Parameterized Complexity

29.07.2001 - 03.08.2001

organized by

Rod G. Downey (Wellington, New Zealand),
Michael R. Fellows (Newcastle, Australia),
Rolf Niedermeier (Tübingen), Peter Rossmanith (TU München)

Parameterized complexity is a new and promising approach to the central issue of how to cope with problems that are NP-hard or worse — as is so frequently the case in the natural world of computing. The key idea is to isolate some aspect(s) or part(s) of the input as the *parameter*, and to confine the seemingly inevitable combinatorial explosion of computational difficulty to an additive function of the parameter, with other costs being polynomial (called FPT complexity). An example is the NP-complete Vertex Cover ("conflict resolution") problem that is now known to be solvable in less than $1.29^k + kn$ steps for conflict graphs of size n. This algorithm works well for $k \leq 200$ and has several applications in computational biology.
Many important "heuristic" algorithms currently in use are FPT algorithms, previously unrecognized as such. Type-checking in ML provides another example. Although complete for EXPTIME in general, it is solved in practice in time $2^k + n$ for programs of size n, where the k is the nesting depth of declarations. Although many naturally parameterized problems are in FPT, some are not. The rich positive toolkit of novel techniques for designing and improving FPT algorithms is accompanied in the theory by a corresponding negative toolkit that supports a rich structure theory of parametric intractability. But the real excitement is in the rapidly developing systematic connections between FPT and useful heuristic algorithms — a new and exciting bridge between the theory of computing and computing in practice.
The organizers of the seminar strongly believe that knowledge of parameterized complexity techniques and results belongs into the toolkit of every algorithm designer. The purpose of the seminar was to bring together leading experts from all over the world, and from the diverse areas of computer science that have been attracted to this new framework. The seminar was intended as the first larger international meeting with a specific focus on parameterized complexity, and it hopefully serves as a driving force in the development of the field.

We had 49 participants from Australia, Canada, India, Israel, New Zealand, USA, and various European countries. During the workshop 25 lectures were given. Moreover, one night session was devoted to open problems and Thursday was basically used for problem discussions in smaller groups (one outcome of these is given in Gerhard Woeginger's contribution).
Schloss Dagstuhl and its staff provided an ideal setting for a very fruitful week of parameterized complexity studies. We are grateful to Dagstuhl and all participants for an exciting and inspiring time.
Finally, we thank Jochen Alber, Frederic Dorn, and Jens Gramm (all Tübingen) for helping in various ways to organize this meeting.

Contents

1 Towards Optimal FPT–Algorithms for Planar Graph Problems

Jochen Alber

A parameterized problem is called *fixed parameter tractable* if it admits a solving algorithm whose running time on input instance (I, k) is $f(k) \cdot |I|^{\alpha}$, where f is an *arbitrary* function depending only on k. Skimming through the literature, typical functions that appear for FPT-problems are, e.g., $f(k) = c^k$ (VERTEX COVER), $f(k) = c^{k^3}$ (TREEWIDTH), or even $f(k) = k^k$ (FEEDBACK VERTEX SET), or $f(k) = k!$ (MULTIDIMENSIONAL MATCHING).
In this talk I want to focus on obtaining a new qualitative behaviour of the exponential function f by presenting different techniques for designing algorithms where $f(k) = c^{\sqrt{k}}$ for various planar graph problems (see [1, 2, 3] for our recent work on this issue).
In particular, I will concentrate on an approach followed in [2], where we coined the notion of what we call the "Layerwise Separation Property" (LSP) of a planar graph problem. Problems having this property include PLANAR VERTEX COVER, PLANAR INDEPENDENT SET, or PLANAR DOMINATING SET. We prove that the LSP is sufficient for quickly computing a tree decomposition of a "yes"-instance of the problem with guaranteed treewidth of $O(\sqrt{k})$, which then can be used to solve the problem in the desired time. As a sideproduct of this, we derive some theoretical results relating, e.g., the domination number or the vertex cover number, to the treewidth of a planar graph.
Besides, I will report on first experimental results of our algorithms that were implemented using the LEDA library.

References

[1] J. Alber, H. Bodlaender, H. Fernau, and R. Niedermeier. Fixed parameter algorithms for planar dominating set and related problems. In *Proc. 7th SWAT*, vol. 1851 of *LNCS*, Springer, pp. 97–110, 2000. Full version accepted for *Algorithmica*.

[2] J. Alber, H. Fernau, and R. Niedermeier. Parameterized complexity: exponential speed-up for planar graph problems. In *Proc. 28th ICALP*, vol. 2076 of *LNCS*, Springer, pp. 261–272, 2001.

[3] J. Alber, H. Fernau, and R. Niedermeier. Graph separators: a parameterized view. In *Proc. 7th COCOON*, vol. 2108 of *LNCS*, Springer, pp. 318–327, 2001.

2 Efficient Algorithms for Horizontal Gene Transfer Problems

Michael Hallett

This talks covers our efforts to develop a model for *lateral gene transfer events* (a.k.a. horizontal gene transfer events) between a set of gene trees $T_1, T_2, \ldots, T_k$ and a species tree S. To the best of our knowledge, this model possesses a higher degree of biological and mathematical soundness than any other model proposed in the literature. Among other biological considerations, the model respects the partial order of evolution implied by S. Within our model, we identify an *activity parameter* that measures the number of genes that are allowed to be simultaneously active in the genome of a taxa and show that finding the most parsimonious scenario that reconciles the disagreeing gene trees with the species tree is doable in polynomial time when the activity level and number of transfers are small, but intractable in general.

