Recent progress in parameterized upper and lower bounds

Rod Downey
Victoria University
Wellington
Cornell, November, 2012

THIS LECTURE:

- Basic Definitions
- Basic Hardness results
- Kernelization lower bounds

- A mathematical idealization is to identify “Feasible” with P. (I won’t even bother looking at the problems with this.)
- With this assumption, the theory of NP-hardness is an excellent vehicle for mapping an outer boundary of intractability, for all practical purposes.
- Indeed, assuming the reasonable current working assumption that NTM acceptance is $\Omega(2^n)$, NP-hardness allows for practical lower bound for exact solution for problems.
- A very difficult practical and theoretical problem is “How can we deal with P?”.
- More importantly how can we deal with $P - FEASIBLE$, and map a further boundary of intractability.

- Arora 1996 gave a $O(n^{\frac{3000}{\epsilon}})$ PTAS for EUCLIDEAN TSP
- Chekuri and Khanna 2000 gave a $O(n^{12(\log(1/\epsilon)/\epsilon^8)})$ PTAS for MULTIPLE KNAPSACK
- Shamir and Tsur 1998 gave a $O(n^{2^{2^{\frac{1}{\epsilon}}}-1})$ PTAS for MAXIMUM SUBFOREST
- Chen and Miranda 1999 gave a $O(n^{(3mm!)^{\frac{m}{\epsilon}+1}})$ PTAS for GENERAL MULTIPROCESSOR JOB SCHEDULING
- Erlebach et al. 2001 gave a $O(n^{\frac{4}{\pi}(\frac{1}{\epsilon^2}+1)^2(\frac{1}{\epsilon^2}+2)^2})$ PTAS for MAXIMUM INDEPENDENT SET for geometric graphs.

- Deng, Feng, Zhang and Zhu (2001) gave a $O(n^{5\log_{1+\epsilon}(1+(1/\epsilon))})$ PTAS for UNBOUNDED BATCH SCHEDULING.
- Shachnai and Tamir (2000) gave a $O(n^{64/\epsilon+(log(1/\epsilon)/\epsilon^8)})$ PTAS for CLASS-CONSTRAINED PACKING PROBLEM (3 cols).

Reference	Running Time for a 20% Error
Arora (AR96)	$O(n^{15000})$
Chekuri and Khanna (CK00)	$O(n^{9,375,000})$
Shamir and Tsur (ST98)	$O(n^{958,267,391})$
Chen and Miranda (CM99)	$> O(n^{10^{60}})$ (4 Processors)
Erlebach et al. (EJS01)	$O(n^{523,804})$
Deng et. al (DFZZ01)	$O(n^{50})$
Shachnai and Tamir (ST00)	$O(n^{1021570})$

The Running Times for Some Recent PTAS's with 20% Error.

What is the problem here?

- Arora (1997) gave a PTAS running in nearly linear time for EUCLIDIAN TSP. What is the difference between this and the PTAS's in the table. Can't we simply argue that with more effort all of these will eventually have truly feasible PTAS's.
- The principal problem with the baddies is that these algorithms have a factor of $\frac{1}{\epsilon}$ (or worse) in their exponents.
- By analogy with the situation of *NP* completeness, we have some problem that has an exponential algorithm. Can't we argue that with more effort, we'll find a much better algorithm? As in Garey and Johnson's famous cartoon, we cannot seem to prove a better algorithm. BUT we prove that it is NP hard.

I'M DUBIOUS; EXAMPLE?

- Then assuming the working hypothesis that there is basically no way to figure out if a NTM has an accepting path of length n except trying all possibilities there is no hope for an exact solution with running time significantly better than 2^n. (Or at least no polynomial time algorithm.)
- Our new working hypothesis that there is basically no way to figure out if a NTM has an accepting path of length k except trying all possibilities. Note that there are $\Omega(n^k)$ possibilities. (Or at least no way to get the "k" out of the exponent or an algorithm deciding k-STEP NTM.)
- One then defines the appropriate reductions from k-STEP TURING MACHINE HALTING to the PTAS using $k = \frac{1}{\epsilon}$ as a parameter to argue that if we can "get rid" of the k from the exponent then it can only be if the working hypothesis is wrong.

