# CALDAM 2018 Pre-Conference School on Algorithms and Combinatorics

**Indian Institute of Technology**
**Guwahati 781039, India**
**February 12-13, 2018**

## Organizers

- Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati 781039, India.
- Department of Computer Science, School of Mathematical Sciences, RKM Vivekananda University, Belur Math, Howrah 711202, India.
- School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada.

## Coordinators

- Sucheta Dutt, Department of Applied Sciences, PEC University of Technology, Chandigarh 160012, India.
- Subir Kumar Ghosh (Co-Convener), Department of Computer Science, School of Mathematical Sciences, RKM Vivekananda University, Howrah 711202, India.
- R. Inkulu (Co-Convener), Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati 781039, India.
- Ramesh Krishnamurti (Co-Convener), School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada.
- R.S. Lekshmi, Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore 641004, India.
- Pinaki Mitra, Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati 781039, India.

# Preface

Discrete Mathematics studies mathematical structures that are discrete, rather than continuous, such as integers, graphs, logic, algorithms and their operations. Discrete Mathematics concerns itself with problems of the following kinds: (1) finding an optimal/extremal object from a large or infinite family of discrete objects, and, (2) Combinatorics, or the mathematics of counting the number of objects satisfying a set of properties among a large family of discrete objects. Most computationally hard problems are precisely the problems of determining the optimal object from a large family of discrete objects or counting the size of such a family. All non-trivial solutions to these problems emerge from the theory of Discrete Mathematics. Often constructive proofs of theorems in Discrete Mathematics lead to algorithms in this domain. Today the applications of Discrete Mathematics form the foundations of Graph Theory, Cryptography, Operations Research, Logic, Computational Geometry, Combinatorics, Algorithms, Theoretical Computer Science, Information Theory, and many others.

The field of Discrete Mathematics in all its branches is a rich and continuously evolving area of research. The school proposes to bring together prominent and leading researchers in Algorithms and Combinatorics to give lectures on recent developments in these overlapping areas of Discrete Mathematics. This will benefit university teachers, researchers and doctoral students in the area of Discrete Mathematics and Computer Science, by exposing them to the recent developments in Computational Geometry, Algorithms, Combinatorics, Graph Theory and their applications.

The school is aimed at fulfilling two purposes: (i) as a *CALDAM 2018 Pre-Conference School*, and (ii) as an *Indo-Canadian School on Algorithms and Combinatorics.* The school is jointly organized by (i) *Indian Institute of Technology, Guwahati, India*, (ii) *RKM Vivekananda University, Belur Math, Howrah, India* and (iii) *Simon Fraser University, Burnaby, Canada.* The school is funded by (i) *Microsoft Research Lab India Pvt. Ltd.* and (ii) *Science and Engineering Research Board, Government of India.*


R. Inkulu — IIT Guwahati

Subir Kumar Ghosh — RKM Vivekananda University

Ramesh Krishnamurti — Simon Fraser University


January 26, 2018

# Lecture Schedule

| Time | February 12, 2018 (Monday) | February 13, 2018 (Tuesday) |
|---|---|---|
| 8.00 hr.– 9.00 hr. | Registration & Inauguration | |
| 9.00 hr.– 10.15 hr. | Bojan Mohar: *Excluded Structure Results: Graph Minors and Graph Immersions* | Valentine Kabanets: *Algorithms Versus Complexity: Two sides of the same coin* |
| 10.15 hr.– 10.45 hr. | Tea | Tea |
| 10.45 hr.– 11.30 hr. | Kamyar Khodamoradi: *Approximation Schemes for Clustering Problems* | Arti Pandey: *On the Complexity of Domination and its Variations* |
| 11.30 hr.– 12.45 hr. | Sunil Chandran: *Spanning Tree Congestion Problem and Generalized Győri-Lovász Th.* | Antonina Kolokolova *Proof Complexity and its Applications* |
| 12.45 hr– 14.00 hr. | Lunch | Lunch |
| 14.00 hr.– 14.45 hr. | Joydeep Mukerjee: *Bend the Strings* | Ambat Vijayakumar:*The Median (anti-Median) Problem on Graphs* |
| 14.45 hr.– 16.00 hr. | Amitabha Bagchi: *Decentralized Random Walk-Based Data Collection in Networks* | Daya Gaur: *An Introduction to Approximation Algorithms* |
| 16.00 hr.– 16.20 hr. | Tea | Closing Session and Tea |
| 16.20 hr.– 18.00 hr. | Short Presentations on Open Problems | |

# Lectures Outline

# Excluded Structure Results: Graph Minors and Graph Immersions

Bojan Mohar
Simon Fraser University, Canada
mohar@sfu.ca

We say that a graph $G$ contains another graph $H$ as a *minor* if a graph isomorphic to $H$ can be obtained from $G$ by first taking a subgraph and then contracting some of the edges. In a series of 22 papers, Robertson and Seymour proved that finite graphs are well-quasi-ordered for the graph minors containment relation [3]. To prove this seminal result they developed a theory of graph minors and one of the centerpieces is a powerful structure theorem (the *Excluded Minor Theorem*) that describes a rough structure of graphs that do not contain a fixed graph $H$ as a minor.

At the same time they resolved a similar result involving graph immersions [4], which was conjectured by Nash-Williams.

The talk will give a gentle introduction to these results, with an emphasis on recent progress in this area [1,2]. Some algorithmic applications may be presented as well.

## References

[1] Ross Churchley, Bojan Mohar, A submodular measure and approximate Gomory-Hu theorem for packing odd trails, SODA 2018.

[2] Matt DeVos, Zdeněk Dvořák, Jacob Fox, Jessica McDonald, Bojan Mohar, and Diego Scheide, A minimum degree condition forcing complete graph immersion, Combinatorica 34 (2014) 279–298.

