

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

دانشگاه یزد

دانشکده ریاضی

گروه علوم کامپیوتر

پایان نامه

جهت دریافت درجه کارشناسی ارشد

علوم کامپیوتر

پوشش‌های هندسی برای نقاط در حال حرکت

استاد راهنما:

دکتر محمد فرشی

استاد مشاور:

دکتر محمدرضا هوشمند اصل

پژوهش‌گر:

ذریه سلطانی گردفرامرزی

مهر ۱۳۹۰

تقدیم به

پدر و مادر عزیزم

و همه کسانی که درست اندیشیدن را به من آموختند.

سپاس‌گزاری

سپاس خداوند یکتای عزتمندی که رحمت و دانش او در سراسر گیتی گسترده شده، آسمان‌ها و زمین همه از آن اوست و علم و دانش حقیقی را بر هر که بخواهد موهبت می‌فرماید. رحمت و لطف او را بی‌نهایت سپاس می‌گویم چرا که فهم و درک مطالب این پژوهش را بر من ارزانی داشت و مرا به این اصل رساند که علم و ایمان دو بال یک پروازند. توفیق تلاش به من داد و هر بار که خطا کردم فرصتی دوباره، تا با امید، تلاشی تازه را آغاز کنم و به خواست او به نتیجه‌ی مطلوب نائل آیم. به‌راستی که همه چیز از آن اوست و همه‌چیز به خواست اوست.

از استاد راهنمای گرانقدرم، جناب آقای دکتر محمد فرشی کمال تشکر و قدردانی را دارم. ایشان با ارائه‌ی درس هندسه‌ی محاسباتی، مرا با یکی از زیباترین شاخه‌های علوم کامپیوتر آشنا کردند. روش تدریس خردمندانه و دقیق ایشان، روند حل مسائل هندسی را به من آموخت و دریافتم که چگونه به کمک هندسه می‌توان به سوالات و مسائل مهم دنیای واقعی جوابی کارا و منطقی داد. آموزه‌ها و راهنمایی‌های ارزشمندشان، به من کمک کرد تا موضوع مورد علاقه‌ام را برای پژوهش انتخاب کنم. با حمایت و دقت‌نظر ایشان روش‌های یک پژوهش را فراگرفتم. در تمام دوره‌ای که مقالات علمی مرتبط با پایان‌نامه را مطالعه و بررسی می‌کردم، ایشان با صبر، درایت و اندوخته‌های علمی ارزشمندشان، راه و روش تحقیق و پژوهش را به من آموختند. بسیار مفتخرم که این دوره را تحت این رهنمودهای ارزشمند به پایان بردم، تمام این آموخته‌ها را ارج می‌نهم و از ایشان بسیار سپاس‌گزارم.

از همراهی‌های دلسوزانه‌ی جناب آقای دکتر محمدرضا هوشمند اصل به عنوان استاد مشاور این پژوهش نیز تقدیر و تشکر می‌کنم.

از خانواده‌ام، به‌ویژه از پدر و مادرم که پشتیبان همیشگی من بوده‌اند کمال سپاس و تشکر را دارم و توفیق جبران محبت این عزیزان را از خداوند خواستارم.

از دوستان عزیزم که زحمت خواندن این پایان‌نامه را متحمل شدند و با نظرات خود مرا در هر چه بهتر کردن این پژوهش یاری کردند نیز بسیار متشکرم.

چکیده

در این پایان‌نامه، $\varepsilon + 1$ -پوشش‌های هندسی برای یک مجموعه از n نقطه در صفحه و در فضای \mathbb{R}^d مورد مطالعه قرار می‌گیرند که این پوشش‌ها می‌توانند زمانی که نقاط آن‌ها حرکت می‌کنند به صورت کارایی نگهداری شوند. پوشش وابسته به حرکت در صفحه، دارای اندازه‌ی $O(n/\varepsilon^2)$ است و با فرض این که نقاط دارای مسیرهای حرکت چندجمله‌ای از درجه‌ی حداکثر s هستند، تعداد $O(n^2\beta(n))$ رویداد را بررسی می‌کند (تابع $\beta(n)$ دارای رشدی آهسته‌تر از توابع لگاریتمی است) و پوشش می‌تواند در هر رویداد در زمان $O(1)$ به‌روزرسانی شود. پوشش وابسته به حرکت در فضای \mathbb{R}^d ، دارای اندازه‌ی $O(n/\varepsilon^{d-1})$ و حداکثر درجه‌ی $O(\log^d n)$ است و با فرض این که نقاط دارای مسیرهای حرکت چندجمله‌ای از درجه‌ی محدود هستند، تعداد $O(n^2/\varepsilon^{d-1})$ رویداد را بررسی می‌کند و هر رویداد می‌تواند در زمان $O(\log^{d+1} n)$ ، با استفاده از یک ساختمان داده‌ی کمکی که به $O((n/\varepsilon^{d-1}) \log^d n)$ حافظه نیاز دارد، پردازش شود. به‌روزرسانی یک طرح پرواز نیز تنها به $O(\log n/\varepsilon^{d-1})$ زمان نیاز دارد. علاوه بر این نتایج، این پوشش‌ها اولین $\varepsilon + 1$ -پوشش‌های وابسته به حرکتی هستند که عملکرد آن‌ها به میزان پراکندگی نقاط از هم وابسته نیست.

فهرست مطالب

ح	جدول نمادها
۱	۱ مقدمات
۲	۱.۱ مدل‌های حرکت
۳	۲.۱ دنباله‌ی داوونپورت-شینزل
۵	۳.۱ غشای محدب یک مجموعه از نقاط در صفحه
۶	۴.۱ مثلث‌بندی یک مجموعه از نقاط در صفحه
۷	۱.۴.۱ مثلث‌بندی مجاز
۹	۵.۱ تابع فاصله‌ی محدب و منصف دو نقطه بر اساس آن
۱۲	۶.۱ نمودار ورنوی
۱۴	۷.۱ مثلث‌بندی دلونی
۱۷	۸.۱ L_p -متریک‌ها
۱۸	۹.۱ نمودار ورنوی و مثلث‌بندی دلونی، بر اساس یک تابع فاصله‌ی محدب
۱۹	۱.۹.۱ تابع فاصله‌ی محدب لوزی \diamond و منصف آن
۲۱	۲.۹.۱ نمودار ورنوی بر اساس d_\diamond
۲۲	۳.۹.۱ مثلث‌بندی دلونی بر اساس d_\diamond و ویژگی‌های آن
۲۷	۱۰.۱ پوشش‌های هندسی روی مجموعه نقاط
۲۷	۱.۱۰.۱ شبکه‌های هندسی
۲۹	۲.۱۰.۱ t -پوشش‌های هندسی

۳۱	ساختارهای وابسته به حرکت	۲
۳۲	۱.۲ مفاهیم کلی و تعاریف	
۳۴	۱.۱.۲ یک مثال با شرح و توضیح	
۴۵	۲.۲ مسابقه‌ی وابسته به حرکت	
۴۷	۳.۲ نزدیک‌ترین جفت از بین نقاط در حال حرکت	
۴۷	۱.۳.۲ الگوریتم ایستای پیمایش صفحه	
۵۲	۲.۳.۲ وابسته به حرکت کردن	
۶۵	تغییرات توپولوژیکی در مثلث‌بندی دلونی بر اساس متر L_∞	۳
۶۶	۱.۳ مفاهیم مقدماتی	
۶۸	۲.۳ شمارش تعداد تغییرات توپولوژیکی	
۶۹	۱.۲.۳ شمارش تغییرات یال‌های گوشه	
۷۱	۲.۲.۳ شمارش تغییرات یال‌های غیر گوشه	
۷۲	۳.۳ تعمیم نتایج به متر d_\diamond	
۷۵	یک پوشش هندسی وابسته به حرکت در صفحه	۴
۷۶	۱.۴ کارهای مرتبط	
۷۷	۲.۴ یک پوشش ایستای مناسب	
۷۸	۱.۲.۴ ساخت پوشش	
۷۹	۲.۲.۴ تحلیل ضریب کشش و اندازه‌ی پوشش	
۸۷	۳.۴ نگهداری پوشش تحت حرکت نقاط	
۹۰	۴.۴ تحلیل پوشش وابسته به حرکت	
۹۳	یک پوشش هندسی وابسته به حرکت در فضای \mathbb{R}^d	۵
۹۴	۱.۵ مقدمه	
۹۵	۲.۵ یک پوشش ایستای مناسب	
۹۵	۱.۲.۵ ساخت پوشش بر اساس دیدگاه θ -گراف	

۱۰۰	ساخت پوشش از $CSPD$ ها	۲.۲.۵
۱۰۸	یک پوشش با اندازه‌ی خطی	۳.۲.۵
۱۰۹	نگهداری پوشش تحت حرکت نقاط	۳.۵
۱۰۹	درخت‌های محدوده بر اساس رتبه	۱.۳.۵
۱۱۵	نگهداری وابسته به حرکت T_{σ}^{rb} و گراف S	۲.۳.۵
۱۱۹	نگهداری پوشش هرس شده‌ی $S^* = (P, E_{S^*})$	۳.۳.۵
۱۲۰	تحلیل پوشش وابسته به حرکت	۴.۵

۱۲۳ ۶ نتیجه‌گیری

۱۲۵ واژه‌نامه فارسی به انگلیسی

۱۲۸ واژه‌نامه انگلیسی به فارسی

۱۳۱ مراجع

فهرست شکل‌ها

۴	پوشش پایینی یک مجموعه از توابع متناظر با دنباله‌ای از نمادها	۱.۱
۶	یک مجموعه محدب در مقایسه با یک مجموعه غیرمحدب	۲.۱
	منحنی ایجاد شده توسط کش پلاستیکی معادل غشای محدب یک مجموعه متناهی از	۳.۱
۶	نقاط در صفحه	
	شکل (الف): وجه نامحدود مثلث‌بندی یک مجموعه از نقاط P در صفحه، مکمل غشای	۴.۱
	محدب P است. شکل‌های (ب) و (ج): دو مثلث‌بندی متفاوت از یک مجموعه از نقاط در	
۷	صفحه	
۸	یک چرخش یال	۵.۱
۹	شکل مربوط به لم ۱.۴.۱	۶.۱
۱۰	شکل مربوط به محاسبه‌ی مقدار تابع فاصله‌ی محدب	۷.۱
۱۱	یک نقطه متعلق به منصف $B_C(a_1, a_2)$	۸.۱
۱۲	حالت‌های مختلف منصف	۹.۱
۱۳	نمودار ورنوی یک مجموعه از نقاط در صفحه	۱۰.۱
۱۴	یک سلول در نمودار ورنوی	۱۱.۱
۱۵	شکل (الف): گراف دوگان نمودار ورنوی و شکل (ب): گراف دلونی	۱۲.۱
۱۸	شکل (الف): دایره‌ی متر اقلیدسی، شکل (ب): دایره‌ی متر L_1 و شکل (ج): دایره‌ی متر L_∞	۱۳.۱
۱۹	لوزی باریک عمودی	۱۴.۱
۲۰	شکل‌های متفاوت منصف در متر d_\diamond	۱۵.۱
۲۱	دو حالت خاص برای نمودار ورنوی بر اساس متر d_\diamond	۱۶.۱

۱۷.۱	شکل‌های (الف) و (ب) به ترتیب: نمودار ورنوی و مثلث‌بندی دلونی بر اساس تابع فاصله‌ی
۲۳	محدب d_{\diamond}
۱۸.۱	چهار \diamond -مخروط متفاوت برای یک لوزی \diamond
۱۹.۱	شکل مربوط به اثبات لم ۳.۹.۱
۲۰.۱	یک مثلث‌بندی دلونی $DT_{\diamond}(P)$ با غشای غیرمحدب
۱.۲	نمودار ty
۲.۲	شکل (الف): تاییدیه‌های مربوط به نگهداری ترتیب نقاط متحرک روی محور y ، شکل (ب):
۳۶	بروز یک رویداد بر اثر حرکت نقاط
۳.۲	وضعیت (الف): ترتیب مرتب شده‌ی نقاط قبل از رویداد، وضعیت (ب): ترتیب مرتب شده‌ی
۳۸	نقاط بعد از رویداد
۴.۲	رویدادهای داخلی و رویدادهای خارجی برای مسئله بالاترین نقطه از بین نقاط متحرک
۵.۲	بررسی کارایی ساختار وابسته به حرکت
۶.۲	بروز یک رویداد در یک هرم وابسته به حرکت
۷.۲	شکل (الف): یک درخت مسابقه‌ی ایستا و شکل (ب): یک درخت مسابقه‌ی وابسته به
۴۵	حرکت برای مسئله‌ی بالاترین نقطه از بین نقاط متحرک روی محور y
۸.۲	قاش نفوذ یک نقطه
۹.۲	شکل اثبات لم ۱.۳.۲
۱۰.۲	شکل اثبات لم ۲.۳.۲
۱۱.۲	مجموعه نقاط ماکسیما و کاندیدای نقطه‌ی p و نقطه‌ی $l_{cand}(p)$
۱۲.۲	نقاط q و q' مقصدهای نقطه p در نمودار ماکسیما
۱۳.۲	سه نوع رویداد متفاوت
۱۴.۲	عدم پاسخ‌گویی ساختار هنگام پیاده‌سازی صریح نمودار ماکسیما
۱۵.۲	مجموعه نقاط $Parents(p)$ و تقسیم‌بندی آن به دو دسته‌ی $Parents_a(p)$ و $Parents_b(p)$
۱۶.۲	یک x رویداد و تغییر در مجموعه‌های $Cands$
۱۷.۲	دو x رویداد بدون نیاز به به‌روزرسانی فیلدهای $Cands()$ و $Parents()$

۶۰	یک $60^\circ +$ رویداد	۱۸.۲
۶۷	شکل (الف): نمودار ورنوی و شکل (ب): مثلث‌بندی دلونی متناظر با آن بر اساس متر L_∞	۱.۳
۶۸	همان مثلث‌بندی دلونی بر اساس متر L_∞ ، با نقطه‌هایی اضافه شده در بی‌نهایت	۲.۳
۶۸	تغییرات در مثلث‌بندی دلونی بر اساس متر L_∞ ، چنان‌چه تنها یک نقطه‌ی آن حرکت کند.	۳.۳
	شکل‌های (الف) و (ب): یک یال گوشه و یک یال غیر گوشه، شکل (ج): نزدیک‌ترین برخورد	۴.۳
۶۹	متناظر با یک یال گوشه برای نقطه‌ی p	
۷۲	قرار گرفتن هر یال غیر گوشه‌ی e ، در یک سلول متشکل از چهار یال گوشه	۵.۳
۷۸	شکل (الف): لوزی اولیه‌ی \diamond ، شکل (ب): لوزی \diamond که به اندازه‌ی $i \cdot \varphi$ چرخیده است .	۱.۴
۷۹	اگر لوزی \diamond به اندازه ضریب معیار s بزرگ شود، نقطه‌ی q روی مرز آن قرار می‌گیرد. .	۲.۴
۸۱	شکل مربوط به مشاهده‌ی ۳.۲.۴	۳.۴
۸۳	شکل‌های (الف) و (ب): به ترتیب برای اثبات حالت‌های اول و دوم لم ۴.۲.۴	۴.۴
	شکل (الف): چهار i -مخروط متفاوت برای یک i -لوزی، شکل (ب): (p, q) یک یال غشا	۵.۴
۸۸	و v و u رئوس غشا	
۸۹	نامعتبر شدن تاییدیه‌ی <i>NotInDiamond</i> رأس s	۶.۴
۸۹	نامعتبر شدن تاییدیه‌ی <i>NotInCone</i> رأس r مربوط به یال غشای (p, q)	۷.۴
۹۰	نامعتبر شدن تاییدیه‌ی <i>NotInCone</i> رأس s_1 ، مربوط به یال غشای (p, q)	۸.۴
۹۵	مخروط‌های کانونی در صفحه و برای $\theta = 22/5^\circ$	۱.۵
۹۷	نقطه‌ی q از لحاظ فاصله $dist_\sigma$ نزدیک‌ترین به نقطه p است.	۲.۵
۹۸	شکل مربوط به اثبات لم ۱.۲.۵	۳.۵
۱۰۰	درجه‌ی نقطه‌ی p_1 در θ -گراف، $n - 1$ است.	۴.۵
۱۰۱	$\bar{\sigma}(p)$: مخروط مزدوج $\sigma(p)$	۵.۵
۱۰۱	انتخاب نقطه‌ی z	۶.۵
۱۰۳	تعیین مجموعه $A(v)$ هر گره v	۷.۵
۱۰۵	A_i و B_i توسط مخروط‌های $\sigma(z)$ و $\bar{\sigma}(z)$ از هم مجزا می‌شوند.	۸.۵

