Overview of my research

Igor Razgon

November 2, 2015

My main research area is parameterized complexity, in particular, applications of the parameterized complexity methodology to the following topics:

- algorithms for graph separation problems;
- complexity of Boolean functions;
- structural graph parameters (e.g. treewidth and cliquewidth) and their applications to understanding well quasi-orderability of graph classes.

I also have experience in design of exponential time algorithms both in applied (AI planning and constraint solvers) and theoretical (moderate exponential algorithms) settings.

In the following three sections I give a short overview of my results related to the above three topics and outline some interesting (in my opinion) directions of future research

1 Parameterized complexity of Boolean functions

Complexity of a Boolean function is the amount of memory needed to represent the function. For example, a function of n variables requires $O(2^n)$ bits to be represented in the form of a truth table. Other forms of representation, such as Boolean circuits can be much more succinct for certain classes of functions.

Understanding the complexity of a particular representation for a particular class of functions is very important from both theoretical and applied perspectives. I came to this field through my interest in knowledge representation. In this field knowledge is often represented in a propositional form. This representation is nothing else than a Boolean function. There are two quite contradictory requirements to this Boolean function. On the one hand, the representation should be sufficiently concise to be recorded in a computer memory. On the other hand, this representation should allow efficient querying. There is no universally good solution. However, a few reasonable representations reconciliating the above requirements have been proposed. It is important to study upper and lower bounds of these representations to understand the classes of functions they can represent and the classes they cannot.

1.1 Lower bounds

Lower bounds for OBDDs The starting point of my research in this topic was the result [8] stating a parameterized upper bound on the size of the famous Ordered Binary Decision Diagrams (OBDDs), a model widely used in the areas of verification and knowledge representation because of possibility of its efficient querying. The disadvantage of OBDDs is that for many useful types of functions their size is exponential in the number of variables. A standard way to obtain an OBDD is by transformation from a Conjunctive Normal Form (CNF). The result [8] states that a CNF of treewidth k (of its primal graph) [1] can be transformed into an OBDD of size $O(n^k)$.

A natural open question (for a parameterized complexity researcher) is whether or not the above upper bound can be turned into a *fixed-parameter tractable* one, i.e. to the form $f(k) * n^c$ where c is a fixed constant independent on k. In [14], we show that this is *impossible* by demonstrating for each k, there is a class of CNFs of treewidth at most k for which the size of equivalent OBDDs is $\Omega(n^{k/4})$. Thus we provide a parameterized lower bound, essentially matching the upper bound of [8].

This lower bound has one more consequence. A relatively recent development in the area of knowledge representation is introduction of Sentential Decision Diagrams (SDD) [7]. This model is more succinct than OBDDs and yet many queries that can be efficiently answered for OBDDs can also be efficiently answered for SDDs. Thus the SDD has a potential to replace the OBDD in applications. Using the *parameterized* lower bound of [14], I provided the first *non-parameterized* separation between OBDD and SDD [12]. That is, I demonstrated a class of CNFs which have polynomial size representation as SDDs but do not have a polynomial size representation as OBDDs.

Lower bounds for non-deterministic read-once branching programs. In terms of circuit complexity classification, OBDDs are quite a restrictive model called *oblivious deterministic read-once branching programs.* A natural follow-up question for the result [14] is to clarify if the non-parameterized lower bound still holds if some of these restrictions are waived. This turned out the case for the first two restrictions. In particular, I have shown in [13] that the $n^{\Omega(k)}$ lower bound holds for non-deterministic read-once branching programs (NROBP). This parameterized lower bound has been used to obtain a *non-parameterized* one, essentially matching the upper bound established in [3]. The extended version of [13] (together with the non-parameterized lower bound) has been recently accepted to the Algorithmica journal, please see [15] for the corresponding preprint.

Further development. The result [13] shows that read-once branching programs are incapable to efficiently represent CNFs of bounded treewidth. A natural direction for further investigation is to try to see if the same is true for read-twice branching programs, read-3-times and, more general, for read-c-times

[1] Treewidth is a well known structural graph parameter measuring closeness of the graph to a tree. See e.g. here `https://en.wikipedia.org/wiki/Treewidth` for a short description of this parameter.

branching programs for an arbitrary but fixed constant c.

My current working hypothesis is that this is indeed true, i.e. that read-c-times branching programs cannot efficiently represent CNFs of bounded treewidth. In fact I have already made a progress towards verifying this hypothesis. The corresponding preprints are [16] and [17].

