Basic Parameterized Complexity Primer

Rod Downey
Victoria University
Wellington

San Diego, Joint Meetings, January, 2012

THIS LECTURE:

- Basic Definitions
- Basic Hardness results
- Kernelization lower bounds

Parameterized Complexity

- 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.

- How to explain practical behaviour of combinatorial problems?
- When does a problem come equipped only with size as the only aspect of the imput you know? Never!
- Think of iteratively designed objects, any width metric you have met, genus, etc.
- How to attack a problem via knowledge of the structure of the input? Are there standard design tools
- How to show that thay are optimal at least up to current knowledge?

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 $\mathfrak{A}$ 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.

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)

Speeding things up

- Cleverer reduction rules (e.g. branch on higher degree vertices)
- 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.
- Jianer Chen and others use this in many FPT algorithms such as the state of the art for FEEDBACK VERTEX SET and VERTEX COVER. $O^*(1.2745^k)$(Chen10 using this, iterative compression, struction, measure and conquer, and other methods).
- Smaller kernels ; Nemhauser-Trotter has a size $2k$ kernel using LP relaxation.

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 $O^*(1+\epsilon)^k)$ (O^* looks only at the exponential part) algorithm is possible unless ETH fails.
- ETH n-valued 3SAT is not in DTIME($2^{o(n)}$).

CROWN REDUCTION RULES

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 $(G, k) \rightarrow (G', k - |H|)$.

- 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.

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.

ITERATIVE COMPRESSION

- 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.

- 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 vertices and get a acyclic graph?) (Cao, Chen, Liu, 2009).

Less practical algorithms

- colour coding for e.g. length k path.
- treewidth and other width metrics. Look at graphs which are treelike and use dynamic programming or automata.
- 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.
- Excluding minors.
- Logical metatheorems based on formal problem description

Hardness and Intractability

- 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.

OF RELEVANCE TO THIS MEETING

- Work on online algorithms e.g. online colouring pathwidth k graphs with $3k + 1$ colours (Kierstead-Trotter)
- Also online approximation: BIN PACKING.
- Online width metrics (Downey-McCartin). e.g. persistence.
- FPT (online) approximation, ie no k or certificate for $f(k)$.
- FPT incremental algorithms, e.g. Harting and Niedermeier LIST COLOURING

ANALOG OF COOK'S THEOREM

- 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.

W-HIERARCHY

- 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.

The Minimob

- 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

XP-OPTIMALITY

- 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.
- New generic techniques for showing that problems don't have kernels assuming PH does not collapse. e.g. k-PATH. TREEWIDTH (by a current PhD student at MIT).

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!