Parameterized Complexity News

The Newsletter of the Parameterized Complexity Community fpt.wikidot.com May 2009

Welcome

Frances Rosamond, Editor
Welcome to the Parameterized Complexity Newsletter. We expecially congratulate new graduates, and ask you to help them find postdoc or other positions.

The Newsletter is archived at the IWPEC website at (`http://www.scs.carleton.ca/~dehne/iwpec`) and at Mike Fellows' website `http://www.mrfellows.net`. The WIKI is located at `http://www.fpt.wikidot.com`. Additions are very appreciated as new great work in parameterized complexity is happening too quickly for only one person (me) to post to the WIKI.

The next Newsletter will include a report on the excellent Corsica meeting. An article by Michael Lampis, Georgia Kaouri and Valia Mitsou will describe new directions in treewidth, continuing work reported at ISAAC. The Table of FPT and Kernelization Races will return. A new plan is to add a picture to the first page of each newsletter (partly to differentiate them, other than by the date). Please send me a note if you have difficulty receiving the newsletter. I am experimenting with a new mailing program. Many thanks to many people for helping with this newsletter.

This Newsletter features an article by Iris van Rooij who develops a theory of cognitive psychology using parameterized complexity. The journal *Cognitive Science, 32, 2008* has a 45 page paper by Iris introducing the tractable cognition thesis. She has summarized some of those ideas for us here. We have an exciting article by Saket Saurabh on k-FEEDBACK ARC SET: FAST, and CHROMATIC CODING, a Report on the Adaptive, Output-Sensitive, Online and Parameterized Algorithms Dagstuhl Seminar by Nadja Betzler and Britta Dorn, and a New Ideas column by Mike Fellows.

Congratulations to all Authors

The ICALP and ESA accepted papers have been announced. Congratulations to everyone for the *outstanding* success. Special applause to Bergen with five papers accepted at ICALP, and four are coauthored by Saket Saurabh.

WG 2009: 35th Int Workshop on Graph-Theoretic Concepts in Computer Science, Montpellier, France, Jn 24-26. PC Workshop beforehand: **GRAAL 2009** "Graph Decompositions and Algorithms", Montpellier, France, Jn 22-23.

Happy Birthday Jan Kratochvíl

Over 40 researchers gathered at Charles University, Prague for a birthday celebration of Jan Kratochvíl. The Seminar concluded with a sumptous banquet, and lively piano playing by the guest-of-honour.

Figure 1: The Seminar concluded with a banquet, and piano playing by the guest-of-honour.

Contents of this issue:

IWPEC 2009

4th International Workshop on Parameterized and Exact Computation (IWPEC) 10-11 September 2009, co-located with ALGO/ESA. IT University of Copenhagen, Denmark.

Special IWPEC competition: prior to IWPEC, send to Fran (www.mathgypsie@yahoo.com) a song with a simple tune (that we can all sing) with lyrics changed to include parameterized complexity. Example: *Happy PC to you, Happy PC to you,...*. We will announce a location for a sing-along.
Guest Speakers: Hans Bodlaender and Noga Alon.

New Ideas Column

by Mike Fellows, University of Newcastle, AU

The Dagstuhl Workshop in April led to some interesting conversations with researchers in the areas of adaptive and on-line algorithms that led to some fresh perspectives on FPT kernelization — new and seemingly well-motivated questions.

In on-line algorithms, the input is gradually revealed and you have to start "solving" the instance incrementally. A key notion is the competitive ratio of an on-line algorithm, meaning the worst case, over all instances I, of the ratio $A(I)/opt(I)$, where $A(I)$ is the cost of the solution that the algorithm would produce, and $opt(I)$ is the minimal cost that could have been achieved if you were first given the entire input, and then solved it (called the optimal off-line cost), rather than having to solve it incrementally as I is revealed.

Something analogous occurs in FPT kernelization. A kernelization algorithm for an FPT problem consists of two things: a set $\mathcal{R}$, and an algorithm for applying them. When a rule is applied to an instance I, you typically get a modified instance I' that in turn may present (possibly many different) further opportunities to apply some reduction rule in $\mathcal{R}$ — we sometimes say that the reduction rules "cascade", resulting eventually in a kernelized instance. The analogy with the on-line situation is that at each step we have to decide which opportunity to exploit.

