CALDAM 2023

Indo-Dutch Pre-Conference School on Algorithms and Combinatorics

February 6 - 7, 2023

Outline of Lectures

Organized by

Sponsored by

CALDAM 2023 Pre-Conference School on Algorithms and Combinatorics

February 6-7, 2023

Organized by:

Dhirubhai Ambani Institute of
Information and Communication Technology,
Gandhinagar, Gujarat 382007, India

Coordinated by:

- Bodo Manthey, (Co-Convenor)
  University of Twente, The Netherlands
- R. S. Lekshmi,
  PSG College of Technology, India
- Sunitha VadivelMurugan, (Co-Convenor)
  DA-IICT, India

Sponsored by:

Science and Engineering Research Board,
Department of Science and Technology,
Government of India

Preface

Discrete Mathematics studies mathematical structures that are discrete, rather than continuous, such as integers, graphs, logic, algorithms, and their operations, etc. Discrete Mathematics concerns itself with problems of the following kinds: (1) finding an optimal/extremal object from a large or an 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 working in the area of Discrete Mathematics and Computer Science, by exposing them to the recent developments and applications in Computational Geometry, Algorithms, Combinatorics, and Graph Theory.

The school is aimed at fulfilling two purposes: (i) as a Pre-Conference School for CALDAM 2023, and (ii) as an Indo-Dutch School on Algorithms and Combinatorics. The school is organized by the Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India. The school is funded by the Science and Engineering Research Board, Department of Science and Technology, Government of India.

Bodo Manthey	R.S. Lekshmi	Sunitha VadivelMurugan
University of Twente	PSG College of Tech.	DA-IICT

February 05, 2023

Lecture Schedule

Time	February 06, 2023	February 07, 2023
08:00 – 09:15	**Registration & Inauguration**	
09:15 – 10:30	**Technical Session - 1** **Arrangements, Partitions, and Applications** Mark de Berg	**Technical Session - 4** **Smoothed Analysis and its Applications to Local Search Heuristics** Jessie van Rhijn
10:30 – 11:00	**Break**	
	Technical Session - 2 **Graph Coloring Problems** Rishi Ranjan Singh	**Technical Session - 5** **Graph Classes arising from the Perfect Matching Polytope** Nishad Kothari
11:45 – 12:30	**Games on Graphs and Eternal Vertex Cover** Neeldhara Misra	**Some Open Problems in Computational Group Theory** Bireswar Das
12:30 – 13:45	**Lunch**	
13:45 – 15:00	**Technical Session - 3** **From Approximate to Exact Integer Programming** Daniel Dadush	**Technical Session - 6** **Random Metrics in the Analysis of Algorithms** Bodo Manthey
15:00 – 15:30	**Break**	
15:30 – 17:00	**Young Researchers Forum** **Short Presentations** by Manoj Changat, A. Mohanapriya, Pavan P.D., Supraja D.K., Adri Bhattacharya, Ritam Manna Mitra, Sangam Balachandar Reddy, Saraswati Girish Nanoti, Pavithra R.	**Participants' Feedback & Conclusion**
19:30 onwards	**Dinner**	

Lecture Outline

Arrangements, Partitions, and Applications

Mark de Berg *
Department of Mathematics and Computer Science,
TU Eindhoven, the Netherlands, M.T.d.Berg@tue.nl.

Computational geometry [5] is the branch of algorithms research that deals with spatial data. It has many applications, ranging from geographic information science and computer-aided design to robotics and molecular biology. Scientifically, the area is closely related to discrete and combinatorial geometry. Computational geometry started in 1980s and since then many beautiful techniques have been developed. In this lecture I will discuss *(substructures in) arrangements* and techniques for *partitioning* arrangements, which are geometric tools that underlie many old as well as many recent geometric algorithms.

(Substructures in) arrangements. Let L be a set of n lines in $\mathbb{R}^2$. Then $\mathcal{A}(L)$, the *arrangement* induced by L is the subdivision of $\mathbb{R}^2$ induced by L. More precisely, the arrangement $\mathcal{A}(L)$ consists of 2-dimensional faces (the *cells* of the arrangement), 1-dimensional faces (the *edges* of the arrangement), and 0-dimensional faces (the *vertices* of the arrangement); see Figure 1(i) for an illustration. Arrangements can also be studied for line segments in $\mathbb{R}^2$, and for hyperplanes, surfaces, or surface patches in $\mathbb{R}^d$ for $d \geqslant 2$.

Arrangements play a fundamental role in computational geometry, because many problems can be phrased in terms of arrangements when translated into a suitable parametric space. Depending on the problem at hand, one is then often interested in only a part of the arrangement, such as a *single cell*, or the *upper envelope*; see Figure 1(i) and (iii). Therefore it is important to analyze the complexity of substructures in arrangements. For example, how many vertices and edges can the upper envelope (or: a single cell) of n curves in $\mathbb{R}^2$ have?

In the lecture I will discuss some basic results on the complexity of substructures in arrangements. We will see the connection to Davenport-Schinzel sequences [8], as well as the celebrated *Clarkson-Shor technique* for analyzing the complexity of the $(\leqslant k)$-*level* in arrangements. I will also give examples of problems that can be solved by computing or analyzing (substructures in) arrangements, in particular for problems involving moving points.

