به نام حضرت دوست

هندسه ی محاسباتی

فصل ۴

فاطمه سلطانی

بهار ۸۹

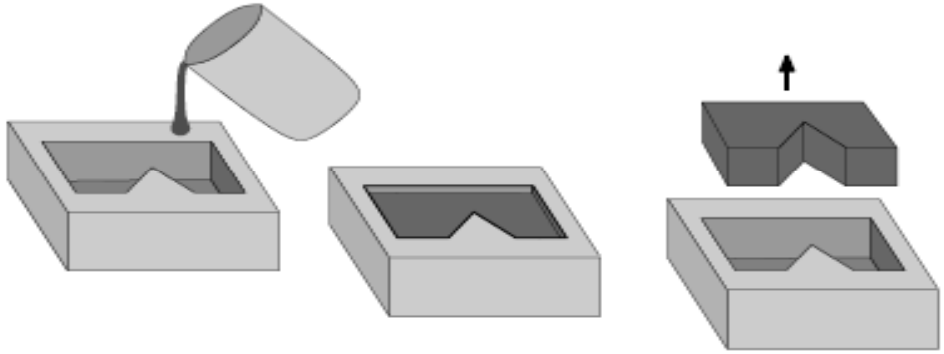
مقدمه

بسياري از اجسام اطراف ما، مانند شكل ماشينها، كاپ هاي پلاستيكي و ... با استفاده از شكل دهي اتوماتيكي ساخته مي شوند. كامپيوترها نقش مهمي را هم در فاز طراحي و هم در فاز ساختاربندي ايجاد مي كنند، بنابراين امكانات CAD/CAM بخش حياتي هر كارخانه مدرن مي باشد. فاز ساختاربندي در واقع به بيان ساختن يك شئ خاص باتوجه به فاكتورهايي مثل: ماده شئ اي كه بايد ساخته شود، شكل شئ، اينكه آيا شئ به مقدار زياد قابل ساختن مي باشد يا نه و... مي پردازد.

در اين فصل به مطالعه منظرهاي هندسي ساختن با قالبها، كه يك پروسه رايج استفاده شده براي اشياء فلزي يا پلاستيكي است، پرداخته مي‌شود. براي اشياء فلزي اين پروسه به عنوان ريخته گري بيان مي‌شود.

مقدمه

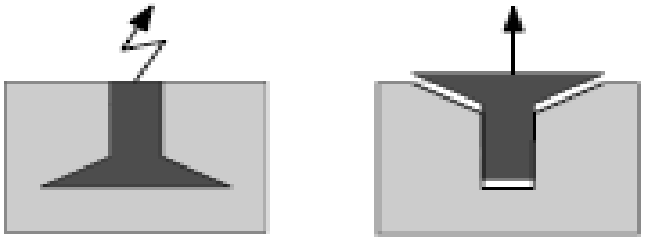
شكل ۱-۴ پروسه ريخته گري را شرح مي دهد. فلز مايع داخل يك قالب ريخته مي‌شود، منجمد مي گردد و شئ از قالب بيرون مي آيد. مرحله آخر به همين سادگي انجام نمي‌شود، ممكن است بدون شكستن قالب شئ بيرون نيايد. ممكن است تنها با چرخاندن قالب شئ بيرون بيايد، وگاهي اشيائي وجود دارند كه قالبي براي آنها وجود ندارد، مثل كره. بنابراين در اين فصل بر روي وجود يا عدم وجود قالبي براي يك شئ كه بتواند از آن خارج شود بحث مي گردد.



شكل ۱-۴

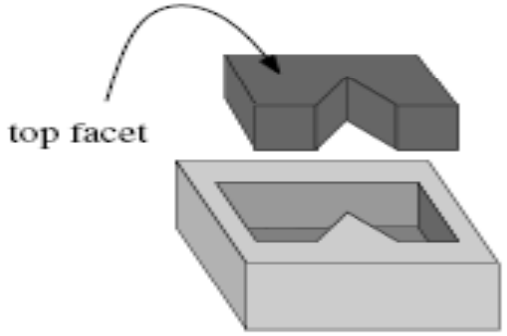
مقدمه

- قرارداد ۱: شئ چندوجهي درنظرگرفته مي‌شود.
- قرارداد ۲: قالبها فقط يك تكه اي درنظر گرفته مي شوند.(قالبهاي دوتكه اي براي ساخت اشيائي مثل كره مورد استفاده قرار مي گيرند.)
- قرارداد ۳: هر شئ را تنها با يك حركت مي توان از قالب بيرون آورد.(مثلا يك پيچ را نمي توان از قالبش بيرون آورد.) براي اين كار حركت انتقالي مناسب است.



۴-۱ هندسه ي ريخته گري

- مسئله: يك شئ با ريخته گري قابل ساخت است يا نه؟
- پاسخ اول: بايد براي آن يك قالب مناسب بيابيم كه از آن خارج شود. شكل حفره داخل قالب توسط شكل شئ معين مي‌شود ولي جهتهاي مختلف شئ قالب هاي متمايزي را ايجاد مي كنند، كه اين انتخاب جهت مي تواند سخت باشد، چون در بسياري از جهتها شئ از قالبش خارج نمي‌شود، درحاليكه در جهتهاي ديگر ممكن است خارج شود.
- يك محدوديت روي جهت اين است كه شئ بايد يك top facet افقي داشته باشد. اين سطح تنها جايي است كه درتماس با قالب نخواهد بود.
- پاسخ دوم: شئ قابل ريخته گري است اگرحداقل از يك جهت با top facet متناسب با آن جهت قابل بيرون آمدن از قالبش باشد.



۴-۱ هندسه ي ريخته گري

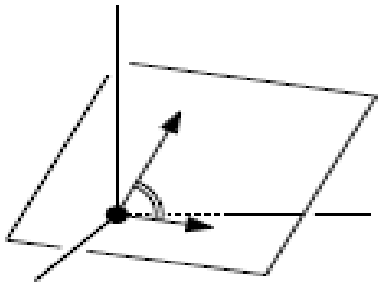
- در ادامه بر روي تعيين اينكه "آيا يك شئ با يك انتقال قابل بيرون آمدن از يك قالب معين مي باشد يا نه" بحث مي‌شود و براي تصميم گيري روي قابليت ريخته گري يك شئ "هر جهت ممكن" مورد بررسي واقع مي‌شود.
- فرض كنيد $\mathcal{P}$ يك چندوجهي سه بعدي است كه با سطوح دو بعدي و يك top facet معين طراحي شده است.(دراينجا نيازي نيست كه يك تعريف دقيق از polyhedron ارائه شود.)
- قالب يك بلوك مستطيلي با حفره اي دقيقا مطابق $\mathcal{P}$ مي باشد. وقتي چندوجهي داخل قالب قرار مي گيرد، top facet آن بايد هم سطح با بالاترين سطح قالب باشد، يعني قالب هيچ قسمت اضافي كه ممكن است مانع خروج شئ شود در سطح بالايي اش ندارد.

۴-۱ هندسه ي ريخته گري

- هر سطح $\mathcal{P}$ به غير از top facet سطح ordinary ناميده مي‌شود.
- هر سطح ordinary به نام f با يك سطح از قالب به نام $f\hat{}$ متناظر است.
- آيا يك بردار جهت $\vec{d}$ وجود دارد كه $\mathcal{P}$ بتواند بدون برخورد با سطوح داخلي قالب در طول انتقال خارج شود؟ (لغزيدن درطول قالب ايرادي ندارد.)
- ازآنجا كه تنها سطح بدون تماس با قالب top facet است، جهت خروج بايد در جهت مثبت محور zها باشد و اين تنها شرط موجود بر روي جهت است.

۱-۴ هندسه ي ريخته گري

- اگر f يك سطح ordinary، $\mathcal{P}$ باشد، آنگاه اين سطح يا بايد از يك طرف سطح متناظر $f\hat{}$ قالب حركت كند و يا بر روي $f\hat{}$ بلغزد. براي ايجاد اين دقت نياز است به زاويه بين دو بردار در فضاي سه بعدي.
- زاويه بين دو بردار: دو بردار از يك مبدا درنظر گرفته شده و صفحه گذرنده از آن دو را ساخته، كوچكترين زاويه بين دو بردار زاويه موردنظر است.

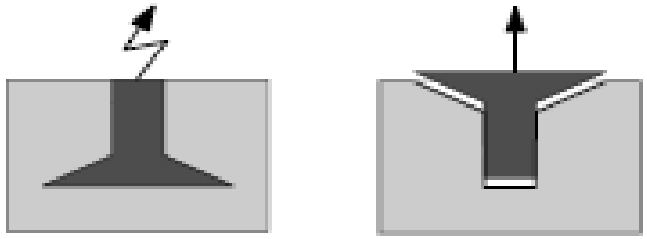


- شرط لازم روي $\vec{d}$: زاويه آن با بردار نرمال خارجي هر سطح ordinary، p حداقل ۹۰ باشد.(لم ۱-۴)

۱-۴ هندسه ي ريخته گري

- لم ۱-۴: چندوجهي $\mathcal{P}$ مي تواند در جهت $\vec{d}$ از قالبش خارج شود، اگر و فقط اگر زاويه حداقل ۹۰ با نرمال خارجي تمام سطوح ordinary، $\mathcal{P}$ بسازد.
- اثبات: only if:

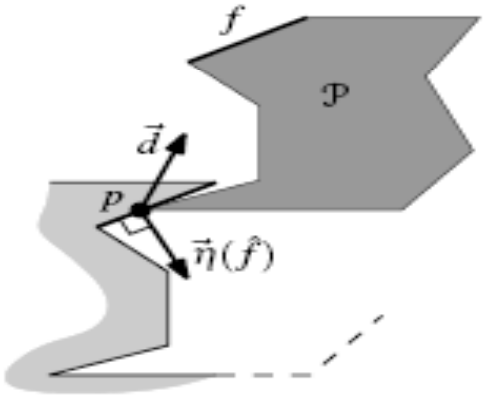
اگر $\vec{d}$ با نرمال خارجي يکي از سطوح ordinary، $(\mathcal{P})$ يعني زاويه کمتر از ۹۰ بسازد، هنگام انتقال تمام نقاط داخل f با قالب برخو



لم ۱-۴: چندوجهي $\mathcal{P}$ مي تواند در جهت $\vec{d}$ از قالبش خارج شود، اگر و فقط اگر زاويه حداقل ۹۰ با نرمال خارجي تمام سطوح ordinary، $\mathcal{P}$ بسازد.

