

بسمه تعالی

برنامه ریزی خطی

Linear programming Manufacturing with models

ارائه دهنده : سولماز آقائی

پاییز ۹۳

موضوع فصل :

آیا برای یک شی داده شده قالب مناسبی وجود دارد تا بتوان شی را به راحتی از قالب خارج نمود؟

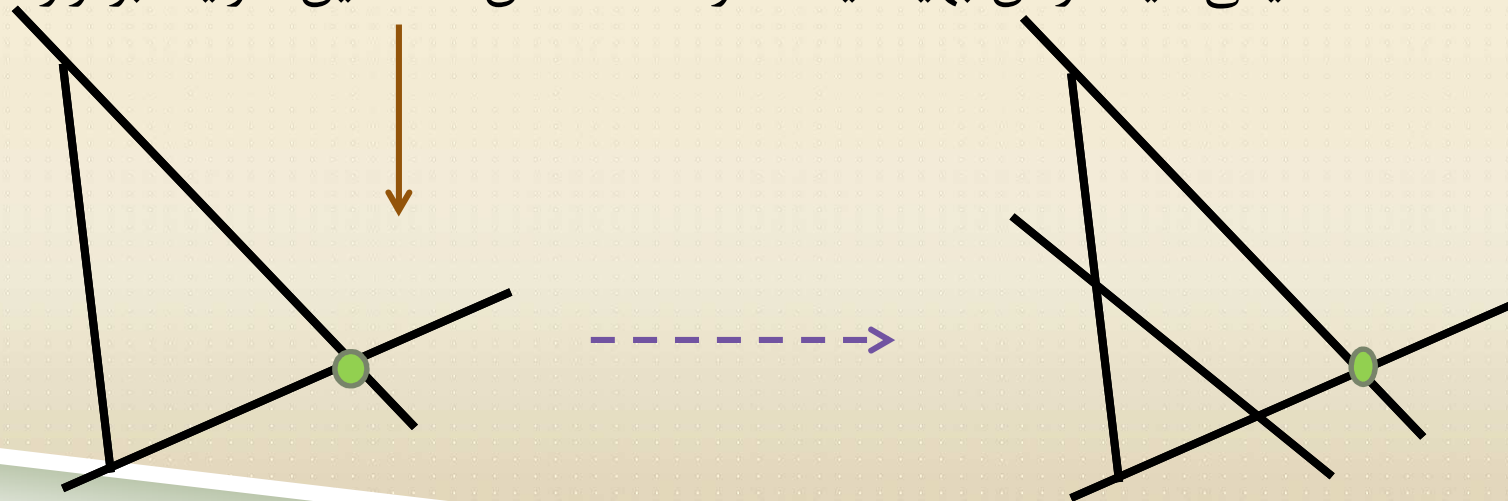
اهداف فصل:

- تبدیل مساله قالب گیری به مساله برنامه ریزی خطی
- حل مساله برنامه ریزی خطی به کمک روش های هندسه محاسباتی
- ارائه یک الگوریتم تصادفی برای بهبود زمان حل مسئله ی برنامه ریزی خطی

حل هندسی مساله برنامه‌ریزی خطی

راه حل مسئله استفاده از یک الگوریتم افزایشی است. بدین صورت که :

محدودیت‌ها را یک به یک به مساله اضافه می‌کنیم و جواب مرحله‌ی قبل را با در نظر گرفتن این محدودیت به روز می‌نماییم تا جوابی برای برنامه‌ریزی خطی جدید به دست آید. لازم است که جواب مساله در هر مرحله یکتا باشد به عبارت دیگر فرض می‌شود که هر ناحیه **feasible** میانی، یک راس بهینه یکتا دارد. اما ممکن است این شرایط برقرار نباشد.



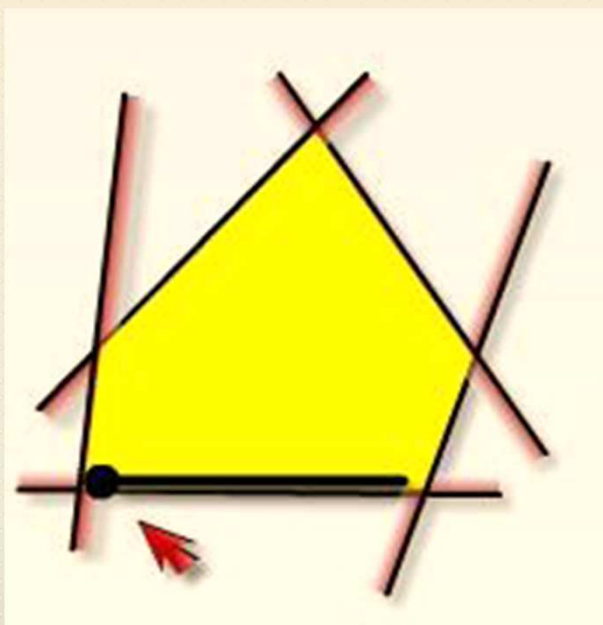
یکی از شرایطی که ممکن است راس یکتا نباشد این است که مسئله بی کران باشد.
برای حل این مسئله دو محدودیت جدید به محدودیت‌های مساله اضافه می کنیم.

$$m_1 := \begin{cases} p_x \leq M & c_x > 0 \\ -p_x \leq M & o.w \end{cases}$$

$$m_2 := \begin{cases} p_y \leq M & c_y > 0 \\ p_y \leq M & o.w \end{cases}$$

مقدار M آنقدر بزرگ انتخاب می شود که اضافه شدن این محدودیت‌ها تاثیری روی محدودیت‌های دیگر نداشته باشد.

