پایان نامه

برای دریافت درجه کارشناسی ارشد در رشته

ریاضی کاربردی

عنوان

کاربرد اصل تجزیه در حل مسایل برنامه ریزی عددصحیح

نگارنده

احمد خورسندعلیزاده

فهرست مطالب

فهرست شکل‌ها

فهرست جدول‌ها

مقدمه

بسیاری از مسایل مهم در زمینه بهینه سازی و همچنین بسیاری از پدیده‌های عالم واقعی در صورت مدل سازی با مقادیر عددصحیح بیان می‌شوند. به عنوان مثال تعداد سدهای ساخته‌شده روی رودخانه و یا تعداد نیروی انسانی نمی‌توانند با اعداد اعشاری بیان شوند. هرگاه تمامی متغیرهای موجود در مدل مساله متغیر صحیح باشند به آن برنامه‌ریزی عددصحیح محض[1] گفته می‌شود که از کاربردهای آن می‌شود به مسایل هزینه ثابت یا مسایل تخصیص نیروی انسانی اشاره کرد. چنانچه بعضی از متغیرها عددصحیح و برخی متغیرها پیوسته باشند به آن مدل برنامه‌ریزی عددصحیح آمیخته[2] گفته می‌شود که ازکاربردهای آن می‌توان به مسایلی چون تخصیص و زمان‌بندی[3] اشاره کرد.

یک زمینه دیگر از کاربردهای برنامه‌ریزی عددصحیح که از اهمیت زیادی برخوردار است، مسایل تصمیم‌گیری از نوع ''بله یا نه'' می‌باشد. به عنوان نمونه آیا منطقه x مکان مناسبی برای ایجاد یک مرکز فروش است یا خیر؟ یا در مسایل روی گراف و شبکه آیا یال e_{ij} انتخاب شود یا خیر؟

این‌گونه محدودیت‌ها را که متغیرهای آن فقط دو انتخاب در پیش داشته باشند می‌توان بر حسب متغیرهایی بیان کرد که فقط دو مقدار اختیار می‌کنند به طوری که اگر تصمیم i بله باشد، $x_i = ۱$ و اگر خیر باشد،

[1] Pure
[2] Mixed
[3] Scheduling

$x_i = ۰$. به مسایلی که فقط دارای متغیرهای دودویی باشند، مسایل برنامه‌ریزی صفر و یک یا دودویی گفته می‌شود .

در حالت کلی هدف برنامه‌ریزی عدد صحیح پیدا کردن مقدار کمینه یا بیشینه از یک تابع خطی بر روی فضایی با محدودیت‌های خطی است، اما به دلیل وجود متغیرهای گسسته، این فضا پیوسته و محدب نیست بلکه فضایی گسسته و در نتیجه، نامحدب است.

استفاده از الگوریتم شاخه‌وکران برای حل این مسایل روش موفقی بوده است و مرتبا به‌روز می‌شود.

در فصل اول به بیان تعاریف و معرفی نمادهای مورد استفاده می‌پردازیم.

در فصل دوم مثال‌هایی از کاربرد تجزیه را بیان می‌کنیم.

در فصل سوم به بررسی روش‌های تجزیه می‌پردازیم که به بهبود کران در هر گام از الگوریتم شاخه‌وکران کمک می‌کنند. به این منظور ابتدا به بررسی سه روش مرسوم یعنی روش‌های صفحه‌برش[۱] و تجزیه دانتزیگ و ولف[۲] و روش لاگرانژ[۳] می‌پردازیم و سپس روش‌های ادغامی[۴] را که ترکیبی از روش‌های مرسوم هستند بررسی می‌کنیم.

در فصل چهارم مروری بر نرم‌افزارهای حل مسایل عددصحیح و نتایج محاسباتی آن‌ها داریم.

[۱] Cutting Plane
[۲] Dantzig-Wolfe Decomposition
[۳] Lagrangian Method
[۴] Integrated

