Nonapproximation and the minimob

Rod Downey

Victoria University

Wellington

New Zealand

Joint work with

- Vladimir Estervill-Castro (Griffiths)
- Fran Rosamond, Mike Fellows, Elena Prieto-Rodrigez (Newcastle, NSW)

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 $c + 1.4^k|G|$, 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,

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

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

- E.g. VERTEX COVER has $C = 1$.

Why this makes a diffference

- The table below illustrates why this might be interesting.

- Note that we are using arbitarily $f(k) = 2^k$, and sometimes we can do better.
- In fact the current VERTEX COVER algorithm is feasible for $k = 70$, and large input. In fact the ETH group in Zurich have implementations that work for graphs of size 200 vertices of more.
- *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.*

Why else this interesting

- Lots of applications, and potential applications.
- Most computational problems involve several pieces of input, one or more of which may be a relevant parameter for various applications.
- For instance, VERTEX COVER can be often used to represent the *conflict graph* of some computational problem. Some Interesting parameters below.
- *Graph linear layout width metrics* are of interest in VLSI layout and

routing problems and have important applications for width values of $k \leq 10$. Interval graphs of pathwidth $k \leq 10$ have applications in DNA sequence reconstruction problems and in other evolutionary problems such as language evolution and phylogenic tree reconstruction.

- *Logic and database problems* frequently are defined as having input consisting a formula (which may be small and relatively invariant), and some other structure (such as a database) which is typically quite large and changeable. Formula size, or other aspects of formula structure may be a relevant parameter(See e.g. Yannakakis STOC). keys. Fellows-Koblitz demonstrated that this might not be a good idea since it wwas randomly FPT. The number of degrees of freedom in a robot motion-planning problem is commonly in the range $k \leq 10$. The number of wiring layers in VLSI chip manufacture is typically bounded by $k \leq 30$.

- *Network problems* may be naturally concerned with optimally locating a small number of facilities.

  These few examples are only suggestive and by no means exhaustive. There are myriad ways in which numbers that are small or moderately large (e.g., $k \leq 40$) arise naturally in problem specifications.

Whis is it interesting 3

- How Parameters arise in practice.
- In classical complexity a decision problem is specified by two items of information:
  (1) The input to the problem.
  (2) The question to be answered.
- In parameterized complexity there are three parts of a problem specification:
  (1) The input to the problem.
  (2) The aspects of the input that constitute the parameter.
  (3) The question.

- We have already seen *Graph Width Metrics*. These can often be *hidden* parameters in some computational problem. Examples include TREEWIDTH, CUTWIDTH, BANDWIDTH and PATHWIDTH. (For example, it has been proposed that the syntactic structure of sentences of natural languages can be modeled by dependency graphs of pathwidth less than 10 (Kornai and Tuza).)
- We have already seen *Implicit Parameters in Logic Programming, Compiler Design and Database Applications.*
- *Possible explanation of unpredicted*

tractability. Problems can be much easier in practice than one would expect from theoretical analyses. E. g. the problem of type inference arises in implementations of programming languages such as ML that are based on the polymorphic typed λ-calculus. (Henglein and Mairson) The problem is complete for deterministic exponential time, yet it has been widely noted that in practice the problem is *efficiently* solved by an algorithm that is known to be exponential in the worst case. An explanation for this discrepancy comes from noting that the logic formulas that occur in natural programs tend to have small bounded depth of *let*'s. For a parameter k bounding this depth, the problem is linear fixed-parameter tractable.

- Another example concerns the register allocation problem for imperative programming languages. This is a problem that is *NP*-complete in general. It has been shown that for structured programs (with short-circuit evaluation) the control-flow graphs have treewidth less than 10, which allows linear FPT algorithms based on this hidden structural parameter to be developed in, for instance, Alstrup, Lauridsen and Thorup.

- Such considerations give a different

possible explanation from those currently offered such as probabilistic behaviour.

- *Problems considered* in practice *come from "people" situations and hence can often have very small parameteres asscoiated with them because of our "brain" limitations.*

Why is this interesting 4

- A Distinctive Positive Toolkit. Lots of general techniques, begging for refinement. For example.