حالت دیگری که ممکن است بر یکتایی جواب تاثیر بگذارد حالتی است که در آن همه نقاط روی یک ضلع از ناحیه شدنی، جواب مسئله باشند.
برای حل این مسئله ما راس سمت چپ را انتخاب می کنیم.



فرض مسئله: فرض کنید $(H, \vec{c})$ یک مسئله خطی باشد و نیم صفحه های $h_1, h_2, \dots, h_n$ را داشته باشیم و H_i مجموعه ای از i محدودیت اول همراه با دو محدودیت m_1, m_2 باشد و C_i ناحیه feasible تعریف شده توسط این محدودیت ها می باشد.

$$H_i := \{m_1, m_2, h_1, h_2, \dots, h_i\}$$

$$C_i := m_1 \cap m_2 \cap h_1 \cap h_2 \cap \dots \cap h_i$$

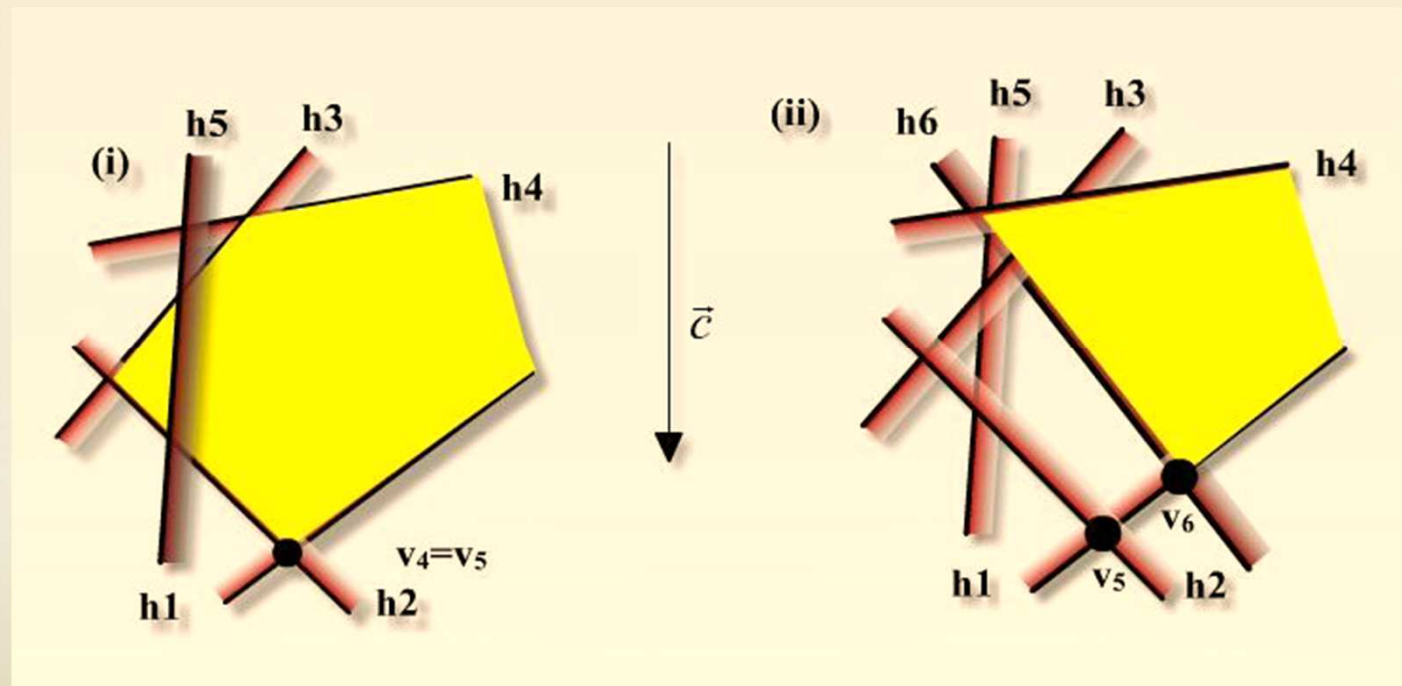
پس هر ناحیه شدنی C_i دارای یک نقطه بهینه یکتاست که آن را با v_i نشان می دهیم.

$$C_0 \supseteq C_1 \supseteq C_2 \dots C_n = C$$

اگر برای یک i ، $C_i = \emptyset$ آنگاه برای هر $j \geq i$ ، $C_j = \emptyset$ و مسئله خطی infeasible می شود.