فصل ۱

تعاریف و نمادها

۱.۱ روش شاخه‌وکران

یکی از موفق‌ترین روش‌ها برای حل مسایل عددصحیح، روش شاخه‌وکران است؛ البته کاربردهای این روش محدود به مسایل عددصحیح نمی‌شود و در انواع دیگری از مسایل هم استفاده می‌گردد. رویکرد شاخه وکران بر این اصل شکل گرفته است که فضای کل جواب های شدنی را به زیر مجموعه‌های کوچکتری تقسیم می کنیم به گونه‌ای که با استفاده از این زیرمجموعه‌ها و به کمک یک ارزیابی نظام‌مند بتوان بهترین جواب شدنی را پیدا کرد. مثال زیر را در نظر بگیرید:

$$\max \;\; Z = 100x_1 + 150x_2$$

$$s.t.$$

$$8000x_1 + 4000x_2 \leq 40000$$

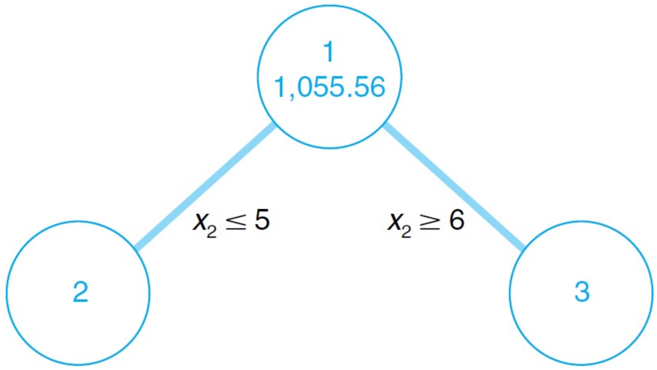
$$15x_1 + 30x_2 \leq 200$$

$$x_i \geq 0, \text{integers}, \;\; i = 1, 2$$

درگام اول مساله را بدون در نظرگرفتن محدودیت صحیح بودن متغیرها حل می کنیم تا یک کران بالا به دست آید $UB = 1055/56 (x_1 = 2/22, x_2 = 5/56)$

با توجه به علامت محدودیت‌ها و مثبت بودن متغیرها اگرx_1, x_2 را به سمت پایین گرد کنیم یک جواب شدنی (کران پایین) به دست می‌آید. $LB = 950$ و $(x_1 = 2, x_2 = 5)$

درگام بعد با شاخه زدن روی متغیر x_2 مطابق شکل(۱.۱) زیر مساله‌های ۲ و ۳ ایجاد می‌شوند. درگره ۲



شکل ۱.۱: شاخه‌زنی روی متغیر $x_۲$

مساله‌ی زیر را حل می‌کنیم:

$$\max \quad Z = ۱۰۰x_۱ + ۱۵۰x_۲$$

$$s.t.$$

$$۸۰۰۰x_۱ + ۴۰۰۰x_۲ \leq ۴۰۰۰۰$$

$$۱۵x_۱ + ۳۰x_۲ \leq ۲۰۰$$

$$x_۲ \leq ۵$$

$$x_i \geq ۰, \text{integers}, \ i = ۱, ۲$$

در این گام $x_۱ = ۲/۵, x_۲ = ۵, z = ۱۰۰۰$ به دست می‌آید. درگره۳ مساله‌ی زیر را حل می‌کنیم:

$$\max \quad Z = ۱۰۰x_۱ + ۱۵۰x_۲$$

$$s.t.$$

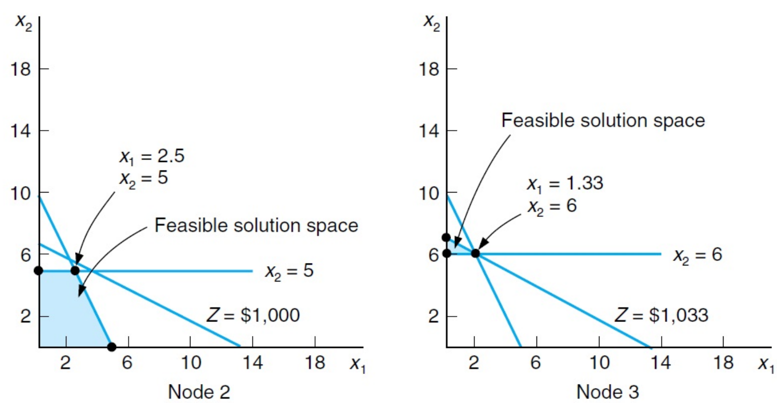
$$8000x_1 + 4000x_2 \leq 40000$$

$$15x_1 + 30x_2 \leq 200$$

$$x_2 \geq 6$$

$$x_i \geq 0, \text{integers}, \; i = 1, 2$$

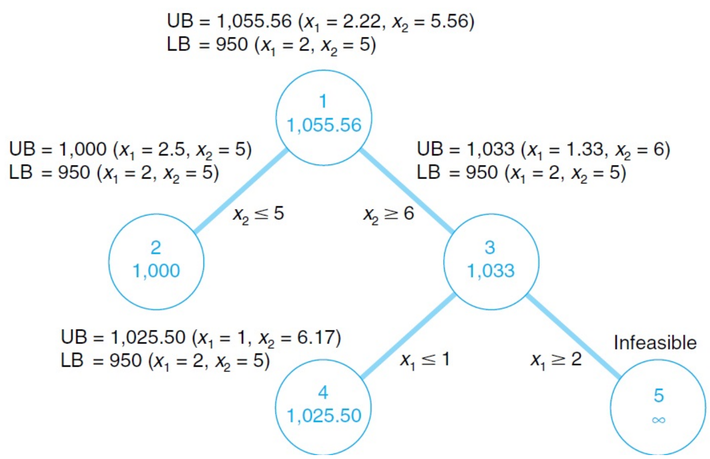
و جواب بهینه $x_1 = 1.33, x_2 = 6, z = 1033.33$ می‌باشد.



شکل ۲.۱: فضای شدنی برای زیر مساله های ۲و۳

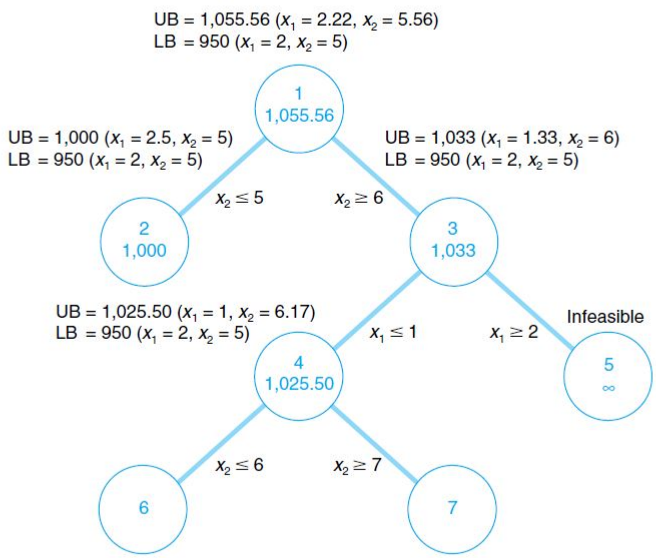
در شکل (۲.۱) فضای شدنی زیرمساله‌های ۲ و ۳ را مشاهده می‌نماییم. کران بالا درگره ۲، ۱۰۰۰ و درگره ۳، ۱۰۳۳ می‌باشد یعنی با ادامه شاخه زنی ازگره ۲، حداکثر مقداری که برای تابع هدف به دست می‌آید، $z^* = 1000$ است و با ادامه ازگره ۳ حداکثر مقداری که ممکن است برای تابع هدف به دست آید $z^* = 1033$ است. بنابراین با توجه به این که تابع هدف از نوع بیشینه‌سازی است، ازگره ۳ شاخه زنی را

ادامه می‌دهیم. مطابق شکل (۳.۱) با اضافه کردن محدودیت‌های $x_1 \leq 1, x_1 \geq 2$ به گره ۳ ، گره‌های ۴ و ۵ ساخته می‌شوند.



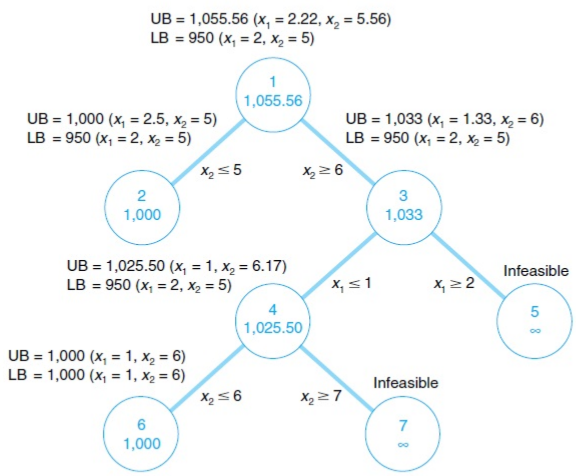
شکل ۳.۱: نمودار شاخه و کران به همراه کران بالا و پایین در گره‌های ۴ و ۵

جواب بهینه‌ی گره ۴ عبارت است از: $x_1 = 1, x_2 = 6/17, Z = 1025$ و فضای شدنی مساله‌ی گره ۵ تهی می‌شود. در ادامه با توجه به این که کران بالا در گره ۴ بیشتر از کران بالا در گره ۲ است، شاخه‌زنی را از گره ۴ ادامه می‌دهیم و مطابق شکل (۴.۱)، محدودیت‌های $x_2 \leq 6, x_2 \geq 7$ را به محدودیت‌های قبلی گره ۴ اضافه می‌کنیم.



