Elementary path ← رأس‌ها تکرار نشود *

Simple path ← یال‌ها تکرار نشود *

* اثبات معادل بودن ۲ محدودیت حذف زیر تور:

$$\sum_{e\in\delta(X)} x_e \geq 2 \Longrightarrow \underbrace{\sum_{e\in\delta(X)} x_e}_{\text{یال‌ها خروجی مجموعه X}} = \underbrace{\sum_{v\in X}\sum_{e\in\delta(v)} x_e}_{\text{یال های خروجی همه رأس‌های X}} - 2\underbrace{\sum_{e\in E(G(X))} x_e}_{\text{یال‌هایی خروجی که درون خود X هستند}}$$

به دلیل محدودیت اول ← ۲=

$$= \sum_{v\in X} 2 - 2\sum_{e\in E(G(X))} x_e = 2\Big(|X| - \sum_{e\in E(G(X))} x_e\Big) \geq 2$$

$$\Longrightarrow \boxed{\sum_{e\in E(G(X))} x_e \leq |X| - 1}$$

← چند وجهی مرتبط با مسئله‌ی TSP، Integral نیست *

* مثلاً در مسئله‌ی حمل و نقل، اگر عرضه‌ها و تقاضاها عدد صحیح باشد، چند وجهی متناظر با آن Integral خواهد بود و مطمئن خواهیم بود با حل به روش سیمپلکس، حتماً جواب بهینه‌ی حاصل عدد صحیح خواهد شد و بهینه‌ی کلی نیز خواهد بود *

* جلسه‌ی دوازدهم:

* حل مسئله‌ی Minimum Perfect Matching :

$$\text{Min} \sum c_e x_e$$

$$\text{s.t.} \quad \sum_{e \in \delta(v)} x_e \le 1 \;\; (=) \quad \forall v \in V \longrightarrow$$

$$x_e \in \{0,1\}$$

کلاً برای مسئله‌ی Matching این محدودیت صادق است ولی در این مسئله چون Min سازی است می‌تواند مساوی 1 نیز باشد، ولی در Max perfect matching باید کوچکتر مساوی 1 باشد

* Maximum perfect matching:

$$\text{Max} \sum c_e x_e$$

$$\text{s.t.} \quad \sum_{e \in \delta(v)} x_e \le 1$$

$$x_e \in \{0,1\}$$

* در مسئله‌ی MST، خود محدودیت‌ها برای Integral شدن چندوجهی کفایت می‌کند ولی در دو مدل فوق، می‌خواهیم به مدل‌های یک Valid inequality به گونه‌ای اضافه کنیم که هم فرمولاسیون قوی‌تر شود و هم چندوجهی حاصل Integral شود.

* مسئله‌ی MST یکی از ساده‌ترین مدل‌های ترکیباتی است (از کلاس P است) ولی تعداد محدودیت‌هایی که باعث می‌شود چندوجهی آن Integral شود، بسیار زیاد است. ←

$$\sum_{e \in E(G(M))} x_e \le |M| - 1$$

← همان Facet defining است

← ولی در مسئله‌ی TSP، هنوز که هنوز است نتوانسته‌ایم بگوییم با اضافه کردن کدام یک از Valid inequality ها، می‌توانیم چندوجهی آن را Integral کنیم. به عبارتی هنوز نتوانسته‌ایم مجموعه‌ی Valid inequality ها را برای TSP پیدا کنیم که چندوجهی آن را Integral کند. (ولی این مجموعه‌ی Valid inequality ها برای مسائل Min perfect matching و Max perfect matching و MST، پیدا شده است)

* Minimum Spanning Tree problem (MST):

$$\min \sum c_e x_e$$

① s.t. $\sum_{e\in E} x_e = n-1$

② $\sum_{e\in E(G(x))} x_e \le |x|-1 \; ; \; x\subset V$

$x_e\in\{0,1\}$

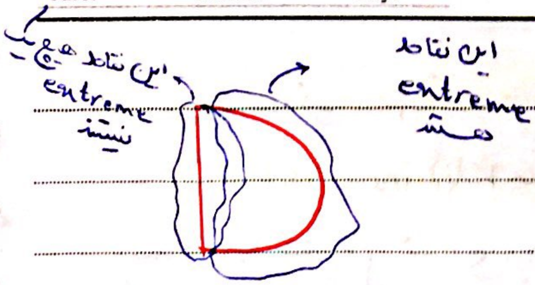
② * MST با مدل بندی فوق، کاملاً چندوجهی Integral دارد و محدودیت شماره‌ی ۲ سبب این اتفاق شده است. (البته تعداد این محدودیت بسیار زیاد است!!!)