Two Basic Examples

- VERTEX COVER
  Input: A Graph G.
  Parameter : A positive integer k.
  Question: Does G have a size k vertex cover? (Vertices cover edges.)
- DOMINATING SET
  Input: A Graph G.
  Parameter : A positive integer k.
  Question: Does G have a size k dominating set? (Vertices cover vertices.)

- VERTEX COVER is solvable by an algorithm $\mathfrak{O}$ in time $f(k)|G|$, a behaviour we call fixed parameter tractability, (Specifically $1.28^k k^2 + c|G|$, with c a small absolute constant independent o f k.)
- Whereas the only known algorithm for DOMINATING SET is complete search of the possible k-subsets, which takes time $\Omega(|G|^k)$.

- In the below I will mostly talk for convenience about graphs.
- I could just as easily be talking about many other areas.
- In the Computer Journal alone, there is biological, artificial intelligence, constraint satisfaction, geometric problems, scheduling, cognitive science, voting, combinatorial optimzation, phylogeny. Model check is the basis of Flum-Grohe.

Basic Definition(s)

- Setting : Languages $L \subseteq \Sigma^* \times \Sigma^*$.
- Example (Graph, Parameter).
- We say that a language L is fixed parameter tractable if there is a algorithm M, a constant C and a function f such that for all x, k,

$$(x, k) \in L \text{ iff } M(x) = 1 \text{ and}$$

$$\text{the running time of } M(x) \text{ is} f(k)|x|^C.$$

Parameters

- Without even going into details, think of all the graphs you have given names to and each has a relevant parameter: planar, bounded genus, bounded cutwidth, pathwidth, treewidth, degree, interval, etc, etc.
- Also nature is kind in that for many practical problems the input (often designed by us) is nicely ordered.

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.

- It is worth remarking that there are problems notably FPT by bounded search tree (type checking in ML) that are not known to have polynomial size kernels, and some “provably” don’t.
- Another easy example for bounded search trees is PLANAR INDEPENDENT SET. (Start with a degree 5 vertex, branching rule of size 6)

- 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.
- This method has a 2005 (Fomin, Grandoni, Kratsch) incarnation called measure and conquer where the branching rules are given *rational valued* weights, and decisions as to what to do are figured out by optimization.
- For example the best exact algorithm for SET COVER and DOMINATING SET use this. (van Rooij-Bodlaender point out that this can be used for algoritm design as well.)
- Jianer Chen and others use this in many FPT algorithms such as the state of the art for FEEDBACK VERTEX SET and VERTEX COVER.

Shrink the Kernel

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) The best is $O^*(1.2745^k)$(Chen10 using this, iterative compression, struction, measure and conquer, and other methods).
- 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?
- Assuming the "Unique Games Conjecture" the $2k$ kernel is tight (Khot etc).
- Actually we know that no $O^*(1+\epsilon)^k)$ algorithm is possible unless ETH fails.
- ETH n-valued 3SAT is not in DTIME($2^{o(n)}$).

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 (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 $O(n + m)$.

- Can get the crown: Take a maximal matching M of G. If $|M| > k$ say no. Else $I = G - M$ is an independent set $(\leq k)$, and then use bipartite matching to match I and its neighbours. Combinatorial arguments show that this has a submatching which is a crown. Delete and repeat.
- Other examples found in SIGACT News Gou-Niedermeier's survey on kernelization.

- (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.

- 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 $O(2^k mn)$.)

- This was forst successflly applied by Reed, Smith, Vetta to GRAPH BIPARTITIZATION. The algorithm is similar, building a minimal bipartitization at each step and using what we can call acceptable partitions for the search step.
- The best now is $O^*(3.83^k)$, and it works better with algorithm engineering (Gray Codes, tree pruning) with (e.g.) biological data Hüffner 2004.
- It is a crucial step for the best two algorithms for VERTREX COVER (Chen, Kanj, Xia 2010, $O^*(1.2745^k)$ and FEEDBACK VERTEX SET (Can I remove k verteices and get a acyclic graph?) (Cao, Chen, Liu, 2009).

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.

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

- 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 high probability of obtaining the required colouring.

- 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-Path
- 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-path. We first construct a directed graph G' as follows:
  For each edge $(u, v) \in E$, if $h(u) = c_i$ and $h(v) = c_{i+1 \pmod k}$ 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 path C from v to v of length k.
- $2^{O(k)} \cdot \mathrm{1}og\,|V|$ colourings, and $k!$ orderings. k-path in time $O(k \cdot |V|^2)$.

