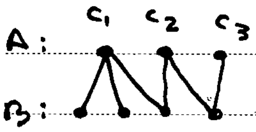
الگوریتم های تقریبی:

اگر برای یک مساله‌ی بیشینه سازی با جواب بهینه OPT الگوریتمی ارائه دهیم که جوابش همیشه حداقل $\frac{OPT}{\alpha}$ باشد، الگوریتم ما یک الگوریتم α-تقریبی است.

اگر برای یک مساله‌ی کمینه سازی با جواب بهینه OPT الگوریتمی ارائه دهیم که جوابش همیشه حداکثر αOPT باشد، الگوریتم ما یک الگوریتم α-تقریبی است.



مساله‌ی پوشش مجموعه‌ی کمینه (min. set cover):

یک گراف دوبخشی با مجموعه رئوس A و B را در نظر بگیرید به طوری که به هر عضو از مجموعه‌ی A، یک وزن نامنفی انتساب داده باشیم. می‌خواهیم $I \subseteq A$ را انتخاب کنیم به طوری که هر عضو از B به حداقل یکی از اعضای I متصل باشد و مجموع وزن اعضای I کمینه شود.

به بیان دیگر می‌توان هر عضو $s \in A$ را نماینده‌ی یک $S \subseteq B$ گرفت به این صورت که $s \in A$ به $v \in B$ متصل است اگر $v \in S$ باشد. پس مساله انتخاب تعدادی از زیرمجموعه های B است که کل B را بپوشانند و مجموع وزنشان کمینه باشد.

هدف این است که با استفاده از روش اولیه-دوگان یک الگوریتم α-تقریبی ارائه دهیم.

$$(IP):\ \min\ c^T x \qquad\qquad (P):\ \min\ c^T x$$

$$y \rightsquigarrow \sum_{S: v \in S} x_S \geq 1 \quad \forall v \in B \qquad y \rightsquigarrow \sum_{S: v \in S} x_S \geq 1 \quad \forall v \in B$$

$$x_S = \{0,1\} \quad \forall s \in A \xrightarrow[\text{شده}]{\text{ریلکس}} x_S \geq 0 \quad \forall s \in A$$

Subject:
Date

$$(D): \max \ 1^T y$$
$$\sum_{v \in S} y_v \le c_S \quad \forall S \in A$$
$$y_v \geq 0 \quad \forall v \in B$$

$$(DRP): \max \sum_{v \in B} z_v$$
$$\sum_{v \in S} z_v \le 0 \quad \forall S \in J$$
$$z_v \geq 0 \quad v \in J$$
$$z \le 1$$

اگر قرارداد کنیم که به z ها فقط مقادیر 0 و 1 اختصاص دهیم، برای S های در A که $S \in J$ باشد داریم $\sum_{v \in S} z_v \le 0$ پس v های در B که به آن S ها متصل هستند را باید z شان را صفر دهیم. در نتیجه تنها z_v هایی را می‌توانیم 1 دهیم که توسط اعضای J پوشیده نشده باشند.

هر جواب مثبت برای (DRP) می‌تواند مقدار تابع هدف در (D) را افزایش دهد. پس پیدا کردن حتی یک راس نپوشیده در B هم کافی است.

الگوریتم تقریبی برای حل مسئله‌ی پوشش کمینه راسی:

مقداردهی اولیه:

$I \leftarrow \emptyset$ نشان دهنده‌ی اعضای A که انتخاب می‌کنیم و البته این اعضا همگی tight خواهند بود (یعنی قیدهایشان در دوگان tight است)

$y \leftarrow 0$ یک جواب شدنی برای دوگان که در طول اجرای الگوریتم آن را بهتر می‌کنیم و از روی آن I را حساب می‌کنیم.

