Color Coding: Variations and applications

Noga Alon, Tel Aviv U.

Based on joint papers with **Gutner, Lokshtanov and Saurabh**

AofA, 2010

Color Coding

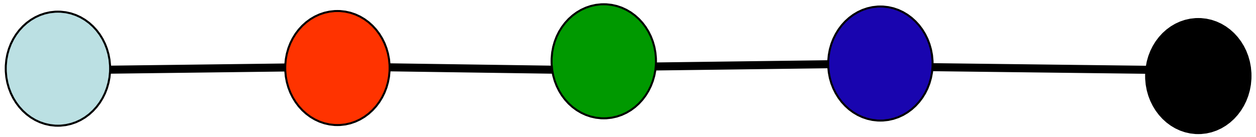
Question [Papadimitriou+ Yannakakis (93)]:
Is it possible to decide, in polynomial time, if a graph G=(V,E) with |V|=n vertices contains a path of $k=[\log_2 n]$ vertices ?

Answer [A+ Yuster+Zwick (95)]:
Yes !

A randomized algorithm:

-Color the vertices randmly by k colors

-Find efficiently if there is a multicolored k-path using Dynamic Programming



The probability that a k-path is multicolored is

$$k!/k\hat{\ }k=e^{-(1+o(1))k}$$

Dynamic Programming: check for every vertex v and every subset T of the set of k colors, if there is a multicolored path of |T| vertices ending at v, using the colors in T. Total time: $O(k\ 2^k\ |E|)$.

Expected running time: $(2e)^{(1+o(1))k}$ $|E|$

Koutis and Williams (09): can be improved to $2^{(1+o(1))k}$ $|E|$

Derandomization:

Use explicit schemes of Perfect Hashing to construct a family of $2^{O(k)}$ log n k-colorings of V, so that every set of size k is multicolored in at least one of them

[Fredman, Komlos and Szemerédi (84),
Schmidt and Siegel (90),
Naor, Schulman and Srinivasan(95)]:
There are such explicit families of size

$$e^{(1+o(1))k} \log n$$

The method extends for finding copies of any graph with a bounded tree width.

This is useful in Computational Biology, in the study of Protein Interaction Networks.

Counting the number of paths (or other graphs of bounded tree-width) on k vertices can be more useful.

Dynamic Programming can work here too, but for the implementation we need Balanced Families of Hash Functions.

Def: A family of functions from [n] to [k] is a (perfectly) balanced family of hash functions if there is a number T>0 so that for every subset K of [n], |K|=k, the number of functions f in the family so that f(K)=[k] is exactly T.

The bad news: such families must be large.

Thm: If F is a perfectly balanced family of functions From [n] to [k], then |F| ≥ c(k) $n^{[k/2]}$.

Proof : Given such a family F, define, for each subset R of size k/2 of [n], two vectors u_R,w_R of length $\begin{matrix}k\\k{=}2\end{matrix}$ jFj each, indexed by the ordered pairs (f ; S) with f 2 F; S ½ [k]; jSj = k=2 as follows:

u_R(f ; S) = 1 if f (R) = S (0 otherwise).

w_R(f ; S) = 1 if f (R) = [k] ¡ S (0 otherwise).

The inner product of two vectors u_R and w_Q is 0, if the two sets R and Q intersect, and is precisely T if R and Q are disjoint.

Therefore, the product of the matrix whose rows are the vectors u_R and the matrix whose columns are the vectors w_Q is the Disjointness Matrix of subsets of size k/2 in [n].

This matrix has full rank, and thus

$$\binom{k}{k/2} |F| \le \binom{n}{k/2}; \text{ implying } |F| \le c(k) n^{k/2}.$$

□

Vassilevska and Williams (09)
Bjorklund, Husfeldt, Kaski and Koivisto (09):

The precise number of k-paths in an n-vertex graph can be computed in time $c(k)n^{k/2+O(1)}$

Parameterized Complexity [Downey and Fellows (99)]:

A problem with a parameter is Fixed Parameter Tractable if there is an algorithm that solves a problem of size n with parameter k in time at most $f(k)n^{O(1)}$.

Example: deciding if a graph of size n contains a path of length k is fixed parameter tractable, deciding if it contains a clique of size k is (probably) not.

Flum and Grohe (04): The problem of counting the number of paths of length k in a graph of size n is # W[1] – complete.

It is therefore not surprising that there are no small families of perfectly balanced hash functions.

What about approximate counting ? Suppose we only want to approximate the number of paths up to a relative error of 1% ?

There is a simple randomized algorithm that does that (e.g., [A, Dao, Hajirasouliha, Hormozdiari + Sahinalp (08)])

Can we do it deterministically ?

Def: A family of functions from [n] to [k] is an **ε-balanced family of hash functions** if there is a number T>0 so that for every subset K of [n], |K|=k, the number of functions f in the family so that f(K)=[k] is at least (1-ε)T and at most (1+ε)T.

Fact: There are such families of size $e^{(1+o(1))k}$ log n.

Can we construct such a family explicitly ?