*MdB is supported by the Dutch Research Council (NWO) through Gravitation-grant NETWORKS-024.002.003.

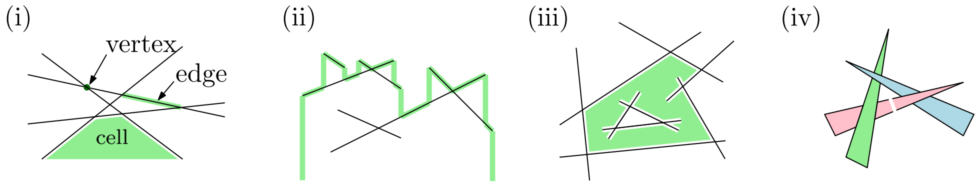

Fig. 1: (i) An arrangement induced by four lines, with 11 cells, 16 edges, and 6 vertices. (ii) The upper envelope of a set of segments. (iii) A single cell in an arrangement of segments. (iv) Three triangles with cyclic overlap in the depth order relation. Cutting the red triangle as indicated produces a set of four objects (the green and blue triangle and the two red pieces) without cyclic overlap.

Partitioning arrangements. Divide-and-conquer is one of the most important algorithmic design techniques. Typically the input is divided into subsets, on which suitable subproblems are defined which are then solved recursively. When applied to problems involving spatial data, one often uses *geometric divide-and-conquer*. Here the space is partitioned into regions, and a subproblem is defined for the objects lying (partially) inside each region. I will discuss techniques to obtain good partitionings for arrangements. In particular, we will study *cuttings*, an old but still powerful technique, and *polynomial partitions* [7, 1], a more recent technique. As an example of an application of these techniques, we will study the following problem [3, 4]: Suppose we are given a set S of n line segments (or: triangles) in $\mathbb{R}^3$, and we wish to cut these segments (or triangles) into pieces such that their depth-order relation is acyclic, as in Figure 1(iv). How many cuts are needed in the worst case?

References

[1] P.K. Agarwal, B. Aronov, E. Ezra, and J. Zahl. An efficient algorithm for generalized polynomial partitioning and its applications. *SIAM J. Comput.* 50:760–787, 2021.

[2] B. Aronov Y. Miller, and M. Sharir. Eliminating depth cycles among triangles in three dimensions. *Discret. Comput. Geom.* 64: 627–653, 2020.

[3] B. Aronov and M. Sharir. Almost tight bounds for eliminating depth cycles in three dimensions. *Discret. Comput. Geom.* 59: 725–741, 2018.

[4] M. de Berg. Removing depth-order cycles among triangles: An algorithm generating triangular fragments. *Discret. Comput. Geom.* 65: 450–469, 2021.

[5] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications.* Springer, 2008.

[6] K.L. Clarkson and P.W. Shor. Application of Random Sampling in Computational Geometry, II. *Discret. Comput. Geom.* 4: 387–421, 1989.

[7] L. Guth. Polynomial partitioning for a set of varieties. *Math. Proc. Camb. Phil. Soc.* 159:459–469, 2015.

[8] M. Sharir and P.K. Agarwal. *Davenport-Schinzel sequences and their Geometric Applications*. Cambridge University Press, 1995.

Graph Coloring Problems

Rishi Ranjan Singh
Indian Institute of Technology Bhilai,
Sejbahar, Raipur, India, rishi@iitbhilai.ac.in

Scheduling problems are promptly visible in designing time-tables, scheduling jobs on CPU, accessing data from hard-discs, routing vehicles to minimize fuel consumption etc. [9, 7, 6]. Graph Coloring problems are closely related to scheduling problems. Graph coloring problems also exhibit similar real-world applications [5, 3, 2, 1]. The objective in a general graph coloring problem is to find the least number of colors to fulfill the demand of coloring. Coloring problems relate to those scheduling problems where the objective is to minimize the number of machines required to meet the demand. For example, the vertex coloring problem requires finding the minimum cost color assignment on the vertices such that the assigned colors on the vertices satisfy some given constraints.

This talk will describe a few variants of graph coloring problems and some primary results for these versions of graph coloring problems. It will start with proper vertex, edge, and total coloring problems. Various generalizations of graph coloring problems or coloring with additional constraints, for example, list coloring, path coloring etc. [10, 4] will be explained next. Afterward, this talk will touch upon improper graph coloring problems [8], which are different type of graph coloring problems. A brief discussion on clustered graph coloring [11] will be done towards the end of this talk.

References

[1] Shamim Ahmed. Applications of graph coloring in modern computer science. *International Journal of Computer and Information Technology*, 3(2):1–7, 2012.

[2] Nicolas Barnier and Pascal Brisset. Graph coloring for air traffic flow management. *Annals of operations research*, 130(1):163–178, 2004.

[3] D. de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, 1985.

[4] Piotr Formanowicz and Krzysztof Tanaś. A survey of graph coloring-its types, methods and applications. *Foundations of Computing and Decision Sciences*, 37(3):223–238, 2012.

