Basic Parametric Complexity I: Intractability

Rod Downey
Victoria University
Wellington

This lecture:

- Basic Definitions
- Classical Motivations
- Basic Hardness results

WHAT'S IN THIS FOR ME?

- 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.

- Lower bounds in P are really hard to come by. But this theory will alow you establish infeasibility for problems in P, under a reasonable complexity hypothesis.
- Also it will indicate to you how to attack the problem if it looks bad.
- It is thus both a positive and negative tool kit.

I'M DUBIOUS; EXAMPLE?

- Below is one application that points at why the completeness theory might interest you.
- The great PCP Theorem of Arora et. al. allows us to show that things don't have PTAS's on the assumption that P$\neq$NP.
- Some things actually do have PTAS's. Lets look at a couple taken from recent major conferences: STOC, FOCS, SODA etc.

- 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)	$O(n^{1021570})$

TABLE: 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.

- 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.)
- Moreover, if the PCP theorem applies,then using this basic hypothesis, there is also no PTAS.

- In the situation of the bad PTAS's the algorithms are polynomial. Polynomial lower bound are hard to come by.
- It is difficult to apply classical complexity since the classes are not very sensitive to things in P.
- Our idea in this case is to follow the NP analogy but work within P.

- What parametric complexity has to offer:
- Then assume the 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.

Efficient PTAS's

- Even if you are only interested in "classical" problems you would welcome a methodology that allows for "practical" lower bounds in P, modulo a reasonable complexity assumption.
- An optimization problem Π has an efficient P-time approximation scheme e (EPTAS) if it can be approximated to a goodness of $(1+\epsilon)$ of optimal in time $f(k)n^c$ where c is a constant and $k = 1/\epsilon$.

- (without even the formal definition) (Bazgan (Baz95), also Cai and Chen (CC97)) Suppose that Π_{opt} is an optimization problem, and that Π_{param} is the corresponding parameterized problem, where the parameter is the value of an optimal solution. Then Π_{param} is fixed-parameter tractable if Π_{opt} has an EPTAS.

- It is unknown if the PTAS's in the table have EPTAS's or not.
- In this talk, I will give a tourist guide through the area or parameterized complexity, making sure to mention a number of applications like the above to "classical" complexity.
- In this an the next talk, I would also like to highlight the way that parameterized complexity allows for an extended "dialog" with the problem at hand. (More on this soon).

- Others to use this technique include the following
- (Alekhnovich and Razborov (AR01)) Neither resolution not tree-like resolution is automizable unless $W[P]$ is randomized FPT by a randomized algorithm with one-sided error. (More on the hypothesis later)
- Frick and Grohe showed that towers of twos obtained from general tractability results with respect to model checking can't be gotten rid of unless $W[1] = FPT$, again more later.

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.

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 $\mathcal{O}$ in time $f(k)|G|$, a behaviour we call fixed parameter tractability, (Specifically $1.4^k k^2 + c|G|$, with c a small absolute constant independent of k.)
- Whereas the only known algorithm for DOMINATING SET is complete search of the possible k-subsets, which takes time $\Omega(|G|^k)$.

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,

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

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

- E.g. VERTEX COVER has $C = 1$. Vertex Cover has been implemented and shown to be practical for a class of problems arizing from computational biology for k up to about 400 (Stege 2000, Dehne, Rau-Chaplin, Stege, Taillon 2001) and n large.

- Keep in mind:an FPT language is in P "by the slice", and more: each k-slice is in the same polynomial time class via the same machine.
- Let L_k denote the k-th slice of L and $L_k^{(>m)}$ denote $\{\langle x, k\rangle : |x| > m\}$, the part of L_k from m onwards.
- (Cai, Chen Downey, Fellows; Flum and Grohe) $L \in$ FPT iff there is an algorithm M, a constant c, and a computable function g such that M witnesses that

$$L_k^{(>g(k))} \in DTIME(n^c).$$

- e.g. For Vertex Cover, g is about 2^k.
- Can do this with other classes, such as LOGSPACE, etc.

- The table below illustrates why this might be interesting.