مراحل زیر را تکرار کن:

یک v نپوشیده انتخاب کن و y_v را افزایش بده تا جایی که حداقل یکی از شرط‌های (D) tight شود. یعنی: $\epsilon = \min_{S: v \in S} \left(c_S - \sum_{w \in S} y_w\right)$ قرار بده: $y_v \leftarrow y_v + \epsilon$

اگر نامعادله‌ی (D) برای $S \in A$ با این کار tight شود قرار بده: $I \leftarrow I + S$

تحلیل هزینه‌ی جواب الگوریتم: می‌خواهیم بدانیم $\sum_{S\in I} c_S$ با IP^* (یعنی جواب بهینه IP) چه رابطه‌ای دارد.

$$\sum_{S\in I} c_S = \sum_S c_S x_S \overset{①}{=} \sum_S x_S \Big(\sum_{v\in S} y_v\Big)$$

①: به این دلیل که فقط $S\in A$ هایی را انتخاب کردیم که $S\in J$ بود، یعنی قیدِ متناظرش در (D) tight بود. به عبارت دیگر:

$$x_S > 0 \rightarrow \sum_{v\in S} y_v = c_S$$

حالا جای $\sum$ ها را عوض می‌کنیم:

$$\sum_{S\in I} c_S = \sum_v y_v \Big(\sum_{S: v\in S} x_S\Big)$$

اگر شرط مکمل لنگی برای y ها و قیدهای متناظرشان در (P) برقرار بود می‌توانستیم بنویسیم:

$$y_v > 0 \rightarrow \sum_{S: v\in S} x_S = 1$$

و نتیجه بگیریم:

$$\sum_{S\in I} c_S = \sum_v y_v \le D^* = P^* \le IP^* \rightsquigarrow$$ جواب الگوریتم بهینه است

اما مشکل اینجاست که برای $y_v > 0$ لزوماً شرط متناظرش در (P) tight نیست پس برای ادامه‌ی کار، تلاش می‌کنیم نشان دهیم: $y_v > 0 \rightarrow \sum_{S: v\in S} x_S \le \alpha$ که نتیجه می‌دهد:

$$\sum_{S\in I} c_S = \alpha \sum_v y_v \le \alpha D^* = \alpha P^* \le \alpha IP^* \rightsquigarrow$$ الگوریتم ما یک الگوریتم α-تقریب است

می‌دانیم که چون x_S ها را ۰ و ۱ انتخاب کرده‌ایم پس $y_v > 0 \rightarrow \sum_{S: v\in S} x_S \le |\{S : v\in S\}|$

در نتیجه اگر ماکسیمم این مقدار را حساب کنیم: $f = \max_v |\{S : v\in S\}|$

نشان دادیم که الگوریتم ما f-تقریبی است که f حداکثر درجه برای اعضای B را نشان می‌دهد.

مساله حذف دورها:

در گراف $G(V)$ داده شده که هر کدام از گره‌های آن یک وزن نامنفی دارند، می‌خواهیم $I \subseteq V$ را از گراف حذف کنیم به طوری که $G(V \setminus I)$ دور نداشته باشد و $\sum_{v \in I} c_v$ کمینه شود.

برای اینکه یک دور حذف شود باید حداقل یکی از گره‌های آن را حذف کنیم. می‌توان به این مساله به صورت یک مساله‌ی پوشش مجموعه‌ای کمینه نگاه کرد که در آن مجموعه‌ی A گره‌های گراف باشند و مجموعه‌ی B دورهای گراف. $v \in A$ به $c \in B$ وصل است اگر v یکی از گره‌های دور c در گراف باشد.

با همان الگوریتم قبلی می‌توان این مساله را در زمان چندجمله‌ای حل کرد. به این دلیل که عملیات پیدا کردن یک حلقه در گراف چندجمله‌ای است و با اینکه تعداد اعضای B به نسبت $|V|$ می‌تواند نمایی باشد اما چون در هر مرحله اجرای الگوریتم یک عضو A که قبلاً انتخاب نشده را انتخاب می‌کنیم و $|V| = |A|$ پس تعداد دفعاتی که حلقه‌ی الگوریتم اجرا می‌شود $|V|$ است.

اما مشکل اینجاست که الگوریتم قبلی برای حل این مساله $|V|$-تقریبی است. به این دلیل که اگر دوری در گراف شامل همه‌ی گره‌ها باشد، درجه‌ی $c \in B$ متناظر آن دور، $|V|$ می‌شود.

الگوریتم قبلی را یک بار برای این مساله می‌نویسیم اما با این تفاوت که یک مرحله‌ی تمیزکاری به آن اضافه می‌کنیم. با توجه به اینکه راس‌های درجه ۱ و ۰ در تولید جواب نهایی نقشی ندارند می‌توان آن‌ها را به صورت تکراری حذف کرد. بدین ترتیب مسیرهای موجود در G حذف می‌شوند و گراف باقی مانده فقط شامل راس‌های با درجه ۲ یا بیشتر است.

