Applications of Color Coding in Randomised Algorithms (for CS 602-Applied Algorithms)

Presentation · March 2019

Meet Taraviya
Indian Institute of Technology Bombay

Color Coding

CS 602 - Applied Algorithms

K. Mittal, M. Pareek and M. Taraviya

Department of Computer Science
Indian Institute of Technology, Bombay

March 22, 2019

Outline

1. Problem Definitions
2. Random Orientations
3. Random Colorings
4. Derandomization
5. Related results

Problem Definitions

k-PATH and k-CYCLE

Given a graph $G = (V, E)$ we are interested in the following problems (in both the directed and undirected case):

- Does the graph contain a path of length k?
- Does the graph contain a cycle of length k?

Problem Definitions

k-PATH and k-CYCLE

Given a graph $G = (V, E)$ we are interested in the following problems (in both the directed and undirected case):

- Does the graph contain a path of length k?
- Does the graph contain a cycle of length k?

We'll present ideas from a paper by Alon, Yuster and Zwick [AYZ95].

Problem Definitions

k-PATH and k-CYCLE

It is easy to see that both of these problems are NP-complete.

- (k-CYCLE) A cycle of length n is the same as a Hamiltonian cycle.
- (k-PATH) A path of length $n-1$ is the same as a Hamiltonian path.

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph. [a]

[a] $|E|, |V|$ are denoted by E, V for convenience.

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

Consider the **directed** case first.

- Choose random permutation π on V.

□

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

Consider the **directed** case first.

- Choose random permutation π on V.
- Consider only edges (u, v) where $\pi(u) < \pi(v)$.

$\square$

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

Consider the **directed** case first.

- Choose random permutation π on V.
- Consider only edges (u, v) where $\pi(u) < \pi(v)$.
- Check if longest path is of length $\geq k$ in this DAG.

□

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

Consider the **directed** case first.

- Choose random permutation π on V.
- Consider only edges (u, v) where $\pi(u) < \pi(v)$.
- Check if longest path is of length $\geq k$ in this DAG.

If there is a path of length k, it is found with probability $\geq \frac{1}{(k+1)!}$, giving the required expected runtime. $\square$

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

Now consider the **undirected** case.

- Choose random permutation π on V.

$\square$

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

Now consider the **undirected** case.

- Choose random permutation π on V.
- For all edges $\{u, v\}$, direct them from u to v where $\pi(u) < \pi(v)$.

$\square$

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

Now consider the **undirected** case.

- Choose random permutation π on V.
- For all edges $\{u, v\}$, direct them from u to v where $\pi(u) < \pi(v)$.
- Check if longest path is of length $\geq k$ in this DAG.

$\square$

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

Now consider the **undirected** case.

- Choose random permutation π on V.
- For all edges $\{u, v\}$, direct them from u to v where $\pi(u) < \pi(v)$.
- Check if longest path is of length $\geq k$ in this DAG.

If there is a path of length k, it is found with probability $\geq \frac{2}{(k+1)!}$. □

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

This gives expected runtime $O((k+1)! \cdot E)$. $\square$

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

This gives expected runtime $O((k+1)! \cdot E)$.
This can be made more efficient. □

k-PATH using Random Orientations

Theorem

A path of length k, if present, can be found in expected time $O((k+1)! \cdot E)$ in a directed graph and $O((k+2)! \cdot V)$ in an undirected graph.

Proof.

This gives expected runtime $O((k+1)! \cdot E)$.
This can be made more efficient.
First do a DFS from an arbitrary vertex. If no path of length k is found, then use the previous algorithm. In such a case, it must be true that $|E| \leq k|V|$. □

Other results

Deterministic Algorithms

By combining techniques of Monien [Mon85] and Bodlaender [Bod93], one can also get deterministic algorithms achieving runtime of $O(k! \cdot V)$ and $O(k! \cdot E)$ for undirected and directed graphs respectively.

k-CYCLE using Random Orientations

Theorem

A cycle of length, if present, k can be found in expected time $O((k-1)! \log k \cdot V^{\omega})$ in a directed or an undirected graph, where ω is the matrix multiplication exponent.

k-CYCLE using Random Orientations

Theorem

A cycle of length, if present, k can be found in expected time $O((k-1)!\log k \cdot V^{\omega})$ in a directed or an undirected graph, where ω is the matrix multiplication exponent.

