Open Problems from Dagstuhl Seminar 09511:

Parameterized Complexity and Approximation Algorithms

Erik D. Demaine
M.I.T.
edemaine@mit.edu

MohammadTaghi Hajiaghayi
AT&T Research
hajiagha@mit.edu

Dániel Marx
Tel Aviv University
dmarx@cs.bme.hu

The following is a list of the problems presented on Monday, December 14, 2009 at the open-problem session of the Seminar on Parameterized Complexity and Approximation Algorithms, held at Schloss Dagstuhl in Wadern, Germany.

Reduction without Sparsification

Mihai Pătraşcu
AT&T Research
mip@alum.mit.edu

The Exponential Time Hypothesis (ETH) states that 3SAT cannot be solved in $2^{o(n)}$ time. We have strong relations between ETH and fixed-parameter tractability. For instance, assuming ETH, a k-clique cannot be found in $n^{o(k)}$ time, and k-SUM cannot be solved in $n^{o(k)}$ time. (The k-SUM problem is this: given n numbers, do some k of them sum to zero?)

We can also make a stronger assumption: CNF SAT cannot be solved in $2^{\alpha n}$ time, for constant $\alpha < 1$. This assumption gives us a handle on the constant in the exponent, so that we can hope for tight powers for other problems. Can we prove that k-clique requires $\Omega(n^{2k/3})$ time under this assumption? (This would be a tight bound if fast matrix multiplication takes quadratic time.) Can we prove that k-SUM requires $\Omega(n^{\lfloor k/2 \rfloor})$ time under the assumption? (Again, this would be tight.)

The main issue is to avoid the sparsification lemma, which is crucially needed in the current reductions.

A different starting assumption is the following: if the complexity of k-SAT is $2^{s_k n}$, then $s_k \to 1$ as $k \to \infty$. This assumption is not known to be equivalent to the CNF SAT assumption, but an equivalence is conceivable (and another open problem).

Vertex Cover versus Maximum Degree

Gregory Gutin
Royal Holloway U. London
gutin@cs.rhul.ac.uk

Given a graph G with m edges and maximum degree $\Delta(G)$, what is the parameterized complexity of deciding whether G has a vertex cover of size at most $m/\Delta(G) + k$, where k is the parameter? If $\Delta(G) \leq B = O(1)$, then the problem is fixed-parameter tractable, even parameterized by both k and B, as there is an algorithm of running time $2^{Bk} n^{O(1)}$. See [GKLM09], which also poses this problem.

Dagstuhl Seminar Proceedings 09511
Parameterized complexity and approximation algorithms
http://drops.dagstuhl.de/opus/volltexte/2010/2499

Henning Fernau asks the following follow-up: what is the status of approximating the corresponding minimization problem, i.e., minimize k such that there is a vertex cover of size $m/\Delta(G) + k$ in the given graph G?

References

[GKLM09] Gregory Gutin, Eun Jung Kim, Michael Lampis, and Valia Mitsou. Vertex cover problem parameterized above and below tight bounds. arXiv:0907.4488, August 2009. http://arXiv.org/abs/0907.4488

Planar Directed *k*-path

Fedor Fomin
U. Bergen
fomin@ii.uib.no

Does the planar directed k-path problem have a subexponential fixed-parameter algorithm? More precisely, given an n-vertex planar directed graph G and an integer k, is it possible to find in time $2^{o(k)}n^{O(1)}$ a directed path of length at least k?

For undirected planar graphs, the problem is solvable in subexponential time $2^{O(\sqrt{k})}n^{O(1)}$ [DPBF05]. For directed planar graphs, for every $\varepsilon > 0$, the problem is solvable in time $(1+\varepsilon)^k n^{O(f(\varepsilon)}$ [DFLRS10].

References

[DFLRS10] Frederic Dorn, Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Beyond bidimensionality: parameterized subexponential algorithms on directed graphs. To appear in *STACS 2010*.