- Well-Quasi-Ordering. E. g. Robertson-Seymour. In their general form, these methods yield polynomial-time ($|G|^3$.) and fixed-parameter tractability results that are supremely impractical, primarily because they involve astronomical hidden constants. Currently primarily a complexity classification tool — a way of showing, often quite easily, that fixed-parameter tractability is possible.

- *Bounded Treewidth and Pathwidth,* Currently approaching practicality, Bodlaender: $f(k) = 2^{2^k}$.

- Related here are *automata theoretical methods*, practical depending on the presentation of the objects.

- *Color-Coding (Hashing)* and derandomization. (Alon, Yuster and Zwick) The method works quite well for certain kinds of problems, such as finding small subgraphs in a graph (such as paths, cycles and matchings). Practical: $f(k) = 2^k$ and the resulting algorithms offer significant practical parameter ranges and no sources of large hidden constants.

- *Elementary Methods. Search trees* and *reductions to a problem kernel* An interesting aspect of these approaches is that while they are simple algorithmic strategies, they are in some sense "new ideas" that might not immediately come to mind, *because they involve costs that are exponential in the parameter.*

Why is this intersting 5

- A new paradigm for seeking tractability.
- *The Deal with the Devil.*
- Okay so the problem is e.g. NP-PSPACE complete, or worse. What to do? Lets actively search for a parameter to absorb the intractability and make the problem feasible by the slice. One way is to use the hidden or otherwise parameters above.
- *Lichenism* BANDWIDTH appears *not* to be fixed parameter tractable. The best known algorithm for

k-BANDWIDTH requires time $O(n^k)$. What this means in the practical sense is that we must consider giving more to the devil for this problem.(e.g.)

BANDWIDTH II
Instance: A graph G and a positive integer k.
Parameter: The pair $(k, \Delta(G))$, where $\Delta(G)$ is is the maximum degree of G.
Question: Is the bandwidth of G at most k?

Why is this interesting 6

- *It cannot hurt.* Using these techniques have improved many known algorithms. For instance, using the technique of Kernelization first we improved VERTEX COVER from $2^k|G|$ to $|G| + 2^k.k^2$.

- Sometimes a FPT algorithm can outperform a poly algorithm. Using Matching one can only get $\Omega(|G|^3)$, but one might prefer $2^k|G|$ for instance.

- Sometimes they give the best alg we know anyway! (VERREX COVER)

- Many heuristic algs in the literature *already use* these techniques. The notion "points the way."

Okay so it does not seem tractable, then what?

- Natural Measures of *fixed parameter INtractability*. They tell you when this approach with this particular parameter will fail.

- The "classical" 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.

- Other $W[1]$ complete problems: k-INDEPENDENT SET, k-CLIQUE, k-3-SAT, etc.

Why $W[1]$ is interesting

- You can use it to prove unparameterized problems to not be feasible even though they are probably not NP-complete or are unknown. E.g. VC-DIMENSION, PRECEDENCE CONSTRAINED k-PROCESSOR SCHEDULING.
- Deep connections with approximation. $W[1]$-hardness means no good PTAS (Cai-Chen, Bagzan).
- Connections with logic. E.g. if in Max-SNP then FPT.
- Finally it's mathematically rich! and fun!

PTAS's

- A recurrent theme in "theory" is that hardness frontiers imply "try another strategy"
- A classic example (Garey and Johnson) Strong NP completeness means no fully polynomial time approximation.
- the famous Arora et. al. result on PCP showed that unless P=NP there are no PTAS's for a whole bunch of approximation problems.
- Our theory allows one to show that there are no *efficient* PTAS's. To wit:

Some recent PTAS's

How does this stuff help?

- The *trouble* is that the problems since they are in P cannot really be addressed by conventional complexity unless LOGSPACE completeness comes in.
- we use the parameter $k = \frac{1}{\epsilon}$.
- Fellows et. al. in recent work has shown that essentially this cannot be gotten rid of from the exponent by proving $W[1]$ hardness.
- Similarly Alekhnovich and Razborov have shown that Resolution is not axiomatizable unless the W-hirearchy collapses.

