دانشکده علوم ریاضی
دانشگاه صنعتی شریف

تحقیق در عملیات ۱

محمدهادی فروغمنداعرابی
پاییز ۱۳۹۹

ادامه سیمپلکس عمومی

جلسه هشتم

نگارنده: الهه قاسمی

۱ مروری بر مباحث گذشته

در جلسه قبل روش سیمپلکس را آموختیم. یک برنامه ریزی خطی به صورت

$$\begin{aligned} \text{بیشینه کن} \quad & c'x \\ \text{که} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

را در نظر بگیرید. روش سیمپلکس برای حل این برنامه‌ریزی خطی دنباله‌ای از تابلوهای سیمپلکس را محاسبه می‌کند. هر تابلو مربوط به یک جواب شدنی با پایه B است و یک جواب شدنی پایه‌ای را تعیین می‌کند. یک تابلو به صورت یک دستگاه معادلات خطی است که در آن متغیرهای پایه و متغیر z که نشان‌دهنده‌ی مقدار تابع هدف است، در سمت چپ تابلو قرار داشته و آن‌ها به صورت معادلات آفینی از متغیرهای غیرپایه‌ای نشان داده می‌شوند.

simplex tableau $\mathcal{T}(B)$

$$\begin{array}{lllll} \mathbf{x}_B & = & \mathbf{p} & + & Q\,\mathbf{x}_N \\ \hline z & = & z_0 & + & \mathbf{r}^T\mathbf{x}_N \end{array}$$

که در آن جواب پایه‌ای شدنی و مقدار بیشینه‌ی تابع هدف به صورت زیر بدست می‌آیند.

$$X_N = 0$$
$$X_B = p$$
$$z = z_0$$

در این روش در هر گام لولا یکی از متغیرهای غیرپایه مثل x_v را که در سطر آخر تابلو ضریب مثبت دارند انتخاب کرده و وارد پایه می‌کردیم. سپس معادلات را بررسی کرده و معادله‌ای را که افزایش x_v را محدودتر می‌کند انتخاب می‌کنیم و متغیر پایه‌ی x_u مربوط به آن را از پایه حذف می‌کنیم.

$$\mathcal{T}(B) \longrightarrow \mathcal{T}(B')$$
$$B' = (B \setminus \{u\}) \cup \{v\}$$

$$x_v \text{ enters the basis}$$
$$x_u \text{ leaves the basis}$$

همچنین نشان دادیم که جواب بهینه زمانی بدست می‌آید که داشته باشیم $r \leq 0$. یعنی هیچ‌کدام از متغیرهای سطر آخر ضریب مثبت نداشته باشند و قابل وارد شدن به پایه نباشند. یکی از چالش‌ها در استفاده از این روش پیدا کردن اولین تابلو است. در مثال‌های گذشته که برنامه‌ریزی خطی به صورت نامعادله‌ای $Ax \leq b$ بود، برای پیدا کردن یک جواب اولیه، بعد از تبدیل برنامه‌ریزی به فرم معادله‌ای آن با استفاده از m متغیر جدید، به سادگی $x = 0$ قرار داده و باقی متغیرها را برابر b (یا به عبارتی اختلاف متغیرهای اصلی از مقدار b مربوط به آن نامعادله) قرار می‌دادیم. اما پیدا کردن یک جواب اولیه برای برنامه‌ریزی های فرم معادله‌ای کار ساده‌ای نیست. درواقع این کار به سختی حل کردن یک برنامه‌ریزی خطی است. در ادامه راهی برای پیدا کردن جواب اولیه ارائه می‌دهیم و قانون لولا و زمان اجرای روش سیمپلکس را بررسی می‌کنیم.

۲ پیدا کردن جواب اولیه

قصد داریم برای برنامه‌ریزی خطی به صورت

$$\begin{aligned} \text{بیشینه کن} \quad & c'x \\ \text{که} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

یک جواب اولیه پیدا کنیم. در ابتدا اگر $b \leq 0$ بود، معادلات را در یک منفی ضرب می‌کنیم تا b مثبت شود. سپس m متغیر جدید از x_{n+1} تا x_{n+m} تعریف می‌کنیم. هرکدام از x_i ها اختلاف مقدار معادله‌ی iام از مقدار b مربوط به آن را نشان می‌دهد. در نتیجه برای پیدا کردن یک جواب شدنی قصد داریم برنامه ریزی خطی بنویسیم که مقدار این متغیرها را کمینه کند. فرم این برنامه‌ریزی به صورت زیر است:

$$\begin{aligned} \text{subject to} \quad & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

$$\mathbf{b} \geq \mathbf{0}$$

$$\overline{\mathbf{x}} = (x_1, \ldots, x_{n+m})$$
$$\bar{A} = (A \,|\, I_m)$$

$$\begin{aligned} \bar{A}\,\overline{\mathbf{x}} &= \mathbf{b} \\ \overline{\mathbf{x}} &\geq \mathbf{0} \end{aligned}$$