Let me say a few more words about [17]. If the hypothesis above is correct, it would imply a polynomial *separation* between read-c-times non-deterministic branching programs and CNFs. In other words, it would follow that there are Boolean functions expressible as CNF of polynomial size (in the number of variables) that cannot be represented as polynomial size read c-times non-deterministic branching programs. However, this latter question is still open as well. Therefore, it looks meaningful to try first to separate CNF and read c-times non-deterministic branching programs with the hope that the obtained insight would help to understand the complexity of these branching programs on CNFs of bounded treewidth. The work [17] is a progress in this direction. It does not yet capture non-deterministic read-c-times branching programs in their full generality. However, it achieves the desired separation for two important subclasses of these branching programs. It also introduces a new graph structural parameter that may be important for construction of *hard* CNFs of bounded treewidth.

A rather unexpected spin-off of results [16] and [17] is that the lower bounds for *oblivious* branching programs hold for *semantic* rather than *syntactic* restriction. A well known open problem in the area of circuit complexity is to obtain a non-polynomial lower bound for *semantic* non-deterministic read-once branching programs. Now, one of lower bounds of [17] requires one more restriction that the branching program is *oblivious*. Thus, it is very interesting to investigate whether the structural parameter machinery developed in [17] and earlier in [13] can be upgraded to tackle this well known open problem.

1.2 Upper bounds

An important formalism in the area of propositional knowledge representation is a Decomposable Negation Normal Form (DNNF) [6]. A notable result regarding DNNFs is that a CNF of treewidth at most k can be transformed to a DNNF of size $O(2^k n)$. In terms of parameterized complexity, this means that there is a fixed-parameter tractable transformation from CNF to a DNNF parameterized by treewidth of the CNF.

The above fact naturally leads to the following question. In the area of parameterized algorithms, it is customary, once a problem turns *fixed-parameter tractable* (FPT) parameterized by treewidth, to check if this problem remains FPT being parameterized by *cliquewidth*. [2] Thus it is natural to ask if the complexity of transformation from CNF to DNNF remains FPT parameterized by the cliquewidth of the input CNF rather than its treewidth.

[2] Another structural graph parameter `https://en.wikipedia.org/wiki/Clique-width` that, unlike treewidth, is applicable to dense graphs.

In [19], we answered the above question positively. The most interesting part of the proposed transformation is a 'sparcification' algorithm transforming any circuit of cliquewidth of k into a circuit of treewidth at most $18k$ computing the same function and having size at most four times the original circuit. This transformation essentially makes algorithms for circuits of bounded treewidth applicable to circuits of bounded cliquewidth.

Conceptually this transformation means the following. There is a data storage whose underlying structure is dense. Instead of querying this data storage directly, it is *preprocessed* to a *more tree like* storage equivalent to the former one subject to the specified type of queries. The above result shows that bounded cliquewidth of the initial storage is a sufficient condition for such a transformation. It seems a very interesting research direction to consider possibility of such transformation in a broader perspective, applied to other data storages such as databases, Bayesian networks or ontologies, and to obtain an extensive characterisation of the cases when such a transformation is possible. This research direction can have a strong applied effect because tree-like structures are easier to analyse and their querying is easier to parallelise (the last factor is especially important when dealing with big data).

2 Well quasi-ordered classes of graphs

My research in this area has been focused so far on understanding structural properties of hereditary classes of graphs that are well quasi-ordered by the induced subgraph relation. Let us refer to these classes as WQO.

In [5] it was asked whether every WQO class has bounded cliquewidth (that is, whether for each WQO class $\mathbf{G}$ there is a constant c such that the cliqeuwidth of all the graphs of $\mathbf{G}$ is at most c). The beauty of this question is that it connects two seemingly unrelated notions, namely WQO classes and classes graphs of bounded cliqeuwidth. Very recently, we have answered this question negatively by demonstrating a particular WQO having unbounded cliquewidth [9].

Our earlier result shows that the bounded cliquewidth property *does* hold if some additional restrictions are posed on the graph classes [2]. An interesting algorithmic spin-off of this research is a fixed-parameter linear time algorithm for a restricted version of the famous biclique problem where long induced paths are forbidden [1].

The result [9] is disappointing in the sense that it indicates that WQO classes and classes of bounded cliquewidth might *indeed* (and not *seemingly* as suggested above) be unrelated. However, there are still a few questions to settle before such conclusion can be made (and my hope is that *some* relation will be found). These questions are the following.

- We can consider classes of graphs whose vertices are labelled with a constant number of labels and the induced subgraph relation preserves the labels. It is conjectured in [5] that even two labels are enough to ensure that well quasi orderability under this relation implies bounded cliquewidth. If we take the positive outcome as the working hypothesis then a reasonable first step seems to consider an arbitrary (but constant) number of labels and to try to establish bounded cliquewidth under this stronger restriction.