دو حالت پس از اضافه کردن نیم صفحه جدید رخ می دهد :

- راس بهینه تغییر نمی کند.
- راس بهینه تغییر می کند در نتیجه باید راس بهینه جدید را یافت.



لم ۴.۵:

فرض کنید $1 \leq i \leq n$ و C_i و v_i به صورت قبل تعریف شده باشند ، آنگاه داریم :

i. اگر $v_{i-1} \in h_i$ آنگاه $v_i = v_{i-1}$

ii. اگر $v_{i-1} \notin h_i$ آنگاه $C_i = \emptyset$ یا $v_i \in l_i$ که l_i خط کران h_i است .

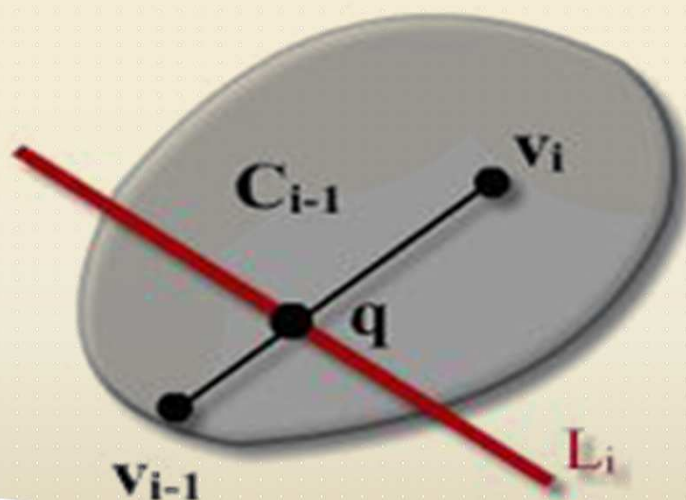
اثبات i:

فرض کنید $v_{i-1} \in h_i$ ، چون $C_i = C_{i-1} \cap h_i$ و $v_{i-1} \in C_{i-1}$ در نتیجه $v_{i-1} \in C_i$.
به علاوه نقطه بهینه در C_i نمی تواند بهتر از نقطه بهینه در C_{i-1} باشد

چون $C_i \subseteq C_{i-1}$ ، بنابراین v_{i-1} راس بهینه در C_i می باشد و $v_i = v_{i-1}$.

اثبات ii :

با استفاده از برهان خلف اثبات می کنیم. اگر $v_{i-1} \notin h_i$ آنگاه برای رسیدن به تناقض فرض می کنیم C_i تهی نباشد و v_i روی خط l_i قرار نداشته باشد. پاره خط $\overline{v_{i-1}v_i}$ درون C_{i-1} قرار دارد و می دانیم تابع هدف در جهت v_i به v_{i-1} در حال افزایش است. اگر نقطه تقاطع پاره خط با خط l_i را q در نظر بگیریم آنگاه $f_{\vec{c}}(q) > f_{\vec{c}}(v_i)$ که با تعریف v_i در تناقض است.



با توجه به لم قبل ، اگر $v_{i-1} \notin h_i$ باشد مسئله به صورت زیر در می آید:
 پیدا کردن نقطه p روی خط l_i که تابع هدف را ماکزیمم می نماید و برای هر $h \in H_i$ ،
 $p \in h$.

برای حل مسئله فرض می کنیم l_i عمودی نباشد ، پس تابع هدف جدید را به صورت یک
 بعدی به صورت زیر تعریف می کنیم :

$$\overline{f_{\rightarrow}}_c : R \rightarrow R$$

$$f_{\rightarrow}_c(p) = \overline{f_{\rightarrow}}_c(p_x) \quad \text{for } p \in l_i$$

برای هر نیم صفحه h فرض می کنیم $\sigma(h, l_i)$ مختص X نقطه تقاطع l_i و خط محدود
 کننده h باشد در این صورت مساله به صورت زیر در می آید :