درنتیجه حالا m معادله و $m+n$ متغیر داریم. پیدا کردن یک جواب اولیه برای این برنامه‌ریزی خطی کار ساده‌ای است. برای این کار قرار می‌دهیم: $x_1 = x_2 = \ldots = x_n = 0$ و متغیر های x_{n+1} تا x_{n+m} را برابر با مقدار مربوط به آن معادله از بردار b قرار می‌دهیم. حالا برای پیدا کردن جواب اولیه برای برنامه‌ریزی خطی اصلی، مجموع متغیرهای جدید را کمینه می‌کنیم. اگر این مقدار به صفر برسد، آنگاه جوابی شدنی برای مسئله‌ی اصلی داریم چرا که مقدار اختلاف هر معادله از مقدار b مربوط به آن به صفر رسیده است. اما اگر این مقدار مخالف صفر باشد یعنی هیچ جوابی برای این برنامه‌ریزی خطی وجود ندارد.

بعد از اجرای الگوریتم سیمپلکس روی برنامه‌ریزی خطی جدید در تابلوی نهایی آن، مقدار متغیرهای اضافی x_{n+1} تا x_{n+m} برابر صفر بوده و آن‌ها را از تابلو حذف کنید و مقدار z را برابر با مقدار تابع هدف قرار دهید تا به تابلوی اولیه برای برنامه‌ریزی خطی اصلی برسید.

$$\begin{array}{ll} \text{maximize} & -(x_{n+1}+x_{n+2}+\cdots+x_{n+m}) \\ \text{subject to} & \bar{A}\,\overline{\mathbf{x}} = \mathbf{b} \\ & \overline{\mathbf{x}} \geq \mathbf{0}, \end{array}$$

۳ قانون لولا

قانون لولا، قانونی برای انتخاب یکی از متغیرهای واجد شرایط برای وارد کردن به پایه است. برای این کار چند پیشنهاد وجود دارد. مثلا انتخاب متغیر با بزرگترین ضریب[1] در سطر آخر سیمپلکس (یعنی در بردار r). پیشنهاد دیگر این است که متغیری را انتخاب کنیم که به بیشترین مقدار ممکن بتوان آن را زیاد کرد[2]. پیشنهاد دیگر این است که متغیری را انتخاب کنیم که یالی را هدف قرار می‌دهد که این یال در چندوجهی مورد نظر بیشترین مقدار ضرب داخلی در تابع هدف(کمترین زاویه با تابع هدف) را دارد.[3] پیشنهاد دیگر که قانون بلند نام دارد، این است که از بین متغیرهایی که ضریب آن‌ها مثبت است، متغیری را انتخاب کنیم که شماره اندیس آن کمتر از بقیه است و برای خارج کردن یک متغیر نیز از متغیر با اندیس کمتر استفاده کنیم.[4]. پیشنهاد آخر این است که یکی از یال‌ها را به صورت تصادفی انتخاب کنیم.[5] در ادامه قانون بلند را بررسی می‌کنیم و ادعا می‌کنیم که با استفاده از آن هرگز در دور نمی‌افتیم.

۴ قانون بلند

همان‌طور که گفته شد قانون بلند یک قانون لولا است که در آن هربار برای انتخاب یک متغیر ورودی به پایه از متغیری با کمترین اندیس در میان متغیرهای غیرپایه‌ای استفاده می‌کنیم و برای خارج کردن یک متغیر از پایه نیز از متغیر واجد شرایط با کمترین اندیس ممکن استفاده می‌کنیم.

قضیه : در اجرای روش سیمپلکس با استفاده از قانون بلند هیچ‌گاه در دور نمی‌افتیم و اجرای آن در متناهی مرحله پایان می‌پذیرد.

اثبات: فرض خلف: در اجرای سیمپلکس با استفاده از قانون بلند در دور افتاده‌ایم. در دور افتادن یعنی اینکه در یک مرحله از اجرای الگوریتم که در آن پایه B است و تابلوی مربوط به آن نیز $\mathcal{T}(B)$ است و در مرحله‌ای دیگر نیز به همین پایه برمی‌گردیم. فرض کنید مجموعه F شامل اندیس مربوط به همه‌ی متغیرهایی است که حداقل یک بار در طور دور وارد(و درنتیجه خارج) پایه می‌شوند. این متغیرها را متغیرهای بی‌ثبات[6] می‌نامیم. ابتدا یک ادعا درباره‌ی دور داشتن در اجرای سیمپلکس می‌کنیم که درمورد هر قانون لولایی صادق است.