- In the class demonstrated in [9], the cliquewidth (though unbounded) is logarithmic in the number of vertices. It is interesting to see if, in general, cliquewidth of graphs of WQO class is at most logarithmic. If this is the case then restricted application of Courcelle's theorem is possible to show that all the problems definable by Monadic Second Order logic formulas with alternating depth 1 are polynomially solvable for WQO classes.

3 Graph separation problems

Roughly speaking, this area concerns with finding a smallest number of vertices or edges (sometimes subject to a particular property) whose removal break a certain subset of paths in the input graph.

Two of my most cited works are [4] and [18] where we solved two challenging open problems in the area of parameterized complexity. In [11] and [10], in addition to solving open problems, we proposed new methodologies. The work [10] is my favourite (though less cited) because of the nice treewidth reduction theorem we managed to prove there. The idea is briefly outlined below.

Let S be a subset of vertices of a graph G. Let us call the *torso* of S the graph G_S obtained from $G[S]$ (the subgraph of G induced by S) by adding edges between pairs of vertices connected in G by a path whose intermediate vertices lie outside S. Intuitively speaking, in G_S, the vertices of S are connected 'like' they are in G.

Let s and t be two vertices and let S be the union of $\{s, t\}$ and all minimal sets of vertices of size at most k that separate s and t. The treewidth reduction theorem states that the treewidth of G_S is bounded by a function of k and, moreover, this set S can be computed in time fixed-parameter w.r.t. k. The power of the theorem is that if we need to find a minimal $s - t$ separator of size at most k subject to a certain property, instead of searching for it in the *original* graph G, we can look for such a separator in G_S. Since G_S is of bounded treewidth, many of such problems can be straightforwardly solved by the famous Courcelle's theorem. Thus the treewidth reduction theorem reduces a problem instance on a graph of unbounded treewidth to an instance of the same problem on a graph of a bounded treewidth! Moreover, this reduction is generic and domain independent and allowed to prove fixed-parameter tractability of many seemingly unrelated problems.

References

[1] Aistis Atminas, Vadim V. Lozin, and Igor Razgon. Linear time algorithm for computing a small biclique in graphs without long induced paths. In

SWAT, pages 142–152, 2012.

[2] Aistis Atminas, Vadim V. Lozin, and Igor Razgon. Well-quasi-ordering, tree-width and subquadratic properties of graphs. *CoRR*, abs/1410.3260, 2014.

[3] Paul Beame, Jerry Li, Sudeepa Roy, and Dan Suciu. Lower bounds for exact model counting and applications in probabilistic databases. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, Bellevue, WA, USA, August 11-15, 2013*, 2013.

[4] Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.

[5] Jean Daligault, Michaël Rao, and Stéphan Thomassé. Well-quasi-order of relabel functions. *Order*, 27(3):301–315, 2010.

[6] Adnan Darwiche. Decomposable negation normal form. *J. ACM*, 48(4):608–647, 2001.

[7] Adnan Darwiche. SDD: A new canonical representation of propositional knowledge bases. In *22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 819–826, 2011.

[8] Andrea Ferrara, Guoqiang Pan, and Moshe Y. Vardi. Treewidth in verification: Local vs. global. In *Logic for Programming, Artificial Intelligence, and Reasoning, 12th International Conference (LPAR)*, pages 489–503, 2005.

[9] Vadim V. Lozin, Igor Razgon, and Viktor Zamaraev. Well-quasi-ordering does not imply bounded clique-width. In *WG2015, to appear.*

[10] Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transactions in Algorithms*, 9(4):30, 2013.

[11] Dániel Marx and Igor Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. *SIAM J. Comput.*, 43(2):355–388, 2014.

[12] Igor Razgon. On OBDDs for CNFs of bounded treewidth. *CoRR*, abs/1308.3829, 2013.

[13] Igor Razgon. No small nondeterministic read-once branching programs for cnfs of bounded treewidth. In *IPEC*, pages 319–331, 2014.

[14] Igor Razgon. On OBDDs for CNFs of bounded treewidth. In *Principles of Knowledge Representation and Reasoning(KR)*, 2014.

[15] Igor Razgon. On the read-once property of branching programs and cnfs of bounded treewidth. *CoRR*, abs/1411.0264, 2014.

[16] Igor Razgon. On oblivious branching programs with bounded repetition that cannot efficiently compute CNFs of bounded treewidth. *CoRR*, abs/1510.02951, 2015.

[17] Igor Razgon. Two types of branching programs with bounded repetition that cannot efficiently compute monotone 3-cnfs. *CoRR*, abs/1510.05486, 2015.

[18] Igor Razgon and Barry O'Sullivan. Almost 2-sat is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.

[19] Igor Razgon and Justyna Petke. Cliquewidth and knowledge compilation. In *Theory and Applications of Satisfiability Testing (SAT)*, pages 335–350, 2013.