

درس ۳۵

پیاده سازی الگوریتم تولید ستون برای حل مسائل بزرگ برنامه ریزی خطی

تهیه شده توسط گروه آموزشی بهینه یاب



روش تولید ستون یا *column generation* یکی از روش های موثر در حل مسائل برنامه ریزی خطی با ابعاد بزرگ به خصوص در مواردی است که تعداد متغیرها بسیار بیشتر از تعداد محدودیت ها است و به عبارتی ماتریس ضرایب مسئله دارای ستون های زیادی است. این روش را در دهه شصت میلادی گیلمور و گموری طراحی کردند. در چنین مسائلی در حالت ازداسازی خطی تعدد متغیرهای غیر پایه که صفر هستند، بسیار زیاد است در حالی که فقط تعداد محدودی از متغیرها در پایه قرار دارند و دارای مقادیر غیر صفر می باشند. برای این منظور در این روش مسئله اولیه به دو زیر مسئله تفکیک می شود، زیر مسئله اصلی و زیر مسئله فرعی. زیر مسئله اصلی در حقیقت زیر مسئله کوچکی از مسئله اولیه است که با فقط زیر مجموعه ای از متغیرها تشکیل می شود. حل زیر مسئله اصلی که یک مدل برنامه ریزی خطی است، باعث شناسایی مقادیر متغیرهای همزاد می شود. از مقادیر متغیرهای همزاد حاصل از زیر مسئله اصلی استفاده می شود تا زیر مسئله فرعی تشکیل شود و با حل آن، متغیر ورودی و ستون مربوط به آن شناسایی و دوباره به زیر مسئله اصلی وارد می شود و این فرایند ارتباط بین پاسخ های مسئله اصلی و فرعی تا آنجا ادامه می یابد که شرط توقف الگوریتم برآورده شود. جزییات به کارگیری این روش در ادامه آمده است.

شرح الگوریتم تولید ستون برای حل یک مسئله برنامه ریزی خطی

یک مسئله برنامه ریزی خطی را به صورت زیر در نظر بگیرید:

$$LP : \begin{cases} \text{Max } Z = cx \\ s.t. \\ Ax = b \\ x \geq 0 \end{cases}$$

اگر ماتریس A در این مدل که یک ماتریس $m*n$ (m سطر و n ستون) فرض می شود بسیار بزرگ باشد، به معنی آن است که دارای تعداد قبل توجهی ستون است که در هر پاسخ پایه مسئله فقط از m ستون آن برای تشکیل وارون ماتریس پایه و پاسخ های پایه استفاده شده و سایر ستون ها آن یعنی $n-m$ ستون وابسته به متغیرهای غیرپایه ای صفر است که در تکرار فعلی استفاده نشده و در تکرار بعد هم قرار است فقط یکی از ستون های آن استفاده شود که وابسته به متغیر ورودی است. افزون بر این مسئله در بسیاری از مسائل واقعی اصولاً تشکیل ماتریس A از ابتدا در مسئله اصلی نیازمند صرف زمان و وقت بسیار است و در مواردی اصولاً تعیین تمام ستون ها تشکیل دهنده آن امکان پذیر نیست.

در روش تولید ستون می خواهیم بدون این که نیاز به شناسایی تمام ستون های ماتریس A باشد، ابتدا با حداقل تعداد مورد نیاز یعنی m ستون از ماتریس A شروع کنیم و زیر مسئله برنامه ریزی خطی آن را موسوم به زیر مسئله اصلی حل کنیم و سپس با وجود این که تمام ستون های ماتریس A را نمی دانیم، با تشکیل زیر مسئله فرعی و حل آن یکی از ستون های آن را به عنوان ستون متغیر ورودی به پایه تولید کرده و سپس آن را وارد زیر مسئله اصلی کنیم. حل جزییات زیرمسائل اصلی و فرعی را شرح می دهیم:

زیر مسئله اصلی محدود شده RMP

این زیر مسئله در ابتدا فقط شامل بخش کوچکی از مسئله اولیه، یعنی یک زیرماتریس $m \times m$ از ماتریس A است. تولید این ستون ها با توجه به مسئله اولیه انجام می شود. سپس در هر تکرار قرار است یک ستون جدید که از حل زیر مسئله فرعی به دست می آید، به آن اضافه شود. زیر مسئله اصلی RMP یک زیر مسئله از نوع LP است و می توان پاسخ را از روش سیمپلکس به دست آورد. البته با حل این مسئله، پاسخ متغیرهای دوگان آن نظیر u به دست خواهند آمد که برای این روش مهم است زیرا این متغیرهای همزاد در زیر مسئله فرعی استفاده می شوند.

