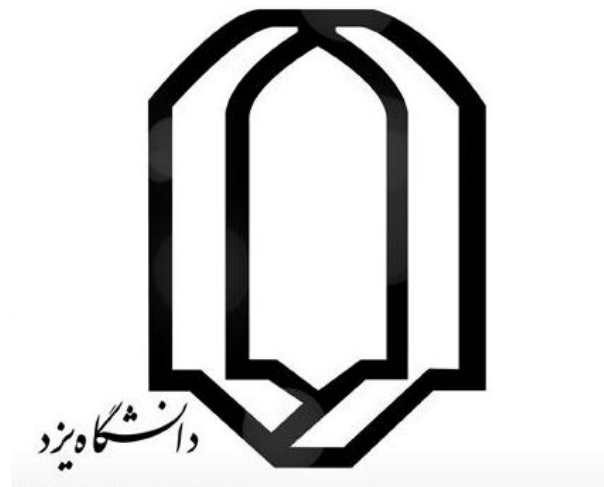




سمینار درس هندسه محاسباتی

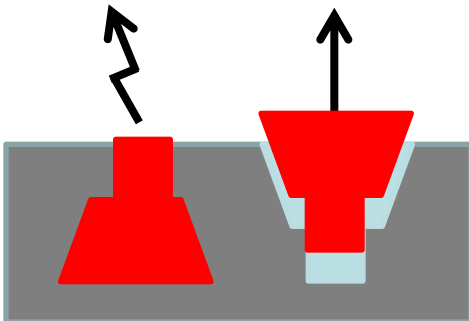
ارایه دهنده :
سعید ریخته گر غیاثی

زمستان ۹۰

برنامه ریزی خطی

Linear programming

Manufacturing with models



منبع:

Computational Geometry,
Theory and Applications, de Berg,
et al.

اهداف

- ۱ - تبدیل مساله قالب گیری به مساله برنامه ریزی خطی
- ۲ - حل مساله برنامه ریزی خطی با کمک روش های هندسه محاسباتی
- ۳ - ارائه یک الگوریتم تصادفی برای بهبود زمان حل مساله برنامه ریزی خطی

برنامه ریزی خطی

- الگوریتم بحث شده در مورد یافتن اشتراک نیم صفحه ها از مرتبه $O(n \log n)$ بود.
- آیا الگوریتم بهتری هم وجود دارد؟
- در مساله قالب گیری کافی است تنها یک نقطه در اشتراک نیم صفحه ها پیدا کنیم، پس ممکن است الگوریتم بهتری هم وجود داشته باشد.

برنامه ریزی خطی

- مساله پیدا کردن یک جواب برای مجموعه ای از محدودیت های خطی به مساله برنامه ریزی خطی یا بهینه سازی خطی معروف است.

$$\max \quad c_1x_1 + c_2x_2 + \cdots + c_dx_d$$

$$s.t. \quad a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,d}x_d \leq b_1$$

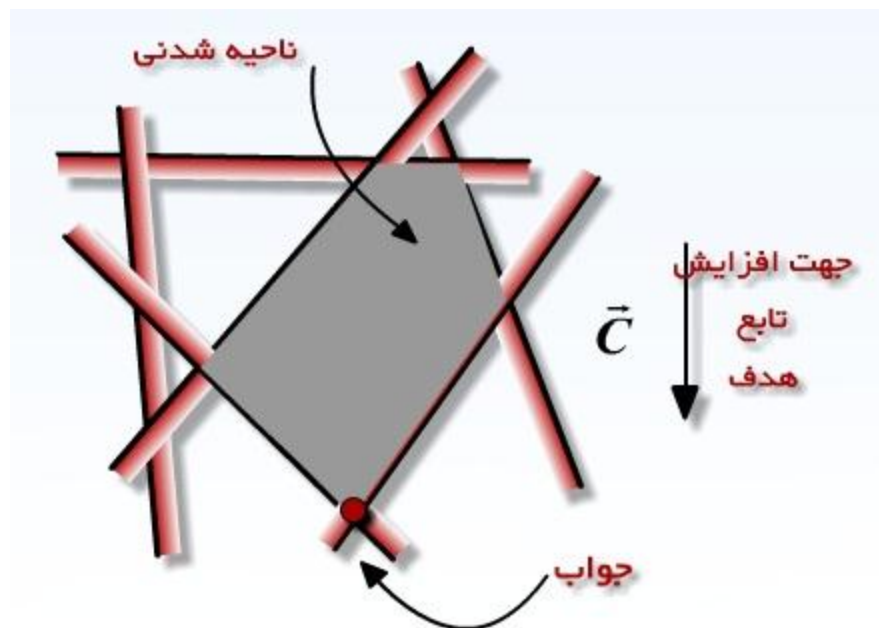
$$a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,d}x_d \leq b_2$$

$$\vdots$$

$$a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,d}x_d \leq b_n$$

برنامه ریزی خطی

- هر نقطه ای که در تمام قیدها صدق کند یک نقطه **شدنی** نام دارد. اشتراک همه نیم صفحه ها نیز ناحیه شدنی نام دارد.
- نقطه شدنی که مقدار تابع هدف به بیشترین مقدار ممکن می رسد، جواب مساله است.