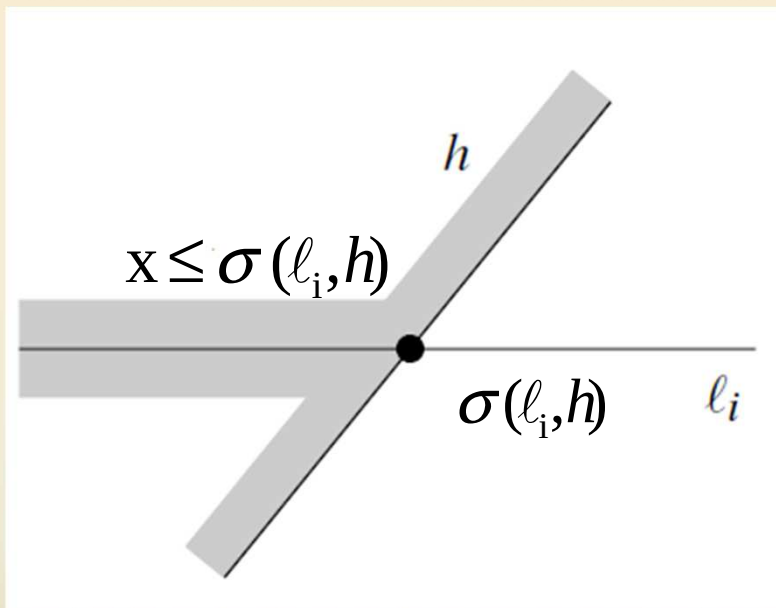
$$\max \vec{f}_{\vec{c}}(x)$$

$$s.t. \quad x \geq \sigma(\ell_i, h),$$

$$x \leq \sigma(\ell_i, h),$$

اگر $\ell_i \cap h$ از چپ کراندار باشد و $h \in H_{i-1}$

اگر $\ell_i \cap h$ از راست کراندار باشد و $h \in H_{i-1}$



برای حل مسئله ، ناحیه شدنی بازه $[X_{\text{left}}, X_{\text{right}}]$

$$X_{\text{left}} = \max\{ \sigma(\ell_i, h), \quad x \geq \sigma(\ell_i, h) \}$$

$$X_{\text{right}} = \min\{ \sigma(\ell_i, h), \quad x \leq \sigma(\ell_i, h) \}$$

لم ۴.۶ :

یک مساله برنامه ریزی خطی یک بعدی در زمان خطی قابل حل است . بنابراین جواب مسئله را در می توان در زمان $O(i)$ تعیین نمود.
حال الگوریتم را ارائه می دهیم.

Algorithm 2DBOUNDEDLP($H, \vec{c}, m_1, m_2$)

Input. A linear program $(H \cup \{m_1, m_2\}, \vec{c})$, where H is a set of n half-planes, $\vec{c} \in \mathbb{R}^2$, and m_1, m_2 bound the solution.

Output. If $(H \cup \{m_1, m_2\}, \vec{c})$ is infeasible, then this fact is reported. Otherwise, the lexicographically smallest point p that maximizes $f_{\vec{c}}(p)$ is reported.

1. Let v_0 be the corner of C_0 .
2. Let $h_1, \dots, h_n$ be the half-planes of H .
3. **for** $i \leftarrow 1$ **to** n
4. **do if** $v_{i-1} \in h_i$
5. **then** $v_i \leftarrow v_{i-1}$
6. **else** $v_i \leftarrow$ the point p on ℓ_i that maximizes $f_{\vec{c}}(p)$, subject to the constraints in H_{i-1} .
7. **if** p does not exist
8. **then** Report that the linear program is infeasible and quit.
9. **return** v_n

لم ۴.۷ :

الگوریتم 2DBOUNDEDLP جواب یک مسئله خطی کراندار با n محدودیت و دو متغیر را در زمان $O(n^2)$ و حافظه خطی محاسبه می کند .

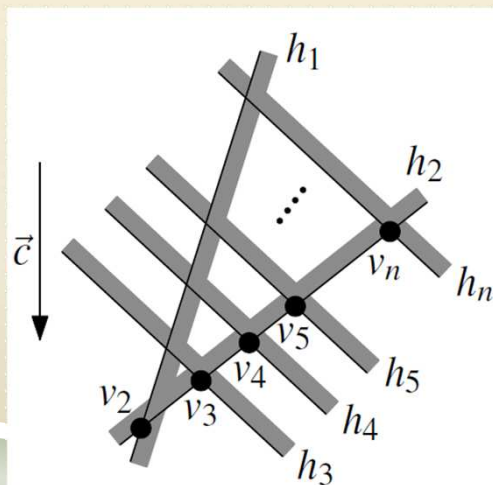
اثبات :

درستی الگوریتم با توجه به توضیحاتی که داده شد نتیجه می گیرد. برای زمان اجرا باتوجه به اینکه نیم صفحه ها یکی یکی اضافه می شوند n مرحله داریم و زمانی که در مرحله i ام صرف می شود برابر است با حل مسئله خطی یک متغیره که $O(i)$ است . بنابراین زمان اجرای الگوریتم برابر است با :

