Subject: «برنامه ریزی عدد صحیح»

Yare: Month: Day: Page: ( )

* جلسه‌ی اول :

پیش فرض : OR1 + OR2

* مسئله‌ی TSP ، همواره یکسان است ولی آن را می توان با مدل های مختلفی نمایش داد.

← لذا حل شدن مسئله ، اهمیت بسیار زیادی خواهد داشت ؛ پس این که بتوانیم مدل سازی کنیم ، هیچ ارزشی نخواهد داشت ، بلکه باید به دنبال حل مدل باشیم.

← اگر بتوانیم مثلاً مسئله‌ای را در قالب IP مدل کنیم ، می توانیم از امکانات آن تئوری برای حل آن مسئله استفاده نماییم*

⇐ پس چون مدل سازی های مختلف ، امکانات خاصی را برای ما فراهم می آورند ، ما یک مسئله را به طرق مختلفی مدل می کنیم.

مسئله‌ی تصمیم گیری ← مدل سازی ← طراحی الگوریتم ← ارزیابی الگوریتم

طراحی الگوریتم ↓ پیاده سازی الگوریتم

مسئله‌ی تصمیم گیری ↙ مسئله شناسی

مسئله‌ی تصمیم گیری ↘ دسته‌بندی مسائل از منظر سختی حل

← هر مدل ، اجازه‌ی پیاده سازی تمام الگوریتم ها را نمی دهد.

Maral

انواع اجرا

I_1
I_2
I_3
⋮

انواع الگوریتم

A_1
A_2
A_3
⋮

انواع مدل

M_1
M_2
M_3
⋮

GT
NT
DP
IP
CP
SF

* انواع روش‌ها
مدل‌سازی
(problem)

← وظیفه‌ی الگوریتم، حل یک مسئله است. حال این الگوریتم‌ها را با چه معیارهایی ارزیابی کنیم؟

* جلسه‌ی دوم :

$$\min f(x)$$
$$x \in S$$

→ این مسئله "برنامه‌ریزی ریاضی" یا "مدل برنامه‌ریزی ریاضی" گویند. (یا "مدل بهینه‌سازی" گویند)

mathematical program = mathematical programming model

← شاخه‌ای از ریاضیات که به علم فوق می‌پردازد mathematical programming گویند.

$f(x)$: یک تابع حقیقی

$S \subseteq \mathbb{R}^n$

$f: S \to \mathbb{R}$

* مسئله‌ی فوق ممکن است شدنی یا ناشدنی باشد.

شدنی $(S \neq \emptyset)$ ← کران‌دار $(\bar{f} > -\infty)$ ← قابل دسترس / غیرقابل دسترس

ناشدنی $(S = \emptyset)$

بی‌کران $(\bar{f} = -\infty)$

$$\bar{f} = \inf_{x \in S} f(x)$$

* inf هر تابع روی یک مجموعه حتماً وجود دارد که می تواند منفی بینهایت ($-\infty$) هم باشد *

inf = بزرگترین کران پایین تابع بر روی یک مجموعه

← پس اگر inf، یک عدد باشد، اصطلاحاً می گوییم کران دار است.

← وقتی کران یک تابع قابل دسترس است که یک x^* وجود داشته باشد به ازای آن:

$$f(x^*) = \bar{f} \Rightarrow \forall x \in S \Rightarrow f(x^*) \leq f(x)$$

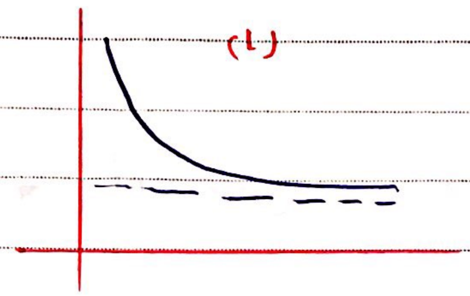
و اگر چنین x^* ای، وجود نداشته باشد، اصطلاحاً می گوییم غیر قابل دسترس است *