۹.۵	شکل (الف): ساده‌ترین راه برای خوش اتصال کردن، شکل (ب): تقسیم‌بندی نقاط به
۱۰.۶	گروه‌های لگاریتمی برای خوش اتصال کردن
۱۰.۷	امکان رسم یال (u, w) به جای یال (u, v)
۱۱.۵	رتبه‌ی نقطه‌ی w در هر مختصات
۱۲.۵	اسکلت یک درخت محدوده بر اساس رتبه‌ی دو بعدی برای ۴ نقطه
۱۳.۵	ذخیره‌ی مجموعه‌های $P(v)$ برای گره‌های سطح آخر اسکلت درخت محدوده بر اساس رتبه
۱۴.۵	ذخیره‌ی مجموعه‌های $R(v)$ برای گره‌های سطح آخر اسکلت درخت محدوده بر اساس رتبه
۱۵.۵	ایجاد یک مسیر انفرادی تا برگ
۱۶.۵	درج نقطه‌ی p در گروه $P(v)^2$
۱۷.۵	درج نقطه‌ی p در گروه $P(v)^\circ$
۱۸.۵	رویداد: جابه‌جایی تصویر دو نقطه‌ی p و q باهم روی یال نماینده‌ی مخروط
۱۹.۵	رویداد: جابه‌جایی تصویر دو نقطه‌ی r و q باهم روی یال نماینده‌ی مخروط

جدول نمادها

نماد	توضیح	صفحه
$\lambda_s(n)$	طول طولانی‌ترین (n, s) دنباله‌ی داوونپورت-شینزل	۴
\mathcal{F}	یک مجموعه از توابع	۴
$\Gamma(\mathcal{F})$	پوشش پایینی مجموعه توابع \mathcal{F}	۴
$\alpha(n)$	معکوس تابع آکرمان	۵
\overline{pq}	پاره‌خط متصل کننده‌ی دو نقطه‌ی p و q	۵
$\mathcal{CH}(S)$	غشای محدب یک مجموعه S	۶
\mathcal{T}	مثلث‌بندی یک مجموعه از نقاط در صفحه	۷
$\ pa\ $	فاصله‌ی اقلیدسی بین p و a	۹
C	یک شکل محدب	۹
d_C	تابع فاصله‌ی محدب بر اساس شکل محدب C	۹

۹ منصف دو نقطه بر اساس تابع فاصله‌ی محدب d_C	B_C
۱۰ خطی که از نقاط a_1 و a_2 می‌گذرد	$a_1 a_2$
۱۱ اشعه‌ای که از a_1 شروع می‌شود و از t_1 می‌گذرد	$\overrightarrow{a_1 t_1}$
۱۳ نمودار ورنوی مجموعه نقاط P	$Vor(P)$
۱۳ مولفه‌ی x نقطه‌ی p	p_x
۱۳ مولفه‌ی y نقطه‌ی p	p_y
۱۳ سلول نقطه‌ی p_i در $Vor(P)$	$V(p_i)$
۱۴ بزرگ‌ترین دایره‌ی خالی q نسبت به P	$Circ_P(q)$
۱۴ گراف دوگان نمودار ورنوی	$DGVD$
۱۵ گراف دلونی مجموعه نقاط P	$DG(P)$
۱۵ مثلث‌بندی دلونی مجموعه نقاط P	$DT(P)$
۱۷ متر L_p	L_p
۱۷ فاصله‌ی L_p -متریک بین u و v	$\delta_p(u, v)$
۱۹ لوزی باریک عمودی	\diamond
۱۹ تابع فاصله‌ی شکل محدب \diamond	d_\diamond
۲۲ نمودار ورنوی مجموعه نقاط P بر اساس d_\diamond	$Vor_\diamond(P)$
۲۲ سلول نقطه‌ی p در $Vor_\diamond(P)$	$V_\diamond(p)$
۲۲ مثلث‌بندی دلونی مجموعه نقاط P بر اساس d_\diamond	$DT_\diamond(P)$
۲۲ دایره با مرکز بی‌نهایت	C_∞
۲۲ لوزی \diamond با مرکز بی‌نهایت	\diamond_∞
۲۷ گراف اقلیدسی	\mathcal{G}
۲۹ ضریب کشش یک پوشش	t
۲۹ کوتاه‌ترین مسیر بین دو نقطه‌ی p و q در \mathcal{G}	$d_{\mathcal{G}}(p, q)$
۲۹ مقداری کوچک بین صفر و یک	ε
۴۸ قاش نفوذ نقطه‌ی p	$Dom(p)$

۴۹ دایره‌ای با شعاع r و مرکز p	$Circ(p, r)$
۴۹ p در سمت چپ q (یا در پایین q ، در صورتی که مولفه‌ی x آن‌ها یکسان باشد)	$p \prec q$
۵۱ مجموعه نقاط ماکسیمای نقطه‌ی p	$Maxima(p)$
۵۱ مجموعه نقاط کاندید برای نقطه‌ی p	$Cands(p)$
۵۱ سمت چپ‌ترین نقطه‌ی $Cands(p)$	$lcand(p)$
۵۲ اندازه‌ی مجموعه‌ی A	$ A $
۵۶ تمام نقاطی که p یک مقصد برای آن‌ها در نمودار ماکسیما است	$Parents(p)$
۷۶ پراکندگی مجموعه نقاط یک پوشش	ρ
۷۸ یک مجموعه از نسخه‌های چرخیده شده‌ی لوزی \diamond	\mathcal{D}
۷۸ لوزی باریک اولیه برای ساخت پوشش	\diamond
۷۸ زاویه‌ی کوچک‌تر \diamond	ϕ
۷۸ لوزی \diamond که حول مبدأ به اندازه‌ی زاویه‌ی $i \cdot \phi$ چرخیده است	\diamond_i
۷۸ زاویه‌ی چرخش \diamond حول مبدأ	φ
۷۸ تابع فاصله‌ی محدب بر اساس شکل محدب \diamond_i	d_i
۷۹ مثلث‌بندی دلونی مجموعه نقاط P بر اساس d_i	$DT_i(P)$
۷۹ پوشش لوزی-دلونی مجموعه نقاط P در صفحه	$DDS(P)$
۸۴ زاویه‌ی بین \vec{pq} و محور مثبت x	$\alpha(p, q)$
۹۵ t -پوشش در فضای \mathbb{R}^d	\mathcal{S}
۹۵ کوتاه‌ترین مسیر بین دو نقطه‌ی p و q در \mathcal{S}	$d_{\mathcal{S}}(p, q)$
۹۵ زاویه‌ی متعلق به دیدگاه θ -گراف	θ
۹۵ یک مجموعه از مخروط‌های کانونی	TC
۹۶ یک مخروط کانونی متعلق به TC	σ
۹۶ نسخه‌ی انتقال یافته‌ی σ که رأس آن روی نقطه‌ی p است	$\sigma(p)$
۹۶ فاصله‌ی p تا تصویر عمودی q روی یال نماینده‌ی $\sigma(p)$	$dist_{\sigma}(p, q)$
۱۰۱ نمادی برای مخفف شده‌ی عبارت "یک تجزیه‌ی جفت مخروط-مجزا"	$CSPD$

۱۰۱	یک تجزیه‌ی جفت مخروط-مجزا برای مخروط σ	Ψ_σ
۱۰۱	یک جفت مخروط-مجزا متعلق به Ψ_σ	(A_m, B_m)
۱۰۲	درخت محدوده‌ی چندبعدی	\mathcal{T}_σ
۱۰۲	زیرمجموعه‌ی کانونی گره v در سطح d در \mathcal{T}_σ	$B(v)$
۱۰۲	زیرمجموعه‌ای از نقاط P که هنگام پرسش با مخروط آن‌ها $B(v)$ را انتخاب کرده‌اند	$A(v)$
۱۰۷	مجموعه یال‌هایی که بین نقاط متعلق به A_i و B_i رسم شده است	$E(A_i, B_i)$
۱۰۸	پوشش هرس شده‌ی \mathcal{S}	\mathcal{S}^*
۱۱۰	صفحه i ام از صفحات سازنده‌ی مخروط σ	h_i
۱۱۰	محور مختصات صفحه‌ی h_i	x_i
۱۱۰	مولفه‌ی x_i نقطه‌ی p	$x_i(p)$
۱۱۰	رتبه‌ی $x_i(p)$ در مجموعه‌ی مرتب شده‌ی نقاط بر اساس مولفه‌ی x_i	$Rank_i(p)$
۱۱۱	اسکلت یک درخت محدوده بر اساس رتبه	$\mathcal{T}_\sigma^{skel}$
۱۱۲	زیرمجموعه‌ی کانونی v در $\mathcal{T}_\sigma^{skel}$	$P(v)$
۱۱۳	زیرمجموعه‌ای از نقاط P که هنگام پرسش با مخروط آن‌ها $P(v)$ را انتخاب کرده‌اند	$R(v)$
۱۱۵	درخت محدوده بر اساس رتبه	\mathcal{T}_σ^{rb}
۱۱۵	یک آرایه از تمام نقاط P ، مرتب شده طبق مولفه‌ی x_i آن‌ها	$A_i[1..n]$
۱۱۹	یک آرایه از تمام نقاط P ، مرتب شده طبق تصویرشان روی یال نماینده‌ی σ	$A_\sigma[1..n]$
۱۲۴	وزن درخت پوشای کمینه‌ی مجموعه نقاط P	$wt(MST(P))$

فصل ۱

مقدمات

۱.۱ مدل‌های حرکت

بسته به کاربرد، حرکت مجموعه نقاط در فضای دلخواه از راه‌های مختلف نمایش داده می‌شود. حرکت می‌تواند به صورت صریح با توابع چندجمله‌ای، به صورت ضمنی با معادلات دیفرانسیلی، یا به صورت آماری با مدل‌های احتمالی نمایش داده شود. همان‌طور که بعداً دیده خواهد شد، در مسائل وابسته به حرکت که نقاط مسئله به‌طور پیوسته در حال حرکت هستند، لازم است تا وضعیت نقاط در هر زمان با مجموعه‌ای از شرط‌های جبری مشخص شود. علاوه بر این نیاز است تا مسیر حرکت نقاط به‌گونه‌ای تعریف شود که تعداد دفعاتی که یک شرط جبری ممکن است نامعتبر شود ثابت باشد. بدین منظور در این پایان‌نامه، فرض می‌شود که مسیر حرکت نقاط در صفحه به صورت حرکت‌های شبه‌جبری^۱ است. برای تعریف این نوع حرکت‌ها لازم است توابع شبه‌جبری تعریف شوند، که از تعریف زانگ^۲ مطرح شده در [۴۰] استفاده می‌شود.

تعریف ۱.۱.۱ توابع پیوسته‌ی یک متغیره‌ی $f_1(x), f_2(x), \dots, f_m(x)$ توابع شبه‌جبری از درجه‌ی حداکثر s

هستند، هرگاه برای هر تابع چندجمله‌ای m متغیره‌ی g از درجه‌ی s_1 ، تابع

$$h(x) = g(f_1(x), f_2(x), \dots, f_m(x))$$

متحد صفر باشد یا حداکثر $s \times s_1$ ریشه داشته باشد.

برای نمونه، توابع چندجمله‌ای یا منطقی با درجه‌ی ثابت توابع شبه‌جبری هستند. یک مجموعه از نقاط دارای حرکت‌های شبه‌جبری از زمان هستند هرگاه مسیر حرکت آن‌ها با توابع شبه‌جبری از زمان توصیف شود. در ادامه، تعریف ۱.۱.۱ با شرح یک مثال توضیح داده می‌شود.

مسئله‌ی یک بعدی زیر که ترتیب n نقطه‌ی در حال حرکت روی محور x ها را گزارش می‌کند، در نظر بگیرید. هر نقطه‌ی p_i دارای مسیر حرکت پیوسته‌ای از زمان است که با $f_i(t)$ نشان داده می‌شود و t به زمان اشاره می‌کند. مقدار هر نقطه‌ی p_i در زمان t ، یعنی مولفه‌ی x نقطه، با $v_i(t)$ نشان داده می‌شود. فرض کنید مجموعه توابع حرکت نقاط، یعنی $f_1(t), f_2(t), \dots, f_n(t)$ توابع شبه‌جبری از درجه‌ی s باشند و شرط‌های جبری برای تعیین موقعیت نقاط متحرک از نوع مقایسه تعریف شوند. اگر برای نشان دادن ترتیب هر دو نقطه‌ی متوالی p_i و p_j روی محور x در زمان t ، به‌طوری که $v_j(t) > v_i(t)$ باشد، از شرط جبری $v_j(t) - v_i(t) > 0$ استفاده شود، آنگاه با توجه به تعریف ۱.۱.۱، می‌توان تابع دو متغیره‌ی خطی $g(f_j, f_i)(t) = f_j(t) - f_i(t)$

^۱Pseudo algebraic motions

^۲Zhang

(که طبق تعریف دارای $s_1 = 1$ و $m = 2$ است)، را به عنوان تابع $h(t)$ در نظر گرفت و به این نتیجه رسید که تابع $h(t) = g(f_j, f_i)(t)$ یا متحد صفر است یا دارای حداکثر $1 \times s$ ریشه است. این بدان معنی است که شرط جبری $v_j(t) - v_i(t) > 0$ با گذشت زمان حداکثر به تعداد s بار صفر می‌شود و متعاقباً تغییر علامت می‌دهد (تغییر ترتیب نقاط).