برنامه ریزی خطی

- برای حل این مساله در تحقیق در عملیات روش های مختلفی وجود دارد. مهم ترین این روش ها که در عمل نیز بسیار مورد استفاده قرار می گیرد، روش **سیمپلکس** است.
- در این روش مساله ما یک مساله ۲-بعدی با n محدودیت است.
- برای استفاده از روش های بهینه سازی خطی باید یک تابع هدف دل خواه تعریف و مساله را حل کنیم تا یک جواب بدست آوریم.

برنامه ریزی خطی

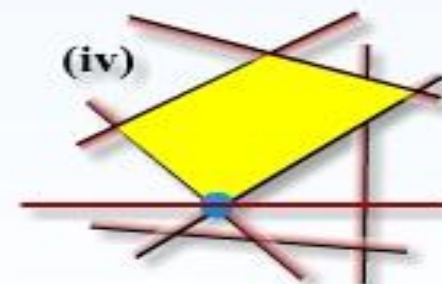
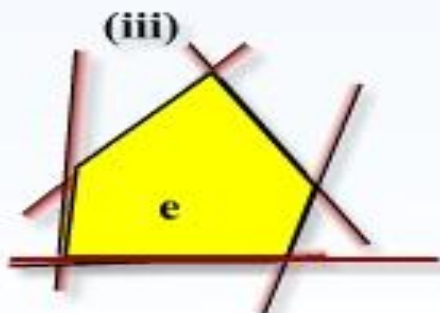
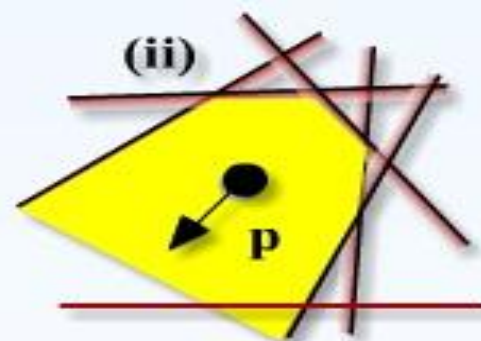
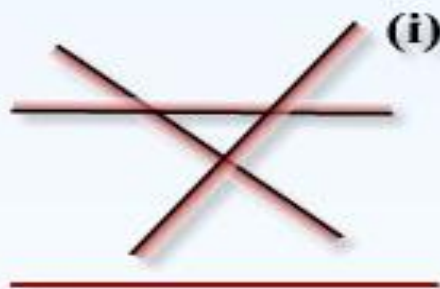
- روش های تحقیق در عملیات برای مسایل با اندازه کوچک مناسب نیستند، پس ما به دنبال حل مساله با روش **هندسه محاسباتی** هستیم.

برنامه ریزی خطی

- مجموعه محدودیت های مساله را با $H = \{h_1, h_2, \dots, h_n\}$ نشان می دهیم.
- بردار $\vec{c} = (c_x, c_y)$ تابع هدف را تعریف می کند، پس تابع هدف مساله به صورت $f_{\vec{c}}(p) = c_x p_x + c_y p_y$ است.
- هدف ما تعیین نقطه ای مانند $p \in \mathbb{R}^2$ است که $p \in \cap H$ و $f_{\vec{c}}(p)$ ماکزیمم است.