زیرمسئله فرعی قیمت گذاری

در این زیر مسئله قرار است بدون این که تمام بردارهای ستونی مربوط به ماتریس A را بدانیم، بردار ستونی مربوط به متغیر ورودی به پایه را برای زیر مسئله محدود شده RMP به دست آوریم. می دانیم که در یک مسئله برنامه ریزی خطی با تابع هدف Max متغیر ورودی به پایه متغیری است که دارای شرط بهینگی به صورت زیر باشد:

$$Min \{z_j - c_j \mid z_j - c_j < 0\}$$

از طرف دیگر داریم:

$$z_j - c_j = c_B B^{-1} a_j - c_j = u a_j - c_j$$

که در آن u متغیرهای دوگان حاصل از پاسخ زیر مسئله اصلی RMP است. از نظریه برنامه ریزی خطی می دانیم $(z_j - c_j)$ را هزینه کاهش یافته یا $Reduced\ cost$ می نامند زیرا تفاوت z_j یعنی قیمت تمام شده برای تولید یک واحد محصول z_j و سود حاصل از هر واحد محصول c_j است. در یک مسئله حداکثر سازی، متغیر ورودی متغیری است که در آن تفاوت سود تولید

هر واحد آن نسبت به قیمت تمام شده هر واحد آن بیشتر باشد، به عبارتی $C_j - Z_j$ بزرگتری داشته باشد. برای این که این مقدار بزرگتر باشد چون معمولاً امکان افزایش سود وجود ندارد، باید هزینه های تولید محصول و یا به عبارتی همان قیمت تمام شده آن کاست و زیر مسئله فرعی را که براساس تولید می شود، زیر مسئله قیمت گذاری یا *pricing subproblem* نامند. حال از نظریه دوگانگی در برنامه ریزی خطی می دانیم که مقدار متغیر دوگان وابسته به محدودیت i ام مسئله اولیه برابر است با:

$$u_i = (c_B B^{-1})$$

پس زمانی که می خواهیم بردار a_j را از بین بردارهای ستون ماتریس A بدست آوریم، به نحوی که مربوط به متغیر ورودی به پایه مسئله اصلی باشد، کافی است داشته باشیم.

$$\begin{aligned} \text{Min } \{z_j - c_j \mid z_j - c_j < 0\} &= \text{Min } \{c_B B^{-1} a_j - c_j \mid c_B B^{-1} a_j - c_j < 0\} \\ &= \text{Min } \{u a_j - c_j \mid u a_j - c_j < 0\} \end{aligned}$$

چون بردار سطری مقادیر متغیرهای همزاد یعنی $u = (u_1, u_2, \dots, u_i, \dots, u_m)$ را می دانیم، پس می توان تعیین بردار a_j را با حل مسئله زیر تعیین کرد.

$$\text{Min } w = (u_1, u_2, \dots, u_i, \dots, u_m) \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix} - c_j$$

در مسئله بالا عناصر بردار a_j باید در بین خودشان شرایط موجود در مسئله اولیه را رعایت کنند که محدودیت های آن معمولاً به صورت خطی است که فضای موجه آن را S در نظر می گیریم.

همان طور که مشخص است، تابع هدف زیر مسئله بالا یا همان $Z_j - C_j$ مربوط به هر متغیر در جدول سیمپلکس است که در آن مقادیر متغیرهای همزاد از حل زیر مسئله اصلی محدود شده RMP تولید می شود. با حل زیر مسئله فرعی بردار a_j به دست می آید. اگر پاسخ تابع هدف زیر مسئله فرعی در یک مسئله با تابع هدف حداکثر سازی مقدار مثبتی شد بدین معنی

است که شرط بهینگی برآورد شده است و به نقطه پایانی و پاسخ بهینه رسیده ایم. ولی در صورتی که مقدار تابع هدف منفی شد، بردار ستون a_j باید به زیر مسئله اصلی محدود شده RMP اضافه شود و الگوریتم در تکرار بعدی ادامه می یابد.