Thm [A+ Gutner (09)]: There is an explicit construction of an ε-balanced family of functions from [n] to [k] consisting of $e^{(1+o(1))k}$ log n functions. Such a family can be constructed in time $e^{(1+o(1))k}$ n log n.

The construction combines:

- **Small sample spaces supporting nearly pairwise independent random variables**
- **A recursive construction based on properties of expanders**
- **The method of conditional expectations**

Therefore, it is possible to approximate the number of k-paths in a given input graph G=(V,E) up to a relative error of 1/poly(k) in time $e^{(1+o(1))k}$ |E| log |V|.

A similar result holds for approximating the number of copies of other subgraphs of size k with bounded tree-width.

Chromatic Coding and Universal Coloring Families

[A, Lokshtanov+ Saurabh (09)]:

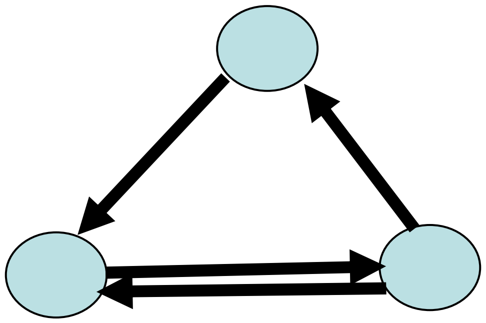
Def: A family of functions from [m] to [r] is a universal (m,k,r)-coloring family if for any graph G on the set of vertices [m] with at most k edges, there is a function f in the family that is a proper vertex coloring of G.

Note that each such family is a perfect hash family for sets of size $\sim \sqrt{k}$

Thm: There is an explicit family of $2^{\tilde{O}(\sqrt{k})} \log n$ functions from $[n]$ to $[\tilde{O}(\sqrt{k})]$ which is a universal $(n;k;\tilde{O}(\sqrt{k}))$ coloring family.

This is useful in tackling several algorithmic problems, including the Feedback Arc Set problem for Tournaments.

Def: A **tournament** T is an oriented complete graph



Def: A **feedback arc set** in T is a set of arcs whose reversal makes T acyclic.

The feedback arc set problem for tournaments:
Given T and k, does T have a feedback arc set of size at most k ?

A (06), following Ailon, Charikar and Newman (05): This problem is NP-hard [also shown by Charbit,Thomasse and Yeo (07)]

What about the parameterized version ?

Raman and Saurabh (06): It can be solved in time $O(2.415^{k} k^{4.752} + n^{O(1)})$.

Faster: (A,Lokshtanov, Saurabh): it can be solved in time

$$2^{O(\sqrt{k})} + n^{O(1)}:$$

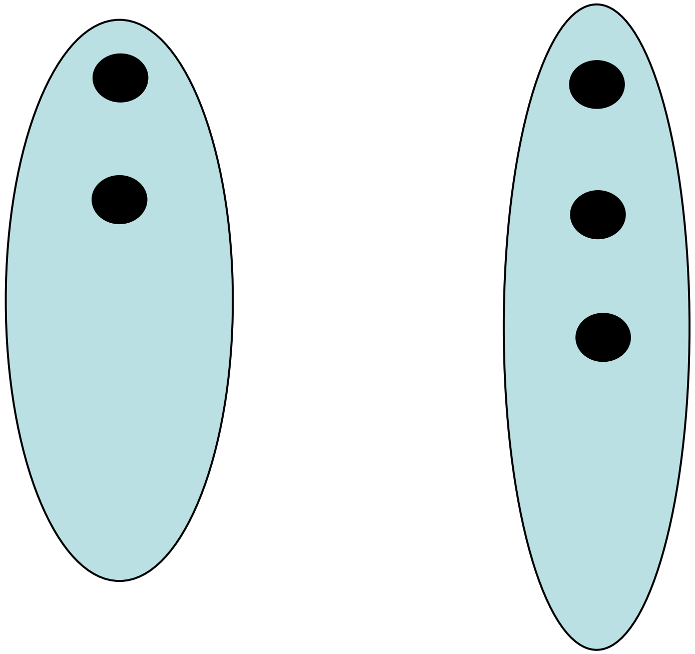
This settles a question of Guo, Moser and Niedermeier

Even Faster: [Feige (10), Karpinski and Schudy(10)]:
It can be solved in time:

$$2^{O(\sqrt{k})} + n^{O(1)}:$$

Fomin, Lokshtanov, Raman and Saurabh (10):
Local search algorithm for feedback arc set in tournaments

The key idea: if the graph consisting of all arcs in an optimal feedback arc set is properly t-colored, this optimal set can be found efficiently using Dynamic Programming.



Prop: A random coloring of a graph with k edges by $O(\sqrt{k})$ colors is a proper coloring with probability at least $2^{-O(\sqrt{k})}$

This (+ kernelization) gives a randomized

$$O(2^{\Theta(\sqrt{k})} + n^{O(1)})$$

algorithm.

Universal coloring families can serve to derandomize it.

The idea can be extended to universal coloring families for hypergraphs, which are useful in tackling additional algorithmic problems.

Conclusion:

Balanced Hashing and **Chromatic Coding** are useful techniques in Parameterized Complexity.

It seems interesting to further explore their possible applications.