So we can make an analogous definition, studying the ratio of the kernel size that our algorithm for applying the rules of $\mathcal{R}$ might achieve, as compared with the kernel size that might be achieved by an optimal sequence of choices concerning reduction rule opportunties. For example, consider the "Buss kernelization" algorithm for Vertex Cover, where $\mathcal{R}$ consists of the two rules: (1) delete vertices of degree zero, and (2) the high degree rule: if $deg(v) > k$, then delete v and set $k' = k - 1$. It can be shown (unpublished result of Daniel Lokshtanov, Daniel Marx and Saket Suarabh) that a simple greedy algorithm to apply these rules can result in a kernelized instance that is a factor of k larger than could be achieved by an optimal sequence of applications of rules in $\mathcal{R}$.

We propose to call this ratio the *adaptivity* of a kernelization algorithm. If the ratio is 1, we will call such an algorithm *fully adaptive*. Interestingly, the greedy policy for applying crown reductions for the Vertex Cover problem can be shown to be fully adaptive.

The motivation is clear; we would like to know that our kerenelization algorithms will fully exploit the power of the reduction rules to reduce the size of the input.

Parameters that make Cognitive Work Light

by Iris van Rooij, Radboud University. Donders Institute for Brain, Cognition and Behaviour

A key objective of cognitive science is to characterize human cognitive capacities and their underlying brain mechanisms. Many believe that computational modeling can help achieve this goal. This is evidenced, for example, by the four computational modeling prizes awarded annually at meetings of the Cognitive Science society (See `http://cognitivesciencesociety.org/conference_overview.html`) Few know, however, that parameterized complexity theory can help computational modelers.

A foundational assumption of cognitive science is that cognitive capacities are computational capacities that are physically realized, e.g., by the brain. The assumption implies that cognitive capacities are limited by computational tractability. The majority of human cognitive capacities—such as, the formation of visual or auditory percepts, social judgments, motor responses, the understanding and production of spoken sentences—operate on a timescale ranging from milliseconds to minutes. Computational models better be computable fast, given human-scale resources, if they are to be explanatorily relevant for cognitive science [1, 3].

Complexity analyzes can guide the design of models of cognitive capacities by introducing model constraints that are necessary and/or sufficient for computational tractability. Several cognitive scientists have proposed to use classical complexity theories to this end. Yet, the univariate analyses characterizing classical complexity seem too crude for dealing with the multivariate nature of real-world inputs to cognitive processes. Parameterized complexity provides a natural refinement of the program. This refinement has been successfully implemented by myself and others in a wide variety of domains, including: similarity judgments [2], abductive reasoning [4], decision-making [5], analogical mapping [8], and problem solving [9]. Analyses have led to new insights about the conditions under which cognitive models can meet the tractability constraint. Such insights are valuable theoretical contributions to cognitive science. In several cases they also lead to new predictions that can be tested in behavioral experiments.

Readers of this newsletter, who likely aren't cognitive scientists, may be asking themselves the following question: This is all nice and well for cognitive scientists, but how is it of use for a parameterized complexity theorist? Besides the obvious advantages for parameterized complexity theory (e.g., that it successfully expands its domains of application), I believe the theory can benefit in a more fundamental way. The application domain of cognitive science is disanalogous to the traditional application domains of parameterized complexity and therefore raises different kinds of theoretical questions.

Whereas traditional domains involve the design of fixed-parameter tractable (fpt) algorithms in order to solve predefined NP-hard problems, in the domain of cognitive science the goal is to design a (fp-)tractable problem in order to model an unknown cognitive capacity. This difference has far reaching consequences for how one is to deal with intractability in the different domains. It also creates a niche for practical applications of parameterized reductions in cognitive science. For details on these and other characteristics of the cognitive science application, I refer to [6, 7].

Let me end with a set of questions that may trigger the interest of complexity theorists. Sometimes a problem is fpt for two or more distinct parameters. Plausibly, the mind/brain exploits many different parameters during computation, even if each individual parameter would be sufficient for fixed-parameter tractability. Can we build a theory of the added value of having two (or more) fpt parameters for the worst-case running time? Other questions follow from the assumption that human cognition is neurally realized: Can we develop theory that supports parameterized complexity analyses relative to a given neural architecture? Can we build a theory of neurally possible fpt-computations? Is neurally possible fpt-computation as powerful as general fpt-computation?

This is just a sample of questions raised by the application domain of cognitive science. I hope this serves as a guide to the application domain and as a stimulus for complexity theorists to help advance the modeling tools of cognitive science.