مسئله کاهش ضایعات برش یا cutting stock problem

در یک کارخانه کاغذ سازی رول های کاغذ به عرض ۱۷ فوت و طول L تولید می شوند. تقاضای دریافتی برای کارخانه براساس عرض کاغذ مورد درخواست تهیه می شود. در حال حاضر کارخانه تقاضای زیر را دریافت کرده است:

عرض مورد تقاضا (فوت)	۳	۵	۹
تعداد رول مورد نیاز	۲۵	۲۰	۱۵

می خواهیم ببینیم چگونه رول های کاغذ را ببریم تا ضمن تامین سفارش های مورد نظر، ضایعات ناشی از الگوهای برش حداقل شوند؟

حل: این مسئله در ادبیات کلاسیک برنامه ریزی خطی یکی از مسائل مهم است و می دانیم که برای مدل سازی آن باید هم الگوهای برش و هم ترکیبات ناشی از آن ها را شناسایی کنیم. یافتن الگوهای برش در مسئله داده شده به دلیل کوچکی آسان اما یافتن ترکیب های آن ها بسیار زیاد است و به همین دلیل از مدل سازی برنامه ریزی خطی برای این منظور استفاده می کنیم.

در مدل سازی کلاسیک مسئله فرض می کنیم ابتدا باید الگوهای برش را تعیین و سپس با کمک مدل برنامه ریزی ریاضی آن را حل کنیم. برای مثال می توان ۷ الگوی زیر را در این مثال به دست آورد:

		الگوهای برش						
		1	2	3	4	5	6	7
عرض برش	3	5	4	2	2	1	0	0
	5	0	1	2	0	1	3	0
	9	0	0	0	1	1	0	1
عرض ضایعات		2	0	1	2	0	2	8

متغیر تصمیم گیری مسئله به صورت زیر تعریف می شود:

x_j = تعداد رولی که براساس الگوی برش j بریده می شود.

چون قرار است کل ضایعات حداقل شود و طول هر رول L است، پس مساحت کل ضایعات برابر است با:

کل مساحت موردنیاز- کل مساحت رول های استفاده شده= کل مساحت ضایعات

$$= 17 \times L (x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7) - L \times (25 \times 3 + 20 \times 5 + 9 \times 15)$$

$$= \boxed{17L} \sum x_j - \boxed{310L}$$

مقادیری که در کادر نوشته شده است، مقادیر ثابت توده و قابل حذف از تابع هدف است بنابراین مدل کلاسیک این مسئله به صورت یک مدل برنامه ریزی عدد صحیح زیر در می آید:

$$\left\{ \begin{array}{l} \text{Min } Z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \\ \text{s.t.} \\ 5x_1 + 4x_2 + 2x_3 + 2x_4 + x_5 \geq 25 \\ x_2 + 2x_3 + x_5 + 3x_6 \geq 20 \\ x_4 + x_5 + x_7 \geq 15 \\ x_j \geq 0, \text{int}, j = 1, \dots, 7 \end{array} \right.$$

مشکل اساسی این نوع مدل سازی برای مسائل واقعی آن است که تعداد الگوهای برش در چنین مسائل عملی بسیار زیاد خواهد بود. برای مثال اگر طول L و 40 تقاضا برای عرض های از 20 تا 80 داشته باشیم، نزدیک 100 میلیون الگو برش خواهیم داشت.

برای غلبه برای این مشکل می خواهیم بدون شناسایی تمام الگوهای اولیه، مسئله را از روش تولید ستونی حل کنیم. البته چون x_j باید عدد صحیح باشد و غالباً چنین مسائلی در ابعاد بسیار بزرگ اتفاق می افتد، می توان پاسخ بهینه عدد صحیح را برابر با پاسخ حاصل از گرد کردن پاسخ ازادسازی برنامه ریزی خطی فرض کرد. چنین کاری در ادبیات این مسئله مرسوم است. اکنون ازادسازی مسئله برنامه ریزی خطی مسئله بالا را در نظر می گیریم و آن را با روش تولید ستونی حل می کنیم.

