there exists constants $F_1, F_2 > 0$ such that the set cover problem does not admit an FPT optimum approximation algorithm with ratio $\rho(opt) = opt^{F_1}$ in $2^{opt^{F_2}} poly(N,M)$ time where N is the size of the universe and M is the number of sets.

Thm 2: Unless $NP \subseteq SUBEXP$, for every $0 < \delta < 1$ there exists a constant $F(\delta) > 0$ such that clique has no FPT optimum approximation with ratio $\rho(opt) = opt^{1-\delta}$ in $2^{opt^F} poly(n)$ time, where n is the number of vertices in the graph.

There are a few $f(opt)$ approximation for some W-hard problems but still the field is very new and we expect more results to be known in the field.

FPT & Streaming: Though in streaming & semi-streaming, we often assume the space needed is a function of n. However for many reasonable graph problems, we can assume the solution is not large on real-world input instances. Thus we can aim for Parametrized streaming in which we require the memory be in $\tilde{O}(k) = O(k \, polylog \, n)$.

Chitnis, Cormode, Hajiaghayi & Monemizadeh [SODA'15] introduce the concept of parametrized streaming and obtain such algorithms for (maximal) matching and vertex cover. The consider two models:

1- insertion only in which edges will be just added.

2- dynamic streaming in which we have both insertion and deletion of edges.

CCHM'15 obtain results for both cases. However for dynamic streaming they only consider the case that the solution size is at most k during the entire course of the stream. This area is novel & new and obtaining new hardness results for it is very interesting, e.g., if the size of the solution just at the end of the stream is at most k and in the middle can be large.