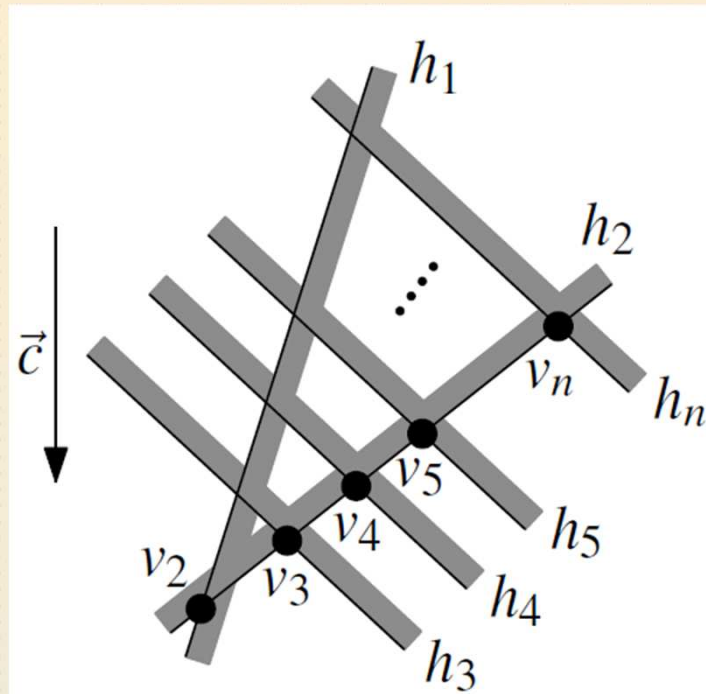
$$\sum_{i=1}^n O(i) = O(n^2)$$

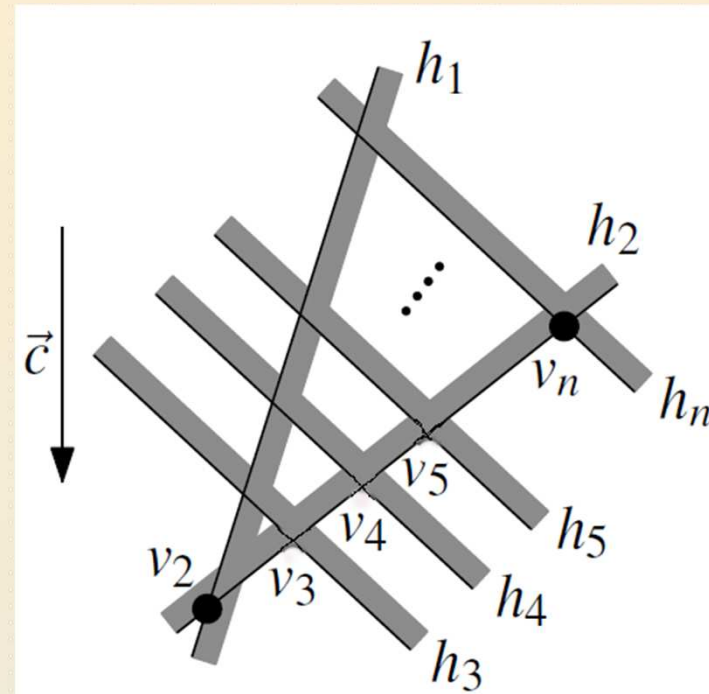
زمان اجرای الگوریتم نسبت به الگوریتم بخش اول کندتر است. حال مسئله این است که:
آیا می توان زمان اجرای الگوریتم را کاهش داد؟

برای هر مرحله از الگوریتم زمان $O(i)$ در نظر گرفتیم در صورتی که زمان لازم فقط مربوط به وقتی است که نقطه بهینه تغییر می کند و در باقی حالت ها زمان اجرا ثابت است ، پس اگر بتوانیم تعداد دفعات تغییر نقطه بهینه را بدست آوریم زمان اجرای الگوریتم دقیق تر می شود.



در برخی حالت ها ممکن است تعداد تغییر نقطه بهینه $O(n)$ باشد.
حال اگر ترتیب اضافه نمودن پاره خط ها تغییر دهیم چه اتفاقی رخ می دهد؟

$$h_1, \dots, h_n$$


$$h_1, h_2, h_n, \dots, h_3$$


برای یافتن ترتیب مناسب از تکنیک ترتیب تصادفی استفاده می کنیم . برای این کار از یک زیر روال به نام RandomPermutation استفاده می کنیم که یک ترتیب تصادفی از اعداد ایجاد می نماید :

Algorithm RANDOMPERMUTATION(A)

Input. An array $A[1 \cdots n]$.

Output. The array $A[1 \cdots n]$ with the same elements, but rearranged into a random permutation.

1. **for** $k \leftarrow n$ **downto** 2
2. **do** $rndindex \leftarrow \text{RANDOM}(k)$
3. Exchange $A[k]$ and $A[rndindex]$.

حال الگوریتم زیر به دست می آید :

Algorithm 2DRANDOMIZEDBOUNDEDLP($H, \vec{c}, m_1, m_2$)

Input. A linear program $(H \cup \{m_1, m_2\}, \vec{c})$, where H is a set of n half-planes, $\vec{c} \in \mathbb{R}^2$, and m_1, m_2 bound the solution.

Output. If $(H \cup \{m_1, m_2\}, \vec{c})$ is infeasible, then this fact is reported. Otherwise, the lexicographically smallest point p that maximizes $f_{\vec{c}}(p)$ is reported.