x^* = optimal solution = optimum

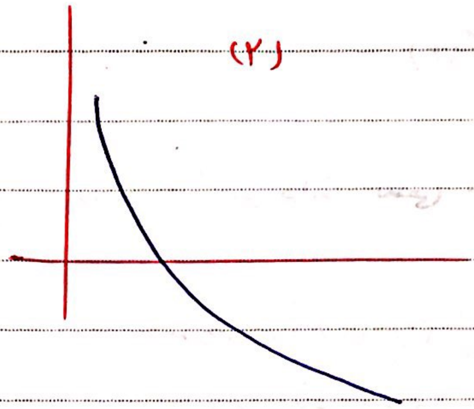
$f(x^*)$ = optimal value

- قابل دسترس (attainable)
  - $x^* = \{$ جواب ها چندتا ← جواب یکتا
  - بهینه چندگانه
    - x^* کران دار
    - x^* بی کران
- غیر قابل دسترس

(۱)

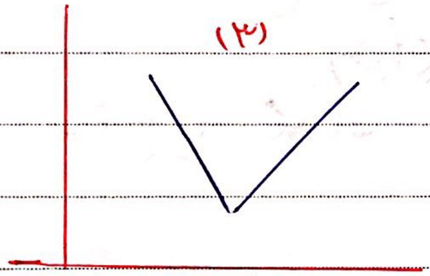


(۲)

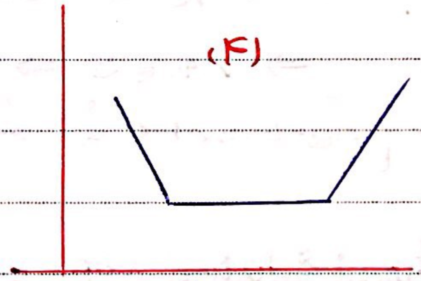


← در نمودار (۱)، f^*، بی‌کران است ولی در نمودار (۲)، خود تابع بی‌کران خواهد برد★

(۳)



(۴)



نمودار (۳) = جواب بهینه یکتا

نمودار (۴) = جواب بهینه‌ی چندگانه

★ تعریف ساده‌ی مدل پایه‌ی بهینه‌سازی ترکیباتی (گسسته):

$\min f(x)$
$x \in S$
$|S| < \infty$

← این، یک مدل پایه‌ی بهینه‌سازی ترکیباتی است وقتی که تعداد اعضای S، متناهی باشد، یعنی: $|S| < \infty$

← پس آن شرط $|S| < \infty$، موجود باشد، حتماً مسئله بهینه‌سازی گسسته است★

← به عبارتی وجه تمایز بهینه‌سازی گسسته آن است که فضای جواب آن متناهی باشد★

← پس مسئله‌ی فوق همیشه جواب بهینه دارد. چرا؟؟

ج ← زیرا فضای جواب متناهی است *

← همچنین این جواب بهینه را همواره می‌توان پیدا کرد. چگونه؟؟

ج ← با طراحی الگوریتمی که تمام جواب‌ها را یکی یکی بررسی کند * (با تعداد قدم‌های متناهی)
که به روش فوق به روش شمارش کامل گویند *

* حال پس چرا باید این درس را ادامه بدانند؟؟ هدف از این درس چیست؟؟

← اگرچه روش شمارش کامل، حتماً جواب بهینه را پیدا می‌کند، ولی همواره روش خوبی برای حل مسائل نیست. مثلاً مسئله کوله پشتی را در نظر می‌گیریم که در آن یک سری اقلامی داریم که هرکدام یک وزن و یک ارزش مشخص دارند. اکنون می‌خواهیم بتوانیم ترکیب بهینه‌ی آن‌ها را به گونه‌ای تعیین کنیم که بهترین و بالاترین ارزش کسب شود *

$$\max \sum_{i=1}^{n} v_i \cdot x_i$$

$$\text{s.t.} \quad \sum w_i \cdot x_i \leq W$$

$$x_i \in \{0,1\} \;,\; i = 0, \ldots, n$$

* مراحل روش شمارش کامل:

← تمام ترکیبات مختلف را تشکیل دهیم و بررسی کنیم که آیا شدنی هستند یا خیر. (مرحله‌ی checking) ← 2^n (فرض ⇐ $n = 100$)

$$2^{100} = (2^{10})^{10} > 10^{30}$$

← لذا با کامپیوتری که در هر ثانیه 10^{15} عملیات را انجام دهد، فقط این مسئله checking، 10^9 سال طول می‌کشد!!!!

