Theoretical Computer Science Lab

Dr. L. Sunil Chandran, Dr. Sathish Govindrajan

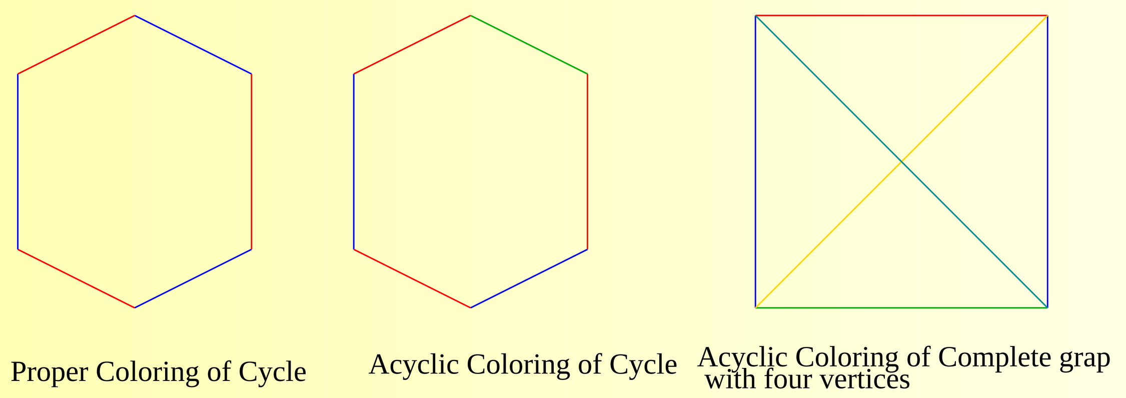
Areas of Interest

Theoretical Computer Science Lab works mainly in the areas of Graph Theory, combinatorics and Geometry-

- Graph Theoretic Problems
- Algorithms on Graphs
- Combinatorial Geometry
- Computational Geometry
- Randomized and Approximation Algorithms

Acyclic edge coloring

- Proper edge coloring of a graph is the assignment of colors to the edges such that adjacent edges get different colors.
- A proper edge-coloring with the property that every cycle contains edges of at least three distinct colors is called an "acyclic edge-coloring".
- We concentrate on graph theoretic concepts like obtaining lower and upper bounds for the acyclic edge chromatic number.



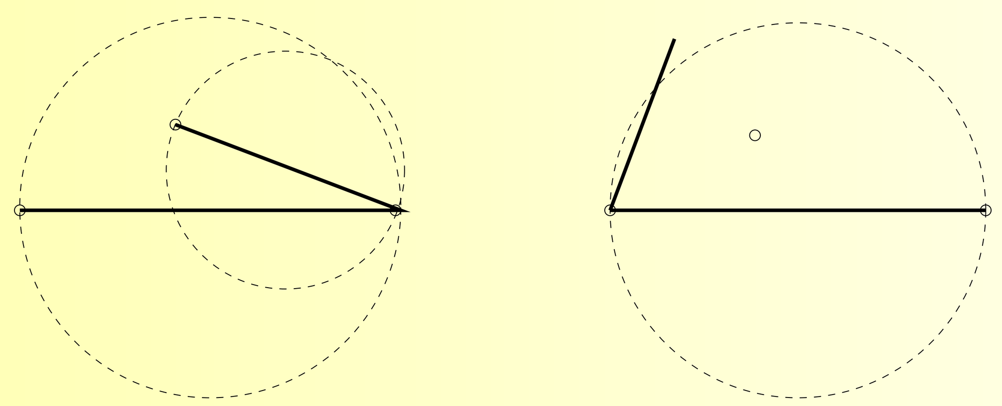
- We are developing the concepts needed by looking at special classes of graphs and obtaining bounds on them.
- Most of our proofs are constructive yielding efficient polynomial time algorithms for the acyclic edge coloring of the graphs under consideration.

Food for Thought

- Prove that to acyclically edge color a d-regular graph at least $d+1$ colors are required.
- Can you say anything about the lower bound of acyclic edge chromatic number for Complete graphs? How about upper bound (This problem is open)?