در تحلیل مسائلی که بعداً مطرح خواهند شد، بسیار اهمیت دارد که تعداد دفعاتی که شرط جبری مربوط به یک نقطه با گذشت زمان صفر می‌شود ثابت باشد. از آنجایی که در بیش‌تر مسائل شرط‌های جبری که برای تعیین وضعیت نقاط معرفی می‌شوند، چند جمله‌ای‌هایی از درجه‌ی کم، تعریف شده روی یک تعداد ثابت از نقاط هستند، فرض شبه‌جبری بودن حرکت نقاط کافی است تا برقراری این موضوع را تضمین کند. برای نمونه، در مثال بالا شرط‌های جبری (مقایسه‌ی دو مقدار) از درجه‌ی یک و تعریف شده روی دو نقطه هستند، بنابراین با فرض شبه‌جبری بودن حرکت نقاط، شرط جبری مربوط به یک نقطه همان تابع g مطرح شده در تعریف ۱.۱.۱ می‌شود که حداکثر به تعداد $1 \times s$ بار صفر خواهد شد (s یک عدد ثابت که درجه‌ی توابع شبه‌جبری حرکت را نشان می‌دهد).

۲.۱ دنباله‌ی داوونپورت-شینزل

تعریف ۱.۲.۱ یک (n, s) دنباله‌ی داوونپورت-شینزل^۳، که n و s اعداد صحیح مثبت هستند، یک دنباله‌ی ساخته شده از n نماد با این خواص است که هیچ دو نماد مجاور در دنباله یکسان نیستند و این‌که برای هر دو نماد مجزای a و b حداکثر s تناوب از آن‌ها در دنباله وجود دارد.

در تعریف بالا، منظور از تناوب a و b این است که نماد b بعد از نماد a و نماد a بعد از نماد b در دنباله ظاهر شود، ولی نه الزاماً مجاور به هم (کنار هم). مثلاً دنباله‌ی زیر، تشکیل شده از نمادهای a, b, c, d را در نظر بگیرید:

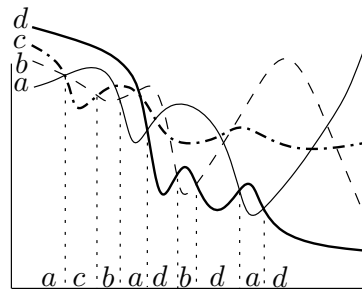
$$\underline{acbdbca_cdb_a_dcd_cadcb}$$

تعداد تناوب‌های a و b در دنباله، یعنی مجموع تعداد دفعاتی که b بعد از a و a بعد از b در دنباله ظاهر شده است، برابر با ۵ است. با توجه به تعریف (n, s) دنباله‌ی داوونپورت-شینزل، می‌توان دریافت که با یک n و s معلوم (داده شده)، بسته به مقدارهای n و s ، دنباله‌های داوونپورت-شینزل متعددی می‌توان یافت، اما همگی

^۳Davenport-Schinzel sequence

دارای طول‌های متنهای هستند؛ زیرا با توجه به تعریف، امکان وجود دو نماد مجاور یکسان در دنباله نیست و نیز تعداد تناوب‌های هر دو نماد مجزا در دنباله به تعداد حداکثر s محدود شده است. مثلاً برای $n = 3$ و $s = 2$ و نمادهای a, b, c طول $(3, 2)$ دنباله‌های داونپورت-شینزل ممکن، حداکثر ۵ است؛ زیرا هر دنباله با طول ۶ یا بیش‌تر متشکل از این نمادها، یا حداقل برای یک جفت نماد مجزا دارای بیش از ۲ تناوب خواهد بود یا دو نماد مجاور یکسان خواهد داشت و در نتیجه، شرایط تعریف یک $(3, 2)$ دنباله‌ی داونپورت-شینزل را نخواهد داشت، به عنوان مثال دنباله‌ی $abc bab$ که ۳ تناوب از a و b را دارد. بنابراین طول طولانی‌ترین (n, s) دنباله‌ی داونپورت-شینزل قابل تعریف خواهد بود و با $\lambda_s(n)$ نشان داده می‌شود. در ادامه به بیان اهمیت و کاربرد این دنباله‌ها در تحلیل مسائلی مهم در هندسه‌ی محاسباتی پرداخته می‌شود.

اگر $\mathcal{F} = \{f_i\}$ یک مجموعه از توابع باشد، پوشش پایینی^۴ برای مجموعه \mathcal{F} برابر تابع $\min f_i(x)$ است که با $\Gamma(\mathcal{F})$ نشان داده می‌شود. به طور مشابه $\max f_i(x)$ به عنوان پوشش بالایی^۵ تعریف می‌شود. پیچیدگی $\Gamma(\mathcal{F})$ نیز برابر با تعداد دفعاتی که تابع موجود در $\Gamma(\mathcal{F})$ عوض می‌شود، یعنی تعداد نقاط شکست $\Gamma(\mathcal{F})$ تعریف می‌شود.



شکل ۱.۱: پوشش پایینی یک مجموعه از توابع متناظر با دنباله‌ای از نمادها

اگر \mathcal{F} مجموعه‌ای از n تابع چندجمله‌ای با درجه‌ی s باشد. با توجه به این که هر دو تابع چندجمله‌ای از درجه‌ی s حداکثر در s نقطه با یکدیگر برخورد می‌کنند (این بدان معنی است که حداکثر s تناوب از هر دو تابع مجزا از \mathcal{F} در $\Gamma(\mathcal{F})$ وجود خواهد داشت) و نیز وجود دو تابع یکسان مجاور به هم در $\Gamma(\mathcal{F})$ هم امکان ندارد، به راحتی می‌توان نتیجه گرفت که $\Gamma(\mathcal{F})$ متناظر با یک (n, s) دنباله‌ی داونپورت-شینزل است و پیچیدگی آن نیز، برابر با طول دنباله‌ی داونپورت-شینزل متناظر با آن خواهد شد. شکل ۱.۱ را ببینید. پس می‌توان گفت که

^۴Lower envelope

^۵Upper envelope

پیچیدگی $\Gamma(\mathcal{F})$ از مرتبه $O(\lambda_s(n))$ است. تمام نتایج به طور مشابه برای پیچیدگی پوشش بالایی نیز برقرار است. اگر دامنه‌ی تعریف $\Gamma(\mathcal{F})$ در نظر گرفته شود و هر تابع چندجمله‌ای از \mathcal{F} روی قسمتی از این دامنه (یک بازه مشخص از دامنه) تعریف شود، یعنی نمودار هر تابع تکه‌ای از نمودار آن تابع با دامنه نامحدود باشد، آنگاه پیچیدگی $\Gamma(\mathcal{F})$ برابر $O(\lambda_{s+2}(n))$ می‌شود [۳۸]. برای یک ثابت $s \geq 3$ ، $\lambda_s(n)$ یک تابع ابرخطی است اما خیلی آهسته رشد می‌کند. در ادامه قضیه‌ای بیان می‌شود که فرمول‌های مربوط به محاسبه $\lambda_s(n)$ را بیان می‌کند. تابع $\alpha(n)$ به معکوس تابع آکرمان^۶ اشاره می‌کند [۳۸].

قضیه ۲.۲.۱ [۴۰] پیچیدگی پوشش پایینی (یا بالایی) یک مجموعه از n تابع چندجمله‌ای با درجه s برابر $O(\lambda_s(n))$ است، که $\lambda_s(n)$ در روابط زیر صدق می‌کند.

$$\lambda_1(n) = n$$

$$\lambda_2(n) = 2n - 1$$

$$\lambda_3(n) = \Theta(n\alpha(n))$$

$$\lambda_s(n) \leq n 2^{(1+o(1))\alpha(n) \frac{s-2}{3}} \quad \text{اگر } n \text{ زوج باشد،}$$

$$\lambda_s(n) \leq n 2^{(1+o(1))\alpha(n) \frac{s-2}{3} \log \alpha(n)} \quad \text{اگر } n \text{ فرد باشد،}$$

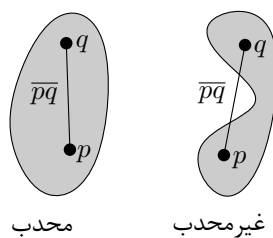
از آن جایی که $\alpha(n)$ تابعی است که بی‌نهایت آهسته رشد می‌کند (تقریباً برای مقادیر عملی و منطقاً بزرگ n مقدار ثابت است)، بنابراین برای یک مقدار ثابت s ، $\lambda_s(n)$ تقریباً خطی از n است.

۳.۱ غشای محدب یک مجموعه از نقاط در صفحه

تعریف ۱.۳.۱ یک زیرمجموعه S از صفحه محدب^۷ نامیده می‌شود اگر و تنها اگر برای هر جفت نقاط $p, q \in S$ پاره خط \overline{pq} کاملاً در داخل S واقع شود. شکل ۲.۱ را ببینید.

^۶Ackermann function

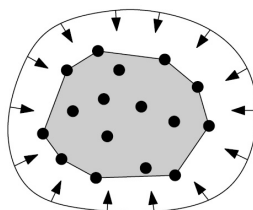
^۷Convex



شکل ۲.۱: یک مجموعه محدب در مقایسه با یک مجموعه غیرمحدب

تعریف ۲.۳.۱ غشای محدب^۱ یک مجموعه S ، کوچک‌ترین مجموعه محدبی است که S را دربر می‌گیرد و با $CH(S)$ نشان داده می‌شود.

با دقت بیشتر، غشای محدب در واقع اشتراک تمام مجموعه‌های محدبی است که S را در برمی‌گیرند. برای درک بهتر غشای محدب یک مجموعه منتهای P از n نقطه در صفحه، می‌توان تصور کرد که نقاط میخ‌هایی هستند که در صفحه کوبیده شده‌اند، و غشای محدب P ، منحنی ایجاد شده توسط کش پلاستیکی است که دور میخ‌ها انداخته شده است. شکل ۳.۱ را ببینید. این دیدگاه منجر به تعریف دیگری از غشای محدب P می‌شود: چندضلعی محدب یکتایی که رأس‌هایش نقاطی از P هستند و سایر نقاط P را در برمی‌گیرد. شکل ۳.۱ را ببینید.



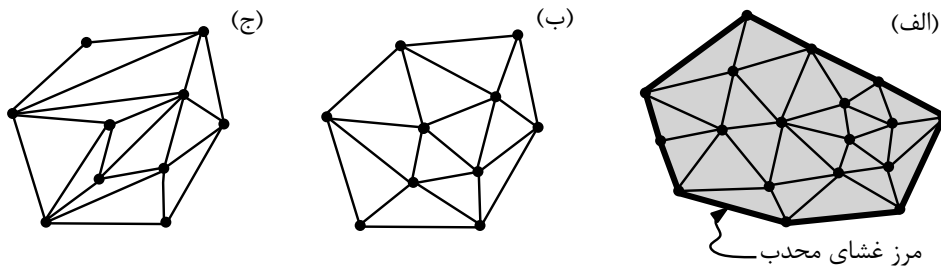
شکل ۳.۱: منحنی ایجاد شده توسط کش پلاستیکی معادل غشای محدب یک مجموعه منتهای از نقاط در صفحه

۴.۱ مثلث‌بندی یک مجموعه از نقاط در صفحه

در تعاریفی که در ادامه بیان می‌شوند، منظور از یال، خط مستقیم متصل‌کننده دو نقطه (رأس) است و منظور از یال‌های نامتقاطع^۲، یال‌هایی است که همدیگر را به غیر از رئوس در جای دیگری قطع نکنند. برای این‌که

^۱Convex hull

^۲Non-crossing edges



شکل ۴.۱: (الف) وجه نامحدود مثلث‌بندی یک مجموعه از نقاط P در صفحه، مکمل غشای محدب P است. شکل‌های (ب) و (ج): دو مثلث‌بندی متفاوت از یک مجموعه از نقاط در صفحه

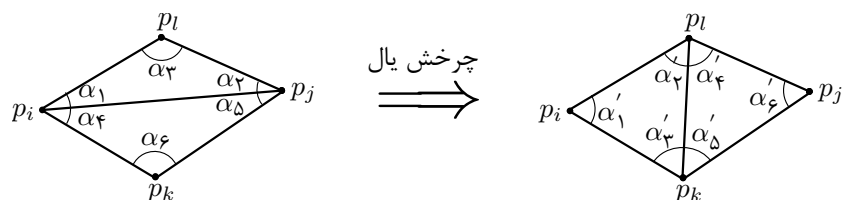
بتوان تعریف استاندارد از مثلث‌بندی P بیان کرد، لازم است ابتدا یک تقسیم‌بندی ماکسیمال یال‌های نامتقاطع^{۱۰} از صفحه، به این صورت تعریف شود: یک تقسیم‌بندی از صفحه به تعداد متناهی وجه به طوری که نتوان هیچ یال دیگری به تقسیم‌بندی اضافه کرد، به طوری که یال‌ها همچنان نامتقاطع باقی بمانند. به عبارت دیگر، هر یالی که به تقسیم‌بندی ماکسیمال یال‌های نامتقاطع تعلق ندارد، در صورت اضافه شدن به آن، در نقطه‌ای به غیر از رئوس، با یکی از یال‌های موجود برخورد خواهد داشت. یک مثلث‌بندی از مجموعه نقاط P در صفحه، یک تقسیم‌بندی ماکسیمال یال‌های نامتقاطع است که مجموعه رأس‌هایش P است.

بدیهی است که با این تعریف یک مثلث‌بندی وجود دارد. اما آیا تمام وجه‌های آن مثلث‌بندی، مثلث هستند؟ جواب مثبت است و تمام وجه‌ها به جز وجه نامحدود باید مثلث باشند؛ زیرا هر وجه محدود یک چندضلعی است، و چنانچه در ([۲۴]، فصل ۳) بیان شده است، هر چندضلعی با اضافه کردن تعداد محدودی یال می‌تواند مثلث‌بندی شود و از آنجایی که طبق تعریف ذکر شده، مثلث‌بندی، یک مجموعه ماکسیمال یال‌های نامتقاطع است (امکان اضافه کردن یال وجود ندارد)، بنابراین تمام وجه‌های محدود مثلث هستند. در مورد وجه نامحدود نیز به راحتی دیده می‌شود که هر پاره‌خطی که دو نقطه متوالی روی مرز غشای محدب P ، را به هم وصل می‌کند یک یال در مثلث‌بندی T است. این بدان معنی است که اجتماع وجه‌های محدود T همیشه غشای محدب P است و وجه نامحدود همیشه مکمل غشای محدب است. قسمت (الف) شکل ۴.۱ را ببینید.

۱.۴.۱ مثلث‌بندی مجاز

با توجه به امکان مثلث‌بندی یک مجموعه از نقاط به شکل‌های مختلف، در بسیاری از کاربردها، نیاز به مثلث‌بندی‌های خاصی است که در آن‌ها، تا جای ممکن، مثلث‌هایی باریک و بلند با زاویه‌هایی کوچک وجود

^{۱۰}Maximal subdivision of non-crossing edges



شکل ۵.۱: یک چرخش یال

نداشته باشند. قسمت‌های (ب) و (ج) شکل ۴.۱ را ببینید.