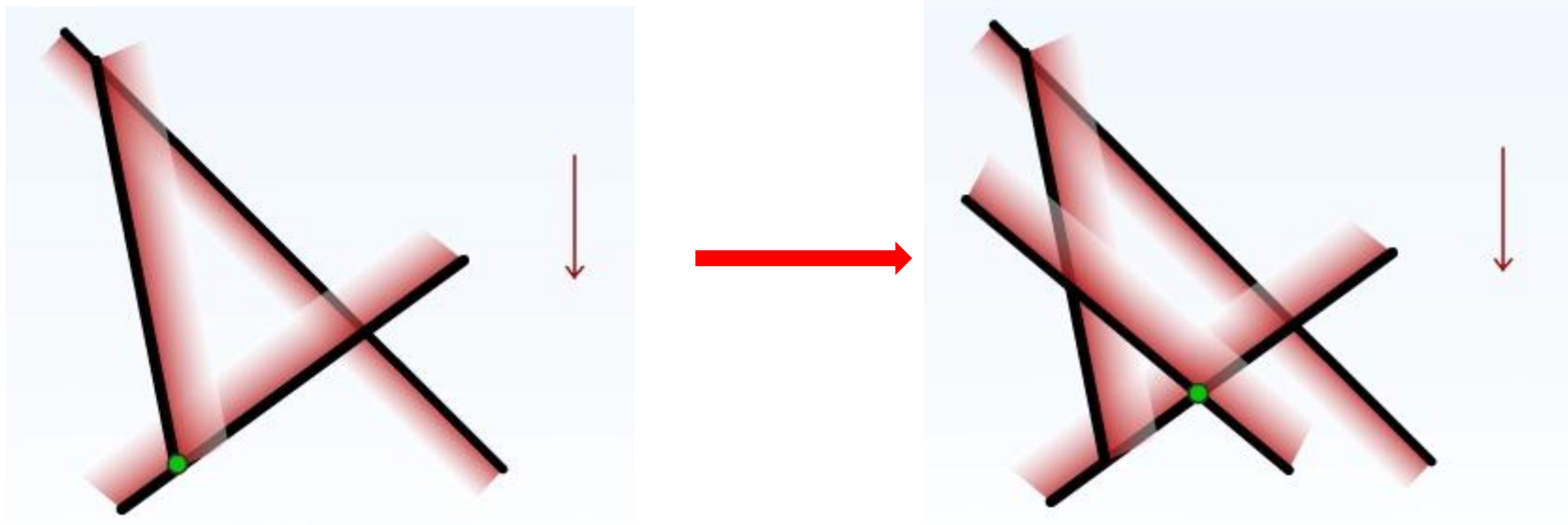
برنامه ریزی خطی

- این مساله برنامه ریزی خطی را با $(H, \bar{c})$ و ناحیه شدنی آن را با C نشان می دهیم.
- حالت های مختلف مساله :



برنامه ریزی خطی

- ایده راه حل هندسی مساله استفاده از یک الگوریتم افزایشی است.
- در هر مرحله یک محدودیت اضافه می شود و برای مساله جدید یک جواب بهینه بدست می آید.



برنامه ریزی خطی

- در این الگوریتم لازم است که در هر مرحله مساله دارای جواب بهینه یکتا باشد. اما ممکن همیشه این شرط برقرار نشود.

- - مساله بی کران باشد.

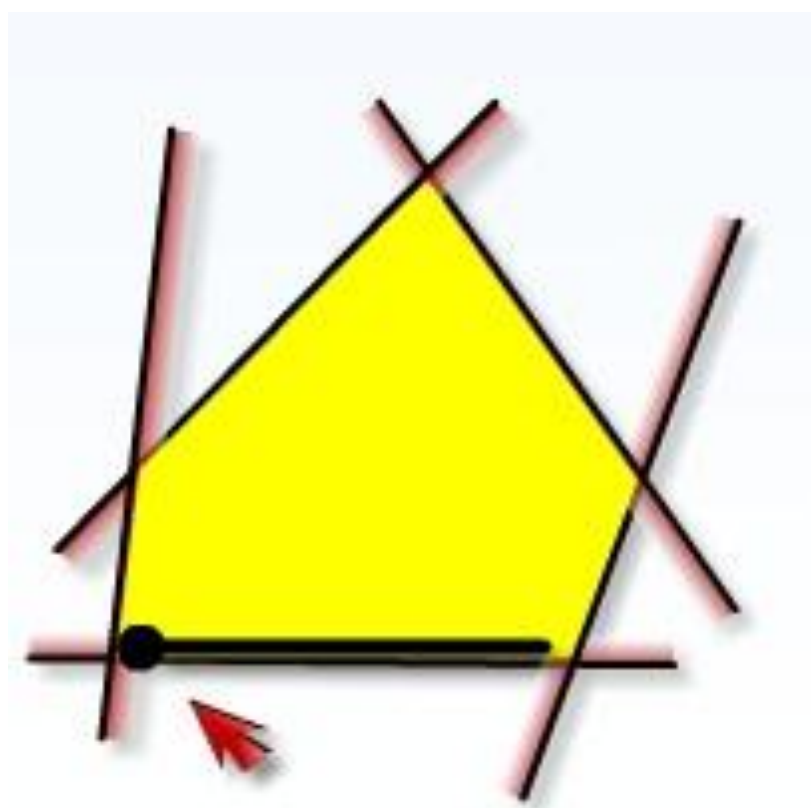
دو محدودیت جدید اضافه می کنیم.

$$m_1 := \begin{cases} p_x \leq M & \text{if } c_x > 0 \\ -p_x \leq M & \text{otherwise} \end{cases}$$

$$m_2 := \begin{cases} p_y \leq M & \text{if } c_y > 0 \\ -p_y \leq M & \text{otherwise} \end{cases}$$

برنامه ریزی خطی

- - مساله بی نهایت جواب داشته باشد، در این حالت از ترتیب الفبایی استفاده می کنیم.



