Basic Parametric Complexity II: Positive Techniques

Rod Downey
Victoria University
Wellington

Positive techniques

- Elementary ones
- Logical metatheorems
- Limits

Kernelization

- I believe that the most important practical technique is called kernelization.
- pre-processing, or reducing

Karsten Weihe's Train Problem

- **Train Covering by Stations**
  Instance: A bipartite graph $G = (V_S \cup V_T, E)$, where the set of vertices V_S represents railway stations and the set of vertices V_T represents trains. E contains an edge $(s, t), s \in V_s, t \in V_T$, iff the train t stops at the station s.
  Problem: Find a minimum set $V' \subseteq V_S$ such that V' covers V_T, that is, for every vertex $t \in V_T$, there is some $s \in V'$ such that $(s, t) \in E$.

Weihe's Solution

- Reduction Rule TCS1:
  Let $N(t)$ denote the neighbours of t in V_S. If $N(t) \subseteq N(t')$ then remove t' and all adjacent edges of t' from G. If there is a station that covers t, then this station also covers t'.
- Reduction Rule TCS2:
  Let $N(s)$ denote the neighbours of s in V_T. If $N(s) \subseteq N(s')$ then remove s and all adjacent edges of s from G. If there is a train covered by s, then this train is also covered by s'.

- European train schedule, gave a graph consisting of around $1.6 \cdot 10^5$ vertices and $1.6 \cdot 10^6$ edges.
- Solved in minutes.
- This has also been applied in practice as a subroutine in practical heuristical algorithms.

The Idea

- Reduce the parameterized problem to a kernel whose size depends solely on the parameter
- As compared to the classical case where this process is a central heuristic we get a provable performance guarantee.
- We remark that often the performance is much better than we should expect especially when elementary methods are used.

Vertex Cover

- Reduction Rule VC1:
  Remove all isolated vertices.
- Reduction Rule VC2:
  For any degree one vertex v, add its single neighbour u to the solution set and remove u and all of its incident edges from the graph.
- Note $(G, k) \to (G', k - 1)$.
- (S. Buss) Reduction Rule VC3:
  If there is a vertex v of degree at least $k + 1$, add v to the solution set and remove v and all of its incident edges from the graph.
- The result is a graph with no vertices of degree $> k$ and this can have a VC of size k only if it has $< k^2$ many edges.

DEFINITION (KERNELIZATION)

Let $L \subseteq \Sigma^* \times \Sigma^*$ be a parameterized language. Let $\mathcal{L}$ be the corresponding pa rameterized problem, that is, $\mathcal{L}$ consists of input pairs (I, k), where I is the main part of the input and k is the parameter. A reduction to a problem kernel, or kernelization, comprises replacing an instance (I, k) by a reduced instance (I', k'), called a problem kernel, such that

(i) $k' \leq k$,
(ii) $|I'| \leq g(k)$, *for some function* g *depending only on* k, *and*
(iii) $(I, k) \in L$ *if and only if* $(I', k') \in L$.

The reduction from (I, k) to (I', k') must be computable in time polynomial in $|I|$.

A Useless Theorem

Theorem (Cai, Chen, Downey and Fellows)

$L \in FPT$ iff L is kernelizable.

- Proof Let $L \in FPT$ via a algorithm running in time $n^c.f(k)$. Then run the algorithm which in time $O(n^{c+1})$, which eventually dominates $f(k)n^c$, either computes the solution or understands that it is in the first $g(k)$ many exceptional cases. ("Eventually polynomial time")

Strategies for improving I: Bounded Search Trees

- Buss's algorithm gives crudely a $2n + k^{k^2}$ algorithm for k-VC.
- Here is another algorithm: (DF) Take any edge $e = v_1 v_2$. either v_1 or v_2 is in any VC. Begin a tree T with first children v_1 and v_2. At each child delete all edges covered by the v_i.
- repeat to depth k.
- Gives a $O(2^k \cdot n)$ algorithm.
- Now combine the two: Gives a $2n + 2^k k^2$ algorithm.

- If G has paths of degree 2, then there are simple reduction rules to deal with them first. Thus we consider that G is of min degree 3.
  BRANCHING RULE VC2:
  If there is a degree two vertex v in G, with neighbours w_1 and w_2, then either both w_1 and w_2 are in a minimum size cover, or v together with all other neighbours of w_1 and w_2 are in a minimum size cover.
- Now when considering the kernel, for each vertex considered either v is included or all of its neighbours (at least) $\{p, q\}$ are included.
- Now the tree looks different. The first child nodes are labelled v or $\{p, q\}$, and on the right branch the parameter drops by 2 instead of 1. or similarly with the w_i case.