Proof.

- Form the DAG G' as in the PATH problem with adjacency matrix A.

$\square$

k-CYCLE using Random Orientations

Theorem

A cycle of length, if present, k can be found in expected time $O((k-1)!\log k \cdot V^{\omega})$ in a directed or an undirected graph, where ω is the matrix multiplication exponent.

Proof.

- Form the DAG G' as in the PATH problem with adjacency matrix A.
- Consider A^{k-1}, which represents all paths of length $k-1$.

$\square$

k-CYCLE using Random Orientations

Theorem

A cycle of length, if present, k can be found in expected time $O((k-1)! \log k \cdot V^{\omega})$ in a directed or an undirected graph, where ω is the matrix multiplication exponent.

Proof.

- Form the DAG G' as in the PATH problem with adjacency matrix A.
- Consider A^{k-1}, which represents all paths of length $k-1$.
- For each such path, check if there is an edge in G between endpoints.

$\square$

k-CYCLE using Random Orientations

Theorem

A cycle of length, if present, k can be found in expected time $O((k-1)!\log k \cdot V^{\omega})$ in a directed or an undirected graph, where ω is the matrix multiplication exponent.

Proof.

- Form the DAG G' as in the PATH problem with adjacency matrix A.
- Consider A^{k-1}, which represents all paths of length $k-1$.
- For each such path, check if there is an edge in G between endpoints.

If there is a cycle of length k, it is found with probability $\geq \frac{1}{(k-1)!}$. $\square$

Random Colorings

Random colorings

- Choose a random coloring $c : V \to [k]$.

Random Colorings

Random colorings

- Choose a random coloring $c : V \to [k]$.
- A path is said to be **colorful** if each vertex on it is colored by a distinct color.

Random Colorings

Random colorings

- Choose a random coloring $c : V \to [k]$.
- A path is said to be **colorful** if each vertex on it is colored by a distinct color.
- Each simple path of length $k-1$ has a chance of $\frac{k!}{k^k} > e^{-k}$ to become colorful.

k-PATH using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, a colorful path of length $k-1$, if it exists, can be found in $2^{O(k)} \cdot E$ time.

k-PATH using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$*, a colorful path of length* $k-1$*, if it exists, can be found in* $2^{O(k)} \cdot E$ *time.*

Proof.

- Add a new vertex s and connect it to all vertices. Now we find a colorful path of length k starting at s.

$\square$

k-PATH using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, a colorful path of length $k-1$, if it exists, can be found in $2^{O(k)} \cdot E$ time.

Proof.

- Add a new vertex s and connect it to all vertices. Now we find a colorful path of length k starting at s.
- Use dynamic programming. A state consists of the following:

$\square$

k-PATH using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, a colorful path of length $k-1$, if it exists, can be found in $2^{O(k)} \cdot E$ time.

Proof.

- Add a new vertex s and connect it to all vertices. Now we find a colorful path of length k starting at s.
- Use dynamic programming. A state consists of the following:
  - A vertex v and a length i $(1 \leq i \leq k)$.
  - A set $S \subseteq \binom{[k]}{i}$ of all possible color sets on some colorful path of length i from s to v.

□

k-PATH using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, a colorful path of length $k-1$, if it exists, can be found in $2^{O(k)} \cdot E$ time.

Proof.

- Add a new vertex s and connect it to all vertices. Now we find a colorful path of length k starting at s.
- Use dynamic programming. A state consists of the following:
  - A vertex v and a length i $(1 \leq i \leq k)$.
  - A set $S \subseteq \binom{[k]}{i}$ of all possible color sets on some colorful path of length i from s to v.
- Perform updates by iterating over i, and over E within each iteration.

$\square$

k-PATH using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, a colorful path of length $k-1$, if it exists, can be found in $2^{O(k)} \cdot E$ time.

Proof.