1. Let v_0 be the corner of C_0 .
2. Compute a *random* permutation $h_1, \dots, h_n$ of the half-planes by calling RANDOMPERMUTATION($H[1 \dots n]$).
3. **for** $i \leftarrow 1$ **to** n
4. **do if** $v_{i-1} \in h_i$
5. **then** $v_i \leftarrow v_{i-1}$
6. **else** $v_i \leftarrow$ the point p on ℓ_i that maximizes $f_{\vec{c}}(p)$, subject to the constraints in H_{i-1} .
7. **if** p does not exist
8. **then** Report that the linear program is infeasible and quit.
9. **return** v_n

زمان اجرای الگوریتم تصادفی :

زمان اجرا به ترتیب تصادفی که محاسبه می شود بستگی دارد . چون n نیم صفحه داریم ، $n!$ حالت برای اجرای الگوریتم وجود دارد که هر کدام زمان اجرای مربوط به خود را دارند . چون انتخاب جایگشت تصادفی است احتمال هر کدام از زمان های اجرا برابر است . بنابراین برای بدست آوردن زمان اجرای الگوریتم ، میانگین زمان اجرا روی $n!$ حالت مختلف را بدست می آوریم و آن را زمان مورد انتظار اجرای الگوریتم می گوئیم .

قضیه :

مساله برنامه ریزی خطی ۲ بعدی با n محدودیت را می توان در زمان قابل انتظار $O(n)$ حل نمود.

یادآوری :

امید ریاضی یا میانگین متغیر تصادفی
 $f(x)$ تابع چگالی احتمال

$$E(x) = \begin{cases} \sum x f(x) \\ \int x f(x) \end{cases}$$

لم ۴.۸ :

مسئله برنامه ریزی خطی دوبعدی با n معادله در زمان مورد انتظار $O(n)$ و ذخیره سازی خطی حل می شود.

اثبات : زمان اجرای RANDOMPERMUTATION برابر است با $O(n)$. برای اضافه کردن صفحات، متغیر تصادفی X_i را بدین صورت تعریف می کنیم:

$$X_i = \begin{cases} 1 & v_{i-1} \notin h \\ 0 & o.w \end{cases}$$

پس زمان سپری شده برای n صفحه در خط ۶ برابر است با:

$$\sum_{i=1}^n o(i) X_i$$

با توجه به خطی بودن امید ریاضی:

$$E\left(\sum_{i=1}^n o(i) \cdot X_i\right) = \sum_{i=1}^n o(i) \cdot E(X_i)$$

$E(X_i)$ چیست؟

احتمال اینکه $v_{i-1} \notin h_i$

برای تحلیل این احتمال از روند BackwardsAnalysis استفاده می شود، یعنی فرض می شود که الگوریتم انجام گرفته و راس بهینه v_n بدست آمده، حال نیم صفحه h_n را حذف می کنیم .

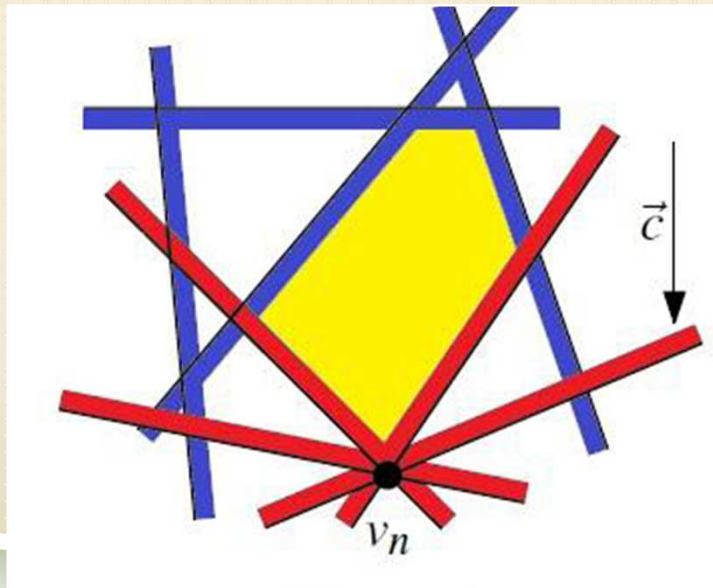
با چه احتمالی نقطه بهینه تغییر می کند؟

اگر v_n راس C_n باشد، حداقل توسط دو تا از نیم صفحه ها تعریف می شود .
و از آنجا که نیم صفحه ها به طور تصادفی اضافه شده اند، احتمال اینکه h_n یکی از نیم صفحه های تعریف کننده v_n باشد حداکثر $\frac{2}{n}$ می باشد.

چرا حداکثر $\frac{2}{n}$ ؟

اولا اگر v_n تعریف کننده بیش از دو یال باشد، آنگاه حذف h_n ممکن است باعث تغییر v_n نشود، چون ممکن است v_n بعد از حذف h_n باز هم راس باشد.