[DPBF05] Frederic Dorn, Eelko Penninkx, Hans L. Bodlaender, and Fedor V. Fomin. Efficient exact algorithms on planar graphs: exploiting sphere cut branch decompositions. In *ESA 2005*, pages 95–106.

Effectivization Generalizes Approximation

Michael Fellows
U. Newcastle
Michael.Fellows@newcastle.edu.au

Suppose for two parameters π and π' of graphs, a true mathematical implication holds:

$$\pi(G) \leq k \implies \pi'(G) \leq k'$$

where k' is some function of k. It is in general interesting to look for *algorithmic effectivizations* of such mathematical knowledge, with respect to complexity regimes of interest.

For example, if the length of a longest cycle in G is at most k, then the treewidth of G is at most $k' = k$. Because both LONG CYCLE and TREEWIDTH are NP-hard, it is interesting to investigate how well this implication can be made algorithmic in polynomial time. In fact, it can be completely effectivized in polynomial time: on input (G, k), we can either determine that G has cycle of length greater than k, or determine that the treewidth of G is at most k.

(Even better, we can produce either a long cycle or a bounded-width tree decomposition.) This is a classic example of a *win/win algorithm* and such algorithms have many uses in fixed-parameter algorithm design as subroutines. For example, for the LONG CYCLE problem, we can in polynomial time either know the answer, or in polynomial time "kernelize" to bounded treewidth (a strategy that has been termed *relative kernelization*).

For another example, it is a true implication that, if the bandwidth of G is at most k, then the pathwidth of G is at most k. In this case, we know only an *approximate polynomial-time effectivization*: in linear time we can either determine that the bandwidth of G is greater than k, or determine that the pathwidth of G is at most 2^{k+1} [FHRS09].

Can we do better? Is there a polynomial-time effectivization where the approximation function is polynomial in k? Is there some way to prove lower bounds for approximate effectivization questions of this sort?

These issues can be interesting even for trivial implications. For example, it is certainly true that, if the minimum domination number of G is at most k, then the minimum domination number of G is at most k (!). Because DOMINATING SET is $W[2]$-complete, it is interesting to consider FPT effectivizations of this trivial implication, and it is of course a major open problem whether there is an FPT approximate effectivization of this trivial implication: the issue of *FPT approximation* for DOMINATING SET.

Thus, in some sense, FPT approximate effectivization of mathematical implications between W-hard (when "standing alone") structural parameters, is an issue that generalizes the issue of FPT approximation of W-hard parameterized problems—about which little is so far known. Studying the complexity of such effectivization problems may shed some light on this area.

References

[FHRS09] Michael Fellows, Juraj Hromkovič, Frances Rosamond, and Monika Steinov. Fixed-parameter tractability, relative kernelization and the effectivization of structural connections. In *Proceedings of Computability in Europe 2009*, Heidelberg, Germany, July 2009.

Subexponential Steiner Tree

Siamak Tazari
Humboldt U. Berlin
siamak.tazari@googlemail.com

Does k-Steiner tree admit a subexponential fixed-parameter algorithm, say for planar graphs? Recall the k-Steiner tree problem: given an unweighted graph G and a set $S \subseteq V(G)$ of terminals, is there a tree in G consisting of k nodes (including the terminals) that contains all terminals in S? We have obtained an algorithm with running time $\inf_{0<\varepsilon\leq 1} O((1+\varepsilon)^k + n^{O(1/\varepsilon)})$ on H-minor-free graphs [MT09]. But is there a subexponential algorithm, with running time $2^{o(k)}n^{O(1)}$, on H-minor-free graphs or even planar graphs? We have shown that this is possible if and only if the problem admits a subexponential kernel [MT09].

References

[MT09] Matthias Mnich and Siamak Tazari. (Almost) subexponential FPT-algorithms on H-minor free graphs and a polynomial kernel for connected dominating set. Presentation at this seminar by Siamak Tazari.

Compatible Coloring
Dániel Marx
Tel Aviv University
dmarx@cs.bme.hu

[Reposing of a classic problem.]