* مسئله‌ی MST را ابتدا در حالت ریلکس شده (بدون محدودیت ۲) حل می‌کنیم، سپس می‌بینیم آیا جواب‌های به دست آمده، این محدودیت‌ها را نقض می‌کند؟ هر یک از محدودیت‌ها که نقض شده باشد در مرحله‌ی بعدی به مدل اضافه می‌شود *

* نقطه‌ی گوشه‌ای (extreme point):

~~نقطه‌ایست که نتوانیم آن را از ترکیب خطی دو نقطه‌ی دیگر به دست آوریم *~~

(نقطه‌ای از S که نتوان آن را به صورت ترکیب خطی دو نقطه‌ی دیگر از S نوشت *)



* نقاط بحرانی:

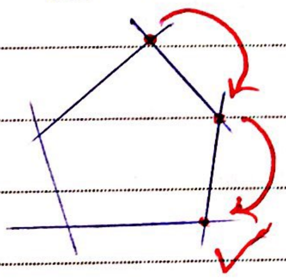
- مقاطعی که مشتق تابع در آن ها صفر است. $\Leftarrow$ $\partial f(x)=0$

* علت اهمیت Entreme point ها در مجموعه های محدب:

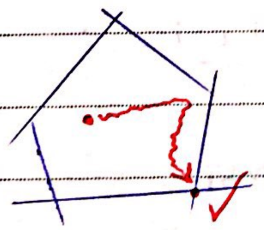
- در جواب های بهینه، حداقل یکی از نقاط، entreme point هستند *

* در روش های حل برنامه ریزی ریاضی، ۲ دسته کلی داریم:

۱- روش هایی که فقط نقاط گوشه ای را بررسی می کنند، مانند روش سیمپلکس:



۲- روش هایی که از نقاط درونی برای رسیدن به جواب بهینه استفاده می کنند مانند روش کارماکار:



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

* اضافه کردن valid inequality به Min perfect Matching و Mean perfect Matching:

O = زیر مجموعه هایی از V که تعداد رئوس آن فرد باشد*

$$O=\{T\subseteq V : 2 \nmid |T|\}$$

*Min PMP:

$$\min \sum_{e\in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{e\in\delta(v)} x_e = 1 \qquad ; v\in V$$

valid inequality یک است که اجباری نیست ولی حل را بهبود می دهد

$$\sum_{e\in\delta(X)} x_e \geq 1 \qquad X\in O,\ |X|>1$$

$$x_e\in\{0,1\}$$

← چون تطابق perfect است، $\sum x_e \geq 1$ برای تعداد رئوس فرد صادق است ولی اگر تعداد رئوس، زوج باشد ممکن است $\sum x_e=0$ نیز بشود. ولی چون تعداد رئوس را فرد و تطابق را نیز perfect درنظر گرفتیم، حتما $\sum x_e$ باید بزرگتر مساوی یک شود* (اگر تطابق، perfect نبود، ممکن بود: $\sum x_e=0$ شود)

← valid inequality حالت Mean را می توانیم در اینجا هم به کار ببریم ولی ~~چون از طریق جمع جبری همین دو محدودیت نیز می توانیم آن را به~~ دست آوریم، دیگر آن را اضافه نخواهیم کرد*

Maral

(۲۱)

* Max PMP :

$$\text{Max} \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \quad \sum_{e \in \delta(v)} x_e \leq 1 \quad ; v \in V$$

$$\sum_{e \in E(G(X))} x_e \leq \frac{|X|-1}{2} \quad ; X \in O, \; |X| > 1$$

$$x_e \in \{0,1\}$$

← این valid inequality می‌گوید که در یک perfect matching در یک گراف با
تعداد رئوس فرد (مثلاً ۷ تا) نهایتاً ۳ تا (نصف رو به پایین) یال
می‌توانیم داشته باشیم *

← طبیعتاً این valid inequality را می‌توانیم در حالت Min هم به کار ببریم
ولی VI های حالت Min را نمی‌توان برای حالت Max به کار برد *

* اثبات این که جواب بهینه این دو مدل Min pmp و Max pmp با اضافه کردن
این valid inequality ها integral می‌شود در کتاب Korte هست *

* ۴ فرمولاسیون TSP از صفحه ۱۴ پی دی اف فرستاده شده خوانده شود
→ Ip Modeling

* جلسه سیزدهم :

* ما ترجیح می‌دهیم مسائل TSP را با روش DFJ حل کنیم، زیرا با روش Branch and cut راحت‌تر حل می‌شود *

← LB حاصل از DFJ از SCF و SCF نیز از MTZ بهتر است.