دوما علاوه بر n معادله m_1 و m_2 هم می توانند در ساختن v_n دخالت داشته باشند، که این هم احتمال را از $\frac{2}{n}$ کمتر می کند.



حال اگر همین روند Backwards برای i معادله اول H در نظر گرفته شود، باز هم داریم:

$$E(X_i) \leq \frac{2}{n}$$

در نتیجه زمان کل مورد انتظار برای حل تمام برنامه های خطی دو بعدی برابر است با:

$$\sum_{i=1}^n O(i) \frac{2}{i} = O(n)$$

برنامه خطی بی کران :

در قسمت قبل برنامه خطی نامحدود را با اعمال دو محدودیت اضافی کراندار کردیم، اما این کار همیشه ممکن نیست و حتی گاهی برنامه خطی محدود می شود ولی کران یا همان مرز ناحیه ممکن ، بسیار بزرگ می شود به گونه ای که قابل شناخت نیست.

مسئله اول : چگونه کراندار بودن یا نبودن برنامه خطی را تشخیص دهیم ؟

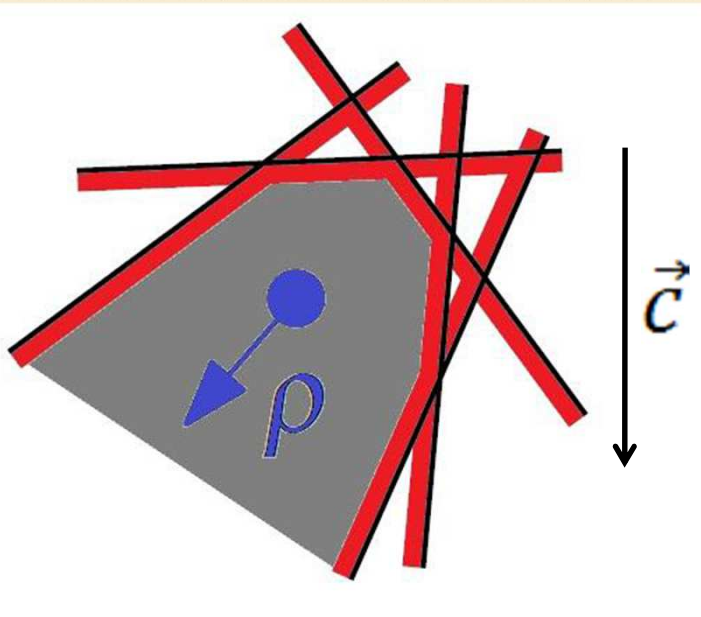
لم ۹-۴: یک برنامه خطی $(H, \vec{c})$ غیر کراندار است اگر و فقط اگر یک بردار $\vec{d}$ وجود داشته باشد بطوریکه $\vec{d} \cdot \vec{c} > 0$ و برای تمام $h \in H$ ، $\vec{d} \cdot \vec{\eta}(h) \geq 0$ و برنامه خطی $(H, \vec{c})$ ممکن می باشد

اگر

$$H' = \{h \in H \mid \vec{\eta}(h) \cdot \vec{d} = 0\}$$

اثبات : (رفت) قبلا گفته شد که بی کران بودن C به معنی وجود پرتوی ρ به گونه ایست که کامل در ناحیه ممکن قرار بگیرد و تابع هدف در طول ρ مقادیر بزرگ دلخواه بگیرد.

p نقطه شروع ρ و $\vec{d}$ بردار جهت ρ است.

$$\rho = \{p + \lambda \vec{d} : \lambda > 0\}$$


تابع هدف $f_{\vec{c}}$ بیشترین مقدار را می گیرد اگر $\vec{d} \cdot \vec{c} > 0$
 به عبارت دیگر $h \in H, \vec{d} \cdot \vec{n}(h) \geq 0$

$\vec{n}(h)$: بردار نرمال نیم صفحه h و عمود بر کران h رو به ناحیه feasible

اثبات : (برگشت) چون $(H', \vec{c})$ یک مسئله feasible است یک نقطه $p_0 \in \cap h$ وجود

دارد .
 $\rho_0 = \{p_0 + \lambda \vec{d} : \lambda > 0\}$

چون $\vec{d} \cdot \vec{n}(h) = 0$ برای هر $h \in H'$ اشعه ρ_0 کاملاً در h قرار می گیرد و تابع هدف $f \vec{c}$ مقادیر بزرگ دلخواه را در طول ρ_0 خواهد داشت.

برای یک نیم صفحه $h \in H \setminus H'$ داریم $\vec{d} \cdot \vec{n}(h) > 0$ در این صورت باید یک پارامتر λ_h وجود داشته باشد به طوریکه $p_0 + \lambda \vec{d} \in h$ برای هر $\lambda \geq \lambda_h$.