ادعا: تمام پایه‌هایی که در دور ساخته می‌شوند به یک جواب شدنی پایه‌ای یکسان می‌رسند و تمام متغیرهای بی‌ثبات در آن‌ها برابر با ۰ هستند.

اثبات ادعا: چون مقدار تابع هدف(z) هرگز کاهش پیدا نمی‌کند درنتیجه باید همواره ثابت بماند تا به حالت اول برگردیم. فرض کنید B یک پایه‌ی شدنی باشد که در طول دور پیدا کرده‌ایم و داشته باشیم:

$$N = \{1, 2, \ldots, n\} \backslash B$$

و برای پایه‌ی بعدی آن داشته باشیم:

$$B' = (B \backslash \{u\}) \cup \{v\}$$

تنها متغیر غیرپایه‌ای که می‌تواند تغییری در مقدار z در اجرای گام لولا از B به B' ایجاد کند متغیر ورودی x_v است. چرا که بقیه‌ی متغیرهای غیرپایه‌ای، غیرپایه‌ای و در نتیجه ۰ باقی می‌مانند. همچنین می‌دانیم که این متغیر ضریب مثبتی در سطر آخر $\mathcal{T}(B)$ دارد. و از آن جایی که مقدار z

[1] Largest Coefficient
[2] Largest Increase
[3] Steepest Edge
[4] Bland's Rule
[5] Random Edge
[6] fickle

از رابطه‌ی $z = z_\circ + r^T X_N$ بدست می‌آید، اگر مقدار این متغیر مثبت باشد منجر به افزایش مقدار z می‌شود. در نتیجه دو جواب شدنی B و B' در مقادیر مربوط به متغیرهای N برابر هستند. در نتیجه این دو جواب شدنی پایه‌ای برابر هستند. در نهایت چون هر متغیر بی‌ثبات حداقل یک بار در طول دور غیرپایه‌ای می‌شود، باید همواره برابر با ۰ باشد. و ادعا ثابت می‌شود.

حال برای ادامه اثبات قضیه بزرگترین اندیس در F را در نظر بگیرید و آن را v بنامید. فرض کنید B پایه‌ای در دور در لحظه‌ای که x_v وارد می‌شود باشد. همچنین فرض کنید B' پایه‌ای از دور در لحظه‌ای که x_v خارج می‌شود(و مثلا x_u وارد می‌شود) باشد. و فرض کنید $p, Q, r, z_\circ$ پارامترهای تابلوی $\mathcal{T}(B)$ باشند و $p', Q', r', z'_\circ$ پارامترهای تابلوی $\mathcal{T}(B')$ باشند. حال ابتدا مجموعه‌های B و N را به صورت مجموعه‌های مرتب زیر می‌نویسیم:

$$B = \{k_1, K_2, ..., K_m\}$$

$$K_1 \leq K_2 \leq ... \leq K_m$$

$$N = \{1, 2, ..., n\} \backslash B$$

$$N = \{l_1, l_2, ..., l_{n-m}\}$$

$$l_1 \leq l_2 \leq ... \leq l_{n-m}$$

چون v بزرگترین اندیس در مجموعه F است و قانون بلند کوچکترین اندیس ممکن را انتخاب می‌کند، در نتیجه سایر متغیرهای بی‌ثبات در این لحظه کاندیدای انتخاب کردن نیستند و درنتیجه ضریب آن‌ها در سطر آخر تابلو نامثبت است. به عبارتی دیگر داریم:

$$r_\beta \geq 0 \text{ and } r_j \leq 0 \text{ for all j such that } l_j \in F \backslash \{v\} \ (1)$$

حال لحظه‌ای را درنظر بگیرید که می‌خواهیم v را خارج کنیم. مشابه حالت قبل برای پارامترهای تابلوی $\mathcal{T}(B')$ داریم:

$$B' = \{k'_1, K'_2, ..., K'_m\}$$

$$K'_1 \leq K'_2 \leq ... \leq K'_m$$

$$N' = \{1, 2, ..., n\} \backslash B'$$

$$N' = \{l'_1, l'_2, ..., l'_{n-m}\}$$

$$l'_1 \leq l'_2 \leq ... \leq l'_{n-m}$$

فرض کنید α' اندیس متغیر خروجی x_v در B' باشد. و β' اندیس مربوط به متغیر ورودی x_u در N' باشد.