Optimal TSP

* LB(MCF) = LB(DFJ)
$O(n^2)$ $O(n^3)$

* LB(SCF)
$O(n^2)$

* LB(MTZ)
$O(n^2)$

سؤال: اثبات کنید LB مدل‌سازی DFJ از مدل‌سازی MTZ بهتر است *

← حال با اضافه کردن Valid inequality ها، سعی داریم تا باز هم مدل‌های DFJ، MTZ، MCF و SCF را بهبود ببخشیم ولی تعداد این VI ها زیاد است
باید با مطرح کردن ایده‌های مختلف آن‌ها را هم کار ببریم *

* فقط مطرح کردن Valid inequality مهم نیست، بلکه باید بگوییم هر یک از آن‌ها چگونه استفاده خواهند شد *

← معمولاً اگر درجه سختی VI ها از نوع $O(2^n)$ یا $O(n!)$ و ... باشد، ابتدا مسئله را
در حالت ریلکس شده و بدون آن VI ها حل می‌کنیم و می‌بینیم کدام یک ...

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

آن‌ها نقض شده اند ← پس آن‌ها را اضافه می‌کنیم. (گاهی اوقات قبل از اضافه کردن آن‌ها تقویت نیز می‌کنیم ← Strong valid inequality)

← اگر این محدودیت های Sepration جزء محدودیت های الزامی ما نیز باشد باید کاملا قادر باشیم آن‌ها را خیلی سریع و Exact حل کنیم و آن‌هایی که نقض شده اند را پیدا کنیم*

* محدودیت های معتبر شانه ای برای مسائل TSP (در حالت متقارن):

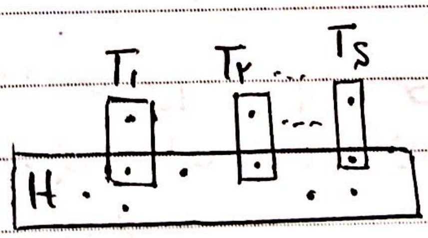
- محدودیت های معتبر برخلاف محدودیت های الزامی به صورت نباید وجود داشته باشد*

مجاز لازم: (هر H و T_1 تا T_S که خصوصیات زیر را داشته باشند)

① $S \geq 3$ عددی فرد

② $T_1, \dots, T_S \subset V(G)$
(دو به دو مجزا)

③ $H \subset V(G)$: $T_i \cap H \neq \emptyset \quad \forall i = 1, \dots, S$
$T_i \setminus H \neq \emptyset \quad \forall i = 1, \dots, S$
$H \setminus T_i \neq \emptyset \quad \forall i = 1, \dots, S$



* هر مجموعه ای با شرایط فوق را داشته باشیم، براساس آن می‌توانیم یک محدودیت شانه ای بنویسیم*

* به عبارتی ۳ شرط فوق، ورودی های محدودیت شانه ای هستند*

Maral

$$\sum_{e\in\delta(H)} x_e + \sum_{i=1}^{s}\sum_{e\in\delta(T_i)} x_e \geq 3s+1$$

یا III

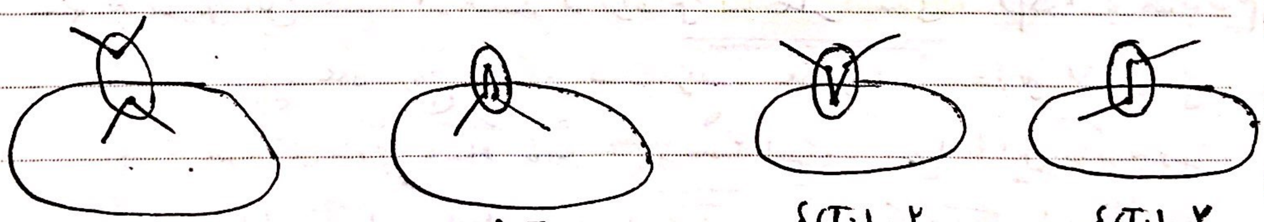
$$\sum_{e\in E(G(H))} x_e + \sum_{i=1}^{s}\sum_{e\in E(G(T_i))} x_e \leq |H| + \sum_{i=1}^{s}|T_i| - \frac{3s+1}{2}$$

← از محدودیت بالایی با استفاده از اتحادهای گفته شده در صفحه‌ی ۲۰ می‌توانیم محدودیت پایینی را به دست آوریم و بالعکس *

* اثبات محدودیت بالایی:

$\xrightarrow[\text{داریم}]{\forall i=1,\dots}$ $\sum_{e\in\delta(T_i)} x_e + \sum_{e\in\delta(H)\cap E(G(T_i))} x_e \geq 3$

برخی از حالات مختلف ممکن عبارت است از: (براساس ورود و خروج ها دسته‌بندی کرده‌ایم)



$\delta(T_i)=4$, $\delta(H)\cap E(G(T_i))=0$

$\delta(T_i)=2$, $\delta(H)\cap E(G(T_i))=2$

$\delta(T_i)=2$, $\delta(H)\cap E(G(T_i))=2$

$\delta(T_i)=2$, $\delta(H)\cap E(G(T_i))=1$

* همان طور که مشخص است، تمامی جمع‌های فوق از ۳ بزرگتر مساوی است *

این عبارت شامل یال‌های H در درخت‌هاست که نقطه‌ای از یک
~~یال‌های H کمتر خواهد بود، پس با جایگذاری آن به جای هم~~
تساوی برقرار خواهد بود .

* اول رابطه را بر روی تمام iها جمع می‌زنیم:

$$\sum_{i=1}^{s}\sum_{e\in\delta(H)\cap E(G(T_i))} x_e + \sum_{i=1}^{s}\sum_{e\in\delta(T_i)} x_e \geq 3s$$

این بخش هم که عیناً در محدودیت وجود دارد

$$\sum_{e\in\delta(H)} x_e \geq$$

$$\underbrace{\sum_{e\in\delta(H)} x_e}_{\text{زوج}} + \underbrace{\sum_{i=1}^{s}\sum_{e\in\delta(T_i)} x_e}_{\text{زوج}} \geq 3s$$

- حال می‌دانیم در مسئله‌ی TSP ، خروجی هر مجموعه حتماً زوج است. S فرد است و ۳ ضرب در S نیز فرد باقی خواهد ماند ← پس قطعاً مجموع ۲ عدد زوج در محدودیت فوق بزرگتر از 3S و بزرگتر مساوی (3S+1) خواهد بود*

$$\xrightarrow{\text{پس}} \boxed{\sum_{e\in\delta(H)} x_e + \sum_{i=1}^{s}\sum_{e\in\delta(T_i)} x_e \geq 3s+1}$$

← به طور کلی این قسمت از مسئله که براساس ساختار گسستی TSP ، متوجه شدیم که سمت چپ عدد زوج است و سمت راست عدد فرد است و باید از حداقل یک به علاوه آن بزرگتر مساوی باشد ، ارزشمند است ، زیرا تا اینجا که به دست آوردیم با استفاده از روابط جبری بوده است و ارزش زیادی ندارند!!!

* اگر S زوج بود ، آن گاه چنین استدلالی نمی‌توانستیم داشته باشیم *

Maral

* محدودیت های معتبر دوجورسازی (مرتبط با محدودیت های ستاره ای) :

① $H \subset V$
② $F \subset \delta(H) \longrightarrow$ * F زیر مجموعه ای از یال های خروجی H است.
③ $|F| =$ فرد

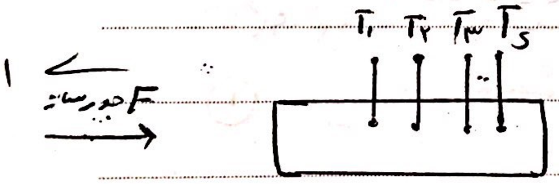
- بنابراین برای هر F که تعداد اعضای آن فرد باشد داریم:

$$\sum_{e\in E(G(H))} x_e + \sum_{e\in F} x_e \leq |H| + \frac{|F|-1}{2}$$

- اثبات رابطه ی فوق برای F هایی که جورسازی هستند (یعنی هیچ رأسی از آن بر روی ۲ یال قرار نگرفته باشد): (برای آن از محدودیت ستاره ای دوم گفته شده استفاده می کنیم)

$$\text{———} \leq |H| + \sum_{i=1}^{s} |T_i| - \frac{3s+1}{2}$$

* حال وقتی F جورسازی باشد؛ هر i از 1 تا s خودش یک یال F می باشد
یعنی: $|F| = S$



$$\text{———} \leq |H| + \underbrace{\sum_{i=1}^{s} \overbrace{|T_i|}^{\text{مجموعه رئوس}}}_{} - \frac{3s+1}{2}$$

$$\text{———} \leq |H| + \underbrace{2}_{\text{چون هر یال ۲ رأس دارد}}\underbrace{|F|}_{\text{مجموعه ی یال ها}} - \frac{3|F|+1}{2}$$

$$\Rightarrow \text{———} \leq |H| + \frac{|F|-1}{2}$$

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

* با اضافه کردن Valid inequality ها به ۲ حالت ممکن است پیش بیاید:

۱- یا فضای جواب کاملاً Integral می شود که در این صورت می توانیم محدودیت Integrality را کلاً حذف نماییم. ($x \in Z$)

- برای این حالت باید از Cutting plane Method ها استفاده کنیم.