- An example:

  Graph k-CUT:

  INPUT: A Graph Gi and an edge weight function $w : E(G) \mapsto \mathbb{N}$. PARAMETER: k. QUESTION : is there a set $C \subseteq E$ s.t. $\sum_{e \in C} w(e) \leq m$ such that the graph $G - C$ has at least k connected components.

- Goldsmith and Hochbaum [FOCS 88] showed the problem is $\mathcal{O}(n^{k^2})$.

- Theorem Graph k-CUT is $W[1]$ hard. Thus assuming $w[1] \neq FPT$, there is *no* feasibel algo.

Lowering the Hardnesss Barrier

- lowering the hardness barrier
- The problem is $W[1]$ does not seem sensitive enough to capture the desired nonapproximations. In recent work we decided to parameterize the *size* of the (unrestricted) problem.
- constructing parametric reductions seems much more complex than the classical ones:

$$\langle G, k\rangle \mapsto \langle \widehat{G}, \widehat{k}\rangle$$

- the new work allows for "classical" type reductions $G \mapsto \widehat{G}$ provided that they are linear.

The Core Problem

- MINICIRCUITa SATISFIABILITY

  INPUT: $k, n \in \mathbb{N}$ in unary, and a circuit C of size $k \log n$.
  PARAMETER k QUESTION is C satisfiable?

  Theorem: MINICIRCUIT SATISFIABILITY is FPT iff there is no algo with running time $2^{o(n)}$ which determines whether boolean circuit C of size n is satisfiable.

- Now we can kind of "recycle" some classical reductions:

  the following are MINI-1 COMPLETE:

MINI3 SAT, MINI VERTEX COVER MINI INDEP SET MINI DOM SET etc:

Related Stuff

- other parametric classes e.g. parametric counting (McCartin, Grohe) $\#W[1]$, etc.
- Recently, Fellows and Chor have shown that $M[1] \subseteq FPT$ implies that k-independent set is $2^{o(k)}$.

Some Open Problems I

- *Industrial Strength FPT.*
- Case Study: VERTEX COVER.
- 1986. Fellows-Langston used Robertson-Seymour $f(k)n^3$, where $f(k)$ was a tower of 2's $500k$ high. and no "known" algorithm, in a technical sense.
- 1987. Johnson, $O(n^3)$ algorithm for VERTEX COVER could be based on a combination of tree-decomposition and finite-state dynamic programming techniques

- 1988 Fellows using Bounded Search Trees explicit $2^k|G|$
- 1989 S. Buss (Problem Kernel Method) $O(kn + 2^k k^{2k+2})$.
- 1992 R. Balasubramanian, R. Downey, M. Fellows and V. Raman $O(kn + 2^k k^2)$.
- 1993 Using Maximum Matching Papadimitriou and Yannakakis $O(3^k n)$ which is P-time for the classical problem if k is $O(\log n)$.
- 1996, Balasubramanian, Fellows and Raman $O(kn + (4/3)^k k^2)$.

- Develop methods to make the various FPT methods some of which are currently impractical all practical. Or show this is impossible in general and point out when it can be done.
- Good candidate: Bounded treewidth. Bring down Bodlaender's constant.
- possible candidate. More extensive use of Colour Coding and other hashing techniques to replace urrent ones.
- Good candidate. What about linear programming?
- Bad Candidate. WQO methods.
- Look at current *Heuristics* and suorce techniques there.

Some Open Problems 2

- *Connections with Classical Issues.* Exploring how the parametric and the classical intertwine.

- What's the best we can do?... any consequences to VERTEX COVER being $O(kn + \epsilon^k k^2)$, say. (Cai-Juedes) The minimob might be useful here

- Structure of FPT. Are there classifications here accoding to $f(k) = 2^k, 2^{2^k}, ...$ for example?

- Maybe arithmetization and connections with approximability relevant. There seem deep connections with approximability. (Also parametric approximability, e.g. is there an algo for DOMINATING SET that either easy no or gives on of size $2k$.)