[5] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[6] Daya Ram Gaur, Apurva Mudgal, and Rishi Ranjan Singh. Routing vehicles to minimize fuel consumption. *Operations Research Letters*, 41(6):576–580, 2013.

[7] Robert Geist and Stephen Daniel. A continuum of disk scheduling algorithms. *ACM Transactions on Computer Systems (TOCS)*, 5(1):77–92, 1987.

[8] Ross J Kang. Improper colourings of graphs. 2008.

[9] ES Page. An approach to the scheduling of jobs on machines. *Journal of the Royal Statistical Society: Series B (Methodological)*, 23(2):484–492, 1961.

[10] Panos M Pardalos, Thelma Mavridou, and Jue Xue. The graph coloring problem: A bibliographic survey. In *Handbook of combinatorial optimization*, pages 1077–1141. Springer, 1998.

[11] David Wood. Defective and clustered graph colouring. *The Electronic Journal of Combinatorics*, 1000:23–13, 2018.

Games on Graphs and Eternal Vertex Cover

Neeldhara Misra
Indian Institute of Technology Gandhinagar,
Palaj, Gandhinagar, Gujarat, India, neeldhara.m@iitgn.ac.in

Abstract

Pursuit-Evasion games typically involve two types of players: those in pursuit (aka cops) and the so-called evaders (aka robbers) [1]. The backdrop is usually a network with specific rules that dictate how the players can move. These are turn-based games, and one is usually interested in knowing whether and how the evaders can be cornered. We often want to do this as inexpensively and quickly as we can, which leads to questions about optimizing the number of cops we deploy and the number of rounds that the game will last. It turns out that answers to these questions often have deep connections with the structure of the underlying network. This first part of this talk will involve a few glimpses of such connections.

In the second part of the talk, we will focus on the ETERNAL VERTEX COVER problem, which is a natural dynamic variant of the vertex cover problem. Here, have a two player game in which guards are placed on some vertices of a graph. In every move, one player (the attacker) attacks an edge. In response to the attack, the second player (the defender) moves some of the guards along the edges of the graph in such a manner that at least one guard moves along the attacked edge. If such a movement is not possible, then the attacker wins. If the defender can defend the graph against an infinite sequence of attacks, then the defender wins. The minimum number of guards with which the defender has a winning strategy is called the eternal vertex cover number of the graph G. We will survey some recent developments around Eternal Vertex Cover and closely related problems.

References

[1] Bonato, A. The game of cops and robbers on graphs. (American Mathematical Soc.,2011)

From approximate to exact integer programming

Daniel Dadush
Centrum Wiskunde & Informatica (CWI),
the Netherlands, dadush@cwi.nl

Approximate integer programming is the following: For a convex body $K \in R^n$, either determine whether $K \cap Z^n$ is empty, or find an integer point in the convex body scaled by 2 from its center of gravity c. Approximate integer programming can be solved in time $2^{O(n)}$ while the fastest known methods for exact integer programming run in time $O(n)^n$. So far, there are no efficient methods for integer programming known that are based on approximate integer programming. Our main contribution are two such methods, each yielding novel complexity results.

First, we show that an integer point $x \in K \cap Z^n$ can be found in time $2^{O(n)}$, provided that the remainders of each component $x(mod\ l)$ for some arbitrarily fixed $l \geq 5(n+1)$ of x are given. The algorithm is based on a cutting-plane technique, iteratively halving the volume of the feasible set. The cutting planes are determined via approximate integer programming. Enumeration of the possible remainders gives a $O(n)^n$ algorithm for general integer programming. This matches the current best bound of an algorithm by Dadush (2012) that is considerably more involved. Our algorithm also relies on a new asymmetric approximate Carathéodory theorem that might be of interest on its own.

Our second method concerns integer programming problems in equation-standard form $Ax = b, 0 \leq x \leq u, x \in Z^n$. Such a problem can be reduced to the solution of $\prod_{i=1}^{n} O(\log u_i + 1)$ approximate integer programming problems. This implies, for example that knapsack or subset-sum problems with polynomial variable range $0 \leq x_i \leq p(n)$ can be solved in time $(logn)^{O(n)}$. For these problems, the best running time so far was $O(n)^n$.

Smoothed Analysis and its Applications to Local Search Heuristics

Jesse van Rhijn
University of Twente,
Enschede, the Netherlands, j.vanrhijn@utwente.nl

In the analysis of algorithms, one often considers the worst case time complexity of an algorithm. This quantity brings with it strong performance guarantees. However, the running time may in practice be much shorter than the worst case time complexity. As an example, take the simplex method for linear programming. Although its worst case running time is exponential in the number of variables, it often terminates in polynomial time. This has its consequences in practice: the simplex method remains a popular tool for solving linear programs, even though guaranteed polynomial-time algorithms are available.

As a way to resolve this conflict between practice and theory, Spielman & Teng developed smoothed analysis (SA). In this framework, one perturbs worst case instances by random variables, in the hopes that the running time of an algorithm on the resulting instance is closer to practical experience. In its first application, Spielman & Teng provided polynomial smoothed time complexity bounds for the simplex method.