- حال یا فقط Valid inequality هارا وارد می کنیم که نقض شده اند و یا تمام Valid inequality های ساده را وارد خواهیم کرد.

- حال اگر از Seperation Sup-problem هم بتوانیم نیز را داشته باشیم و تابع تشخیص باشد در صورتی که:

* Integrality Gap = ۰ ⟸ حتماً باید محدودیت های نقض شده را وارد کنیم چون فضای چند وجهی (چند وجهی رأس ها Integral است) با وجود رأس ها Integral شده است و اگر یکی از آن ها نباشد ممکن فضای جواب غیر Integral شود *

* Integrality Gap ≠ ۰ ⟸ نمی توانیم محدودیت های نقض شده را نادیده بگیریم *

* مثلاً فرض کنیم مسئله زیر را بدون در نظر گرفتن محدودیت سوم حل کرده ایم و یک x^* به دست آمده است: (محدودیت ۴ نیز ۳ و $0 < x < 1$ به تعبیر پذیرفته است)

$$MST \Rightarrow \text{Min} \sum_{e \in E} c_e x_e \quad (1)$$

$$s.t. \quad \sum_{e \in E} x_e = n-1 \quad (2)$$

$$\sum_{e \in E(G(X))} x_e \geq 2 \quad ; \quad X \subset V, \quad 3 \leq |X| \leq \lfloor \tfrac{|V|}{2} \rfloor \quad (3)$$

$$x_e \in \{0,1\} \quad (4)$$

Maral

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

* حال از کجا متوجه بشیم این x^* ها ($\bar{x}$) در تمام محدودیت های شماره ۳ که تعدادشان نیز فوق العاده زیاد است صدق می کند ؟ و همچنین اگر تعدادی از این محدودیت ها نقض می شود چطوری بفهمیم کدام یک از آن ها نقض شده است ؟

* مسئله ی Min CUT :

- پیدا کردن مجموعه ای از رئوس که مجموع وزن یال های خروجی آن، Min شود *

ج: کافیست سمت چپ محدودیت سوم را به عنوان تابع هدف یک مسئله ی Min CUT در نظر بگیریم و آن را حل کنیم ← اگر جواب حاصل از ۲ بزرگتر شد که یعنی تمام x ها از ۲ بزرگتراست و لذا قطعاً محدودیت سوم برقرار است ولی اگر جواب حاصل از ۲ کمتر شد یعنی محدودیت سوم نقض شده است.

* خوشبختانه Min CUT یک مسئله متعلق به کلاس p است. $(O(n^3))$

$$\bar{x}_e \longrightarrow G=(V, E, W)$$

وزن یال ها را x^*_e قرار می دهیم. $\longrightarrow \bar{x}_e$

$$\min_{X\subset V} \sum_{e\in\delta(X)} \bar{x}_e$$

* برنامه ریزی عدد صحیح Min CUT و Mean CUT خوانده شود *

② حالتی که Integrality Gap $\neq 0$:

- در این حالت تمام VI ها را نیز که چک بکنیم برقرار هستند یا خیر، باز هم ممکن است جواب نهایی عدد صحیح نباشد و لذا مجبور خواهیم شد که از شاخه زنی کمک بگیریم و پس دوباره در هر شاخه نیز این فرآیند را بررسی کنیم ← به آن Branch and cut گویند *

* این دسته از روش ها را CUT generation Method یا Constraint generation Method یا cutting plane method هم می نامند.

Maral

(۲۸)

← حل Branch and Bound وقتی که با cut generation method ها ترکیب می‌شود، به آن Branch and cut and Bound گفته می‌شود که دیگر Bound آن ذکر نخواهد شد *

← شاخه و کرانی که در هر Node آن یک cutting plane method حل می‌شود را Branch and cut گویند *

* جلسه چهاردهم:

* cutting plane Method : (cut generation method)

یک سری محدودیت‌ها را بررسی دارند و مدل را حل می‌کنند، پس جواب به دست آمده را در محدودیت‌های ریلکس شده بررسی می‌کنی تا ببینند کدام یک از محدودیت‌ها را نقض می‌کند. اگر هیچ محدودیتی نقض نشود که به جواب بهینه کلی رسیده‌ایم. ولی اگر نقض شود باید بگردیم تا ببینیم کدام یک از محدودیت‌ها نقض شده اند و بعد حداقل یکی از آن‌ها را به مدل وارد می‌کنیم.