برنامه ریزی خطی

- جواب بهینه هر مرحله یک راس از ناحیه شدنی است که آن را **راس بهینه** می نامیم.

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

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

- قرار می دهیم :

- و جواب یکتای مرحله i را با v_i نشان می دهیم.

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

- داریم :

- در نتیجه اگر برای یک i ، $C_i = \emptyset$ آنگاه برای هر $j \geq i$ ، $C_j = \emptyset$.

برنامه ریزی خطی

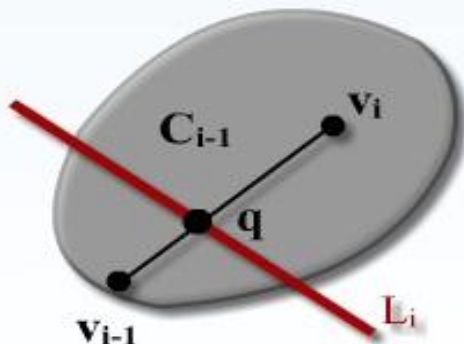
- **لم :** با فرض های قبل برای هر $1 \leq i \leq n$ داریم :
- (۱) اگر $v_{i-1} \in h_i$ آنگاه $v_i = v_{i-1}$.
- (۲) اگر $v_{i-1} \notin h_i$ آنگاه $C_i = \emptyset$ یا $v_i \in l_i$ که در آن l_i خطی است که h_i را محدود می کند.

برنامه ریزی خطی

• ایده اثبات :

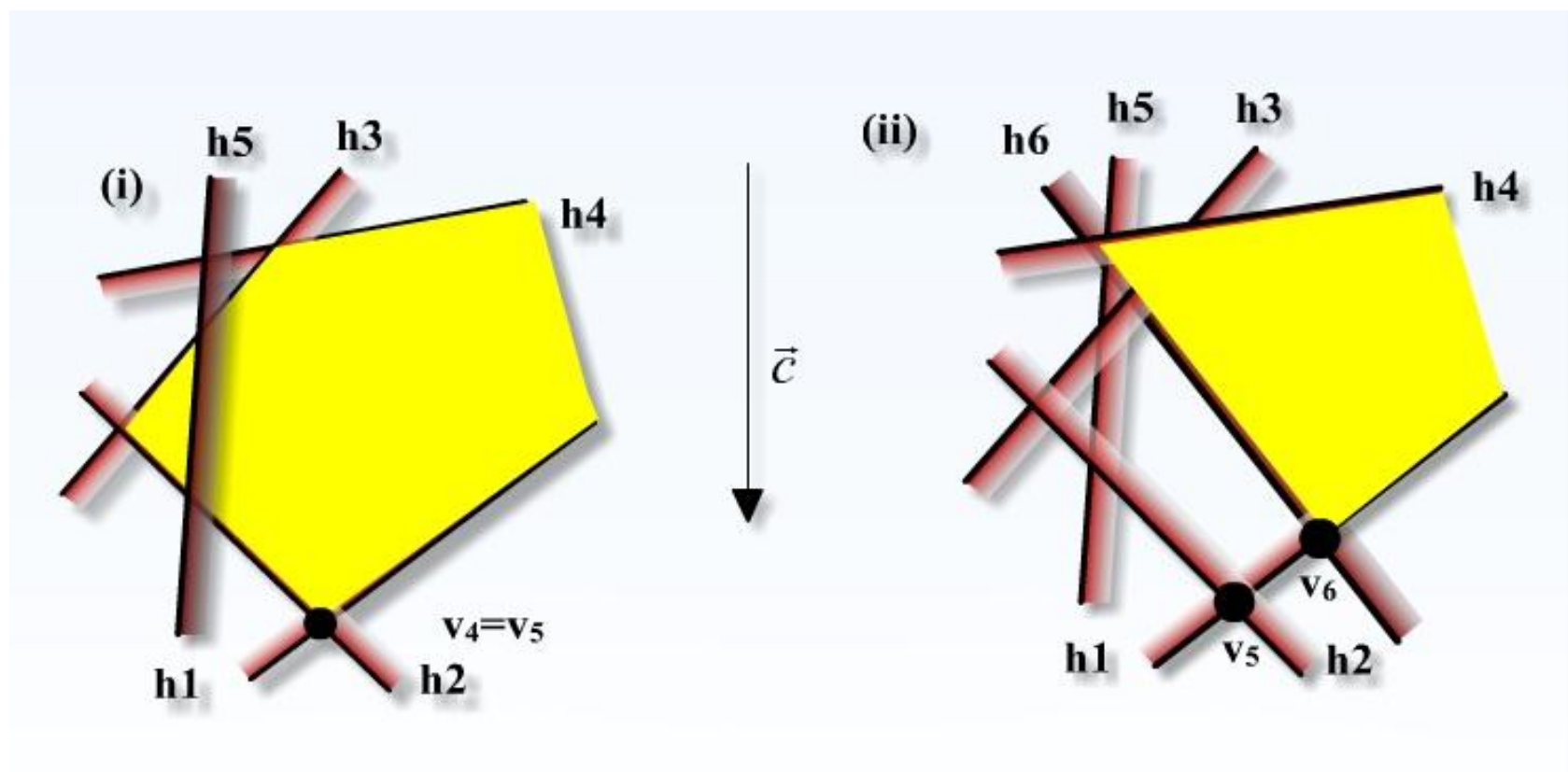
• اگر $v_{i-1} \in h_i$ آنگاه $v_{i-1} \in C_i$. راس بهینه C_i نمی تواند از راس بهینه C_{i-1} بهتر باشد. زیرا C_i زیر مجموعه C_{i-1} است.