مثلث‌بندی \mathcal{T} را برای مجموعه نقاط $P = \{p_1, p_2, \dots, p_n\}$ در نظر بگیرید. فرض کنید $e = \overline{p_i p_j}$ یک یال از \mathcal{T} است که متعلق به وجه نامحدود نیست و مجاور به دو مثلث $p_i p_j p_l$ و $p_i p_j p_k$ است. اگر این دو مثلث یک چهارضلعی محدب را شکل دهند، می‌توان یک مثلث‌بندی جدید \mathcal{T}' را با حذف یال $\overline{p_i p_j}$ و درج یال $\overline{p_k p_l}$ به دست آورد. که به این عمل **چرخش یال**^{۱۱} گفته می‌شود (شکل ۵.۱). در نتیجه‌ی یک چرخش یال، شش زاویه‌ی $\alpha_1, \dots, \alpha_6$ در \mathcal{T} با شش زاویه‌ی $\alpha'_1, \dots, \alpha'_6$ در \mathcal{T}' جابه‌جا می‌شود. شکل ۵.۱ را ببینید.

یال $e = \overline{p_i p_j}$ یک **یال غیرمجاز**^{۱۲} نامیده می‌شود اگر

$$\min_{1 \leq i \leq 6} \alpha_i < \min_{1 \leq i \leq 6} \alpha'_i.$$

در ادامه، در لم زیر یک روش هندسی برای تشخیص مجاز یا غیرمجاز بودن یک یال بیان می‌شود که منجر به یک تعریف مناسب برای مثلث‌بندی مجاز^{۱۳} خواهد شد.

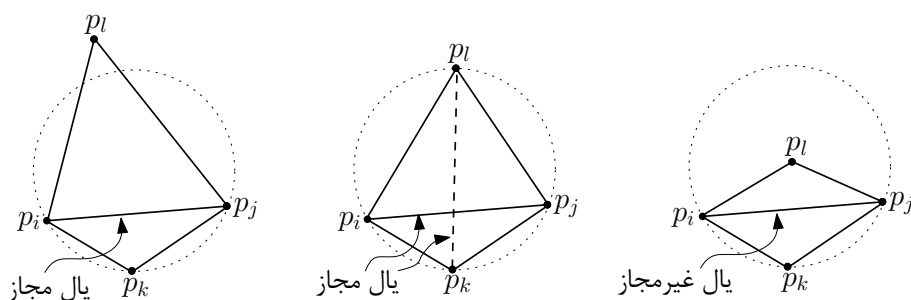
لم ۱.۴.۱ ([۲۴]) یال $\overline{p_i p_j}$ را مجاور به مثلث‌های $p_i p_j p_l$ و $p_i p_j p_k$ و $Circ$ را دایره‌ای که از p_i و p_j و p_k می‌گذرد در نظر بگیرید. یال $\overline{p_i p_j}$ غیرمجاز است اگر و تنها اگر نقطه p_l در داخل $Circ$ واقع شود. یال $\overline{p_i p_j}$ مجاز است اگر و تنها اگر نقطه p_l روی $Circ$ یا خارج از آن واقع شود. شکل ۶.۱ را ببینید.

بنابراین، با توجه به لم ۱.۴.۱ می‌توان گفت، یک مثلث‌بندی از یک مجموعه از نقاط در صفحه که هیچ یال غیرمجازی نداشته باشد، یک مثلث‌بندی مجاز از آن مجموعه نقاط است.

^{۱۱}Edge flip

^{۱۲}Illegal edge

^{۱۳}Legal triangulation



شکل ۶.۱: شکل مربوط به لم ۱.۴.۱

۵.۱ تابع فاصله‌ی محدب و منصف دو نقطه بر اساس آن

فرض کنید C یک شکل محدب در صفحه باشد (که لزوماً متقارن نیست) به طوری که مبدأ مختصات O را به عنوان مرکزش دربر می‌گیرد. برای هر دو نقطه‌ی دلخواه p و a ، ابتدا C با بردار p انتقال داده می‌شود، به طوری که مرکز C روی نقطه p قرار گیرد. سپس اشعه‌ای که از p شروع می‌شود و از a می‌گذرد را در نظر بگیرید. v را نقطه یکتای حاصل از برخورد اشعه با مرز C بگیرید. شکل ۷.۱ را ببینید. یک تابع فاصله‌ی محدب^{۱۴} بر اساس C بین دو نقطه‌ی p و a به این صورت تعریف می‌شود:

$$d_C(p, a) = \frac{\|pa\|}{\|pv\|}$$

و با d_C نشان داده می‌شود که $\|pa\|$ به فاصله‌ی اقلیدسی بین p و a اشاره می‌کند. مقدار $d_C(p, a)$ دقیقاً برابر است با ضریبی که C با مرکز p باید به اندازه‌ی آن انبساط یا انقباض داده شود تا مرزش a را لمس کند و به آن ضریب معیار^{۱۵} گفته می‌شود. تابع d_C همواره دو خاصیت از متر را دارد.

$$(۱) \text{ انعکاسی: } d_C(p, p) = ۰ \text{ و } d_C(p, a) \geq ۰$$

$$(۲) \text{ نامساوی مثلثی: } d_C(p, a) \leq d_C(p, b) + d_C(b, a)$$

در صورتی که C نسبت به مرکزش متقارن باشد خاصیت تقارنی $d_C(p, a) = d_C(a, p)$ را نیز خواهد داشت و در نتیجه d_C یک متر می‌شود [۳۵، ۱۹].

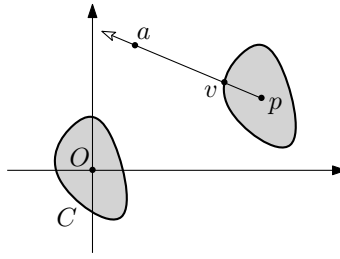
a_1 و a_2 را دو نقطه‌ی مجزا در صفحه در نظر بگیرید. منصف^{۱۶} این دو نقطه بر اساس تابع فاصله‌ی محدب

d_C ، با نماد $B_C(a_1, a_2)$ نشان داده می‌شود و تعریف آن در ادامه بیان شده است.

^{۱۴}Convex distance function

^{۱۵}Scale factor

^{۱۶}Bisector



شکل ۷.۱: شکل مربوط به محاسبه‌ی مقدار تابع فاصله‌ی محدب

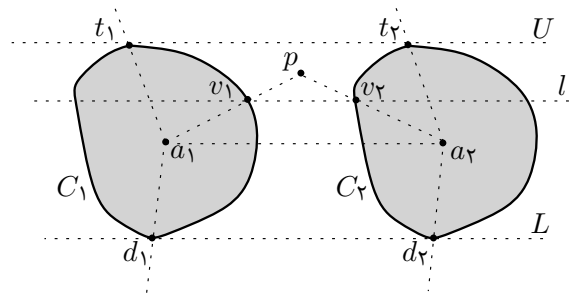
تعریف ۱.۵.۱ منصف $B_C(a_1, a_2)$ بر اساس تابع فاصله‌ی محدب d_C مکان هندسی تمام نقاط p ای است که

$$d_C(a_1, p) = d_C(a_2, p)$$

مثال شناخته شده، منصف دو نقطه در متر اقلیدسی است که خط عمود بر پاره خط $\overline{a_1 a_2}$ است که از وسط a_1 و a_2 می‌گذرد. اما در حالت کلی، $B_C(a_1, a_2)$ یک خط نیست که در ادامه بیان می‌شود.

برای رسم منصف دو نقطه‌ی a_1 و a_2 بر اساس یک تابع فاصله‌ی محدب d_C ، لازم است تا دو نسخه از شکل محدب C که تابع فاصله بر اساس آن تعریف شده است، به گونه‌ای انتقال داده شود که مرکز C روی آن نقاط واقع شود. این نسخه‌های انتقال داده شده، C_1 و C_2 نامیده می‌شوند. بدون از دست دادن کلیت مسئله فرض می‌شود که خط $a_1 a_2$ افقی است و a_1 در سمت چپ a_2 واقع شده است. خط گذرنده از بالاترین نقاط در C_1 و C_2 که موازی با خط $a_1 a_2$ است با U نشان داده می‌شود. به طور مشابه خط گذرنده از پایین‌ترین نقاط در C_1 و C_2 که موازی با خط $a_1 a_2$ است با L نشان داده می‌شود. نقطه‌ی بالایی در C_1 نقطه‌ی t_1 است که نقطه‌ی برخورد C_1 و U است و نقطه‌ی پایینی در C_1 نقطه‌ی d_1 است که نقطه‌ی برخورد C_1 و L است (در صورت منحصر به فرد نبودن نقاط بالایی و پایینی، سمت چپ‌ترین نقطه‌ی برخورد انتخاب می‌شود). نقاط بالایی و پایینی در C_2 که به طور مشابه به عنوان نقاط برخورد C_2 با U و L تعریف می‌شوند، به ترتیب نقاط t_2 و d_2 هستند (در صورت منحصر به فرد نبودن این نقاط، سمت راست‌ترین نقطه‌ی برخورد انتخاب می‌شود). شکل ۸.۱ را ببینید.

در ادامه روش به دست آوردن نقطه‌ی p متعلق به منصف $B_C(a_1, a_2)$ توضیح داده می‌شود. شکل ۸.۱ را ببینید. ابتدا خط l موازی با خط وصل کننده دو مرکز a_1 و a_2 در فاصله‌ی کمی بالاتر (یا پایین‌تر) از پاره خط $\overline{a_1 a_2}$ رسم می‌شود به طوری که با مرزهای C_1 و C_2 برخورد داشته باشد. نقاط یکتای حاصل از برخورد l با مرزها، v_1 و v_2 نامیده می‌شوند. سپس نقطه برخورد خطوط $a_1 v_1$ و $a_2 v_2$ که p نامیده می‌شود، یک نقطه متعلق به منصف $B_C(a_1, a_2)$ خواهد بود؛ زیرا، در مثلث $a_1 p a_2$ خط $v_1 v_2$ موازی با خط $a_1 a_2$ رسم شده است



شکل ۸.۱: یک نقطه متعلق به منصف $B_C(a_1, a_2)$

و طبق قضیه تالس^{۱۷}،

$$\frac{\|pa_1\|}{\|pv_1\|} = \frac{\|pa_2\|}{\|pv_2\|}$$

$$d_C(a_1, p) = d_C(a_2, p).$$

لم زیر نشان می‌دهد که منصف دو نقطه بر اساس تابع فاصله‌ی محدب شبیه به یک خط رفتار می‌کند، یعنی همومورفیک^{۱۸} به یک خط است. لازم به ذکر است که دو فضای X و Y همومورفیک هستند، اگر یک تابع یک به یک f از X به Y وجود داشته باشد به طوری که f و f^{-1} پیوسته باشند. تابع f یک همومورفیسم^{۱۹} نامیده می‌شود.

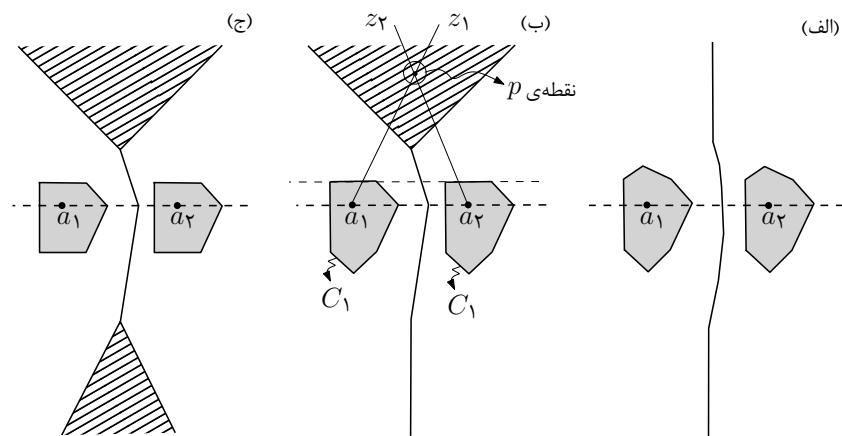
لم ۲.۵.۱ ([۳۵]) اگر خط a_1a_2 با هیچ کدام از پاره خط‌های مرز شکل محدب C موازی نباشد، آنگاه منصف $B_C(a_1, a_2)$ همومورفیک به یک خط است و کاملاً در داخل نوار خمیده‌ای که با اشعه‌های $\overrightarrow{a_1d_1}$ ، $\overrightarrow{a_2t_2}$ ، $\overrightarrow{a_1t_1}$ و $\overrightarrow{a_2d_2}$ تعریف می‌شود واقع شده است.

منصف تنها در موارد خاص و معینی شبیه به یک خط نیست، در این وضعیت، منصف می‌تواند شامل یک یا دو ناحیه‌ی دو بعدی نامحدود شود. شکل ۹.۱ را ببینید. در قسمت (الف) این شکل خط a_1a_2 موازی با هیچ کدام از مرزهای شکل محدب نیست و منصف شبیه به یک خط می‌شود. در قسمت (ب) خط a_1a_2 موازی با یک پاره خط از مرز شکل محدب است و یک قسمت از منصف یک ناحیه‌ی نامحدود می‌شود و در قسمت (ج) خط a_1a_2 موازی با دو پاره خط از مرز شکل محدب است و دو قسمت از منصف، ناحیه‌ی نامحدود می‌شود.

^{۱۷}Thales

^{۱۸}Homeomorphic

^{۱۹}Homeomorphism



شکل ۹.۱: حالات‌های مختلف منصف

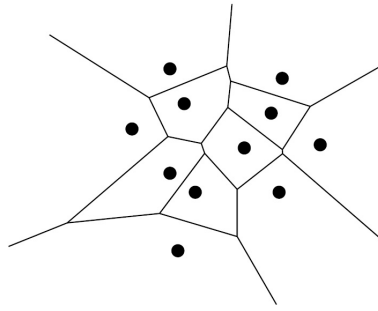
نتیجه ۳.۵.۱ ([۳۵]) منصف $B_C(a_1, a_2)$ به ترتیب یک یا دو ناحیه‌ی نامحدود را در برمی‌گیرد اگر و تنها اگر خط $a_1 a_2$ موازی با به ترتیب یک یا دو پاره‌خط از مرز شکل محدب C باشد.

همان‌طور که در شکل ۹.۱، قسمت (ب) می‌بینید، خط $a_1 a_2$ موازی با یک پاره‌خط از مرز شکل محدب C_1 شده است. خط z_1 را هر خط گذرنده از این پاره‌خط در شکل محدب C_1 و مرکز آن a_1 بگیرید. خط z_2 را نیز، هر خط گذرنده از این پاره‌خط در C_2 و مرکز a_2 بگیرید. در این وضعیت، هر نقطه‌ی p که محل برخورد دو خط z_1 و z_2 است، یک نقطه از منصف خواهد بود. در نتیجه، مکان هندسی نقاط p ، یک ناحیه را شکل می‌دهد (به‌جای یک خط).

بنابراین، زمانی که بیان می‌شود منصف $B_C(a_1, a_2)$ شامل یک یا دو ناحیه‌ی نامحدود می‌شود، این بدین معنی است که هر نقطه از این نواحی نامحدود دارای فاصله‌ی یکسان تا دو نقطه‌ی a_1 و a_2 هستند. به عبارت دیگر اگر قسمتی از منصف شبیه به یک خط و قسمتی از آن ناحیه‌ی نامحدود باشد، هر نقطه‌ی ناحیه‌ی نامحدود همان خواصی از منصف را دارد که هر نقطه‌ی متعلق به قسمت شبیه به خط دارد.

