دانشگاه صنعتی شیراز

دانشکده ریاضی

پایان نامه کارشناسی ارشد
در رشته ریاضی گرایش تحقیق در عملیات

یک روش شاخه و کران کارا مبتنی بر تولید ستون

نگارش:
فهیمه نظری

استاد راهنما:
دکتر سید مصطفی خرمی زاده

شهریورماه ۱۳۹۳

تقدیم به پدر، مادر و همسر مهربانم

// سپاس گزاری...

سپاس خداوندگار حکیم را که با لطف بی‌کران خود، آدمی را زیور عقل آراست. اکنون با به پایان رسیدن این مقطع تحصیلی، بر خود لازم می‌دانم از تمامی استادان، دوستان و عزیزانی که در طول این دوره تا به ثمر رسیدن آن مرا یاری و همراهی نمودند سپاسگزاری نمایم....

چکیده

یک روش شاخه و کران کارا مبتنی بر تولید ستون

نگارش:
فهیمه نظری

هدف این پایان نامه، بررسی ترکیب روش شاخه و کران و روش تولید ستون برای حل مسایل برنامه ریزی با اعداد صحیح با تعداد زیادی متغیر می باشد. بدین منظور تقسیم بندی متغیر ها را به متغیر های اندیس، الگوریتمی و هسته ای شرح داده و نقش هر دسته از این متغیر ها را بررسی می کنیم. سپس، برخی از چالش های پیاده سازی یک روش شاخه و کران مبتنی بر تولید ستون را مورد بحث قرار می دهیم. از جمله برخی روش های شاخه سازی، مفهوم بازیابی شدنی بودن و پیاده سازی مدولی را مورد بحث قرار می دهیم. این مفاهیم بیان شده ما را قادر می سازد که الگوریتمی برای مسایل برنامه ریزی با اعداد صحیح معرفی کنیم. سپس با مفاهیم بیان شده با الگوریتم حاصل از ترکیب روش شاخه و کران و تولید ستون مساله جریان چند کالایی صحیح را حل کرده و سعی می کنیم با ارایه نتایج عددی کارایی این الگوریتم را برای حل مساله جریان چند کالایی صحیح مورد ارزیابی قرار دهیم.

فهرست مطالب

عنوان	صفحه

فصل ۱: مفاهیم و تعاریف مورد نیاز ۱
۱-۱ مقدمه ۲
۱-۲ نظریه گراف ۳
۱-۲-۱ روش های جستجو ۴
۱-۳ برنامه ریزی خطی ۵
۱-۳-۱ روش سیمپلکس متغیر های کراندار ۷
۱-۴ مساله برنامه‌ریزی صحیح آمیخته ۱۰
۱-۵ روش شاخه و کران ۱۱
۱-۶ مساله شبکه جریان با هزینه کمینه ۱۳

فصل ۲: روش شاخه و هزینه ۱۸
۲-۱ مقدمه ۱۹
۲-۲ روش تولید ستون ۲۰
۲-۲-۱ قیمت گذاری هزینه کاهش یافته ۲۴
۲-۲-۲ بازیابی شدنی بودن ۲۷
۲-۲-۳ اثبات همگرایی روش تولید ستون ۳۰
۲-۳ روش تولید ستون بلوکی ۳۲
۲-۴ روش شاخه و هزینه ۳۷
۲-۴-۱ کیفیت سست سازی ۴۰
۲-۵ شاخه سازی ۴۳
۲-۵-۱ شاخه سازی روی متغیر های مساله بنیادی ۴۴
۲-۶ تفاوت بین روش تولید ستون و تجزیه بندرز ۴۸