شکل ۴.۱: شاخه‌زنی مجدد روی متغیر $x_۲$

جواب بهینه‌ی گره ۶ عبارت است از $x_۱ = ۱, x_۲ = ۶, Z = ۱۰۰۰$ چون به یک نقطه‌ی صحیح رسیدیم کران پایین هم تغییر می‌کند و در این گام $UB = LB = ۱۰۰۰$ مساله‌ی گره ۷ نیز نشدنی می‌شود. از آن جا که کران بالا در گره ۲ با کران پایین حاصل از حل مساله گره ۶ برابر است، با شاخه‌زنی از گره ۲ نمی‌توان کران بهتری از کران موجود به دست آورد بنابراین جواب بهینه همان جواب گره ۶ می‌باشد.



شکل ۵.۱: نمودار شاخه و کران و به دست آمدن جواب بهینه در گره ۶

الگوریتم شاخه‌وکران را می‌توان به صورت گام‌های زیر بیان کرد:

الگوریتم ۱ الگوریتم شاخه‌وکران برای حل مساله عدد صحیح: (در حالت بیشینه‌سازی و محدودیت‌های بزرگتر مساوی)

گام ۱. جواب بهینه‌ی آزادسازی خطی مساله را به‌دست می‌آوریم. (گره ۱) (می‌توان به‌جای آزادسازی خطی، آزاد سازی‌های دیگری را هم استفاده کرد.)

گام ۲. در گره ۱، یک کران بالا برای تابع هدف به‌دست آوردیم. در این گام با گرد کردن مقادیر متغیرها به سمت پایین، به یک جواب شدنی صحیح برای این مساله می‌رسیم که یک کران پایین برای مقدار تابع هدف به‌دست خواهد داد.

گام ۳. یکی از متغیرهای ناصحیح را انتخاب می‌کنیم و با شاخه زدن روی آن متغیر دو گره جدید خواهیم داشت.

گام ۴. مسایل گره‌های جدید را حل می‌کنیم تا کران بالا در هر گره مشخص شود و اگر در یک گره همه‌ی متغیرها صحیح شدند، کران پایین نیز به‌روز می شود.

گام ۵. در ادامه شاخه زدن را ابتدا برای گره‌هایی انجام می‌دهیم که کران بالای بیشتری دارند. اگر همه‌ی گره‌ها دارای یکی از سه شرط زیر هستند، متوقف می‌شویم:

۱) مساله نشدنی است.

۲) تمام متغیرها، صحیح‌اند.

۳) کران بالای حاصل از حل مساله‌ی گره، با کران پایین مساله برابر است یا کمتر از آن است.

گام ۶. به گام سه می رویم.

نکته: برای مساله‌ی کمینه‌سازی با محدودیت‌های کوچکتر مساوی ، کافی است در گام دو متغیرها را

به بالا گرد کنیم و در الگوریتم، جای کران بالا و کران پایین را عوض کنیم.

۲.۱ برنامه ریزی پویا[۱]

یکی از مفاهیمی که آشنایی با آن به درک بهتر مطالبی که در ادامه بیان می‌شود کمک می‌کند، برنامه‌ریزی پویا است ([۱۵] را ببینید).

برنامه‌ریزی پویا روشی بهینه برای تبدیل مسایل پیچیده به مجموعه‌ای از مسایل ساده‌تر است. برنامه‌ریزی پویا یک قالب کلی برای حل دسته‌ای از مسایل است و بر خلاف برنامه‌ریزی خطی، چارچوبی استاندارد برای طبقه‌بندی مسایل مربوط به آن وجود ندارد و معمولاً برای حل این دسته از مسایل به خلاقیت نیاز است. در واقع آن‌چه برنامه‌ریزی پویا انجام می‌دهد ارایه روش برخورد کلی جهت حل دسته‌ای از مسایل است.

به صورت خلاصه می‌توان راه‌کارهای برنامه‌ریزی پویا را به صورت زیر طبقه‌بندی کرد:

۱. تعیین زیرمساله‌های مرتبط

۲. حل سوال مساله‌های بزرگتر با استفاده از زیرمساله‌های کوچکتر

۳. به دست آوردن جواب نهایی مساله با استفاده از جواب زیرمساله‌ها (ممکن است مساله اصلی هم یکی از زیرمساله‌ها باشد.)