Subject:
Yare: Month: Day:

- مدل بهینه سازی عمومی (یا فرم عمومی برنامه ریزی ریاضی):

$$\min_{x \in S} f(x)$$

- فرم استاندارد برنامه ریزی ریاضی (یا برنامه ریزی ریاضی):

$$\min f(x)$$

$$\begin{cases} g_i(x) \leq 0 \quad ; \quad i=1,\dots,m \\ h_j(x) = 0 \quad ; \quad j=1,\dots,p \\ x \in X \end{cases}$$

$(\mathbb{R}^n \subseteq D)$

دامنه مسئله

← حال می گوئیم اگر $x \in \mathbb{Z}^n$ ← برنامه ریزی عدد صحیح

← اگر $f(x)$ ، $g(x)$ و $h(x)$ ، همگی "آفین" باشد، مدل برنامه ریزی خطی گویند
و اگر فقط یکی از آن ها آفین نباشد، مدل برنامه ریزی غیرخطی گویند *

← وقتی می گوئیم LP ← یعنی متغیرها پیوسته هستند و اگر پیوسته نباشد، مدل عدد صحیح گفته می شود *

← هر تابع کوادراتیک (quadratic) به صورت زیر است:

$x^{\beta} y^{\alpha}$ if $(\beta+\alpha \leq 2)$

آن گاه می توان اثبات کرد که هر تابع کوادراتیک را می توان به صورت زیر نوشت:

یک ماتریس $Q = m \times n$

$$x_1 x_2 + x_1^2 - x_2^2 + \beta \xrightarrow[\text{نوشت}]{\text{می توان}} \frac{1}{2} x^T Q x + b^T x + c$$

convex optimization model ها را می توانیم بسیار efficient، حل کنیم.

convex ← توابع کاسه‌ای شکل هستند *

* تابع آفین، یک تابع خطی است که با یک عدد ثابت، جمع شده است. ولی اگر عدد ثابت نداشته باشیم، به آن تابع خطی گویند.

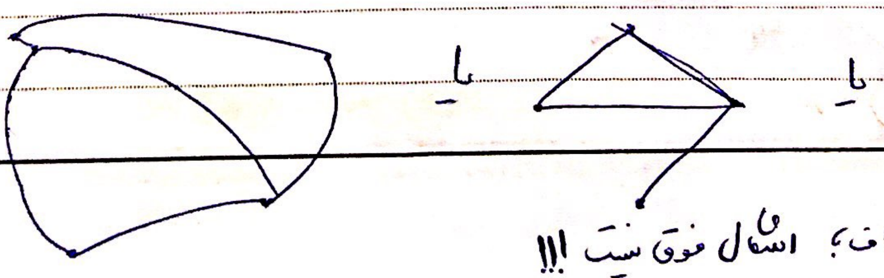
← شرط لازم و کافی برای آن که یک تابع کوادراتیک، convex باشد، آن است که: $(Q \succeq 0)$

← و شرط لازم و کافی برای $Q \succeq 0$ باشد، آن است که تمام مقادیر ویژه‌ی آن بزرگتر یا مساوی صفر باشد *

* جلسه سوم:

* گراف تئوری:

- "یک نشاندن از گراف در یک صفحه‌ی مسطح"، یا همان "نمایش گراف"، عبارت است از:



⇐ تعریف گراف، اشکال فوق نیست !!!

* در ریاضیات ، چینش‌هایی مختلفی داریم ، به عنوان مثال :

R ← میدان اعداد حقیقی (× و + و ≤)

VS (Vector space) ← فضاهای برداری (. و + و ≤ و ×)

(ضرب داخلی)

PS (probability space) ← فضای احتمالاتی ، که یک سه‌تایی شامل (P و Γ و Ω) دارد *

GT ← گراف تئوری نیز یک چینشی در ریاضیات است که با یک مجموعه متناهی تعریف می‌شود

(البته می‌تواند مجموعه‌ی نامتناهی ولی شمارا هم باشد)

- مجموعه‌ی مورد نظر همان مجموعه‌ی رئوس است.