فصل ۳: نحوه پیاده سازی کارای روش شاخه و هزینه ۵۲
۳-۱ مقدمه ۵۳

۳-۲ ساختار درخت جستجو ۵۶
۳-۳ پیاده سازی مدولی ۵۷

فصل ۴: مساله جریان چند کالایی ۶۳
۴-۱ مقدمه ۶۴
۴-۲ معرفی مساله ۶۵
۴-۳ شرح روش شاخه و هزینه برای $(IMFP)$ ۶۶
۴-۴ نتایج عددی ۷۲
۴-۴-۱ شرح محیط پیاده سازی ۷۲
۴-۴-۲ مسایل آزمون ۷۳

فصل ۵: نتایج و پیشنهاد ها ۱۰۱

آ پیوست ۱۰۵

مراجع ۱۱۲

واژه‌نامه فارسی به انگلیسی ۱۱۵

فهرست جدول‌ها

عنوان	صفحه
جدول ۴–۱: جدول مربوط به مثال ۳	۷۴
جدول ۴–۲: جدول مربوط به مثال ۴	۷۶
جدول ۴–۳: جدول مربوط به مثال ۵	۷۶
جدول ۴–۴: جدول مربوط به مثال۶	۷۸
جدول ۴–۵: جدول مربوط به مثال ۷	۷۸
جدول ۴–۶: جدول مربوط به مثال ۸	۷۹
جدول ۴–۷: جدول مربوط به مساله ۹	۸۰
جدول ۴–۸: جدول مربوط به مساله ۱۰	۸۱

فهرست شکل‌ها

عنوان	صفحه
شکل ۱–۱: یک مثال جریان شبکه با کران بالا و پایین	۱۶
شکل ۱–۲: روش حل یک مثال جریان شبکه	۱۷
شکل ۴–۱: مساله کمینه هزینه جریان دوکالایی	۷۰
شکل ۴–۲: جدول مربوط به مثال ۳	۷۵
شکل ۴–۳: جدول مربوط به مثال ۴	۷۷
شکل ۴–۴: ادامه جدول مربوط به مثال ۴	۷۷
شکل ۴–۵: جدول مربوط به مثال ۵	۸۳
شکل ۴–۶: ادامه جدول مثال ۵	۸۴
شکل ۴–۷: ادامه جدول مثال ۵	۸۵
شکل ۴–۸: ادامه جدول مثال ۵	۸۶
شکل ۴–۹: ادامه جدول مربوط به مثال ۵	۸۷
شکل ۴–۱۰: ادامه جدول مربوط به مثال ۵	۸۸
شکل ۴–۱۱: جدول مربوط به مثال ۶	۸۹
شکل ۴–۱۲: ادامه جدول مثال ۶	۹۰
شکل ۴–۱۳: ادامه جدول مثال ۶	۹۱
شکل ۴–۱۴: ادامه جدول مثال ۶	۹۲
شکل ۴–۱۵: جدول مربوط به مثال ۷	۹۳
شکل ۴–۱۶: ادامه جدول مثال ۷	۹۴
شکل ۴–۱۷: ادامه جدول مثال ۷	۹۵
شکل ۴–۱۸: ادامه جدول مثال ۷	۹۶
شکل ۴–۱۹: جدول مربوط به مثال ۸	۹۷
شکل ۴–۲۰: جدول مربوط به مثال ۹	۹۸
شکل ۴–۲۱: ادامه جدول مثال ۹	۹۹