الگوریتم تقریبی برای حل مسأله حذف دورها:

$\emptyset \to I$

$0 \to y$

تا زمانی که گراف دور دارد مراحل زیر را تکرار کن:

۱- تا زمانی که راس درجه ۰ یا ۱ در گراف هست آن را حذف کن

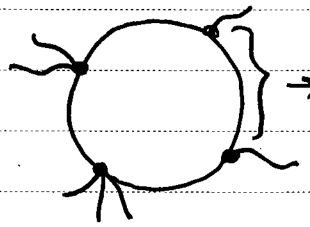
۲- یک دور مانند C را در نظر بگیر

۳- $\epsilon = \min_{v \in C}(c_v - \sum_{v \in C} y_c)$ ، $y_c \leftarrow y_c + \epsilon$ و آن رئوسی که به ازای $v \in C$ tight شده باشد قرار بده: $I \leftarrow I + v$ و v را از گراف حذف کن.

حال در نظر داریم برای $y_c \ge 0$ یک حد بالا روی $\sum_{v \in C} x_v$ بدهیم. برای این کار الگوریتم را به صورتی تغییر می‌دهیم که c هایی را انتخاب کند که برای حذفشان نیاز باشد حداکثر x_v $v \in C$ را انتخاب کنیم (برابر با ۱ قرار دهیم).

اگر درجه‌ی همه‌ی راس‌های یک دور برابر با ۲ باشد کافی است یکی از گره‌هایش را از گراف حذف کنیم تا کل دور با مرحله‌ی ۱ الگوریتم (عملیات ساده‌سازی) از گراف حذف شوند.

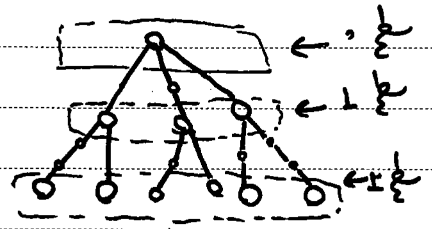
اما اگر k تا گره با درجه‌ی بیشتر از ۲ داشته باشیم، در بدترین حالت مقدار $\sum_{v \in C} x_v$ یعنی $|I \cap C|$ برابر با $2k$ می‌شود. k تا برای راس‌های درجه ۳ یا بیشتر و k تا برای مسیرهای درجه ۲ بین این گره‌ها:



مسیر متشکل از گره‌های درجه ۲ بین دو راس با درجه بیشتر از ۲

پس کافی است نشان دهیم در هر مرحله وجود دارد دوری که تعداد راس‌های درجه ۳ یا بیشتر (یعنی k) خیلی زیاد نمی‌شود و یک الگوریتم چندجمله‌ای برای پیدا کردن چنین دوری ارائه دهیم.

یک الگوریتم شبیه (breadth first search) BFS پیاده کنید به صورتی که فقط گره های با درجه ۳ یا بیشتر در سطح های آن باشند. یعنی با مسیرهای متشکل از راس های درجه ۲ بین دو تا راس درجه ۳ یا بیشتر، مثل یال رفتار کنید. از یک راس درجه ۳ یا بیشتر دلخواه درخت را گسترش دهید.



اگر هر کدام از گره های سطح h' به هم یال داشته باشند، یک دور در گراف G داریم که شامل $2h'+1$ راس با درجه ی حداقل ۳ است.

حداکثر ارتفاع این درخت $\lceil \log n \rceil$ است پس وجود دارد دوری در G که حداکثر $2\lceil \log n \rceil$ گره با درجه ی بیشتر از ۲ دارد و با این الگوریتم شبیه BFS در زمان چند جمله ای پیدا می شود.

پس داریم $|I \cap C| \leq 4\lceil \log n \rceil \rightarrow$ $\% > 0$.

اگر مرحله ی ۲ الگوریتم را تغییر دهیم که به جای C دلخواه، C را به صورت گفته شده پیدا کند الگوریتم ما یک الگوریتم $4\lceil \log n \rceil$ - تقریبی خواهد بود.

مساله‌ی جنگل اشتینر:

یک سری زوج از رأس‌ها $\{s_i, t_i\}$ داریم که در گراف وزن‌دار، می‌خواهیم زیرمجموعه‌ی F از یال‌ها را انتخاب کنیم که در G_F هر s_i به t_i متناظرش وصل باشد به طوری که مجموع وزن‌های F کمینه شود.