تکرار شروع در زیر مسئله اصلی RMP

ابتدا به سه ستون برای تولید پایه اولیه نیاز داریم به نحوی که این سه ستون یک ماتریس اولیه 3×3 برای B^{-1} با ستون های مستقل از هم را تشکیل دهند، یعنی الگوهای اولیه هر یک به تنهایی فقط از عرض های $3, 5, 9$ استفاده کنند. اگر متغیرهای وابسته به آن ها را برای هماهنگی با مدل اولیه که ساختیم، به ترتیب x_1 و x_6 و x_7 بنامیم و مجموعه متغیرهای پایه را با BV_0 نشان دهیم، در این صورت داریم:

$$BV_0 = \begin{bmatrix} x_1 \\ x_6 \\ x_7 \end{bmatrix}, B_0 = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow B_0^{-1} = \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

چون ضریب هر متغیر در تابع هدف این مسئله یک است، پس:

$$(u_1, u_2, u_3) = c_B B_0^{-1} = (1, 1, 1) \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \left(\frac{1}{5}, \frac{1}{3}, 1 \right)$$

زیر مسئله فرعی قیمت گذاری

می خواهیم متغیر ورودی به پایه را در زیر مسئله اصلی تعیین کنیم. چون تابع هدف مسئله حداقل سازی است، باید با توجه به مثبت ترین $Z_j - C_j$ در بین متغیرهای غیرپایه مثبت، متغیر واردشونده به پایه تعیین شود و چون مسئله با ابعاد بسیار بزرگ است و تمام ستون ها متغیرها را در ماتریس A نداریم، روش تولید ستونی استفاده می شود و ستون مربوط به متغیر ورودی را به صورت زیر

$$\begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix}$$

در نظر می گیریم که هر عنصر آن بیانگر آن است که در الگوی انتخابی قرار است چه مقدار از عرض های مورد نظر

بریده شوند:

$$\begin{aligned} z_j - c_j &= c_B B_0^{-1} \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = (u_1, u_2, u_3) \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = \left(\frac{1}{5}, \frac{1}{3}, 1 \right) \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 \\ &= \frac{1}{5}a_3 + \frac{1}{3}a_5 + a_9 - 1 \end{aligned}$$

از طرف دیگر a_3, a_5, a_9 و a_9 باید به گونه ای انتخاب شوند که مجموع عرض مصرفی بیش از ۱۷ فوت نشود، یعنی:

$$\begin{cases} 3a_3 + 5a_5 + 9a_9 \leq 17 \\ a_3, a_5, a_9 \geq 0, \text{int} \end{cases}$$

پس به طور خلاصه به دنبال یافتن ترکیبی هستیم که

$$\text{Max } Z = \frac{1}{5}a_3 + \frac{1}{3}a_5 + a_9 - 1$$

s t.

$$3a_3 + 5a_5 + 9a_9 \leq 17$$

$$a_3, a_5, a_9 \geq 0, \text{int}$$

مسئله بالا یک مسئله کوله پشتی است. لذا:

$$Z_{IP}^* = \frac{8}{15}, a_9^* = a_3^* = a_5^* = 1$$

یعنی اولاً چون مقدار تابع هدف مثبت است، پس شرط توقف صادق نیست و ثانیاً الگوی فوق برای ورود به پایه انتخاب می شود که فرض کنید وابسته به متغیر مثلاً x_5 باشد. در این صورت دوباره به زیر مسئله اصلی بر می گردیم. در این مثال مشخص است که متغیر ورودی مربوط به متغیر x_5 است. در مواردی که مشخص نیست نام دلخواه برای آن در نظر گرفته می شود (مثلاً s_1).