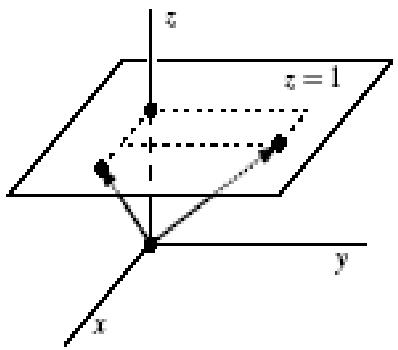
✓ If:

فرض خلف: در جايي $\mathcal{P}$ با قالب برخورد كند، بايد نشان دهيم كه يك نرمال خارجي كه زاويه كمتر از ۹۰ با $\vec{d}$ مي سازد، وجود دارد. فرض كنيد p نقطه اي از $\mathcal{P}$ است كه با يك سطح $\hat{f}$ قالب برخورد كرده است، اين بدين معني است كه p نمی تواند به سمت بيرون حركت كند، پس بايد زاويه بزرگتر از ۹۰ با $\vec{\eta}(\hat{f})$ بسازد، در نتيجه $\vec{\eta}(f)$ با $\vec{d}$ زاويه كمتر از ۹۰ مي سازد.



۱-۴ هندسه ي ريخته گري

- يك نتيجه جالب لم: اگر $\mathcal{P}$ را بتوان با دنباله اي از انتقال هاي كوچك از قالبش بيرون آورد، با يك انتقال هم مي توان. پس امكان استفاده بيش از يك انتقال كمكي به حل مسئله نمي كند.
- هدف: يافتن بردار $\vec{d}$ با ويژگي بيان شده در لم ۱-۴.
- يك جهت در فضاي سه بعدي با يك بردار كه از مبدا مختصات شروع مي‌شود، معين مي گردد. با توجه به مثبت بودن z بردار اين بردارها را به عنوان نقاطي در صفحه z=1 درنظر مي گيريم.



- نتيجه: هر نقطه در صفحه z=1 مثل (x,y,1)معرف بردار منحصر بفرد(x,y,1) مي باشد و برعكس.

۱-۴ هندسه ي ريخته گري

- سؤال: لم ۱-۴ شرايط لازم و كافي را روي جهت $\vec{d}$ ارائه داد، چگونه اين شرايط به صفحه جهت(صفحه z=1) انتقال يابد؟
- فرض كنيد $\vec{\eta}=(\eta_x,\eta_y,\eta_z)$ نرمال خارجي يك سطح ordinary باشد، بردار $\vec{d}=(d_x,d_y,1)$ يك زاويه حداقل ۹۰ با $\vec{\eta}$ مي سازد، اگر و فقط اگر ضرب داخلي $\vec{\eta}$ و $\vec{d}$ غيرمثبت شود:

$$\vec{\eta}_x d_x+\vec{\eta}_y d_y+\vec{\eta}_z \le 0$$

اين نامساوي توصيف كننده يك نيم صفحه روي صفحه z=1 مي باشد كه در طرف راست يا چپ يك خط روي اين صفحه مي باشد.

- نكته: اين شرط براي سطوح افقي صحيح نمي باشد، چون $\vec{\eta}_x=\vec{\eta}_y=0$ برقرار است و يا $\vec{\eta}_z\le 0$ برقرار نيست. درنتيجه يا $\vec{\eta}_z>0$ كه همواره

۱-۴ هندسه ي ريخته گري

- بنابراين هر سطح غير افقي $\mathcal{P}$ معرف يك نيم صفحه روي صفحه z=1 است، كه هر نقطه در تقاطع مشترك اين نيم صفحه ها معادل بردار جهتي است كه با آن مي توان شئ را از قالبش درآورد. اين تقاطع مشترك مي تواند تهي نيز باشد كه بدين معني است كه شئ قابل خارج شدن از قالبش نمي باشد.
- تبديل مسئله به يك مسئله هندسي: يافتن يك نقطه در تقاطع مشترك يك مجموعه داده شده از نيم صفحه ها و يا تصميم گيري اينكه تقاطع تهي است.
- اگر چندوجهي موردنظر n سطح داشته باشد، حداكثر n-1 نيم صفحه مورد بحث است.(حداكثر به دليل عدم حضور سطوح افقي و منهاي ۱ به دليل عدم حضور top facet)

۱-۴ هندسه ي ريخته گري

- از آنجا كه گفته شد براي تست قابليت ريخته گري $\mathcal{P}$ بايد تمام سطوح به عنوان top facet در نظر گرفته شوند، قضيه زير بيان مي‌شود.
- قضيه ۲-۴: اگر $\mathcal{P}$چندوجهي با n سطح باشد در زمان مورد انتظار $O(n^2)$ و حافظه مورد نياز $O(n)$ براي ذخيره سازي مي توان تصميم گرفت كه آيا $\mathcal{P}$ قابل ريخته گري است يا نه. به علاوه اگر $\mathcal{P}$ قابل ريخته گري باشد در زمان مشابهي مي توان يك قالب و يك جهت مناسب براي خروج آن يافت.

۲-۴ تقاطع نیم صفحه ها

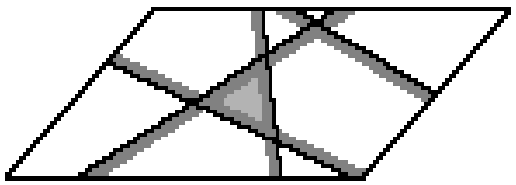
- فرض کنید که $H=\{h_1,h_2,..,h_n\}$ مجموعه اي از معادلات خطي دومتغیره باشد، که هر معادله به صورت $a_ix+b_iy \le c_i$ است. c_i و b_iو a_i ثابت هستند و b_iو a_i همزمان نمي توانند صفر باشند.

- از لحاظ هندسي يك معادله به عنوان يك نيم صفحه بسته در R^2 كه با خط $a_ix+b_iy = c_i$ كراندار شده بيان مي‌شود.

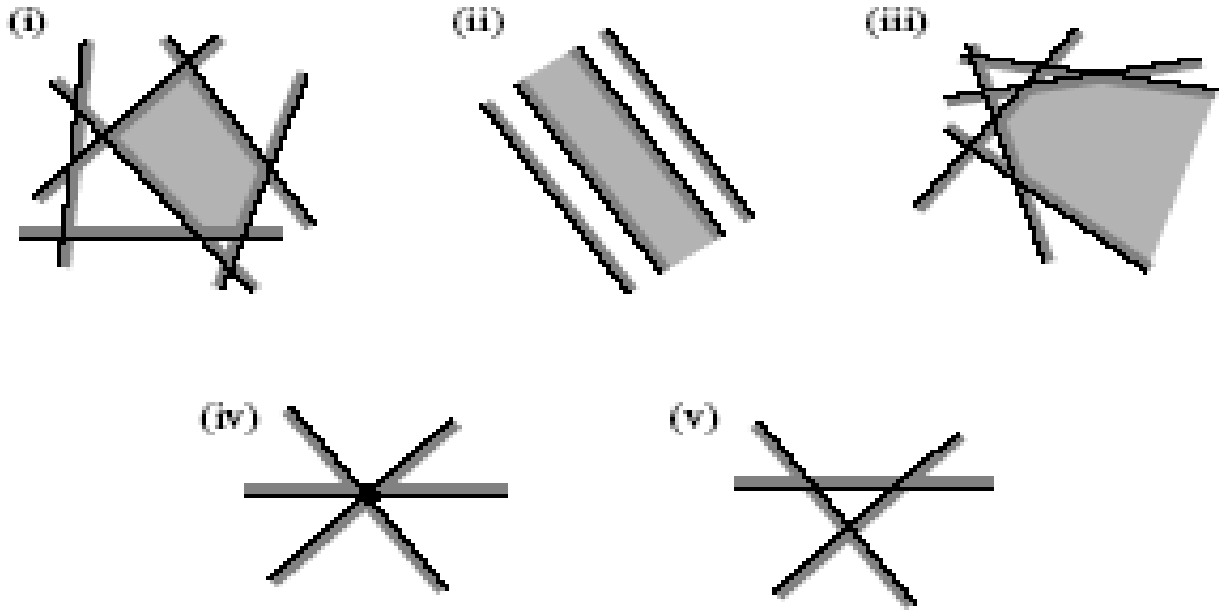
- مسئله: يافتن "تمام" نقاط $(x,y) \in R^2$ كه در همه n معادله صدق كند، به عبارت ديگر يافتن تمام نقاط واقع شده در تقاطع مشترك نيم صفحه هاي H.(در اين قسمت مسئله نسبت به قبل كلي تر مي‌شود.)

۲-۴ تقاطع نيم صفحه ها

- تعيين شكل ناحيه مشترك نيم صفحه ها:
  - هر نيم صفحه يك ناحيه محدب است واشتراك نواحي محدب نيز محدب است.
  - هر نقطه روي مرز تقاطع مشترك بايد روي مرز يكي ازنيم صفحه ها واقع شود، درنتيجه يالهاي سازنده ناحيه مشترك همان خطهاي مرزي بعضي از نيم صفحه ها مي باشند.
  - از آنجا كه ناحيه تقاطع محدب است، هر خط مرزي يك نيم صفحه مي تواند حداكثر سازنده يك يال باشد.
  - نتيجه: تقاطع مشترك n نيم صفحه يك چندضلعي محدب است كه حداكثر با n يال محدود شده است.



۲-۴ تقاطع نيم صفحه ها



- شكل فوق (۲-۴) چند مورد از اين تقاطع ها را نشان مي دهد. تقاطع هم مي تواند نامحدود باشد(شكلii و iii)، هم مي تواند به صورت يك نقطه يا خط باشد(شكلiv) و هم مي تواند تهي باشد(شكل v).

۲-۴ تقاطع نيم صفحه ها

- الگوريتم divide-and-conquer براي يافتن تقاطع مشترك n نيم صفحه ارائه مي‌شود، كه بر پايه يك INTERSECTREGIONCONVEX عادي براي محاسبه تقاطع دو ناحيه محدب است.