گراف ساده (بدون جهت) (V, E) $E \subseteq \{\{u, v\} , u, v \in V : u \neq v\}$

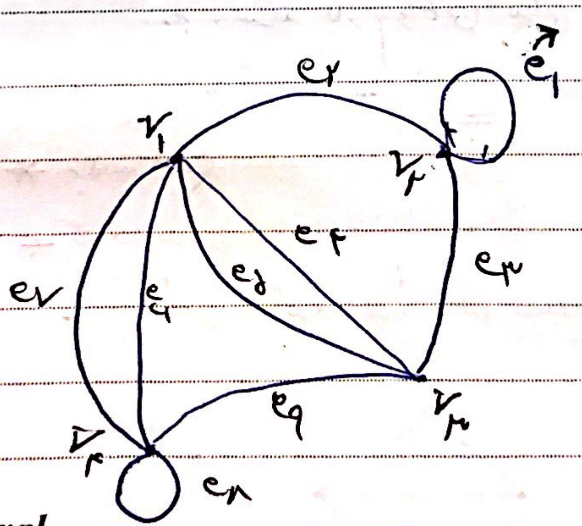
مجموعه رئوس ← V ، مجموعه‌ی یال‌ها ← E

گراف جهت‌دار (V, E) $E \subseteq V \times V$

← پس به طور خیلی ساده ، گراف یک دوتایی در ریاضیات است *

← در ریاضیات تکرار یک حرف معنا ندارد ، بنابراین طوقه را نمی‌توانیم $\{u, u\}$ بنویسیم!!

← برای تعریف گراف‌های پیشرفته‌تر ، باید یک تابعی تعریف کنیم که اعضای E را به V مرتبط سازد *



$\Psi E \rightarrow \{\{u, v\} , u, v \in V\}$

* برای گراف های جهت دار:

ψ^+ : رأسی که یال به آن وارد می‌گردد:

ψ^- : رأسی که یال از آن خارج می‌شود

$\psi^+ , \psi^- : E \to V$

مثال شکل قبل

$\psi(e_1) = \{v_2\}$

$\psi(e_2) = \{v_1, v_2\}$

$\psi(e_4) = \psi(e_3) = \{v_1, v_3\}$

— حال می‌توانیم به گراف، انواع وزن های یالی و وزن های رأسی را اضافه کنیم:

$(V_1, E_1, \psi_1, \underset{E\to R}{w_e^1}, \underset{E\to R}{w_e^2}, \underset{V\to R}{w_v})$

— چون بسیاری از مسائل بهینه سازی گسسته را می‌توان با گراف نمایش داد و حل کرد
بحث گراف تئوری را خواهیم خواند *

* Planner embeding نوعی گراف مسطح است که هیچ یک از یال های آن یکدیگر را قطع نکنند.

* گراف Planner:
گرافی که برای آن حداقل یک نشاندن در بعدی مسطح وجود داشته باشد.

* Label و رنگ: نوعی وزن دادن است که به جای مجموعه ی R، به مجموعه ی ~~...~~ نسبی می‌روند

Maral

*** ماتریس مجاورت :**

- برای گرافی با n رأس ، یک ماتریس $n \times n$ در نظر گرفته می شود و برای هر رأس به رأس دیگر با یک عدد ارتباط برقرار می کند*

*** ماتریس وقوع (M) :**

در این ماتریس ، ردیف ها رئوس و ستون ها ، یال ها هستند . به عبارتی به خود یال ها هویت می بخشد*

یال e_1 از v_1 خارج شده است.

$$\begin{array}{c|cccc} & e_1 & e_2 & \cdots & e_n \\ \hline v_1 & +1 & 2 & & \\ v_2 & & & & \\ \vdots & & & & \\ v_n & & & & \end{array}$$

- اگر ماتریس مجاورت را به توان k برسانیم ، تعداد گشت ها (walk) به طول k به دست می آید*

*** درجه ی رأس ($deg(v)$) :**

- تعداد یال هایی که به رأس متصل شده اند.