شکل ۴–۲۲: جدول مربوط به مثال ۱۰ . . . . ۱۰۰

فصل ۱

مفاهیم و تعاریف مورد نیاز

۱–۱ مقدمه

هدف از نوشتن این پایان نامه بررسی ترکیب روش شاخه و کران با روش تولید ستون است. ریشه های اولیه روش تولید ستون در سال ۱۹۶۰ بوده است[۲]. ترکیب فرآیند تولید ستون با الگوریتم شاخه و کران به منظور حل مشکل صحیح بودن مسایل در ابتدا دارای چالش های بسیاری بود. بنابراین چندین سال طول کشید تا در عمل تولید ستون به طور موفقیت آمیزی با شاخه و کران ترکیب شد[۷]. بحثی که در سال های اخیر پیش آمده این است که آیا نرم افزار های چارچوب جعبه سفید برای حل مسایل موثر هستند. در این پایان نامه همچنین کارایی یک نرم افزار جعبه سفید را بررسی می کنیم. در ضمن الگوریتم شاخه و هزینه را با استفاده از یکی از این نرم افزار پیاده سازی می کنیم و برای حل مساله جریان چند کالایی صحیح از آن استفاده خواهیم کرد. در این فصل، مفاهیم مورد نیاز مرتبط با فصل های بعد را بیان می کنیم. در ابتدا، تعاریف مربوط به نظریه گراف را در بخش ۱–۲ می آوریم. این تعاریف برای حل و بررسی مساله جریان چند کالایی و نحوه انتخاب گره برای پردازش به کار می آیند. سپس، در بخش ۱–۳ مساله برنامه ریزی خطی را معرفی کرده و روش سیمپلکس با متغیر های کراندار را شرح می دهیم. در بخش ۱–۴ مساله برنامه ریزی صحیح آمیخته را مرور می کنیم و روش شاخه و کران را که یکی از روش های حل این نوع مسایل است، در بخش ۱–۵ می آوریم. در نهایت، مباحث مربوط به جریان در شبکه را در بخش ۱–۶ بیان می کنیم.

۱-۲ نظریه گراف

تعاریف این بخش برگرفته از [۱۲] است.

تعریف ۱.۱: (گراف ساده) فرض کنید V مجموعه ای ناتهی باشد و $E \subseteq V \times V$. گراف G، یک جفت (V, E) است به طوری که V، یک مجموعه متناهی و مخالف تهی از گره‌ها و E، یک مجموعه متناهی (در صورت امکان تهی) از یال‌هاست که گره ها را به هم وصل می کند و آن را با نماد $G(V, E)$ نمایش می‌دهیم.

تعریف ۲.۱: (گراف جهت دار) جفت مرتب (V, E) را گراف جهت دار (بر V) می نامند، که در آن V مجموعه راس ها و E مجموعه یال ها است و آن را با نماد $G(V, E)$ نمایش می‌دهیم.

تعریف ۳.۱: (گشت) فرض کنید x و y (که لزوما متمایز نیستند) در گراف بی جهت $G(V, E)$ راس باشند. یک گشت $x - y$ در G، دنباله متناهی

$$x = x_۰, e_۱, x_۱, e_۲, x_۲, ..., e_{n-۱}, x_{n-۱}, e_n, x_n = y$$

از راس ها و یال های G است، که از راس x شروع شده و با n یال $e_i = x_{i-۱}, x_i$، $۱ \leq i \leq n$، به راس y ختم می شود.

تعریف ۴.۱: (مسیر) یک گشت $x - y$ را در گراف بدون جهت $G(V, E)$ در نظر بگیرید. اگر هیچ راسی از گشت $x - y$ بیش از یک بار تکرار نشود، نتیجه را یک مسیر $x - y$ می نامند.

تعریف ۵.۱: (گراف همبند) گراف G را همبند گوییم اگر به ازای هر دو راس متمایز u و v در V، مسیری در G چنان موجود باشد که u و v را به هم وصل کند.

تعریف ۶.۱: (طوقه) برای هر یال مثل (b, c) می گوییم که این یال از راس های b و c می گذرد؛ b را مجاور به c می گویند. یال (b, c) یک طوقه نامیده می شود اگر $b = c$ باشد.

تعریف ۷.۱: (درخت) فرض کنید $G(V,E)$ گراف بدون جهت و بی طوقه ای باشد. گراف G را درخت می نامند اگر همبند بوده و شامل هیچ دوری نباشد.