$$k'_{\alpha'} = v, l'_{\beta'} = u$$

چون v بزرگترین اندیس در مجموعه F است و قانون بلند کوچکترین اندیس ممکن را انتخاب می‌کند، در نتیجه سایر متغیرهای بی‌ثبات در این لحظه کاندیدای حذف شدن نیستند. طبق ادعا در این لحظه p متناظر با همه‌ی متغیرهای بی‌ثبات برابر با ۰ است. درنتیجه از این نظر با هم تفاوتی ندارند و چیزی که باعث می‌شود فقط v کاندیدا باشد این است که بقیه آن‌ها محدودیتی برای ورود x_u ایجاد نمی‌کنند. یعنی ضریب x_u در معادله مربوط به آن‌ها منفی نیست و ضریب x_u در معادله مربوط به x_v منفی است. به عبارتی دیگر داریم:

$$q'_{\alpha'\beta'} \leq 0 \text{ and } q'_{i\beta'} \geq 0 \text{ for all i such that } k'_i \in F \backslash \{v\} \ (2)$$

می‌خواهیم از دو نتیجه‌ی بالا استفاده کنیم و یک برنامه‌ریزی خطی بنویسیم که این جواب برای آن هم بهینه باشد و هم کران نداشته باشد. که این به وضوح یک تناقض است.

برنامه‌ریزی خطی زیر را در نظر بگیرید:

$$\begin{array}{ll} \text{Maximize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x}_{F\backslash\{v\}} \geq \mathbf{0} \\ & x_v \leq 0 \\ & \mathbf{x}_{N\backslash F} = \mathbf{0}. \end{array}$$

ادعا می‌کنیم که جواب پایه‌ای شدنی $\widetilde{x}$ مربوط به برنامه‌ریزی خطی اصلی با پایه‌ی B یک جواب شدنی برای این برنامه‌ریزی خطی است. زیرا طبق ادعا داریم $\widetilde{x}_N = \circ, \widetilde{x}_F = \circ$. همچنین به ازای هر x که در $Ax = b$ صدق کند، مقدار تابع هدف به صورت زیر نشان داده می‌شود:

$$c^T x = z = z_\circ + r^T x_N$$

به ازای همه‌ی جواب‌های شدنی برای برنامه‌ریزی خطی جدید داریم:

$$x_{\ell_j} \begin{cases} \geq 0 & \text{if } \ell_j \in F \setminus \{v\} \\ \leq 0 & \text{if } \ell_j = \ell_\beta = v, \end{cases}$$

و از عبارت (۱) داریم:

$$r_j x_{l_j} \leq 0 \text{ for all j such that } l_j \in F$$

حال چون $x_{N\setminus F} = \circ$ در نتیجه داریم $r^T x_N \leq \circ$ و در نتیجه $z \leq z_\circ$. پس نتیجه می‌گیریم که $\widetilde{x}$ یک جواب بهینه برای این برنامه‌ریزی خطی است.

از ادعاهای قبلی می‌دانیم که $\widetilde{x}$ ساخته شده از $B^{'}$ یک جواب شدنی پایه‌ای برای برنامه‌ریزی خطی اصلی است. به ازای همه‌ی جواب‌های x برای $Ax = b$ داریم:

$$x_{B'} = p^{'} + Q^{'} X_{N'} (3)$$

حالا با میل دادن مقدار $\widetilde{x}_u$ از $\circ$ به مقدار $t > \circ$ ، $\widetilde{x}_{N'}$ را تغییر می‌دهیم. این یک جواب جدید $\widetilde{x}(t)$ برای $Ax = b$ بدست می‌دهد. نشان می‌دهیم که برای همه‌ی $t > \circ$ این جواب، یک جواب شدنی پایه‌ای برای برنامه‌ریزی خطی جدید است و مقدار تابع هدف $c^T\widetilde{x}(t)$ در صورتی که داشته باشیم $t \to \infty$ به بی‌نهایت میل می‌کند.