Bounded Width Metrics

- Graphs constructed inductively. Treewidth, Pathwidth, Branschwidth, Cliquewidth 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* $X_i \cap X_k \subseteq X_j$.

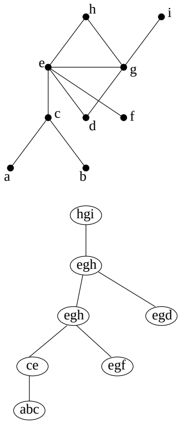
- This gives the following well-known definition.

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 Independent Set



$\emptyset$	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^3}.

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 known decomps run well at least sometimes.

- 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

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

Model Checking

- 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.

Courcelle's and Seese's Theorems

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 $f : \mathbb{N} \to \mathbb{N}$ 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

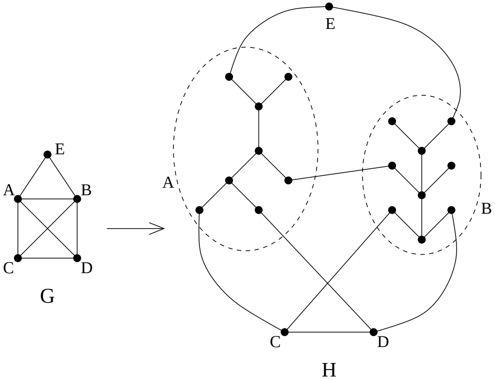
THEOREM (FRICK AND GROHE 1999)

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.

- Natural basic hardness class: $W[1]$. Does not matter what it is, save to say that the analog of Cook's Theorem is SHORT NONDETERMINISTIC TURING MACHINE ACCEPTANCE
  Instance: A nondeterministic Turing Machine M and a positive integer k.
  Parameter: k.
  Question: Does M have a computation path accepting the empty string in at most k steps?

- If one believes the philosophical argument that Cook's Theorem provides compelling evidence that SAT is intractible, then one surely must believe the same for the parametric intractability of SHORT NONDETERMINISTIC TURING MACHINE ACCEPTANCE.
- Moreover, recent work has shown that if SHORT NTM is fpt then n-variable 3SAT is in DTIME($2^{o(n)}$)

- Given two parameterized languages $L, \widehat{L} \subseteq \Sigma^* \times \Sigma^*$, say $L \leq_{FPT} \widehat{L}$ iff there are (computable) $f, x \mapsto x', k \mapsto k'$ and a constant c, such that for all x,

$$(x, k) \in L \text{ iff } (x', k') \in \widehat{L},$$

  in time $f(k)|x|^c$.
- Lots of technical question still open here.

- Analog of Cook's Theorem: (Downey, Fellows, Cai, Chen)
  WEIGHTED 3SAT$\equiv_{FTP}$ SHORT NTM ACCEPTANCE.
  WEIGHTED 3SAT
  Input: A 3 CNF formula ϕ
  Parameter: k
  Question: Does ϕ has a satisfying assignment of Hamming weigth k, meaning exactly k literals made true.

- Think about the usual poly reduction from SAT to 3SAT. It takes a clause of size p, and turns it into many clauses of size 3. But the weight control goes awry. A weight 4 assignment could go to anything.
- We don't think WEIGHTED CNF SAT$\leq_{ftp}$WEIGHTED 3 SAT.
- Gives rise to a heirarchy:

$$W[1] \subseteq W[2] \subseteq W[3] \ldots W[SAT] \subseteq W[P] \subseteq XP.$$

- XP is quite important, it is the languages which are in DTIME$(n^f(k))$ with various levels of uniformity, depending on the choice of reductions.