A popular optimization paradigm that often exhibits poor worst case behavior is local search. Such optimization algorithms function by modifying candidate solutions of an optimization problem in order to reduce their objective value. Examples of local search methods include k-opt for the Travelling Salesperson Problem (TSP), k-means for clustering, and the flip heuristic for Max Cut. These three examples also each show exponential worst case running time, and polynomial smoothed time complexity.

In this talk, we will acquaint ourselves with the paradigm of smoothed analysis, and in particular its application to local search heuristics. We will survey some results on this topic, which demonstrate the effectiveness of SA in advancing our understanding of local search heuristics. We will moreover perform a smoothed analysis of 2-opt, a simple heuristic for the TSP. We conclude with some open problems and challenges for the method.

Some references related to this work are [1–7]

References

[1] David Arthur, Bodo Manthey, and Heiko Röglin. Smoothed Analysis of the k-Means Method. *Journal of the ACM*, 58(5):19:1–19:31, October 2011.

[2] Ali Bibak, Charles Carlson, and Karthekeyan Chandrasekaran. Improving the Smoothed Complexity of FLIP for Max Cut Problems. *ACM Transactions on Algorithms*, 17(3):19:1–19:38, July 2021.

[3] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP. *Algorithmica*, 68(1):190–264, January 2014.

[4] Matthias Englert, Heiko Röglin, and Berthold Vöcking. Smoothed Analysis of the 2-Opt Algorithm for the General TSP. *ACM Transactions on Algorithms*, 13(1):10:1–10:15, September 2016.

[5] Bodo Manthey. Smoothed Analysis of Local Search. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, pages 285–308. Cambridge University Press, Cambridge, 2021.

[6] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, May 2004.

[7] Andrea Vattani. k-means Requires Exponentially Many Iterations Even in the Plane. *Discrete & Computational Geometry*, 45(4):596–616, June 2011.

Graph Classes arising from the Perfect Matching Polytope

Nishad Kothari
Indian Institute of Technology Madras,
Chennai, Tamil Nadu, India, nishad@cse.iitm.ac.in

Matchings and perfect matchings have received considerable attention in Graph Theory as well as in other related areas (such as, but not limited to, algorithms and optimization); see Lovász and Plummer [LP86]. In particular, the perfect matching polytope has been studied extensively; see Schrijver [Sch03]. However, there still remain many questions — pertaining to this polytope and related graph classes — to which we do not know the answers. Before going further, let's make sure that we are on the same page. (In other words, let's state things formally.)

For a (loopless undirected) graph G and a vertex $v \in V(G)$, we use $\partial(v)$ to denote the set of edges incident with v. A set of edges, say $M \subseteq E(G)$, is a *perfect matching* if it contains precisely one edge incident with each vertex (that is, if $|M \cap \partial(v)| = 1$ for each $v \in V(G)$). One may view each perfect matching M as a 0-1 vector, say χ^M, in $\mathbb{R}^{|E(G)|}$: if an edge e belongs to M, the corresponding component χ^M_e is 1; otherwise, χ^M_e is 0. We refer to these vectors as the *perfect matching vectors* of G; now, the *perfect matching polytope of* G, denoted by $\mathcal{PMP}(G)$, is their convex hull. For instance, the famous Petersen graph (on 10 vertices and 15 edges) has precisely 6 perfect matchings (convince yourself — it is a good exercise); consequently, its perfect matching polytope is a convex body in $\mathbb{R}^{15}$ with precisely 6 extreme points (aka corners).

In what follows, I will describe two graphs classes whose definitions are motivated by natural questions and properties pertaining to the perfect matching polytope. We will first see polyhedral definitions of these graphs classes, and later discuss equivalent graph-theoretical formulations (in terms of 'conformal subgraphs').

Birkhoff-von Neumann Graphs:

It is well-known that every convex polytope is the set of solutions to a system of linear equations and inequalities (that is, the feasible region of a linear program). This leads to a natural problem (in polyhedral combinatorics): given a graph G, formulate a linear program whose feasible region is precisely its perfect matching polytope $\mathcal{PMP}(G)$. Below, we discuss two necessary conditions for a vector $x \in \mathbb{R}^{|E(G)|}$ to belong to $\mathcal{PMP}(G)$.

Suppose that $x \in \mathcal{PMP}(G)$. In other words, x can be expressed as a convex combination of perfect matching vectors of G. Since each perfect matching vector is a 0-1 vector, clearly

x is non-negative. That is, $x_e \geq 0$ for each edge $e \in E(G)$ — these inequalities are called *non-negativity constraints*. Furthermore, if χ^M is a perfect matching vector, it satisfies $\sum_{e\in\partial(v)} \chi^M_e = 1$ at each vertex $v \in V(G)$. This implies that x also satisfies the same — i.e., $\sum_{e\in\partial(v)} x_e = 1$ — these equations are called *degree constraints*.