زیر مسئله اصلی RMP

ابتدا با توجه به ستون ورودی باید ستون زیر را محاسبه کرد.

$$\bar{a}_j = B_0^{-1}a_j = B_0^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{5} & 1 & 1 \\ 1 & \frac{1}{3} & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{5} \\ \frac{1}{3} \\ 1 \end{bmatrix}$$

اکنون می توان آزمایش نسبت روش سیمپلکس برای تعیین متغیر خروجی از پایه را در مورد آن اجرا کرد. مقدار سمت راست با توجه به پایه این مرحله به صورت زیر بدست می آید.

$$\bar{b} = B_0^{-1}b = \begin{bmatrix} \frac{1}{5} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 25 \\ 20 \\ 15 \end{bmatrix} = \begin{bmatrix} 5 \\ \frac{20}{3} \\ 15 \end{bmatrix}$$

$$Min \left\{ \begin{matrix} x_6 \\ x_1 \\ \frac{5}{\frac{1}{5}}, \frac{20}{\frac{1}{3}}, \frac{15}{1} \end{matrix} \right\} = 15$$

پس، متغیر x_7 از پایه خارج می شود. محاسبه مقادیر متغیرهای پایه جدید و وارون ماتریس پایه جدید که از طریق حاصل

ضرب ماتریس مقدماتی به دست می آید، به صورت زیر است:

$$BV_1 = \begin{bmatrix} x_1 \\ x_6 \\ x_7 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_6 \\ x_5 \end{bmatrix} \rightarrow B = \begin{bmatrix} 5 & 0 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow B^{-1} = \begin{bmatrix} \frac{1}{5} & 0 & -\frac{1}{5} \\ 0 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix}$$

$$u = c_B B^{-1} = (1, 1, 1) \begin{bmatrix} \frac{1}{5} & 0 & -\frac{1}{5} \\ 0 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} = \left(\frac{1}{5}, \frac{1}{3}, \frac{7}{15} \right)$$

زیر مسئله فرعی قیمت گذاری

$$z_j - c_j = c_B B_0^{-1} \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = (u_1, u_2, u_3) \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = \left(\frac{1}{5}, \frac{1}{3}, \frac{7}{15} \right) \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = \frac{1}{5}a_3 + \frac{1}{3}a_5 + \frac{7}{15}a_9 - 1$$

$$\begin{cases} Max Z = \frac{1}{5}a_3 + \frac{1}{3}a_5 + \frac{7}{15}a_9 - 1 \\ s.t. \\ 3a_3 + 5a_5 + 9a_9 \leq 17 \\ a_3, a_5, a_9 \geq 0, \text{int} \end{cases} \Rightarrow a_3^* = 4, a_5^* = 1, a_9^* = 0, Z_{IP}^* = \frac{2}{15} > 0$$

یعنی اولاً چون مقدار تابع هدف مثبت است، پس شرط توقف صادق نیست و ثانیاً الگوی فوق که وابسته به متغیر x_2 است، برای ورود به پایه انتخاب می شود و دوباره روش از زیر مسئله اصلی محدود شده ادامه می یابد.

تکرار ۲:

زیر مسئله اصلی RMP

ابتدا با توجه به ستون ورودی باید ستون زیر را محاسبه کرد.

$$\bar{a}_j = B_0^{-1}a_j = B_0^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{5} & 0 & -\frac{1}{5} \\ 0 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{4}{5} \\ \frac{1}{3} \\ 0 \end{bmatrix}$$

اکنون می توان آزمایش نسبت روش سیمپلکس برای تعیین متغیر خروجی از پایه را در مورد آن اجرا کرد. مقدار سمت راست با توجه به پایه این مرحله به صورت زیر بدست می آید.

$$\bar{b} = B_0^{-1}b = \begin{bmatrix} \frac{1}{5} & 0 & -\frac{1}{5} \\ 0 & \frac{1}{3} & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 25 \\ 20 \\ 15 \end{bmatrix} = \begin{bmatrix} 2 \\ \frac{5}{3} \\ 15 \end{bmatrix}$$

$$\text{Min} \left\{ \begin{matrix} x_6 \\ x_1 \\ \frac{2}{4}, \frac{3}{1} \\ \frac{5}{5}, \frac{3}{3} \end{matrix} \right\} = 2.5$$