- Add a new vertex s and connect it to all vertices. Now we find a colorful path of length k starting at s.
- Use dynamic programming. A state consists of the following:
  - A vertex v and a length i $(1 \leq i \leq k)$.
  - A set $S \subseteq \binom{[k]}{i}$ of all possible color sets on some colorful path of length i from s to v.
- Perform updates by iterating over i, and over E within each iteration.

The total number of states is $\left(\sum_{i=0}^{k} i \cdot \binom{k}{i}\right) \cdot V = 2^{O(k)} \cdot V$ and the runtime is $2^{O(k)} \cdot E$. □

k-CYCLE using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, all pairs of vertices connected by colorful paths of length $k-1$ can be found in $2^{O(k)} \cdot VE$ or $2^{O(k)} \cdot V^{\omega}$ time.

k-CYCLE using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, all pairs of vertices connected by colorful paths of length $k-1$ can be found in $2^{O(k)} \cdot VE$ or $2^{O(k)} \cdot V^{\omega}$ time.

Proof.

The $2^{O(k)} \cdot VE$ algorithm is obtained by simply running the previous algorithm $|V|$ times.

$\square$

k-CYCLE using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, all pairs of vertices connected by colorful paths of length $k-1$ can be found in $2^{O(k)} \cdot VE$ or $2^{O(k)} \cdot V^{\omega}$ time.

Proof.

- Consider a partition $\{C_1, C_2\}$ of $[k]$, each of size $k/2$.

$\square$

k-CYCLE using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, all pairs of vertices connected by colorful paths of length $k-1$ can be found in $2^{O(k)} \cdot VE$ or $2^{O(k)} \cdot V^{\omega}$ time.

Proof.

- Consider a partition $\{C_1, C_2\}$ of $[k]$, each of size $k/2$.
- Let G_1 be a graph having vertices with colors in C_1 and similarly G_2.

$\square$

k-CYCLE using Random Colorings

Lemma

Given a graph G and a coloring $c : V \rightarrow [k]$, all pairs of vertices connected by colorful paths of length $k-1$ can be found in $2^{O(k)} \cdot VE$ or $2^{O(k)} \cdot V^{\omega}$ time.

Proof.

- Consider a partition $\{C_1, C_2\}$ of $[k]$, each of size $k/2$.
- Let G_1 be a graph having vertices with colors in C_1 and similarly G_2.
- Recursively, find pairs of vertices connected by colorful paths of length $k/2-1$ in G_1 and G_2, as boolean matrices A_1 and A_2.

$\square$

k-CYCLE using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, all pairs of vertices connected by colorful paths of length $k-1$ can be found in $2^{O(k)} \cdot VE$ or $2^{O(k)} \cdot V^{\omega}$ time.

Proof.

- Consider a partition $\{C_1, C_2\}$ of $[k]$, each of size $k/2$.
- Let G_1 be a graph having vertices with colors in C_1 and similarly G_2.
- Recursively, find pairs of vertices connected by colorful paths of length $k/2 - 1$ in G_1 and G_2, as boolean matrices A_1 and A_2.
- Let B be the adjacency matrix of the edges in G from G_1 to G_2.

□

k-CYCLE using Random Colorings

Lemma

Given a graph G and a coloring $c : V \to [k]$, all pairs of vertices connected by colorful paths of length $k-1$ can be found in $2^{O(k)} \cdot VE$ or $2^{O(k)} \cdot V^{\omega}$ time.

Proof.

- Consider a partition $\{C_1, C_2\}$ of $[k]$, each of size $k/2$.
- Let G_1 be a graph having vertices with colors in C_1 and similarly G_2.
- Recursively, find pairs of vertices connected by colorful paths of length $k/2-1$ in G_1 and G_2, as boolean matrices A_1 and A_2.
- Let B be the adjacency matrix of the edges in G from G_1 to G_2.
- Take OR of the matrices $A_1 B A_2$ over all partitions $\{C_1, C_2\}$.

The number of matrix multiplications is given by
$N(k) = (2 \cdot N(k/2) + 2) \times \binom{k}{k/2} = 2^{O(k)}$. $\square$

Random Colorings

Theorem

In a graph G, a path of length $k-1$, if it exists, can be found in $2^{O(k)} \cdot V$ expected time in the undirected case and in $2^{O(k)} \cdot E$ expected time in the directed case.