- Any Mahaney-type result relating FPT to classical hypotheses. (Some weak connections known, Abrahamson-Downey-Fellows.)
- An *online* theory.

Some Open Problems 3

- Unknown classifications. There are several we feel important. here are a couple.

- GRAPH ISOMORPHISM
  Instance: Graphs G and H.
  Parameter: (Various, e.g. Max degree, max treewidth)
  Question: Is G isomorphic to H.

- (FPT?) TOPOLOGICAL CONTAINMENT
  Instance: A graph $G = (V, E)$.
  Parameter: A graph H.
  Question: Is H topologically contained in G?

- (FPT?) DIRECTED FEEDBACK VERTEX SET
  Instance: A directed graph $D = (V, A)$.
  Parameter: A positive integer k.
  Question: Is there a set S of k vertices such that each directed cycle of G contains a member of S?

- (FPT?) PLANAR DIRECTED DISJOINT PATHS

  Input: A directed planar graph G and k pairs $\langle r_1, s_1 \rangle, ..., \langle r_k, s_k \rangle$ of vertices of G.

  Parameter: A positive integer k.

  Question: Does G have k vertex-disjoint paths $P_1, ..., P_k$ with P_i running from $r - i$ to s_i?

  Known to be solvable in $O(n^{f(k)})$ time by Schrijver.

- (FPT?) PLANAR MULTIWAY CUT
  Instance: A weighted planar graph $G = (V, E)$ with terminals $\{x_1, \ldots, x_k\}$.
  Parameter: A positive integer k.
  Question: Is there a set of edges of total weight $\leq k'$ whose removal disconnects each terminal from all the others?

  The general version of this problem is NP-complete by Dahlhaus et. al. [**?**]. Best known complexity is $O((4^k)^k n^{2k-1} \log n)$ by [**?**] where it is asked if the problem is FPT

- (W[1] Hard?) FIXED ALPHABET LONGEST COMMON SUBSEQUENCE (LCS)
  Instance: k sequences X_i over an alphabet Σ of fixed size (such as $|\Sigma| = 4$ for DNA sequences), and a positive integer m.
  Parameter: A positive integer k.
  Question: Is there a string $X \in \Sigma^*$ of length m that is a subsequence of each of the X_i

  $W[t]$ hard for all t when the alphabet is bounded by the parameter k. Unfortunately this is not the problem that is really of interest to biologists. A simple dynamic programming algorithm that is widely used in the packages available to molecular biologists runs in time $O(n^k)$, and is frequently applied with $k = 5$ (with the help of supercomputers).

- (W[1] Hard?) BOUNDED HAMMING WEIGHT DISCRETE LOGARITHM
  Instance: An n-bit prime, a generator g of F_p^*, an element $a \in F_p^*$, a positive integer k.
  Parameter: A positive integer k.
  Question: Is there a positive integer x whose binary representation has at most k 1's (that is, x has a Hamming weight of k) such that $a = g$?

- (W[1] Hard?) CROSSING NUMBER
  Instance: A graph $G = (V, E)$.
  Parameter: k
  Question: Can G be embedded in the plane with at most k edges crossing?

- (W[1] Hard?) JUMP NUMBER
  Instance: A poset $P = (P, \leq)$.
  Parameter: A positive integer k.
  Question: Is the jump number of $P \leq k$?

Some Open Problems 4

- *Important Technical Questions.*
- E.g. Density of the degree structures, Ladner-Type theorem.
- Does the W-Hierarchy collapse, Does collapse propogate anyways?
- Oracle results, separations.
- Valiant-Vaziarini etc. Toda. analogs.

Your Strategy

- Buy that very reasonably priced book *"Parameterized Complexity"* (It is an excellent Present!)
- Some of the papers can be found in `www.mcs.vuw.ac.nz/research /math-pubs.shtml`
- Especially "Parameterized computational feasibility" and "Parameterized Complexity Analysis"
- my e-mail `rod.downey@vuw.ac.nz`

- Thank you very much for your attention.