[۱] Dynamic Programming

۳.۱ نمادها

در ادامه برای سهولت فقط مساله‌ی عددصحیح محض با ناحیه‌ی موجه کراندار و ناتهی را در نظر می‌گیریم. هرچند می‌توان روش‌های ارایه شده را به حالت عمومی‌تر گسترش داد. در فصل‌های بعد از نمادهای زیر برای مساله‌ی ILP استفاده می‌کنیم؛

$$\mathcal{Q} = \{x \in \mathbb{R}^n \big| Ax \geq b\}$$

که $A \in \mathbb{Q}^{m\times n}$ و $b \in \mathbb{Q}^m$ است. همچنین قرار می‌دهیم $\mathcal{F} = \mathcal{Q} \cap \mathbb{Z}^n$ و پوسته محدب $\mathcal{F}$ را با $\mathcal{P}$ نمایش می‌دهیم. در این صورت مساله به شکل زیر تعریف می‌شود:

$$Z_{IP} = \min_{x\in\mathbb{Z}^n}\{c^Tx \big| Ax \geq b\} = \min_{x\in\mathcal{F}}\{c^Tx\} = \min_{x\in\mathcal{P}}\{c^Tx\}$$

برای بردار داده شده‌ی $c \in \mathbb{Q}^n$ اگر $\mathcal{F}$ تهی بود قرار می‌دهیم $z = \infty$. این مساله را به طور خلاصه با نماد $\mathrm{ILP}(\mathcal{P}, \mathrm{c})$ نمایش می‌دهیم.

مساله‌ی مرتبط دیگر مساله‌ی جداسازی[۱] است که روی $\mathcal{P}$ تعریف می‌شود و از مسایل تصمیم‌گیری[۲] به حساب می‌آید. برای x داده شده از فضای $\mathbb{R}^n$؛ مساله جداسازی x از $\mathcal{P}$ مشخص می‌کند که یا $x \in \mathcal{P}$ است و یا در غیر این صورت $a \in \mathbb{R}^n$ و $\beta \in \mathbb{R}$ را به گونه‌ای تعیین می‌کند که به ازای هر $y \in \mathcal{P}$ داشته باشیم $a^Ty \geq \beta$ اما $a^Tx < \beta$. با این شرایط جفت $(a, \beta) \in \mathbb{R}^{n+1}$ معرف یک نامعادله معتبر[۳] برای $\mathcal{P}$ است که توسط x نقض شده است.

[۱] Separation
[۲] Decision Making Problems
[۳] Valid Inequality

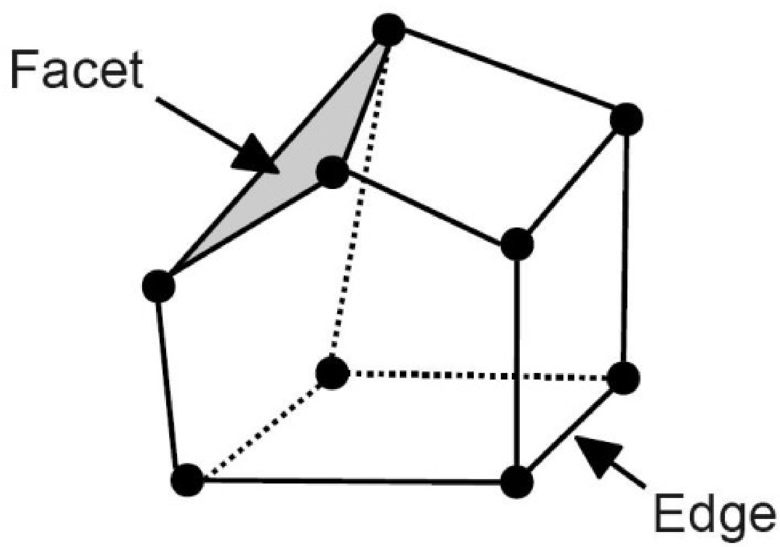
مساله جداسازی را با نماد $\mathrm{SEP}(\mathcal{P}, \mathrm{x})$ نمایش می‌دهیم. خروجی این مساله یا مجموعه تهی است و یا مجموعه‌ای از نامعادلات معتبر برای $\mathcal{P}$ که توسط x نقض شده‌اند. این نامعادلات را بُرش می‌نامیم.