پس، متغیر x_1 از پایه خارج می شود. محاسبه مقادیر متغیرهای پایه جدید و وارون ماتریس پایه جدید که از طریق حاصل ضرب ماتریس مقدماتی به دست می آید، به صورت زیر است:

$$BV_1 = \begin{bmatrix} x_1 \\ x_6 \\ x_5 \end{bmatrix} \rightarrow \begin{bmatrix} x_2 \\ x_6 \\ x_5 \end{bmatrix} \rightarrow B = \begin{bmatrix} 4 & 0 & 1 \\ 1 & 3 & 1 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow B^{-1} = \begin{bmatrix} \frac{1}{4} & 0 & -\frac{1}{4} \\ -\frac{1}{12} & \frac{1}{3} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix}$$

$$u = c_B B^{-1} = (1, 1, 1) \begin{bmatrix} \frac{1}{4} & 0 & -\frac{1}{4} \\ -\frac{1}{12} & \frac{1}{3} & -\frac{1}{4} \\ 0 & 0 & 1 \end{bmatrix} = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{2} \right)$$

زیر مسئله فرعی قیمت گذاری

$$z_j - c_j = c_B B_0^{-1} \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = (u_1, u_2, u_3) \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = \left(\frac{1}{6}, \frac{1}{3}, \frac{1}{2} \right) \begin{bmatrix} a_3 \\ a_5 \\ a_9 \end{bmatrix} - 1 = \frac{1}{6}a_3 + \frac{1}{3}a_5 + \frac{1}{2}a_9 - 1$$

$$\begin{cases} \text{Max} Z = \frac{1}{6}a_3 + \frac{1}{3}a_5 + \frac{1}{2}a_9 - 1 \\ \text{s.t.} & \Rightarrow a_3^* = 1, a_5^* = 1, a_9^* = 1, Z_{LP}^* = 0 \neq 0 \\ 3a_3 + 5a_5 + 9a_9 \leq 17 \\ a_3, a_5, a_9 \geq 0, \text{int} \end{cases}$$

چون مقدار تابع هدف صفر است، پس شرط توقف صادق است و به جواب بهینه می رسیم.

پاسخ نهایی مسئله با این روش به صورت زیر است:

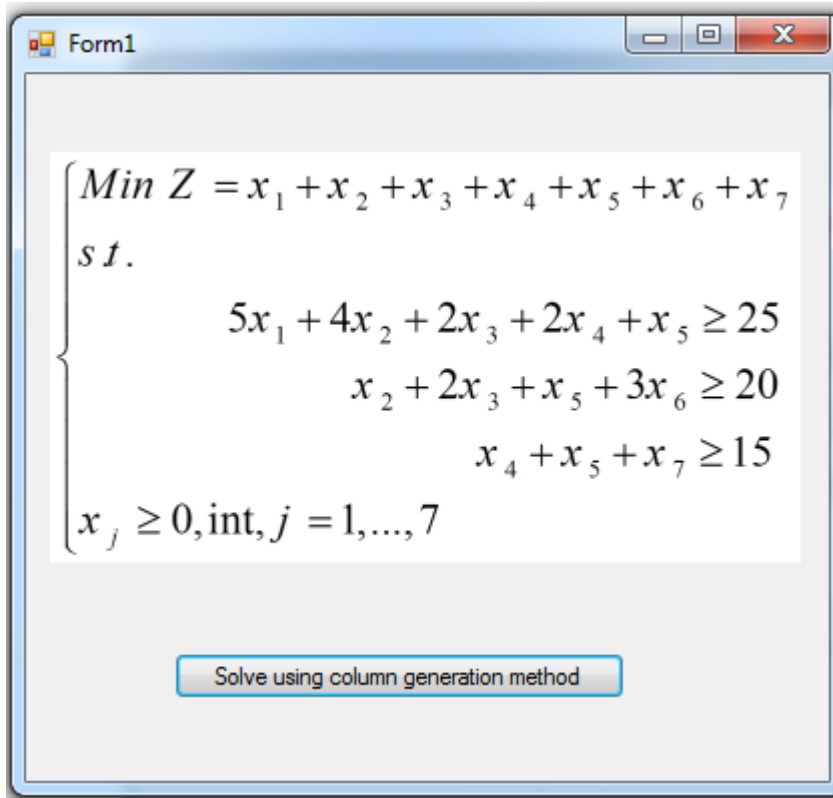
$$Z_{LP}^* = 18\frac{1}{3}, x_2^* = 2.5, x_5^* = 15, x_6^* = \frac{5}{6}$$