$deg(v) \longrightarrow$ برای گراف های بدون جهت

برای گراف های جهت دار:

$deg^+(v) \longrightarrow$ تعداد یال هایی که از رأس v خارج می شود

$deg^-(v) \longrightarrow$ " " " وارد " " "

Δ : بیشترین درجه‌ی رأسی } هرکدام (+) و (−) هم در گراف‌های جهت‌دار دارند*
δ : کمترین درجه‌ی رأسی

* در گراف‌های ساده‌ی بدون جهت، همواره داریم:

$$\sum_{v \in V} dg(v) = 2|E|$$

واز رابطه‌ی بالا، متوجه می‌شویم همواره تعداد رئوس با درجه‌ی فرد، زوج است.

* در گراف ساده همواره:

$$0 \leq dg(v) \leq |V| - 1$$

* گراف‌های کامل:

- یک گراف کامل n رأسی، به گرافی می‌گویند که مجموعه‌ی رئوس آن، n تا باشد و تمام یال‌های ممکن، در آن رسم شده باشد. (گراف هم ساده است)

* گراف‌های ۲ بخشی کامل:

- ۲ مجموعه‌ی رئوس دارد که بین رئوس آن دو مجموعه هیچ یالی نیست، ولی هر مجموعه تعداد یال بیشینه کامل می‌باشد*

* گراف ۲ بخشی:

گرافی است که رئوس آن را بتوان به دو دسته غیرتهی، تقسیم کرد و بین رئوس ۲ مجموعه هیچ یالی وجود نداشته باشد

درخت: گراف همبندی که هیچ دوری نداشته باشد*
جنگل: گرافی که هیچ دوری نداشته باشد*

Scanned by CamScanner

گراف K - منتظم:

گرافی است که درجه ی تمام رئوس آن K می باشد.

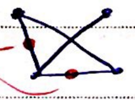
*** گراف مکمل ساده:**

- گرافی است که مجموعه ی رئوس به عنوان مجموعه رئوس گراف اولیه است ولی یال های آن یال های گراف اولیه ی ۲ گراف را باهم کامل می کنند *

*** گراف زیر تقسیم:**

- گراف اولیه داریم که روی یال های آن به تعدادی node (راس) اضافه می کنیم

رئوس اضافه شده



*** گراف فراگیر:**

یه زیرگراف گرافی که مجموعه ی رئوس همان است ولی تعداد یال هایش کمتر می شود *

*** گراف الحاقی:**

یک قسمت مشخصی از گراف به شامل تعدادی راس و یال را بدست می دهیم *

*** گراف Merged:**

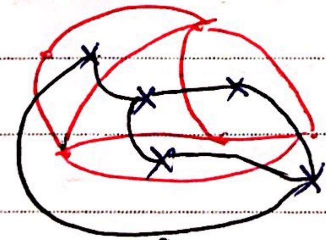
- گرافی که از یکی کردن چند راس گراف اولیه ایجاد شده است *

*** گراف های Dual:**

- گرافی است که از وصل کردن ناحیه های مجاور در یک گراف موجود به دست می آید.

Face: هر ناحیه ای که توسط تعدادی یال محصور شده باشد.

External Face: ناحیه ی بیرونی



External face

گراف Dual

* جلسه‌ی چهارم:

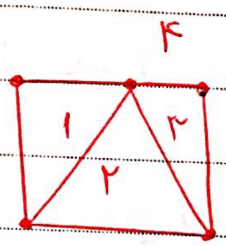
* قضیه‌ای در ارتباط با گراف‌های مسطح (اویلر):

* وجه کاملاً به نشان‌دادن گراف در صفحه مرتبط می‌شود.

f = تعداد face ها

e = تعداد یال‌ها

v = تعداد رأس‌ها



$+f - e + v$

$۴ - ۷ + ۵ = ۲$

$\Rightarrow$ ۲ = تعداد یال‌ها − تعداد رئوس + تعداد وجه‌ها

* این قضیه با استقراء ریاضی اثبات می‌شود*

* دنباله‌هایی از رأس‌ها و یال‌ها:

- در این جا با دنباله‌های یکی در میان سروکار داریم؛ یعنی پس از هر رأس به یال و پس از هر یال به یک رأس می‌رسیم.