In the *compatible coloring* problem [HN08], we are given a complete graph with each edge assigned one of three colors, and we want to assign one of three colors to each vertex such that no edge has the same color as both of its endpoints. (More formally, if $c : V \cup E \to \{1, 2, 3\}$ denotes the union of the input edge coloring and the output vertex coloring, then $c(uv) = x$ implies either $c(u) \neq x$ or $c(v) \neq x$.) Does this problem have a polynomial-time algorithm?

This is a well-known problem in constraint satisfaction, but we believe that fixed-parameter tools (either upper or lower bounds) may be the missing link in finding a solution. The best known algorithm runs in $n^{O(\log n/ \log\log n)}$ time [FHKS05], and its approach suggests that there may be an underlying parameterized problem of relevance. In particular, it seems somewhat related to 2SAT Deletion (removing the fewest clauses to make a 2SAT formula satisfiable), which is fixed-parameter tractable [RS08]. Can we use fixed-parameter tools to solve compatible coloring in polynomial time, or to prove an $n^{\Omega(\log n/ \log\log n)}$ lower bound assuming ETH?

References

[FHKS05] Tomás Feder, Pavol Hell, Daniel Král, and Jiří Sgall. Two algorithms for general list matrix partitions. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, Vancouver, Canada, 2005, pages 870–876.

[HN08] Pavol Hell and Jaroslav Nesetril. Colouring, constraint satisfaction, and complexity. *Computer Science Review* 2(3):143–163, 2008.

[RS08] Igor Razgon and Barry O'Sullivan. Almost 2-SAT is fixed-parameter tractable (extended abstract). In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, Reykjavik, Iceland, 2008, pages 551–562.

Approximating Star Discrepancy
Magnus Wahlström
MPI für Informatik
wahl@mpi-inf.mpg.de

Given a set P of n points in the unit d-dimensional hypercube $[0, 1]^d$, we want to find a box with one corner at the origin having either high volume and few points or low volume and many points. More precisely, the *star discrepancy* of P is $d^*(P) = \max_B |Vol(B) - |B \cap P|/|B|$, where the max is taken over boxes $B = [0, x_1] \times [0, x_2] \times \cdots \times [0, x_d]$, $0 \leq x_i \leq 1$.

The best known methods for computing the star discrepancy take $n^{O(d)}$ time, and if P is a good point set (i.e., its star discrepancy is near-optimal), then even obtaining a constant-factor approximation takes $n^{O(d)}$ time with known methods. It is known that the problem is NP-hard, and we have proved that it is W[1]-hard when parameterized by d [work under submission]. But little is known about approximability. Is it possible to obtain a constant-factor approximation (or better), either classically or with a fixed-parameter algorithm?

References

[DEM96] D. P. Dobkin, D. Eppstein, and D. P. Mitchell. Computing the discrepancy with applications to supersampling patterns. *ACM Trans. Graph.* 15(4):354–376, 1996.

[Gne08] Michael Gnewuch. Bracketing numbers for axis-parallel boxes and applications to geometric discrepancy. *J. Complexity* 24(2):154–172, 2008.

[GSW09] M. Gnewuch, A. Srivastav, and C. Winzen. Finding optimal volume subintervals with k points and calculating the star discrepancy are NP-hard problems. *J. Complexity* 25:115–127, 2009.

[Thi01] Eric Thiémard. An algorithm to compute bounds for the star discrepancy. *J. Complexity* 17(4):850–880, 2001.

Speeding Up Fixed-Parameter Approximation

Henning Fernau
U. Trier
fernau@uni-trier.de

For e.g. Vertex Cover, we know that there is no $2^{o(k)}$ exact algorithm assuming the Exponential Time Hypothesis. With approximation factor α (smaller than what we could do in polynomial time), can you show that there is still no $2^{o(k)}$ α-approximation algorithm? For Vertex Cover, I don't know, and this question applies to many other problems too.

