Erdős-Rényi véletlen gráfok színezése

Felsmann Dániel

Bevezetés

- Erdős-Rényi véletlen gráfok: $G(100, \frac{1}{2})$

Bevezetés

- Erdős-Rényi véletlen gráfok: $G(100, \frac{1}{2})$
- gráfszínezési feladat: úgy szeretnénk G csúcsait $V_1, ..., V_k$ osztályokba sorolni, hogy egyik osztály se feszítsen élt

Bevezetés

- Erdős-Rényi véletlen gráfok: $G(100, \frac{1}{2})$
- gráfszínezési feladat: úgy szeretnénk G csúcsait $V_1, ..., V_k$ osztályokba sorolni, hogy egyik osztály se feszítsen élt
- $\chi(G)$ a minimálisan szükséges osztályok száma

Bevezetés

- Erdős-Rényi véletlen gráfok: $G(100, \frac{1}{2})$
- gráfszínezési feladat: úgy szeretnénk G csúcsait $V_1, ..., V_k$ osztályokba sorolni, hogy egyik osztály se feszítsen élt
- $\chi(G)$ a minimálisan szükséges osztályok száma
- egy fontos elméleti eredmény:

$$\chi(G) = \frac{n}{2\log_2 n - 2\log_2 \log_2 n - 2\log_2 2 + o(1)}$$

Mohó színező algoritmus

- Inputok: $G = (V, E)$, G csúcsainak egy sorrendje

Mohó színező algoritmus

- Inputok: $G = (V, E)$, G csúcsainak egy sorrendje
- az éppen következő v_i csúcs színe legyen a legkisebb olyan szín, amellyel eddig egyetlen szomszédját sem színeztük

Mohó színező algoritmus

- Inputok: $G = (V, E)$, G csúcsainak egy sorrendje
- az éppen következő v_i csúcs színe legyen a legkisebb olyan szín, amellyel eddig egyetlen szomszédját sem színeztük

Stratégiák a csúcssorrend megválasztására:

Mohó színező algoritmus

- Inputok: $G = (V, E)$, G csúcsainak egy sorrendje
- az éppen következő v_i csúcs színe legyen a legkisebb olyan szín, amellyel eddig egyetlen szomszédját sem színeztük

Stratégiák a csúcssorrend megválasztására:

- 100 legjobb

Mohó színező algoritmus

- Inputok: $G = (V, E)$, G csúcsainak egy sorrendje
- az éppen következő v_i csúcs színe legyen a legkisebb olyan szín, amellyel eddig egyetlen szomszédját sem színeztük

Stratégiák a csúcssorrend megválasztására:

- 100 legjobb
- fokszám szerint csökkenő sorrend

Mohó színező algoritmus

- Inputok: $G = (V, E)$, G csúcsainak egy sorrendje
- az éppen következő v_i csúcs színe legyen a legkisebb olyan szín, amellyel eddig egyetlen szomszédját sem színeztük

Stratégiák a csúcssorrend megválasztására:

- 100 legjobb
- fokszám szerint csökkenő sorrend
- szaturáció szerint csökkenő sorrend. Egy v csúcs szaturációs foka v szomszédain felhasznált színek száma.

Javított mohó színezés

- Adott G egy jó színezése

Javított mohó színezés

- Adott G egy jó színezése
- Ha G csúcsait úgy soroljuk fel, hogy az egy színosztályba eső csúcsok kövessék egymást, akkor a mohó színezés által használt színek száma nem nőhet

Javított mohó színezés

- Adott G egy jó színezése
- Ha G csúcsait úgy soroljuk fel, hogy az egy színosztályba eső csúcsok kövessék egymást, akkor a mohó színezés által használt színek száma nem nőhet
- Algoritmus: ismételjük a javító eljárást a színosztályok véletlen sorrendjeire

Eredmények

A mohó színezés eredményei:

	16	17	18	19	20	>20	Átlagos futásidő (ms)
10 legjobb	-	-	0.0006	0.0201	0.1661	0.8132	0.9111
	0.0002	0.0275	0.3832	0.5301	0.0588	0.0002	
Largest first	-	0.0010	0.0415	0.3020	0.4608	0.1947	0.1654
	0.0003	0.0438	0.4659	0.4586	0.0314	0.0001	
Saturation largest first	0.0036	0.1317	0.5108	0.3133	0.0394	0.0012	2.7447
	0.0058	0.2074	0.6088	0.1749	0.0031	-	