- *XP* has k-CAT AND MOUSE GAME and some other games ((DF99a)),
- $W[P]$ has LINEAR INEQUALITIES, SHORT SATISFIABILITY, WEIGHTED CIRCUIT SATISFIABILITY ((ADF95)) and MINIMUM AXIOM SET((DFKHW94)).
- Then there are a number of quite im portant problems from combinatorial pattern matching which are $W[t]$ hard for all t: LONGEST COMMON SUBSEQUENCE ($k =$ number of seqs.,$|\Sigma|$-two parameters) ((BDFHW95)), FEASIBLE REGISTER ASSIGNMENT, TRIANGULATING COLORED GRAPHS, BANDWIDTH, PROPER INTERVAL GRAPH COMPLETION ((BFH94)), DOMINO TREEWIDTH ((BE97)) and BOUNDED PERSISTE NCE PATHWIDTH ((McC03)).
- $W[2]$ include WEIGHTED $\{0, 1\}$ INTEGER PROGRAMMING, DOMINATING SET ((DF95a)), TOURNAMENT DOMINATING SET ((DF95c)) UNIT LENGTH PRECEDENCE CONSTRAINED SCHEDULING (hard) ((BF95)), SHORTEST COMMON SUPERSEQUENCE (k)(hard) ((FHK95)), MAXIMUM LIKELIHOOD DECODING (hard), WEIGHT DISTRIBUTION IN LINEAR CODES (hard), NEAREST VECTOR IN INTEGER LATTICES (hard) ((DFVW99)), SHORT PERMUTATION GROUP FACTORIZATION (hard).
- $W[1]$ we have a collection including k-STEP DERIVATION FOR CONTEXT SENSITIVE GRAMMARS, SHORT NTM COMPUTATION, SHORT POST CORRESPONDENCE, SQUARE TILING ((CCDF96)), WEIGHTED q–CNF SATISFIABILITY ((DF95b)), VAPNIK–CHERVONENKIS DIMENSION ((DEF93)) LONGEST COMMON SUBSEQUENCE (k, $m =$ LENGTH OF COMMON SUBSEQ.) ((BDFW95)), CLIQUE, INDEPENDENT SET ((DF95b)), and MONOTONE DATA COMPLEXITY FOR RELATIONAL DATABASES

- Notice that there are at least two ways to parameterize: Parameterize the part of the problem you want to look at and to parameterize the problem itself.
- This point of view makes this sometime a promise problem. Input something, promise it is parameterized, and ask questions about it.
- Two interpretations one with certificate one only with a promise. e.g. CLIQUEWIDTH, PATHWIDTH.
- Some recent work “lowers the hardness barrier”; perhaps giving better inapproximability results.

ETH

- Recall the exponential time hypthesis is (ETH) n-variable 3-SATISFIABILITY is not solvable in DTIME($2^{o(n)}$). (Impagliazzo Paturi and Zane.)
- This is seen an important refinement of $P \neq NP$ that is widely held to be true.
- It is related to FPT as we now see.

- INPUT A parametrically minature problem QUESTION Is it in the class
  e.g. INPUT a graph G of size $k \log n$ with n in unary.
  Does it have a vertex cover of size d?
- Get mini Vertex cover, mini Dominating set, Minisat etc.
- Core problem: minicircuitsat.

THEOREM (CHOR, FELLOWS AND JUEDES ; DOWNEY ET. AL.)

The $M[1]$ complete problems such as MIN-3SAT *are in FPT iff the exponential time hypothesis fails.*

- That is, more or less, EPT is the “same” as $M[1] \neq FPT$.
- And now we have a method of demonstrating no good subexponential algorithm; Show M[1] hardness.
- Chen-Grohe established an insomorphism between the complexity degree structures.
- Fellows conjectures that PCP like techniques will show

- This new programme regards the classes like W[1] as artifacts of the basic problem of proving hardness under reasonable assumptions, and strikes at membership ofXP.
- Eg INDEPENDENT SET and DOMINATING SET which certainly are in XP. But what's the best exponent we can hope for for slice k? They are clearly solvable in time $O(n^{k+1})$.

THEOREM (CHEN ET. AL 05)

The following hold:

(i) INDEPENDENT SET *cannot be solved in time* $n^{o(k)}$ *unless* FPT=M[1].

(ii) DOMINATING SET *cannot be solved in time* $n^{o(k)}$ *unless* FPT=M[2].

- The proofs recycle and miniaturize various NP and W[1] completeness results.
- Many recent results of similar ilk based on ETH or SETH, such as results on treewidth etc.

Where else?

- Another area is approximation. Here we ask for an algorithm which either says “no solution of size k” or here is one of size $2k$ (say).
- For example Bin Packing is has to $(k, 2k)$-approx, but k-Independent Dominating Set has not approx of the form $(k, F(k))$ for any computable F unless $FPT = W[1]$. (DFMccartin)
- Flum Grohe show that all natural $W[P]$ complete problems don’t have approx of the form $(k, F(k))$ for any computable F unless $FPT = W[P]$.