[3] Neil Robertson, Paul Seymour, Graph minors. XX. Wagner's conjecture, J. Combin. Theory Ser. B 92 (2004), no. 2, 325–357.

[4] Neil Robertson, Paul Seymour, Graph minors XXIII. Nash-Williams' immersion conjecture, J. Combin. Theory Ser. B 100 (2010), no. 2, 181–205.

# Approximation Schemes for Clustering Problems

Kamyar Khodamoradi
University of Alberta, Canada
khodamor@ualberta.ca

## 1. Introduction

Clustering is a prevalent problem in science and technology, business, and our daily lives. Roughly speaking, it is the task of grouping items of data into separate categories or *clusters* based on a notion of similarity or "closeness". It has been studied in a variety of disciplines such as data mining, data science, operations research, and computer science to name a few. In this talk, we will focus on a class of clustering problems known as centre-based clustering. This class contains the well-known problems of Facility Location (or FL for short), $k$-median and $k$-means .

In this presentation, we look at the clustering problems through the lens of approximation algorithms. The main message is that a standard local search can obtain arbitrarily good approximation factors in some cases. The intricate part is, of course, in the analysis, which has become possible thanks to the recent techniques developed for analyzing the local search.

The talk will have two parts. The first part deals with the approximation of $k$-means and some related problems. In the second part, we see the application of local search to setting where we are allowed to ignore a fraction of the input data. The main problems that we consider in this section are known as the k-means problem with outliers ($k$-means-out ) and the k-median problem with outliers ($k$-median-out ).

## 2. PTAS for k-Means

Given a set of points $\mathcal{X}$ and a set of potential centres $\mathcal{C}$, the task in the $k$-means problem is to choose $k$ cluster centres from $\mathcal{C}$ and assign all the points in $\mathcal{X}$ to them in such a way that the sum of squares of the distances between these points and centres is minimized ($k$-median is very similar, the only difference is that the objective there is to minimize the sum of the distances instead). The following is a highlight of the results we will see in this part.

- **PTAS for $k$-means :** a local search algorithm that provides the very first *Polynomial Time Approximation Scheme* (PTAS) for $k$-means in Euclidean metrics with a constant dimension [4][1]. In fact, the analysis is carried out for more general metrics known as "doubling metrics", of which the Euclidean metrics with a constant dimension are a special case.
- **PTAS for $k$-median and FL with arbitrary opening costs:** PTASs were already known in the community for these problems through the classic works of Sanjeev Arora et al. (see [1] for instance). Those results used dynamic programming along the notion of "quad-trees". We, in this part, see how local search can recreate the same results.

[1] Simultaneously and independently in [2] as well.

## 3. PTAS for k-Means with Outliers

In the $k$-MEANS-OUT problem, we are still choosing $k$ cluster centres from $\mathcal{C}$. The difference is that we are not required to assign all the points in $\mathcal{X}$ to the chosen centres. Instead, we are allowed to discard a given number $z$ of them as outliers. Outliers are very common in real-world applications. Consider a situation where the data points are provided by sensory readings. A fraction of the data may be altered by the noise and thus unreliable. For the sake of obtaining a better clustering, it is natural to discard the noisy data.

The main focus of this part is on introducing a framework for transforming an analysis for the clustering problems without outliers to an analysis for the case with outliers, while preserving the approximation guarantee. This transformation comes with a caveat for $k$-MEDIAN-OUT /$k$-MEANS-OUT . We have to chose slightly more than $k$ centres from $\mathcal{C}$ (to be precise, we need to open $(1+\varepsilon)k$ centres). If we are not allowed a violation on the number of centres, there are examples showing that local search can have arbitrarily large cost compared to an optimal clustering. The main results that we observe are:

- **UFL with Uniform Opening Costs:** standard local search achieves the same approximation guarantee for UNCAPACITATED FACILITY LOCATION as the case without outliers.
- **$k$-means-out and $k$-median-out :** standard local search with $(1+\varepsilon)k$ centres gives the same approximation guarantees as the local search with $k$ centres does for the case without outliers. This observation implies a PTAS for doubling metrics, a PTAS for "minor-closed" metrics (e.g., the shortest path metric of a planar graph), and matching constant factors as the best known results for the general metrics.

## 4. Closing Remarks

For the most of this talk, I will be presenting the results of my joint work with Zahary Friggstad, Mohsen Rezapour, and Mohammad Salavatipour that appeared in SODA 2018 [3], and a prior work by the other three authors in FOCS 2016 [4], which forms the basis for our recent results.

## References

[1] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation Schemes for Euclidean $k$-Medians and Related Problems. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, STOC 1998, Dallas, Texas, USA, May 23-26, 1998*, pages 106–113, 1998.

[2] Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 353–364, 2016.

[3] Zachary Friggstad, Kamyar Khodamoradi, Mohsen Rezapour, and Mohammad R. Salavatipour. Approximation Schemes for Clustering with Outliers. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 398–414, 2018.

[4] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a ptas for k-means in doubling metrics. In *IEEE 57th Annual Symposium on*

*Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 365–374, 2016.

# Spanning Tree Congestion Problem and Generalized Győri-Lovász Theorem [2]

L. Sunil Chandran
Indian Institute of Science, Bangalore
sunil@iisc.ac.in

**The Spanning Tree Congestion (STC) Problem:**

Given a connected graph $G = (V, E)$, let $T$ be a spanning tree. For an edge $e = (u, v) \in E$, its detour with respect to $T$ is the unique path from $u$ to $v$ in $T$; let $\mathsf{DT}(e, T)$ denote the set of edges in this detour. The stretch of $e$ with respect to $T$ is $|\mathsf{DT}(e, T)|$, the length of its detour. The dilation of $T$ is $\max_{e \in E} |\mathsf{DT}(e, T)|$. The edge-congestion of an edge $e \in T$ is $\mathsf{ec}(e, T) := |\{f \in E : e \in \mathsf{DT}(f, T)\}|$, i.e., the number of edges in $E$ whose detours contain $e$. The congestion of $T$ is $\mathsf{cong}(T) := \max_{e \in T} \mathsf{ec}(e, T)$. The spanning tree congestion (STC) of the graph $G$ is $\mathsf{STC}(G) := \min_T \mathsf{cong}(T)$, where the minimization is among all spanning trees of $G$.

