Algoritmos Parametrizados

Métodos Aleatorizados

Lehilton Pedrosa

Segundo Semestre de 2016

Instituto de Computação – Unicamp

Roteiro

1. Métodos Aleatorizados
2. Conjunto de retroalimentação de Vértices
3. Codificação de cores
4. Separação aleatória
5. Codificação cromática
6. Desaleatorização

Métodos Aleatorizados

Algoritmo Aleatorizado

Modelo probabilístico

- $P \subseteq \Sigma^*$ um problema
- $r \geq 0$ o número de bits aleatórios

Seja um algoritmo Alg com parâmetros:

- $I \in \Sigma^*$
- $\omega \in \{0, 1\}^r$

Dizemos que Alg responde R para I com probabilidade:

$$p := \frac{|\{\omega : Alg(I, \omega) = R\}|}{2^r}$$

Algoritmo Monte Carlo

Definição

Um algoritmo Monte Carlo com falsos negativos e probabilidade p é um algoritmo aleatorizado tal que:

1. Tem tempo de execução limitado
2. Tem as seguintes probabilidades:

Entrada	**Prob. da resposta correta**
Instância não: $I \in P$	1
Instância sim: $I \notin P$	p

Amplificação de probabilidade

O algoritmo ALG′ amplifica a probabilidade de ALG:

ALG′(I, t)

1. Para $i = 1, \ldots, t$: execute ALG(I)
2. Se todas execução de ALG responder não, responda não.
3. Senão, responda sim.

O algoritmo ALG′ tem probabilidade $p' = e^{-pt}$.

Observação para FPT:

- Se $t = \lceil \frac{1}{p} \rceil$, então p' é constante
- Se ALG tem $p = 1/\mathcal{O}^*(f(k))$ e tempo $\mathcal{O}^*(g(k))$, então ALG' tem p' constante e tempo $\mathcal{O}^*(f(k)g(k))$.

Conjunto de retroalimentação de Vértices

Conjunto de retroalimentação de Vértices

Lema

Se G é multigrafo com $d(v) \geq 3$ para $v \in V(G)$ e X é conjunto de retroalimentação de vértices, então mais da metade das arestas incidem em X.

FPT aleatorizado

Teorema

Dados G e k, existe algoritmo polinomial que

1. *ou encontra um conjunto de retroalimentação com tamanho k;*
2. *ou reponde não.*

Se G tiver conjunto de retroalimentação de tamanho k, então a probabilidade de sucesso é pelo menos 4^{-k}.

Corolário

Existe um algoritmo aleatorizado que, em tempo $4^k n^{\mathcal{O}(1)}$ encontra um conjunto de retroalimentação com tamanho k de G, ou reporta falha. Se a instância for sim, então devolve um conjunto com probabilidade constante.

Codificação de cores

Isomorfismo de subgrafo

Isomorfismo de subgrafo

Dados grafos G e H, H é subgrafo de G?

Problema do Caminho mais longo: caso particular quando H é um caminho P_k, com k vértices.

Codificação de Cores

Ideia

- damos uma coloração própria para o caminho H
- adivinhamos as cores no grafo G

Flexibilidade da aleatorização:

- como se testássemos todas possibilidades
- mas com tempo de execução para somente uma

Dificuldades:

- número de possibilidades certas é “grande”

Contando o número de possibilidades

Se k for pequeno comparado a n, então precisamos acertar as cores de poucos vértices!

Lema

Seja U um conjunto de n elementos e $X \subseteq U$ com $|X| = k$. Seja $\chi : U \to [k]$ uma coloração de U, escolhida aleatória e uniformemente.

Então a probabilidade de que χ induz uma coloração própria de X é pelo menos e^{-k}.

Encontrando um caminho colorido

Nos concentramos no seguinte problema:

- Dados G e uma coloração de $V(G)$ com k cores;
- Queremos um caminho de G com as k cores

Ideia

- Sem cores:
  - guardamos que vértices foram visitados: tempo $\binom{n}{k}$
- Com cores:
  - precisamos guardar que cores foram visitadas

Lema

Seja $\chi : V(G) \to [k]$. Existe algoritmo que verifica em tempo $2^k n^{\mathcal{O}(1)}$ se G contém um caminho com k vértices e k cores; se sim, também devolve o caminho.

