دانشکده علوم ریاضی
دانشگاه صنعتی شریف

بهینه‌سازی ترکیبیاتی

محمدهادی فروغمنداعرابی
بهار ۱۳۹۶

الگوریتم یافتن تطابق کامل با وزن کمینه

جلسه بیست و چهارم

نگارنده: درسا فتح اللهی، گلنوش شاه کرمی

۱ تعاریف

۱. M : مجموعه ای از یال ها که دو به دو راس مشترک ندارند. در ابتدا این مجموعه تهی است. (درواقع یک تطابق نیمه کاره است.)

۲. Ω : ساختار تودرتوی غنچه ها

۳. $\delta(u)$: همه ی یال هایی که از یک غنچه خارج میشوند.

۴. تابع $\Pi : \Omega \to \mathbb{R}$ که در آن:

$\Pi(u) \geq 0 \quad |u| \geq 3$ u غنچه است.۳

$\Pi(u) = val \quad |u| = 1$ u تک راسی است.۱

در ابتدای همه ی مقادیر Π را ۰ قرار میدهیم.

$$\Sigma_{u \in \Omega : e \in \delta(u)} \Pi(u) \leq \omega_e$$

اگر M تطابقی کامل باشد و تابع Π را مطابق شرط بالا تعریف کرده باشیم،داریم:

$$\omega(M) \geq \Sigma_{u \in \Omega} \Pi(u)$$

در حین اجرای الگوریتم و به روزرسانی مقادیر تابع Π هدف زیاد کردن مقدار $\Sigma\Pi(u)$ می باشد اما درصورتی که در الگوریتم به حالت تساوی برای نامساوی بالا برسیم مچینگ با وزن کمینه نیز یافت شده است پس این کار را ادامه میدهیم تا وزن تطابق با $\Sigma\Pi(u)$ ها برابر شود.

جنگل M درمیان

در این جنگل رئوس عمق $\circ$ رئوس پوشیده نشده توسط یال های مچینگ می باشند و هر مسیری از رئوس عمق $\circ$ به رئوس عمق دیگر با استفاده از یال های درخت مسیری متناوب اما غیر افزایشی است . در این صورت در هر مرحله تعداد رئوس اعماق زوج از رئوس اعماق فرد بیشتر است . (به اندازه ی تعداد درخت های جنگل)

۲ الگوریتم

پایه الگوریتم

در ابتدا $M = \emptyset$ هم چنین Ω برابر با مجموعه رئوس است و Π هرکدام از آنها برابر $\circ$ می باشد .

مراحل

در هر مرحله یک جنگل داریم. میخواهیم تلاش کنیم تطابق را بزرگ تر کنیم. باید π مربوط به راس Ω را اضافه کنیم.

Π مربوط به اعماق زوج را با یک سرعت زیاد می کنیم و اعماق فرد را ب همان اندازه کم می کنیم.

آیا می توان این کار را تا انتها انجام داد؟ خیر، شرط ها خراب می شوند. (هر دو شرط ممکن است خراب شود.)

α : اولین جایی که خراب می شود.

حال دو حالت داریم:

۱) شرط مربوط به یال e خراب شود.

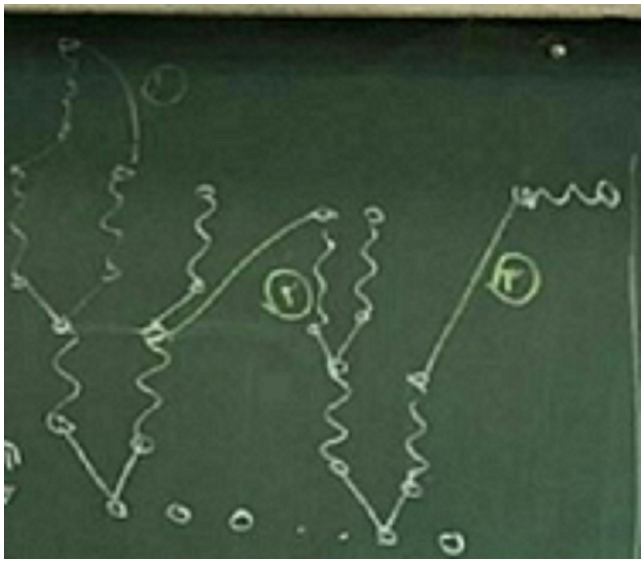
۲) شرط مربوط به راس v خراب شود.

۱.۲ حالت اول

در این حالت یال های درخت خراب نمی شوند زیرا یکی کم و دیگری اضافه می شود.

حال یالی را در نظر بگیرید که دو سر آن اضافه می شود. یک یا دو سر آن عمق زوج است.

یکی از ۳ حالت زیر رخ می دهد:



۳: جنگل را گسترش می دهیم

۲: مسیر افزایشی را اعمال می کنیم و درخت را از اول شروع می کنیم. (تعداد حالت هایی که رخ می دهد خیلی کم است)

۱: غنچه بکنیم.

بین دو عمق زوج یک درخت مسیر پیدا شده است.

راس های روی دور را حذف می کنیم. (C)

C را به Ω اضافه می کنیم و $\pi(C) = \circ$ می گذاریم.

۲.۲ حالت دوم

U ای که $\pi(U)=0$ را باز می کنیم.
تطابق را روی U می چرخانیم تا راس پوشیده شده داخلی روی یال تطابق بیفتد.
مسیر زوج روی درخت را در جنگل نگه می داریم و مچینگ های داخل مسیر فرد را از جنگل جدا می کنیم .
U را از Ω حذف می کنیم.

۳ زمان اجرا

تعداد بارهای افزایش تطابق (O(v)) * حداکثر زمان بین دو افزایش تطابق

| مجموعه راس های درون یکی از غنچه های عمق زوج| + | تعداد راس های نا زوج (فرد و رها) در جنگل| ۲

وقتی یک دور فرد را باز می کنیم، یکی از دو عبارت بالا اضافه می شوند.
وقتی جنگل را گسترش می دهیم تعداد راس های نا زطوج دوتا کم می شود و تعداد راس های زوج یکی اضافه می شود. (ضریب دو عبارت بالا برای این حالت گسترش جنگل نیاز است.)
در نتیجه حداکثر تعداد بارهایی که الگوریتم می خواهد π را اضافه کند $O(v^2)$ است.
زمان اجرا = $O(v^2)O(ElogE)$
الگوریتمی با زمان اجرای چندجمله ای برای تطابق کامل با وزن کمینه در گراف با تطابق کامل وجود دارد.

۴ بهینه بودن الگوریتم

نشان می دهیم

$$\omega(M) = \Sigma_{u\in\Omega}\Pi(u)$$

که معادل نشان دادن دو عبارت زیر است:

۱. $e \in M : w_e = \Sigma_{v,e\in\delta(v)}\Pi(v)$

۲. $v \in \omega : |M \cap \delta(v)| \leq 1$

طبق الگوریتم به دلیل اینکه هر یال مچینگ زمانی عضو جنگل بوده است و با خروج این از جنگل جمع Π های شامل آن تغییری نمی کند شرط تساوی برقرار است و رابطه اول درست است.
طبق غنچه بودن این رابطه دوم برقرار است . پس وزن M کمینه است و حکم ثابت می شود.