Random Colorings

Theorem

In a graph G, a path of length $k-1$, if it exists, can be found in $2^{O(k)} \cdot V$ expected time in the undirected case and in $2^{O(k)} \cdot E$ expected time in the directed case.

Theorem

In a graph G, a cycle of length k, if it exists, can be found in expected time $2^{O(k)} \cdot VE$ or $2^{O(k)} \cdot V^{\omega}$.

Derandomizing Random Colorings

k-Perfect Family of Hash Functions

A family of hash functions from $[n] \to [k]$ is called k-perfect if for all $S \subseteq [n]$, $|S| = k$, there is a function in the family that is one to one on S.

Derandomizing Random Colorings

k-Perfect Family of Hash Functions

A family of hash functions from $[n] \to [k]$ is called k-perfect if for all $S \subseteq [n]$, $|S| = k$, there is a function in the family that is one to one on S.

Using ideas from Schmidt and Siegal [SS90] and Moni Naor, the existence of such families of size $2^{O(k)} log(n)$ can be shown.

Derandomizing Random Colorings

k-Perfect Family of Hash Functions

A family of hash functions from $[n] \rightarrow [k]$ is called k-perfect if for all $S \subseteq [n]$, $|S| = k$, there is a function in the family that is one to one on S.

Using ideas from Schmidt and Siegal [SS90] and Moni Naor, the existence of such families of size $2^{O(k)} log(n)$ can be shown.

Derandomization

We can use the above family to get colorings in which for each k subset V' of V, there is a coloring that assigns each vertex in V' a distinct color. This incurs an extra $log(|V|)$ multiplicative factor in the runtime.

Related results

- k-PATH problem is in P for $k \leq log(|V|)$ (this is the LOG PATH problem).
- The derandomization of color-coding method can be easily parallelized, yielding efficient NC algorithms.
- Cycles of length k for $k \leq 7$ can be found in time $O(V^{\omega})$. [AYZ97]

Counting the number of paths of length k

k-Perfect Family of Balanced Hash Functions

A family of hash functions from $[n] \to [k]$ is called k-perfect balanced if for some $T > 0$, we have that for all $S \subseteq [n]$, $|S| = k$, the number of functions in the family such that $f(S) = [k]$ is exactly T.

Counting the number of paths of length k

k-Perfect Family of Balanced Hash Functions

A family of hash functions from $[n] \to [k]$ is called k-perfect balanced if for some $T > 0$, we have that for all $S \subseteq [n]$, $|S| = k$, the number of functions in the family such that $f(S) = [k]$ is exactly T.

Do we have a small k-perfect balanced family of functions?

Counting the number of paths of length k

k-Perfect Family of Balanced Hash Functions

A family of hash functions from $[n] \to [k]$ is called k-perfect balanced if for some $T > 0$, we have that for all $S \subseteq [n]$, $|S| = k$, the number of functions in the family such that $f(S) = [k]$ is exactly T.

Do we have a small k-perfect balanced family of functions?
No (Proof later).

Counting the number of paths of length k

k-Perfect Family of Balanced Hash Functions

A family of hash functions from $[n] \rightarrow [k]$ is called k-perfect balanced if for some $T > 0$, we have that for all $S \subseteq [n]$, $|S| = k$, the number of functions in the family such that $f(S) = [k]$ is exactly T.

Do we have a small k-perfect balanced family of functions?
No (Proof later).
But we can **approximate!**
Allow the number to be between $(1 - \epsilon)T$ and $(1 + \epsilon)T$.

Counting the number of paths of length k

k-Perfect Family of Balanced Hash Functions

A family of hash functions from $[n] \to [k]$ is called k-perfect balanced if for some $T > 0$, we have that for all $S \subseteq [n]$, $|S| = k$, the number of functions in the family such that $f(S) = [k]$ is exactly T.

Do we have a small k-perfect balanced family of functions?
No (Proof later).
But we can **approximate!**
Allow the number to be between $(1-\epsilon)T$ and $(1+\epsilon)T$.