Also, a methodology question: For fixed-parameter approximation, I know how to use the classic FPT techniques of reduction rules and search trees to speed up over exact fixed-parameter algorithms. But I don't know how to get speedup using the classic technique of iterative compression. Can this help? In the approximation world, bootstrapping is similar to iterative compression; perhaps that is useful?

Counting Lattice Points

Michael Fellows
U. Newcastle
Michael.Fellows@newcastle.edu.au

[Open problem by Nadia Betzler.]

Consider integer linear programming with k variables and n constraints but no objective function, parameterized by k. It is famously fixed-parameter tractable to determine whether the linear program has any solution. If the number of solutions is finite (the polytope is bounded), is it fixed-parameter tractable to count (or approximately count) the number of solutions? This is a parameterized version of the classic problem of counting lattice points in a bounded polytope.

Some recent papers by Alexander Barvinov may be relevant.

Making Graphs Distance-Balanced

Sergio Cabello
U. Ljubljana
sergio.cabello@fmf.uni-lj.si

What is the parameterized complexity of finding the minimum number of edges to add to a graph to make it "distance-balanced"? Informally, a graph is distance-balanced if every vertex has the same average distance to all other vertices. Formally, for each vertex v of an unweighted undirected graph G, let $\sigma(v) = \sum_{u \neq v} d_G(u, v)$; then the graph G is *distance-balanced* if $\sigma(u) = \sigma(v)$ for all vertices u, v.

Together with Primož Lukšič, we can prove that this problem is NP-hard. Is it W[1]-hard when parameterized by the number of edges you add? Similar problems could also be considered for edge removal.

Subset Feedback Vertex Set

Saket Saurabh
U. Bergen
saket.saurabh@ii.uib.no

What is the parameterized complexity of the following problem?

Subset Feedback Vertex Set
Input: Graph $G = (V, E)$, $S \subseteq V$, and a positive integer k.
Parameter: k
Question: Does there exist $F \subseteq V$, $|F| \leq k$, such that every cycle having at least one vertex in S is also hit by F?

Is this problem fixed-parameter tractable or W-hard? We know that the problem is fixed-parameter tractable on planar graphs.

Also, MohammadTaghi Hajiaghayi asks the following: are there subexponential fixed-parameter algorithms for Subset Feedback Vertex Set on planar graphs?

Approximating Disjoint Paths

Dimitrios Thilikos
National and Kapodistrian U. Athens
sedthilk@math.uoa.gr

Recall the Disjoint Path Problem: given a graph with k sources $s_1, s_2, \ldots, s_k$ and k targets $t_1, t_2, \ldots, t_k$, find k disjoint paths $p_1, p_2, \ldots, p_k$ where p_i is from s_i to t_i. This problem is well-known to be fixed-parameter tractable, with an algorithm running in time $O(f(k)n^3)$. The algorithm is relatively simple, but the proof of its correctness is based on the entire Graph Minors Project which makes the function f immense.

Can we improve f? Kawarabayashi et al. claim they can improve the proof of correctness so that f becomes something like i-ly exponential on k where i is relatively small. (A bulk estimation would say $i \leq 9$.) Can we obtain an $O(2^k n)$ algorithm, ignoring Graph Minors? This would be impressive, but perhaps hard to believe.

But what about fixed-parameter approximation? That is, can we devise a fixed-parameter algorithm that runs in $2^{O(k)} \cdot n$ time and outputs either that "no k paths exists" or returns $\geq k/2$ of the paths in question. We can show that this problem has no polynomial-size kernel, based on the results of [BDFH09].

References

[BDFH09] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

Subexponentiality Gap

Xiuzhen Huang
Arkansas State U.
huang.xiuzhen@cs.astate.edu

For Planar Vertex Cover, Planar Independent Set, Planar Dominating Set, etc., Baker [Bak94] obtained efficient PTASs (EPTASs) with running time $2^{O(1/\varepsilon)}n^{O(1)}$. We [CHKX04] proved a lower bound of $2^{o(\sqrt{1/\varepsilon})}n^{O(1)}$ assuming the Exponential Time Hypothesis. Marx [Mar07] proved a lower bound of $2^{(1/\varepsilon)^{1-\delta}}n^{O(1)}$ assuming the Exponential Time Hypothesis.