Algorithm INTERSECTHALFPLANES(H)
Input. A set H of n half-planes in the plane.
Output. The convex polygonal region $C := \bigcap_{h \in H} h$.

```
1.  if card(H) = 1
2.      then C ← the unique half-plane h ∈ H
3.      else Split H into sets H1 and H2 of size ⌈n/2⌉ and ⌊n/2⌋.
4.           C1 ← INTERSECTHALFPLANES(H1)
5.           C2 ← INTERSECTHALFPLANES(H2)
6.           C ← INTERSECTCONVEXREGIONS(C1, C2)
```

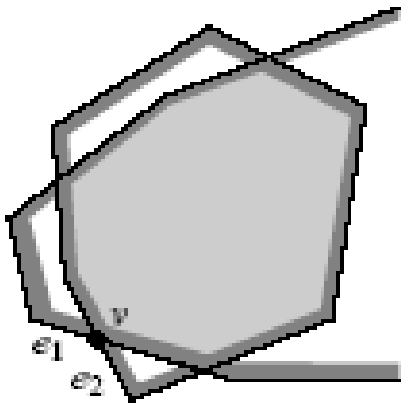
- البته در اينجا بايد توجه كرد كه نواحي مشترك مي توانند خط و يا نقطه هم باشند يعني لزوما چندضلعي نمي باشند، پس الگوريتم بايد اندكي اصلاح شود.

۲-۴ تقاطع نیم صفحه ها

- در نتيجه ۲-۷ ديده شد كه نقاط اشتراك دو چندضلعي در زمان O(nlogn+klogn) قابل محاسبه است، كه n مجموع تعداد رئوس در دو چندضلعي و k تعداد نقاط اشتراك بين دو ناحيه مي باشد.

- اگر v نقطه مشترك يال e_1 از C_1 و يال e_2 از C_2 باشد، بدون توجه به چگونگي اشتراك e_1 و e_2، v بايد يك راس $C_1 \cap C_2$ باشد،

و از آنجا كه $C_1 \cap C_2$ اشتراک n نيم صفحه است، حداكثر n يال و n راس دارد. اين نتيجه می دهد كه $k \le n$، پس زمان اجرا برابر است با:O(nlogn).



۲-۴ تقاطع نيم صفحه ها

- زمان اجراي كل الگوريتم:

$$T(n)=\begin{cases} o(1) & n=1 \\ o(n\log n)+2T(n/2) & n>1 \end{cases}$$

- بنابراين:

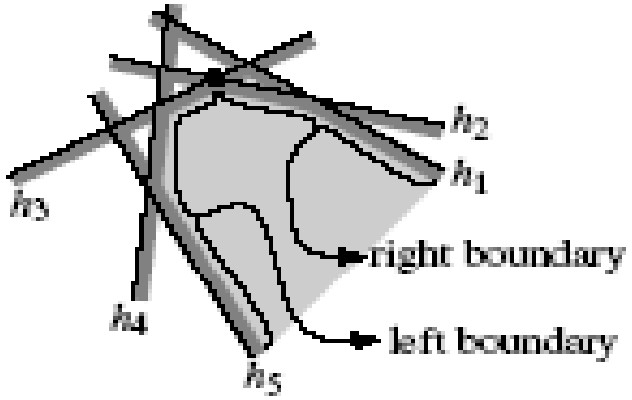
$$T(n)=o(n\log^2 n)$$

- به دنبال الگوريتم سريعترمي باشيم.

- توجه: در الگوريتمي كه ارائه خواهد شد، نواحي اي كه در INTERSECTCONVEXREGIONS از آنها اشتراك گرفته مي‌شود دو بعدي درنظر گرفته مي شوند، اگر يكي يا هردوي آنها خط يا نقطه بودند، كار آسانتر مي گردد و به عنوان تمرين رها مي‌شود.

۲-۴ تقاطع نيم صفحه ها

- مشخصات جزئي تر براي معرفي يك چندضلعي محدب C: مرز چپ و راست C به طور جداگانه، كه شامل ليست مرتب شده نيم صفحه ها به ترتيبي كه وقتي مرز(چپ يا راست) از بالا به پايين طي مي‌شود، اتفاق مي افتد. اين دو مرز را با $L_{left}(C)$ و$L_{right}(C)$ نمايش داده مي‌شود.



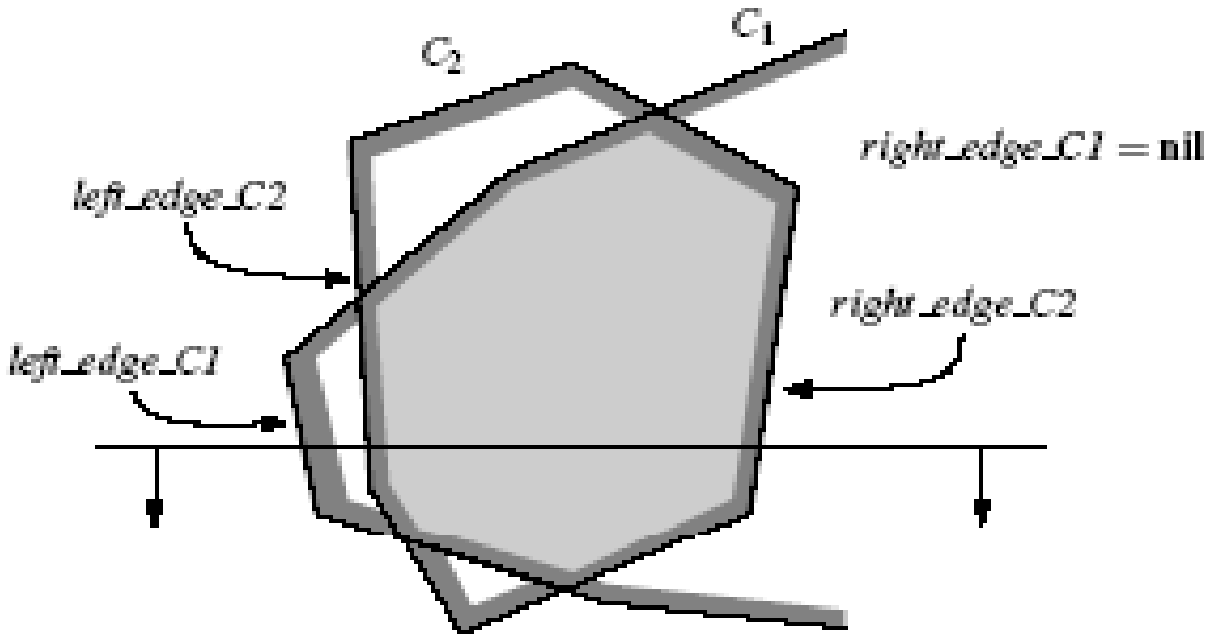
$$L_{\text{left}}(C) = h_3, h_4, h_5$$
$$L_{\text{right}}(C) = h_2, h_1$$

- رئوس در اينجا مرتب نشده اند كه با يافتن اشتراك خطوط متوالي به دست مي آيند.

۲-۴ تقاطع نيم صفحه ها

- براي سادگي توصيف الگوريتم، فرض مي‌شود كه يال افقي وجود ندارد.(براي اينكه يالهاي افقي نيز موجود باشند، مي توان گفت اگر يال مذكور از بالا C را محدود كند متعلق به مرز چپ باشد و يا برعكس.)
- الگوريتم جديد، يك الگوريتم plane sweep است، يك sweep line را روي صفحه به سمت پايين حركت داده و يالهايي از C_1 وC_2 كه با آن تقاطع دارند نگهداري مي شوند.
- با توجه به محدب بودن C_1 وC_2 حداكثر ۴ يال متقاطع وجود دارد، پس به جاي استفاده از ساختمان داده پيچيده از ۴ اشاره گر $left\_edge\_C_1$ و $left\_edge\_C_2$ و $right\_edge\_C_1$ و $right\_edge\_C_2$ استفاده مي‌شود.(درصورت عدم تقاطع nill)

۴-۲ تقاطع نیم صفحه ها



شکل ۴-۳

۲-۴ تقاطع نیم صفحه ها

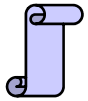
- سؤال: اشاره گرها چگونه مقداردهي اوليه مي شوند؟
- پاسخ: اگر y_1 مؤلفه بالاترين راس C_1 باشد،(اگر C_1 يال نامحدودي از بالا داشته باشد، $y_1 = \infty$) و y_2 به طور مشابه باشد آنگاه $y_{start} = \min(y_1, y_2)$. براي محاسبه تقاطع C_1 و C_2 توجه لازم تنها روي قسمتي از صفحه با مؤلفه y كمتر يا مساوي y_{start} است. درنتيجه sweep را از y_{start} شروع به حركت داده و مقدار اوليه تمامي اشاره گرها خط $y=y_{start}$ داده مي‌شود.
- سؤال: در الگوريتم sweep صفحه اي به صف eventها نيازي است؟
- پاسخ: eventها درواقع ابتدا و انتهاي يالهاي C_1 و C_2 كه با sweep برخورد داشته اند، پس event بعدي كه مشخص كننده يال بعدي است كه به آن مي رسيم، درواقع بالاترين end pointي است كه يال آن در sweep قرار دارد، درنتيجه event بعدي را در زمان ثابت با استفاده از ۴ اشاره گر موجود مي توان به دست آورد و نيازي به صف نمي باشد.(end point با yهاي مساوي را از چپ به راست درنظر گرفته و در end pointهايي كه روي هم قرار دارند، چپ ترين يال را اول درنظر مي گيريم.)

۲-۴ تقاطع نيم صفحه ها

- در هر event point يك سري يال جديد e روي مرز ايجاد مي‌شود، براي كار روي e ابتدا بررسي مي‌شود كه اولا e متعلق است به C1 يا C2 و دوما روي مرز چپ قرار دارد يا مرز راست و سپس با توجه به اين دو الگوريتم مناسب فراخواني مي‌شود.
- در اينجا الگوريتمي كه e بر روي مرز چپ C1 قرار دارد بيان مي‌شود، بقيه به طور مشابه است.
- فرض كنيد p بالاترين end point ، e است، با سه حالت ممكن e به عنوان يالي از C درنظر گرفته مي‌شود: يا يال با P به عنوان بالاترين end point، يا $e \cap left\_edge\_C_2$ به عنوان بالاترين end point، يا $e \cap right\_edge\_C_2$ به عنوان بالاترين يال، كه به صورت زير بررسي مي‌شود.

۲-۴ تقاطع نیم صفحه ها

- آيا p مابين $left\_edge\_C_2$ و $right\_edge\_C_2$ قرار دارد؟ اگر باشد، يال e را با شروع از p در C شركت مي دهيم و سپس نيم صفحه اي را كه خط مرزي آن شامل e مي‌شود را به $L_{left}(C)$ اضافه مي كنيم.