Algoritmo para Caminho mais longo

Teorema

Existe um algoritmo aleatorizado que, dada uma instância do Caminho mais longo (G, k)*, em tempo* $(2e)^k n^{\mathcal{O}(1)}$*, ou reporta falha, ou encontra um caminho de tamanho pelo menos* k*. Se a instância for sim, então ele encontra uma solução com probabilidade constante.*

Separação aleatória

Separação aleatória

Ideia

Colorimos arestas de forma que:

- arestas das solução sejam vermelhas
- arestas adjacentes seja azuis

→ componentes conexas vermelhas são “componentes” da solução

Isomorfismo de subgrafo

Seja (G, H) uma instância sim de Isomorfismo de subgrafo:

- i.e., existe $\hat{H} \subseteq G$ tal que $\hat{H} \cong H$

Coloração independente: pintamos cada aresta de G de vermelho ou azul, com probabilidade $1/2$ para cada cor.

Seja Γ o conjunto das arestas adjacentes de H.

Coloração bem-sucedida

1. H é vermelho
2. Γ é azul

Lema

A probabilidade de obter uma coloração independente bem-sucedida é pelo meno $1/2^{dk}$.

Algoritmo para isomorfismo

- acreditamos que isomorfismo não é **FPT**
- caso particular: $\Delta(G) \leq d$, para algum d fixo

Teorema

Existe um algoritmo que dados (G, H), com $\Delta(G) \leq d$, ou reporta falha ou decide que existe um subgrafo de G isomorfo a H que executa em tempo $2^{dk} k! n^{\mathcal{O}(1)}$.

Se a reposta for sim, então ele devolve o subgrafo com probabilidade constante.

Codificação cromática

Problema do d-agrupamento

Problema de edição de arestas:

- queremos adicionar ou remover arestas
- obter um novo grafo de um certa classe

Termos:

- $A \subseteq \binom{V(G)}{2}$ é um conjunto de adjacências
- um d-agrupamento é a união disjunta de d cliques

Problema do d-agrupamento de grafo

Dado um grafo G e um inteiro k, existe um conjunto de adjacências A, com $|A| \leq k$, tal que o grafo $(V(G), E(G) \triangle A)$ é um d-agrupamento?

Codificação cromática

- seja uma coloração de vértices $\chi : V \to [q]$
- $\binom{uv \in V(G)}{2}$ é colorida propriamente se $\chi(u) \neq \chi(v))$

Codificação cromática: solução A é colorida propriamente

Obtendo uma coloração própria

Lema

Seja $q = \lceil \sqrt{8k} \rceil$ e G um grafo com k arestas cujos vértices foram coloridos aleatoriamente com q cores.

Então a probabilidade de $E(G)$ ser colorida propriamente é pelo menos $2^{-\sqrt{k/2}}$.

Encontrando uma solução colorida

Lema

Seja G um grafo e $\chi : V(G) \to [q]$. Seja V_i os vértices com cor i.

Se existe uma solução A colorida propriamente por χ, então, para todo i, $G[V_i]$ é um l-agrupamento com $l \leq d$.

Algoritmo:

Combinando

Lema

Dada uma instância (G, k) *de* d*-agrupamento com coloração* $\chi : V(G) \to [\lceil\sqrt{8k}\rceil]$*, podemos decidir se há uma solução colorida propriamente em tempo* $2^{\mathcal{O}(d \log d \sqrt{k})}$.

Teorema

Existe um algoritmo aleatorizado que, dada uma instância (G, k) *do* d*-agrupamento, em tempo* $2^{\mathcal{O}(d\sqrt{k})} n^{\mathcal{O}(1)}$ *ou reporta falha, ou encontra uma solução. Além disso, dada uma instância sim, ele retorna uma solução com probabilidade constante.*

Desaleatorização

Desaleatorizando

Famílias pseudoaleatórias

Queremos colorações $\chi : [n] \to [k]$ boas

A invés de obter χ aleatório, retiramos de uma família $\mathcal{F}$:

- $\mathcal{F}$ contém pelo menos uma coloração boa
- $\mathcal{F}$ é pequena

Vamos usar algumas construções prontas: separadores, conjuntos universais, hashs perfeitos e famílias independentes

Separadores

Definição

Um (n, k, l)-splitter é uma família $\mathcal{F}$ de funções $f : [n] \to [l]$.