As noted above, given a graph G, every vector in $\mathcal{PMP}(G)$ satisfies the corresponding non-negativity as well as degree constraints. Unfortunately, these necessary conditions are not always sufficient! For instance, the Petersen graph has two vertex-disjoint 5-cycles; see if you can use this hint to come up with a vector that satisfies the non-negativity and degree constraints, but does not belong to the perfect matching polytope (of the Petersen graph). Interestingly however, classical results of Birkhoff and of von Neumann, imply that these necessary conditions are in fact sufficient in the case of bipartite graphs.

We say that a graph G is *Birkhoff-von Neumann* if $\mathcal{PMP}(G)$ is completely characterized by the non-negativity and degree constraints. It follows from the above discussion that all bipartite graphs are Birkhoff-von Neumann, whereas the Petersen graph is not. Interestingly, many nonbipartite graphs are also Birkhoff-von Neumann; one such example is an odd wheel (that is, any graph obtained from an odd cycle graph by adding a new vertex and making it adjacent to all other vertices). This leads us to the following decision problem.

Decision Problem 1: Given a graph G, decide whether G is Birkhoff-von Neumann.

The above decision problem is in the complexity class **co-NP**, but is <u>not</u> known to be in **NP**; thus it is also <u>not</u> known to be in **P**.

<u>PM-compact Graphs</u>:

Given any polytope $\mathcal{P}$, one may construct a simple undirected graph known as the *skeleton of* $\mathcal{P}$ and denoted by $\mathbb{S}(\mathcal{P})$, as follows. The graph $\mathbb{S}(\mathcal{P})$ has precisely one vertex corresponding to each extreme point (that is, face of dimension zero) of $\mathcal{P}$, and two vertices of $\mathbb{S}(\mathcal{P})$ are adjacent if and only if the corresponding extreme points belong to a common "edge" (that is, face of dimension one) of $\mathcal{P}$. For example, if $\mathcal{P}$ denotes the three-dimensional cube, then $\mathbb{S}(\mathcal{P})$ is the unique bipartite cubic (aka 3-regular) simple graph on 8 vertices (and 12 edges) — known as the cube graph (for obvious reasons).

We say that a polytope $\mathcal{P}$ is *compact* if its skeleton $\mathbb{S}(\mathcal{P})$ is a complete graph. For instance, the three-dimensional cube is not compact; however, the tetrahedron is compact. In the same spirit, a graph G is *perfect matching compact*, or simply *PM-compact*, if its perfect matching polytope $\mathcal{PMP}(G)$ is compact. This leads us to our next decision problem.

Decision Problem 2: Given a graph G, decide whether G is PM-compact.

As before, the above decision problem is in the complexity class **co-NP**, but is <u>not</u> known to be in **NP**; thus it is also <u>not</u> known to be in **P**.

Conformal Subgraphs:

For most problems pertaining to the study of perfect matchings, one may restrict attention to *matching covered graphs* — that is, connected nontrivial graphs with the additional property that each edge belongs to some perfect matching. The same holds for the above mentioned decision problems.

In our previous discussion, we defined the Birkhoff-von Neumann and PM-compact properties using the perfect matching polytope. Is there a way to define them using just graph-theoretical notions and concepts? Yes, there is.

A subgraph H of a graph G is *conformal* if the graph $G - V(H)$ has a perfect matching. For instance, in the case of the Petersen graph, all of its 8-cycles are conformal; whereas, none of its 6-cycles are conformal (check for yourself). Conformality plays an important role in the study of perfect matchings, and it is intrinsically related to the ear decomposition theory of matching covered graphs (which we will skip here).

We use the term *bicycle* to refer to a pair of vertex-disjoint cycles. Observe that if (Q_1, Q_2) is a conformal bicycle in a matching covered graph G, then $G - V(Q_1) - V(Q_2)$ has a perfect matching; thus, either Q_1 and Q_2 are both odd cycles, or they are both even cycles. In the former case, we say that (Q_1, Q_2) is an *odd conformal bicycle*; in the latter case, we say that (Q_1, Q_2) is an *even conformal bicycle.*

The following result of Balinski and Russakoff [BR74], and independently due to Chvátal [Chv75], gives a graph-theoretical way of seeing why Decision Problem 2 is in **co-NP**.

Theorem 1. *A matching covered graph is not PM-compact if and only if it has an even conformal bicycle.*

Likewise, the following result of Balas [Bal81] provides a graph-theoretical explanation as to why Decision Problem 1 is in **co-NP**.

Theorem 2. *A matching covered graph is not Birkhoff-von Neumann if and only if it has an odd conformal bicycle.*

Recent Developments:

In this section, by 'graph', we mean 'matching covered graph'.

In a recent paper with Carvalho, Lucchesi and Murty [LCKM18], we established that the problem of characterizing Birkhoff-von Neumann graphs is equivalent to another important problem in Matching Theory — that of characterizing "solid" graphs. Earlier, they [CLM06] provided a complete characterization of Birkhoff-von Neumann planar graphs.

In the case of PM-compact graphs, Wang, Lin, Carvalho, Lucchesi, Sanjith and Little [WLC+13] characterized those that are either bipartite or "near-bipartite", whereas Wang, Shang, Lin and Lucchesi [WSLL14] characterized those that are cubic and claw-free.