← در برش گموری محدودیت $x \in Z$ حذف می‌شود *

* ایده‌ی اصلی روش‌های برش صفحات:

۱- جواب اشتباه قبلی مجدداً ظاهر نشود *

۲- فضای موجه کوچک نشود *

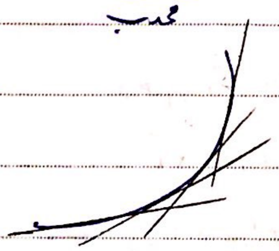
* چه موقعی از Branch and cut استفاده می‌کنیم؟

وقتی یک مسئله در قالب MILP مدل شده است و این مدل یک سری محدودیت مشکل‌ساز دارد که مسئله قابل حل نباشد و حتی پس از آزادسازی نیز نتوانیم حل کنیم از Branch and cut استفاده می‌کنیم *

MILP = Mixed Integer Linear programming

* هر تابع محدب را می‌توان به صورت سوپریمم یک سری از صفحه‌ی مماس بر آن به نوشت :

$$\underbrace{f(x) \le a}_{\text{محدب}} \Longrightarrow \sup_{\theta \in T}(a_\theta x + b_\theta) \le a$$



* صفحه ۱۵۸ کتاب Wolsey ، (18':00")

* مدل تخصیص :

$$\max \sum \sum c_{ij} \cdot x_{ij}$$

$$\text{s.t.} \quad \sum x_{ij} \le 1$$

$$\sum a_{ij} x_{ij} \le b_j$$ → محدودیت مسئله کوله‌پشتی

* CUT Passes = تعداد Itteration هایی که cutting plane method به اجرا (Run) می‌شود

* cutting PM در شرایط عادی زمانی متوقف می‌شوند که تمام محدودیت‌ها برقرار شده باشد.
ولی زمانی می‌توانیم با واسطه روش حل آن را متوقف کنیم که محدودیت‌های نقض شده
جزء valid inequality های غیراجباری باشد. ولی اگر یک محدودیت الزامی
نقض شده باشد می‌بایست حتماً تا انتها cutting plane method را حل کرده و
محدودیت را به سرعت و Exact تشخیص دهیم و به مدل وارد کنیم *

* در cutting plane method، حداقل باید هر بار یک برش اضافه شود ولی می‌توانیم
در یک دور بیش از یک CUT هم پیدا شود که براساس قرارداد، تعیین می‌کنیم که
در هر دور نهایتاً چه تعداد برش اضافه شود *

* روش برش صفحات در مسئلهٔ TSP

در TSP ، محدودیت زیر حذف می‌شود:

$$\sum_{e\in\delta(X)} x_e \geq 2$$

← حال برای بررسی محدودیت فوق، باید یک مسئلهٔ Min Cut را حل کنیم تا ببینیم آیا نقض شده است یا خیر؟!

← حال علاوه بر این محدودیت‌های حذف زیرتور که اجباری هستند، می‌توانیم محدودیت‌های سایر را نیز اضافه کنیم

* حل مسئلهٔ Max Flow معادل حل مسئلهٔ Min S-T Cut است *

← علت اصلی استفاده از Valid inequality ها، بهبود دادن lower bound (LB) می‌باشد. ← حال اگر LB بهبود یابد، زمان حل نیز ممکن است کاهش پیدا کند.

* الگوریتم جدید حل مسئلهٔ TSP: (constraint generation)

← ابتدا محدودیت حذف زیرتور را حذف می‌کنیم ولی مدل باقی‌مانده را با همان شرط $x\in Z$ (یعنی به صورت IP) حل می‌کنیم. (برخلاف روش قبلی که محدودیت $x\in Z$ نیز حذف می‌شد) سپس می‌بینیم که آیا زیرتوری ایجاد شده است؟ مثلاً می‌توانیم جواب را رسم کنیم. (چون قطعاً جواب حاصل، اعداد صحیح خواهد بود) سپس اگر زیرتوری ایجاد شده بود، محدودیت حذف زیرتور مربوطه را اضافه خواهیم کرد.