$v_1 e_1 v_2 e_2 \dots e_{n-1} v_n$

- به هر یک از توالی‌های فوق، یک گشت (walk) گویند*

- بسته به این که یک رأس چه تعداد بار تکرار شده یا یک یال چه تعداد تکرار شده و یا رأس اول در انتها نیز تکرار شده است یا خیر، گشت‌ها را (دنباله‌ها) دسته‌بندی می‌کنیم*

Subject:
Yare: Month: Day: Page: ( )

* مسیر (Path):

← اگر در یک گشت، هیچ رأسی تکرار نشود، به آن مسیر می‌گوییم *

* گذر:

← گشتی که هیچ یک از یال‌های آن دوبار تکرار نشود را گذر گوییم *

* گذر بسته:

گذری است که رأس ابتدایی و انتهای آن یکسان باشد.

* دور:

یک گذر بسته‌ای که رأس‌های آن هم تکرار نشده‌اند

← وقتی بگوییم رأس‌ها تکرار نمی‌شوند، قطعاً یال‌ها نیز تکرار نمی‌شوند ولی عکس این موضوع برقرار نیست.

* سیر (Tour):

گشت بسته‌ای که از هر یال G، حداقل یکبار عبور کند *

* سیر اویلری (گذر بسته‌ی اویلری):

گشت بسته‌ای که از هر یال G، دقیقاً یکبار عبور کند *

* گذر اویلری:

اگر گذری داشته باشیم که بسته نباشد ولی همه‌ی یال‌ها را پوشش دهد *

* دور همیلتونی:

دوری که تمام رأس‌ها را پوشش دهد *

* گراف همیلتونی:

گرافی که در آن حداقل یک دور همیلتونی وجود داشته باشد *

* گراف اویلری:

گرافی که یک سیر اویلری در آن وجود داشته باشد *

★ گراف همبند:

← گرافی که بین هر دو رأس آن حداقل یک مسیر وجود داشته باشد

★ جنگل:

گرافی که هیچ دوری در آن نداشته باشد، جنگل گفته خواهد شد

★ درخت:

جنگلی که همبند باشد، درخت نامیده می‌شود★

★ قضیه:

یک گراف اویلری است اگر و فقط اگر درجه‌ی تمام رئوس آن زوج باشد★

★ قضیه:

یک گراف دارای گذر اویلری است اگر و فقط اگر دقیقاً دارای دو رأس درجه‌ی فرد باشد★ (همبند نیز باید باشد)

★ گراف k اویلری:

← اگر بتوانیم با k تا گذر، تمام آن را پوشش دهیم، گراف را k اویلری گوییم★

★ قضیه:

- یک گراف k- اویلری است اگر و فقط اگر دارای دقیقاً 2k رأس با درجه‌ی فرد باشد★

← براساس شرط لازم می‌توان حذف کرد، براساس شرط کافی می‌توان اثبات کرد و بر اساس شرط لازم و کافی می‌توان یک رابطه‌ی دوتایی برقرار کرد

★ کمر گراف:

- کوچکترین دوری که در یک گراف وجود دارد را "کمر گراف" گویند★

- قطعاً کمر گراف عدد ۱ و ۲ را نمی‌تواند بگیرد و در گراف های ساده، حداقل این دور ۳ می‌باشد★

★ قطر گراف:

بزرگترین دور یک گراف را قطر گراف گویند★

Subject:
Yare: Month: Day: Page: ( )

← با پیدایش این مباحث max و min و ... و ای دیچینن اعدادی در گراف ها، آرام آرام به مبحث optimization نزدیک می شویم ← بنابراین خود قطری گراف ها، یک سری مسئلی دارد که ذاتاً درمورد بهینه سازی است ★ (حتی اگر اصلاً با مهندسی صنایع سروکار نداشته باشد)

★ رنگ آمیزی رأسی:

← اگر رأس ها را برچسب رنگی بزنیم به آن رنگ آمیزی رأسی گویند★

- رنگ آمیزی صحیح:

- اگر هر دو رأس مجاور، دارای رنگ یکسان نباشد★

★ رنگ آمیزی یالی صحیح:

هر دو یالی که هم رس هستند، رنگ یکسان نداشته باشد.