Most recently, in a joint work with Carvalho, Wang and Lin [CKWL20], we provided a complete characterization of graphs that are Birkhoff-von Neumann as well as PM-compact — or, equivalently, graphs that do not have a conformal bicycle. Thus, the problem of deciding whether a matching covered graph has a conformal bicycle is in **co-NP**, in **NP** as well as in **P**; unfortunately, the parity requirement is the obstacle!

References

[Bal81] E. Balas. Integer and fractional matchings. In P. Hansen, editor, *Studies on Graphs and Discrete Programming*, volume 59, pages 1–13. North Holland Mathematics Studies, 1981.

[BR74] M. L. Balinksi and A. Russakoff. On the assignment polytope. SIAM *Review*, 16(4):516–525, 1974.

[Chv75] V. Chvátal. On certain polytopes associated with graphs. *J. Combin. Theory Ser. B*, 18:138–154, 1975.

[CKWL20] M. H. Carvalho, N. Kothari, X. Wang, and Y. Lin. Birkhoff-von neumann graphs that are pm-compact. SIAM *J. Discrete Math.*, 34(3):1769–1790, 2020.

[CLM06] M. H. Carvalho, C. L. Lucchesi, and U. S. R. Murty. How to build a brick. *Discrete Math.*, 306:2383–2410, 2006.

[LCKM18] C. L. Lucchesi, M. H. Carvalho, N. Kothari, and U. S. R. Murty. On two unsolved problems concerning matching covered graphs. SIAM *J. Discrete Math.*, 32(2):1478–1504, 2018.

[LP86] L. Lovász and M. D. Plummer. *Matching Theory*. Number 29 in Annals of Discrete Mathematics. Elsevier Science, 1986.

[Sch03] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, 2003.

[WLC+13] X. Wang, Y. Lin, M. H. Carvalho, C. L. Lucchesi, G. Sanjith, and C. H. C. Little. A characterization of PM-compact bipartite and near-bipartite graphs. *Discrete Math.*, 313:772–783, 2013.

[WSLL14] X. Wang, W. Shang, Y. Lin, and C. L. Lucchesi. A characterization of PM-compact claw-free cubic graphs. *Discrete Math. Algorithms Appl.*, 6(2), 2014.

Some Open Problems in Computational Group Theory

Bireswar Das
Indian Institute of Technology Gandhinagar,
Palaj, Gandhinagar, Gujarat, India, bireswar@iitgn.ac.in

In this talk we will first introduce some basic algorithmic tools in computational group theory. One of the most important problems in computational group theory, whose complexity status is not yet known, is the group isomorphism problem when the groups are given by their Cayley tables. It is open whether this problem has a polynomial time algorithm. On the other hand, it is unlikely that the problem is NP-complete. In this talk, we will discuss some other computational problems in group theory whose complexity status is still unresolved.

References:

1. V. Arvind, and J. Torán: "The Complexity of Quasigroup Isomorphism and the Minimum Generating Set Problem", ISAAC 2006: 233-242.

2. J. A. Grochow, and Y. Qiao, "Algorithms for group isomorphism via group extensions and cohomology", SIAM J. Comput., vol. 46, pp. 1153-1216, 2017.

3. E. M. Luks: "Permutation groups and polynomial-time computation", in Groups and Computation, DIMACS series in Discrete Mathematics and Theoretical Computer Science 11 (1993), 139-175.

4. Á. Seress: "Permutation Group Algorithms", Cambridge Univ. Press, 2003.

Random Metrics in the Analysis of Algorithms

Bodo Manthey

University of Twente, Enschede, Netherlands, b.manthey@utwente.nl

1 Introduction

For large-scale optimization problems, finding optimal solutions within reasonable time is often impossible, because many such problems, like the traveling salesman problem (TSP), are NP-hard. Nevertheless, we often observe that simple heuristics succeed surprisingly quickly in finding close-to-optimal solutions. Many such heuristics perform well in practice but have a poor worst-case performance. In order to explain the performance of such heuristics, probabilistic analysis has proved to be a useful alternative to worst-case analysis. Probabilistic analysis of optimization problems has been conducted with respect to arbitrary instances (without the triangle inequality) or instances embedded in Euclidean space.

However, the average-case performance of heuristics for general metric instances is not well understood. This lack of understanding can be explained by two reasons: First, independent random edge lengths (without the triangle inequality) and random geometric instances are relatively easy to handle from a technical point of view – the former because of the independence of the lengths, the latter because Euclidean space provides a structure that can be exploited. Second, analyzing heuristics on random metric spaces requires an understanding of random metric spaces in the first place.

In order to initiate systematic research of heuristics on general metric spaces, we use the following model, proposed by Karp and Steele [9, Section 3.4]: given an undirected complete graph, we draw edge weights independently at random according to exponential distributions with parameter 1. The distance between any two vertices is the total weight of the shortest path between them, measured with respect to the random weights. We call such instances *random shortest path metrics*.