فرض کنید $\lambda' = \max_{h \in H \setminus H'} \lambda_h$ و $p = p_0 + \lambda' \vec{d}$

در نتیجه $\rho = \{p + \lambda \vec{d} : \lambda > 0\}$ کاملاً در هر نیم صفحه $h \in H$ قرار می گیرد ،
بنابراین $(H, \vec{c})$ بدون کران است .

اگر مختصات را طوری دوران دهیم که بردار $\vec{c}$ به صورت عمودی به سمت بالا باشد داریم
 $c=(0,1)$ و هر بردار جهت $d=(dx,dy)$ که را می توان به صورت $d=(dx,1)$ نوشت که با
 dx روی خط $y=1$ نشان داده می شود.

$$\vec{\eta}(h) = (\eta_x, \eta_y)$$

$$\vec{d} \cdot \vec{\eta}(h) = d_x \eta_x + \eta_y \geq 0$$

$$d_x \eta_x \geq -\eta_y.$$

بنابراین یک دستگاه از نامعادله خطی یا به عبارت دیگر یک مسئله خطی یک متغیره $\bar{H}$ داریم .

• مسئله جواب داشته باشد:

d_x^* جواب برنامه خطی $\bar{H}$ باشد. حال برای $h \in H' \subseteq H$ داریم: $d_x^* \eta_x + \eta_y = 0$

پس نرمال خارجی همه صفحات H' بر بردار $\vec{d} = (d_x^*, 1)$ عمود هستند یعنی خودشان با

هم موازی هستند و با محور x ها برخورد دارند که به برنامه ی خطی یک بعدی مشابه قبل

تبدیل می شود که در زمان خطی قابل حل است. حال اگر این برنامه خطی یک بعدی که $\bar{H}'$

نامیده می شود، ممکن باشد آنگاه H' هم ممکن است آنگاه $\bar{H}$ بی کران است پس برنامه خطی

حالت بی کران دارد.

به طور خلاصه: بردار d یافت گردید حال اگر d در شرایط صدق کرد ناحیه بی کران و اگر H'

ناممکن باشد پس H ناممکن است.

- مسئله جواب نداشته باشد:

اگر $\bar{H}$ نشدنی باشد طبق لم برنامه خطی کراندار است.

$\bar{H}$ نشدنی است اگر و تنها اگر راست ترین محدودیتی که از سمت چپ مسئله را کراندار می کند سمت راست چپ ترین محدودیتی باشد که از سمت راست مسئله را محدود می کند.

اگر دو محدودیت مشترک تهی داشته باشند آنگاه مسئله کراندار است.

ما این دو محدودیت را به عنوان m_1, m_2 در نظر می گیریم پس کرانداری برنامه خطی اثبات می شود.

Algorithm 2DRANDOMIZEDLP($H, \vec{c}$)

Input. A linear program $(H, \vec{c})$, where H is a set of n half-planes and $\vec{c} \in \mathbb{R}^2$.

Output. If $(H, \vec{c})$ is unbounded, a ray is reported. If it is infeasible, then two or three certificate half-planes are reported. Otherwise, the lexicographically smallest point p that maximizes $f_{\vec{c}}(p)$ is reported.

1. Determine whether there is a direction vector $\vec{d}$ such that $\vec{d} \cdot \vec{c} > 0$ and $\vec{d} \cdot \vec{\eta}(h) \geq 0$ for all $h \in H$.
2. **if** $\vec{d}$ exists
3. **then** compute H' and determine whether H' is feasible.
4. **if** H' is feasible
5. **then** Report a ray proving that $(H, \vec{c})$ is unbounded and quit.
6. **else** Report that $(H, \vec{c})$ is infeasible and quit.
7. Let $h_1, h_2 \in H$ be certificates proving that $(H, \vec{c})$ is bounded and has a unique lexicographically smallest solution.
8. Let v_2 be the intersection of ℓ_1 and ℓ_2 .
9. Let $h_3, h_4, \dots, h_n$ be a random permutation of the remaining half-planes in H .


```

10. for  $i \leftarrow 3$  to  $n$ 
11.     do if  $v_{i-1} \in h_i$ 
12.         then  $v_i \leftarrow v_{i-1}$ 
13.         else  $v_i \leftarrow$  the point  $p$  on  $\ell_i$  that maximizes  $f_{\vec{c}}(p)$ , subject to the
            constraints in  $H_{i-1}$ .
14.         if  $p$  does not exist
15.             then Let  $h_j, h_k$  (with  $j, k < i$ ) be the certificates (possibly
                 $h_j = h_k$ ) with  $h_j \cap h_k \cap \ell_i = \emptyset$ .
16.             Report that the linear program is infeasible, with
                 $h_i, h_j, h_k$  as certificates, and quit.
17. return  $v_n$ 

```

قضیه: یک مساله برنامه ریزی خطی دو بعدی با n محدودیت در زمان مورد انتظار $O(n)$ و حافظه مصرفی خطی حل می شود.

با تشکر از توجه شما