← در این جا "تعداد رنگ های کمینه" اهمیت پیدا می کند. که به این تعداد، عدد رنگی گراف گفته★

★ مؤلفه (component):

بزرگترین زیرگراف همبند در یک گراف غیر همبند را مؤلفه گویند★ (زیرگراف همبند بیشینه)

maximal ← یعنی در خاصیت مورد نظر ما، بزرگترین باشد.

← عناصر روی قطر ماتریس مجاورت به توان ۲، برابر با درجه هر رأس می باشد زیرا تعداد گشت ها به طول ۲ را نشان می دهد

★ زنجیره (chain):

~~دنباله های جهت دار به این مبحث مطرح می شود که در این صورت الزاماً دیگر~~ ممکن است رأس اول نباشد.

$V_1 e_1 V_2 e_2 \ldots$

Maral

Subject:
Yare: Month: Day: Page: ( )

- فصل ۱، ۲، ۳ و ۴ کتاب باندی - موری پایان یافت*

Maximal planner SubGraph* :

زیرگرافی را در یک گراف G در دست داریم که بیشینه باشد؛ یعنی اگر فقط یک یال دیگر به آن اضافه کنیم خاصیت مسطح بودن آن از بین می رود*

* جلسه پنجم: ۱۰.۵

* برش یالی:

- مجموعه ای از یال ها که اگر آن ها را از گراف برداریم، رئوس گراف G دو دسته ی غیر تهی تقسیم شده باشد. (به عبارتی مجموعه رئوس را به دو دسته افراز می کنیم و سپس e یال های بین آن دو را برمی داریم)

← به عبارتی دو دسته رئوس ایجاد می شود که اصلاً با هم ارتباط ندارند!

* یال برشی:

یالی است که اگر آن را برداریم، تعداد بندهای گراف اضافه خواهد شد*

* شرط لازم و کافی برای آن که یال e برشی باشد:

یال e برشی است اگر و فقط اگر یال e در هیچ دوری واقع نشده باشد*

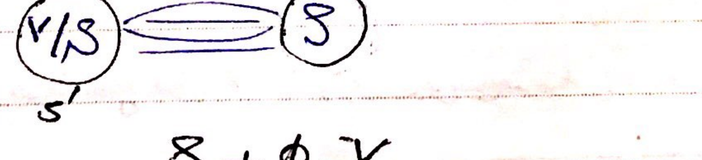
درخت فراگیر ← زیرگرافی از G که درخت است و فراگیر هم می باشد*

← اگر یک یال به درخت فراگیر اضافه شود، دور ایجاد خواهد شد*

* بند:

← برش اگر Minimal باشد، به آن بند گوییم*

(یک سری ویژگی هایی که اگر اضافه یا کم شوند، آن ویژگی از بین می رود)



Maral $S \neq \phi$ و V

$E(S, S')$

← در «برش یالی» لزوماً بندها اضافه نخواهد شد!! ولی در تعریف یال برشی داریم که حتماً می‌بایست تعداد بندها اضافه شود*

← نتیجه‌ی برش یالی، آن است که در گراف G، دو مجموعه‌ی رئوس S و S' را جدا می‌کند.