واضح است که F حتماً یک جنگل است. به این دلیل که اگر دور داشته باشد می‌توان یک یال از آن دور را حذف کرد و $\{s_i, t_i\}$ ها هنوز به هم متصل باشند و به این صورت مجموع وزن F کاهش یابد.

$$(IP):\ \min\ c^T x \qquad\qquad (P):\ \min\ c^T x$$

$$y \rightsquigarrow \sum_{e\in\delta(S)} x_e \geq 1 \quad \forall S \in \cup S_i \qquad y \rightsquigarrow \sum_{e\in\delta(S)} x_e \geq 1 \quad \forall S\in \cup S_i$$

$$x_e \in \{0,1\} \xrightarrow{\text{ریلکس شده}} x_e \geq 0$$

که S_i ها به این صورت تعریف می‌شوند: $S_i = \{S \subseteq V \mid |S \cap \{s_i, t_i\}| = 1\}$

یعنی S_i مجموعه‌ای از زیرمجموعه‌های V است که دقیقاً یکی از s_i یا t_i را داشته باشند.

به این S_i ها زیرمجموعه‌های فعال V می‌گوییم.

$$(D):\ \max \sum_{S\in\cup S_i} y_S \qquad\qquad (DRP):\ \max \sum_{S\in \cup S_i} z_S$$

$$\sum_{S: e\in\delta(S)} y_S \leq c_e \quad \forall e\in E \qquad\qquad \sum_{S: e\in\delta(S)} z_S \leq 0 \quad \forall e \in J$$

$$y_S \geq 0 \quad \forall S\in \cup S_i \qquad\qquad z_S \geq 0 \quad S\in J$$

$$z_S \leq 1 \quad \forall S \in \cup S_i$$

اگر مقدار دار کنیم که z_S ها را نامنفی نگه داریم، با توجه به شرط $\sum_{S: e\in\delta(S)} z_S \leq 0$, $e \in J$ ، z_S ها را برای S هایی که مجموعه‌ی J را قطع کنند، برابر با ۰ قرار می‌دهیم.
در نتیجه فقط z_S هایی را زیاد می‌کنیم که S یک مولفه‌ی همبندی در G_J باشد.

الگوریتم تقریبی برای حل مسائل جنگل اشتاینر:

مقداردهی اولیه:

$F \leftarrow \emptyset$ نشان دهنده‌ی یال‌هایی که خریداری می‌کنیم.

$Y \leftarrow 0$ یک جواب شدنی دوگان.

در G_F یک مولفه‌ی همبندی فعال S پیدا کن

Y_S را اضافه کن تا حداقل یکی از نامساوی‌های دوگان tight شود:

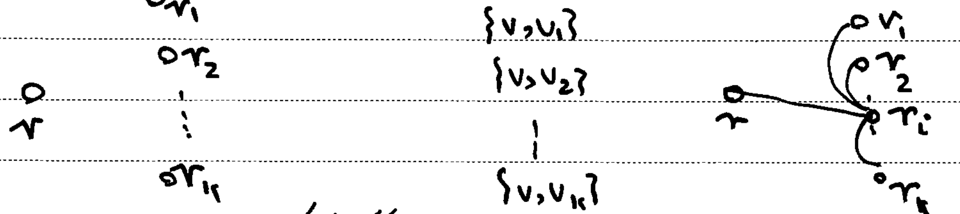
$\varepsilon = \min_{e\in\delta(S)} \left(c_e - \sum_{S'\in\delta(e)} Y_{S'} \right)$ و $Y_S \leftarrow Y_S + \varepsilon$. اگر قیدی در دوگان برای یالی مانند e
tight شود آن یال را خریداری می‌کنیم: $F \leftarrow F + e$

تحلیل هزینه‌ی جواب الگوریتم:

$$\sum_{e\in F} c_e = \sum_e x_e c_e = \sum_e x_e \sum_{S: e\in\delta(S)} Y_S = \sum_{S\in US_i} Y_S \sum_{e\in\delta(S)} x_e$$

اگر برای $Y_S > 0$ ثابت کنیم $\sum_{e\in\delta(S)} x_e \leq \alpha$ داریم: $\sum_{e\in F} c_e = \alpha \sum_{S\in US_i} Y_S = \alpha D^* \leq \alpha IP^*$
یعنی یک الگوریتم α-تقریبی داریم.