This model is also known as *first-passage percolation*, and has been introduced by Broadbent and Hammersley as a model for passage of fluid in a porous medium [2]. The appealing feature of random shortest path metrics is their simplicity, which enables us to use them for the analysis of heuristics. Dyer and Frieze [5], answering a question raised by Karp and Steele [9], analyzed the patching heuristic for the asymmetric TSP (ATSP) in this model. They showed that it comes within a factor of $1 + o(1)$ of the optimal solution with high probability. Hassin and Zemel [7] applied their findings to the 1-center problem.

2 Model and Structural Properties

We consider undirected complete graphs $G = (V, E)$ without loops. First, we draw *edge weights* $w(e)$ independently at random according to the exponential distribution with parameter 1.

Second, let the *distances* $d : V \times V \to [0, \infty)$ be given as follows: the distance $d(u, v)$ between u and v is the minimum total weight of a path connecting u and v. In particular, we have $d(v, v) = 0$ for all $v \in V$, $d(u, v) = d(v, u)$ because G is undirected, and the triangle inequality: $d(u, v) \leq d(u, x) + d(x, v)$ for all $u, x, v \in V$. We call the complete graph with distances d obtained from random weights w a *random shortest path metric*.

We use the following notation: Let $\Delta_{\max} = \max_{u,v} d(u, v)$ denote the *diameter* of the random shortest path metric. Let $B_\Delta(v) = \{u \in V \mid d(u, v) \leq \Delta\}$ be the ball of radius Δ around v, i.e., the set of all nodes whose distance to v is at most Δ.

We denote the minimal Δ such that there are at least k nodes within a distance of Δ of v by $\tau_k(v)$. Formally, we define $\tau_k(v) = \min\{\Delta \mid |B_\Delta(v)| \geq k\}$.

By $\mathrm{Exp}(\lambda)$, we denote the exponential distribution with parameter λ. If a random variable X is distributed according to a probability distribution P, we write $X \sim P$. In particular, $X \sim \sum_{i=1}^m \mathrm{Exp}(\lambda_i)$ means that X is the sum of m independent exponentially distributed random variables with parameters $\lambda_1, \ldots, \lambda_m$.

For $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$ and let $H_n = \sum_{i=1}^n 1/i$ be the n-th harmonic number.

There has been significant study of random shortest path metrics or first-passage percolation. The expected length of an edge is known to be $\ln n/n$ [4, 8]. Asymptotically the same bound holds also for the longest edge almost surely [7, 8].

To understand random shortest path metrics, it is convenient to fix a starting vertex v and see how the lengths from v to the other vertices develop. In this way, we analyze the distribution of $\tau_k(v)$.

The values $\tau_k(v)$ are generated by a simple birth process as follows. (The same process has been analyzed by Davis and Prieditis [4], Janson [8], and also in subsequent papers.) We have $\tau_1(v) = 0$.

For $k \geq 1$, we are looking for the closest vertex to any vertex in $B_{\tau_k(v)}(v)$ in order to obtain $\tau_{k+1}(v)$. This conditions all edges (u, x) with $u \in B_{\tau_k(v)}(v)$ and $x \notin B_{\tau_k(v)}(v)$ to be of length at least $\tau_k(v) - d(v, u)$. By the memoryless property of exponential distribution, this additional length is also exponentially distributed. The set $B_{\tau_k(v)}(v)$ contains k vertices. Thus, there are $k \cdot (n - k)$ edges to the rest of the graph. Consequently, the difference $\delta_k = \tau_{k+1}(v) - \tau_k(v)$ is distributed as the minimum of $k(n - k)$ exponential random variables (with parameter 1), or, equivalently, as $\mathrm{Exp}(k \cdot (n - k))$. We obtain that $\tau_{k+1}(v) \sim \sum_{i=1}^k \mathrm{Exp}\big(i \cdot (n - i)\big)$. Note that these exponential distributions as well as the random variables $\delta_1, \ldots, \delta_n$ are independent.

Lemma 1. *For any $k \in [n]$ and any $v \in V$, we have $\mathbb{E}\big(\tau_k(v)\big) = \frac{1}{n} \cdot \big(H_{k-1} + H_{n-1} - H_{n-k}\big)$ and $\tau_k(v) \sim \sum_{i=1}^{k-1} \mathrm{Exp}\big(i \cdot (n - i)\big)$.*

Lemma 2 (Janson [8, p. 352])**.** *For $c > 3$, we have $\mathbb{P}(\Delta_{\max} > c \ln(n)/n) \leq O(n^{3-c} \log^2 n)$.*

A key structural insight for the analysis of algorithms is that nodes can be clustered in a few clusters of relatively small diameter according to the following lemma.

Lemma 3 (Bringmann et al. [1])**.** *Consider a random shortest path metric and let $\Delta \geq 0$. If we partition the instance into clusters, each of diameter at most 6Δ, then the expected number of clusters needed is $O(1 + n/\exp(\Delta n/5))$.*

3 Algorithmic Results

Construction Heuristics for the TSP. Insertion heuristics and the nearest neighbor heuristic are prominent heuristics for construct a reasonably short tour for the TSP. Insertion heuristics achieve an approximation ratio between constant and $O(\log n)$, depending on the insertion rule used [13]. The nearest neighbor is known to achieve an approximation ratio of $O(\log n)$. Both heuristics can be shown to achieve constant approximation ratio on RSP instances using the aforementioned clustering argument (Lemma 3) [1].