We note that there is an equivalent cut-based definition for edge-congestion, which we will use in our proofs. For each tree-edge in $e \in T$, its removal from $T$ results in two connected components; let $U_e$ denote one of the components. The edge-congestion of the edge $e$ is $\mathsf{ec}(e, T) := |E(U_e, V \setminus U_e)|$.

Various types of congestion, stretch and dilation problems are studied in computer science and discrete mathematics. In these problems, one typically seeks a spanning tree (or some other structure) with minimum congestion or dilation. We mention some of the well-known problems, where minimization is done over all the spanning trees of the given graph:

1. The Low Stretch Spanning Tree (LSST) problem is to find a spanning tree which minimizes the total stretch of all the edges of $G$. [AKPW95] It is easy to see that minimizing the total stretch is equivalent to minimizing the total congestion of the edges of the selected spanning tree.

2. Spanning Tree Congestion (STC) problem is to find a spanning tree of minimum congestion. [Ost04]

3. Tree Spanner Problem is to find a spanning tree of minimum dilation. [CC95]

There are other congestion and dilation problems which do not seek a spanning tree, but some other structure. The most famous among them is the Bandwidth problem and the Cutwidth problem; see the survey [RSV00] for more details.

Among the problems mentioned above, several strong results were published in connection with the LSST problem. In comparison, there were not many strong and general results for the STC Problem, though it was studied extensively by many researchers in the past 13 years. The problem was formally proposed by Ostrovskii [Ost04] in 2004. Prior to Ostrovskii, Simonson [Sim87] had studied the same parameter under a different name to approximate the cut width of outer-planar graph. A number of graph-theoretic results were presented on this

[2] The presentation is based on the recent work done in collaboration with Yun Kuen Cheung and Davis Issac of Max Planck Institute für Informatik, Saarbrücken, Germany.

topic. Some complexity results were also presented recently, but most of these results concern special classes of graphs.

The most general results regarding STC of general graphs is an $\mathcal{O}(n\sqrt{n})$ upper bound by Löwenstein, Rautenbach and Regen in 2009 [LRR09], and a matching lower bound by Ostrovskii in 2004 [Ost04]. Note that the above upper bound is not interesting when the graph is sparse, since there is also a trivial upper bound of $m$. In our paper we come up with a strong improvement to these bounds after 8 years:

**Theorem:** For a connected graph $G$ with $n$ vertices and $m$ edges, its spanning tree congestion is at most $\mathcal{O}(\sqrt{mn})$. In terms of average degree $d_{avg} = 2m/n$, we can state this upper bound as $\mathcal{O}(n\sqrt{d_{avg}})$. There is a matching lower bound.

Our proof is constructive. Though in general the time complexity of our algorithm can be exponential, for graphs with $m = \omega(n \log^2 n)$ edges, it works in sub-exponential time. Thus for relatively dense graphs we have a sub-exponential algorithm. We raise the existence of a polynomial time algorithm as an open problem.[3]

Though in the general case we do not have any polynomial time algorithm for the $\mathsf{STC}$ problem, we are able to provide efficient constant factor approximation algorithms for two important cases. In both cases we prove that the spanning tree congestion is $\Theta(n)$ and provide efficient polynomial time algorithms to find spanning trees with congestion $\mathcal{O}(n)$.

- For random graphs $\mathcal{G}(n,p)$ with $1 \geq p \geq \frac{c \log n}{n}$ for some small constant $c > 1$. It should be noted that the $\mathsf{STC}$ problem is relevant only for connected graphs and since the threshold function for graph connectivity is $\frac{\log n}{n}$, we are providing the polynomial time algorithm for almost all of the relevant range of values of $p$.
- The other important case where we can give a constant factor approximation algorithm is for the class of graphs with minimum degree $(1/2 + f)n$ for any fixed positive constant $f$.

As a crucial ingredient for the above results, we prove the following lemma:

**Lemma 1** *Let $G$ be a $k$-connected graph with $m$ edges. Then its spanning tree congestion is $\mathcal{O}(m/k)$.*

**The Key Tool: The Generalized Győri-Lovász Theorem.** The result of Löwenstein, Rautenbach and Regen [LRR09] relied on a theorem called the Győri-Lovász Theorem: [Győ76, Lov77]

**Theorem 2** *Let $G = (V,E)$ be a $k$-connected graph. Given any $k$ distinct terminal vertices $t_1, \cdots, t_k$, and $k$ positive integers $n_1, \cdots, n_k$ which sum to $|V|$, there always exists a $k$-partition of $V$ into $\cup_{j=1}^k V_j$, such that for each $j \in [k]$, $t_j \in V_j$, $|V_j| = n_j$ and $G[V_j]$ is connected.*

To derive our stronger upper bound, we formulate and prove a generalization of the above theorem, which we believe might be a useful tool in related areas. The statement of our generalized Győri-Lovász Theorem is given below: (for any $U \subset V$, $w(U) = \sum_{v \in U} w(v)$)

**Theorem 3** *Let $G = (V,E)$ be a $k$-connected graph. Let $w$ be a weight function $w : V \to \mathbb{R}^+$. Given any $k$ distinct terminal vertices $t_1, \cdots, t_k$, and $k$ positive integers positive integers*

[3] Finding the optimal spanning tree is $\mathsf{NP}$-hard [OOUU11, BFG[+]12], but our requirement here is lesser, which is finding a spanning tree with congestion $\mathcal{O}(\sqrt{mn})$.