[1] Frixione, M. (2001). Tractable competence. *Minds and Machines, 11*, 379–397.

[2] Müller, M., van Rooij, I. & Wareham, T. (2009). Similarity as tractable transformation. To appear in the *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, Amsterdam, The Netherlands.

[3] Tsotsos, J. K. (1990). Analyzing vision at the complexity level. *Behavioral and Brain Sciences, 13*(3), 423-469.

[4] van Rooij, I. (2003). *Tractable cognition: Complexity theory in cognitive psychology.* Doctoral dissertation. Department of Psychology, University of Victoria, Victoria, Canada.

[5] van Rooij, I., Stege, U., & Kadlec, H. (2005). Sources of complexity in subset choice. *Journal of Mathematical Psychology, 49*(2), 160-187.

[6] van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science, 32*, 939-984.

[7] van Rooij, I. & Wareham, T. (2008). Parameterized complexity in cognitive modeling: Foundations, applications and opportunities. *Computer Journal, 51*(3), 385-404.

[8] van Rooij, I., Evans, P., Müller, M., Gedge, J. & Wareham, T. (2008). Identifying sources of intractability in cognitive models: An illustration using analogical structure mapping. In B. C Love, K. McRae, and V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society, Austin, TX: Cognitive Science Society* (pp. 915-920).

[9] Wareham, T. (2008). On the computational complexity of analogy-based models of problem solving: Implications and opportunities. Paper presented at the Workshop on New Perspectives on Human Problem Solving, Purdue University, November 8-9, 2009, USA.

Chromatic Coding and Universal (Hyper-) Graph Coloring Families

by Saket Saurabh, University of Bergen

In a seminal paper Alon, Yuster and Zwick [3] introduced the method of COLOR CODING as a tool to detect a k-sized subgraph F of constant treewidth in an input graph G in time $2^{O(k)}n^{O(1)}$. The color coding technique is typically applied when we are searching for a small structure S of size k in a larger structure X. We randomly color X with a set of k colors and show that if X contains S as a substructure then the probability that a particular copy of S in X has been colored with distinct colors, that is, S has become colorful, is roughly $e^{-O(k)}$. Finally applying dynamic programming on the colored structure X we find the desired colorful S in $2^{O(k)}n^{O(1)}$. Overall this leads to an algorithm with running time $2^{O(k)}n^{O(1)}$. Since the introduction of COLOR CODING, various applications and variations of it have been found. One of the most prominent variation is known as DIVIDE AND COLOR.

Recently, a novel variation of COLOR CODING, called CHROMATIC CODING has been introduced in [2] which has led to subexponential time algorithms for various problems on dense graphs. We explain the method of CHROMATIC CODING through an example of k-FEEDBACK ARC SET IN TOURNAMENTS. The problem is defined as follows.

> k-Feedback Arc Set in Tournaments (k-FAST)
> Instance: A tournament $T = (V, A)$,and an integer k.
> Question: Is there an arc set $S \subseteq A$ such that $|S| \leq k$ and $T \setminus S$ is acyclic?

The basic idea of Chromatic Coding is to randomly color the input tournament T in a way that the graph induced by the feedback arc set S is *properly colored*. That is, we consider the solution set S as a k-edge graph and require the random coloring to properly color this graph, that is the endpoints of every edge in this graph are colored with different colors. It can be easily shown that any graph on k edges can be properly colored with $O(\sqrt{k})$ colors. In fact, a random coloring with $O(\sqrt{k})$ colors properly colors the graph with probability at least $e^{-O(\sqrt{k})}$. So we startby coloring the input tournament T with $O(\sqrt{k})$ colors and hence with probability $e^{-O(\sqrt{k})}$ the feedback arc set we are looking for is properly colored. The next question to be addressed is how to find the desired S given the coloring of T. The trick is that every color class must induce an acyclic tournament, because we are not allowed to delete edges whose endpoints have the same color. Now we can exploit that an acyclic tournament has a unique topological ordering to give a dynamic programming or (divide and conquer) based algorithm to find the optimal properly colored feedback arc set. This algorithm has running time $|V|^{O(\sqrt{k})}$. Finally we use the fact that k-FAST has a polynomial kernel of size $O(k^2)$ to obtain a randomized algorithm with running time $2^{O(\sqrt{k}\log k)}n^{O(1)}$.