تعریف ۱.۳.۱. • محدودیت $a^T x \geq b$ در نقطه‌ی y فعال است اگر و تنها اگر $a^T y = b$.

- M را یک وجه[۱] از چندوجهی $\mathcal{P}$ می‌نامیم هرگاه حداقل در یکی از محدودیت های آن فعال باشد.

- e یک یال[۲] $\mathcal{P}$ است اگر یک وجه $\mathcal{P}$ باشد و بعد آن ۱ باشد.

- G را یک منظر[۳] $\mathcal{P}$ می‌نامیم هرگاه یک وجه آن باشد و اختلاف بعد آن با $\mathcal{P}$، ۱ باشد یعنی $dim(p) - ۱ = dim(G)$.



شکل ۶.۱: نمایش یال و منظر برای یک چندوجهی

مساله‌ی دقیق‌تر مساله شناسایی منظر[۴] است که در آن برش‌هایی تولید می‌شوند که شامل منظرهای

[۱]Face
[۲]Edge
[۳]Facet
[۴]Facet Identification Problem

$\mathcal{P}$ باشند.

۴.۱ اصل تجزیه

همان طور که در مقدمه بیان شد برای اجرای الگوریتم شاخه و کران نیاز به تولید کران پایین برای مقدار بهینه (Z_{IP}) داریم و عمومی‌ترین روش برای این کار استفاده از آزادسازی خطی[۱] است.

$$Z_{LP} = \min_{x\in\mathbb{R}^n}\{c^t x \big| Ax \geq b\} = \min_{x\in\mathcal{Q}}\{c^t x\} \quad (۱.۱)$$

واضح است که $Z_{LP} \leq Z_{IP}$ چون $\mathcal{P} \subseteq \mathcal{Q}$.

برای بهبود کران خطی، روش‌های تجزیه چندوجهی دومی را تقریب می‌زنند که از قطع دادن آن با $\mathcal{Q}$ کران بهتری حاصل می‌شود. متاسفانه بر خلاف $\mathcal{Q}$ این چندوجهی دوم معمولاً اندازه نمایی[۲] دارد و باید بخشی از آن به طور پویا ساخته شود. بدین منظور می‌توان از روش صفحه برش استفاده کرد که یک روش تقریب بیرونی است و همچنین می‌توان از روش‌های تولید ستون مانند روش تجزیه دانتزیگ و ولف [۵]و یا روش لاگرانژ [۳،۸] که روش‌های تقریب درونی هستند استفاده کرد.

در فصل بعد توضیح مختصری درباره‌ی تفاوت تقریب درونی و بیرونی خواهیم داد. در ادامه آزادسازی (۲.۱) را برای (۱.۱) در نظر بگیرید.

$$\min_{x\in\mathbb{Z}^n}\{c^T x \big| A'x \geq b'\} = \min_{x\in\mathcal{F}'}\{c^T x\} = \min_{x\in\mathcal{P}'}\{c^T x\} \quad (۲.۱)$$

[۱] LP Relaxation
[۲] Exponential Size

که در آن $\mathcal{F} \subset \mathcal{F}' = \{x \in \mathbb{Z}^n \big| A'x \geq b'\}$ و $A' \in \mathbb{Q}^{m' \times n}, b' \in \mathbb{Q}^{m'}$ و $\mathcal{P}'$ پوسته محدب $\mathcal{F}'$ و $\mathcal{Q} = \{x \in \mathbb{R}^n \big| A'x \geq b', A''x \geq b''\}$.

بنابراین اگر چندوجهی‌های $\mathcal{Q}'$ و $\mathcal{Q}''$ به ترتیب فضای جواب محدودیت‌های $[A'|b']$ و $[A''|b'']$ را مشخص کنند داریم

$$\mathcal{Q} = \mathcal{Q}' \cap \mathcal{Q}''$$

اساس کار روش‌های تجزیه بر این نکته است که با بهینه‌سازی روی $\mathcal{P}' \cap \mathcal{Q}''$ کران بهتری از کران آزادسازی خطی بدست می‌آید چون $\mathcal{P}' \cap \mathcal{Q}'' \subset \mathcal{Q}$ در این تقسیم‌بندی محدودیت‌ها فرض بر این است که یک الگوریتم برای حل مسایل بهینه‌سازی یا جداسازی روی $\mathcal{P}'$ در دست است، به عبارتی $[A'|b']$ محدودیت‌های آسان و $[A''|b'']$ محدودیت‌های پیچیده هستند.