۶.۱ نمودار ورنوی

مجموعه‌ی $P = \{p_1, p_2, \dots, p_n\}$ را یک مجموعه‌ی متناهی از نقاط در صفحه در نظر بگیرید. هدف تقسیم‌بندی صفحه، به‌گونه‌ای است که به هر نقطه‌ی $p_i \in P$ ناحیه‌ای اختصاص داده شود، به‌طوری‌که هر نقطه‌ی x متعلق به این ناحیه، به نقطه‌ی p_i نزدیک‌تر از سایر نقاط دیگر در P باشد. به تقسیم‌بندی که



شکل ۱۰.۱: نمودار ورنوی یک مجموعه از نقاط در صفحه

این ویژگی را داشته باشد نمودار ورنوی^{۲۰} گفته می‌شود و با $Vor(P)$ نشان داده می‌شود. شکل ۱۰.۱ را ببینید. نمودار ورنوی که یک ساختار هندسی پرکاربرد در هندسه محاسباتی است، کاربردهایی در جغرافیای جمعیتی، فیزیک، نجوم، رباتیک و بسیاری از زمینه‌های دیگر دارد.

اگر فاصله‌ی بین دو نقطه‌ی p و q در صفحه با متر اقلیدسی سنجیده شود،

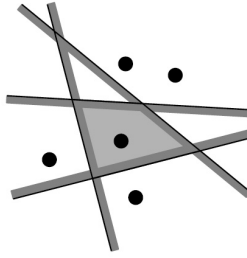
$$\|pq\| = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}.$$

همان‌طور که بیان شد به هر نقطه‌ی $p_i \in P$ ناحیه‌ای اختصاص داده می‌شود، این ناحیه، سلول p_i در نمودار ورنوی نامیده می‌شود و با $V(p_i)$ نشان داده می‌شود. برای رسم نمودار ورنوی P لازم است بین هر دو نقطه‌ی p_i ، خط منصف آن دو رسم شود و هر خط منصف در محل برخورد با سایر خطوط منصف، یک رأس نمودار را ایجاد می‌کند. بنابراین تمام یال‌ها در نمودار ورنوی، پاره‌خط یا نیم‌خط منصف بین دو نقطه هستند. مثلاً یال بین دو نقطه‌ی p_i و p_j برابر با مکان هندسی تمام نقاط a است که در تساوی $\|p_j a\| = \|p_i a\|$ صدق می‌کنند. بنابراین به راحتی دیده می‌شود که یک سلول $V(p_i)$ ناحیه‌ی حاصل از اشتراک تمام نیم‌صفحه‌هایی است که نقطه‌ی p_i را در بردارند. در شکل ۱۱.۱ می‌توانید نحوه‌ی ساخت یک سلول در نمودار ورنوی را ببینید.

در ادامه قضیه‌ی زیر مطرح خواهد شد که بیان می‌کند پیچیدگی نمودار ورنوی، خطی از تعداد نقاط مجموعه‌ی سازنده آن است.

قضیه ۱.۶.۱ ([۲۴]) برای $n \geq 3$ ، نمودار ورنوی یک مجموعه از n نقطه در صفحه، دارای حداکثر $2n - 5$ رأس و حداکثر $3n - 6$ یال است.

^{۲۰} Voronoi diagram



شکل ۱۱.۱: یک سلول در نمودار ورنوی

قبل از بیان قضیه‌ی زیر که اصلی‌ترین ویژگی نمودار ورنوی را مطرح می‌کند، لازم است مفهوم بزرگ‌ترین دایره‌ی خالی^{۲۱} تعریف شود. بزرگ‌ترین دایره‌ی خالی q نسبت به P ، بزرگ‌ترین دایره با مرکز q است که هیچ نقطه‌ای از P در داخل خود ندارد (دربر نمی‌گیرد) و با $Circ_P(q)$ نشان داده می‌شود.

قضیه ۲.۶.۱ ([۲۴]) برای نمودار ورنوی مجموعه نقاط P ، ویژگی‌های زیر برقرار است:

(الف) یک نقطه q یک رأس در $Vor(P)$ است اگر و تنها اگر بزرگ‌ترین دایره‌ی خالی‌اش $Circ_P(q)$ سه یا بیش‌تر نقطه از P را روی مرزش داشته باشد.

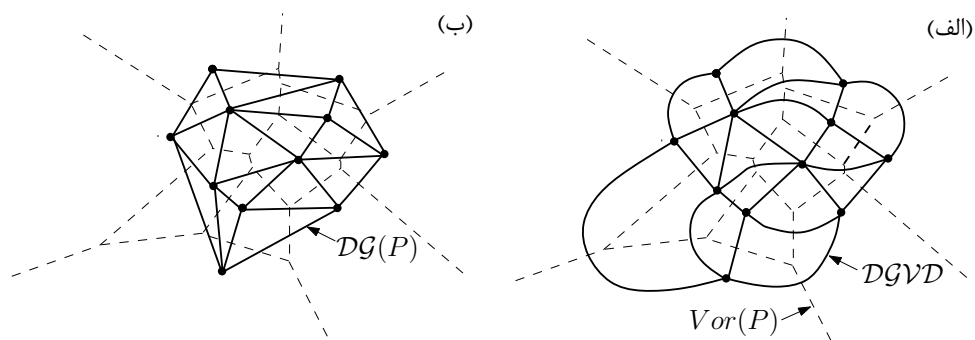
(ب) منصف بین p_i و p_j یک یال $Vor(P)$ را تعریف می‌کند اگر و تنها اگر یک نقطه‌ی r روی منصف وجود داشته باشد، به طوری که $Circ_P(r)$ تنها دو نقطه‌ی p_i و p_j را روی مرزش داشته باشد.

۷.۱ مثلث‌بندی دلونی

برای تعریف مثلث‌بندی دلونی لازم است گراف دوگان^{۲۲} نمودار ورنوی تعریف شود. اگر نمودار ورنوی $Vor(P)$ برای مجموعه نقاط P در نظر گرفته شود، گراف دوگان $DGVD$ به این صورت تعریف می‌شود: متناظر با هر سلول در $Vor(P)$ ، یک رأس در $DGVD$ وجود دارد. اگر بین دو سلول در $Vor(P)$ یک یال مشترک وجود داشته باشد، بین دو رأس متناظر با این دو سلول در $DGVD$ هم یک کمان وجود خواهد داشت (قسمت (الف) شکل ۱۲.۱). به عبارت دیگر $DGVD$ یک کمان برای هر یال $Vor(P)$ دارد و یک تناظر یک به یک بین وجه‌های محدود $Vor(P)$ و رأس‌های $DGVD$ وجود خواهد داشت. قسمت (الف) شکل ۱۲.۱ را ببینید.

^{۲۱}Largest empty circle

^{۲۲}Dual graph



شکل ۱۲.۱: (الف): گراف دوگان نمودار ورنوی و شکل (ب): گراف دلونی

اگر به جای رأس متناظر با سلول $V(p_i)$ در $DGVD$ ، نقطه‌ی p_i قرار گیرد و به جای کمان وصل‌کننده‌ی رأس‌های نظیر سلول‌های $V(p_i)$ و $V(p_j)$ در $DGVD$ ، پاره‌خط $\overline{p_i p_j}$ درج شود (قسمت (ب) شکل ۱۲.۱)، گراف حاصل، **گراف دلونی**^{۲۳} مجموعه نقاط P نامیده می‌شود و با $DG(P)$ نمایش داده می‌شود.

به راحتی دیده می‌شود که هر وجه f در $DG(P)$ متناظر با یک رأس v در $Vor(P)$ است. بنابراین هر یال f متناظر با یک یال مجاور به v است. به‌ویژه، اگر v مجاور به سلول‌های $V(p_1), V(p_2), \dots, V(p_k)$ باشد، آن‌گاه f نقاط p_1, p_2, \dots, p_k را به عنوان رأس‌هایش دارد. قضیه‌ی ۲.۶.۱ بیان می‌کند که در این وضعیت نقاط p_1, p_2, \dots, p_k (رئوس f) روی بزرگ‌ترین دایره‌ی خالی $Circ_P(v)$ واقع می‌شوند، یعنی تمام زوایای f کم‌تر از 180° خواهد شد و نتیجه می‌شود که f علاوه بر این که یک k -ضلعی است، محدب نیز است.

هر مثلث‌بندی حاصل از اضافه کردن تعدادی یال به $DG(P)$ ، **مثلث‌بندی دلونی**^{۲۴} مجموعه نقاط P نامیده می‌شود و با $DT(P)$ نمایش داده می‌شود. از آنجایی که وجه‌های گراف دلونی محدب هستند به دست آوردن این چنین مثلث‌بندی آسان است. مشاهده می‌شود، مثلث‌بندی دلونی مجموعه نقاط P منحصر به فرد است اگر و تنها اگر $DG(P)$ یک مثلث‌بندی باشد و این زمانی ممکن است که نقاط در وضعیت عمومی^{۲۵} باشند. با استناد به قسمت الف قضیه‌ی ۲.۶.۱، اگر هنگام تشخیص هر رأس q در $Vor(P)$ ، بزرگ‌ترین دایره‌ی خالی $Circ_P(q)$ دقیقاً سه نقطه روی مرز داشته باشد، گفته می‌شود که مجموعه نقاط P در **وضعیت عمومی** هستند. به راحتی دیده می‌شود که اگر مجموعه‌ی نقاط P در وضعیت عمومی باشند، آن‌گاه تمام رأس‌های $Vor(P)$ از درجه‌ی سه خواهند بود و در نتیجه تمام وجه‌های $DG(P)$ مثلث می‌شوند. این توجیه می‌کند

^{۲۳}Delaunay graph

^{۲۴}Delaunay Triangulation

^{۲۵}General Position

که چرا اغلب $DG(P)$ را مثلث‌بندی دلونی P می‌نامند. اگر مجموعه نقاط P به‌طور تصادفی انتخاب شوند احتمال این‌که در وضعیت عمومی باشند بسیار زیاد است. با توجه به این مطلب و برای راحتی کار در سراسر این پایان‌نامه فرض می‌شود که نقاط در وضعیت عمومی هستند.

در ادامه، قضیه ۲.۶.۱ در مورد نمودار ورنوی، این‌بار برای گراف دلونی بیان می‌شود. و از آن‌جایی که در این پایان‌نامه فرض شد، نقاط در وضعیت عمومی هستند، بنابراین گراف دلونی همان مثلث‌بندی دلونی می‌شود و این قضیه برای مثلث‌بندی دلونی نیز برقرار خواهد بود.

قضیه ۱.۷.۱ ([۲۴]) فرض کنید P یک مجموعه از نقاط در صفحه باشد.

الف) نقاط $p_i, p_j, p_k \in P$ رأس‌های یک وجه از گراف دلونی P هستند اگر و تنها اگر هیچ نقطه‌ای از P داخل دایره‌ی گذرنده از p_i, p_j و p_k نباشد.

ب) دو نقطه‌ی $p_i, p_j \in P$ یک یال گراف دلونی P را شکل می‌دهند اگر و تنها اگر دایره‌ی گذرنده از p_i و p_j وجود داشته باشد به‌طوری‌که هیچ نقطه‌ی دیگری از P را در برنگیرد.

قضیه ۱.۷.۱ در واقع به یک ویژگی اختصاصی در مثلث‌بندی دلونی که در قضیه‌ی زیر بیان شده است، اشاره می‌کند.

قضیه ۲.۷.۱ ([۲۴]) فرض کنید P یک مجموعه از نقاط در صفحه و T یک مثلث‌بندی از P باشد. مثلث‌بندی T یک مثلث‌بندی دلونی است اگر و تنها اگر هیچ نقطه‌ای از P در داخل دایره‌ی محیطی هر مثلث T نباشد. و آخرین قضیه که بسیار مهم است، بیان می‌کند که، هر مثلث‌بندی دلونی یک مثلث‌بندی مجاز است و برعکس. تعریف مثلث‌بندی مجاز را می‌توانید در زیربخش ۱.۴.۱ ببینید. این ویژگی نقش مهمی در ساخت مثلث‌بندی دلونی و نگهداری درست آن، زمانی که نقاط حرکت می‌کنند، دارد.

قضیه ۳.۷.۱ ([۲۴]) فرض کنید P یک مجموعه از نقاط در صفحه باشد. یک مثلث‌بندی T از P یک مثلث‌بندی مجاز است اگر و تنها اگر T یک مثلث‌بندی دلونی از P باشد.

در پایان این بخش لازم است این تعاریف نیز بیان شوند: در یک مثلث‌بندی دلونی از مجموعه نقاط P ، رأس **غشای**^{۲۶}، رأسی است که روی غشای مثلث‌بندی واقع شده است و مجاور به وجه نامحدود است. همچنین دو نوع یال در مثلث‌بندی دلونی وجود دارد: یال **غشای**^{۲۷} و یال داخلی^{۲۸}. به یال‌هایی که مرز مشترک بین مثلث‌بندی و

^{۲۶}Hull vertex

^{۲۷}Hull edge

وجه نامحدود است، **یال غشا** و به یال‌های باقیمانده که در داخل مثلث‌بندی و مجاور به دو مثلث هستند، **یال داخلی** گفته می‌شود. به عبارتی دیگر هر یال داخلی، یک یال پایه برای مجاور شدن دو مثلث است.

۸.۱ L_p -متریک‌ها

برای دو نقطه‌ی $u = (u_1, \dots, u_d)$ و $v = (v_1, \dots, v_d)$ متعلق به فضای \mathbb{R}^d ، فاصله‌ی L_p -متریک بین u و v به این صورت تعریف می‌شود:

$$\delta_p(u, v) = \left(\sum_{i=1}^d |u_i - v_i|^p \right)^{1/p}.$$

به راحتی دیده می‌شود که L_2 -متریک همان فاصله‌ی اقلیدسی شناخته شده می‌باشد. در ادامه برای راحتی کار L_p -متریک‌ها در صفحه در نظر گرفته می‌شوند. بنابراین، برای دو نقطه‌ی $u, v \in \mathbb{R}^2$ ، فاصله‌ی L_p -متریک بین u و v به این صورت تعریف می‌شود:

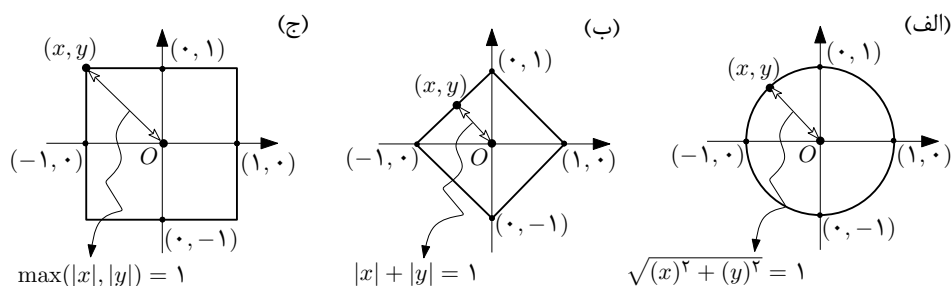
$$\delta_p(u, v) = (|u_1 - v_1|^p + |u_2 - v_2|^p)^{1/p}.$$

لازم به ذکر است که فاصله‌ی L_∞ -متریک بین u و v نیز به طور متفاوت و به این شکل تعریف می‌شود:

$$\delta_\infty(u, v) = \max(|u_1 - v_1|, |u_2 - v_2|).$$

برای معرفی هر L_p -متریک یک شکل واحد وجود دارد. مبدأ مختصات را در نظر بگیرید، مکان هندسی تمام نقاطی که دارای فاصله‌ی L_p -متریک یک از مبدأ هستند، یک شکل هندسی واحد را برای آن متر تعریف می‌کنند. مثال شناخته شده‌ی آن شکل واحد دایره به مرکز مبدأ در متر اقلیدسی (L_2 -متریک) است که فاصله‌ی اقلیدسی تمام نقاط روی مرز دایره تا مبدأ برابر با یک است. شکل ۱۳.۱ قسمت (الف) را ببینید. به راحتی برای سایر L_p -متریک‌ها نیز این شکل واحد به دست می‌آید که در اصطلاح به عنوان دایره‌ی هر L_p -متریک شناخته می‌شود (چون متر اقلیدسی از سایر مترها شناخته شده‌تر است و شکل واحد آن دایره است، از این اصطلاح در سایر مترها نیز استفاده می‌شود). به عنوان نمونه دایره‌ی L_1 -متریک برابر با مکان هندسی تمام نقاط $u = (x, y)$ است که دارای فاصله‌ی L_1 -متریک یک از مبدأ هستند، یعنی $|x| + |y| = 1$ ، که به شکل یک مربع واحد ۴۵ درجه چرخیده شده است. شکل ۱۳.۱ قسمت (ب) را ببینید. نمونه‌ی دیگر دایره‌ی L_∞ -متریک است که یک مربع (بدون چرخش) است. شکل ۱۳.۱ قسمت (ج) را ببینید.