⟸ مسئلهٔ TSP نامتقارن = مسئلهٔ تخصیص + محدودیت‌های حذف زیرتور

* جلسه‌ی پانزدهم:

* سؤال: در بررسی Branch and Bound، اگر مسئله محدودیت زمانی داشته باشیم و قبل از اتمام زمان، نتوانیم تمام Node ها را بررسی کنیم، جواب به دست آمده را چگونه بفهمیم بهینه است یا خیر؟!!! آیا error bound قابل تعیین است؟

* Profitable TSP:

- در این مسئله هر مشتری، یک profit به ازای بازدید آن خواهد داشت. هدف این مسئله ملاقات کردن مشتریانی است که جمع سود حاصل از بازدید آن‌ها بیشینه شود.
- به عبارتی در این مسئله برخلاف مسئله‌ی TSP معمولی، لازم نیست تمام مشتریان ملاقات شوند*

* Branch and price:

- روش طراحی الگوریتم B&P:
- Branch and Bound فقط مختص Integer programming نیست بلکه کاربردها بسیار متنوعی نیز دارد، از جمله در Flow shop و .... ولی علت آنکه B&B در Ip بسیار پرکاربردتر و موفق‌تر عمل کرده است آن است که LB ها را بسیار سریع محاسبه می‌کند*

* تفاوت الگوریتم و Method:

* الگوریتم، آن چیزی است که دقیقاً برای مسئله‌ی ما ساخته شده و ایجاد گردیده است ولی وقتی از Method حرف می‌زنیم، منظورمان روش کلی حل مسئله می‌باشد. مثلاً ممکن است از Branch and Bound Method برای طراحی یک الگوریتم مختص مسئله‌ی خودمان استفاده کنیم.

← B&B لزوماً برای Ip نیست و B&B هایی هستند که اصلاً با mathematical prog کاری نخواهند داشت.

Subject:

Yare: Month: Day: Page: ( )

* جواب سؤال صفحه‌ی قبل:

* error bound = عددی که مطمئن هستیم خطا از آن بیشتر نخواهد شد.

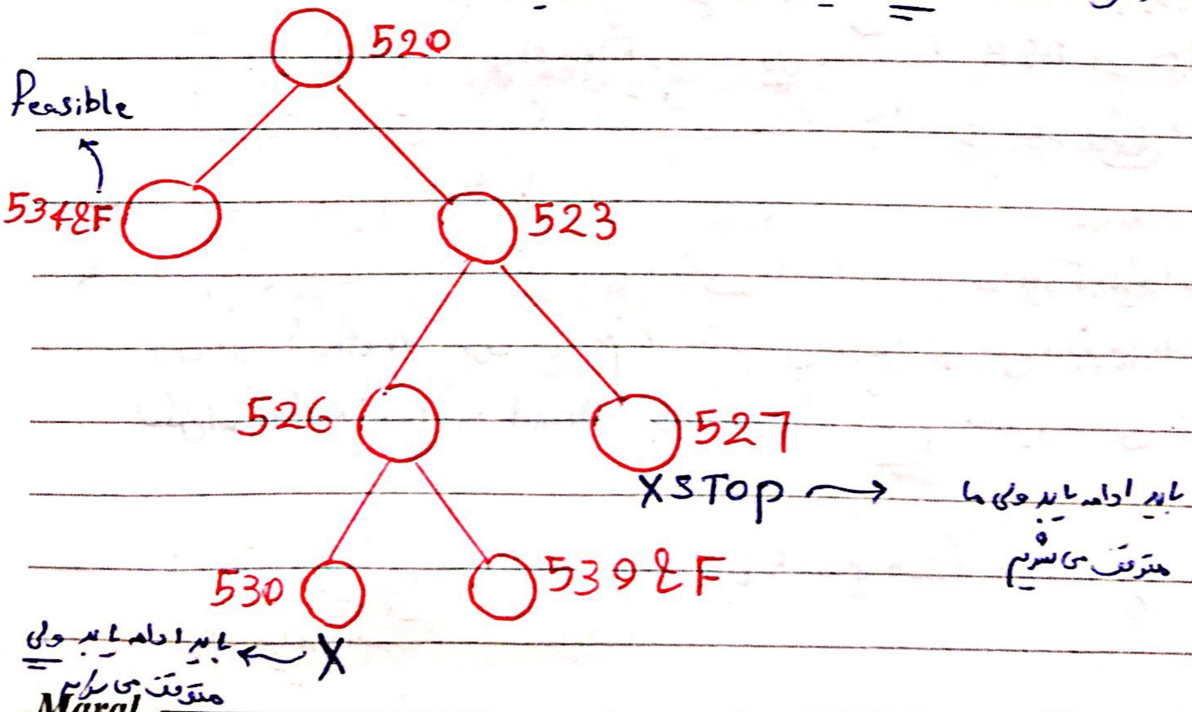
* به معنی STOP کردن الگوریتم B&B ، ۲ حالت پیش می‌آید: (مسئله‌ی Min ساز)

① یا UB به دست نیامده است ← که در این صورت جواب های به دست آمده اصلاً قابل اطمینان نیستند* (چرا که اگر حتی فقط یک گره هم feasible شده باشد، upper bound به دست خواهد آمد*) فقط می‌توانیم مینیمم LB ها را تعیین کنیم و بگوییم قطعاً جواب بهینه، بهتر از آن نخواهد شد*

② و یا یک یا تعدادی UB به دست آمده است، در این صورت باید از بین این حدود بالا، بهترین UB را تشخیص دهیم و تعیین کنیم که متعلق به کدام گره است.