	$n = 50$	$n = 100$	$n = 150$
$k = 2$	625	2,500	5,625
$k = 3$	15,625	125,000	421,875
$k = 5$	390,625	6,250,000	31,640,625
$k = 10$	1.9×10^{12}	9.8×10^{14}	3.7×10^{16}
$k = 20$	1.8×10^{26}	9.5×10^{31}	2.1×10^{35}

TABLE: The Ratio $\frac{n^{k+1}}{2^k n}$ for Various Values of n and k

- Note that we are using arbitarily $f(k) = 2^k$, and sometimes we can do better. (Such as the case of VERTEX COVER)
- So the FPT is interesting since it works better than complete search for problems where we might be interested in small parameters but large input size.

- 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.

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:

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

- Example: $W[1,s] \equiv_{fpt}$ Antimonotone $W[1,s]$.
- $W[1,s]$ is the problem based around weight k for circuits of depth 2, and maximum fanin for the top Or gates of size 2. (board) Specifically, $W[1,s]$ are the problems fpt reducible to weighted s-SAT.
- Red/Blue nonblocker: Input A Red/Blue bipartite graph $G = (V_R \cup V_B = V, E)$.
  Question is there a set of red vertices V' of size k, such that each blue vertex has a neighbour not in V'?
- $\Pi_{u \in V_B} \Sigma_{x_i \in N[u] \cap V_R} \overline{x_i}$.

- X is a boolean expressiuon in CNF of max clause size s.
- m clauses $C_1, \ldots, C_m$.
- Construct $G = (V_R \cup V_B, E)$ with a nonblocker of size $2k$ iff X has a satisfying assignment of weight k.

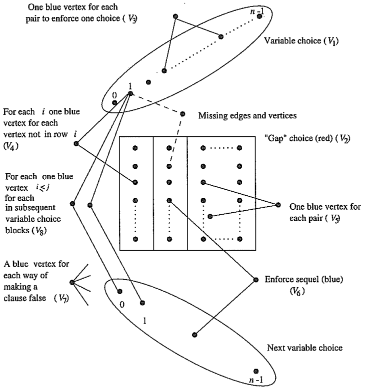

FIGURE 10.2. Gadget for **RED/BLUE NONBLOCKER**

The blue vertex set V_{blue} of G is the union of the following sets of vertices:

$V_3 = \{c[r_1, r_2, r_2'] : 0 \leq r_1 \leq k-1, 0 \leq r_2 < r_2' \leq n-1\}$,

$V_4 = \{d[r_1, r_2, r_2', r_3, r_3'] : 0 \leq r_1 \leq k-1, 0 \leq r_2, r_2' \leq n-1, 0 \leq r_3,$
$r_3' \leq n-1$ and either $r_2 \neq r_2'$ or $r_3 \neq r_3'\}$,

$V_5 = \{e[r_1, r_2, r_2', r_3] : 0 \leq r_1 \leq k-1, 0 \leq r_2, r_2' \leq n-1, r_2 \neq r_2',$
$1 \leq r_3 \leq n-k+1\}$,

$V_6 = \{f[r_1, r_1', r_2, r_3] : 0 \leq r_1, r_1' \leq k-1, 0 \leq r_2 \leq n-1,$
$1 \leq r_3 \leq n-k+1, r_1' \neq r_2 + r_3\}$,

$V_7 = \{g[j, j'] : 1 \leq j \leq m, 1 \leq j' \leq m_j\}$,

$V_8 = \{h[r_1, r_1', j, j'] : 0 \leq r_1 < r_1' \leq k-1$ and $j \geq j'\}$.

In the description of V_7, the integers m_j are bounded by a polynomial in n and k whose degree is a function of s, which will be described below. Note that since s

- Theorem $W[1,s] = W[1,2] =$ Antimonotone $W[1,2]$.
- Given an antimonotone $W[1,s]$ circuit C, we construct an antimonotone $W[1,2]$ circuit such that $W[1,s]$ has a weight k accepting input iff C' has a weight k' one where

$$k' = k2^k + \sum_{i=2}^{s} \binom{k}{i}$$

- Let the input variable $x[j], j = 1, \ldots, n$ to C, create new variables for each possible set of at most s and at least 2 of the $x[i]$'s. Let $A_1, \ldots, A_p$ be the enumeration of all such sets. These are the circuit inputs. Think of them as variables $v[i]$ representing A_i.
- Rearrange the circuit using these variables and enforcement variables.