Can we close the remaining small gap between the upper bound and the lower bound of EPTASs for these problems on planar graphs?

References

[Bak94] Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the Association for Computing Machinery* 41(1):153–180, 1994.

[CHKX04] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Linear FPT reductions and computational lower bounds. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 2004, pages 212–221.

[Mar07] Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60-78, 2008.

Fixed-Parameter Approximation Reduction

Xiuzhen Huang
Arkansas State U.
huang.xiuzhen@cs.astate.edu

Our framework for fixed-parameter approximation (FPA) [CH06] uses L-reduction to prove the positive results for MAX SNP problems, namely, that all problems in the class MAX SNP admit fixed-parameter approximation schemes in time $2^{O((1-\varepsilon/O(1))k)}n^{O(1)}$ for any $\varepsilon > 0$.

It would be very nice to formalize a standard/uniform notion of *FPA reduction* to study the negative and positive results under the framework of fixed-parameter approximation.

References

[CH06] Liming Cai and Xiuzhen Huang. Fixed-parameter approximation: conceptual framework and approximability results. In *Proceedings of the 2nd International Workshop on Parameterized and Exact Computation*, Zürich, Switzerland, September 2006, pages 96–108.

Weighted FPT

MohammadTaghi Hajiaghayi
AT&T Research
hajiagha@mit.edu

Most fixed-parameter algorithms for parameterized problems, like k-vertex cover or k-dominating set or k-Steiner tree, are inherently about unweighted graphs. Of course, we could add integer weights to the problem (which, with suitable scaling, can approximate any real weights), but this can lead to a huge increase in the parameter. By contrast, many algorithms such as those based on linear programs (and approximation algorithms based on rounding such linear programs) very naturally extend to arbitrary real weights with little to no increase in running time.

Can we devise fixed-parameter algorithms for weighted graphs that have less severe dependence on weights? For example, can we obtain polynomial instead of exponential dependence on the maximum weight divided by the minimum weight (assuming integer weights, say)? Or what are the "right" parameters for such problems? Is there a nice framework for designing fixed-parameter algorithms on weighted graphs? It may also be natural to allow a little approximation in the goal, so that we can round the weights to values more suitable for fixed-parameter algorithms.

Metric Dimension

Daniel Lokshtanov
U. Bergen
daniello@ii.uib.no

What is the parameterized complexity of *metric dimension*: given a graph G and a parameter k, is there a set S, $|S| \leq k$, such that any two vertices $u, v \notin S$ have a vertex $s \in S$ with $d(u, s) \neq d(v, s)$? In other words, we want a set S whose distances to a vertex $v \notin S$ uniquely define v, providing a kind of coordinate system. For example, a path graph has metric dimension 1 (pick S to be the leftmost vertex); the $n \times n$ grid graph has metric dimension 2 (pick S to be the two left corners); and more generally a d-dimensional grid has metric dimension d.

This problem is known to be NP-hard [GJ79]. Is it fixed-parameter tractable? We conjecture that it is W[1]-hard.

Dimitrios Thilikos points out that there is a subgraph characterization of having metric dimension $\leq k$ [HMPSW07].

References

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

[HMPSW07] Carmen Hernando, Merce Mora, Ignacio M. Pelayo, Carlos Seara, and David R. Wood. Extremal graph theory for metric dimension and diameter. arXiv:0705.0938, http://arXiv.org/abs/0705.0938.

Better Parameters for Cluster Editing
Frank Dehne
Carleton U.
dehne@scs.carleton.ca

[Joint open problem with Mike Langston.]

Recall the cluster editing problem: given a graph, how many edge edits (insertions or deletions) do you need to make it into a disjoint union of cliques? This problem is well-studied in terms of fixed-parameter tractability: it has a linear kernel and a 2.27^k tree-search algorithm, where k is the number of edits. But we argue that this parameter is not so useful in at least one practical setting, because it is essentially quadratic in clique sizes. Can we define a new parameter that counts vertices instead of edges, and obtain similarly efficient fixed-parameter algorithms for this parameter?