REMEMBER KERNELIZATION?

- When can we show that a FPT problem likely has no polynomial size kernel?
- Notice that if P=NP then all have constant size kernel, so some reasonable assumption is needed.

DEFINITION (BODLAENDER, DOWNEY, FELLOWS, HERMELIN)

labelDefinition: DistillationA OR-distillation algorithm for a classical problem $L \subseteq \Sigma^*$ is an algorithm that

- receives as input a sequence $(x_1, \ldots, x_t)$, with $x_i \in \Sigma^*$ for each $1 \leq i \leq t$,
- uses time polynomial in $\sum_{i=1}^{t} |x_i|$,
- and outputs a string $y \in \Sigma^*$ with
  1. $y \in L \iff x_i \in L$ for some $1 \leq i \leq t$.
  2. $|y|$ is polynomial in $\max_{1 \leq i \leq t} |x_i|$.

- Similarly AND-distillation.

LEMMA (FORTNOW AND SANTHANAM 2007)

If any NP complete problem has a distillation algorithm then $PH = \Sigma_3^P$. That is, the polynomial time hierarchy collapses to three or fewer levels That is, the polynomial time hierarchy collapses to three or fewer levels

- Here Σ_3^p is $NP^{NP^{NP}}$.
- Strictly speaking the prove that $co - NP \subseteq NP\backslash poly$.

The Proof

- Let L be NP complete. We show that $\overline{L}$ is in NP\poly if L has dist.
- Let $\overline{L}_n = \{x \notin L : |x| \leq n\}$.
- Given any $x_1, \ldots, x_t \in \overline{L}_n$, the distillation algorithm $\mathcal{A}$ maps $(x_1, \ldots, x_t)$ to some $y \in \overline{L}_{n^c}$, where c is some constant independent of t.

- The main part of the proof consists in showing that there exists a set $S_n \subseteq \overline{L}_{n^c}$, with $|S_n|$ polynomially bounded in n, such that for any $x \in \Sigma^{\leq n}$ (PHP) we have the following:
  - If $x \in \overline{L}_n$, then there exist strings $x_1, \ldots, x_t \in \Sigma^{\leq n}$ with $x_i = x$ for some i, $1 \leq i \leq t$, such that $\mathcal{A}(x_1, \ldots, x_t) \in S_n$.
  - If $x \notin \overline{L}_n$, then for all strings $x_1, \ldots, x_t \in \Sigma^{\leq n}$ with $x_i = x$ for some i, $1 \leq i \leq t$, we have $\mathcal{A}(x_1, \ldots, x_t) \notin S_n$.
- to decide if $x \in \overline{L}$, guess t strings $x_1, \ldots, x_t \in \Sigma^{\leq n}$, and checks whether one of them is x. If not, it immediately rejects. Otherwise, it computes $\mathcal{A}(x_1, \ldots, x_t)$, and accepts iff the output is in S_n. It is immediate to verify that M correctly determines (in the non-deterministic sense) whether $x \in \overline{L}_n$.

How does this relate to kernelization?

Definition (Bodlaender, Downey, Fellows, Hermelin)

A OR-composition algorithm for a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that

- receives as input a sequence $((x_1, k), \ldots, (x_t, k))$, with $(x_i, k) \in \Sigma^* \times \mathbb{N}^+$ for each $1 \leq i \leq t$,
- uses time polynomial in $\sum_{i=1}^{t} |x_i| + k$,
- and outputs $(y, k') \in \Sigma^* \times \mathbb{N}^+$ with
  1. $(y, k') \in L \iff (x_i, k) \in L$ for some $1 \leq i \leq t$.
  2. k' is polynomial in k.

Lemma (Bodlaender, Downey, Fellows, Hermelin)

Let L be a compositional parameterized problem whose derived classical problem L_c is NP-complete. If L has a polynomial kernel, then L_c is also distillable.

LEMMA (BODLAENDER, DOWNEY, FELLOWS, HERMELIN)

Let L be a parameterized graph problem such that for any pair of graphs G_1 and G_2, and any integer $k \in \mathbb{N}$, we have $(G_1, k) \in L \vee (G_2, k) \in L \iff (G_1 \cup G_2, k) \in L$, where $G_1 \cup G_2$ is the disjoint union of G_1 and G_2. Then L is compositional.