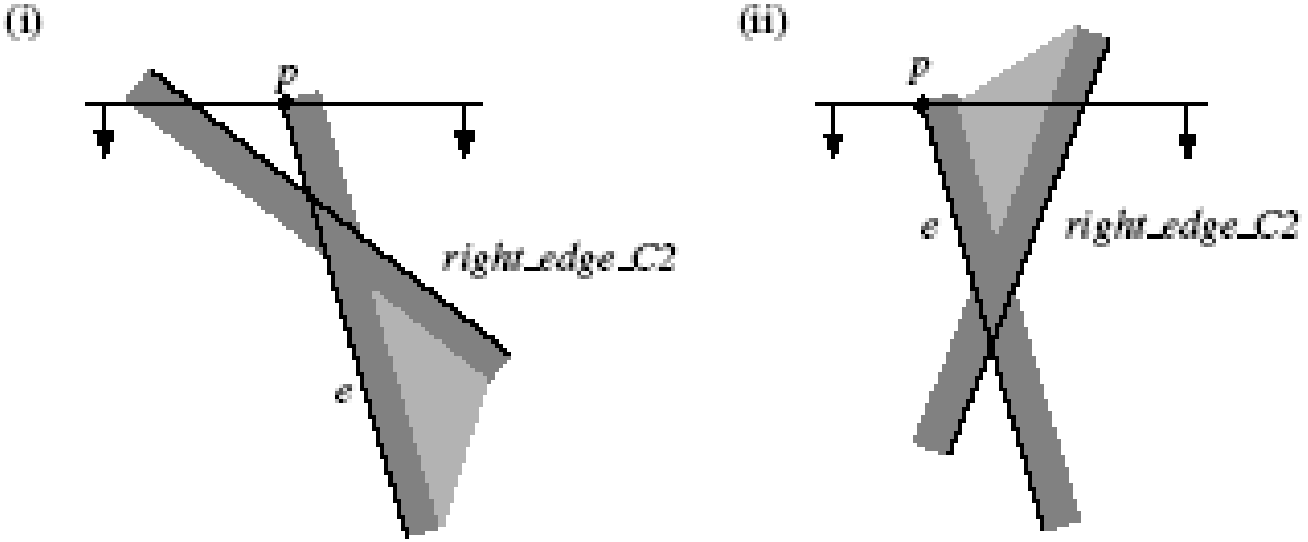
- آيا e با $right\_edge\_C_2$ تقاطع دارد؟ اگر داشت آنگاه نقطه تقاطع يك راس C است و دو حالت ممكن دارد:

۱. p سمت راست $right\_edge\_C_2$ قرار مي گيرد، آنگاه نقطه تقاطع شروع يالي است كه بايد در C قرار گيرد.

۲. P سمت چپ $right\_edge\_C_2$ قرار مي گيرد، آنگاه نقطه تقاطع پايان يالي است كه بايد در C قرار گيرد.

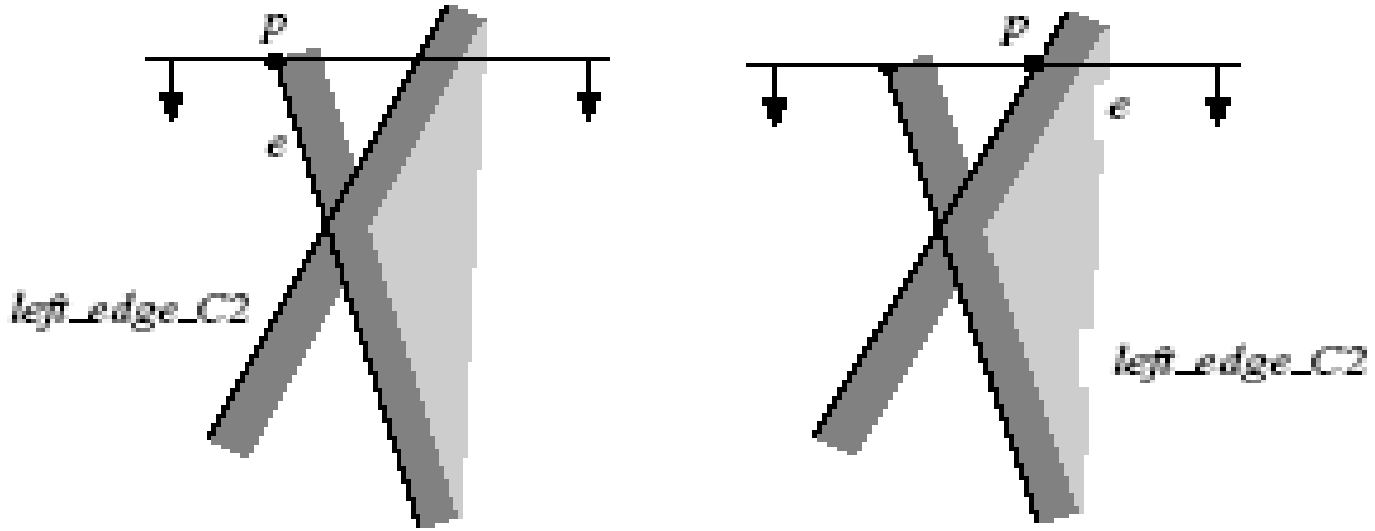
۲-۴ تقاطع نیم صفحه ها

- در حالت اول نیم صفحه اي كه e را تعريف مي كند در $L_{left}(C)$ قرار داده شده و نيم صفحه اي كه right_edge_C2 را تعريف مي كند در $L_{right}(C)$. در حالت دوم هيچ كاري انجام نمي‌شود، چون اين قبلا مورد بررسي قرار گرفته اند و در صورت نياز وارد ليستها شده اند.



۲-۴ تقاطع نیم صفحه ها

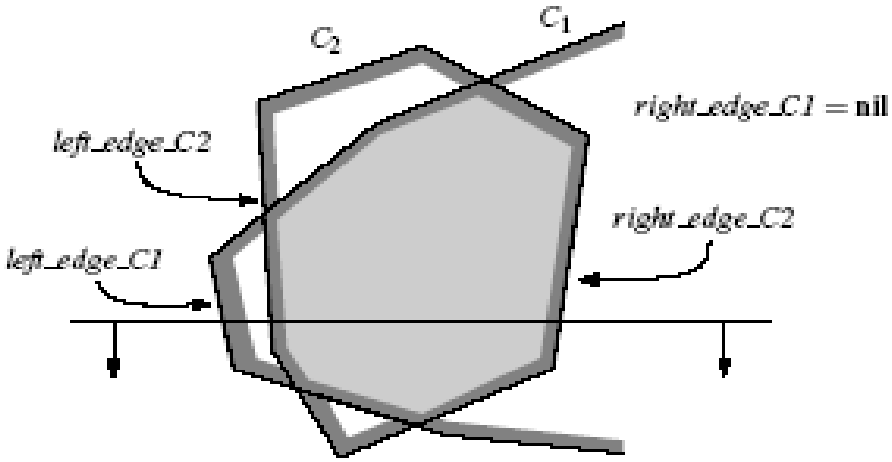
- آیا e با $left\_edge\_C_2$ تقاطع دارد؟ اگر داشت، نقطه تقاطع راسي از C است و يال C كه با نقطه تقاطع شروع مي‌شود دو حالت ممكن دارد، يا قسمتي از e است يا قسمتي از $left\_edge\_C_2$ كه تصميم گيري بين اين دو در زمان ثابت صورت مي گيرد. اگر p سمت چپ $left\_edge\_C_2$ باشد آن يال قسمتي از e است وگرنه قسمتي از $left\_edge\_C_2$. پس با توجه به اين دو حالت يا e يا $left\_edge\_C_2$ به C اضافه شده و نيم صفحه متناظرش به $L_{left}(C)$.

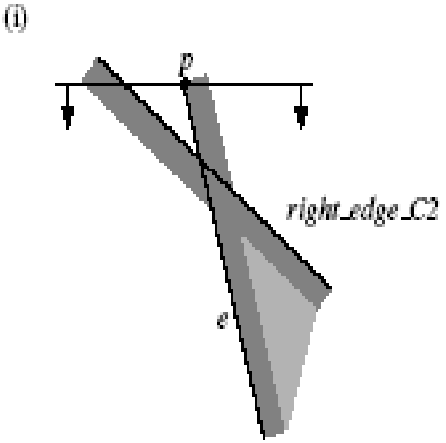


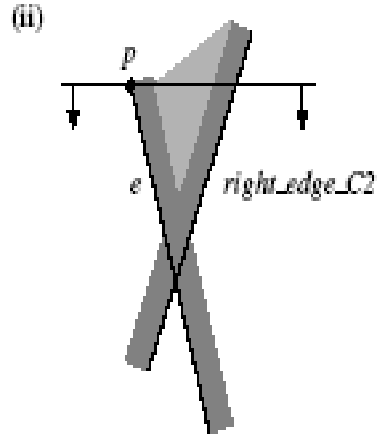
۲-۴ تقاطع نيم صفحه ها

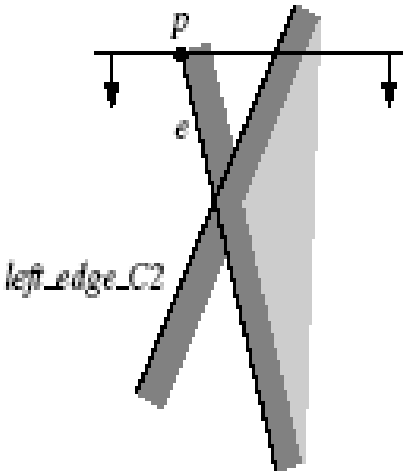
- سؤال: ممكن است هم نيم صفحه متناظر با e و هم نيم صفحه متناظر با $left\_edge\_C_2$ به $L_{left}(C)$ اضافه شود، به چه ترتيبي اين دو به ليست اضافه مي شوند؟
- پاسخ: $left\_edge\_C_2$ تنها وقتي اضافه مي‌شود كه به عنوان يك يال C با شروع از نقطه تقاطع $left\_edge\_C_2$ و e باشد. براي اضافه كردن صفحه e دو دليل ممكن است وجود داشته باشد:

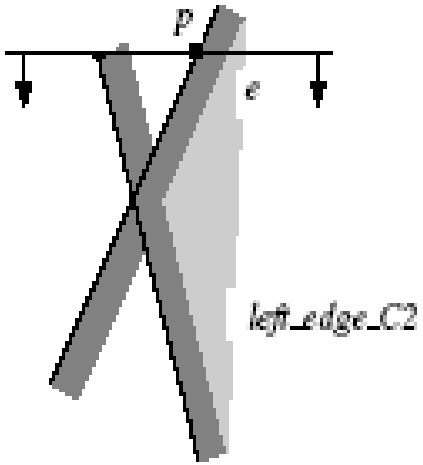
يا e به عنوان يال C با شروع در بالاترين end point آن باشد(يعني همان حالت اول) يا e به عنوان يال C با شروع از تقاطعش با $right\_edge\_C_2$ باشد، در اين صورت e به $L_{left}(C)$ اضافه مي‌شود و $right\_edge\_C_2$ به $L_{right}(C)$(طبق حالت دوم) و $left\_edge\_C_2$ هم به $L_{left}(C)$(طبق همين حالت سوم). پس در هردو حالت اول بايد e اضافه شود بعد $left\_edge\_C_2$.