حال LB مربوط به گره‌ی ریشه هم می‌دانیم چون کمترین محدودیت را دارد کمترین LB را نشان می‌دهد یعنی جواب بهینه قطعاً از LB گره‌ی ریشه (Parent) کمتر نخواهد شد*

* در هر Node، حتماً یک LB به دست می‌آید*



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

← مثلاً در شکل صفحه‌ی قبل، 534 یک جواب شدنی هست است. یعنی قطعاً جوابی برتر از 534 را قبول نخواهیم کرد. (زیرا یک جواب 534 شدنی را در دست داریم)

Best UB = Min{feasible جواب‌های} = Min{534, 539} = 534

Best LB = Min{گره هایی که هنوز تعیین تکلیف نشده اند. (یعنی تا پایان زمان حل هنوز فرض نشده اند.)} = Min{527, 530} = 527

← پس می‌گوییم جواب بهینه قطعاً بین 534 و 527 خواهد بود.

$۵۲۷ \leq z^* \leq ۵۳۴$

* بخش ۱۰.۵ کتاب wolsey:

← در حالت کلی در Lagrangian Relaxation اگر کل محدودیت ها به تابع هدف بروند، LB ای که از آن به دست می‌آید با مشابه LP Relaxation برابر خواهد بود *

و همچنین اگر فقط تعدادی از محدودیت ها را به تابع هدف ببریم به گونه‌ای که چندوجهی حاصل از محدودیت های باقی مانده Integral باشد، باز هم LB حاصل از آن با LB مشابه LP Relaxation برابر خواهد بود *

* فقط در حالتی جواب Lagrangian Relaxation ممکن است از حل مشابه در حالت LP Relaxation بهتر شود که چند وجهی حاصل از محدودیت های باقی مانده Integral نشود، وگرنه در غیر این صورت که دو مشابه برابر خواهند بود *

* جلسه سیزدهم:

مثال: یک سری مفتول با طول استاندارد L داریم. حال برش‌هایی از این مفتول ها را برای برآوردن احتیاجات a_iها نیاز خواهیم داشت.

~~* تعداد کل شاخه ها = K~~

* تعداد الگوهای مورد نیاز = n

← می‌خواهیم تقاضا برآورده شود و در عین حال تعداد شاخه های مصرفی کمینه شود. ($\forall i \;\; a_i \leq L$)

Maral

* مدل برش ۱ در حالت عمومی:

$$x_j = \begin{cases} 1; & \text{اگر شاخه‌ی } j \text{ انتخاب شود.} \quad (j = 1, \dots, K) \\ 0; & o.w \end{cases}$$

$$y_{ij} = \begin{cases} 1; & \text{اگر تقاضای } i \text{ بر شاخه‌ی } j \text{ زده شود.} \\ 0; & o.w \end{cases}$$

$L =$ طول شاخه

$a_i =$ طول تقاضای i

$$\text{Min} \sum_{j=1}^{k} x_j$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{n} y_{ij} \cdot a_i \leq x_j \cdot L & ; \forall j \in \{1, \dots, k\} \\ (y_{ij} \leq x_j) & \forall i \in \{1, \dots, n\} \quad \forall j \in \{1, \dots, k\} \\ x_j, y_{ij} \in \{0, 1\} & \forall i, j \\ \sum_{j=1}^{k} y_{ij} = 1 & \forall i \in \{1, \dots, n\} \end{cases}$$

این محدودیت الزامی نیست ولی سرعت حل را افزایش می‌دهد

* ماکسیمم تعداد الگوهایی که برای هر مفتول می‌توانیم بزنیم چه‌قدر است 2^n است *

$$\begin{cases} S \subseteq \{a_1, a_2, \dots, a_n\} \\ \sum_{i \in S} a_i \leq L \quad (1) \end{cases}$$

← حال فرض کنید N تا از این الگوها شرط 1 را دارا هستند و لذا $(2^n - N)$ تعداد الگوهایی است که شدنی نبوده اند *