تعریف ۸.۱: (درخت جهت دار) اگر G گرافی جهت دار با شد آن گاه G را درخت جهت دار می نامند اگر گراف بدون جهت مربوط به G، یک درخت (گراف همبند بدون طوقه و دور) باشد.

تعریف ۹.۱: فرض کنید $G(V,E)$ گرافی جهت دار باشد، برای هر $v \in V$،
الف) درجه ورودی v تعداد یال های G است که به v می رسند و آن را با $deg^-(v)$ نشان می دهیم.
ب) درجه خروجی v تعداد یال های G است که از v خارج می شوند و آن را با $deg^+(v)$ نشان می دهیم.

تعریف ۱۰.۱: (درخت ریشه دار) درخت جهت دار G را درخت ریشه دار می نامند اگر یک راس یکتای r، به نام ریشه در G با درجه $r = deg^-(r) = 0$ وجود داشته باشد و برای همه راس های دیگر V، $deg^-(v)$، درجه درونی v، برابر ۱ باشد.

تعریف ۱۱.۱: (سطح) فرض کنید $G(V,E)$ درخت جهت دار با گره ریشه $r \in V$ باشد. می گوییم $v \in V$ در سطح n قرار دارد اگر مسیری از v به r با طولی به تعداد n یال وجود داشته باشد.

تعریف ۱۲.۱: اگر u_1 و u_2 راس هایی از درخت ریشه دار باشند و u_1 دارای شماره سطح کوچکتری باشد، آنگاه u_1 نیای u_2 (یا u_2 زاده u_1) است اگر مسیری از u_1 به u_2 موجود باشد.

۱-۲-۱ روش های جستجو

روش جستجوی نخستین ژرفا: فرض کنید $G(V,E)$ درخت بدون جهت و همبندی با $r \in V$ باشد. با شروع از r مسیری در G می سازیم. اگر این مسیر شامل همه راس های V باشد، آن

گاه مسیر، درخت فراگیر T برای G است و کار تمام است. اگر نه، فرض کنید x و y دو راس آخری از مسیر باشند که به آنها می رسیم و y آخرین راس باشد. فرض کنید گره قبل از گره x که نسبت به این گره به گره ریشه نزدیک تر است، گره p باشد. در این صورت به راس x برمی گردیم یا عقب نشینی می کنیم و مسیر دومی در G می سازیم که با x شروع شود و شامل هیچ راسی که قبلا ملاقات شده است نباشد. اگر چنین مسیری وجود نداشته باشد، به پدر pی x عقب نشینی می کنیم و می بینیم که کجا ممکن است از p منشعب شود و مسیری (با راس هایی که قبلا به آنها رسیده ایم) با برگ جدید y_1 بسازد. اگر تمام یال های موسوم از راس p به راس هایی که قبلا با آنها برخورد شده است منتهی شوند، به یک سطح بالاتر عقب نشینی می کنیم و فرایند را ادامه می دهیم. چون گراف متناهی و همبند است، این تکنیک، نهایتا درخت فراگیر T را برای G تعیین می کند، که در آن r به عنوان ریشه T در نظر گرفته می شود. در این صورت با استفاده از T، راس های G را در فهرست پیش ترتیب مرتب می کند و با توجه به این فهرست، گره ها ی درخت جستجوی روش شاخه و کران را پردازش می کنیم.

روش جستجوی نخستین پهنا: فرض کنید $G(V,E)$ درخت بدون جهت و همبند باشد. در اینجا راس را به عنوان ریشه تعیین و همه راس های مجاور را به صورت بادبزنی به ریشه وصل می کنیم. سپس از هر فرزند ریشه به صورت بادبزنی به آن راس هایی که (قبلا به آن نرسیده ایم) مجاور یکی از فرزندان اند، وصل می کنیم. وقتی که این فرایند را ادامه می دهیم هرگز راسی را دوبار ثبت نمی کنیم، لذا دوری تشکیل نمی شود و چون G متناهی است، سرانجام فرایند خاتمه می یابد.