- For each top or gate g choose the correct A_i's for the input lines to g in C, all negated of course.
- Now add an enforcement mechanism for consistency of the $v[i]$. This is done by 2^k copies of each of the $x[j]$, $x[j,d] : d = 1, \ldots, 2^k$. Now write out the exponentially many implications saying that the simulation is faithful.
- Details DF or see FG, for their approach using logic.

- CLIQUE is $W[1]$-complete as is INDEPENDENT SET.
- (DF, Cai and Chen) SHORT TURING MACHINE ACCEPTANCE is $W[1]$ complete.
- Generic reduction for hardness from CLIQUE. (write the vertices on the tape and check in $\binom{k}{2}$ moves if they are adjacent.) Membership is another generic simuation by a circuit.

- *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

- (Chandra and Merlin, 77) the complexity of query languages in the study of database theory.
- Vardi 1982 notes that classical complexity seemed wrong: suggested evaluation of a query when the size of the query was fixed as a function of the size of the database

- (Standard sort of problem) Input: A boolean query φ and a database instance I.
  Parameter: Some parameter of φ, such as the size of φ. (or its complexity etc)
  Problem: Evaluate φ in I.
- Yananakakis 1995 suggested that parameterized complexity good framework to address this.
- (Downey-Fellows-Taylor, 95 Papadimitriou-Yannakakis 97) The morning edition of the news is bad.
- Papadimitriou and Yannakakis systematically also looked at other parameters such as bounding the number of variables following ideas of Vardi (Va95). They looked at positive queries, conjunctive queries, first order theories and datalog ones and found them to be all $W[1]$ hard and at various levels of the W-hierarch y.
- Other analyses look at other parametric aspects and give even more bad news. (e.g. Demri, Laroussinie and Schnoebelen (DLS02).)

- You might well ask now, with “good” news like this provided by parameterized complexity, what use is it? You could argue that once we knew these problems were NP-hard and likely PSPACE complete. Now we know that even when you bound the obvious parameters then they are still hard!
- One interpretation is that we should learn to live with this by searching for new coping strategies

- The parametric point of view is to continue the dialog with the problem. To cope by finding new, and maybe more appropriate parameters.
- (Frick and Grohe (FrG02), Flum and Grohe (FG02a)) Let C be a class of relational database instances such that underlying graph of instances in C are any of the following forms: bounded degree, bounded treewidth, bounded local treewidth, planar or have an excluded minor. Then the query evaluation problem for the relational calculus on C is FPT.
- Frick and Grohe also looked at things beyong query languages, such as XML and temporal logics such as LTL and CTL* are used for specification languages for automated verification, proving sometimes practical FPT results.

- 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.
- Some recent work “lowers the hardness barrier”; perhaps giving better inapproximability results.

ETH

- 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
  Does it have a vertex cover of size d?
- Get minivertex 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.
- Fellows conjectures that PCP like techniques will show M[1]=W[1] using randomized reductions

Where else?

- We can formulate a notion of counting complexity and get $\# W[1]$ (Flum and Grohe, McCartin). A sample theorem:

Theorem (FG)

Counting k-cycles in a graph is $\# W[1]$-complete. (The existence problem is FPT, as we see next time.)

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)

Where else?

- EPT and bounded alternation. (Flum, Grohe and Weyer) A parameterized problem is in EPT iff it is solvable in time $2^{0(k)}|x|^c$.
- Partial formulation of an operator calculus such as *BP* analog. (Downey, Fellows, Regan) (Moritz Meuller) Unknown if analogs of, say, Toda's Theorem holds.
- A few results on parameterized logspace, and even less on parameterized PSPACE, except through game analogs.