قرار می‌دهیم:

$$\tilde{x}_{\ell'_j}(t) := \begin{cases} 0 & \text{if } \ell'_j \in N' \setminus u \\ t & \text{if } \ell'_j = \ell'_{\beta'} = u. \end{cases}$$

اگر $\widetilde{x} = \circ$ و $t > \circ$ با توجه به (۳) و (۲) داریم:

$$\tilde{x}_{k'_i}(t) = \tilde{x}_{k'_i} + tq'_{i\beta'} \begin{cases} \geq 0 & \text{if } k'_i \in F \setminus \{v\} \\ < 0 & \text{if } k'_i = k'_{\alpha'} = v. \end{cases}$$

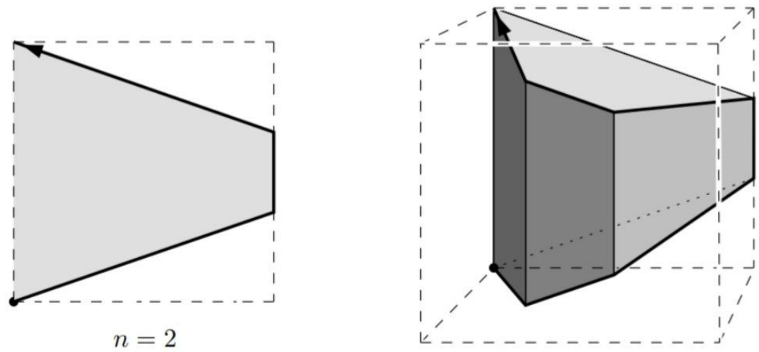
توجه کنید که $\widetilde{x}(t)$ هنوز هم جوابی شدنی برای برنامه‌ریزی خطی جدید است. چون متغیر $x_u = x_{l'_{\beta'}}$ یک کاندیدا برای وارد شدن به پایه‌ی $B^{'}$ بود، می‌دانیم که $r^{'}_{\beta'} > \circ$ در نتیجه داریم:

$$\mathbf{c}^T\tilde{\mathbf{x}}(t) = z'_0 + \mathbf{r}'^T\tilde{\mathbf{x}}_{N'}(t) = z'_0 + tr'_{\beta'} \to \infty \quad \text{for } t \to \infty.$$

این یعنی برنامه‌ریزی خطی جدیدمان کران ندارد. پس فرض خلف باطل است و قضیه اثبات می‌شود. □

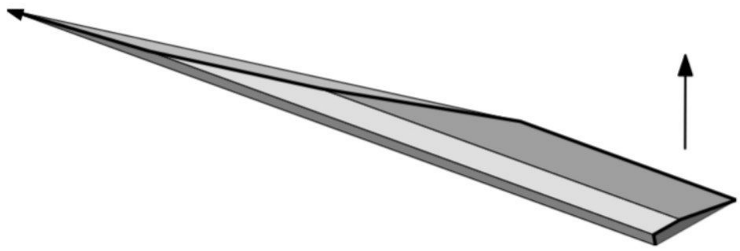
۵ آیا روش سیمپلکس سریع است؟

در عمل سیمپلکس بین $۲m$ تا $۳m$ بار گام لولا را اجرا می‌کند. این عدد بسیار خوبی است اما بعدها توانستند مثال‌هایی ارائه دهند که در آن‌ها برای رسیدن به جواب، همه‌ی رأس‌های چندوجهی طی می‌شوند. مطابق شکل زیر که در آن برای رسیدن به هدف از همه‌ی یال‌های مشخص شده عبور می‌کنیم. این یعنی مثال‌هایی از روش سیمپلکس وجود دارد که در آن زمان اجرای الگوریتم نمایی است.



$n = 2$ $n = 3$

درواقع شکل مثال بالا به صورت زیر است:



اثبات می‌شود که سیمپلکس حداکثر $e^{C\sqrt{n \ln n}}$ بار لولا می‌زند و این کران بالای مطلوبی نیست.

حالا فرض کنید الگوریتمی برای اجرای سیمپلکس داشته باشیم که در انتخاب یال‌ها، هربار کوتاهترین مسیر ممکن را پیدا کند. سوالی که مطرح می‌شود این است که این مسیر را چقدر سریع می‌توان پیدا کرد؟ با داشتن یک کران بالا برای پیدا کردن کوتاه‌ترین مسیر، می‌توانیم مشخص کنیم که الگوریتم‌های حل سیمپلکس در کل چقدر می‌توانند سریع بشوند. در این زمینه حدسی وجود دارد که در هر چندوجهی بین هردو رأس مسیر کوتاهی وجود دارد که با $O(n)$ حرکت بتوان آن را طی کرد. این حدس هنوز اثبات نشده است. اما چیزی که با علم امروز می‌دانیم این است که با حدود $n^{1+\ln n}$ حرکت روی یال‌ها می‌توان به رأس مورد نظر رسید. اما با این حال، حتی با داشتن الگوریتمی برای پیدا کردن کوتاهترین مسیر در چندوجهی نمی‌توانیم سیمپلکس را در زمان خطی اجرا کنیم.