In order to derandomize our algorithm we construct a new kind of universal hash functions, that we coin *universal coloring families*. For integers m, k and r, a family $\mathcal{F}$ of functions from $[m]$ to $[r]$ is called a universal (m, k, r)-coloring family if for any graph G on the set of vertices $[m]$ with at most k edges, there exists an $f \in \mathcal{F}$ which is a proper vertex coloring of G. We give an explicit construction of a universal $(n, k, O(\sqrt{k}))$-coloring family $\mathcal{F}$ of size $|\mathcal{F}| \leq 2^{O(\sqrt{k}\log k)} \log n$.

The notion of universal coloring can be easily generalized to q-uniform hypergraphs. A hypergraph H is a pair $H = (V, E)$ where V is a set of vertices, and E is a set of non-empty subsets of V called hyperedges. If all edges have the same cardinality q, the hypergraph is said to be uniform or q-uniform hypergraphs. Given a hypergraph $H = (V, E)$, a function $f : V \to [r]$ is a proper vertex coloring of H if any edge $e \in E$ is not monochromatic under f. For integers m, k and r, a family $\mathcal{F}$ of functions from $[m]$ to $[r]$ is called a universal (m, k, r)-q-uniform coloring family if for any q-uniform hypergraph H on the set of vertices $[m]$ with at most k edges, there exists an $f \in \mathcal{F}$ which is a proper vertex coloring of H. We provide an explicit construction of a $(n, k, O(k^{1/q}))$-q-uniform coloring family $\mathcal{F}$ of size $|\mathcal{F}| \leq 2^{O(k^{1/q}\log k)} \log n$. Using $(n, k, O(k^{1/q}))$-q-uniform coloring family one can solve several other dense problems. Notable ones include Minimum Quartet Inconsistency (MQI) and Betweenness. We refer to [4] and [1] for the definitions of MQI and Betweenness, respectively. While the algorithm for MQI runs in time $2^{O(k^{1/4}\log k)}n^{O(1)}$, the algorithm for Betweenness runs in time $2^{O(k^{1/3}\log k)}n^{O(1)}$.

Algorithms based on Chromatic Coding run in subexponential time, a trait uncommon to parameterized algorithms. In fact, to the authors best knowledge the only parameterized problems for which non-trivial subexponential time algorithms are known are *bidimensional* problems in planar graphs or graphs excluding a certain fixed graph H as a minor. A preliminary version of the article containing Chromatic Coding, *universal coloring families* and its applications in obtaining sub-exponential time algorithm for k-FAST appeared in [2].

[1] Nir Ailon and Noga Alon: Hardness of fully dense problems. Inf. Comput. 205(8): 1117-1129 (2007).

[2] Noga Alon, Daniel Lokshtanov and Saket Saurabh: Fast FAST. To appear in the proceedings of ICALP 2009.

[3] Noga Alon, Raphael Yuster and Uri Zwick: Color-Coding. J. ACM 42(4): 844-856 (1995).

[4] Maw-Shang Chang, Chuang-Chieh Lin and Peter Rossmanith: New Fixed-Parameter Algorithms for the Minimum Quartet Inconsistency Problem. IWPEC 2008: 66-77.

Figure 2: At Dagstuhl, what is Magdalena Gruber saying to Britta Dorn? Nadja Betzler on left with Stefan Kratsch behind.

Adaptive, Output-Sensitive, Online and Parameterized Algorithms

by Nadja Betzler[1] and Britta Dorn[2], Friedrich Schiller University, Jena[1], University Tuebingen[2]

Dagstuhl Seminar 09171, organized Jrmy Barbay (Chile), Rolf Klein (Bonn), Alejandro Lopez-Ortiz (Waterloo) Rolf Niedermeier (Jena, Germany) had a goal of bringing together researchers from Adaptive, Output Sensitive, Online and Parameterized Algorithms. Each community came up with its own set of techniques to take advantage of the fact that for some problems the worst case among all instances of same size is too pessimistic. For example, an *adaptive* sorting algorithm takes advantage of an existing order in the input, with its running time being a function of the disorder in the input. In Computational Geometry, the complexity of many algorithms is expressed as a function of the size of their output, which in many cases is a better indicator of the difficulty of the instance than the size of the input — this approach is pursued in the field of *Output Sensitive Algorithms*. In *Online Analysis*, the performance of the off-line optimum plays the role of a per-instance measure of the input difficulty, and the online algorithm is expected to match it, usually up to a constant factor.