3 Parameterized Problems in Automata Theory

Todd Wareham

Consider the Deterministic Finite State Automaton (DFA) Intersection problem, which, given a set A of DFA over an alphabet Σ, asks if there is a string $x \in \Sigma^*$ that is accepted by every DFA in A. This problem underlies various problems involving the composition and intersection of sets of finite state transducers (FST). All of these problems have applications within natural language processing; unfortunately, all of these problems are NP-hard and their best known algorithms compute the composite automaton by an iterative application of the pairwise state Cartesian-product construction, which requires $O(|Q|^{|A|})$ time where $|Q|$ is the maximum number of states in any finite state automaton in A. Can we do better?
In this talk, I apply techniques from parameterized complexity theory to assess the non-polynomial time algorithmic options for a subproblem of these problems, Bounded DFA Intersection, which requires x to be in Σ^k for a given $k > 0$ such that $k <= |Q|$. This analysis shows that relative to the set of problem-aspects $\{|A|, k, |Q|, |\Sigma|\}$, the only FPT algorithms are those whose running times are

non-polynomial in sets $\{|Q|, |A|\}$, $\{|\Sigma|, k\}$, or $\{|Q|, |\Sigma|\}$ or one of their supersets; all other possibilities have been ruled out by W-hardness results derived by reductions from parameterized versions of the Longest Common Subsequence and Dominating Set problems. All of the hardness results and various of the algorithms also apply to the original problems mentioned above, including the composition of restricted types of FST, e.g., p-subsequential FST.

References

[1] Wareham, H.T. (2000) "The Parameterized Complexity of Intersection and Composition Operations on Sets of Finite-State Automata." In Proceedings of the Fifth International Conference on Implementation and Application of Automata. Lecture Notes in Computer Science no. 2088. Springer-Verlag; Berlin.

4 Complexity and Management Decisions

Detlef Seese

The analysis of many complex problems in different areas of application shows that there seems to be a correlation between high complexity (NP-hardness) of the problem and the existence of large grids (as minors) in the underlying communication-structure defined by the problem. This criterion is specified and analysed in different areas of application. It is given a survey on related results in the areas of decidability of theories, graph algorithms, capital markets, risk-management, auctions, VLSI circuits, neural networks, software engineering and dynamical systems.

5 Algorithmic Aspects of the Feferman-Vaught Theorem

Johann Makowsky

(Based on joint work with B. Courcelle, J. Marino, J. Przytycki, E. Ravve, and U. Rotics)

A. Tarski initiated the study of the behaviour of validity of formulas in structures when passing to substructures, forming union of chains, or other algebraic operations. The Feferman-Vaught Theorem says how the truth value of a formula of First Order Logic in a generalized product of structures depends on the factors and the index set. For generalized sums this can be extended to Monadic Second Order Logic MSOL (Laeuchli, Shelah, Gurevich). For finite structures this can be used to check MSOL properties of structures and to compute polynomial invariants (graph polynomials) of structures provided the structure was built inductively using sum-like operations. Graphs of bounded tree width and bounded clique width are built inductively using such operations. We give a precise definition of sum-like operations on structures and survey algorithmic applications in the realm of graph polynomials and link polynomials.

6 Applying Parameterized Complexity to DNA Primer Design

Patricia Evans

(joint work with A. Smith and H.T. Wareham)

Designing universal DNA primers for a set of strings can be done by finding substrings that are within a short Hamming distance from a substring of each string. This problem is known as the Closest Substring problem, and is known to be NP-hard. The use of long sequences and short substrings, with few errors, make this problem a good target for parameterized complexity. Parameters for the number of strings (m), length of strings (n), length of substring (l), and

Hamming distance (k) are specified and examined. We provide a survey of the fixed-parameter tractability of some of the parameterized variants of the problem, and specify two algorithms. The first FPT algorithm is based on tabulation (with parameters ($|\Sigma|$, l, and k), and the second is based on sets of similar substrings with reduction to problem kernel (using parameters $|\Sigma|$, l, k, and m). Each algorithm is suitable for different specific applications.

7 Recent Progress in Computing the Stability Number

John Michael Robson

We present three improvements to a known recursive branching algorithm (Robson, J.Alg 7, (1986), p. 425) for computing the stability number of an n vertex graph which reduce the time complexity to $O(2^{cn})$ for c slighly smaller than 1/4. Firstly, we improve the analysis of the effectiveness of a dynamic programming approach to handling small ($\leq m$ vertex) induced subgraphs of non-regular graphs of degree at most 9. Since it suffices to consider induced subgraphs which are connected and have no vertex of degree 1, we can upper bound the number of such subgraphs by n times the weighted sum of the 8-ary trees of size up to m with weight equal to $2^{-number\ of\ leaves}$ (the previous algorithm used weights all equal to 1).
Secondly, a slightly more detailed case analysis of the neighbourhoods of a chosen vertex of degree at most 3 together with a more systematic use of constants for all cases bounding the factor by which the case is treated faster than the general case give better information on the time required when the graph has a vertex of low degree or when extra information is available on the stable sets to be considered.
Finally, where the minimum vertex degree d lies in $[4, 7]$, the algorithm considers the ball of radius 2 around one vertex A of degree d and applies the fact that there is a maximum stable set containing either A or at least two neighbours of A. For each of an exhaustive set of neighbourhood structures (i.e. partial specifications of this ball), a set of recursive calls is generated which minimises the time bound on the total computation provable by using the constants already derived as described above. The analysis of this algorithm requires consideration of more than $200,000$ neighbourhood structures and the $d!$ permutations of the neighbour vertices for each one and is, of course, carried out by computer.