*$T_1, \cdots, T_k$ such that for each $j \in [k]$, $T_j \geq w(t_j)$, and $\sum_{i=1}^{k} T_i = w(V)$, we can find a $k$-partition of $V$ into $\cup_{j=1}^{k} V_j$, such that for each $j \in [k]$, $t_j \in V_j$, $w(V_j) \leq T_j + \max_{v \in V} w(v) - 1$, and $G[V_j]$ is connected.*

We also give an exponential time algorithm to find the $k$-partition; if we need only $k/2$ partitions instead of $k$ (the input graph remains assumed to be $k$-connected), the algorithm's running time improves to $\mathcal{O}^*(2^{\mathcal{O}((n/k)\log k)})$.[4] We emphasize that the running time does not depend on the weights.

The Győri-Lovász Theorem was proved independently by Győri [Győ76] and Lovász [Lov77]. Győri's proof is elementary, while Lovász used homology theory. Both proofs are non-constructive.

## References


[AKPW95] Noga Alon, Richard M. Karp, David Peleg, and Douglas B. West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.

[BFG+12] Hans L. Bodlaender, Fedor V. Fomin, Petr A. Golovach, Yota Otachi, and Erik Jan van Leeuwen. Parameterized complexity of the spanning tree congestion problem. *Algorithmica*, 64(1):85–111, 2012.

[CC95] Leizhen Cai and Derek G. Corneil. Tree spanners. *SIAM J. Discrete Math.*, 8(3):359–387, 1995.

[Győ76] E. Győri. On division of graphs to connected subgraphs. *Colloq. Math. Soc. Janos Bolyai*, 18:485–494, 1976.

[Lov77] László Lovász. A homology theory for spanning trees of a graph. *Acta Math. Acad. Sci. Hungaricae*, 30(3–4):241–251, 1977.

[LRR09] Christian Löwenstein, Dieter Rautenbach, and Friedrich Regen. On spanning tree congestion. *Discrete Math.*, 309(13):4653–4655, 2009.

[OOUU11] Yoshio Okamoto, Yota Otachi, Ryuhei Uehara, and Takeaki Uno. Hardness results and an exact exponential algorithm for the spanning tree congestion problem. *J. Graph Algorithms Appl.*, 15(6):727–751, 2011.

[Ost04] M. I. Ostrovskii. Minimal congestion trees. *Discrete Math.*, 285:219–226, 2004.

[RSV00] André Raspaud, Ondrej Sýkora, and Imrich Vrto. Congestion and dilation, similarities and differences: A survey. In *SIROCCO 2000*, pages 269–280, 2000.

[Sim87] Shai Simonson. A variation on the min cut linear arrangement problem. *Mathematical Systems Theory*, 20(4):235–252, 1987.


[4] $\mathcal{O}^*$ notation hides all polynomial factors in input size.

# Bend the Strings

Joydeep Mukherjee

Indian Statistical Institute, Kolkata

joydeep.m1981@gmail.com

Graphs arising from the intersection of geometric objects has been a major focus of research due to their various practical applications as well as the elegant combinatorial structure they contain. String graphs are a class of geometric intersection graphs which can be defined as the intersection graphs of simple curves on the plane. As mentioned, they also offer nice combinatorial structures which make them amenable to various beautiful combinatorial as well as algorithmic results. For example most of the important optimization problems are polynomial time solvable in interval graphs which are otherwise NP-Hard in general string graphs. Another famous subclass of string graphs is planar graphs [4]. For planar graphs many important optimization problems admit nice algorithmic upper bounds in terms of approximation and fixed parameter tractability (FPT). But most of these problems are intractable in terms of approximation and FPT in general string graphs. Again more famously Maximum Independent Set problem is intractable in general graphs in terms of approximation whereas there is a relatively good approximation for MIS in string graphs [3].

Recently Asinowski et al. [1] introduced the concept of VPG graphs which is expected to give more insight into the study of string graphs. A *path* is a simple, piecewise linear curve made up of horizontal and vertical line segments in the plane. A *k-bend path* is a path made up of $k+1$ line segments. A *$B_k$-VPG representation* of a graph is a collection of $k$-bend paths such that each path in the collection represents a vertex of the graph and two such paths intersect if and only if the vertices they represent are adjacent in the graph. The graphs that have a $B_k$-VPG representation are called *$B_k$-VPG graphs* and the set of all $B_k$-VPG graphs are denoted simply by $B_k$-VPG. A graph is said to be a *VPG graph* if it is a $B_k$-VPG graph for some $k$. The *bend number* of a graph $G$, denoted by $bend(G)$, is the minimum integer $k$ for which $G$ has a $B_k$-VPG representation. Asinowski et al. [1] showed that the family of VPG graphs are equivalent to the family of string graphs. Therefore, bend number of a string graph is finite. These class of graphs gives a natural way of defining subclasses of string graphs.

The study of different algorithmic problems as well as combinatorial problems in these subclasses are expected to give nice algorithmic insights which might lead to better algorithms for these problems in string graphs. Moreover much work has also been done on graph representation using shapes of lower bends like [2, 4]. These works establish that many known graph classes has bounded bend number which might lead to new algorithmic upper bounds for more general classes of graphs.

In this talk we plan to survey some important results in this area including some of our results and also plan to mention some open problems.

## References

[1] Andrei Asinowski, Elad Cohen, Martin Charles Golumbic, Vincent Limouzy, Marina Lipshteyn, and Michal Stern. Vertex intersection graphs of paths on a grid. *Journal of Graph Algorithms and Applications*, 16(2):129–150, 2012.