پیاده سازی روش تولید ستون به زبان ویژال بیسیک

در این قسمت مراحل برنامه نویسی مثال فوق برای حل به روش تولید ستون به شرح توضیح داده می شود. در ابتدا یک فرم به صورت زیر ساخته می شود و دکمه اجرا برنامه ساخته و سپس بر روی آن دوبار کلیک می شود. با توجه به این که در این روش نیاز به استفاده از روابط ماتریسی است، از کتابخانه *accord* استفاده می شود. برای این کار ابتدا این کتاب خانه به

فرمت *dll* را دانلود و سپس در محیط پروژه خود اضافه می شود. برای استفاده از قابلیت های سیپلکس و *Accord*، کد زیر را در بالای صفحه اضافه کنید.

```
Imports ILOG.Concert
Imports ILOG.CPLEX
Imports Accord.Math
```



پس از ورود به ناحیه کد نویسی، موارد زیر وارد می شود.

<pre>Dim Base(2, 2) As Double Base(0, 0) = 5 Base(1, 1) = 3 Base(2, 2) = 1 Dim Baseinver = Base.Inverse</pre>	ماتریس پایه به صورت یک ماتریس دو در دو معرفی می شود و مقادیر پیش فرض در آن وارد می شود. با استفاده از قابلیت <i>accord</i> ، معکوس ماتریس پایه محاسبه می شود.
<pre>Dim cB = {1.0, 1.0, 1.0} Dim u(2, 0) As Double u = Baseinver.Transpose.Multiply(cB.Transpose)</pre>	در این بخش، ضرایب متغیرهای پایه فعلی تعیین می شود که برابر یک برای تمام متغیرهای پایه است. مقدار متغیر همزاد از رابطه زیر بدست می آید که از توابع کتابخانه <i>Accord</i> بدست می آید. $u = c_B B^{-1}$
<pre>Dim stoppingcondition As Boolean stoppingcondition = False Dim b = {25.0, 20.0, 15.0}</pre>	شرایط توقف به صورت یک متغیر درست و غلط تعریف می