Examples

- k-Path, k-Cycle, k-Cheap Tour, k-Exact Cycle, and k-Bounded Treewidth Subgraph
- k, σ-Short Nondeterministic Turing Machine Computation (Needs work)
- Many recent examples, Bodlaender, Kratch, Lokshantov, Saurabh etc. Also using (poly,poly)-reductions, co-nondeterminism, small interactive protocols, etc.

AND-composition and Distillation

- A current PhD student Andrew Drucker from MIT who has shown this also implies collapse. This implies all the below don't have poly kernels. The proof is remarkable.
- Applications: Graph width metrics:
- CUTWIDTH, TREEWIDTH, PROBLEMS WITH TREEEWIDTH PROMISES, EG.. COLOURING

OTHER RESULTS

- BDFH show that there are problems in ETP (FPT in time $O^*(2^{O(k)})$) without polynomial time kernels.
- Fortnow and Santhanam: Satisfiability does not have PCP's of size polynomial in the number of variables unless PH collapse.
- The Harnik-Noar approach to constructing collision resistant hash functions won't work unless PH collapses.
- Burhmann and Hitchcock: There are no subexponential size hard sets for NP unless PH collapses. (Ie many hard instances)
- Chen Flum Müller: Many results, e.g. parameterized SAT has no subexponential "normal" (strong) kernelization unless ETH fails.

- Using transformations, Bodlaender, Thomass'/ e and Yeo show that DISJOINT CYCLES, HAMILTON CIRCUIT PARAMETERIZED BY TREEWIDT H etc don't have poly kernels unless collapse.
- Also the important DISJOINT PATHS, famously FPT by Robertson and Se ymour.
- Similarly using Dell-Mecklebeek Kratz showed the non-poly-kernelizability of k-RAMSEY.
- Fernau et. al. have shown that there are problems with Poly Turing Kernels but no poly kernels unless collapse.(!), and these are natural related to spanning trees (Namely DIRECTED k LEAF SPANNING TREE).

Turing kernelizations

- Possible to avoid the material above. e.g. Binkele-Raible, Fernau, Fomin, Lokshantov, Saurabh and Villanger, k-Leaf Out Tree (directed spanning tree with k-leaves)
- The rooted case has a poly kernel.
- The unrooted case does not unless .....
- So it has a poly Turing Kernel
- Now lower bounds by recent work on completeness.

Definition (Turing Kernelization)

A *Turing Kernel* consists of

1. (i) Three parameterized languages L_1 and L_2 (typically $L_1 = L_2$) with $L_i \subset \Sigma^* \times \mathbb{N}$ and $L_3 \subseteq \Sigma^* \times L_1$
   (ii) and a computable function g
   (iii) and a polynomial time computable function $f : \Sigma^* \times \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$, $\langle \sigma, \tau, k \rangle \mapsto \langle \rho, k' \rangle$ with $|\tau| \leq |\sigma|$ *and* $|f(\langle \sigma, \tau, k \rangle)| \leq g(k)$ such that
   (iv) for all σ, τ, k,

$$\langle \sigma, \tau, k \rangle \in L_3 \text{ iff } \langle \rho, k' \rangle \in L_2.$$

2. Plus an oracle Turing procedure Φ, running in polynomial time on L_1, with oracle L_2, such that on input $\langle \sigma, k \rangle$, if the procedure queries $\langle \tau, k \rangle$ then it answers yes iff $f(\langle \sigma, \tau, k \rangle) \in L_3$

- The idea is that on input $\langle \sigma, k \rangle$ Φ works like a normal polynomial time machine except on oracle queries, it converts the query to a query of the kernel *determined by the query* τ.
- In the case of k-LEAF OUT BRANCHING,
  (I) L_1 are pairs $\langle G, k \rangle$ consisting of digraphs with k or more leaf outbranchings.
  (II) L_2 are pairs $\langle \hat{G}, k \rangle$ consisting of *rooted* digraphs (the input $\hat{G}$ would specify a root r) with with k or more leaf outbranchings.
  (III) L_3 are triples $\langle r, G, k \rangle$ consisting of yes instances of whether G has a k or greater leaf outbranching rooted at r.

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)
- Confronting intractability via parameters, Downey Thilikos, Computing Reviews
- Fundamentals of Parameterized Complexity, Downey-Fellows, this year.

What should you do?

- You should buy that new wonderful book...(and its friends)
- Thanks!