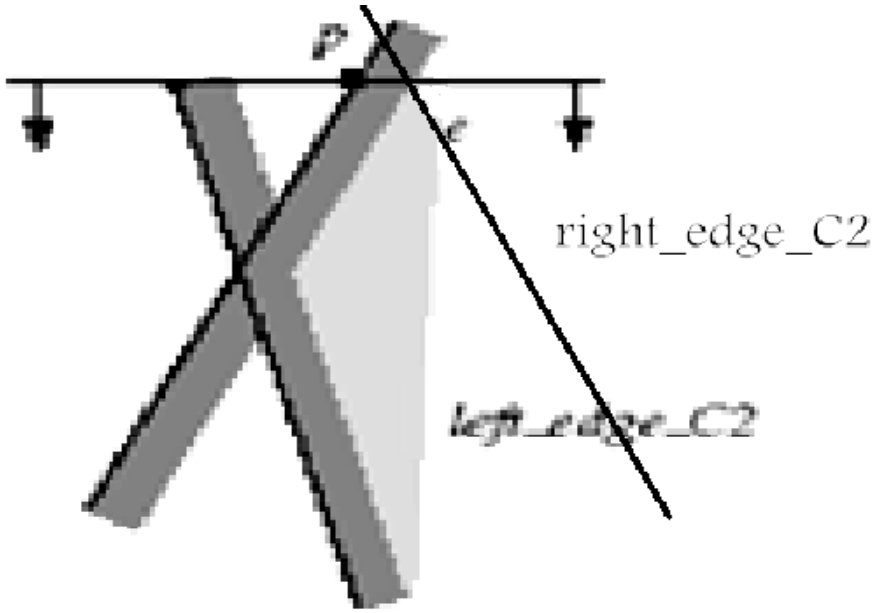

C_2
C_1
$right\_edge\_C1 = \mathbf{nil}$
$left\_edge\_C2$
$right\_edge\_C2$
$left\_edge\_C1$

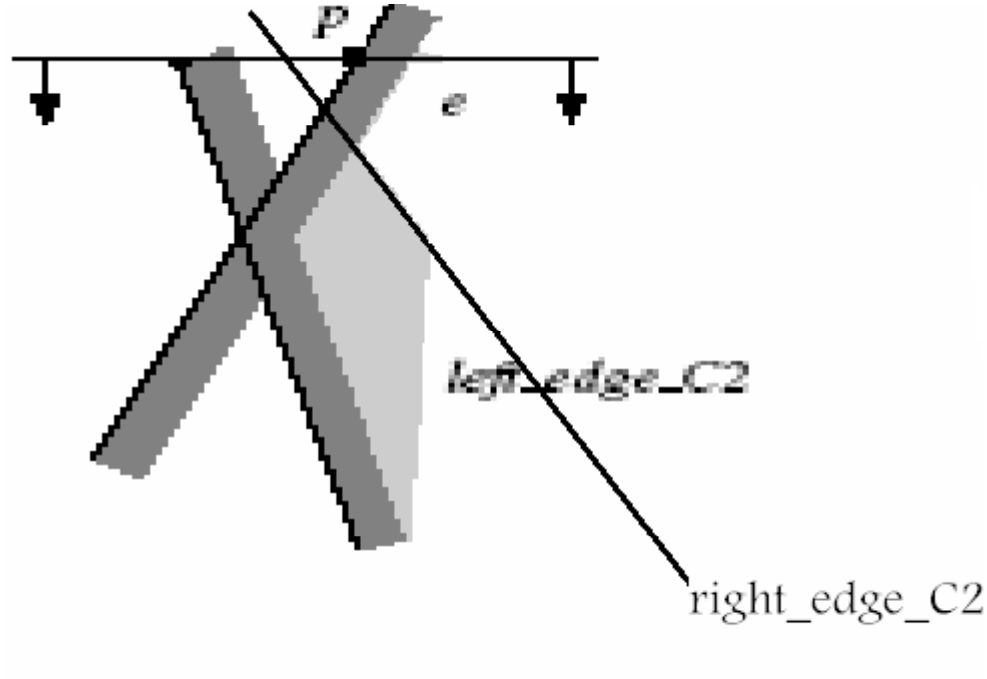

(i)
p
$right\_edge\_C2$
e


(ii)
p
e
$right\_edge\_C2$


p
e
$left\_edge\_C2$

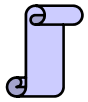

p
e
$left\_edge\_C2$


p
e
right_edge_C2
left_edge_C2


p
e
left_edge_C2
right_edge_C2

۲-۴ تقاطع نیم صفحه ها

- نتیجه: برای رسیدن به یک یال C زمان ثابت مورد نیاز است.
- قضیه ۳-۴: تقاطع دو ناحیه محدب می تواند در زمان O(n) محاسبه شود.(چون حداکثر n نیم صفحه و درنتیجه n یال داریم.)
- صحت الگوريتم منوط به اضافه شدن تمام يالهاي C با ترتيب صحيح مي باشد: يک يال C و نقطه p به عنوان upper end point آن را فرض کنید. آنگاه p یا upper end point ، یک یال از C_1 یا C_2 است و یا نقطه تقاطع e از C_1و e′ از C_2. در مورد اول وقتی که به p می رسیم یال C هم یافت می شود و در مورد دوم وقتی که به upper end point پایین تری اش رسیدیم، یال اضافه می شود.(با توجه به حالتهای اول و دوم). بنابراین همه نیم صفحه ها اضافه می شوند. اثبات ترتیب مناسب اضافه شدن هم ساده می باشد.



۴-۲ تقاطع نیم صفحه ها

- با توجه به قضیه ۴-۳ برای الگوریتم INTERSECTHALFPLANE داریم:

$$T(n)=\begin{cases} o(1) & n=1 \\ o(n)+2T(n/2) & n>1 \end{cases}$$

$$T(n)=o(n\log n)$$

- نتیجه ۴-۴: تقاطع مشترک یک مجموعه با n نیم صفحه می تواند در زمان O(nlogn) محاسبه و با O(n) ذخیره شود.

۳-۴ برنامه های خطی افزایشی

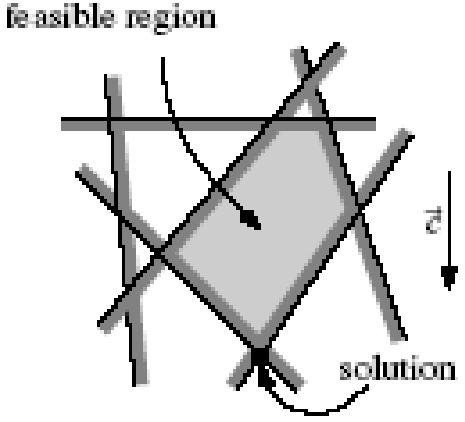
- در قسمت قبل تمام نقاط مشترک نیم صفحه ها یافت شد درحالیکه در بسیاری از مسائل از جمله مسئله ریخته گری یافتن یک جواب کافیست، پس احتمالا می توان به دنبال الگوریتم سریعتری بود.
- يافتن يك جواب براي يك مجموعه از معادلات، ارتباط نزديكي با مسئله شناخته شده درoperations research كه linear optimization و يا linear programming ناميده مي‌شود، دارد. با اين تفاوت كه در برنامه خطي جواب خاصي (جوابي كه تابع خطي را ماكسيمم مي كند.) از مجموعه معادلات مورد نياز است، يعني :

$$\begin{aligned} &\text{Maximize} && c_1x_1+ c_2x_2+\ldots+ c_dx_d \\ &\text{Subject to} && a_{1,1}x_1+\ldots+ a_{1,d}x_d \le b_1 \\ & && a_{2,1}x_1+\ldots+ a_{2,d}x_d \le b_2 \\ & && \vdots \\ & && a_{n,1}x_1+\ldots+ a_{n,d}x_d \le b_n \end{aligned}$$

که $a_{i,j}$ و c_i و b_i ثوابتي هستند كه ورودي مسئله را شكل مي دهند.

۳-۴ برنامه های خطی افزایشي

- تابعي كه ماكسيمم مي‌شود تابع objective، مجموعه معادلات به همراه تابع obj برنامه خطي و تعداد متغيرهاي Lp بعد آن ناميده مي‌شود.
- از آنجا كه معادلات خطي به عنوان نيم فضاهايي در R^d مي باشند، تقاطع اين نيم فضاها كه در همه معادلات صدق مي كند به عنوان ناحيه امكان(feasible) برنامه خطي و نقاط موجود در اين ناحيه نقاط امكان و نقاط خارج از اين ناحيه نقاط ناممكن(infiseable) ناميده مي شوند. اگر نقاط امكان تهي باشد(شكل ۲-۴ قسمت (v)) Lp ناممكن ناميده مي‌شود.
- تابع obj با يك جهت در فضاي بيان شده و ماكسيمم كردن يعني يافتن نقطه $(x_1,x_2,\ldots,x_d)$ كه در جهت بردار $\vec{c}$ ماكسيمم باشد.

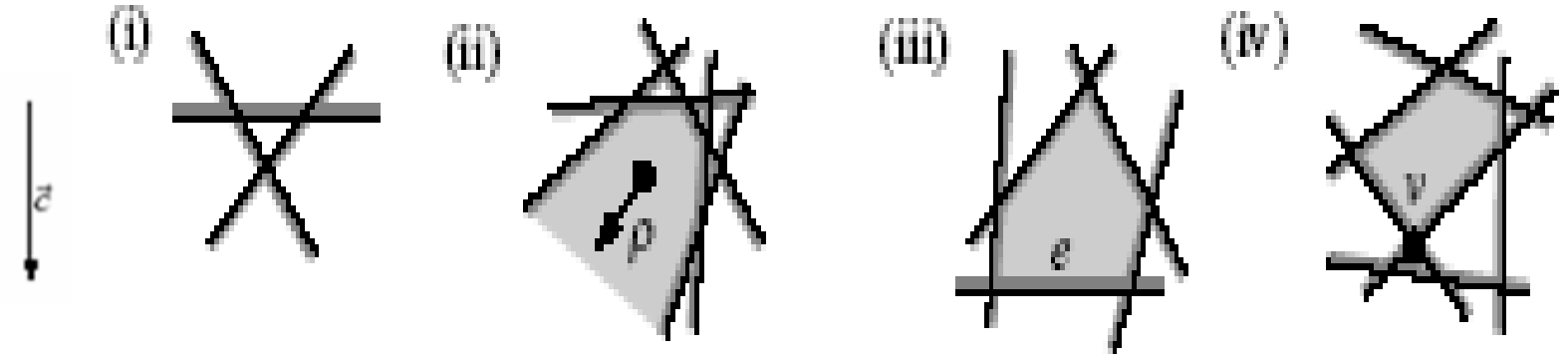


در نتيجه حل Lp يعني يافتن يك نقطه در ناحيه امكان كه در جهت $\vec{c}$ ماكسيمم باشد.