8 Finite variable logics capturing parameterized complexity classes

Jörg Flum

(joint work with Martin Grohe.)

We present descriptive characterizations of the main parameterized complexity classes. For example, a problem Q on ordered graphs parameterized by natural numbers is in $W[1]$ if and only if for some s, every slice of Q is definable by a formula of the form $\exists x_1 \ldots \exists x_n \psi$, where ψ is a Boolean combination of formulas of least fixed point logic containing at most s variables and only one fixed point operator.

9 Probabilistic 3-SAT Algorithms

Uwe Schöning

We present a series of 3 algorithms for 3-SAT (which can be generalized to k-SAT) based on the concept of local search from some randomly selected initial assignment, and restart if no satisfying assignment is found. The first version uses random initial assignments and a deterministic backtracking procedure to search for a satisfying assignment within Hamming distance $n/4$ from the initial assignment. It achieves the bound (1.5^n) (where n is the number of variables). The second algorithm replaces the backtracking search by a random walk, and using a Markov chain analysis (gambler's ruin problem) one can show the improved bound $((4/3)^n)$. The third algorithm, finally, looks out for "independent" clauses and chooses the initial assignment for variables in independent clauses in a biased way. It can be shown that the obtained bound is (1.3301^n).

10 Two Deterministic Algorithms for k-SAT

Edward A. Hirsch

(a survey of joint papers with Dantsin, Goerdt, S.Ivanov, Kannan, Kleinberg, Papadimitriou, Raghavan, Schoening, Vsemirnov)

We survey two constructions for the derandomization of two families of randomized algorithms for k-SAT. The first construction derandomizes Schoening's random walk algorithm and uses covering codes. The second construction derandomizes the randomized unit clause elimination algorithm of Paturi, Pudlak and Zane, and uses projective geometry.

11 Recognizing More Random Unsatisfiable 3-SAT Instances Efficiently

Andreas Goerdt

(joint work with Joel Friedman)

It is known that random k-SAT instances with at least d_n clauses where $d = d_k$ is a suitable constant are unsatisfiable (with high probability). This talk deals with the question to certify the unsatisfiability of such a random 3-SAT instance in polynomial time. A backtracking based algorithm of Beame et al. works for random 3-SAT instances with at least $n^2/\log n$ clauses. This is the best result known by now.
We improve the aforementioned bound by Beame et al. to $n^{3/2+\varepsilon}$ for any $\varepsilon > 0$. Our approach extends the spectral approach introduced to the study of random k-SAT instances for $k \geq 4$ in previous work of the second author.

12 Homeomorphic Embedding of k-Connected Graphs in Graphs of Treewidth k

Torben Hagerup

(joint work with Arvind Gupta and Naomi Nishimura)

We study the problem of homeomorphic embedding of a guest graph G in a host graph H, i.e., of deciding whether G has a subdivision isomorphic to a subgraph of H. Matoušek and Thomas proved that the problem is NP-complete if G is connected and H is of treewidth k, for some constant k, and that the problem can be solved in polynomial time if G additionally is of bounded degree, but they left open the corresponding question for H of constant treewidth k and G being k-connected. We show that the latter problem can be solved in polynomial time for every fixed k, namely in $n^{\binom{k}{2}+O(k)}$ time. It is unknown whether (but seems unlikely to us that) the problem is fixed-parameter tractable with parameter k. The central part of our argument is a lemma showing that for tree decompositions of H of a particular kind, two vertices u and v are separated in H by the vertices in the bag of a node x of bag size k if and only if nodes whose bags contain u and v are separated by x in the tree decomposition of H. This enables us to reduce the number of combinations of partial solutions that need to be considered at x from a potentially exponential to a polynomial level.

13 Fast Fixed-Parameter Tractable Algorithms for Nontrivial Generalizations of Vertex Cover

Prabhakar Ragde

(joint work with Naomi Nishimura and Dimitrios Thilikos)

Our goal in this work is the development of fast algorithms for recognizing general classes of graphs. We seek algorithms whose complexity can be expressed as a linear function of the graph size plus an exponential function of k, a natural parameter describing the class. In particular, we consider the class $\mathcal{W}_k(\mathcal{G})$, where

for each graph G in $\mathcal{W}_k(\mathcal{G})$, the removal of a set of at most k vertices from G results in a graph in $\mathcal{G}$. (If $\mathcal{G}$ is the class of edgeless graphs, $\mathcal{W}_k(\mathcal{G})$ is the class of graphs with bounded vertex cover.)
When $\mathcal{G}$ is a minor-closed class such that each graph in $\mathcal{G}$ has bounded maximum degree, and all obstructions of $\mathcal{G}$ (minor-minimal graphs outside $\mathcal{G}$) are connected, we obtain an $O((g+k)|V(G)| + (fk)^k)$ recognition algorithm for $\mathcal{W}_k(\mathcal{G})$, where g and f are constants (modest and quantified) depending on the class $\mathcal{G}$. If $\mathcal{G}$ is the class of graphs with maximum degree bounded by D (not closed under minors), we can still obtain a running time of $O(|V(G)|(D+k) + k(D+k)^{k+3})$ for recognition of graphs in $\mathcal{W}_k(\mathcal{G})$.
Our results are obtained by considering minor-closed classes for which all obstructions are connected graphs, and showing that the size of any obstruction for $\mathcal{W}_k(\mathcal{G})$ is $O(tk^7 + t^7k^2)$, where t is a bound on the size of obstructions for $\mathcal{G}$. A trivial corollary of this result is an upper bound of $(k+1)(k+2)$ on the number of vertices in any obstruction of the class of graphs with vertex cover of size at most k. These results are of independent graph-theoretic interest.
(This work was also reported at WADS 2001.)