[2] Therese C. Biedl and Martin Derka. 1-string $b_2$-vpg representation of planar graphs. *JoCG*, 7(2):191–215, 2016.

[3] Jacob Fox and János Pach. Computing the independence number of intersection graphs. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1161–1165. Society for Industrial and Applied Mathematics, 2011.

[4] Daniel Gonçalves, Lucas Isenmann, and Claire Pennarun. Planar graphs as l-intersection or l-contact graphs. *CoRR*, abs/1707.08833, 2017.

# Decentralized Random Walk-Based Data Collection in Networks

Amitabha Bagchi

Indian Institute of Technology Delhi

bagchi@cse.iitd.ernet.in

We consider a multi-hop sensor network where each node is equipped with a queue which helps in store and forward process of data. There are $k$ source nodes in an $n$-node network, $k \leq n - 1$, and each source node gathers data at a fixed rate from its surroundings and stores it in its queue along with other data packets that may be generated earlier or received from the neighbouring nodes. Specifically, each of the $k$ source nodes generate data as independent Bernoulli processes with at the same rate, $\beta$, and relay it to a designated node, the *sink*. In our model, we assume that the network is connected, but we do not expect any node to know anything about the network except the identity of its neighbours (the nodes with which it can directly communicate). We also assume that time is slotted and nodes communicate with each other at the start of every time slot. However, this assumption can be easily removed and does not affect our results.

In our setting each node, at each time slot, selects a data packet from its queue uniformly at random and forwards it to a neighbour who is chosen according to a distribution on the neighbours (for example, it may be the uniform distribution). We allow a node to transmit only one packet to one of its neighbours, but, it can receive multiple packets from its neighbours. This is known as transmitter gossip constraint [1, 5] and has been used in literature [3, 1, 4, 5] for energy conservation and to prevent data implosion [2]. The movement of any data packet in such setting can be seen as a random walk of the data packet on the graph. The particular type of random walk would depend on the distribution defined on the neighbours of each node, for example, if the distribution is the uniform distribution for each node then the random walk described by a packet becomes the simple random walk on the graph.

We observe that the Markov chain defined by our data collection process achieves steady-state when the queues of the nodes are stable. We also give a necessary and sufficient condition for the stability of the queueing system implicit in the process and the stationarity of the resulting Markov chain.

Having discussed our data collection process at steady-state, we analyze two important performance metrics of our data collection process: throughput and latency. We show that the data rate which determines the network throughput is lower bounded by the spectral gap of the random walk's transition matrix. In particular, we show if the random walk is simple then the rate also depends on the maximum and minimum degree of the graph modelling the network. We also discuss examples for which our lower bound and upper bound on the data rate are optimal up to constant factors. We also present a upper bound on the average latency which reflects the trade-off between the data rate and latency in data collection.

## References

[1] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: Design, analysis and applications. In *Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Comm. Societies*, INFOCOM '05, pages 1653–1664 vol. 3. IEEE, 2005.

[2] W. R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. of the 5th Annual ACM/IEEE Intl. Conf. on Mobile Computing and Networking*, MobiCom '99, pages 174–185. ACM, 1999.

[3] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proc. of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, pages 565–574. IEEE, 2000.

[4] S. Kashyap, S. Deb, K. V. M. Naidu, R. Rastogi, and A. Srinivasan. Efficient gossip-based aggregate computation. In *Proc. of the 25th ACM Symposium on Principles of Database Systems*, PODS '06, pages 308–317. ACM, 2006.

[5] D. Mosk-Aoyama and D. Shah. Computing separable functions via gossip. In *Proc. of the 25th Annual ACM Symposium on Principles of Distributed Computing*, PODC '06, pages 113–122. ACM, 2006.

# Algorithms versus Complexity: Two Sides of the Same Coin

Valentine Kabanets
Simon Fraser University, Canada
kabanets@cs.sfu.ca

## 1. Introduction

There is an intimate relation between certain efficient algorithms and certain lower bounds. An *algorithm* for a problem is a procedure that allows one to figure out a correct answer for every given input instance of the problem. In contrast, a *lower bound* for a problem is a mathematical proof showing that there is no fast algorithm to solve the problem, no matter how clever your algorithm may be.

People are normally interested in finding efficient algorithms for their problems. When such an algorithm exists, and can be discovered, that is great! However, we know that this can't happen for all problems! Some natural problems simply do not have efficient algorithms. In such cases, people would love to have a mathematical proof that no algorithm (running in certain "small" amount of time) can solve the problem. Such a proof is called a *lower bound* for the problem.

Proving lower bounds appears even more difficult than finding clever efficient algorithms! But let us suppose we have such a lower bound for some "interesting" problem. Can we say anything more than just the "negative" result that there is no fast algorithm? In other words, can we use the clever ideas that went into the usually hard proof of a lower bound for something more, maybe some other *algorithm* for a related problem?

The surprising answer is 'Yes'! Algorithms and Lower Bounds are related: good algorithms for certain problems $X$ can be used to prove lower bounds for certain other problems $Y$, and conversely, lower bounds for certain problems $Y$ can be used to design efficient algorithms for certain other problems $X$.

## 2. Derandomization and Learning Algorithms

In this lecture, I will discuss some known examples of the connection between algorithms and lower bounds. I will start with the classical result in *derandomization* showing that every efficient *randomized* algorithm (the algorithm that flips random coins to help it find the correct solution to the problem) can always be made into an efficient *deterministic* algorithm, assuming we have lower bounds of a certain kind! This is a result of a long line of work in complexity theory [BM84, Yao82, NW94, BFNW93, IW97, Uma03].