- $f_{\vec{c}}$: تابع obj تعريف شده توسط بردار $\vec{c}$.

۳-۴ برنامه های خطی افزایشي

- مسئله: يافتن يك نقطه $p \in R^2$ كه $p \in \cap H$, و $f_{\vec{c}}$ ماكسيمم شود.($\vec{c} = (c_x, c_y)$ و $f_{\vec{c}} = c_x p_x + c_y p_y$)
- برنامه خطي را با (H, $\vec{c}$) نمايش مي دهيم، و از C براي نمايش ناحيه امكان استفاده مي‌شود. كه چهار جواب ممكن دارد.



۳-۴ برنامه های خطی افزایشي

۱. LP ناممکن است ⇐هیچ جوابي براي مجموعه معادلات وجود ندارد.

۲. ناحیه امکان بیکران است ⇐ یك پرتوي ρ وجود دارد که به طور کامل در ناحیه C قرار مي گیرد و تابع $f_{\vec{c}}$ مقادیر بزرگ دلخواه در طول ρ مي گیرد. در اینجا حل مسئله توصیف اینچنین پرتویي است.

۳. ناحیه C یك یال e دارد که نرمال خارجي آن در جهت $\vec{c}$ است ⇐ هر نقطه روي e نقطه ایست که $f_{\vec{c}}(p)$ را ماکسیمم مي کند.

۴. هیچ کدام از سه حالت فوق اتفاق نیفتد ⇐ جواب یگانه v وجود دارد، که در جهت $\vec{c}$ ماکسیمم است.

۳-۴ برنامه های خطی افزایشي

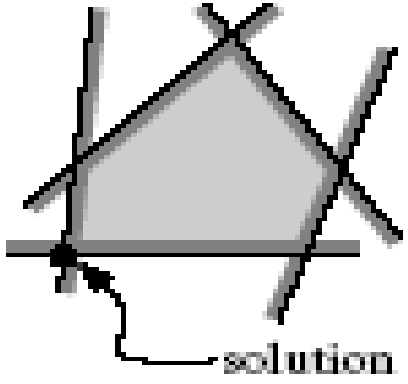
- براي حل مسئله اولا: فرض مي‌شود كه معادلات يكي يكي اضافه شده و جواب هر برنامه خطي مياني، خوش تعريف و يگانه است كه اين نتيجه مي دهد، جواب نهايي نيز يگانه است.(مطابق شكل iv)
- بدين منظور دو معادله به LP اضافه مي‌شود، تا برنامه خطي كراندار شود، مثلا اگر $c_x>0$ و$c_y>0$ آنگاه بازاي يك $M\in R$ بزرگ، معادلات $p_x \le M$ و $p_y \le M$ اضافه مي شوند.(M بايد بزرگ انتخاب شود، تا اثري روي جواب بهينه نداشته باشد.)
- درنتيجه دو معادله زير به مجموعه معادلات اضافه مي‌شود:

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

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

۳-۴ برنامه های خطی افزایشي

- نکته: m_1 وm_2 تابع $\vec{c}$ هستند ومستقل از نیم صفحه هاي H و ناحیه امکان $C_0=m_1\cap m_2$ یك زاویه قائمه است.
- دوما: براي وجود جواب یگانه در مورد(iii) هم کوچکترین نقطه از لحاظ قاموسي درنظر گرفته مي‌شود، که معادل است با اینکه $\vec{c}$ اندکي چرخانده شود تا با هیچ کدام از نرمال صفحات هم جهت نباشد.
- نتیجه: با اعمال این دو قرارداد هر برنامه خطي جواب یگانه دارد که یك راس ناحیه ممکن است و راس بهینه نامیده مي‌شود.



۴-۳ برنامه های خطی افزایشي

- فرض: $(H, \vec{c})$ برنامه خطي مورد نظر، $h_1, h_2, \ldots, h_n$ شماره گذاري صفحات H، H_i i معادله اول به همراه معادلات m_1و m_2 و C_i ناحیه ممکن آنها $\Leftarrow$ با انتخاب C_0 هر ناحیه ممکن C_i يك راس بهینه یگانه دارد v_i

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

$C_i := m_1 \cap m_2 \cap h_1 \cap \ldots \cap h_i$

$\forall j \geq i \quad C_j = \varnothing \Leftarrow C_i = \varnothing$ و اگر $C_0 \supseteq C_1 \supseteq \ldots \supseteq C_n = C$

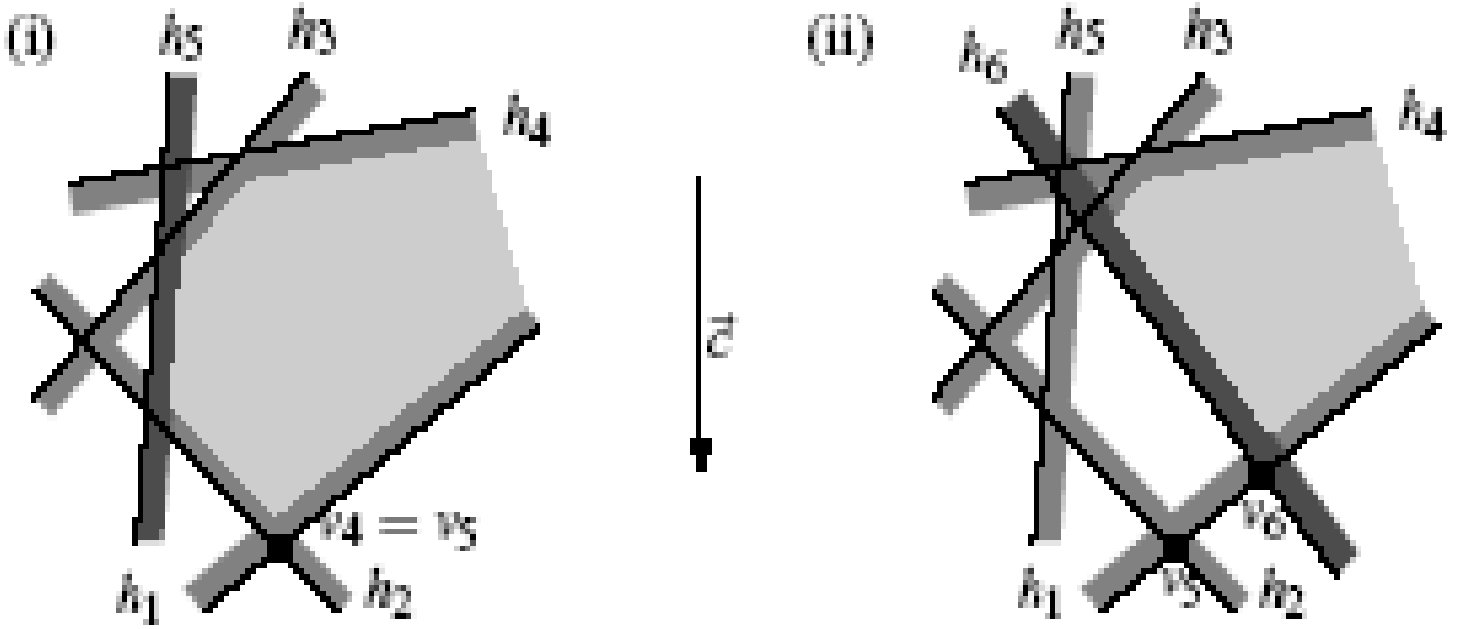
- لم ۴-۵: (بررسي چگونگي تغییر راس بهینه پس از اضافه شدن h_i) اگر $1 \leq i \leq n$ آنگاه داریم:

i) if $v_{i-1} \in h_i \Rightarrow v_i = v_{i-1}$

ii) if $v_{i-1} \notin h_i \Rightarrow C_i = \varnothing$ or $v_i \in l_i$ l_i is the line bounding h_i

۳-۴ برنامه های خطی افزایشی

✓ مثال



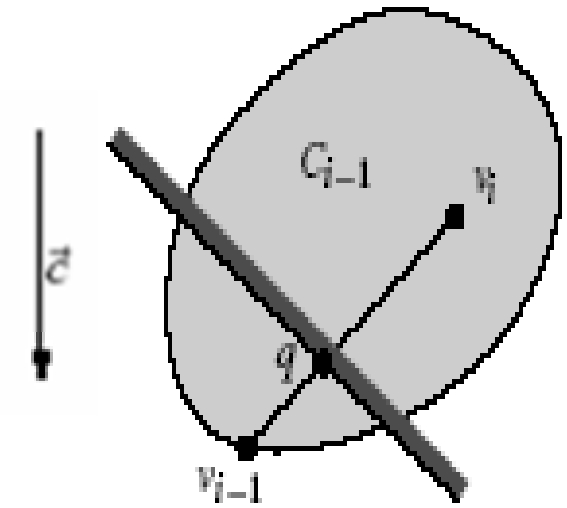
۳-۴ برنامه های خطی افزایشي

کار مضاعف

همت مضاعف

اثبات:

- i) اگر $v_{i-1} \in h_i$ آنگاه با توجه به $C_i = C_{i-1} \cap h_i$ و $v_{i-1} \in C_{i-1}$ در نتیجه $v_{i-1} \in C_i$ ، از آنجا که $C_i \subseteq C_{i-1}$ پس نقطه بهینه C_i نمي تواند بهتر از نقطه بهینه C_{i-1} باشد، در نتیجه v_{i-1}همان راس بهینه C_i مي باشد.
- ii) $v_{i-1} \notin h_i$، فرض خلف: $C_i \neq \varnothing$ و v_i روي l_i قرار نگیرد. آنگاه خط $v_{i-1}v_i$ را داریم. چون $v_i \in C_i \subseteq C_{i-1}$ در نتیجه $v_i \in C_{i-1}$ و با توجه به محدب بودن C_{i-1} و اینکه $v_{i-1} \in C_{i-1}$ و خط $v_{i-1}v_i$ کاملا داخل C_{i-1} است. حال چون v_{i-1} نقطه بهینه C_{i-1} است و تابع obj ، $\vec{f}_c$ خطي مي باشد،



پس در طول خط $v_{i-1}v_i$ از v_i به v_{i-1} افزایش مي یابد.