• اگر $v_{i-1} \in h_i$ آنگاه خط $v_{i-1} v_i$ درون C_{i-1} و تابع هدف در راستای آن صعودی است، اگر نقطه تقاطع l_i با این پاره خط q باشد، آنگاه $f_{\bar{c}}(q) > f_{\bar{c}}(v_i)$ که تناقض است.



برنامه ریزی خطی

- دو حالتی که پس از اضافه کردن نیم صفحه جدید رخ می دهد.



برنامه ریزی خطی

• با توجه به لم قبل، اگر $v_{i-1} \notin h_i$ باشد، مساله به صورت زیر در می آید :

پیدا کردن نقطه p روی خط l_i ،

که تابع هدف را ماکسیمم کند و برای هر $h \in h_i$ ، $p \in h$.

$$\bar{f}\vec{c} : R \mapsto R$$

$$\bar{f}\vec{c}(p) = \bar{f}\vec{c}(p_x)$$

$$p \in l_i$$

برای حل این مساله فرض می کنیم l_i عمودی نباشد و تابع هدف یک بعدی جدید را تعریف می کنیم :

برنامه ریزی خطی

Maximize $\bar{f}c(x)$

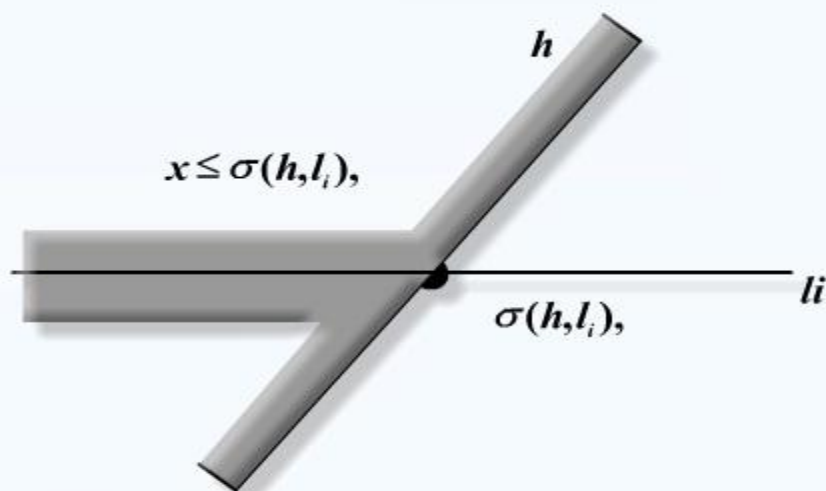
subject to $x \geq \sigma(h, l_i), \quad h \in H_i$

$x \leq \sigma(h, l_i), \quad h \in H_i$

• مساله به صورت زیر در می آید:

اگر $l_i \cap h$ از چپ کران دار باشد :

اگر $l_i \cap h$ از راست کران دار باشد :



که در آن $\sigma(h, l_i)$ مختص
 x تقاطع l_i با h است.

برنامه ریزی خطی

- {اگر $l_i \cap h$ از راست کران دار باشد $\sigma(h, l_i)$ $x_{right} = \min$ }
- {اگر $l_i \cap h$ از چپ کران دار باشد $\sigma(h, l_i)$ $x_{left} = \max$ }
- جواب مساله در بازه $[x_{left}: x_{right}]$ قرار دارد.
- اگر $x_{left} > x_{right}$ مساله جواب ندارد، در غیر اینصورت یکی از نقاط انتهایی جواب است.

