به نام خدا

تمرین سری یک – بهینه سازی کاربردی

ارسلان فیروزی - ۹۷۱۰۲۲۲۵

۱. در این سوال به دنبال یک Match که شرط کاور کردن رئوس و کمینه بودن جمع وزن یال ها را دارد، هستیم. به همین خاطر از متغیر E_{ij} استفاده کردم که نشان دهنده این است که آیا یال بین راس i ام در مجموعه رئوس U و راس j ام در مجموعه رئوس V در Match بهینه وجود دارد یا نه. به همین خاطر هرکدام یک متغیر بولین است و مسئله یک Integer Programming است.

(a

$$\text{Minimize} \sum_{i=1}^{size(U)} \sum_{j=1}^{size(V)} E_{ij} W_{ij}$$

$$\left.\begin{array}{l} \sum_{j=1}^{size(V)} E_{ij} = 1 \quad , i = 1 \dots size(U) \\ \sum_{i=1}^{size(U)} E_{ij} = 1 \quad , j = 1 \dots size(V) \end{array}\right\} \text{شروط لازم برای یک Match}$$

$E_{ij} \leq Edge_{ij} \longrightarrow$ ماتریس $Edge_{ij}$ نشان دهنده این است که در گراف اولیه بین راس i در U و راس j در V یال هست یا نه اگر باشد مقدار یک دارد و اگر نه مقدار صفر

$$E_{ij} \in \{0 , 1\}$$

b) پس از relaxation:

$$\text{Minimize} \sum_{i=1}^{size(U)} \sum_{j=1}^{size(T)} E_{ij} W_{ij}$$

$$\sum_{j=1}^{size(T)} E_{ij} = 1 \quad , \; i = 1 \ldots size(U)$$

$$E_{ij} \in \mathbb{R}$$

$$\sum_{i=1}^{size(U)} E_{ij} = 1 \quad , \; j = 1 \ldots size(T)$$

$$E_{ij} \leqslant Edge_{ij}$$

$$E_{ij} \leqslant 1$$

$$0 \leqslant E_{ij}$$

۲. جهت محاسبه بیشینه شعاع توپ، لازم است دو متغیر C و R که C یک نقطه بیانگر مرکز توپ در این فضای معرفی شده است و R یک طول بیانگر شعاع توپ است، یافته شود. توپ با بیشینه شعاع در حالتی است که R بیشینه باشید به شرطی که فاصله هیچ نقطه ای روی باندهای مشخص شده در این فضا و مرکز از R کمتر نباشد. چرا که این یعنی نقاطی در توپ هستند که در فضای مشخص شده توسط P نیستند. در زیر توضیحات بالا به صورت ریاضی بیان شده است:

$$\text{maximize } R \qquad (Q2$$

$$R \in \mathbb{R}$$

$$AC \leqslant b \qquad C \in \mathbb{R}^n$$

$$A_i C - b_i \leqslant R \, \|A_i\|_2 \quad i = 1 \ldots m$$

$$A_i C - b_i \geqslant -R \, \|A_i\|_2 \quad i = 1 \ldots m$$

۳. جهت بیان این مسئله در حوزه Linear Programming من از یک ماتریس J_{ij} استفاده کردم که مفهوم این را دارد که چه مقدار از کار j ام توسط کارگر i ام انجام می‌شود. یعنی اگر یک باشد، کارگر i ام کل آن کار را انجام می‌دهد و اگر صفر باشد هیچ کاری برای آن نمی‌کند و اگر بین صفر و یک باشد بخشی از آن را انجام می‌دهد. یعنی مقادیر J_{ij} عضو اعداد حقیقی هستند.

در صورتی که یک کار را نتوان بین کارگرها تقسیم کرد، مقادیر J بولین هستند و عضو اعداد صحیح و مسئله Integer Programming خواهد بود.

میخواهیم بیشینه‌ی ساعت کار شده توسط یک کارگر بین تمام کارگرها را مینیمم کنیم. پس از یک متغیر کمکی S برای بیان ماکسیمم در مسئله مان استفاده کردم. همچنین از ماتریس S که درایه S_{ij} آن این مفهوم را برای من دارد که کارگر i ام آیا میتواند کار j ام را انجام دهد یا نه را برای پیاده سازی کارهای قابل انجام توسط هر کارگر، استفاده کردم و شرط آن به این صورت است که همواره هر درایه ماتریس J از درایه متناظر ماتریس S کوچکتر باید باشد:

$$\text{Minimize} \quad S$$

$$S \geq \sum_{j=1}^{n} J_{ij} P_j \quad , \quad i = 1, \dots, m$$

$$\sum_{i=1}^{m} J_{ij} = 1 \quad , \quad j = 1, \dots, n$$

$$J_{ij} \leq S_{ij} \quad , \quad \begin{matrix} i = 1 \dots m \\ j = 1 \dots n \end{matrix}$$

$$J_{ij} \in \{0, 1\} \quad , \quad \begin{matrix} i = 1, \dots, m \\ j = 1, \dots, n \end{matrix}$$