۱–۳ برنامه ریزی خطی

برای یک مجموعه از متغیر های حقیقی مقدار داده شده، برنامه ریزی خطی یک مساله بهینه سازی است که یک تابع هدف خطی را با توجه به بعضی نامساوی ها و یا تساوی های خطی

کمینه یا بیشینه می کند. با تغییر شکل می توانیم هر برنامه ریزی خطی را به فرمی که در تعریف زیر ارایه شده است، تبدیل کنیم. مساله برنامه ریزی خطی (LP) را به صورت زیر تعریف می کنیم.

$$\begin{aligned} Min \quad & c^T\mathbf{x}, \\ s.to. \quad & A\mathbf{x} \leq b, \\ & \mathbf{x} \in R^n_+ \end{aligned} \qquad (۱-۱)$$

که در آن داریم: $m, n \in \mathbb{N}$ ، $c \in \mathbb{R}^n$، $A \in \mathbb{R}^{m\times n}$ و $b \in \mathbb{R}^m$.

تعریف ۱۳.۱: (ترکیب محدب) فرض کنید که $\lambda_۱ x_۱ + \lambda_۲ x_۲ + \ldots + \lambda_n x_n$ یک ترکیب خطی از فضای برداری $\mathbb{R}^n$ باشد. در این صورت این ترکیب خطی را، یک ترکیب محدب گوییم، هرگاه در آن

$$\sum_{i=۱}^{n} \lambda_i = ۱, \lambda_i \geq ۰, i = ۱, ۲, \ldots, n.$$

تعریف ۱۴.۱: (پوسته محدب) برای مجموعه $X \subseteq \mathbb{R}^n$، پوسته محدب X به صورت $Conv(X)$ تعریف می شود:

$$Conv(X) = \{\mathbf{x} : \mathbf{x} = \sum_{t=۱}^{T} \lambda_t \mathbf{x}_t, \sum_{t=۱}^{T} \lambda_t = ۱, \lambda \geq ۰, t = ۱, ۲, ..., T\}$$

که در آن ترکیب محدب روی همه زیر مجموعه های متناهی $\{\mathbf{x}_۱, \mathbf{x}_۲, ..., \mathbf{x}_T\}$ از X می باشد[۲۷].

تعریف ۱۵.۱: (چندوجهی)[۱۹] فرض کنید $n \in N$،

- چندوجهی $P \subseteq \mathbb{R}^n$ مجموعه نقاطی است که در تعداد متناهی $m \in \mathbb{N}$ از نامساوی های خطی

صدق می کند؛ یعنی $P = \{\mathbf{x} \in \mathbb{R}^n |\ A\mathbf{x} \geq b\}$ که (A, b) یک ماتریس $m \times (n+1)$ است.

• چند وجهی که ضرایب آن گویا باشد، یک چند وجهی گویا خوانده می شود.

•یک چند وجهی کراندار خوانده می شود اگر $w \in R_+$ وجود داشته باشد به طوری که $P \subseteq \{\mathbf{x} \in \mathbb{R}^n | -w < x_j < w, j = 1, \ldots, n\}$. یک چندوجهی کران‌دار را چندسقفی گوییم.

قضیه ۱.۱۶: (قضیه نمایش)[4] فرض کنید $X = \{\mathbf{x} | A\mathbf{x} \leq b, \mathbf{x} \geq 0\}$ یک چند وجهی ناتهی باشد. در این صورت مجموعه نقاط راسی X ناتهی و دارای تعداد متناهی نقطه، نظیر، $p_1, p_2, \ldots, p_T$ است. به علاوه، مجموعه جهت های راسی تهی است اگر و تنها اگر X کراندار باشد. اگر X بیکران باشد، آن گاه مجموعه جهت های راسی ناتهی است و دارای تعداد متناهی بردار، نظیر، $r_1, r_2, \ldots, r_L$ است. گذشته از این، $x \in X$ اگر و فقط اگر آن را بتوان به صورت ترکیب محدبی از $p_1, p_2, \ldots, p_T$ به علاوه ترکیب نامنفی از $r_1, r_2, \ldots, r_L$ نوشت، یعنی:

$$\mathbf{x} = \sum_{t=1}^{T} \lambda_t p_t + \sum_{j=1}^{L} r_j \mu_j, \quad \sum_{t=1}^{T} \lambda_t = 1, \quad \lambda_t, \mu_j \geq 0, 0 \leq t \leq T, 0 \leq j \leq L.$$

قضیه ۱.۱۷: [18] اگر $X = P \cap \{\mathbf{x} \in \mathbb{Q}_+^n |\ A\mathbf{x} \geq b, x_i \in \mathbb{Z}, i \in I\} \neq \emptyset$ و P یک چند وجهی گویا باشد آنگاه پوسته محدب X یک چند وجهی گویاست که نقاط فرین آن زیر مجموعه X می باشد و جهت های فرین آن جهت های فرین P است.

در این پایان نامه بدون کاستن از کلیت، توجه خود را به چند وجهی ها، ماتریس ها و بردار های با مقادیر گویا محدود می کنیم.

۱-۳-۱ روش سیمپلکس متغیر های کراندار

در این بخش توضیح مختصری از روش سیمپلکس متعیر های کراندار ارایه می شود. مساله برنامه ریزی خطی زیر را در نظر بگیرید:

$$Min \qquad c^T \mathbf{x},$$

$$s.to. \qquad A\mathbf{x} \leq b,$$

$$l \leq \mathbf{x} \leq u,$$

که در آن A یک ماتریس $m \times n$ با رتبه m است. جواب$\bar{\mathbf{x}}$ معادلات $A\mathbf{x} \leq b$ جواب پایه این سیستم است اگر A را بتوان به صورت $[B, N_۱, N_۲]$ افراز کرد که در آن ماتریس B دارای رتبه m است به طریقی که با افراز $\mathbf{x}$ به صورت $(\mathbf{x}_B, \mathbf{x}_{N_۱}, \mathbf{x}_{N_۲})$ داریم $\bar{\mathbf{x}}_{N_۲} = u_{N_۲}$، $\bar{\mathbf{x}}_{N_۱} = l_{N_۱}$، $\bar{\mathbf{x}_B} = B^{-۱}b - B^{-۱}N_۱ l_{N_۱} - B^{-۱}N_۲ u_{N_۲}$. به علاوه، اگر $l_B \leq \bar{\mathbf{x}_B} \leq u_B$، به $\bar{\mathbf{x}}$ جواب شدنی پایه گفته می شود. می دانیم که در روش سیمپلکس پس از یافتن یک جواب پایه ای شدنی برای مساله اگر این جواب برای مساله بهینه نبود باید به دنبال متغیری باشیم که با ورودش به پایه مقدار تابع هدف را بهبود بخشد. متغیر ورودی را با استفاده از رابطه زیر تعیین می کنیم:

$$max\,(max\{z_j - c_j | j \in R_۱\}, max\{c_j - z_j | j \in R_۲\}),$$

که در آن $R_۱$ مجموعه اندیس های متغیر های غیر پایه است که در کران پایین شان هستند، و $R_۲$ مجموعه اندیس های متغیر های غیر پایه ای است که در کران بالایشان هستند. اگر این مقدار بیشینه مثبت شد k را اندیس مقدار بیشینه می گیریم. اگر k در $R_۱$ قرار داشت، آن گاه x_k را از مقدار فعلی اش یعنی l_k افزایش می دهیم و اگر k در $R_۲$ بود x_k را از مقدار فعلی اش یعنی u_k کاهش می دهیم. اگر x_k از مقدار فعلی اش افزایش یابد، سه حالت پیش می آید. حالت اول این است که با افزایش x_k یک متغیر پایه به کران پایین اش می رسد. در حالت دوم، با افزایش این متغیر غیر پایه ای یک متغیر پایه ای به کران بالایش می رسد و یا در حالت سوم، خود متغیر x_k به کران بالایش می رسد. فرض کنید $x_k = l_k + \Delta_k$ که در آن Δ_k مقدار افزایش x_k است. مقداری از Δ_k که به ازای آن یک متغیر پایه به کران پایین اش برسد را با $\gamma_۱$ نشان می دهیم.