14 Exact Solutions for Closest String and Related Problems

Jens Gramm

(joint work with Rolf Niedermeier and Peter Rossmanith)

Closest String is one of the core problems in the field of consensus word analysis with particular importance for computational biology. Given k strings of same length and a positive integer d, find a "closest string" s such that none of the given strings has Hamming distance greater than d from s. Closest String is NP-complete. In biological practice, however, d usually is very small.
We show how to solve Closest String in linear time for constant d (the exponential growth in d is $O(d^d)$). We extend this result to the closely related problems d-Mismatch and Distinguishing String Selection. Moreover, we give a linear time algorithm for Closest String when $k = 3$ and d is arbitrary. Finally, the practical usefulness of our findings is substantiated by some experimental results and an application in primer design.

15 Parameterized Complexity of Type Checking Logic Programs

Witold Charatonik

Regular types are known in logic programming already for more than 25 years, but usually their definition is restricted to tuple-distributive regular sets of trees, that is, sets recognizable by top-down deterministic tree automata. Types defined this way were considered (by the logic-programming community) to be more efficient than general regular types. On the other hand the automata-theory community argued that general types are more expressive and the complexity of main algorithms (type inference and type checking) is the same (EXPTIME) in both cases.
In this talk we show that there is indeed a difference: the type checking problem for regular directional types for logic programs is fixed-parameter tractable if the types are restricted to be tuple-distributive, and is fixed-parameter intractable in the general case.

16 Generalized Model-Checking Problems

Martin Grohe

A fundamental algorithmic problem, playing an important role in different areas of computer science, is the following model-checking problem:
Given a finite relational structure A and a formula F of some logic L, evaluate F in A.
The name model-checking is most commonly used for the appearance of the problem in automated verification. However, the problem of evaluating a query against a finite relational database is of the same type. Constraint satisfaction problems in artificial intelligence can also be seen as model-checking problems. Moreover, many of the best-known algorithmic problems can be directly translated into model-checking problems. Often, we are not only interested in a model-checking problem itself, but also in certain variants, such as counting problems, which we refer to as generalized model-checking problems.
Generalized model-checking problems admit a natural parameterization in terms of the size of the input formula. I want to argue that parameterized model-

checking problems provide a natural framework for parameterized complexity theory and then discuss a number of recent results on the parameterized complexity of generalized model-checking problems for first-order logic.

17 Structural Aspects of Parameterized Complexity, with an "Expectation Principle"

Kenneth W. Regan

We study the complexity of parameterized languages L along other "gradients" $k(n)$ of the parameter k besides fixed ones. Complexity along these gradients is uniquely well-defined if we use the size of circuits C_{nk} computing $L_{nk} = \{x : |x| = n, < x, k >\in L\}$ as the cost measure, though it is usual to speak of TM running time. Call L "normal" if for all x, the set of values k such that $< x, k >\in L$ is contiguous. Then we can associate to L one or both of the optimization problems Max_L [Min_L] by: given x, maximize [minimize] k such that $L(x, k)$ holds. Given a distribution D on instances x of a given size n, define one or both of

$EMax_L(n) = \sum_{x \text{ of size } n} Max_L(x) * \Pr_D(x)$
$EMin_L(n) = \sum_{x \text{ of size } n} Min_L(x) * \Pr_D(x)$.

When we omit mention of D, it is understood to be a "natural"—usually uniform—distribution on the instances. Then we can state the following informal "Expectation Principles":

EP1: If $EMax_L(n) = \Omega(n^e)$ or $EMin_L(n) = \Omega(n^e)$ for some $e > 0$, then L should be fixed-parameter tractable.
EP2: If L is W[1]-hard [and in the W[t]-hierarchy], then $EMax_L(n)$ or $EMin_L(n)$ should be bounded above by a polynomial in $log(n)$.

We observe that a great many of the parameterized problems in the Downey-Fellows monograph, mainly those over graphs or strings or matrices (which have natural uniform distributions), abide by these principles. There are some technical difficulties, such as "duality" whereby the parameter "k" stands for $n - k$

or some other quantity, and exceptions such as CUTWIDTH and BANDWIDTH both having $EMin(n) \quad = n$ for uniformly random n-vertex graphs, but only CUTWIDTH is FPT while BANDWIDTH is W[1]-hard. The following results, building lightly on the subsequent talk by Venkatesh Raman, offer both support and exception for the principles:

(1) Let Π be a hereditary graph property, and let L be the parameterized language with $Max_L(G)$ = maximize the number of nodes in G that induce a subgraph with property Π. Then L is W[1]-hard iff $EMax_L(n)$ is asymptotic to $2log(n)$ under uniform distribution on n-vertex graphs, and L is FPT otherwise.
(2) Instead define L via $Min_L(G)$ = minimize the number of nodes whose removal leaves a graph in Π. Except for the "all" property, $EMin_L(n)$ is always asymptotic to n. It is known that whenever Π is defined by a finite set of forbidden subgraphs then L is FPT, in agreement with EP1, but when Π is the set of 3-colorable graphs, L is hard for the W[t] hierarchy and hence in partial exception to EP2.

We seek deeper connections to EP1 and EP2, and to study the complexity along gradients defined by $EMax_L(n)$ or $EMin_L(n)$.