حال نقطه تقاطع خط مذکور با l_i را q درنظر مي گیریم.

q) وجود دارد چون: $v_{i-1} \notin h_i$ و $v_i \in C_i$ $\Leftarrow$ qعضو C_i هم است.) $\Leftarrow$ $\vec{f}_c(q) > \vec{f}_c(v_i)$

و این تناقض است با بهینه بودن v_i در ناحیه C_i.

۴-۳ برنامه های خطی افزایشی

- لم ۴-۵ چگونگي يافتن راس جديد را بيان نمي كند.
- مسئله: يافتن نقطه p روي l_i كه $f_{\vec{c}}(p)$ را تحت محدوديت $p \in h$ براي $h \in H_{i-1}$ ماكسيمم مي كند.(درحالت دوم لم)
- براي سادگي فرض مي‌شود l_iعمودي نيست، آنگاه با مؤلفه x پارامتري مي‌شود. پس تابع زير تعريف مي‌شود:

$$\overline{f}_{\vec{c}} : R \to R$$

$$f_{\vec{c}}(p) = \overline{f}_{\vec{c}}(p_x) \quad for \quad p \in l_i$$

- براي هر نيم صفحه h، $б(h,l_i)$ مؤلفه x نقطه تقاطع l_i و خط مرزي h است.(عدم تقاطع: يا هر نقطه روي l_i در معادله h صدق مي كند، كه معادله h ناديده گرفته مي‌شود، و يا هيچ نقطه اي از l_i در h صدق نمي كند كه LP ناممكن گزارش مي‌شود.)

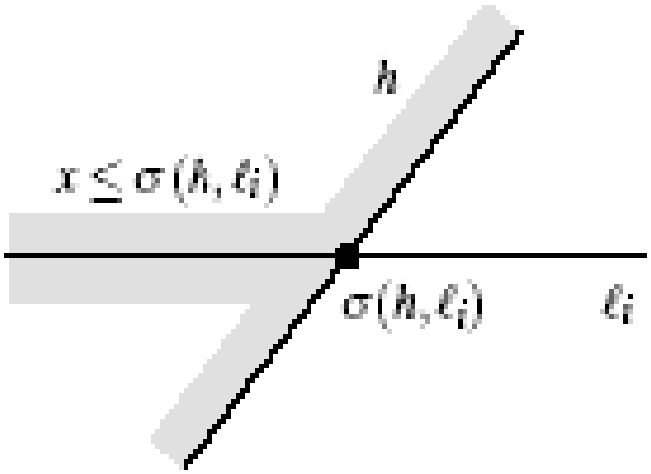
۳-۴ برنامه های خطی افزایشي

✓ مسئله: با توجه به اينكه $l_i \cap h$ به چپ كراندار مي‌شود يا به راست مسئله تبديل مي‌شود به:

Maximize $\overline{f}_{\overrightarrow{c}}(x)$

subject to $x \geq б(h,l_i)$, $h \in H_{i-1}$ and $l_i \cap h$ is bounded to the left

$x \leq б(h,l_i)$, $h \in H_{i-1}$ and $l_i \cap h$ is bounded to the right



كه يك برنامه خطي يك بعدي مي باشد.

۳-۴ برنامه های خطی افزایشي

- فرض کنید:

$x_{left}=max_{\ h\in Hi-1}\{б(h,l_i)\colon l_i\cap h \text{ is bounded to the left}\}$

and

$x_{right}=min_{\ h\in Hi-1}\{б(h,l_i)\colon l_i\cap h \text{ is bounded to the right}\}$

فاصله $[x_{left}:x_{right}]$ ناحيه ممكن LP يك بعدي است، پس اگر $x_{left} > x_{right}$ LP, ناممكن و درغير اينصورت نقطه بهينه نقطه اي روي خط l_i در نقطه x_{left} يا x_{right}مي باشد.

- لم ۶-۴: برنامه خطي يك بعدي در زمان خطي حل مي‌شود.(يافتن ماكسيمم يا مينمم i نيم صفحه) بنابراين آنگاه محاسبه راس بهينه جديد v_i (يعني حالت دوم لم ۵-۴) و يا تصميم گيري ناممكن بودن برنامه خطي در زمان O(i) انجام مي گيرد.

۳-۴ برنامه های خطی افزایشی

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 $h_1, \ldots, h_n$ 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

۳-۴ برنامه های خطی افزایشي

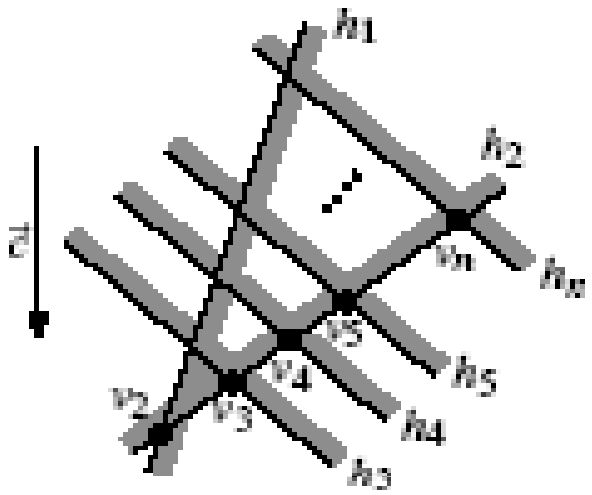
- لم ۴-۷: الگوريتم فوق حل يك برنامه خطي كراندار با n معادله و دو متغير را در زمان $O(n^2)$ و ذخيره خطي محاسبه مي كند.
- اثبات صحت الگوريتم: با توجه به لم ۴-۵ هرگاه نيم صفحه جديد h_i اضافه مي‌شود، نقطه v_i همچنان نقطه بهينه C_i است و اگربرنامه خطي يك بعدي روي l_i ناممكن باشد، آنگاه C_i تهي است و درنتيجه $C=C_n \subseteq C_i$ هم تهي است، پس برنامه خطي ناممكن است.
- اثبات زمان اجرا: نيم صفحه ها يكي يكي در n مرحله اضافه مي شوند، بيشترين زمان اجرا در خط ۶ام سپري مي‌شود، كه در مرحله iام برابر O(i) مي باشد، پس زمان كل برابر است با:

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

- گرچه الگوريتم ساده و خوب است ولي نسبت به الگوريتم قبل كه تمام نقاط را مي يافت كندتر است، آيا اين تحليل صحيح مي باشد؟؟

۳-۴ برنامه های خطی افزایشي

- هزينه هر مرحله i با O(i) كراندار مي‌شود، اگر $v_{i-1} \notin h_i$. ولي اگر $v_{i-1} \in h_i$ مرحله i در زمان ثابت انجام مي گردد.
- راس بهينه حداكثر مي تواند n بار تغيير كند ولي اگر ترتيب اضافه شدن صفحات مناسب باشد، اين مقدار كاهش مي يابد.



- شكل روبرو نشان مي دهد كه زمان اجرا در بدترين حالت $O(n^2)$ مي باشد.

۴-۴ برنامه خطي تصادفي

- سؤال: پس مي توان نيم صفحه ها را به ترتيبي اضافه كرد كه اصلا v_i تغييري نكند تا زمان اجرا خطي شود. اينكه براي هر مجموعه از نيم صفحه ها يك ترتيب مناسب وجود دارد صحيح است ولي آيا يافتن اين ترتيب ممكن و يا مقرون به صرفه است؟
- پاسخ: چون اين ترتيب بايد قبل از شروع الگوريتم كه هنوز هيچ از تقاطع نيم صفحه ها در دست نيست موجود باشد يافتن ترتيب مناسب ساده نيست. كه در اينجا يك ترتيب تصادفي از H را برمي داريم.
- درالگوريتم تصادفي از تابع RANDOM(k) استفاده مي‌شود كه در زمان ثابت يك عدد تصادفي بين ۱ تا k را مي دهد.
- زمان اجرا وابسته به انتخاب تصادفي ساخته شده در الگوريتم ميباشد.

۴-۴ برنامه خطي تصادفي

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 $h_1, \ldots, h_n$ of the half-planes by calling RANDOMPERMUTATION($H[1 \cdots n]$).
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