- Now the size of the search tree and hence the time complexity is determined by some recurrence relation.
- many, many versions of this idea with increasingly sophisticated reduction rules.

THEOREM (NEMHAUSER AND TROTTER (1975))

For an n-vertex graph $G = (V, E)$ with m edges, we can compute two disjoint sets $C' \subseteq V$ and $V' \subseteq V$, in $O(\sqrt{n} \cdot m)$ time, such that the following three properties hold:

(i) There is a minimum size vertex cover of G that contains C'.
(ii) A minimum vertex cover for the induced subgraph $G[V']$ has size at least $|V'|/2$.
(iii) If $D \subseteq V'$ is a vertex cover of the induced subgraph $G[V']$, then $C = D \cup C'$ is a vertex cover of G.

THEOREM (CHEN ET AL. (2001))

Let $(G = (V, E), k)$ be an instance of K-VERTEX COVER. *In $O(k \cdot |V| + k^3)$ time we can reduce this instance to a problem kernel $(G = (V', E'), k')$ with $|V'| \leq 2k$.*

- The current champion using this approach is a $O^*(1.286^k)$ (Chen01).
- Here the useful O^* notation only looks at the exponential part of the algorithm.

INTERACTIONS

- Now we can ask lots of questions. How small can the kernel be?
- Notice that applying the kernelization to the unbounded problem yields a approximation algorithm.
- Using the PCP theorem we know that no kernel can be smaller that 1.36 k unless P=NP (Dinur and Safra) as no better approximation is possible. Is this tight?
- Actually we know that no $O^*(1+\epsilon)^k)$ is possible unless ETH fails.

DEFINITION

A crown in a graph $G = (V, E)$ consists of an independent set $I \subseteq V$ and a set H containing all vertices in V adjacent to I.

- For example a degree 1 vertex and its neighbour is a crown.
- For a crown $I \cup H$ in G, then we need at least $|H|$ vertices to cover all edges in the crown.
- REDUCTION RULE CR:
  For any crown $I \cup H$ in G, add the set of vertices H to the solution set and remove $I \cup H$ and all of the incide nt edges of $I \cup H$ from G.
- Shrinkage $(G, k) \rightarrow (G', k - |H|)$.

How to use Crowns?

Theorem (Abu-Khzam, Collins, Fellows, Langston, Suters, Symons (2004))

A graph that is crown-free and has a vertex cover of size at most k can contain at most $3k$ vertices.

Theorem (Chor, Fellows, Juedes (2004))

If a graph $G = (V, E)$ has an independent set $V' \subset V$ such that $|N(V')| < |V'|$, then a crown $I \cup H$ with $I \subseteq V'$ can be found in G in time $O(n + m)$.

- Other examples found in SIGACT News Gou-Niedermeier's survey on kernelization.

INTERLEAVING

- (Niedermeier and Rossmanith, 2000) showed that iteratively combining kernelization and bounded search trees often performs much better than either one alone or one followed by the other.
- Begin a search tree, and apply kernelization, then continue etc. Analysing the combinatorics shows a significant reduction in time complexity, which is very effective in practice.

An Example

- (NR) As an example, 3-Hitting Set (Given a collection of subsets of size 3 from a set S find k elements of S which hit the sets.) An instance (I, k) of this problem can be reduced to a kernel of size k^3 in time $O(|I|)$, and the problem can be solved by employing a search tree of size 2.27^k. Compare a running time of $O(2.27^k \cdot k^3 + |I|)$ (without interleaving) with a running time of $O(2.27^k + |I|)$ (with interleaving).
- Interesting and not yet developed generalization due to Abu-Khzam 2007 uses pseudo-kernelization. (TOCS, October 2007)

- Reed, Smith and Vetta 2004. For the problem of “within k of being bipartite” (by deletion of edges).

DEFINITION (COMPRESSION ROUTINE)

A compression routine is an algorithm that, given a problem instance I and a solution of size k, either calculates a smaller solution or proves that the given solution is of minimum size.

An Example, VC again!

- $(G = (V, E), k)$, start with $V' = \emptyset$, and (solution) $C = \emptyset$.
- Add a new vertex v to both V' and C, $V' \leftarrow V' \cup \{v\},\ C \leftarrow C \cup \{v\}$.
- Now call the compression routine on the pair $(G[V'], C)$, where $G[V']$ is the subgraph induced by V' in G, to obtain a new solution C'. If $|C'| > k$ then we output NO, otherwise we set $C \leftarrow C'$.
- If we successfully complete the nth step where $V' = V$, we output C with $|C| \leq k$. Note that C will be an optimal solution for G. (Algo runs in time $O(2^k mn)$.)