← جنگل Maximal:

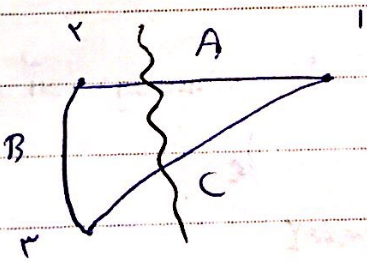
در این نوع جنگل ← اگر یک یال دیگر به جنگل اضافه شود، دور در آن ایجاد می‌شود*

* پس منظور از maximal و minimal، بیشینه بودن یا کمینه بودن وزن‌ها و هزینه‌ها و سود و... نیست!! ← (بلکه طبق تعریف فوق، وقتی می‌گوییم یک ویژگی maximal است که اگر کمی تعریف آن اضافه یا کم شود، آن ویژگی به خصوص از بین می‌رود.)

minimal ← کمین
minimum ← کمینه

← در گراف تئوری، وقتی از کمینه بودن حرف می‌زنیم، بیشتر اوقات منظور کمترین تعداد یال است ولی در Application (کاربرد)، منظور از کمینه یا بیشینه بودن، کمترین یا بیشترین وزن یال‌هاست * ← اگر وزن یال‌ها، ۱ باشد، تعداد یال‌ها در نظر گرفته می‌شود*

ارتباط بین وزن دار و غیروزن دار



$S = \{1\}$

$S' = \{1 , 2\}$

مهم آن است که بین S و S' هیچ گونه ارتباطی وجود نداشته باشد.

$E(S, S') = \{A, C\}$

← با تغییر یال برشی، S و S' عوض می‌شوند*

s_t cut :

- یعنی طوری برش بزنیم که دو رأس مشخص s و t ، هیچ گونه ارتباطی با یکدیگر نداشته باشند.

* رأس برشی :

رأسی که یال ها را به ۲ دسته تقسیم می‌کند به طوری که این یال ها فقط در رأس v ، با یکدیگر در ارتباط اند *

- اگر این رأس را برداریم ، تعداد مؤلفه های گراف اضافه می‌شود *

قضیه :

در هر درخت ، یک رأس ، برشی است ، اگر و فقط اگر ، درجه ی آن رأس از یک بزرگتر باشد.

$\deg(v) > 1$

* در هر گراف همبند ساده ، حداقل ۲ رأس غیر برشی وجود دارد *

← همبند ، حتماً یک درخت فراگیر دارد *

* جورسازی (Matching) :

مجموعه ای از یال ها ، که هیچ ۲ تایی رأس مشترک نداشته باشند *

← در جورسازی ، چون اگر یک یال کم کنیم باز هم جورسازی باقی می‌ماند ، minimal معنا ندارد ولی اگر به آن یک یال اضافه کنیم ، چون احتمال آن که یال ها هم رأس شوند ، وجود دارد ، maximal معنا می‌دهد * ← لذا Maximal Matching ، یک جورسازی است که اگر فقط یک یال به آن اضافه کنیم ، رأس مشترک ایجاد خواهد شد *

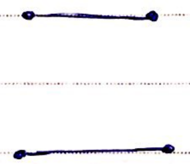
* Perfect Matching (جورسازی کامل) :

- یک نوع جورسازی است که در آن تمام رأس ها ، استفاده شده باشند *

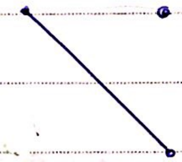
Min mum perfect Matching → مسئله ی تخصیص

Subject:
Yare: Month: Day: Page: ( )

← Maximum Matching یک نوع جورسازی است که تعداد یال های آن بیشینه باشد.



$\underbrace{\text{maximum matching}}_{\text{Perfect}}$



Maximal matching

* اگر یک راس اضافه شود دیگر perfect نیست*

← مثلاً Minimum matching معنا نمی دهد، چون هیچ یالی انتخاب نمی شود!!! ولی Minimum Perfect matching معنا می دهد

Minimum Perfect matching ⟶ مسئله تخصیص متناظر

* مسئله shortest path problem در زمان چندجمله‌ای حل می شود اگر و فقط اگر تمام وزن یال های آن، مثبت باشد.

* پوشش یالی:

مجموعه‌ای از یال ها که هر راس حداقل یک بار توسط آن ها پوشش داده شده باشد*

* پوشش راسی:

مجموعه‌ای از راس که هر یال حداقل یک طرف آن در مجموعه راس باشد*

* Perfect matching به زبان دیگر:

جورسازی که پوشش یالی هم باشد.

Maral ~~* عدد پوشش یالی: کمترین تعداد یالی که با آن می توان همه راس ها را پوشش داد*~~