برنامه ریزی خطی

- مساله گفته شده در زمان $O(i)$ حل می شود.
- حال آماده ایم الگوریتم را به طور رسمی ارائه کنیم.

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

برنامه ریزی خطی

- الگوریتم **2D Bounded LP** مساله برنامه ریزی خطی $O(n^2)$ و با استفاده از $O(n)$ حافظه حل می کند.
- ایده اثبات :** درستی الگوریتم به سادگی از مطالبی که تا کنون گفته شده است، نتیجه می شود. برای بررسی زمان اجرای الگوریتم توجه کنید پس از اضافه شدن i امین نیم صفحه باید مساله بهینه سازی یک بعدی در خط ۶ الگوریتم حل شود و این کار زمان $O(i)$ لازم دارد.

برنامه ریزی خطی

- پس زمان اجرای الگوریتم برابر است با

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



- آیا می توان زمان اجرای الگوریتم را کمتر کرد؟

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

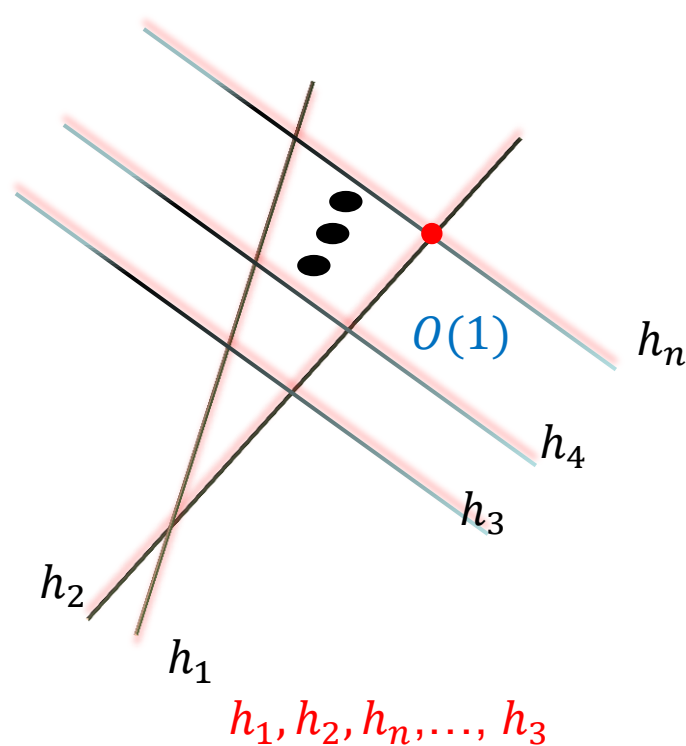
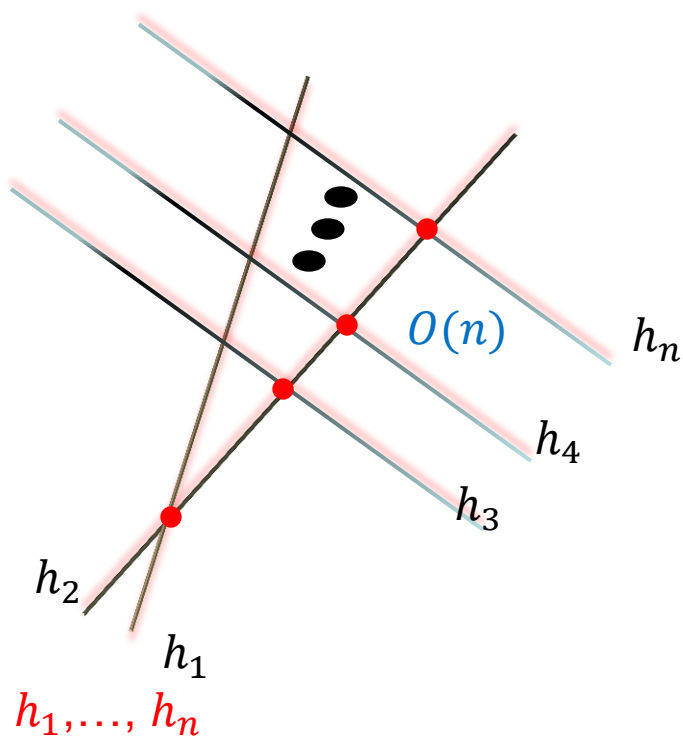
Input. A linear program $(H \cup \{m_1, m_2\}, \vec{c})$, where H is a set of n half-planes, $\vec{c} \in \mathbb{R}^2$, and m_1, m_2 bound the solution.

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 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 h_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

برنامه ریزی خطی

- آیا می توان تعداد دفعات تغییر راس بهینه را محاسبه کرد؟
- برای جواب به این سوال از یک مثال استفاده می کنیم.