Practice

- I remark that in practice these methods work much better than we might expect.
- Langston's work with irradiated mice, ETH group in Zurich, Karesten Weihe
- See The Computer Journal especially articles by Langston et al.

Less practical algorithms

- In what follows we look at algorithms that in general seem less practical but can sometimes work in practice.

Colour Coding

- k-Subgraph Isomorphism
  Instance: $G = (V, E)$ and a graph $H = (V^H, E^H)$ with $|V^H| = k$.
  Parameter: A positive integer k (or V^H).
  Question: Is H isomorphic to a subgraph in G?
- Idea: to find the desired set of vertices V' in G, isomorphic to H, we randomly colour all the vertices of G with k colours and expect that there is a colourful solution; all the vertices of V' have different colours.
- G uniformly at random with k colors, a set of k distinct vertices will obtain different colours with probability $(k!)/k^k$. This probability is lower-bounded by e^{-k}, so we need to repeat the process e^k times to have probability one of obtaining the required colouring.

DERANDOMIZATION

- We need a list of colorings of the vertices in G such that, for each subset $V' \subseteq V$ with $|V'| = k$ there is at least one coloring in the list by which all vertices in V' obtain different colors.

DEFINITION (k-PERFECT HASH FUNCTIONS)

A k-perfect family of hash functions is a family $\mathcal{H}$ of functions from $\{1, 2, ..., n\}$ onto $\{1, 2, ..., k\}$ such that, for each $S \subset \{1, 2, ..., n\}$ with $|S| = k$, there exists an $h \in \mathcal{H}$ such that h is bijective when restricted to S.

Theorem (Alon et al. (1995))

Families of k-perfect hash functions from $\{1, 2, ..., n\}$ onto $\{1, 2, ..., k\}$ can be constructed which consist of $2^{O(k)} \cdot \log n$ hash functions. For such a hash function, h, the value $h(i)$, $1 \leq i \leq n$, can be computed in linear time.

An example

- k-Cycle
- For each colouring h, we check every ordering $c_1, c_2, \ldots, c_k$ of the k colours to decide whether or not it realizes a k-cycle. We first construct a directed graph G' as follows:
  For each edge $(u, v) \in E$, if $h(u) = c_i$ and $h(v) = c_{i+1 (\bmod k)}$ for some i, then replace (u, v) with arc $\langle u, v \rangle$, otherwise delete (u, v).
  In G', for each v with $h(v) = c_1$, we use a breadth first search to check for a cycle C from v to v of length k.
- $2^{O(k)} \cdot \log |V|$ colourings, and $k!$ orderings. k-cycle in time $O(k \cdot |V|^2)$.

- Graphs constructed inductively. Treewidth, Pathwidth, Branschwidth, Cliquewid th, mixed width etc. k-Inductive graphs, plus old favourites such as planarity etc, which can be viewed as local width.
- Example:

DEFINITION

[Tree decomposition and Treewidth] Let $G = (V, E)$ be a graph. A tree decomposition, TD, of G is a pair $(T, \mathcal{X})$ where
1. $T = (I, F)$ is a tree, and
2. $\mathcal{X} = \{X_i \mid i \in I\}$ is a family of subsets of V, one for each node of T, such that

(i) $\bigcup_{i \in I} X_i = V$,
(ii) for every edge $\{v, w\} \in E$, *there is an* $i \in I$ *with* $v \in X_i$ *and* $w \in X_i$, *and*
(iii) for all $i, j, k \in I$, *if* j *is on the path from* i *to* k *in* T, *then* $X_i \cap X_k \subseteq X_j$.

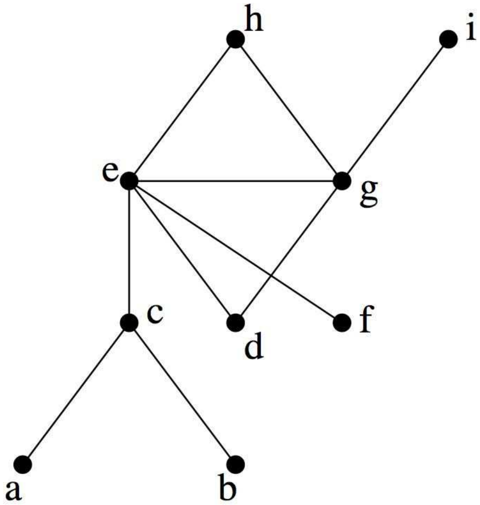
- This gives the following well-known definition.