۴-۴ برنامه خطي تصادفي

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 ← n downto 2
2.      do rndindex ← RANDOM(k)
3.         Exchange A[k] and A[rndindex].
```

- تعداد حالات مختلف n! است كه هركدام زمان خاص اجراي خود را دارد. حال چون هركدام از اين زمان هاي اجرا احتمال يكساني دارند، زمان اجراي مورد انتظار الگوريتم كه متوسط تمام n! حالت مختلف زمان اجراست، مورد بررسي قرار مي گيرد.
- يادآوري: اميد رياضي و يا ميانگين متغير تصادفي

$$E(x)=\begin{cases}\sum xf(x) & gosaste\\ \int xf(x) & peyvaste\end{cases}$$

f(x): تابع چگالي احتمال

۴-۴ برنامه خطي تصادفي

- لم ۸-۴: مسئله برنامه نويسي خطي دوبعدي با n معادله در زمان مورد انتظار O(n) و ذخيره سازي خطي حل مي‌شود.
- اثبات: زمان اجراي RANDOMPERMUTATION برابراست با O(n). براي اضافه كردن صفحات، متغير تصادفي X_i را بدين صورت تعريف مي كنيم:

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

- پس زمان سپري شده براي n صفحه در خط ۶ برابر است با:

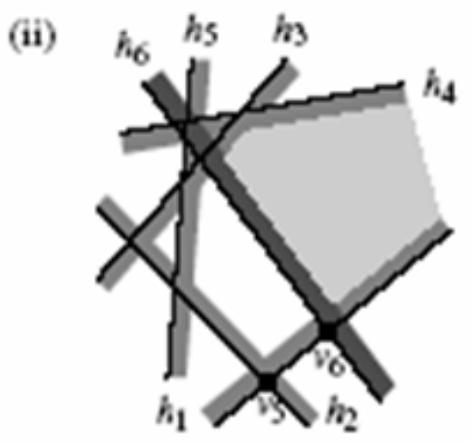
$$\sum_{i=1}^{n} O(i) X_i$$

- با توجه بودن به خطي بودن اميد رياضي:

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

۴-۴ برنامه خطي تصادفي

- سؤال: $E(X_i)$ چيست؟
- پاسخ: احتمال اينكه $v_{i-1} \not\in h_i$، يعني احتمال اينكه نقطه بهينه در اين مرحله تغيير كند. براي تحليل اين احتمال از روند BackwardsAnalysis استفاده مي‌شود، يعني فرض مي‌شود كه الگوريتم انجام گرفته و راس بهينه v_n بدست آمده، حال صفحه h_n را حذف مي كنيم، با چه احتمالي نقطه بهينه تغيير مي كند؟



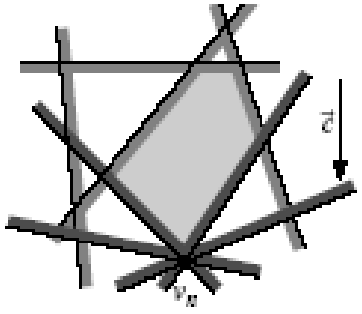
- از آنجا كه v_n راس C_n مي باشد، حداقل توسط دو تا از نيم صفحه ها تعريف مي شوند و وقتي v_n تغيير مي كند كه راسي از C_{n-1} نباشد، يعني h_n يكي از نيم صفحه هايي است كه v_n را تعريف مي كند. و از آنجا كه نيم صفحه ها به طور تصادفي اضافه شده اند، احتمال اينكه h_n يكي از نيم صفحه هاي تعريف كننده v_n باشد حداكثر $2/n$ مي باشد.

کار مضاعف

همت مضاعف

۴-۴ برنامه خطي تصادفي

- سؤال: چرا حداکثر2/n ؟
- پاسخ: اولا اگر v_n تعريف کننده بيش از دو يال باشد، آنگاه حذف h_n ممکن است باعث تغيير v_n نشود، چون ممکن است v_n بعد از حذف h_n باز هم راس باشد. دوما علاوه بر n معادله،m_1و m_2 هم مي توانند در ساختن v_n دخالت داشته باشند،که اين هم احتمال را از 2/n کمتر مي کند. پس



$$E(x_n) \leq 2/n$$

- ✓ ادامه پاسخ: حال اگر همين روند Backwards براي i معادله اول H درنظر گرفته شود، باز هم داريم:

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

- ✓ در نتيجه زمان کل مورد انتظار براي حل تمام برنامه هاي خطي دو بعدي برابر اسْت با:

$$\sum(O(i).2/i) = O(n)$$

۴-۵ برنامه خطي غير كراندار

- در قسمت قبل برنامه خطي نامحدود را با اعمال دو محدوديت اضافي كراندار كرديم، اما اين هميشه ممكن نيست و حتي گاهي برنامه خطي محدود مي‌شود ولي كران يا همان مرز ناحيه ممكن بسيار بزرگ مي‌شود به گونه اي كه قابل شناخت نيست.
- مسئله : چگونگي تشخيص غيركراندار‌ي برنامه خطي $(H, \vec{c})$
- لم ۴-۹: يك برنامه خطي $(H, \vec{c})$ غيركراندار است اگر و فقط اگر يك بردار $\vec{d}$ وجود داشته باشد بطوريكه $\vec{d}.\vec{c}>0$ و براي تمام $h \in H$ $\vec{d}.\vec{\eta}(h) \geq 0$ و برنامه خطي $(H', \vec{c})$ ممكن مي باشد اگر

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

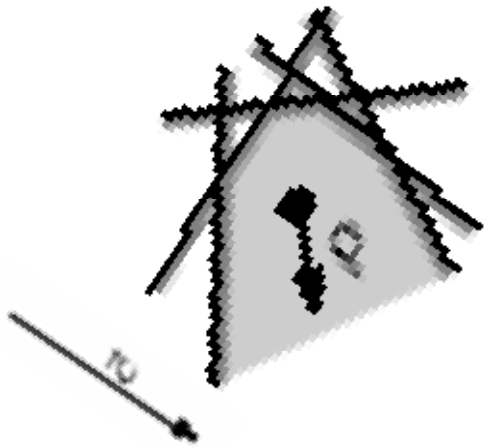
۵-۴ برنامه خطي غير كراندار

اثبات:

- Only if: قبلا گفته شد كه بيكران بودن C به معني وجود پرتوي ρ به گونه ايست كه كامل در ناحيه ممكن قرار بگيرد و تابع obj در طول ρ مقادير بزرگ دلخواه بگيرد.
- p: نقطه شروع ρ، $\vec{d}$: بردار جهت آن $\Leftarrow$ $\rho=\{p+\lambda\vec{d} \ : \lambda>0\}$
- تابع obj در طول ρ مقادير بزرگ مي گيرد اگرو فقط اگر $\vec{d}\,.\,\vec{c}>0$.

از طرف ديگر اگر $\vec{\eta}(h)$ بردار نرمال نيم صفحه h به سمت ناحيه ممكن باشد، آنگاه داريم:

$$\vec{d}\,.\vec{\eta}(h)\geq 0$$



۵-۴ برنامه خطي غير كراندار

- if : برنامه خطي (H,) و بردار $\vec{c}$ با شرايط داده شده در لم مفروض است.چون (H',) ممكن است پس نقطه $p_0 \in \cap_{h \in H'} h$ وجود دارد. فرض كنيد پرتوی $\{p_0 + \lambda : \lambda > 0\}$ ، از آنجا كه برای $h \in H'$، پرتوی $p_0 + \lambda \vec{d}$ به طور كامل در هر كدام از صفحات H' قرار می گيرد. بنابراين از آنجا كه تابع obj در طول $\vec{d}$ مقادير بزرگ دلخواه می گيرد.

برای هرنيم صفحه $h \in H \backslash H'$ ، داريم $\vec{d}.\vec{n}(h) \geq 0$ كه نتيجه می دهد پارامتر λ_hی وجود دارد، كه برای $\lambda \geq \lambda_h$ تمام $p_0 + \lambda \vec{d} \in h$. با فرض $\lambda_h := \max_{h \in H \backslash H'} \lambda_h$ و $P := P_0 + \lambda' \vec{d}$ پرتوی به p_0 طور كامل در هر كدام از صفحات H ضو قرار می گيرد. درنتيجه $(H, \vec{c})$ بيكران است.

۴-۵ برنامه خطي غير كراندار

- براي تست بيكران بودن برنامه خطي سيستم مولفه را مي چرخانيم، پس داريم:

$$\vec{c} = (0,1)$$

$$\text{since} \quad \vec{d}.\vec{c} > 0 \qquad \vec{d} = (d_x,1)$$

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

- پس براي بيكران بودن برنامه خطي بايد n معادله خطي فوق داراي جواب باشد، كه معادل است با حل برنامه خطي يك بعدي $\hat{H}$.
- نكته: در برنامه خطي نياز به تابع obj است كه در اينجا چون فقط روي وجود يا عدم وجود جواب بحث مي‌شود، تابع obj ناديده گرفته مي‌شود.

۴-۵ برنامه خطي غير كراندار

حالت اول: d_x^* جوابي از برنامه خطي $\hat{H}$ است. حال براي $h \in H' \subseteq H$ داريم:

$$d_x^* \eta_x + \eta_y = 0$$

پس نرمال خارجي همه صفحات داخل H' با بردار $\vec{d} = (d_x^*, 1)$ عمود هستند.

يعني خودشان با هم موازي هستند و با محور xها برخورد دارند. كه به برنامه خطي يك بعدي مشابه قبل تبديل مي‌شود، كه در زمان خطي قابل حل است حال اگر اين برنامه خطي يک بعدی که $\hat{H}'$ ناميده مي‌شود ممكن باشد، ← H‘ هم ممكن است ← $\hat{H}$ بيكران است (نوعي از ممكن بودن) ← برنامه خطي حالت بيكران دارد. پس مي‌توان براي آن در زمان O(n) يك پرتوي ρ يافت.

اگر $\hat{H}$‘ ناممكن باشد ← H‘ ناممكن است ← H ناممكن است. $(H' \subseteq H)$

۴-۵ برنامه خطي غير كراندار

- حالت دوم: $\hat{H}$ جواب ممكن نداشته باشد ← طبق لم برنامه خطي كراندار است.
- سؤال: آيا اطلاعات بيشتري مي‌توان بدست آورد؟
- پاسخ: با توجه دوباره به برنامه خطي يك بعدي، $\hat{H}$ ناممكن است ↔ $\hat{h}_1$ صفحه با بيشترين مرز محدود شده به چپ و $\hat{h}_2$ با كمترين مرز محدود شده به راست باشند و آن بيشترين از كمترين بزرگتر باشد، آنگاه تقاطع $\hat{h}_2$و $\hat{h}_1$ ناممكن است ← برنامه خطي سازنده h_2و h_1 كراندار است.
- كراندار بودن $(\{h_1, h_2\},c)$ به معني كرانداري كل (H,c) مي‌باشد چون $\hat{h}_2$و $\hat{h}_1$ به جاي m_2 وm_1 عمل مي كنند.

۴-۵ برنامه خطي غير كراندار

- توجه: با اين روش كرانداري برنامه خطي $(\{h_1, h_2\},c)$ ضمانت مي‌شود ولي نمي‌توان گفت كه جواب يگانه بهينه دارد. يعني وقتي كه ناممكن بودن برنامه خطي يك بعدي به اي دليل باشد كه: $\eta(h_1)=(0,-1)$

آنگاه بقيه نيم صفحات را بررسي كرده، اگر h2اي يافت شد كه $\eta_x(h_2)>0$ آنگاه h_1 و h_2 ضمانت وجود جواب يگانه بهينه را مي‌كنند. درغير اينصورت يا برنامه خطي ناممكن است ويا جواب يگانه بهينه قاموسي ندارد. كه آنرا با برنامه خطي يك بعدي كه با صفحات h با خاصيت $\eta_x(h) = 0$ ساخته مي‌شود، را حل كرده و اگر ممكن بود پرتوي ρ را در جهت (-1,0) را برگردانده كه تمام نقاط روي ρ جواب بهينه ممكن است.

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 $\vec{d} \cdot \vec{\eta}(h) \geqslant 0$ 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 $h_3, h_4, \ldots, h_n$ 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

۴-۵ برنامه خطي غير كراندار

- قضيه ۴-۱۰: يك مسئله برنامه خطي ۲بعدي با n مسئله معادله در زمان مورد انتظار تصادفي O(n) و بدترين حالت ذخيره سازي خطي قابل حل است.