18 Deciding Hamiltonicity in Graphs of Bounded Treewidth

Walker M. White

Courcelle's Theorem states that, for any fixed k and any monadic second order property of graphs, we can construct an automaton that accepts exactly the parse strings of graphs of treewidth k that have this property. The proof of this theorem gives us a method for implementing the automaton; each state corresponds to a test set, or a collection of graphs that represent possible extensions of the parse string.
The difficulty in implementing Courcelle's Theorem is that the automata can be quite large. For example, the test sets in the automaton for deciding whether a graph of treewidth k is Hamiltonian or not have $2^t t!$ many elements. This means that the naive implementation of this automaton has $2^{2^t t!}$ many states. One possible solution to this problem is to avoid constructing the entire automaton

while processing the parse string. Instead we build the automaton dynamically, only constructing states as they are needed.
We demonstrate how to use the test sets to dynamically construct the automaton for Hamiltonicity. This gives us an algorithm for determining whether or not a parse string of length n for a graph of treewidth k is Hamiltonian in time $O(2^t t! n)$. We conclude with a discussion of the difficulty in applying these techniques to Courcelle's Theorem in general.

19 Parameterized Complexity of Finding Subgraphs with Hereditary Properties

Venkatesh Raman

(joint work with Subhash Khot, Princeton University, USA)

We consider the parameterized complexity of the following problem: Given a graph G, an integer parameter k and a non-trivial hereditary property Π, are there k vertices of G that induce a subgraph with property Π? This problem has been proved NP-hard by Lewis and Yannakakis. We show that if Π includes all trivial (edgeless) graphs but not all complete graphs or vice versa, then the problem is complete for the parameterized class $W[1]$ and is fixed parameter tractable otherwise. Our proofs of both the tractability and hardness involve nontrivial use of the theory of Ramsey numbers.

20 On the Parameterized Complexity of Layered Graph Drawing

Naomi Nishimura

(Joint work with V. Dujmovic, M. Fellows, M. Hallett, M. Kitching, G. Liotta, C. McCartin, P. Ragde, F. Rosamond, M. Suderman, S. Whitesides, D. Wood)

We consider graph drawings in which vertices are assigned to layers and edges are drawn as straight line-segments between vertices on adjacent layers. We prove that graphs admitting crossing-free h-layer drawings (for fixed h) have bounded pathwidth. We then use a path decomposition as the basis for a linear-time algorithm to decide if a graph has a crossing-free h-layer drawing (for fixed h). This algorithm is extended to solve a large number of related problems, including allowing at most k crossings, or allowing at most r edge deletions to leave a crossing-free drawing (for fixed k or r). If the number of crossings or deleted edges is a non-fixed parameter then these problems are NP-complete. For each setting, we can also permit downward drawings of directed graphs and drawings in which edges may span multiple layers, in which case the total span or the maximum span of edges can be minimized. In contrast to the so-called Sugiyama method for layered graph drawing, our algorithms do not assume a preassignment of the vertices to layers.

21 A Fast Parameterized Face Cover Algorithm

Faisal N. Abu-Khzam

(joint work with Michael A. Langston)

A face cover of a plane graph, G, is a set of faces whose boundaries contain all vertices of G. When k is fixed and a face cover of size at most k exists, finding such a cover can be accomplished in linear time. Both the $O(12^k n)$ method of Downey-Fellows and the $O(c^{\sqrt{k}}n+n^2)$,where $c = 3^{36\sqrt{34}}$ method of Alber et al rely on reductions to planar dominating set. We present a direct $O(5^k n)$ algorithm.

22 Call Control with k Rejections

Thomas Erlebach

Given a set of connection requests (calls) in a communication network, the call control problem is to accept a subset of the requests and route them along paths in the network such that no edge capacity is violated, with the goal of rejecting as few requests as possible. For the problem of computing a solution that rejects at most k requests, we give FPT algorithms for tree networks with arbitrary capacities and for trees of rings with unit capacities.

23 Parametric Aspects of Parallel Complexity Theory

Klaus-Jörn Lange

The use of reducibilities of polynomial growth in parallel complexity theory often leads to inadequate models which are unable to measure appropriately parameters like speed-up or efficiency. On the other hand, the use of linear time reducibilities doesn't yield any systematic framework adequate for the classification of parallel problems. Parameterized complexity offers tools to handle these questions. The talk presents a rather old idea to apply reductions of linear growth *slicewise* to languages. This approach is demonstrated by the example of fixed vs. general membership problems. This allows one for example to give evidence for certain fixed wordproblems to lie in P but not in a fixed $DTIME(n^k)$. A proof of this fact would imply $P \neq NP$!

24 Parameterized Counting

Venkatesh Raman

(joint work with V. Arvind, IMSc. Chennai)

We look at the counting versions (both exact and approximate) of many parameterized problems whose decisions versions are fixed parameter tractable. We describe examples of fixed parameter tractable problems, whose counting versions are also fixed parameter tractable, as well as examples whose counting versions are W[1]-hard.

25 News on the ICALP'2001 paper by Liming Cai and David Juedes

Gerhard J. Woeginger

The paper "Subexponential parameterized algorithms collapse the W-hierarchy" by L. Cai and D. Juedes (Proceedings of ICALP'2001, Springer LNCS 2076, pp. 273-284, 2001) states the following main result: *In case some MAX SNP-hard problem can be solved in subexponential time, then W[1]=FPT*. On thursday Aug/02/2001 (during the Dagstuhl seminar on parameterized complexity) a working group of ten people met, and discussed, and tried to understand the arguments of this paper. The talk summarizes the observations and conclusions of this working group.
(1) The proof of Lemma 3 (pages 278 and 279) is fatally flawed. The removal of the conflicting unit clauses (x) and $(\neg x)$ messes up the calculations of the argument.
(2) There is no easy repair of the proof of Lemma 3. It can be shown that if for some $c \geq 3$ and some $r > 1/2$ the problem MAX c-SAT$^{(r,1)}$ is contained in XP, then P=NP. For that reason, any argument along the lines of the current proof should be doomed.
(3) The whole proof of the main result of Cai and Juedes breaks down.
(4) There might be hope to save the main result by centering the argument not around MAX c-SAT, but around the following parameterized variant of Vertex

Cover: Given a graph $G = (V, E)$ and a parameter k, does there exist a vertex cover with $k \log |V|$ vertices?