برنامه ریزی خطی

- پس برای کاهش زمان اجرای الگوریتم باید ترتیب نیم صفحه ها اصلاح شود.
- یافتن ترتیبی که گفتیم ساده نیست. اما می توان از یک ترتیب تصادفی استفاده کرد.

برنامه ریزی خطی

• الگوریتم تصادفی

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

برنامه ریزی خطی

- الگوریتم تصادفی

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** $\text{rndindex} \leftarrow \text{RANDOM}(k)$
3. Exchange $A[k]$ and $A[\text{rndindex}]$.

برنامه ریزی خطی

- زمان اجرای الگوریتم
- به دست آوردن زمان اجرای الگوریتم ساده نیست و به ترتیبی که برای نیم صفحه ها تعیین می شود بستگی دارد.
- برای بدست آوردن زمان اجرای الگوریتم میانگین زمان اجرای الگوریتم روی همه $n!$ ترتیب مختلف را بدست می آوریم و آن را **زمان مورد انتظار** اجرای الگوریتم می نامیم.

برنامه ریزی خطی

- **قضیه :** برای مساله برنامه ریزی خطی دو متغیره با n محدودیت زمان مورد انتظار اجرای الگوریتم *2D Randomized bounded LP* خطی است.
- یادآوری : امید ریاضی و یا میانگین متغیر تصادفی

$$E(x) = \begin{cases} \sum xf(x) \\ \int xf(x) \end{cases} \quad f(x) : \text{تابع چگالی احتمال}$$

برنامه ریزی خطی

- ایده اثبات : زمان اجرای *RANDOMPERMUTATION* برابر
- است با $O(n)$.

برای اضافه کردن صفحات، متغیر تصادفی X_i را بدین صورت تعریف می کنیم:

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

برنامه ریزی خطی

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

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

با توجه بودن به خطی بودن امید ریاضی:

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

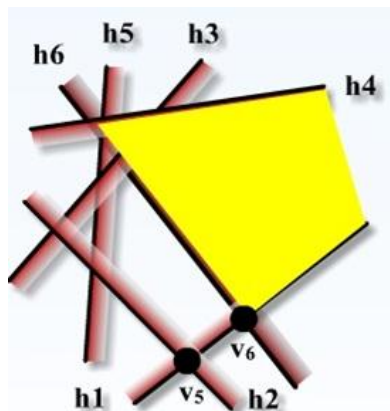
برنامه ریزی خطی

• سؤال: $E(X_i)$ چیست؟

✓ پاسخ: احتمال اینکه $v_{i-1} \notin h_i$

، یعنی احتمال اینکه نقطه بهینه در این مرحله تغییر کند. برای تحلیل این احتمال از روند **Backwards Analysis** استفاده می شود، یعنی فرض می شود که الگوریتم انجام گرفته و راس بهینه v_n بدست آمده، حال صفحه h_n را حذف می کنیم،

با چه احتمالی نقطه بهینه تغییر می کند؟

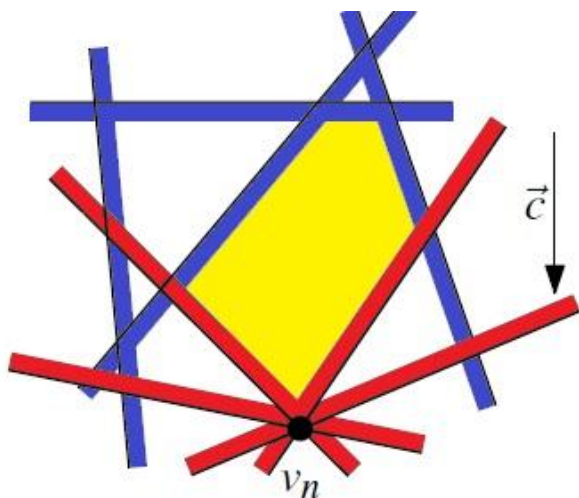


برنامه ریزی خطی

✓ از آنجا که v_n راس C_n می باشد،
حداقل توسط دو تا از نیم صفحه ها تعریف می شوند
و وقتی v_n تغییر می کند که راسی از C_{n-1}
نباشد، یعنی h_n یکی از نیم صفحه هایی است که v_n را
تعریف می کند. 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 $2/n$
می باشد.

برنامه ریزی خطی

- چرا حداکثر $2/n$ ؟
- پاسخ: اولاً 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. دوماً 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 $2/n$ کمتر می کند.