During the exchange sessions of the seminar, it turned out that for most researchers of these other areas, the main questions to the PC-community seemed to be the following: What *is* a parameter? How do we get one/is it computable?

Dániel Marx pointed out the vast variety of possible parameterizations, distinguishing different kinds of parameters. One big sort are *obvious, explicitly given parameters*, such as for example the required size of the solution (e.g., number of vertices in graph problems), the size of "something" (e.g., number of variables, number of clauses), the dimension, or the distance from the trivial solution (E.g., for a planar graph G, is $\alpha(G) \geq \frac{n}{4} + k$, where $\alpha(G)$ denotes the size of a maximum independent set for G? This is an open problem. For $k = 0$ it is trivial due to the Four Color Theorem.). There are further sorts of parameters, which are more of the *structural* kind, meaning that they make use of certain properties of the input and are not always easy to compute, as for example the treewidth of a graph.

As an interesting outcome of the seminar, open problems and possible lines of research were stated. Dniel Marx proposed the following:

k-Clique for line segments
Given a set of line segments in the plane, and given a parameter k, are there k line segments that pairwise intersect?

So far, it is open whether this problem is in P or NP-complete, and whether it is FPT (with k being the parameter). The combinatorial question related to this problem is the following: Given a set of pairwise intersecting segments, build a graph in the following way:

- its vertices are the segments
- add an edge between two vertices if the corresponding segments touch.

The above line segment problem then asks for a k-clique in this graph. Moreover, what can we say about these graphs, e.g. with respect to their maximum treewidth? The k-Clique for line segments problem is open, whereas for more general lines (curves), the corresponding k-Clique problem is known to be NP-complete.

The final discussion brought up some starting points for finding connections between the different areas of research, but this is far from being settled. Inspired by the adaptive approach, Mike Fellows encouraged to have a look at the power of our reduction rules, and at the assessment of the adaptivity of an algorithm with respect to these rules, i.e., a systematic investigation of the best order in which a set of reduction rules should be applied.

Shai Ben-David made a further suggestion, motivated from Machine Learning: If we consider the stability of the input (i.e., small perturbations of the input would have no big influences on the output) as a function of the output, does this maybe fit into the FPT-framework?

Kurt Mehlhorn pointed out in his talk on geometric computing that a good strategy could mean to do the easy cases fast – this has a flavor of data reductions.

Raimund Seidel was talking about the aspects of Turing kernelization and encouraged the community to have a look at the ILP people in order to encouter Turing kernelizations. The relations between LP-solving and data reduction could be an interesting area.

All in all, the concept of parameterizing seems to be natural in the context of adaptive and output-sensitive algorithms, as parameters that measure some kind of regularity arise naturally here. In the case of online algorithms, the gap seems to be larger.

Chinese University of Hong Kong

Prof. Leizhen Cai held a series of seminars on Parameterized Complexity in April, with talks by Mike Fellows and Fran Rosamond.

Figure 3: Discussions following Mike Fellows' talk at Chinese University of Hong Kong. Prof. Irwin King (l), MF (c), Leizhen Cai (r) and students.

Special Event

Congratulations to Dániel Marx and Zsuzsa Martonffy on the birth of their beautiful daughter Veronik.

CONGRATULATIONS!

Please contact our new graduates or their advisors if you know of post-doc or other opportunities for them.

Michael Dom. Dissertation: *Recognition, Generation, and Application of Binary Matrices with the Consecutive-Ones Property.* Institut für Informatik, Friedrich-Schiller-Universität Jena. Published by Cuvillier, 2009. Advisor: Rolf Niedermeier. Congratulations, Dr. Michael Dom.

Serge Gaspers. Dissertation: *Exponential Time Algorithms.* Department of Informatics at the University of Bergen, December 2008. Advisors: Fedor V. Fomin and Pinar Heggernes. Since January, Dr. Gaspers is a Postdoctoral Researcher in the AlGco group (ALgorithms for Graphs and COmbinatorics) at the LIRMM (Laboratoire d'Informatique, de Robotique et de Microlectronique de Montpellier) at the University of Montpellier 2 in France. Congratulations, Dr. Serge Gaspers.

Niko Schwarz. Master's Thesis: *Rank aggregation by criteria. Minimizing the maximum Kendall-Tau distance.* Institut für Informatik, Friedrich-Schiller-Universität Jena. Betreuer Dipl.-Bioinf. Nadja Betzler, Prof. Dr. Rolf Niedermeier. April 2009.