۴. من در این سوال میخواهم جمع ضرب وزن هر کار در زمان پایان آن را مینیمم کنم. پس هدف را مینیمایز کردن عبارت جمع ضرب وزن هر کار در زمان پایان آن میگذارم. بر روی

زمان پایان کارها این شرط را دارم که اندازه اختلاف هیچ ۲ زمان پایان کاری نمیتواند کوچکتر از P_j مربوطه باشد. لذا شرط $|t_i-t_j|>=P_j$ را دارم. اما من به راحتی قبل نمیتوانم قدر مطلق را از این شرط بردارم و مجبورم برای پیاده سازی منطق Or از ۲ متغیر بولین کمکی به نام های B و B' استفاده کنم که در نتیجه مسئله تبدیل به یک Mixed Integer Linear Programming می‌شود:

$$\text{minimize} \quad \sum_{j=1}^{n} T_j w_j$$

$$B_{ij}(T_j - T_i) \geq P_j \times B_{ij}$$

$$B'_{ij}(T_i - T_j) \geq P_j \times B'_{ij}$$

$$B_{ij} = B_{ji}$$

$$B'_{ij} = B'_{ji}$$

$$B_{ij} + B'_{ij} = 1$$

$$B_{ij} \in \{0, 1\}$$

$$B'_{ij} \in \{0, 1\}$$

۵. با پیاده سازی شروط بیان شده در صورت سوال به مسئله زیر رسیدم. با توجه به اینکه تاکید شده است تنها از ۲ متغیر استفاده شود، از متغیر تعداد متریال A، B و C نمیتوانستم استفاده کنم و خودم به صورت دستی نابرابری آن ها را به فرم P1 و P2 یا همان تعداد کالاهای تولید از نوع یک و دو نوشتم:

$$\text{Maximize} \quad 6P1 + 9P2$$

$$P1 \leqslant 130$$

$$0 \leqslant P1$$

$$P2 \leqslant 100$$

$$0 \leqslant P2$$

$$7P1 + 7P2 < 1200$$

$$(P_1 + P_2)\frac{1}{60} < 20$$

۶. برای بیان مسئله به فرم LP، برای هر قدر مطلق یک متغیر کمکی استفاده کردم. و همانطور که در کلاس تدیس شد، نابرابری‌های آن را بازنویسی کردم:

$$\text{minimize} \quad 2x + 3s$$

$$y - 10 \leqslant s$$

$$y - 10 \geqslant -s$$

$$t + y \leqslant 5$$

$$x + 2 \leqslant t$$

$$x + 2 \geqslant -t$$

۷. جهت بدست آوردن بیشینه جریان انتقال یافته از S به t، مسئله را به این صورت بیان کردم که به دنبال بیشینه کردن شار خروجی نود S هستیم. شار گذرنده از هر یال را $x_{i,j}$ نام‌گذاری میکنم و شروط پایداری شار برای رئوس میانی را مینویسم. نیاز داریم که

همچنین شار گذرنده از هر یال از محدودیت آن یال کمتر باشد. لذا از ماتریس L استفاده کردم که در صورتی که یالی در گراف بین i و j وجود داشته باشد و جهت آن از i به j باشد، در ماتریس L درایه L_{ij} برابر مقدار محدودیت یال می‌شود و اگر این یال وجود نداشت مقدار صفر دارد چون هیچ شاری قابلیت گذر ندارد. و شرط آن این می‌شود که درایه های ماتریس x از درایه های ماتریس L کمتر باشد:

$$\text{maximize} \sum_{\{j \mid (s,j)\in A\}} x_{sj}$$

$$x_{ij} \leq l_{ij} \quad , (i,j)\in A$$

$$0 \leq x_{ij} \quad , (i,j)\in A$$

$$\sum_{\{j \mid (i,j)\in A\}} x_{ij} - \sum_{\{j \mid (j,i)\in A\}} x_{ji} = 0 \quad , i = \text{تمام رئوس جز s و t}$$

$$l_{ij} = \begin{cases} \text{Max. flow} & (i,j)\in A \\ 0 & (j,i)\notin A \end{cases}$$

۸. a) برای n=200 و K=15 من رسیدم به اینکه ۷ متغیر اول x مقدار ۲ داشته باشند و متغیر هشتم مقدار ۱ داشته باشه و بقیه صفر باشند. جواب تو مدت زمان ۰.۰۲۰۲۳۱ ثانیه بدست اومد. همچنین مقدار objective function برابر با ۶۴ شد.

برای n=2000 و K=150 من رسیدم به اینکه ۷۵ متغیر اول x مقدار ۲ داره و بقیه صفر. جواب تو مدت زمان ۰.۱۰۸۱۸۲ ثانیه بدست اومد. مقدار objective function برابر با ۵۷۰۰ شد.

b) نتایج کاملا مشابه قسمت قبل شد با این فرق که پاسخ برای n=200 و K=15 در مدت زمان ۰.۰۲۳ ثانیه و برای n=2000 و K=150 در مدت زمان ۰.۰۲۸ ثانیه بدست آمد.

کد هر دو قسمت در فایل main.m به زبان متلب نوشته شده است.