^{۲۸}Interior edge



شکل ۹.۱: (الف): دایره‌ی متر اقلیدسی، شکل (ب): دایره‌ی متر L_1 و شکل (ج): دایره‌ی متر L_∞

۹.۱ نمودار ورنوی و مثلث‌بندی دلونی، بر اساس یک تابع فاصله‌ی

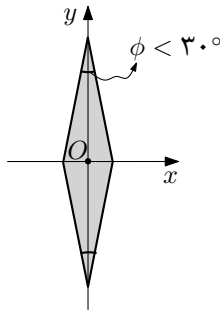
محدب

یک دیدگاه برای ساخت نمودار ورنوی که در [۱۹] معرفی شده است، دیدگاه گسترش موج‌ها^{۲۹} است. در این دیدگاه نمودار ورنوی را می‌توان برای مترهای دیگر (علاوه بر متر اقلیدسی)، و نیز برای توابع فاصله‌ی محدب تعریف کرد. اگر یک سنگ‌ریزه به داخل یک دریاچه‌ی راکد انداخته شود، موج‌های دایره‌ای از نقطه برخورد سنگ‌ریزه با آب به اطراف گسترش می‌یابند. اگر n سنگ‌ریزه به‌طور هم‌زمان به داخل دریاچه انداخته شوند، مکان‌هایی که موج‌های حاصل از آن‌ها، درحین گسترش، با یکدیگر برخورد می‌کنند نمودار ورنوی را روی n نقطه‌ی برخورد سنگ‌ریزه‌ها با آب تعریف می‌کند.

نمودار ورنوی برای هر متر استاندارد، از جمله مترهای L_p ، می‌تواند با تغییر شکل موج از شکل دایره (در متر اقلیدسی)، به شکل دایره در آن متر به‌دست آید. برای توضیحات بیش‌تر در مورد L_p -متریک‌ها و دایره‌ی نظیر آن‌ها بخش ۸.۱ را ببینید. همچنین نمودار ورنوی برای توابع فاصله‌ی محدب نیز می‌تواند با تغییر شکل موج به شکل محدبی که تابع فاصله‌ی محدب بر اساس آن تعریف شده است به‌دست می‌آید. اگر شکل محدب نسبت به مرکز متقارن نباشد (برای مثال، یک مثلث) هرچند که تابع فاصله‌ی محدب حاصل یک متر نخواهد بود، اما باز هم می‌تواند برای تعریف یک نمودار ورنوی استفاده شود. برای توضیحات بیش‌تر در مورد تابع فاصله‌ی محدب بخش ۵.۱ را ببینید.

شبه نمودار ورنوی بر اساس متر اقلیدسی، نمودار ورنوی بر اساس سایر مترها نیز می‌تواند برای حل مسائل مختلف نزدیک‌ترین جفت (تمام نزدیک‌ترین همسایه‌ها، درخت‌های پوشای کمینه و غیره) استفاده شود. تعدادی از این مسائل، برای توابع فاصله‌ی محدبی که لزوماً متر نیستند نیز وجود دارند. به‌ویژه، مسائل بزرگ‌ترین دایره‌ی

^{۲۹}Expanding waves



شکل ۱۴.۱: لوزی باریک عمودی

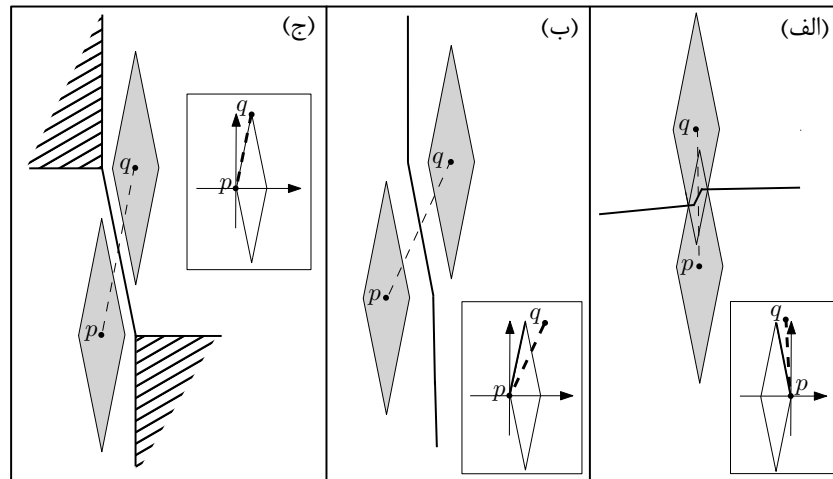
خالی که تبدیل می‌شوند به مسائل بزرگ‌ترین شکل محدب خالی و مسائل مسیریابی برای یک دیسک که به مسائل مسیریابی برای یک شکل محدب تبدیل می‌شوند.

در فصل ۴ نیاز است تا مثلث‌بندی دلونی، بر اساس یک تابع فاصله‌ی محدب، که یک لوزی باریک و متقارن نسبت به مرکز، شکل محدب آن است، تعریف شود. بنابراین مناسب است که با این تابع فاصله‌ی محدب، که یک متر نیز هست (بخش ۵.۱ را ببینید)، مفاهیم مربوط به ساخت نمودار ورنوی و مثلث‌بندی دلونی بر اساس یک تابع فاصله‌ی محدب، شرح داده شود. بدین منظور ابتدا منصف، بر اساس این تابع فاصله‌ی محدب، برای ساخت نمودار ورنوی معرفی می‌شود. سپس بر اساس آن منصف، نمودار ورنوی و مثلث‌بندی دلونی حاصل از آن معرفی می‌شود.

۱.۹.۱ تابع فاصله‌ی محدب لوزی \diamond و منصف آن

شکل محدب مورد نظر، که تابع فاصله بر اساس آن تعریف می‌شود، لوزی باریک \diamond است به طوری که مرکز آن در مبدأ واقع شده است و قطرهای آن موازی با محورهای x و y هستند. قطر اصلی (بلندتر)، به طول ۲ واحد و موازی با محور y است. زاویه‌ی رأس‌های بالایی و پایینی آن ϕ است و کم‌تر از 30° است. شکل ۱۴.۱ را ببینید. بنابراین زاویه‌ای که قطر اصلی لوزی با محور x می‌سازد 90° است. هرچند که، همین لوزی در حالتی که قطر اصلی آن زاویه‌ی دیگری با محور x بسازد (لوزی چرخیده شود)، می‌تواند تابع فاصله‌ی محدب متناظرش را تعریف کند، به طوری که تمام ویژگی‌های مثلث‌بندی دلونی حاصل از آن، مشابه با ویژگی‌های مثلث‌بندی دلونی حاصل از تابع فاصله‌ی لوزی اولیه (عمودی) باشد.

تابع فاصله‌ی شکل محدب \diamond که با نماد d_\diamond نشان داده می‌شود را در نظر بگیرید، منصف دو نقطه‌ی p و q بر اساس این تابع فاصله، طبق مطالب ذکر شده در بخش ۵.۱ به راحتی رسم می‌شود و با توجه به موقعیت دو



شکل ۱۵.۱: شکل‌های متفاوت منصف در متر d

نقطه‌ی p و q نسبت به هم، شکل‌های مختلفی برای آن به دست می‌آید، که می‌توان این شکل‌ها را بر اساس مطالب زیر دسته‌بندی کرد.

پاره خط \overline{pq} را در نظر بگیرید، فرض کنید نقطه‌ی پایینی این پاره خط p باشد، \overline{pq} به گونه‌ای انتقال داده می‌شود که p روی مبدأ مختصات قرار گیرد. \overline{pq} در یکی از این دو وضعیت می‌تواند قرار گیرد: یا در ربع اول یا در ربع دوم صفحه‌ی مختصات. اگر در ربع اول واقع شد، شیب آن با شیب ضلع بالا و سمت چپ \diamond ، و اگر در ربع دوم واقع شد، شیب آن با شیب ضلع بالا و سمت راست \diamond مقایسه می‌شود. شکل ۱۵.۱ را ببینید. نتیجه‌ی مقایسه‌ی شیب پاره خط \overline{pq} و شیب ضلع لوزی متناظر با آن، یکی از سه حالت زیر است:

(الف) شیب پاره خط \overline{pq} بیش‌تر از شیب ضلع لوزی \diamond

(ب) شیب پاره خط \overline{pq} کم‌تر از شیب ضلع لوزی \diamond

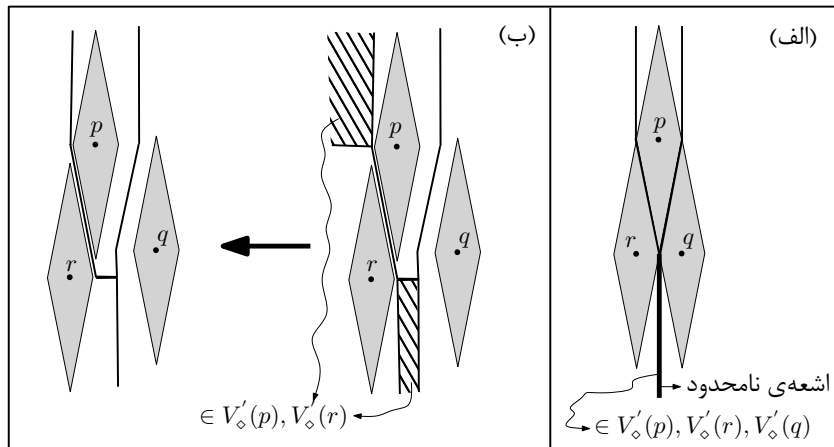
(ج) شیب پاره خط \overline{pq} مساوی با شیب ضلع لوزی \diamond .

منصف‌ها در این سه حالت به ترتیب به سه شکل متفاوت زیر به دست می‌آیند (شکل ۱۵.۱):

(الف) منصف شبیه به یک خط افقی (شامل یک پاره خط و دو خط تقریباً افقی)

(ب) منصف شبیه به یک خط عمودی (شامل یک پاره خط و دو خط تقریباً عمودی)

(ج) منصف شامل یک پاره خط و دو ناحیه‌ی نامحدود.



شکل ۱۶.۱: دو حالت خاص برای نمودار ورنوی بر اساس متر d_{\diamond}

۲.۹.۱ نمودار ورنوی بر اساس d_{\diamond}

اگر P مجموعه‌ای از نقاط در صفحه باشد، نمودار ورنوی بر اساس تابع فاصله‌ی محدب d_{\diamond} ، با معرفی سلول نظیر هر نقطه‌ی $p \in P$ ، تعریف می‌شود.

$$V'_{\diamond}(p) = \{x \in \mathbb{R}^2 : \forall q \in P, d_{\diamond}(x, p) \leq d_{\diamond}(x, q)\}.$$

این تعریف برای نمودار ورنوی، بر اساس یک تابع فاصله‌ی محدب، در دو مورد مشکل ایجاد می‌کند: مورد اول، زمانی است که سلول $V'_{\diamond}(p)$ می‌تواند شامل یک ناحیه‌ی بسته و یک اشعه‌ی نامحدود متصل به آن باشد (در این وضعیت، آن اشعه متعلق به سلول‌های دیگر نیز خواهد بود). برای مثال، سلول $V'_{\diamond}(p)$ در شکل ۱۶.۱ قسمت (الف) این مشکل را دارد. مورد دوم، زمانی است که منصف دو نقطه‌ی p و r بر اساس d_{\diamond} ، می‌تواند شامل دو ناحیه‌ی نامحدود باشد. در این وضعیت، نقاط واقع در این ناحیه‌های نامحدود، هم متعلق به سلول نظیر p و هم متعلق به سلول نظیر r خواهند بود. برای مثال، شکل ۱۶.۱ قسمت (ب) را ببینید. در این دو مورد، نمودار ورنوی لزوماً یک تقسیم‌بندی از صفحه، به طوری که هر یال، مرز مشترک بین تنها دو سلول ورنوی باشد و نیز داخل هر سلول، متناظر با تنها یک نقطه‌ی یکتا از P باشد، نخواهد بود.

برای رفع این مشکلات، از دیدگاه مطرح شده توسط کلین^{۳۰} و وود^{۳۱} [۳۳] (همچنین [۱۱، ۳۵] را ببینید) استفاده می‌شود: برای مورد اول، اشعه‌ی نامحدود دیگر متعلق به سلول $V'_{\diamond}(p)$ نخواهد بود، بلکه جزء ناحیه‌ی سلول‌های دیگر محسوب می‌شود. برای مثال، در شکل ۱۶.۱ قسمت (الف)، اشعه‌ی نامحدود جزء سلول $V'_{\diamond}(p)$

^{۳۰} Klein

^{۳۱} Wood

نخواهد بود، بلکه یال بین دو سلول $V'_\diamond(r)$ و $V'_\diamond(q)$ محسوب خواهد شد. برای مورد دوم، یک نقطه‌ی $x \in \mathbb{R}^2$ که در داخل بیش از یک سلول $V'_\diamond(p)$ قرار دارد، تنها به سلولی که نقطه‌ی متناظرش از لحاظ ترتیب الفبایی^{۳۲} کوچک‌ترین است منتسب می‌شود (تعریف ترتیب الفبایی در پاراگراف بعدی بیان شده است). در شکل ۱۶.۱ قسمت (ب)، نقطه‌ی r بر اساس این ترتیب کوچک‌تر بوده است و نقاط ناحیه‌ی نامحدود به سلول آن تعلق گرفته است. با اعمال این دیدگاه روی تعریف سلول‌های نمودار ورنوی که از این به بعد برای هر نقطه‌ی $p \in P$ با نماد $V_\diamond(p)$ نشان داده می‌شوند، نمودار ورنوی مجموعه نقاط P بر اساس d_\diamond که با $Vor_\diamond(P)$ نشان داده می‌شود، یک تقسیم‌بندی مناسب از صفحه خواهد شد، به طوری که هر نقطه از صفحه در یکی از این سه وضعیت قرار خواهد داشت: الف) متعلق به داخل یک سلول ورنوی یکتا، ب) متعلق به یک یال ورنوی یکتا، ج) متعلق به یک رأس ورنوی یکتا.