مقدار Δ_k را که به ازای آن یک متغیر پایه به کران بالایش می رسد را با $\gamma_۲$ نشان می دهیم. می توان نشان داد که $\gamma_۱$ و $\gamma_۲$ با استفاده از روابط زیر بدست می آیند [۴]:

$$\gamma_۱ = \begin{cases} min\{\frac{\hat{b_i}-l_{B_i}}{y_{ik}} : y_{ik} < ۰ \mid ۱ \geq i \leq m\}, & y_k > ۰; \\ \infty, & y_k \leq ۰. \end{cases}$$
$$\gamma_۲ = \begin{cases} min\{\frac{u_{B_i}-\hat{b_i}}{-y_{ik}} : y_{ik} < ۰ \mid ۱ \geq i \leq m\}, & y_k < ۰; \\ \infty, & y_k \geq ۰. \end{cases}$$

مقداری از Δ_k که به ازای آن x_k به کران بالایش u_k می رسد به وضوح $u_k - l_k$ است. واضح است که Δ_k عبارت است از:

$$\Delta_k = min\{\gamma_۱, \gamma_۲, u_k - l_k\}.$$

جواب فعلی باید به هنگام شود تا جواب شدنی پایه جدید را معرفی کند. اگر $\Delta_k = u_k - l_k$، آن گاه تغییر اساسی در پایه داده نمی شود و x_k هنوز غیر پایه ای است؛ به جز این که در این شرایط، در کران بالایش قرار می گیرد. در این حالت تنها ستون سمت راست تغییر می کند که باعث تغییر مقدار تابع هدف و مقادیر متغیر های پایه می شود. در غیر این صورت، متغیر خروجی را با توجه به Δ_k انتخاب می کنیم. فرض می کنیم r اندیس متغیر خروجی باشد. در هر سه حالت، مقادیر جدول با استفاده از روابط زیر به هنگام می شوند:

$$\mathbf{x}_B = \hat{b} - y_k\Delta_k; \qquad (۲-۱)$$

$$z = \hat{z} - (z_k - c_k)\Delta_k. \qquad (۳-۱)$$

اگر x_k را از مقدار فعلی اش کاهش دهیم، روابط مشابهی بدست می آید (برای مطالعه بیشتر به [۴] مراجعه شود). برای به هنگام کردن جدول سیمپلکس سه عمل لازم است؛ اول، ستون متغیر

غیر پایه ورودی را در مقدار فعلی اش (u_k یا l_k) ضرب می کنیم و حاصل را با بردار سمت راست جمع می کنیم. سپس، ستون متغیر خروجی را در مقداری که می گیرد (u_{B_r} یا l_{B_r}) ضرب می کنیم و حاصل را از سمت راست کم می کنیم. سرانجام، یک عمل محور گیری طبیعی روی بردار تعدیل یافته سمت راست انجام می دهیم.