I will survey some other results illustrating how one gets lower bounds from algorithms, and algorithms from lower bounds. Time permitting, I will also show a much more recent such result, where we get, unconditionally, efficient *learning* algorithms for a certain class of functions, from a known lower bound proof for that same class of functions [CIKK16, CIKK17].

Here the learning setting is as follows: you are given "black-box" (or oracle) access to an unknown function $f\colon \{0,1\}^n \to \{0,1\}$ from a class of "easy" functions (e.g., those functions computable by efficient algorithms of some restricted kind). Your learning algorithm is a randomized algorithm that, with high probability, must produce a new algorithm that computes some function $g\colon \{0,1\}^n \to \{0,1\}$ so that $g$ is "approximately the same as" $f$, i.e., $g$ and $f$ agree on all but a negligible fraction of $n$-bit inputs. In this way, you probabilistically "learn" $f$, by querying $f$ on a small number of inputs of your choice. (Your learning algorithm must be efficient, and so in particular is not allowed to query $f$ on all $2^n$ possible inputs.) This is a variant of the well-known PAC (Probabilistically Approximately Correct) setting for learning algorithms introduced by Valiant [Val84].

The result I'd like to talk about is that there is such a learning algorithm for learning the class of functions computable by circuits (computer chips) made up from constantly many layers of at most polynomially many logical gates, where the allowed logical gates are AND, OR, NOT, and MOD $p$, for a prime number $p \geq 2$; here the MOD $p$ gate outputs 1 if and only if the number of 1-valued inputs to the gate is divisible by $p$. Such a class of functions is called $AC^0[p]$ in complexity theory.

We get the *first* efficient learning algorithm for the class $AC^0[p]$ of functions, and the way we get the algorithm is by exploiting the known lower bounds against $AC^0[p]$-type algorithms (and currently no other, more direct, way is known)!

## 3. Assumed background

In the talk, I will assume that students have seen some algorithms, and are familiar with the concept of efficient (polynomial-time) algorithms. Some basic exposure to complexity, e.g., the famous "$P$ vs. $NP$" question, is helpful but not necessary.

## References


[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13:850–864, 1984.

[CIKK16] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 10:1–10:24. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[CIKK17] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Agnostic learning from tolerant natural proofs. In Klaus Jansen, José D. P. Rolim, David Williamson, and Santosh Srinivas Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, volume 81 of *LIPIcs*, pages 35:1–35:19. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[IW97] R. Impagliazzo and A. Wigderson. P=BPP if E requires exponential circuits: Derandomizing the XOR Lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.

[Uma03] C. Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences*, 67(2):419–440, 2003.

[Val84] Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[Yao82] A.C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

# On the Complexity of Domination and its Variations

Arti Pandey
Indian Institute of Technology Ropar
arti@iitrpr.ac.in

Let $G = (V, E)$ be a simple undirected graph. A set $D \subseteq V$ of a graph $G = (V, E)$ is called a *dominating set* of $G$ if $N_G[v] \cap D \neq \emptyset$ for all $v \in V$. The MINIMUM DOMINATION problem is to find a dominating set of minimum cardinality. The MINIMUM DOMINATION problem has wide applications in facility location problems. The MINIMUM DOMINATION problem is very well studied in literature. The decision version of the MINIMUM DOMINATION problem is NP-complete for general graphs and remains NP-complete for important graph classes like bipartite graphs, chordal graphs, planar graphs etc. Since, the problem remains NP-complete even for important graph classes, the following issues have also been studied in the literature: (i) to look for special classes of graphs where the problem admits polynomial time algorithms, (ii) to design good approximation algorithm and look for the limit of approximability by showing some inapproximation result, (iv) to obtain some structural results which are used in solutions of the above issues and are of independent interests as well.

Many important variations of domination have also been studied in literature, for example total domination, connected domination, disjunctive domination, restrained domination, outer-connected domination, semitotal domination etc. Total domination is also well studied problem in graph theory. The concept of total domination in graphs was introduced by Cockayne, Dawes, and Hedetniemi in 1980. A set $D \subseteq V$ of a graph $G$ is called a *total dominating set* of $G$ if $N_G(v) \cap D \neq \emptyset$ for all $v \in V$. Every total dominating set is also a dominating set. Total domination is the extension of domination to include redundancy, but it is expensive to implement. A relaxed form of total domination called semitotal domination was introduced by Goddard, Henning and McPillan in 2014. A set $D \subseteq V$ is called a *semitotal dominating set* of $G$ if $D$ is a dominating set of $G$, and every vertex in $D$ is within distance 2 of another vertex of $D$. The MINIMUM SEMITOTAL DOMINATION problem is to find a semitotal dominating set of minimum cardinality. The decision version of the MINIMUM SEMITOTAL DOMINATION problem is known to be NP-complete for general graphs.

In this talk, we will discuss about the MINIMUM DOMINATION problem and the MINIMUM SEMITOTAL DOMINATION problem. We will first briefly review the literature available for these problems. Then, we will discuss NP-hardness results, polynomial time algorithms and approximability results for these two problems which are our recent contributions.

## References

[1] W. Goddard and M. A. Henning and C. A. McPillan. Semitotal domination in graphs. *Util. Math.*, 94 (2014) 67-81.

[2] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs*, volume 208. Marcel Dekker Inc., New York, 1998.

[3] M. A. Henning. A survey of selected recent results on total domination in graphs. *Discrete Mathematics*, 309(1):32-63, 2009.

[4] M. A. Henning and A. Yeo, *Total Domination in Graphs*, Springer, 2013.

[5] M. A. Henning, A. J. Marcon. On matching and semitotal domination in graphs. Discrete Math., 324 (2014) 13–18.

[6] M. A. Henning and Arti Pandey, Algorithmic Aspects of Semitotal Domination in Graphs, *on arXiv*.