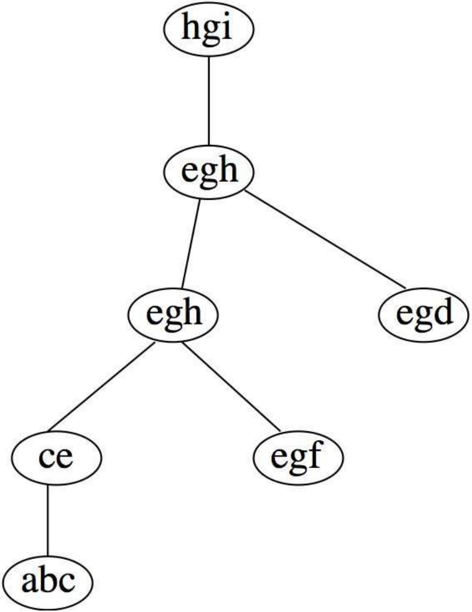
DEFINITION

The width of a tree decomposition $((I, F), \{X_i \mid i \in I\})$ is $\max_{i \in I} |X_i| - 1$. The treewidth of a graph G, denoted by $tw(G)$, is the minimum width over all possible tree decompositions of G.

- The following refers to any of these inductively defined graphs families. Notes that many commercial constructions of, for example chips are inductively defined.
  1. Find a bounded-width tree (path) decomposition of the input graph that exhibits the underlying tree (path) structure.
  2. Perform dynamic programming on this decomposition to solve the problem.

AN EXAMPLE FOR INDEXPENDENT SET




hgi
egh
egh
egd
ce
egf
abc

$\emptyset$	a	b	c	ab	ac	bc	abc
0	1	1	1	2	-	-	-

Bodlaender's Theorem

- The following theorem is shows that treewidth is FPT. Improves many earlier results showing this. The constant is about 2^{35k^2}.

Theorem (Bodlaender)

k-Treewidth is linear time FPT

- Not practical because of large hidden O term.
- Unknown if there is a practical FPT treewidth algorithm
- Nevertheless approximation and algorithms specific to knwon decomps run well at least sometimes.

- There have been some (at least theortical) applications on IP with bounded variables.

THEOREM (LENSTRA)

Integer programming feasibility can be solved with $O(p^{\frac{9p}{2}}L)$ *arithmetical operations in integers of* $O(p^{\frac{9p}{2}}L)$ *bits where* p *is the number of input variables and* L *is the number of input bits for the LIP instance.*

- I don't know much about this but you can look at Rolf Niedermeier's book (Invitation to Fixed Parameter Algorithms)
- Mostly impractical.

- (First order Logic)
    1. **Atomic formulas:** $x = y$ and $R(x_1, ..., x_k)$, where R is a k-ary relation symbol and $x, y, x_1, ..., x_k$ are individual variables, are FO-formulas.
    2. **Conjunction, Disjunction:** If ϕ and ψ are FO-formulas, then $\phi \wedge \psi$ is an FO-formula and $\phi \vee \psi$ is an FO-formula.
    3. **Negation:** If ϕ is an FO-formula, then $\neg\phi$ is an FO-formula.
    4. **Quantification:** If ϕ is an FO-formula and x is an individual variable, then $\exists x\ \phi$ is an FO-formula and $\forall x\ \phi$ is an FO-formula.
- Eg We can state that a graph has a clique of size k using an FO-formula,

$$\exists x_1 ... x_k \bigwedge_{1 \leq i \leq j \leq k} E(x_i, x_j)$$

- Two sorted structure with variables for sets of objects.
- 1. **Additional atomic formulas:** For all set variables X and individual variables y, Xy is an MSO-formula.
  2. **Set quantification:** If ϕ is an MSO-formula and X is a set variable, then $\exists X\ \phi$ is an MSO -formula, and $\forall X\ \phi$ is an MSO-formula.
- Eg k-col

$$\exists X_1,,,\exists X_k \left(\forall x \bigvee_{i=1}^{k} X_i x \wedge \forall x \forall y \left(E(x,y) \rightarrow \bigwedge_{i=1}^{k} \neg (X_i x \wedge X_i y) \right) \right)$$

- Instance: A structure $\mathcal{A} \in \mathcal{D}$, and a sentence (no free variables) $\phi \in \Phi$.
  Question: Does $\mathcal{A}$ satisfy ϕ?
- PSPACE-complete for FO and MSO.