Geometric Graphs

- A geometric graph is drawn on Euclidean plane with points as vertices and line segments joining them serving as edges.
- Edges are defined based on some geometric constraints. Depending on these constraints we get various classses of geometric graphs.
- Two broad research problems on these graphs:
  - Proving combinatorial bounds.
  - Algorithms to compute optimum graphs.
- Currently we are looking at Locally Delaunay Graphs and Locally Gabriel graphs.

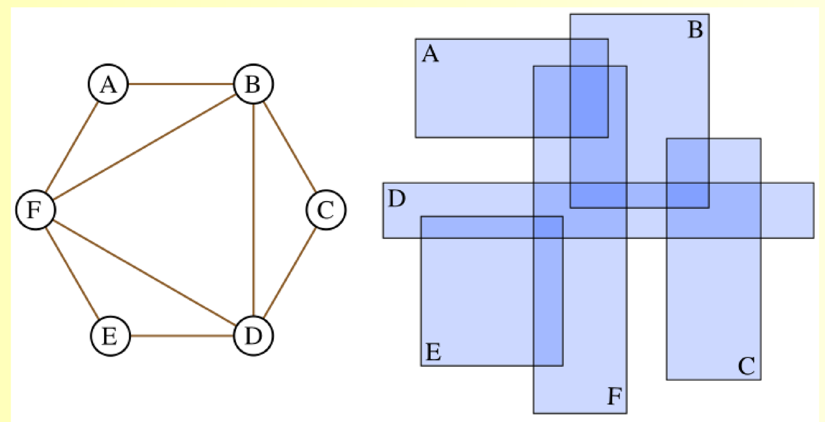


Invalid and valid Locally Gabriel Graphs

- Combinatorial bounds on LDG/LGG:
  - $n^{\frac{5}{4}} \leq |LGG| \leq n^{\frac{3}{2}}$
  - $n^{\frac{4}{3}} \leq |LDG| \leq n^{\frac{3}{2}}$
- No polynomial time algorithm to compute the optimum LDG/LGG is known. It is conjectured that the problem is hard, although no hardness proof is known yet.
- We are looking at developing exact/approximation algorithms for computing optimum LGG on special point sets like convex point set.

Some Special Intersection Graphs

- A k-dimensional box is a Cartesian product of closed intervals $[a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_k, b_k]$.
- A graph is said to have a k-box representation if each vertex can be associated with a k-dimensional box in the Euclidean space such that two vertices are adjacent if and only if their corresponding k-dimensional boxes have a non-empty intersection.
- The minimum k for which G has such a k-box representation is called the boxicity of k. Similarly we can replace k-dimensional boxes with k-dimensional unit cubes, i.e. Cartesian product of k unit intervals and the resulting parameter is referred to as *cubicity* of a graph.



We have developed/proved the following.

- A randomized algorithm to construct a box representation for any graph G on n vertices in $O(\Delta \log n)$ dimensions.
- The boxicity of a graph is at most $2\Delta^2$.
- Almost all graphs have $\Theta(n)$ boxicity. There exists no deterministic polynomial time algorithm to approximate the boxicity of a graph on n elements within a $O(\sqrt{n})$ factor.
- Bounds on the cubicity of interval graphs.

Empty convex k-gon

- Consider a two player game where each player places a point in general position in his turn. The objective of the game is to avoid the formation of an empty convex k-gon among the set of points placed by the players.
- We assume that each player optimally places the point in his turn.
- We focus on finding the exact value/bounds on the number of steps before the game ends for $k \geq 5$.

Bipartite matching problem

Consider the problem of matching applicants to a set of jobs with preferences. A popular matching satisfies maximum number of applicants. We look at instances not admitting popular matching.

	j_1	j_2	j_3	j_4	j_5
a_1	1	×	2	3	×
a_2	3	1	2	4	5
a_3	2	1	×	×	3

M_1

	j_1	j_2	j_3	j_4	j_5
a_1	1	×	2	3	×
a_2	3	1	2	4	5
a_3	2	1	×	×	3

M_2

M1 is better matching than M2

- Computing a least unpopular matching is NP-Hard. We give a simple polynomial time algorithm to compute a matching with bounded unpopularity.
- We consider computing a probability distribution over matchings which is Popular. We show that such a mixed popular matching always exists and can be computed efficiently using a linear programming formulation.