وجود دارد مثالی با $k+1$ راس که این مقدار ($\sum_{e\in\delta(S)} x_e$) ~~[illegible]~~ برای یکی برابر با k شود.



گراف کامل روی این راس‌ها که وزن همه‌ی یال‌ها ۱ باشد

زوج رئوس‌هایی که باید به هم وصل باشند

اگر Y_{v_1} را برای زیاد کردن انتخاب کنیم همه‌ی یال‌هایی که از آن خارج می‌شوند خریده می‌شوند

در این مثال $\sum_{e\in\delta(v_j)} x_e = k$ هست و یک جواب با $\sum_{S\in U_{s_i}} y_S = 1$ داریم.

هدف این بود که تا جایی که می توانیم $\sum_{S\in U_{s_i}} y_S$ را زیاد کنیم. پس یک تغییری در روند انتخاب y_S هایی که افزایش می دهیم می کنیم.

کم تا هم هدف دوگان

مجموعه ی مولفه های همبندی G_F را C بنامید (که فقط شامل مولفه های فعال باشد).
y_S را برای هر $S\in C$ به مقدار ε اضافه می کنیم.
$y_S \leftarrow y_S+\varepsilon \quad \forall S\in C$ و $\varepsilon = \min_{S\in C}\left(\min_{e\in\delta(S)}\left(c_e-\sum_{S'\in\delta(e)} y_{S'}\right)\right)$. یال e که برای آن قید دوگان متناظرش tight شده باشد را خریداری می کنیم. $F\leftarrow F+e$

فرض کنید یال هایی که در طول اجرای الگوریتم خریداری کرده ایم را با شماره تعدادی که در الگوریتم خریداری شده اند اندیس گذاری کنیم و به نسبت اندیس به صورت صعودی مرتب کنیم.

یک مرحله ی تمیزکاری به الگوریتم اضافه می کنیم به این صورت که از آخر به اول e هایی که می توان حذفشان کرد و جواب مساله را ناشدنی نمی کنند، حذف می کنیم. البته ترتیب حذف کردن e ها در جواب تاثیری ندارد ولی اینجا برای ساده سازی یک سری تحلیل های بعدی به ترتیب حذف کردیم.

لم: در هر مرحله از اجرای الگوریتم: $\sum_{S\in C} |\delta(S)\cap F'| \le 2|C|$
که در اینجا F' جواب نهایی الگوریتم است که پس از حذف کردن تعدادی از یال های F بدست می آید. از آنجایی که در هر مرحله از اجرای الگوریتم فقط یال هایی خریداری می شوند که بین $S\in C$ ها باشند (چون y_S را برای مولفه های همبندی که آن را قطع می کنند زیاد می کنیم) پس $|\delta(S)\cap F'|$ تعداد یال های خارج شونده از S را نشان می دهد که در جواب نهایی ظاهر شده است.

اثبات این لم را فعلاً به تعویق می اندازیم.

Subject:
Date

تضمین: الگوریتم زیر یک الگوریتم ۲-تقریبی برای حل مسأله جنگل اشتاینر است.

مقداردهی اولیه:

$\emptyset \to F$

$0 \to Y$

$0 \to l$ شمارش تکرار حلقه که برای اندیس‌دهی یال‌ها استفاده می‌شود

تا زمانی که تمام زوج‌های $\{s_i, t_i\}$ در G_F به هم وصل نشده‌اند مراحل زیر را تکرار کن:

1- $l+1 \to l$

2- C را برابر با مولفه‌های همبندی G_F که فعال هستند $(\exists i,\ |C \cap \{t_i, s_i\}| = 1)$ قرار بده

3- Y_S را برای هر $S \in C$ به مقدار یکنواخت اضافه کن تا برای یک یال e قید دوگان tight شود، $\epsilon = \min_{S \in C}\left(\min_{e \in \delta(S)}\left(c_e - \sum_{e \in \delta(S')} Y_{S'}\right)\right)$ ، $Y_S \leftarrow Y_S + \epsilon, \forall S \in C$ ، $F \leftarrow F + e_l$

$F \to F'$

برای $k \leftarrow l$ تا 1 مراحل زیر را تکرار کن

اگر $F' - e_k$ یک جواب شدنی برای مسأله است

e_k را از F' حذف کن

F' را به عنوان جواب مسأله برگردان

برای اینکه ثابت کنیم این الگوریتم ۲-تقریبی است کافی است نشان دهیم $\sum Y_S \overbrace{|\delta(S) \cap F'|}^{\sum_{e \in \delta(S)} x_e} \le 2\sum Y_S$
برای این کار نشان می‌دهیم که این نامساوی در ابتدای کار برقرار است (Y ها صفر هستند پس چون $0 \le 0$ برقرار است) سپس نشان می‌دهیم در هر مرحله اجرای الگوریتم مقداری که به سمت راست نامساوی اضافه می‌شود از مقداری که به سمت چپ آن اضافه می‌شود بزرگتر یا مساوی است.