The practical setting we consider is cleaning proteome data. A proteome is the graph of all protein–protein interactions in a system of proteins. It is big, and difficult to compute: determining the presence or absence of one edge can take weeks and a publication. Yeast has thousands of proteins, while humans have tens of thousands of proteins, so this is a big effort.

Unfortunately the available edge-presence data is noisy, with many false positives and false negatives. How can we clean up this data? In our experiments, clustering editing is a good predictor for protein interaction, so this seems a useful formulation.

The bad news is that the existing cluster-editing fixed-parameter algorithm is too slow in practice, much much slower than say Vertex Cover. We can solve instances up to aroud 60 vertices, which is much too small for the application. The trouble seems to be in the choice of parameter k, which is essentially like square of clique size, making the linear-in-k kernel into effectively a quadratic kernel.

One possible alternate parameter k' is the number of vertex edits required to transform the graph into a disjoint union of cliques. Note that we still want to solve the original problem—minimizing the number of edge edits—but hope to solve it in time $f(k')n^{O(1)}$ for such an alternate parameter k' that involves vertices instead of edges.

Another practical issue is that programs tend to be more efficient for parameters for which the problem is monotone, that is, increasing the parameter preserves YES instances. This property is not true for the number k of edge edits: requiring more edits may prevent a previous cliquification. This fact makes it difficult to search for the best value of k, because in practice YES instances (where the search-tree algorithm can terminate as soon as it finds an answer) run much faster than NO instance (which must explore the entire search tree). If we could obtain the aforementioned property for a parameter k', we could use linear search from large k' to small k', and thus solve only one NO instance. (In theory, of course, we would use binary search, but in practice avoiding NO instances is more important, making such a linear search the method of choice.)

Periodic Scheduling
Henning Fernau
U. Trier
fernau@uni-trier.de

I want to point to an area that has been neglected so far from the viewpoint of parameterized complexity. It offers many problem variants, three of which are discussed below. They always involve some kind of numberwork which seem to make them hard to place within the W hierarchy.

The basic setup consists of a set C of n customers, where customer i has positive integer period p_i of service desire, say, counted in whole days. For example, imagine you are supposed to deliver milk to customer i every p_ith day. But you have only one car and you can make at most one delivery per day. Or more realistically, you need to service real-time jobs with various periods (e.g., to update various output devices like display and sound, and to poll various input devices like keyboards or sensors), but you have only one processor.

Version 1: Maximize the number of happy customers. More formally, does there exist a subset $C' \subseteq C$ of customers, $|C'| \geq k$, such that C' can be served, i.e., there are starting days s_i for each customer i such that $s_i + xp_i \neq s_j + yp_j$ for all customers $i, j \in C'$, $i \neq j$?

Looking at proofs in the community, it is effectively known that this problem is W[1]-hard. But is it in W[1]?

Version 2: To see how the numberwork comes into play, notice that if the starting days (s_i's) are given, then this problem can be shown to be W[1]-complete.

Version 3: How many cars (or processors) do you need to make all customers happy? Here the parameter k' is the number of cars.

Looking at proofs in the community, it is effectively known that this problem is W[1]-hard, also when start days (s_i's) are given. But is it in W[1]? This membership is open even if the start days (s_i's) are given.

As far as I know, these questions have been investigated from the viewpoint of approximation algorithms, but none of this work leads to APX-membership results. The reductions for NP-completeness indicate hardness results that might be stated somewhere. (I am not from this community but rather stumbled across these problems.)

Because this kind of problem surely has some practical motivation, it would be interesting to find nice FPT approximations, e.g., to validate the heuristics that are in current use.

There was also some brief discussion about the current biggest open problems of the form "is this problem fixed-parameter tractable?" The two big targets that came to many people's minds were Biclique and Edge Multicut.