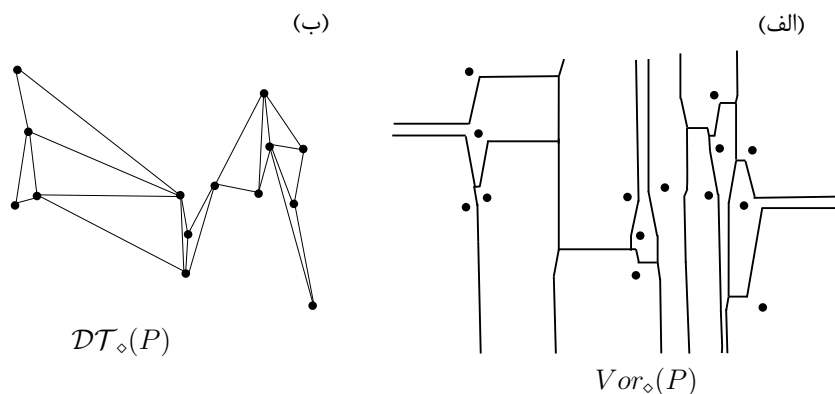
تعریف ۱.۹.۱ دو نقطه‌ی $p_i = (x_i, y_i)$ و $p_j = (x_j, y_j)$ را در نظر بگیرید. ترتیب الفبایی p_i کم‌تر از p_j است، اگر و تنها اگر مولفه‌های x و y این نقاط یکی از این شرایط را داشته باشند: $x_i < x_j$ یا اگر $x_i = x_j$ باشد، آنگاه $y_i < y_j$.

۳.۹.۱ مثلث‌بندی دلونی بر اساس d_\diamond و ویژگی‌های آن

از آنجایی که در این پایان‌نامه فرض شده است، نقاط در وضعیت عمومی هستند، بنابراین گراف دلونی حاصل از گراف دوگان $Vor_\diamond(P)$ ، یک مثلث‌بندی دلونی یکتا از مجموعه نقاط P خواهد بود و با نماد $DT_\diamond(P)$ نشان داده می‌شود. شکل ۱۷.۱ نمودار ورنوی و مثلث‌بندی دلونی یک نمونه از نقاط را نشان می‌دهد.

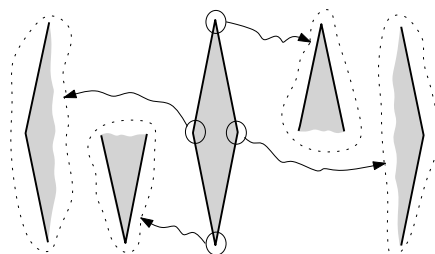
همان‌طور که بیان شد، چون \diamond نسبت به مرکزش متقارن است، پس تابع فاصله‌ی محدب d_\diamond ، طبق مطالب بخش ۵.۱ خاصیت تقارنی را نیز دارد. در نتیجه d_\diamond ، یک متر است. بنابراین ویژگی‌هایی که در بخش‌های ۶.۱ و ۷.۱، به ترتیب برای نمودار ورنوی و مثلث‌بندی دلونی در متر اقلیدسی بیان شدند، قابل تعمیم به نمودار ورنوی و مثلث‌بندی دلونی در متر d_\diamond نیز هستند. کافی است در بیان این ویژگی‌ها به جای دایره‌ی متر اقلیدسی، دایره‌ی متر d_\diamond که لوزی \diamond است، جایگزین شود. در جدول ۱-۱ می‌توانید تعدادی از این ویژگی‌ها را که در دو متر اقلیدسی و d_\diamond در مقایسه‌ی باهم آورده شده است، ببینید. در این جدول نماد \mathcal{C}_∞ ، به یک دایره با مرکز بی‌نهایت و نماد \diamond_∞ ، به یک لوزی \diamond با مرکز بی‌نهایت اشاره می‌کند.

^{۳۲}Lexicographically order



شکل ۱۷.۱: شکل‌های (الف) و (ب) به ترتیب: نمودار ورنوی و مثلث‌بندی دلونی بر اساس تابع فاصله‌ی محدب

d_\diamond



شکل ۱۸.۱: چهار \diamond -مخروط متفاوت برای یک لوزی \diamond

با گسترش دو ضلع مجاور به هر رأس v از یک لوزی \diamond ، به دو نیم خط، یک مخروط با رأس v تعریف می‌شود. چنین مخروطی \diamond -مخروط نامیده می‌شود. شکل ۱۸.۱ را ببینید. بنابراین از هر لوزی \diamond ، چهار \diamond -مخروط متفاوت قابل تعریف است. \diamond -مخروط‌ها در حقیقت همان \diamond_∞ هستند. شکل ۱۸.۱ را ببینید. تمام ویژگی‌های بیان شده در جدول قابل اثبات هستند. اثبات ویژگی‌های ۱ و ۲ در متر d_\diamond ساده و شبیه به اثبات همین ویژگی‌ها در متر اقلیدسی است ([۲۴]، فصل ۷ و ۹). ویژگی ۳ و ۴ در ادامه اثبات می‌شوند. ویژگی ۵ نیز با استفاده از یک مثال، به راحتی نشان داده می‌شود.

لم ۲.۹.۱ دو نقطه‌ی $p, q \in P$ یک یال غشا در $DT_\diamond(P)$ را شکل می‌دهند اگر و تنها اگر یک \diamond -مخروط با p و q روی مرزش وجود داشته باشد، به طوری که هیچ نقطه‌ی دیگری از P را دربر نگیرد.

اثبات. فرض کنید که یک \diamond -مخروط با این شرایط وجود داشته باشد. این مخروط در واقع همان لوزی \diamond به مرکز بی‌نهایت است. بنابراین مرکز این لوزی نقطه‌ی o در بی‌نهایت فرض می‌شود. با توجه به تابع فاصله‌ی محدب لوزی \diamond (لوزی \diamond ، دایره‌ی متر d_\diamond است)، نتیجه می‌شود که $d_\diamond(o, p) = d_\diamond(o, q)$. این بدین معنی

خواص DT در متر d_\diamond	خواص DT در متر اقلیدسی	
نقاط $p, q, r \in P$ رأس‌های یک مثلث هستند اگر و تنها اگر لوزی \diamond گذرنده از p, q, r هیچ نقطه‌ای از P را دربر نگیرد.	نقاط $p, q, r \in P$ رأس‌های یک مثلث هستند اگر و تنها اگر دایره‌ی گذرنده از p, q, r هیچ نقطه‌ای از P را دربر نگیرد.	۱
دو نقطه‌ی $p, q \in P$ یک یال را شکل می‌دهند اگر و تنها اگر یک لوزی \diamond گذرنده از p و q وجود داشته باشد، به طوری که هیچ نقطه‌ی دیگری از P را در بر نگیرد.	دو نقطه‌ی $p, q \in P$ یک یال را شکل می‌دهند اگر و تنها اگر دایره‌ای گذرنده از p و q وجود داشته باشد، به طوری که هیچ نقطه‌ی دیگری از P را در بر نگیرد.	۲
دو نقطه‌ی $p, q \in P$ یک یال غشا را شکل می‌دهند اگر و تنها اگر یک \diamond -مخروط با p و q روی مرز وجود داشته باشد، به طوری که هیچ نقطه‌ی دیگری از P را دربر نگیرد (p و q روی مرز \diamond_∞ خالی).	دو نقطه‌ی $p, q \in P$ یک یال غشا را شکل می‌دهند اگر و تنها اگر تمام نقاط، در یک طرف خط pq باشند (p و q ، روی مرز C_∞ خالی).	۳
نقطه‌ی p یک رأس غشاست اگر و تنها اگر یک \diamond -مخروط با رأس p وجود داشته باشد، به طوری که هیچ نقطه‌ی دیگری از P را دربر نگیرد. (p رأس \diamond_∞ خالی).	نقطه‌ی p یک رأس غشاست اگر و تنها اگر خطی گذرنده از p باشد که تمام نقاط در یک طرفش باشند (p روی مرز C_∞ خالی).	۴
غشای مثلث‌بندی می‌تواند غیرمحدب باشد.	غشای مثلث‌بندی محدب است.	۵

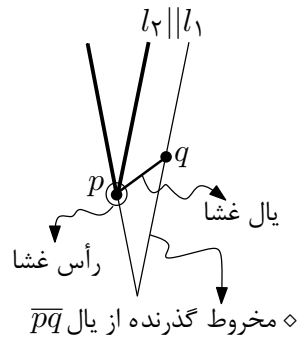
جدول ۱-۱: مقایسه‌ی ویژگی‌های مثلث‌بندی دلونی در مترهای اقلیدسی و d_\diamond

است که نقطه‌ی o روی منصف بین p و q در $Vor_{\diamond}(P)$ قرار دارد و هر دو نقطه‌ی p و q نزدیک‌ترین نقاط از مجموعه‌ی P به آن هستند. بنابراین نقطه‌ی o روی یال بین دو سلول $V_{\diamond}(p)$ و $V_{\diamond}(q)$ واقع می‌شود. به دلیل پیوستگی تابع فاصله‌ی d_{\diamond} ، همیشه می‌توان یک بخشی از منصف بین p و q که نقطه‌ی o را نیز شامل می‌شود یافت، به طوری که هر نقطه از این بخش منصف تنها بین $V_{\diamond}(p)$ و $V_{\diamond}(q)$ مشترک باشد و از آن جایی که o در بی‌نهایت است، این بخش از منصف نامحدود می‌شود. در نتیجه یک یال نامحدود بین $V_{\diamond}(p)$ و $V_{\diamond}(q)$ در نمودار $Vor_{\diamond}(P)$ وجود دارد و بنابراین دو نقطه‌ی p, q یک یال غشا در $DT_{\diamond}(P)$ را شکل می‌دهند. برای اثبات برگشت قضیه، فرض می‌شود که دو نقطه‌ی $p, q \in P$ یک یال غشا در $DT_{\diamond}(P)$ را شکل می‌دهند. بنابراین یک یال نامحدود متناظر با آن در نمودار $Vor_{\diamond}(P)$ بین دو سلول $V_{\diamond}(p)$ و $V_{\diamond}(q)$ وجود دارد. هر نقطه‌ی o از این یال، به جز نقطه‌ی انتهایی آن را که در نظر بگیرید، از آن جایی که p و q نزدیک‌ترین نقاط از P به آن هستند، لوزی \diamond به مرکز o به طوری که از p و q بگذرد، شامل هیچ نقطه‌ی دیگری از P در داخل یا روی مرزش نخواهد بود و از آن جایی که o می‌تواند نقطه‌ای در بی‌نهایت باشد، این لوزی می‌تواند یک \diamond -مخروط نیز باشد. \square

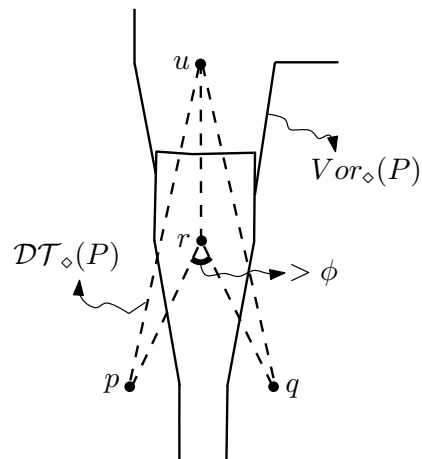
لم ۳.۹.۱ نقطه‌ی p یک رأس غشا در $DT_{\diamond}(P)$ است اگر و تنها اگر یک \diamond -مخروط با رأس p وجود داشته باشد، به طوری که هیچ نقطه‌ی دیگری از P را دربر نگیرد.

اثبات. فرض کنید که یک \diamond -مخروط با این شرایط وجود داشته باشد. بنابراین سلول $V_{\diamond}(p)$ نظیر p در نمودار $Vor_{\diamond}(P)$ یک سلول نامحدود است و در نتیجه p یک رأس غشا در $DT_{\diamond}(P)$ است. برای اثبات برگشت قضیه، فرض می‌شود که p یک رأس غشا در $DT_{\diamond}(P)$ است. از آن جایی که هر رأس غشا نقطه‌ی انتهایی یک یال غشاست، فرض کنید p نیز نقطه‌ی انتهایی یال غشای \overline{pq} باشد. طبق لم ۲.۹.۱، یک \diamond -مخروط با p و q روی مرزش وجود دارد، به طوری که هیچ نقطه‌ی دیگری از P را دربر نمی‌گیرد. حال اگر نیم‌خطی از p موازی با مرزی از این \diamond -مخروط که از نقطه‌ی q گذشته است رسم شود (شکل ۱۹.۱ را ببینید)، یک \diamond -مخروط با رأس p به دست می‌آید که زیرمجموعه‌ای از \diamond -مخروط گذرنده از یال \overline{pq} است و بنابراین هیچ نقطه‌ای از P را نیز دربر نمی‌گیرد. \square

نتیجه ۴.۹.۱ غشای مثلث‌بندی $DT_{\diamond}(P)$ می‌تواند غیرمحدب باشد.



شکل ۱۹.۱: شکل مربوط به اثبات لم ۳.۹.۱



شکل ۲۰.۱: یک مثلث بندی دلونی $DT_{\diamond}(P)$ با غشای غیرمحدب

اثبات. این نتیجه را می‌توان با یک مثال اثبات کرد. شکل ۲۰.۱ را ببینید. فرض کنید P یک مجموعه از چهار نقطه‌ی p, u, r, q در صفحه باشد و این نقاط به‌گونه‌ای قرار گرفته باشند که u, r در یک راستای عمودی و p, q در یک راستای افقی باشند و مثلث فرضی puq غشای محدب این مجموعه از نقاط باشد. یعنی نقطه‌ی r در داخل این مثلث فرضی واقع شود. علاوه بر این، زاویه‌ی بین پاره‌خط‌های \overline{rp} و \overline{rq} نیز بیش‌تر از ϕ باشد. شکل ۲۰.۱ را ببینید. در این وضعیت، طبق مطالبی که در زیربخش ۱.۹.۱ بیان شد، منصف بین r, p و نیز منصف بین r, q ، هر دو شبیه به خط عمودی می‌شوند (شکل ۲۰.۱). دو خط عمودی که در بی‌نهایت موازی می‌شوند و هرگز یکدیگر را قطع نخواهند کرد. بنابراین سلول نظیر نقطه‌ی r در نمودار ورنوی $Vor_{\diamond}(P)$ یک سلول نامحدود می‌شود که بین دو نقطه‌ی p و q قرار دارد. در نتیجه، به دلیل وجود این ناحیه نامحدود، منصف مشترکی بین p و q در $Vor_{\diamond}(P)$ وجود نخواهد داشت و پاره‌خط \overline{pq} یک یال در $DT_{\diamond}(P)$ نخواهد بود (پاره‌خط \overline{pq} یک یال غشای محدب این مجموعه از نقاط است). در حالی که پاره‌خط‌های \overline{rp} و \overline{rq} دو یال در $DT_{\diamond}(P)$ هستند (شکل ۲۰.۱). این مثال نشان داد که غشای $DT_{\diamond}(P)$ می‌تواند غیرمحدب باشد. \square

۱۰.۱ پوشش‌های هندسی روی مجموعه نقاط

۱.۱۰.۱ شبکه‌های هندسی

مجموعه‌ی P شامل n نقطه در فضای \mathbb{R}^d را در نظر بگیرید، یک شبکه‌ی متصل‌کننده‌ی نقاط P ^{۳۳}، یک گراف $\mathcal{G} = (P, E)$ با مجموعه رأس‌های P و مجموعه یال‌های $E \subseteq P \times P$ است، به طوری که هر دو نقطه‌ی $p, q \in P$ با یک مسیر در \mathcal{G} به هم متصل می‌شوند. یک شبکه‌ی هندسی^{۳۴} یا یک گراف اقلیدسی^{۳۵}، یک گراف وزن‌دار \mathcal{G} است که رأس‌ها متناظر با نقاط در فضای اقلیدسی و وزن روی یال‌ها متناظر با فاصله‌ی اقلیدسی بین نقاط انتهایی آن یال است. شبکه‌های هندسی در واقع تعداد زیادی از شبکه‌های حقیقی موجود، مانند شبکه راه‌ها، شبکه مخابرات و غیره را مدل می‌کنند.