برنامه ریزی خطی

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

$$E(X_i) \leq 2/i$$

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

$$\sum \left(o(i) \cdot \frac{2}{i} \right) = o(n)$$



برنامه خطی بی کران

- در قسمت قبل برنامه خطی نامحدود را با اعمال دو محدودیت اضافی کراندار کردیم، اما این 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.
- چگونه کراندار بودن یا نبودن برنامه خطی $(H, \vec{c})$ را تشخیص بدهیم؟

برنامه خطی بی کران

- **قضیه :** يك برنامه خطی $(H, \vec{c})$ غیرکراندار است اگر و فقط اگر يك بردار 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

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

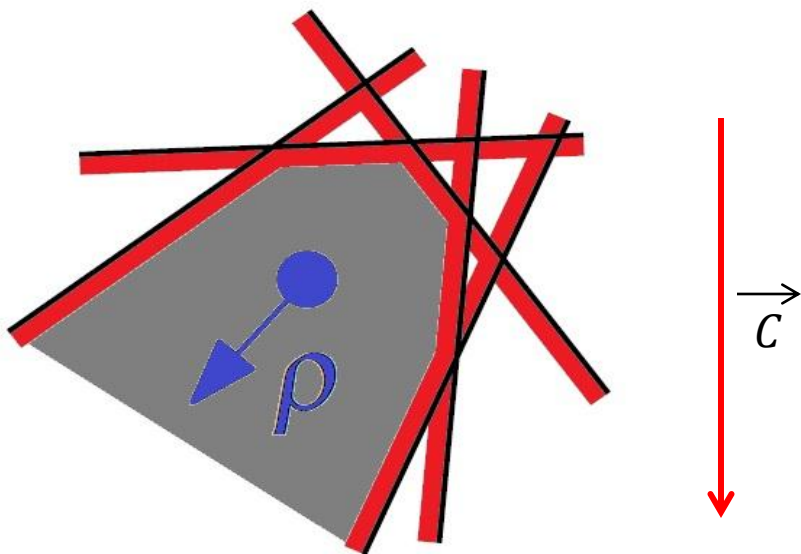
برنامه خطی بی کران

- ایده اثبات :
- **طرف رفت:** قبلا گفته شد که بیکران بودن C به معنی وجود پرتوی p به گونه ایست که کامل در ناحیه ممکن قرار بگیرد و تابع obj در طول p مقادیر بزرگ دلخواه بگیرد.

برنامه خطی بی کران

- p : نقطه شروع p ، $\vec{d}$: بردار جهت آن $\Leftarrow \rho = \{p + \lambda \vec{d} : \lambda > 0\}$
- تابع obj در طول ρ مقادیر بزرگ می گیرد اگر و فقط اگر $\vec{d} \cdot \vec{c} > 0$. از طرف دیگر اگر $\eta(h)$ بردار نرمال نیم صفحه h به سمت ناحیه ممکن باشد، آنگاه داریم:

$$\vec{d} \cdot \vec{\eta}(h) = 0$$



برنامه خطی بی کران

- **طرف برگشت:** برنامه خطی $(H, \vec{c})$ و بردار $\vec{d}$ با شرایط داده شده در لم مفروض است. چون $(H', \vec{c})$ ممکن است پس نقطه $p_0 \in \bigcap_{h \in H'} h$ وجود دارد. فرض کنید پرتوی $\rho_0 := \{p_0 + \lambda \vec{d} : \lambda > 0\}$ ، از آنجا که برای
- $\vec{d} \cdot \vec{\eta}(h) = 0$ ، $h \in H'$ پرتوی ρ_0 به طور کامل در هر کدام از صفحات H' قرار می گیرد. بنابراین از آنجا که $\vec{d} \cdot \vec{c} > 0$ تابع obj در طول ρ_0 مقادیر بزرگ دلخواه می گیرد.

برنامه خطی بی کران

برای هر نیم صفحه $h \in H \setminus H'$ ، داریم $\vec{d} \cdot \vec{\eta}(h) > 0$ که نتیجه می دهد پارامتر λ_h وجود دارد، که برای تمام $\lambda \geq \lambda_h$ ، $p_0 + \lambda d \in h$. با فرض $\lambda' := \max_{h \in H \setminus H'} \lambda_h$ و $P := P_0 + \lambda'$ پرتوی p_0 به طور کامل در هر کدام از صفحات عضو H قرار می گیرد. در نتیجه $(H, \vec{c})$ بیکران است.



برنامه خطی بی کران

- اگر ما مختصات را طوری دوران دهیم که بردار $\vec{c}$ به صورت عمودی به سمت بالا باشد داریم $c=(0,1)$ و هر بردار جهت $\vec{d} = (d_x, d_y)$ که در لم گفته شده صدق کند می توان به صورت $d = (d_x, 1)$ نوشت که با d_x روی خط $y = 1$ نشان داده می شود.

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

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

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

برنامه خطی بی کران

• الگوریتم :

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 ℓ_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

برنامه خطی بی کران

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

تشکر از توجه شما عزیزان