Theorem

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

Size of k-perfect balanced family of functions

Theorem

If F is a k-perfect balanced family of functions $\{f : [n] \to [k]\}$, $|F| \geq c(k)n^{k/2}$.

Size of k-perfect balanced family of functions

Theorem

If F is a k-perfect balanced family of functions $\{f : [n] \to [k]\}$, $|F| \geq c(k)n^{k/2}$.

Proof.

- For each $R \subseteq [n]$, $|R| = k/2$ consider vectors u_R and w_R of length $\binom{k}{k/2} \cdot |F|$, where
  - $u_R(f, S) = 1$ if $f(R) = S$ (0 otherwise)
  - $w_R(f, S) = 1$ if $f(R) = [k] \setminus S$ (0 otherwise)

□

Size of k-perfect balanced family of functions

Theorem

If F is a k-perfect balanced family of functions $\{f : [n] \to [k]\}$, $|F| \geq c(k)n^{k/2}$.

Proof.

- For each $R \subseteq [n]$, $|R| = k/2$ consider vectors u_R and w_R of length $\binom{k}{k/2} \cdot |F|$, where
  - $u_R(f, S) = 1$ if $f(R) = S$ (0 otherwise)
  - $w_R(f, S) = 1$ if $f(R) = [k] \setminus S$ (0 otherwise)
- $\langle u_R, w_Q \rangle = 0$ if $R \cap Q \neq \phi$ and $\langle u_R, w_Q \rangle = T$ otherwise.

$\square$

Size of k-perfect balanced family of functions

Theorem

If F is a k-perfect balanced family of functions $\{f : [n] \to [k]\}$, $|F| \geq c(k)n^{k/2}$.

Proof.

- For each $R \subseteq [n]$, $|R| = k/2$ consider vectors u_R and w_R of length $\binom{k}{k/2} \cdot |F|$, where
  - $u_R(f, S) = 1$ if $f(R) = S$ (0 otherwise)
  - $w_R(f, S) = 1$ if $f(R) = [k] \setminus S$ (0 otherwise)
- $\langle u_R, w_Q \rangle = 0$ if $R \cap Q \neq \phi$ and $\langle u_R, w_Q \rangle = T$ otherwise.
- Let M_u be matrix with u_R as rows.
- Let M_w be matrix with w_Q as columns.

□

Size of k-perfect balanced family of functions

Theorem

If F is a k-perfect balanced family of functions $\{f : [n] \to [k]\}$, $|F| \geq c(k)n^{k/2}$.

Proof.

- For each $R \subseteq [n]$, $|R| = k/2$ consider vectors u_R and w_R of length $\binom{k}{k/2} \cdot |F|$, where
  - $u_R(f, S) = 1$ if $f(R) = S$ (0 otherwise)
  - $w_R(f, S) = 1$ if $f(R) = [k] \setminus S$ (0 otherwise)
- $\langle u_R, w_Q \rangle = 0$ if $R \cap Q \neq \phi$ and $\langle u_R, w_Q \rangle = T$ otherwise.
- Let M_u be matrix with u_R as rows.
- Let M_w be matrix with w_Q as columns.
- $M_u M_w$ has full rank $\implies \binom{k}{k/2} \cdot |F| \geq \binom{n}{k/2} \implies |F| \geq c(k)n^{k/2}$.

□

Thank You!

References I

Noga Alon, Raphael Yuster, and Uri Zwick.
Color-coding.
J. ACM, 42(4):844–856, 1995.

Noga Alon, Raphael Yuster, and Uri Zwick.
Finding and counting given length cycles.
Algorithmica, 17(3):209–223, 1997.

Hans L. Bodlaender.
On linear time minor tests with depth-first search.
J. Algorithms, 14(1):1–23, 1993.

B. Monien.
How to find long paths efficiently.
In G. Ausiello and M. Lucertini, editors, *Analysis and Design of Algorithms for Combinatorial Problems*, volume 109 of *North-Holland Mathematics Studies*, pages 239 – 254. North-Holland, 1985.

References II

Jeanette P. Schmidt and Alan Siegel.
The spatial complexity of oblivious k-probe hash functions.
SIAM J. Comput., 19(5):775–786, 1990.