<pre>Dim bbar = Baseinver.Multiply(b.Transpose())</pre>	<p>شود و در ابتدای الگوریتم غلط در نظر گرفته می شود. مقدار ضرایب سمت راست با توجه به پایه فرض شده از رابطه زیر بدست می آید.</p> $\bar{b} = bB^{-1}$
<pre>Dim cplexCG As New Cplex() Dim rngcg(1)() As IRange Dim a3 As INumVar Dim a5 As INumVar Dim a9 As INumVar a3 = cplexCG.NumVar(0, Double.MaxValue, ILOG.Concert.NumVarType.Int) a5 = cplexCG.NumVar(0, Double.MaxValue, ILOG.Concert.NumVarType.Int) a9 = cplexCG.NumVar(0, Double.MaxValue, ILOG.Concert.NumVarType.Int)</pre>	<p>برای حل زیر مسئله فرعی قیمت گذاری، از کتاب خانه <i>cplex</i> استفاده می شود. متغیرهای این مدل به صورت a_5, a_3 و a_9 به صورت عدد صحیح و نامنفی تعریف می شود.</p>
<pre>Dim experssion(2) As ILOG.Concert.IIntExpr experssion(2) = cplexCG.Sum(cplexCG.Prod(3, a3), cplexCG.Prod(5, a5), cplexCG.Prod(9, a9)) rngcg(0) = New IRange(1) {} rngcg(0)(1) = cplexCG.AddLe(experssion(2), 17, "CG") Dim objfunctioncg As ILOG.Concert.IObjective = cplexCG.AddMaximize()</pre>	<p>برای ایجاد روابط تابع هدف و محدودیت های مسئله فرعی قیمت گذاری، متغیر <i>expression</i> تعریف می شود. محدودیت این مدل در فرایند حل تغییر نمی کند و لذا در ابتدا ساخته می شود. تابع هدف این مدل با توجه به پایه تغییر می کند که به صورت بیشینه سازی تعریف می شود.</p>
<pre>While stoppingcondition = False</pre>	<p>برای کنترل برقراری شرط توقف، از تابع <i>while</i> استفاده می شود. زمانی که شرط توقف برقرار گردد، از این تابع خارج می شود.</p>
<pre>experssion(1) = cplexCG.Prod(a3, 0) experssion(1) = cplexCG.Sum(cplexCG.Prod(u(0, 0), a3), cplexCG.Prod(u(1, 0), a5), cplexCG.Prod(u(2, 0), a9)) objfunctioncg.ClearExpr() cplexCG.AddToExpr(objfunctioncg, experssion(1))</pre>	<p>با توجه به مقدار فعلی متغیرهای همزاد، تابع هدف مدل فرعی قیمت گذاری تعیین می شود.</p>
<pre>Dim a3bar As Double Dim a5bar As Double Dim a9bar As Double</pre>	<p>مقدار بهینه متغیرهای مدل قیمت گذاری در متغیرهای این قسمت ذخیره می شود.</p>
<pre>If cplexCG.Solve() Then a3bar = cplexCG.GetValue(a3) a5bar = cplexCG.GetValue(a5) a9bar = cplexCG.GetValue(a9) Else MessageBox.Show("Error", "Watch out", MessageBoxButtons.RetryCancel) End If</pre>	<p>در این قسمت مدل قیمت گذاری حل می شود. اگر مدل دارای جواب بهینه داشته باشد، جواب بهینه متغیرها در متغیرهای تعریف شده ذخیره می شود. اگر مدل دارای جواب بهینه نباشد، پیام خطا چاپ می شود.</p>
<pre>If cplexCG.ObjValue - 1 > 0 Then Dim abelowentervar(2) As Double abelowentervar(0) = a3bar abelowentervar(1) = a5bar abelowentervar(2) = a9bar Dim newbelow = Baseinver.Multiply(abelowentervar.Transpose()) Dim relativetest0 As Double</pre>	<p>مطابق درس، اگر مقدار تابع هدف مدل فوق منهای یک مثبت باشد، یعنی به جواب بهینه نرسیده ایم و باید فرایند ادامه یابد. در این صورت، متغیر ورودی به پایه مشخص می</p>

```
Dim relativetest1 As Double
Dim relativetest2 As Double
Dim existvari As Integer
relativetest0 = bbar(0, 0) / newbelow(0, 0)
relativetest1 = bbar(1, 0) / newbelow(1, 0)
relativetest2 = bbar(2, 0) / newbelow(2, 0)

If Math.Min(relativetest0, relativetest1) >
relativetest2 Then
existvari = 2
Else
If Math.Min(relativetest0, relativetest1) =
relativetest1 Then
existvari = 1
Else
existvari = 0

End If

End If

Base(0, existvari) = a3bar
Base(1, existvari) = a5bar
Base(2, existvari) = a9bar

Baseinver = Base.Inverse
u = Baseinver.Transpose.Multiply(cB.Transpose)
bbar = Baseinver.Multiply(b.Transpose())
Else
stoppingcondition = True
End If

End While

cplexCG.End()
```

شود که در واقع متغیری است که ضرایب آن در محدودیت ها به صورت $a3bar$, $a5bar$, و $a9bar$ است. برای تعیین متغیر خروجی از پایه، ابتدا باید ستون متغیر ورودی به پایه را تعیین کرد. سپس از تست نسبت متغیر خروجی از پایه تعیین می گردد که به صورت روبرو انجام می شود. پس از تعیین متغیر خروجی از پایه، متغیر ورودی در پایه جای آن را در ماتریس $Base$ می گیرد. ماتریس معکوس پایه بدست می آید و مقادیر سمت راست به هنگام می شود. در غیر این صورت (یعنی مقدار تابع هدف منهای یک مثبت نباشد)، شرط توقف درست می شود.

پس از پایان اجرای الگوریتم فضای حافظه کامپیوتر از حل مسئله پاک می شود.

BEHINEHYAB.COM