26 Open Problems

Benny Chor

Constructing phylogenetic (or evolutionary) trees from biological data is a classical problem in biology, and it still is a major challenge today. Most realistic formulations of the problem, which take errors into account, give rise to hard computational problems. Here, we concentrate on one specific method: quartet based tree reconstruction.
The input is a list of m quartets over n species. Each quartet is an unrooted binary tree on four species. A given quartet is consistent with a binary unrooted tree T if the subtree induced by T on the four leaves is the same as the given quartet. The goal is to construct a binary tree with the n species in its leaves, such that the total number of the satisfied quartets is maximized.
For a full input list ($m = \binom{n}{4}$, where each 4 tuple of species is represented by one quartet) it is easy ($O(n^4)$ time) to solve the decision problem "is there a tree satisfying all m quartets?". For smaller values of m, however, even the decision problem is NP complete. The corresponding maximization problem is MAX SNP hard.
Currently, the best EXACT algorithm runs in time $O(m3^n)$. Values of n in the range 20 to 30 are of significant biological interest (e.g. when considering mammalian evolution). A trivial randomized algorithm (pick a random binary tree) satisfies $m/3$ of quartets. This algorithm is easily derandomized.
From the point of view of parameterized complexity, it is known that given the full list ($m = \binom{n}{4}$), answering "is there a tree which satisfies all but k quartets" is in FPT (dependence on k is 4^k).

Interesting problems:

1. Smaller exponent for an exact algorithm (e.g. 2^n instead of 3^n).
2. A different parameterized approach, capable of handling a fixed proportion cm quartets' errors.
3. An FPT algorithm for satisfying $k + (m/3)$ quartets.

Rod G. Downey

Some problems:

1. Say somethings about the W-hierarchy. That is, make progress on any of the outstanding questions of separation, randomized collapse, collapse propagation, or the relationship between unknown memberships and collapse. For instance (as Mike Hallet asked at the workshop), one might hope to prove something like: if BANDWIDTH is in $W[P]$ then $W[P] = FPT$.

2. Many of the known FPT algorithms use treewidth, and seem genuinely feasible if one is given a tree decomposition. However, there seems to be no really practical tree decomposition algorithm. Find one.

3. Investigate the structure of FPT. Reductions that are polynomial time in both n (the input size) and k (the parameter) may be useful here, or perhaps parametric logspace reductions. Hierarchies of automata are another possibility.

4. Investigate the practicality of COLOR CODING. The constants are, at present, horrendous in the derandomized versions of this FPT technique.

5. Develop other general parameters to explain the tractability of inputs for classes of problems which are theoretically intractable. (Treewidth, for example, has proven to be quite general.) So the parameter is the "topology" of the input — are there other possibilities?

6. Is there any hope for the Cai-Juedes approach? Is there the possibility of an FPT algorithm with a $O(1 + e)^k$ additive exponential parametric contribution for VERTEX COVER, for any $e > 0$, or is there a theory of thresholds that can be shown to apply here? How does the classical theory of, e.g., approximation thresholds, relate to this issue?

Mike Fellows

Some More Problems:
(1) Although the Cai-Juedes paper of ICALP 2001 turned out to be flawed, it nevertheless raised some extremely interesting possibilities and fresh perspectives. In particular, they initiated a program of "optimal" FPT algorithms that is still

viable, despite the collapse of their main theorem, and is, I think, one of the most important new ideas in the field. To be specific, their main theorem would have provided the starting point for proving an endless horizon of concrete results such as:

(a) There can be no FPT algorithm for VERTEX COVER with a running time of $2^{o(k)}n^c$ unless FPT= $W[1]$.

(b) There can be no FPT algorithm for PLANAR VERTEX COVER with a running time of $2^{o(\sqrt{k})}n^c$ unless FPT= $W[1]$.

(c) There can be no FPT algorithm for PLANAR DOMINATING SET with a running time of $2^{o(\sqrt{k})}n^c$ unless FPT= $W[1]$.

The statement (c) would show that the new techniques and FPT results for planar problems presented at the workshop by Jochen Alber (and just a few weeks earlier at ICALP) are in some sense "optimal". This is still an extremely interesting and viable program, and results of Cai and Juedes that do hold up include that (a) implies (b) implies (c), and that all three statements are implied by the following:

Conjecture: The $k \log n$ VERTEX COVER problem is complete for $W[1]$.

There were various discussions at the workshop concerning the plausibility of this conjecture, and no consensus. My intuition favors the conjecture.

The natural way to attack this would be to try to encode the k-INDEPENDENT SET problem directly into the $k \log n$ VERTEX COVER problem. Some useful gadgets are available (based on constructions introduced by Cai and Juedes), but one may argue that a normal many:1 reduction is impossible. The usual sort of many:1 combinatorial reduction of (G, k) to (G', k') allows G to be computably recovered from G'. The difficulty this raises is that because VERTEX COVER is $2k$-kernelizable, we would then (normally) be provided the means to computably represent an arbitrary (e.g., Kolmogorov random) n-vertex graph G by a structure G' of size $2k' \log n$, for the proposed reduction, and this will not be possible. Elbow-deep in gadgeteering this issue arises quite concretely. So what does this mean for the above conjecture? It might only mean that we will have to use Turing reductions to show $W[1]$-hardness. The $k \log n$ VERTEX COVER problem belongs to $W[P]$ — this is not very difficult to prove — but how does one show that it belongs to $W[2]$? Those whose intuition is that the $k \log n$ VERTEX COVER problem is easier than $W[1]$ might start here!