Theorem (Courcelle 1990)

The model-checking problem for MSO restricted to graphs of bounded treewidth is linear-time fixed-parameter tractable.

Detleef Seese has proved a converse to Courcelle's theorem.

Theorem (Seese 1991)

Suppose that $\mathcal{F}$ is any family of graphs for which the model-checking problem for MSO is decidable, then there is a number n such that, for all $G \in \mathcal{F}$, the treewidth of G is less than n.

LOCAL TREEWIDTH

- $ltw(G)(r) = \max \{tw(N_r(v)) \mid v \in V(G)\}$ where $N_r(v)$ is the neighbourhood of radius r about v.
- A class of graphs $\mathcal{C} = \{G : G \in D\}$ has bounded local treewidth if there is a function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that, for $r \geq 1$, $ltw(G)(r) \leq f(r)$i, for all $G \in \mathcal{C}$.
- Examples Bounded degree, bounded treewidth, bounded genus, excluding a minor

THE FRICK GROHE THEOREM

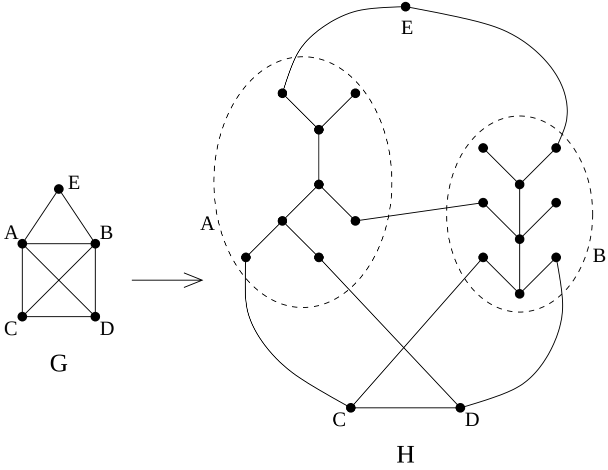
THEOREM (FRICK AND GROHE 1999)

Parameterized problems that can be described as model-checking problems for FO are fixed-parameter tractable on classes of graphs of bounded local treewidth.

For example DOMINATING SET, INDEPENDENT SET, or SUBGRAPH ISOMORPHISM are FPT on planar graphs, or on graphs of bounded degree

More exotic methods

- minor ordering



- Robertson-Seymour Finite graphs are WQO's under minor ordering. $H \leq_{minor} G$ is $O(|G|^3)$ FPT for a fixed H.

- THEOREM (MINOR-CLOSED MEMBERSHIP)

  If $\mathcal{F}$ is a minor-closed class of graphs then membership of a graph G in $\mathcal{F}$ can be determined in time $O(f(k) \cdot |G|^3)$, where k is the collective size of the graphs in the obstruction set for $\mathcal{F}$.

  - Likely I won't have time to discuss what this means but see DF for more details.

Other work

- There has been a lot of recent work exploring the bad behaviour of the algorithms generated by the metatheorems
- Including work by Grohe and co-authors showing that the iterated exponentials cannot be gottent rid of unless P=NP or FPT=W[1] in the MSO case and the local treewidth case respectively.
- Including work of Bodlaender, Downey, Fellows, and Hermelin showing that unless the polynomial time hierarchy collapses no small kernels for e.g. treewidth, and a wide class of problems.
- Still much to do.

Some questions

- Commercially many things are solved using SAT solvers. Why do they work. What is the reason that the instances arizing from real life behave well?
- How to show no reasonable FPT algorithm uaing some assumption?
- Develop a reasonable randomized version, PCP, etc. This is the “hottest” area in TCS yet not really developed in parameterized complexity. (Moritz Meuller has some nice work here)

Some References

- Parameterized Complexity, springer 1999 DF
- Invitation to Parameterized Algorithms, 2006 Niedermeier, OUP
- Parameterized Complexity Theory, 2006, Springer Flum and Grohe
- Theory of Computing Systems, Vol. 41, October 2007
- Parameterized Complexity for the Skeptic, D, proceedings CCC, Aarhus, (see my homepage)
- The Computer Journal, (ed Downey, Fellows, Langston)
- Parameterized Algorithmics: Theory, Practice, and Prospects, Henning Fernau, CUP to appear.

What should you do?

- You should buy that wonderful book...(and its friends)
- Than You