۴–۱ مساله برنامه‌ریزی صحیح آمیخته

در این بخش، به شرح مدل یک مساله برنامه‌ریزی صحیح آمیخته (MIP) می‌پردازیم و بیان می‌کنیم که چگونه با اعمال یک تغییر جزیی در دامنه یک مساله برنامه‌ریزی خطی (فضای جواب)، این دسته از مسائل بهینه سازی، حاصل می‌شوند. طبیعی است که این تغییر در ساختار فضای جواب، سبب تغییر در کیفیت جواب‌ها و زمان حل مساله می‌گردد. فرض کنید که x بردار متغیرهای تصمیم یک مساله برنامه‌ریزی خطی به صورت (۱–۱) باشد. اگر دامنه بعضی از متغیرها را طوری تغییر دهیم که صحیح شوند، آن گاه یک مساله به صورت

$$\begin{aligned} &Min \qquad c^T\mathbf{x}, && (۴–۱)\\ &s.to. \qquad A\mathbf{x} \leq b, \qquad (MIP) &&\\ &\qquad\qquad \mathbf{x} \in \mathbb{R}^n_+,\ x_i \in \mathbb{Z}, i \in I. && \end{aligned}$$

بدست می آید که یک مساله برنامه‌ریزی صحیح آمیخته نامیده می شود. مسایل برنامه ریزی صحیح آمیخته در حالت بیشینه سازی را می توان با ضرب بردار هزینه در عدد منفی یک به مساله کمینه سازی تبدیل کرد. بنابراین در ادامه بدون کاستن از کلیت فرض می کنیم مساله به شکل کمینه سازی است. اگر در مساله برنامه ریزی صحیح آمیخته به شکل (۴–۱) داشته باشیم $I = ۱, ۲, ..., n$ آنگاه آن را یک برنامه ریزی صحیح می نامند. می دانیم که مجموعه

جواب های برنامه ریزی صحیح آمیخته، یعنی $X = \{\mathrm{x} \in \mathbb{R}_+^n |\ A\mathrm{x} \leq b, x_i \in \mathbb{Z}, i \in I\}$ زیر مجموعه چند وجهی $P = \{\mathrm{x} \in \mathbb{R}_+^n |\ A\mathrm{x} \leq b\}$ است. P را مجموعه ی جواب های برنامه ریزی خطی سست شده می نامند؛ یعنی آن مساله برنامه ریزی خطی که از حذف محدودیت های صحیح بودن برنامه ریزی صحیح آمیخته بدست آمده است. به دلیل وجود محدودیت های مربوط به صحیح بودن برخی متغیر ها، مجموعه X یک چند وجهی نیست. با استفاده از قضیه زیر که در مرجع [۱۸] اثبات شده است، می توان نشان داد که پوسته محدب X یک چند وجهی است. تاکنون الگوریتم های موثری برای حل مسایل برنامه ریزی صحیح و برنامه ریزی صحیح آمیخته ارایه شده است [۲۳، ۱۴]. با پیشرفت های چشمگیر اخیر درمورد نرم افزار و سخت افزار رایانه های موجود روش شاخه و کران قادر به حل مسایل با اندازه بزرگتری نسبت به قبل می باشد. از طرفی بسیاری از روش های موثری که در این مورد ارایه شده اند مانند روش شاخه و برش و روش شاخه و هزینه از ایده های روش شاخه و کران نشات گرفته اند. با توجه به این مطالب این روش را در بخش بعد شرح می دهیم.

۱–۵ روش شاخه و کران

فرایند شاخه و کران را می توان توسط درختی نشان داد که هر زیر مساله از این فرایند متناظر با یک گره در درخت است. گره متناظر با مساله اصلی را می توان به عنوان ریشه درخت در نظر گرفت. هر زمان که یک مساله به دو زیر مساله تقسیم می شود، این زیر مساله ها گره های فرزند گره فعلی به حساب می آیند. فرض کنید X مجموعه جواب های یک مساله داده شده باشد و $s \in X$ یک جواب خوب باشد که با روش های ابتکاری بدست آمده است. منظور از جواب خوب جوابی است که مقدار تابع هدف به ازای آن به مقدار تابع هدف بهینه نزدیک باشد. زمانی که یک جواب جدید بهتر پیدا می شود آن را جای s جایگذاری می کنیم پس همواره s یک کران بالای کلی روی مقدار بهینه است.