در روش‌هایی که ذکر کردیم باید تعداد محدودیت‌های پیچیده از اندازه چندجمله‌ای باشد یا به عبارتی دیگر $\mathcal{Q}''$ دارای نمایش صریح[۱] باشد. بنابراین کران حاصل از روش‌های تجزیه به صورت زیر خواهد بود:

$$\begin{aligned} Z_D &= \min_{x \in \mathcal{P}'} \{c^T x \big| A''x \geq b''\} \\ &= \min_{x \in \mathcal{F}' \cap \mathcal{Q}''} \{c^T x\} = \min_{x \in p' \cap \mathcal{Q}''} \{c^T x\} \end{aligned} \tag{۳.۱}$$

[۱] Explicit

مراجع

[1] C. Barnhart, C. A. Hane, and P. H. Vance. Using branch-and-price-and-cut to solve origin destination integer multi-commodity flow problems. *Operations Research*, **48**:318-326, 2000.

[2] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch and price: Column generation for solving huge integer programs. *Operations Research*, **46**:316–329, 1998.

[3] J.E. Beasley. Lagrangean relaxation. In C.R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Optimization*. Wiley, 1993.

[4] L. Chen, S. Khan, K. Li, and E. Manning. Building an adaptive multimedia system using the utility model. In *Proceedings of the 11 IPPS/SPDP'99 Workshops Held in Conjunction with the 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing*, pages 289-298, London, UK, 1999. Springer-Verlag.

[5] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, **8(1)**:101–111, 1960.

[6] Danzig, G. and Ramser, R., The truck dispatching problem, *Management Science*, 6, 80, 1959.

[7] Fisher, M., Optimal solution of vehicle routing problems using minimum ktrees, *Operations Research*, 42, 626, 1994..

[8] M.L. Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management Science*, **27**:1-18, 1981.

[9] M. V. Galati., *Decomposition in Integer Programming*, PhD Thesis, Lehigh University, December 2009.

[10] M. Galati. DIP, 2009. Available from http://www.coin-or.org/projects/Dip. xml.

[11] M. Held and R. M. Karp. The traveling salesman problem and minimum spanning trees. *Operations Research*, **18**:1138-1162, 1970.

[12] N. Kohl, J. Desrosiers, O.B.G. Madsen, M.M. Solomon, and F. Soumis. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, **33**:101–116, 1999.

[13] L. Ladanyi. CoinUtils. Available from http://www.coin-or.org/.

[14] Laporte, G., Nobert, Y., and Desrouchers, M., Optimal routing with capacity and distance restrictions, *Operations Research*, 33, 1050, 1985.

[15] A. Lew. and H. Mauch., *Dynamic Programming*, Springer, 2007.

[16] J. T. Linderoth and Lodi, A. *MILP Software*. Wiley Encyclopedia of Operations Research and Management Science. 2011.

[17] J. T. Linderoth and T. K. Ralphs. Noncommercial software for mixed integer linear programming. *In Integer Programming: Theory and Practice*, pages 253-303. CRC Press Operations Research Series, 2005.

[18] S. Martello and P. Toth. *Knapsack Problems: algorithms and computer implementation.* John Wiley & Sons, Inc., USA, 1st edition, 1990.

[19] C. Martinhon, A. Lucena, and N. Maculan. *A relax and cut method for the vehicle routing problem*. Unpublished working paper, 2001.

[20] D. Miller, A matching based exact algorithm for capacitated vehicle routing problems, *ORSA Journal on Computing*, 7, 1, 1995.

[21] M. de Moraes Palmeira, A. Lucena, and O. Porto. *A Relax and Cut Algorithm for Quadratic Knapsack Problem*. Technical report, Universidade Federal do Rio de Janeiro, 1999.

[22] Müller-Hannemann, M. and Schwartz, A., Implementing weighted b-matching algorithms: Toward a flexible software design, in *Proceedings of the Workshop on Algorithm Engineering and Experimentation (ALENEX99), volume 1619 of Lecture notes in Computer Science*, pages 18–36, Baltimore, MD, 1999, Springer-Verlag.

[23] H. Nagamochi, T. Ono, and T. Ibaraki. Implementing an efficient minimum capacity cut algorithm. *Mathematical Programming*, **67**:325-341, 1994.

[24] T.K. Ralphs. Parallel Branch and Cut for Vehicle Routing. PhD thesis, Cornell University, May 1995.