Para todo $S \subseteq [n]$, como $|S| = k$, existe $f \in \mathcal{F}$ tal que:

$$-1 \leq |f^{-1}(j) \cap S| - |f^{-1}(j') \cap S| \leq 1.$$

Splitters injetores: Se $l \geq k$, então f induz uma função injetora em S.

Ferramenta básica

Teorema

Para $n, k \geq 1$, existe algoritmo que:

- *obtém um (n, k, k^2)-splitter de tamanho $k^{\mathcal{O}(1)} \log n$*
- *tem tempo de execução $k^{\mathcal{O}(1)} n \log n$.*

Hash perfeito

Um (n, k, k)-splitter é chamado de (n, k)-hash perfeito.

Teorema

Para $n, k \geq 1$, existe algoritmo que:

- *obtém um (n, k)-hash perfeito de tamanho $e^k k^{\mathcal{O}(\log k)} \log n$*
- *tem tempo de execução $e^k k^{\mathcal{O}(\log k)} n \log n$.*

Conjunto Universal

Definição

Um (n, k)-universal set é uma família $\mathcal{U}$ de subconjuntos de $[n]$ tal que para cada $S \subseteq [n]$ de tamanho ki, a família $\{A \cap S : A \in \mathcal{U}\} = 2^{[k]}$.

Teorema

Para $n, k \geq 1$, existe algoritmo que:

- *obtém um (n, k)-universal set de tamanho $2^k k^{\mathcal{O}(\log k)} \log n$*
- *tem tempo de execução $2^k k^{\mathcal{O}(\log k)} n \log n$.*

Conjuntos independentes k a k

Definição

Uma família $H_{n,k,q}$ de funções $f : [n] \to [q]$ é chamada de espaço amostral independente k a k se, para todos conjuntos de k elementos de $[n]$, $i_1 \leq \cdots \leq i_k$ e todo $\alpha \in [q]^k$, valer

$$Pr((f(i_1), \ldots, f(i_k)) = \alpha) = 1/q^k,$$

onde $f \in H_{n,k,q}$ é escolhida uniformemente.

Teorema

Existe algoritmo que obtém $H_{n,k,q}$ de tamanho $\mathcal{O}(n^k)$ em tempo linear na saída.

Desaleatorizando Caminho mais longo

Ideia: ao invés de usar uma coloração aleatória, testamos todas colorações de um (n, k)-hash perfeito

Teorema

Caminho mais longo pode ser resolvido em tempo $(2e)^k k^{\mathcal{O}(\log k)} n^{\mathcal{O}(1)}$.

Desaleatorizando d-agrupamento

Definição

Uma (n, k, q)-coloring family é uma família $\mathcal{F}$ de funções de $[n]$ em $[q]$ com a seguinte propriedade:

- para todo grafo $G = (V, E)$ com $V = [n]$ e $|E| \leq k$, existe $f \in \mathcal{F}$ que colore E propriamente.

Objetivo: encontrar uma (n, k, q)-coloring family pequena

Ideia: Compor um (n, k, k^2)-splitter com uma (k^2, k, q)-coloring family

- cada aresta de G deve ser colorida propriamente
- basta que pares de vértices sejam coloridos independentemente

Uma família de coloração explícita para k^2

Lema

Para $k \geq 1$, *exite algoritmo que:*

- *obtém* $(k^3, k, 2\lceil\sqrt{k}\rceil)$*-coloring family de tamanho* $2^{\mathcal{O}(\sqrt{k}\log k)}$;
- *executa em tempo linear na saída.*

Combinando

Teorema

Para $n, k \geq 1$, *exite algoritmo que:*

- *obtém* $(n, k, 2\lceil\sqrt{k}\rceil)$*-coloring family de tamanho* $2^{\mathcal{O}(\sqrt{k}\log k)} \log n$;
- *executa em tempo* $2^{\mathcal{O}(\sqrt{k}\log k)} n \log n$

Teorema

Existe algoritmo para d*-agrupamento em tempo* $2^{\mathcal{O}(\sqrt{k}(d+\log k))} n^{\mathcal{O}(1)}$.