[7] B. S. Panda and Arti Pandey, Algorithmic aspects of open neighborhood location-domination in graphs, *Discrete Appl. Math.*, 216(2017), 290-306.

[8] Arti Pandey and B. S. Panda, Domination in some subclasses of bipartite graphs, *Accecpted subject to minor revision in Discrete Appl. Math.*

[9] B. S. Panda and Arti Pandey, Complexity of total outer-connected domination problem in graphs, *Discrete Appl. Math.*, 199 (2016), 110-122.

# Proof Complexity and its Applications

Antonina Kolokolova
Memorial University of Newfoundland, Canada
kol@cs.toronto.edu

The famous P vs NP question is formulated in terms of decision problems, where solving a computational problem means providing a yes/no answer: for example, "does there exist a schedule satisfying a given system of constraints?" Yet in practice it is the actual witness of the "yes" answer which is often the goal: produce a schedule satisfying given constraints, if it is exists. For problems in NP, by its very definition, we know that not only such witnesses to "yes" answers exist, but also their size is small.

But what if the answer to your problem instance is "no"? Ideally, we would like an explanation, a reason why there was no way to satisfy the constraints. In short, we want a proof that the answer is "no", just like we want a witness for the "yes" answer. But what could such a proof be? And how large could it become?

This question is studied in *proof complexity*, the area combining complexity theory and mathematical logic. Usually, the question is formulated as studying complexity of proofs of tautologies: formulas that are always true. A propositional formula is a tautology if and only if its negation is unsatisfiable, making it a natural counterpart to the Satisfiability problem: given a propositional formula, is it satisfiable? That said, proof complexity extends well beyond the propositional setting, from the complexity of proving quantified statements to reasoning about systems of equations to bounded arithmetic.

I will give an overview of what is known about the complexity of proofs in common (mainly propositional) proof systems, discussing in particular what kinds of reasoning is hard for the simpler and better understood systems such as resolution. I will also talk about stronger systems, closer to natural deduction. Much is unknown about these systems; it is still open whether there is anything hard for them (or even whether there exists a set of rules and axioms in which there is always a proof of size comparable to the input tautology).

Propositional proof complexity started in earnest with the 1979 paper by Cook and Reckhow [CR79]. There, they have formally defined an abstract notion of a propositional proof system (with complexity measures built in), and studied several types of proof systems, in particular proving that a wide class of systems based on a natural deduction rule are all equivalent.

The propositional proof system that received the most attention is a provably weaker system called "resolution". Rather than operating with arbitrary formulas, it works with a formula in CNF (product-of-sums) form: an AND of ORs of (possibly negated) variables; the only rule is a version of *modus ponens* adapted to this setting (the resolution rule). This is one of the few proof systems for which we know that there are formulas requiring exponentially large proofs; moreover, we know a number of examples of such formulas, including the well-studied pigeonhole principle [Hak85]. Though it started as a purely theoretical endeavor, resolution proof system became a basis for a powerful class of modern heuristics used in SAT solvers [BBH09]. Nowadays, SAT solvers are used to solve a wide variety of constraint satisfaction problems from software verification to the US broadband spectrum auction, outperforming traditional equations-based methods from operations research in a number of applications. When exactly they perform so spectacularly well, in spite of the underlying problem being both NP-hard and co-NP hard, is still an open problem.

Time permitting, I will also talk about proof systems based on numerical representations of formulas, and combinations of approaches.

See excellent surveys by Pudlák [Pud08], Beame/Pitassi [BP01], Buss [Bus12] and Nordström [Nor15] for more details on propositional proof systems, and a survey by Pitassi and Tzameret [PT16] on algebraic proof systems.

## References


[BBH09] A Biere, A Biere, M Heule, H van Maaren, and T Walsh. *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications.* IOS Press, 2009.

[BP01] Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. In *Current Trends in Theoretical Computer Science*, pages 42–70. 2001.

[Bus12] Samuel R. Buss. Towards NP-P via proof complexity and search. *Ann. Pure Appl. Logic*, 163(7):906–917, 2012.

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979.

[Hak85] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

[Nor15] Jakob Nordström. New wine into old wineskins: A survey of some pebbling classics with supplemental results. 2015.

[PT16] Toniann Pitassi and Iddo Tzameret. Algebraic proof complexity: Progress, frontiers and challenges. *ACM SIGLOG News*, 3(3):21–43, August 2016.

[Pud08] Pavel Pudlák. Twelve problems in proof complexity. In *Proceedings of the 3rd international conference on Computer science: theory and applications*, pages 13–27. Springer-Verlag, 2008.

# The Median (anti-Median) Problem on Graphs

Ambat Vijayakumar
Cochin University of Science and Technology
vambat@gmail.com

The median of a graph is one of the centrality concepts, together with the notions such as center and centroid, is defined as follows.

The status of a vertex $u$ in a graph $G$ is the sum of the distances from $u$ to all the vertices in $G$. The subgraph induced by the vertices of minimum (maximum) status in $G$ is the median (anti-median) of $G$, denoted by $M(G)(AM(G))$. Thus the median vertices are the facility locations that concern with minimum average distance in a network. The problem of finding a graph $H$ with a given graph $G$ as the median is referred to as the median problem.

In [12], Slater proved that any graph is a median of some connected graph. The number of vertices used for such a construction was shown to be $\leq 2|V(G)|$ in [6] and it was improved to $2|V(G)| - \delta(G) + 1$ in [5]. The anti-median problem can be defined similarly and a solution to it can be seen in [2].