[25] T. K. Ralphs and M. V. Galati. Decomposition and dynamic cut generation in integer linear programming. *Math. Program.*, 106(2):261–285, 2006.

[26] T.K. Ralphs , M. Galati. *Decomposition in Integer Programming, in Integer Programming: Theory and Practice*, John Karlof, ed. (2005), 57-110.

[27] T.K. Ralphs, L. Kopman, W.R. Pulleyblank, and L.E. Trotter Jr. On the capacitated vehicle routing problem. *Mathematical Programming*, **94**:343-359, 2003.

[28] F. Vanderbeck. Lot-sizing with start-up times. *Management Science*, **44**:1409-1425, 1998.

[29] J.M. Van den Akker, C.A.J. Hurkens, and M.W.P. Savelsbergh. Time-indexed formulations for machine scheduling problems: Column generation. *INFORMS Journal on Computing*, **12**:111-124, 2000.

[30] R.K. Watson. *Packet Networks and Optimal Admission and Upgrade of Service Level Agreements: Applying the Utility Model.* PhD thesis, University of Victoria, 2001.

واژه‌نامه فارسی به انگلیسی

ا

آزاد سازی و برش .... Relax and Cut

آزادسازی خطی .... LP Relaxation

آمیخته .... Mixed

ادغامی .... Integrated

ارزش و برش .... Price and Cut

اندازه نمایی .... Exponential Size

ب

برش کمینه .... Minimum Cut

برش‌های تجزیه .... Decomposition Cuts

برنامه ریزی پویا .... Dynamic Programming

ت

تجزیه دانتزیگ و ولف .... Dantzig-Wolfe Decomposition

تجزیه و برش .... Decompose and Cut

ج

جداسازی .... Separation

خ

خاصیت صحیح بودن .... Integrality Property

ر

رابط الگوریتم .... Algorithms Interface

رابط کاربردی .... Applications Interface

روش لاگرانژ .... Lagrangian Method

ز

زمان‌بندی .... Scheduling

ص

صریح .... Explicit

صفحه‌برش .... Cutting Plane

ک

کوله‌پشتی .... Knapsack Problem

م

محض .... Pure

مساله شناسایی منظر .... Facet Identification Problem

مساله کوله پشتی چند انتخابی چند معیاری .... Multi-ChoiceMulti-Dimensional Knapsack Problem

مساله مدیریت پول نقد دستگاه‌های خودپرداز .... ATM Cash Management Problem

مساله‌ی حمل‌ونقل .... Vehicle Routing Problem

مساله‌ی b-جورسازی کامل .... Perfect b-Matching Problem

مسایل تصمیم‌گیری .... Decision Making Problems

منظر .... Facet

ن

نامعادله معتبر .... Valid Inequality

و

وجه .... Face

ی

یال .... Edge

واژه‌نامه انگلیسی به فارسی

A

Algorithms Interface .... رابط الگوریتم

Applications Interface .... رابط کاربردی

ATM Cash Management Problem .... مساله مدیریت پول نقد دستگاه‌های خودپرداز

C

Cutting Plane .... صفحه‌برش

D

Dantzig-Wolfe Decomposition .... تجزیه دانتزیگ و ولف

Decision Making Problems .... مسایل تصمیم‌گیری

Decompose and Cut .... تجزیه و برش

Decomposition Cuts .... برش‌های تجزیه

Dynamic Programming .... برنامه ریزی پویا

E

Edge .... یال

Explicit .... صریح

Exponential Size .... اندازه نمایی

F

Face .... وجه

Facet .... منظر

Facet Identification Problem .... مساله شناسایی منظر

I

Integrality Property .... خاصیت صحیح بودن

Integrated .... ادغامی

K

Knapsack Problem .... کوله‌پشتی

L

Lagrangian Method .... روش لاگرانژ

LP Relaxation .... آزادسازی خطی

M

Minimum Cut .... برش کمینه

Mixed .... آمیخته

Multi-ChoiceMulti-Dimensional Knapsack Problem .... مساله کوله پشتی چند انتخابی چند معیاری

P

Perfect b-Matching Problem .... مساله‌ی b-جورسازی کامل

Price and Cut .... ارزش و برش

Pure .... محض

R

Relax and Cut .... آزاد سازی و برش

S

Scheduling .... زمان‌بندی

Separation .... جداسازی

V

Valid Inequality .... نامعادله معتبر

Vehicle Routing Problem .... مساله‌ی حمل‌ونقل

نمایه