(2) Polynomial time is tolerated as a mathematical model of tractability because it is usually quite well-behaved: if you can do it in polynomial time, then normally the exponent in your polynomial is at most 2 — in some rare extreme cases maybe 6 — and there is some sort of consensus that if your exponent is greater than 3 then your P-time algorithm is useless for practical purposes. If you present an algorithm that is $O(n^{1000})$, then most folks would not consider this as deserving the commercially honored label of "polynomial time" in the usual way. Eventually, the research community that is concerned with polynomial-time approximation schemes will begin to recognize this and begin to apply the in-

evitable tool of Bazgan's Theorem in order to elucidate when PTAS's cannot be improved to EPTAS's. (To state my prejudices quite plainly, I think it is a bit of a scandal that so little attention is currently being paid to the difference between a PTAS and an EPTAS.)

To review the issues, a PTAS computes a solution that is within a factor of $(1+\epsilon)$ of optimal in time that is polynomial for each ϵ. For example, Khanna and Motwani described three planar logic problems in a FOCS '96 paper that have PTAS's running in time $O(n^{35k^2})$, where $k = 1/\epsilon$. This is polynomial for each fixed ϵ, for sure, but for a 20% approximation we are looking at a polynomial-time algorithm that is $O(n^{875})$. It would be much better if we could get an EPTAS (efficient PTAS — terminology due to Cesati and Trevisan) with a running time of, for example, $O(2^k n^3)$, which would purchase quite good approximations. However, Cai, Fellows, Juedes and Rosamond have shown (manuscript available) that all three of these optimization problems considered by Khanna and Motwani have W[1]-hard associated parameterized problems, and so by Cristina Bazgan's theorem (later and independently proved by Cesati and Trevisan), they probably do not have EPTAS's. There are lots of PTAS's for which this rather well-motivated issue has not yet been explored. Get 'em while they're hot!

There is much more to explore here. The goodness of an approximation algorithm can be expressed in a variety of ways, with the PTAS regime of "... within a factor of $(1 + (1/k))$ of optimal," being only one of a wide variety of ways to frame the question. One would of course expect to have to pay more for a better approximation, and the goodness of the approximation (however this is expressed) is an obvious parameter. There really are just two basic ways that a polynomial running time can become more expensive for better approximations: by blowing up the degree of the polynomial, or by blowing up the constant in front. Some examples of how to use this perspective:

(a) Is it $W[1]$-hard to approximate a minimum dominating set in an n-vertex graph to within a factor of $(\log n)^{1/k}$? (David Johnson showed in 1974 that an approximation to within a factor of $(1 + \log n)$ can be computed in polynomial time.)

(b) Is it $W[1]$-hard to find a dominating set in a tournament that is within a factor of $(\log n)^{1/k}$ of minimum size, or can this be done in FPT time? This can easily be accomplished in polynomial time when $k = 1$.

(c) Is there a straightforward combinatorial reduction showing that it is $W[1]$-hard to approximate a minimum coloring of an n-vertex graph to within a factor of $n^{1/k}$ of optimal?

The reductions that might prove such things would presumably require new kinds of gadgets, but this is not necessarily a big obstacle — nothing of this sort has so far been attempted.

Henning Fernau

We study several versions of parameterized enumeration. The idea is always to have an algorithm which outputs all solutions (in a certain sense) to a given problem instance. Such an algorithm will be analyzed from the viewpoint of parameterized complexity. We show how to apply enumeration techniques in a number of examples. In particular, we give a fixed-parameter algorithm for the reconfiguration of faulty chips when providing so-called shared spares.
Kernelizations as well as search trees (which are the most prominent ways to devise fixed-parameter decision algorithms) are very useful techniques also for parameterized enumeration.
Remarkably, lower bounds and non-membership can be shown for several examples of enumeration problems and enumeration classes. In contrast, in the classical area of decision problems, mostly only relativized assertions of this kind are obtainable.

Judy Goldsmith

In Burago, et al.'s 1996 paper [1], they show that the optimal policy for a POMDP with deterministic observations and at most m states per observation can be approximated to within an additive constant ε in time polynomial in the size of the POMDP and $1/\varepsilon$. The complexity is $\mathcal{O}(n^m)$.
My question is where this problem falls in the W-hierarchy.
The problem can be restated as follows. A POMDP can be described as a graph $G = \langle V, E \rangle$ and a finite set of actions that determine transition probabilities from each vertex (for each $v \in V$ and $a \in A$, the probabilities $t(v, a, w)$ add up to 1 over all $w \in V$). There is a set of vertices G, the goal vertices; the goal of the controller is to pass through G with high probability. This task is made more difficult by the fact that the controller may have incomplete information about the current vertex at each time when an action must be chosen. This is represented by a coloring $h : V \to C$ such that for each $c \in C$, $|h^{-1}(c)| \leq m$. The controller only learns the color of the current vertex, though it may access the entire history of the graph traversal (as a sequence of colors) for its decision-making.
The proof can be sketched as follows. At each time step, the controller can maintain a probability distribution over states, indicating the probabilities of being in each possible state. Because of the bounded multiplicity of colors, at most m states have positive probability at any time. The value function mapping a control policy to expected values for each such distribution is continuous. For each color c, the m-simplex of probability distributions over the support of c

(the states mapped to color c) can be partitioned into $poly(m, 1/\varepsilon)$ many sub-simplices so that any value function will vary by at most ε over each sub-simplex. Then one can approximate the actual optimal value function on the corners of the sub-simplices, and take a linear extension of those values for interior points on the simplices.
Can this method be extended? Are there other methods?