2-Opt for the TSP and Sparse Graphs. The 2-opt heuristic is a popular local search heuristic for the TSP. In the worst case on metric instances, it is $O(\sqrt{n})$ [3]. For independent, non-metric edge lengths drawn uniformly from the interval $[0, 1]$, the expected approximation ratio is $O(\sqrt{n} \cdot \log^{3/2} n)$ [6].

We easily get an approximation ratio of $O(\log n)$ based on the two (almost trivial) facts that the length of the optimal tour is $\Theta(1)$ with high probability and that $\Delta_{\max} = O(\log n/n)$ with high probability. (An open problem is to improve this.)

However, we can also consider non-complete graphs to generate random metrics. The process is exactly the same: draw random edge weights and take shortest paths. Given that the graph is connected, the result will be a complete graph with edge lengths that for a metric. If we have sparse graphs to generate the random instances, then 2-opt can be shown to achieve an approximation ratio of $O(1)$ [11].

Facility Location. The trivial algorithm of opening the k cheapest facilities can be shown to do surprisingly well for the the facility location problem. The approximation ratio ranges from $1 + o(1)$ over $O(1)$ to $O(\sqrt[4]{\ln n})$, depending on the costs of the facility [1, 10].

4 Open Problems

To conclude the paper, let us list the open problems that we consider most interesting:

1. While the distribution of distances in asymmetric instances does not differ much from the symmetric case, an obstacle in the application of asymmetric random shortest path metrics seems to be the lack of clusters of small diameter. Is there an asymmetric counterpart for this?

2. Is it possible to prove an $1 + o(1)$ approximation ratio (like Dyer and Frieze [5] for the patching algorithm) for any of the simple construction heuristics for the TSP?

3. What is the approximation ratio of 2-opt in random shortest path metrics? Can we prove that the expected ratio of 2-opt is $o(\log n)$?

4. There is some cautious work on generalizing RSP to the case of a non-complete underlying graph [11, 12]. Are there more general results possible, in particular if the underlying graph is sparse?

References

[1] Karl Bringmann, Christian Engels, Bodo Manthey, and B. V. Raghavendra Rao. Random shortest paths: Non-euclidean instances for metric optimization problems. *Algorithmica*, 73(1):42–62, 2015.

[2] S. R. Broadbent and J.M. Hammersley. Percolation processes. I. Crystals and mazes. *Proceedings of the Cambridge Philosophical Society*, 53(3):629–641, 1957.

[3] Barun Chandra, Howard J. Karloff, and Craig A. Tovey. New results on the old k-opt algorithm for the traveling salesman problem. *SIAM Journal on Computing*, 28(6):1998–2029, 1999.

[4] Robert Davis and Armand Prieditis. The expected length of a shortest path. *Information Processing Letters*, 46(3):135–141, 1993.

[5] Martin E. Dyer and Alan M. Frieze. On patching algorithms for random asymmetric travelling salesman problems. *Mathematical Programming*, 46:361–378, 1990.

[6] Christian Engels and Bodo Manthey. Average-case approximation ratio of the 2-opt algorithm for the TSP. *Operations Research Letters*, 37(2):83–84, 2009.

[7] Refael Hassin and Eitan Zemel. On shortest paths in graphs with random weights. *Mathematics of Operations Research*, 10(4):557–564, 1985.

[8] Svante Janson. One, two, three times $\log n/n$ for paths in a complete graph with edge weights. *Combinatorics, Probability and Computing*, 8(4):347–361, 1999.

[9] Richard M. Karp and J. Michael Steele. Probabilistic analysis of heuristics. In Eugene L. Lawler, Jan Karel Lenstra, Alexander H. G. Rinnooy Kan, and David B. Shmoys, editors, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, pages 181–205. Wiley, 1985.

[10] Stefan Klootwijk and Bodo Manthey. Probabilistic analysis of facility location on random shortest path metrics. In Florin Manea, Barnaby Martin, Daniël Paulusma, and Giuseppe Primiero, editors, *Computing with Foresight and Industry - 15th Conference on Computability in Europe, CiE 2019, Durham, UK, July 15-19, 2019, Proceedings*, volume 11558 of *Lecture Notes in Computer Science*, pages 37–49. Springer, 2019.

[11] Stefan Klootwijk and Bodo Manthey. Probabilistic analysis of optimization problems on sparse random shortest path metrics. In Michael Drmota and Clemens Heuberger, editors, *31st International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms, AofA 2020, June 15-19, 2020, Klagenfurt, Austria (Virtual Conference)*, volume 159 of *LIPIcs*, pages 19:1–19:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[12] Stefan Klootwijk, Bodo Manthey, and Sander K. Visser. Probabilistic analysis of optimization problems on generalized random shortest path metrics. *Theoretical Computer Science*, 866:107–122, 2021.

[13] Daniel J. Rosenkrantz, Richard E. Stearns, and Philip M. Lewis II. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563–581, 1977.

NOTES