The problem of simultaneous embedding of medians deals with the embedding of two graphs $G_1$ and $G_2$ in graph $H$ such that $M(H) \cong G_1$ and $AM(H) \cong G_2$. In [1], such an embedding is given with $d(G_1, G_2) \geq 2$. This work also presents the simultaneous embedding of $G_1$ and $G_2$ as convex subgraphs of $H$ with $M(H) \cong G_1$, $AM(H) \cong G_2$ and $d(G_1, G_2) \geq \lfloor d(G_1)/2 \rfloor + \lfloor d(G_2)/2 \rfloor + 2$. An improved solution to these problems with $d(G_1, G_2) \geq 1$ for every $G_1$ and $G_2$ is in [9].

The study of the median(anti-median) problem for different classes of graphs is also significant since the median constructions for general graphs cannot be directly applied to many networks as their underlying graph belong to different classes of graphs. See [11, 14, 7, 15, 8, 10] for some works on cographs, distance hereditary graphs, Ptolemaic graphs, $k$-partite graphs and symmetric bipartite graphs. A construction with $M(H) = G$ and $G$ is convex in $H$ is in [4].

This talk aims to bring the significance and recent results on the median problem in Graph Theory.

## References

[1] K. Balakrishnan, B. Brešar, M. Changat, S. Klavžar, M. Kovše and A. R. Subhamathi, Simultaneous embeddings of graphs as median and antimedian subgraphs, Networks 56(2010), 90-94.

[2] H. Bielak and M. M. Syslo, Peripheral vertices in graphs, Studia Sci. Math. Hungar. 18(1983), 269-275.

[3] F. Buckley and F. Harary, Distance in Graphs, Addison-Wesley, Redwood City, CA(1990).

[4] P. Dankelmann and G. Sabidussi, Embedding graphs as isometric medians. Discrete Appl. Math. 156(2008), 2420-2422.

[5] G. R. T. Hendry, On graphs with prescribed median I, J. Graph Theory, 9(1965), 477-481.

[6] Z. Miller, Medians and distance sequences in graphs, Ars Combin., 15(1983), 169-177.

[7] J. Nieminen, The center and the distance center ofa Ptolemaic graph, Oper. Res. Lett. 7 (1988) 9194.

[8] K. Pravas and A. Vijayakumar, The median problem on k-partite graphs, Discuss. Math. Graph Theory, 35 (2015), 439-446.

[9] K. Pravas and A. Vijayakumar, Convex median and anti-median at prescribed distance, Journal of Combinatorial Optimization, 33(2017)1021-1029.

[10] K. Pravas and A. Vijayakumar, The median problem on symmetric bipartite graphs, proceedings of the International Conference on Theoretical Computer Science and Discrete Mathematics 2016, Kalasalingam University: ICTCSDM 2016, LNCS 10398, pp. 262-270, 2017.

[11] S. B. Rao and A. Vijayakumar, On the median and the anti-median of a cograph, Int. J. Pure Appl. Math. 46(5)(2008), 703-710.

[12] P. J. Slater, Medians of arbitrary graphs, J. Graph Theory 4 (1980), 389-392.

[13] C. Smart and P. J. Slater, Center, median and centroid subgraphs, Networks, 34, 4 (1999), 303-311.

[14] H. G. Yeh and G. J. Chang, Centers and medians of distance-hereditary graphs, Discrete Math. 265(2003), 297-310.

[15] S.V. Yushmanov; On the median of Ptolemaic graphs, Issled Oper ASU 32 (1988), 67-70.

[16] B. Zelinka, Medians and peripherians of trees, Arch. Math (Brno) (1968), 87-95.

# Understanding Approximation Algorithms

Daya Gaur
University of Lethbridge, Canada
daya.gaur@uleth.ca

Approximation algorithms [11, 10] trade the quality of solution for time. This is a general and natural way to address the "complexity" of a problem. In this talk aimed at research scholars and graduate students, I will illustrate various techniques used to design and analyze approximation algorithms. We will examine approximation algorithms based on dynamic programming, linear programming [8], primal-dual schema and combinatorial methods. We will look at a variety of problems across several domains. Some of the problems that we will consider are i) scheduling of vehicles with the release and handling times [4], ii) vehicle routing problem [9], iii) covering in a geometric setting [3, 5], iv) balanced cuts in graphs [7, 2], v) coloring in hypergraphs [6], and vi) coverage in sensor networks [1]. Each problem will illustrate a different method of design and analysis.

## References

[1] R. Benkoczi, D. R. Gaur, and M. Thom. A 2-approximation algorithm for barrier coverage by weighted non-uniform sensors on a line. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 95–111. Springer, Cham, 2016.

[2] S. Choudhury, D. R. Gaur, and R. Krishnamurti. An approximation algorithm for max k-uncut with capacity constraints. *Optimization*, 61(2):143–150, 2012.

[3] D. R. Gaur and B. Bhattacharya. Covering points by axis parallel lines. In *Proc. 23rd European Workshop on Computational Geometry*, pages 42–45. Citeseer, 2007.

[4] D. R. Gaur, A. Gupta, and R. Krishnamurti. A 5/3-approximation algorithm for scheduling vehicles on a path with release and handling times. *Information Processing Letters*, 86(2):87–91, 2003.

[5] D. R. Gaur, T. Ibaraki, and R. Krishnamurti. Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. *Journal of Algorithms*, 43(1):138–152, 2002.

[6] D. R. Gaur and R. Krishnamurti. Simple approximation algorithms for maxnaesp and hypergraph 2-colorability. *Journal of combinatorial optimization*, 5(2):167–173, 2001.

[7] D. R. Gaur, R. Krishnamurti, and R. Kohli. Conflict resolution in the scheduling of television commercials. *Operations research*, 57(5):1098–1105, 2009.

[8] R. Krishnamurti and D. R. Gaur. LP rounding and extensions., 2007.

[9] R. R. Singh and D. R. Gaur. Cumulative VRP: A simplified model of green vehicle routing. In *Sustainable Logistics and Transportation*, pages 39–55. Springer, Cham, 2017.

[10] V. V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[11] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.