References

[1] Dima Burago and Michel de Rougemont and Anatol Slissenko (1996). *On the Complexity of Partially Observed Markov Decision Processes* .Theoretical Computer Science; volume 157, number 2, pages 161–183.

Venkatesh Raman

Given a 2-CNF formula F with m clauses on n variables, and an integer parameter k, is there an assignment to the variables that satisfies at least $m-k$ clauses of F? Is this problem FPT or W-hard?
I know of at least three problems that (parametrically) reduce to this question.
This problem is clearly W[P]-hard for c-CNF formulas for $c > 2$, as we can reduce (in polynomial time) c-CNF satisfiability question to this problem even for constant k.

Ken Regan, Martin Grohe, Jörg Flum

Consider HITTING CLIQUE: graph G, parameter k, does G have a set U of k vertices such that every k-clique in G has nonempty intersection with U? This is in Flum-Grohe's class "A[2]", but not known to be in W[2] or in the W[t] hierarchy at all. Is it? More generally, what relationships can we establish among W[t], A[t], and Downey-Fellows-Regan's "H[t]" classes?

Todd Wareham

Robot Motion Planning. Given a robot in an environment composed of some set of obstacles and initial and final positions of the robot within this environment, the *motion planning problem* involves finding a sequence of motions that move the robot from the initial to the final position without intersecting any of the obstacles. Though polynomial-time algorithms are known for this problem for very limited kinds of robots, e.g., line segments, disks, rectangles (see [5] and references), the best known algorithm for arbitrarily complex robots requires $O(n^k(\log n \cdot d^{O(k)} + 2dk^{O(k^2)}))$ time [1], where k is the degrees of freedom of movement, n is the number of polynomials required to describe the surfaces of the robot and its environment, and d is the maximum degree of these polynomials [5]. The terms exponential in d and k in these running times are not daunting because values of d and k in practice are typically small, e.g., $d = 4$ for a polygonal robot in a planar polygonal environment and $k \leq 7$ for industrial robot arms. It is thought unlikely that algorithms such as that in [1] can eliminate the n^k term in their running time because such algorithms must compute all points in a special k-dimensional space called *FP space*, and the number of points in this space is $O(n^k)$ in the worst case [5, Theorem 3.1]. However, this does not rule out the existence of other algorithms that are fixed-parameter tractable relative to k.

A formal definition of the problem described above is as follows [3]:

d-Dimensional Euclidean generalized mover's problem (dD-GMP, $d \in \{2,3\}$)

Instance: A set O of obstacle polyhedra, a set P of polyhedra which are freely linked together at a set of linkage vertices V such that P has k degrees of freedom of movement, and initial and final positions p_I and p_F of P in d-dimensional Euclidean space.

Question: Is there a legal movement of P from p_I to p_F, i.e., is there a continuous sequence of translation and rotations of the polyhedra in P such that at each point in time, no polyhedron in P intersects any polyhedron in O and the polyhedra in P intersect themselves only at the linkage vertices in V?

Let k-dD-GMP denote the parameterized version of this problem in which k is the parameter. Reif [3] showed that 3D-GMP is $PSPACE$-hard. Cesati and Wareham [2] in turn used Reif's reduction to show that k-3D-GMP is W[SAT]-hard. This ruled out the existence of the most desirable type of FPT algorithms for motion planning. However, many open problems remain:

1. Does k-3D-GMP become fixed parameter tractable if additional problem-aspects describing the complexity of the robot's environment, *e.g.*, number / surface-complexity of obstacles, are added to the parameter? Reif's reduction essentially encodes a Turing machine's state in the robot and encodes

computations as paths through a hideously complex maze of obstacles. Such an obstacle-set is not a realistic model of instances of motion-planning problems encountered in practice.

2. Is k-2D-GMP fixed-parameter tractable? Much motion planning is done relative to robots that effectively operate in two dimensions on factory floors.

3. What is the fixed-parameter status of more realistic parameterized motion-planning problems that incorporate moving obstacles, optimality constraints on motion plans, and uncertain robot motion (see [3, 4, 5] and references)?

References

[1] Canny, J.F. (1987) *The Complexity of Robot Motion Planning.* The MIT Press; Cambridge, MA.

[2] Cesati, M. and Wareham, H.T. (1995) "Parameterized Complexity Analysis in Robot Motion Planning." In *Proceedings of the 25th IEEE International Conference on Systems, Man, and Cybernetics: Volume 1.* IEEE Press; Los Alamitos, CA. 880–885.

[3] Reif, J.H. (1987) "Complexity of the Generalized Mover's Problem." In Schwartz, J.T., Sharir, M. and Hopcroft, J. (eds.) (1987) *Planning, Geometry, and Complexity of Robot Motion.* Ablex Publishing Corporation; Norwood, NJ. 267–281.

[4] Reif, J.H. and Sharir, M. (1994) "Motion Planning in the Presence of Moving Obstacles." *Journal of the ACM*, **41(4)**, 764–790.

[5] Schwartz, J.T. and Sharir, M. (1990) "Algorithmic Motion Planning in Robotics." In J. van Leeuwen (ed.) *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity.* The MIT Press; Cambridge, MA. 391–430.