Eredmények

A mohó színezés eredményei:

	16	17	18	19	20	>20	Átlagos futásidő (ms)
10 legjobb	-	-	0.0006	0.0201	0.1661	0.8132	0.9111
	0.0002	0.0275	0.3832	0.5301	0.0588	0.0002	
Largest first	-	0.0010	0.0415	0.3020	0.4608	0.1947	0.1654
	0.0003	0.0438	0.4659	0.4586	0.0314	0.0001	
Saturation largest first	0.0036	0.1317	0.5108	0.3133	0.0394	0.0012	2.7447
	0.0058	0.2074	0.6088	0.1749	0.0031	-	

A javított mohó színezés eredményei:

	15	16	17	18	19	>19	Átlagos futásidő (ms)
$k := 10$	-	-	0.0216	0.4223	0.5297	0.0264	0.785
$k := 100$	-	0.0367	0.6582	0.3051	-	-	7.742
$k := 500$	0.0018	0.3451	0.6521	0.0010	-	-	38.517
$k := 1000$	0.0073	0.6072	0.3855	-	-	-	76.875

Tabu keresés általános működése

- heurisztikus optimalizáló algoritmus

Tabu keresés általános működése

- heurisztikus optimalizáló algoritmus
- S keresési tér

Tabu keresés általános működése

- heurisztikus optimalizáló algoritmus
- S keresési tér
- $f : S \to \mathbb{R}$ célfüggvény

Tabu keresés általános működése

- heurisztikus optimalizáló algoritmus
- S keresési tér
- $f : S \to \mathbb{R}$ célfüggvény
- $s \in S$ megoldás $N(s)$ szomszédai

Tabu keresés általános működése

- heurisztikus optimalizáló algoritmus
- S keresési tér
- $f : S \to \mathbb{R}$ célfüggvény
- $s \in S$ megoldás $N(s)$ szomszédai
- Általános lépés: az s megoldásból a legjobb célfüggvény értékű szomszédjába lépünk, ha az nem szerepel a tabu listán:

Tabu keresés általános működése

- heurisztikus optimalizáló algoritmus
- S keresési tér
- $f : S \to \mathbb{R}$ célfüggvény
- $s \in S$ megoldás $N(s)$ szomszédai
- Általános lépés: az s megoldásból a legjobb célfüggvény értékű szomszédjába lépünk, ha az nem szerepel a tabu listán:
- ha s-ből s'-be léptünk, akkor az $s' \to s$ lépés felkerül a tabu listára, a legrégebben oda került lépést pedig töröljük a listáról

Tabu színezés

- megoldások: G csúcsainak egy $V_1, ..., V_k$ k részes partíciója (színezés)

Tabu színezés

- megoldások: G csúcsainak egy $V_1, ..., V_k$ k részes partíciója (színezés)
- célfüggvény: a színosztályon belüli, "rossz" élek száma

Tabu színezés

- megoldások: G csúcsainak egy $V_1, ..., V_k$ k részes partíciója (színezés)
- célfüggvény: a színosztályon belüli, "rossz" élek száma
- egy s megoldás szomszédai: az egyik csúcsot V_i-ből V_j-be rakjuk

Tabu színezés algoritmus

Input	$G = (V, E)$
	k a felhasználható színek száma
	$\|T\|$ a tabu lista mérete
	N a lépésenként generált szomszédok száma
	maxiter: az iterációk maximális száma
Inicializálás	s véletlen színezés generálása
	i := 0
	$T := \emptyset$
while $f(s) > 0$ és $i <$ maxiter	
	s N szomszédjának generálása
	s^* ezek közül a legjobb, ami nem tabu
	$s := s^*$
	T frissítése
	$i := i + 1$
Output	Ha $f(s) = 0$, akkor egy színezés k színnel
	ha $f(s) > 0$, akkor nem találtunk jó színezést

A tabu színezés eredményei

Színszám	Sikeres színezések aránya	Átlagos futásidő (ms)
14	0.023	3206.44
15	0.922	774.97
16	1	41.82