با توجه به اینکه y ها برای $S\in C$ در هر مرحله به مقدار ε اضافه می‌شوند پس تغییرات عبارت سمت راست هست: $2\sum_{S\in C}\varepsilon = \varepsilon 2|C|$

و تغییرات عبارت سمت چپ هست: $\sum_{S\in C}\varepsilon|\delta(S)\cap F'|$

پس کافی است ثابت کنیم در هر مرحله: $\varepsilon\sum_{S\in C}|\delta(S)\cap F'| \le \varepsilon 2|C|$

یعنی $\sum_{S\in C}|\delta(S)\cap F'|\le 2|C|$ که طبق لم فرض کردیم درست است.

یک تعبیر از حل مساله‌ی دوگان: می‌خواهیم دور S ها به عرض y_S خندق بکنیم به طوری که مجموع خندق‌هایی که یک یال باید از روی آن عبور کند حداکثر به اندازه‌ی وزن آن یال باشد

اثبات لم: مرحله‌ی i ام از تکرار الگوریتم را در نظر بگیرید که یال e_i به F اضافه می‌شود. F_i را مقدار همه یال‌هایی که تا اینجای اجرای الگوریتم به F اضافه شده‌اند. یعنی $F_i=\{e_1,\dots,e_i\}$ قرار دهید. $H=F'-F_i$ یعنی مجموعه یال‌هایی که باید به F_i اضافه شوند تا جواب را تولید کنند. دقت کنید که هیچکدام از اعضای H اضافی نیستند.

یک گراف جدید بسازید که در آن مولفه‌های همبندی G_{F_i} منقبض شده باشند. با توجه به اینکه F در هر مرحله یک جنگل است پس با اضافه کردن یال‌های H به گراف جدید، یک جنگل داریم.
فرض کنید $\deg(v)$ درجه‌ی راس v در این گراف جدید را نشان می‌دهد.
راس‌های گراف را به دو رنگ قرمز و آبی رنگ کنید که راس‌های قرمز نماینده‌ی مولفه‌های فعال در G_{F_i} باشند و راس‌های آبی مولفه‌های غیرفعال. مولفه‌های غیرفعال درجه ۰ را حذف کنید.
در نظر داشتیم نامساوی زیر را در G_{F_i} اثبات کنیم:

$$\sum_{S\in C}|\delta(S)\cap F'|\le 2|C|$$

معادل این است که نامساوی زیر را در گراف جدید اثبات کنیم:

$$\sum_{v\in R}\deg(v)\le 2|R|$$

که R مجموعه‌ی راس‌های قرمز رنگ است.

با توجه به اینکه گره های آبی با درجه ی صفر را حذف کردیم پس در حال حاضر درجه ی هر گره ی آبی حداقل 1 است. اما نشان خواهیم داد که نمی تواند دقیقاً 1 هم باشد پس حداقل 2 است. به این دلیل که اگر درجه ی آن 1 باشد، با توجه به اینکه هیچکدام از یال های H اضافه نیستند پس این یال باید در مسیر بین یک $s_i t_i$ وجود داشته باشد. اما با توجه به اینکه دقیقاً یک یال از این مجموعه خارج می شود پس یکی از s_i یا t_i باید داخل این مولفه (که معادلش یک راس آبی قرار داده ایم) بوده باشد که با غیر فعال بودن این مولفه تناقض دارد.

از طرفی داریم:

$$\sum_{v \in R} deg(v) = \sum_{v \in R \cup B} deg(v) - \sum_{v \in B} deg(v)$$

پس داریم:

$$\sum_{v \in R} deg(v) \leq 2(|R| + |B|) - 2|B| = 2|R|$$

چون مجموع درجات راس ها در یک جنگل حداکثر دو برابر تعداد راس هاست

چون درجه ی هر B در جنگل حداقل 2 است

اثبات کامل شد.