برای طراحی یک شبکه برای مجموعه‌ی مشخصی از نقاط، چندین معیار کیفی در نظر گرفته می‌شود. در

^{۳۳}A network connecting the points of P

^{۳۴}Geometric network

^{۳۵}Euclidean graph

زیر تعدادی از مهم‌ترین معیارهای کیفی برای ارزیابی شبکه‌های هندسی بیان شده است.

۱. اندازه^{۳۶}، به‌عنوان تعداد یال‌های شبکه تعریف می‌شود. در حالت کلی ترجیح داده می‌شود که شبکه‌ها تا جای ممکن اندازه‌ی کوچکی (خطی از تعداد نقاط) داشته باشند.

۲. وزن^{۳۷}، به‌عنوان مجموع وزن یال‌های شبکه تعریف می‌شود. از آنجایی که هر شبکه باید تمام نقاط را به هم وصل کند، در نتیجه وزن آن از پایین با وزن درخت پوشای کمینه کران‌دار می‌شود. وزن یک معیار خوب برای سنجش هزینه‌ی ساخت شبکه است. بنابراین، اغلب شبکه‌هایی با وزن کم مورد نظر هستند.

۳. ضریب کشش^{۳۸} یا تاخیر^{۳۹} برای دو نقطه‌ی داده شده، برابر با نسبت کوتاه‌ترین مسیر (مسیر با وزن مینیمم) بین دو نقطه در شبکه، به فاصله‌ی آن دو نقطه بر اساس متر تعریف شده برای آن شبکه است (مثلاً این فاصله در متر اقلیدسی خط مستقیم متصل کننده‌ی دو نقطه است). ضریب کشش یک شبکه به‌عنوان بیش‌ترین ضریب کشش برای هر جفت از نقاط مجزا در شبکه تعریف می‌شود. در بسیاری از موارد، نیاز است که ضریب کشش شبکه با یک ثابت کوچک محدود شود (که حداقل باید یک باشد). شبکه‌ها با ضریب کشش حداکثر t ، t -پوشش‌ها^{۴۰} نامیده می‌شوند.

۴. درجه^{۴۱}، بیش‌ترین تعداد یال‌های مجاور به هر نقطه در شبکه می‌باشد و اغلب نیاز است که با یک ثابت کوچک محدود شود. درجه‌ی محدود یک شبکه، به اندازه‌ی کوچک آن شبکه اشاره می‌کند، اما برعکس این مطلب لزوماً درست نیست.

در حالت کلی، در زمان طراحی یک شبکه، آن‌چه که اهمیت زیادی دارد، اعمال ترکیبی از این معیارهای کیفی بر روی شبکه است و در زمان تحلیل شبکه نیز ویژگی‌های شبکه، نسبت به این معیارها سنجیده می‌شود. یکی از مسائل مهم در این زمینه، مطالعه‌ی شبکه‌هایی با ضریب کشش کم است (در ترکیب با دیگر ویژگی‌ها). از جمله، در بسیاری از کاربردها مانند شبکه‌ی راه‌ها لازم است یک ارتباط سریع (مستقیم) بین هر جفت از

^{۳۶}Size

^{۳۷}Weight

^{۳۸}Stretch factor

^{۳۹}Dilation

^{۴۰}t-Spanners

^{۴۱}Degree

نقاط در P برقرار باشد (یعنی شبکه یک گراف کامل باشد) ولی این نیاز به خاطر هزینه‌های بالا، قابل اجرا شدن نیست. بنابراین نیاز به مطالعه‌ی شبکه‌هایی با ضریب کشش کم، منجر به شکل‌گیری مفهوم پوشش‌های هندسی می‌شود. این پوشش‌ها در واقع یک ساختار برای شبکه‌ها، زمانی که ارتباطات کوتاه بین نقاط اهمیت دارند را فراهم می‌کنند.

۲.۱۰.۱ t -پوشش‌های هندسی

تعریف ۱.۱۰.۱ مجموعه‌ی P شامل n نقطه در فضای \mathbb{R}^d و $t \geq 1$ را یک عدد حقیقی در نظر بگیرید. یک t -پوشش^{۴۲} برای P ، یک گراف بدون جهت \mathcal{G} با مجموعه رأس‌های P است، به طوری که کوتاه‌ترین مسیر بین هر دو نقطه‌ی p و q از P ، در \mathcal{G} که با نماد $d_{\mathcal{G}}(p, q)$ نشان داده می‌شود، این شرط را داشته باشد:

$$d_{\mathcal{G}}(p, q) \leq t \cdot \|pq\|.$$

هر مسیری که این شرط را برآورده سازد یک t -مسیر^{۴۳} بین p و q نامیده می‌شود.

برای هر عدد حقیقی $t' > t$ که $t' > t$ است، اگر \mathcal{G} یک t -پوشش برای مجموعه نقاط P باشد، بدیهی است که \mathcal{G} یک t' -پوشش نیز برای P است. این، منجر به تعریف زیر می‌شود:

تعریف ۲.۱۰.۱ (ضریب کشش) مجموعه‌ی P شامل n نقطه در فضای \mathbb{R}^d و \mathcal{G} را یک گراف اقلیدسی با مجموعه رأس‌های P در نظر بگیرید. ضریب کشش \mathcal{G} ، کوچک‌ترین عدد حقیقی t است به طوری که \mathcal{G} یک t -پوشش از P باشد.

گراف کامل یک 1 -پوشش است، اما تعداد مربعی یال دارد. درحقیقت، اگر فرض شود که هیچ سه نقطه‌ای از P روی یک خط قرار ندارند، سپس گراف کامل تنها 1 -پوشش برای P است. بنابراین، در حالت کلی t -پوشش‌ها برای t های بزرگ‌تر از یک بررسی می‌شوند. اما از آنجایی که در بسیاری از کاربردها، پوشش‌هایی با ارتباطات سریع (ضریب کشش کم) مورد نیاز هستند. بنابراین پوشش‌هایی با ضریب کشش نزدیک به یک، مورد مطالعه قرار می‌گیرند، یعنی $t = 1 + \varepsilon$ ، برای مقادیر ε مثبت و کوچک ($0 < \varepsilon < 1$). به این ترتیب، می‌توان با انتخاب مقادیر خیلی کوچک برای ε ، مقدار t را به یک نزدیک کرد، یعنی t -پوشش را به یک 1 -پوشش (گراف کامل) نزدیک کرد.

^{۴۲} t -spanner

^{۴۳} t -path

فصل ۲

ساختارهای وابسته به حرکت

برای نگهداری مشخصه‌ی مورد نظر در یک مجموعه از اشیای هندسی، که به‌طور مداوم در حال حرکت هستند، (مثلاً نگهداری بالاترین نقطه از بین نقاط متحرک) از ساختارهای وابسته به حرکت استفاده می‌شود. در این فصل یک چهارچوب مفهومی برای این ساختارها، تعدادی معیار برای سنجش کارایی آن‌ها، و نیز روش‌های اساسی برای طراحی آن‌ها معرفی می‌شود. این مفاهیم کلی، با نمایش ساختارهایی برای نگهداری دو نمونه مسئله وابسته به حرکت، یعنی بالاترین نقطه و نزدیک‌ترین جفت از بین نقاط در حال حرکت بررسی می‌شوند. لازم به ذکر است که مطالب اصلی این فصل، با استفاده از مراجع [۵، ۶] تنظیم شده است.

۱.۲ مفاهیم کلی و تعاریف

از آنجایی که حرکت و پویایی بخش جدا نشدنی جهان فیزیکی است، برای شبیه‌سازی دنیای واقعی و مسائلی در حوزه‌ی هندسه محاسباتی، مانند تشخیص برخورد در رباتیک و انیمیشن، تعیین محدوده‌ی قابل رویت در گرافیک کامپیوتری، شبکه‌های بی‌سیم و تلفن همراه و غیره، ضرورت مطرح شدن ساختارهای وابسته به حرکت را توجیه می‌کنند.

در یک سیستم بررسی و کنترل ترافیک هوایی، یکی از مهم‌ترین نیازهایی که سیستم باید برآورده سازد این است که بتواند به این درخواست که کدام دو هواپیما در لحظه‌ی کنونی نزدیک‌ترین جفت هواپیما هستند، در کم‌ترین زمان پاسخ دهد تا از برخورد احتمالی بین هواپیماها جلوگیری شود. این مسئله از دنیای واقعی معادل با این مسئله‌ی هندسی است که یک مجموعه از نقاط در حال حرکت وجود دارد و هدف یافتن نزدیک‌ترین جفت نقاط در هر لحظه زمان درخواست است. بدین ترتیب نقاط در حال حرکت معادل با هواپیماها می‌شوند. برای این مسئله راه‌حل‌های مختلفی وجود دارد از جمله: ساده‌ترین راهی که ممکن است به نظر برسد، محاسبه‌ی نزدیک‌ترین جفت در هر زمان که کاربر درخواست می‌کند است. در این روش هر بار که کاربر به سیستم یک درخواست (اعلام نزدیک‌ترین جفت) می‌دهد، سیستم الگوریتم ایستای یافتن نزدیک‌ترین جفت را روی یک مجموعه از داده‌های ایستا (موقعیت فعلی نقاط)، اجرا و نتیجه را اعلام می‌کند. با این روش هر درخواست کاربر به اندازه‌ی یک‌بار اجرای کامل الگوریتم زمان می‌برد که نامطلوب است، به‌ویژه هنگامی که تعداد داده‌ها (نقاط) زیاد باشند. راه‌حل دیگری که وجود دارد این است که در فاصله‌های زمانی مشخص مثلاً هر ده دقیقه یک بار، الگوریتم یافتن نزدیک‌ترین جفت اجرا و نتیجه به عنوان نزدیک‌ترین جفت تا ده دقیقه‌ی بعد نگهداری شود. بنابراین در این فاصله‌ی زمانی ده دقیقه‌ای، هر زمان که کاربر درخواست داد در یک زمان ثابت می‌توان پاسخ داد.

هرچندکه در این روش زمان پاسخگویی خیلی کوتاه و مطلوب است، اما پاسخ داده شده چندان معتبر نیست و می‌تواند نادرست باشد؛ زیرا ممکن است در طول این فاصله‌ی زمانی نزدیک‌ترین جفت تغییر یابد. بنابراین اگر اندازه‌ی فاصله‌ی زمانی بزرگ انتخاب شود، هرچندکه تعداد محاسبات کاهش می‌یابد ولی خطای پاسخ افزایش می‌یابد. اگر اندازه‌ی فاصله‌ی زمانی کوچک انتخاب شود، هرچندکه خطای پاسخ کاهش می‌یابد ولی تعداد محاسبات افزایش می‌یابد، درحالی‌که بخشی از این محاسبات بی‌فایده نیز می‌باشد (زمانی‌که نزدیک‌ترین جفت در یک فاصله‌ی زمانی طولانی تغییری نکند).

همان‌طورکه دیده شد راه‌حل‌های بیان شده برای مسئله‌ی مطرح شده قابل قبول نیستند. راه‌حل مطلوبی که برای این مسئله در بخش ۳.۲ بیان خواهد شد بر اساس انتخاب هوشمندانه‌ی فاصله‌های زمانی شکل می‌گیرد. هرچندکه حرکت اشیا به‌صورت پیوسته از زمان می‌باشد اما نزدیک‌ترین جفت تنها در لحظه‌های گسسته‌ای از زمان تغییر می‌کند. بنابراین اگر بتوان این لحظات را تعیین و نزدیک‌ترین جفت را تنها در این لحظات به‌روزرسانی کرد، هدف مورد نظر برآورده می‌شود. برای تعیین این لحظات لازم است قبل از شروع حرکت نقاط، مسیرهای حرکت آن‌ها مشخص باشد. اگر مسیر حرکت نقاط به صورت معادلات پارامتری از زمان در نظر گرفته شود و پارامترهای دیگری که وابسته به راه‌حل مسئله است نیز اعمال شود، در مجموع می‌توان این لحظات را تعیین کرد. سپس می‌توان با ایجاد ساختارهای مناسبی نتایج مسئله را با گذشت زمان نگهداری و تنها در این لحظات ساختارها و نتایج را به‌روزرسانی کرد. در ادامه به معرفی ساختارهای وابسته به حرکت که اساس این راه‌حل هستند پرداخته می‌شود.

مشخصه‌ی یک مسئله‌ی هندسی در واقع همان درخواستی است که مسئله برای پاسخ به آن شکل گرفته است. مثلاً مسئله‌ی هندسی یافتن نزدیک‌ترین جفت از بین یک مجموعه از نقاط متحرک در صفحه را در نظر بگیرید. مشخصه‌ی این مسئله‌ی هندسی نزدیک‌ترین جفت است. هرچندکه مشخصه یا مشخصه‌های یک مسئله ثابت هستند، یعنی همواره از مسئله یک درخواست می‌شود، مثلاً: اعلام نزدیک‌ترین جفت. اما پاسخی که داده می‌شود با گذشت زمان و تغییر موقعیت نقاط بر اثر حرکت، می‌تواند تغییر کند. مثلاً نزدیک‌ترین جفت دو نقطه‌ی p, q هستند و در لحظه‌ی بعد بر اثر حرکت نقاط، موقعیت آن‌ها تغییر کرده و نزدیک‌ترین جفت دو نقطه‌ی m, n می‌شوند. بنابراین پاسخی که به مشخصه‌ی (درخواست) یک مسئله داده می‌شود و تابعی از وضعیت کنونی نقاط است **تابع وضعیت**^۱ داده‌های متحرک آن مسئله نامیده می‌شود. بنابراین در مثال نزدیک‌ترین جفت، در هر لحظه، دو نقطه‌ای که نزدیک‌ترین به یکدیگر هستند، تابع وضعیت آن لحظه را

^۱ Configuration function

مشخص می‌کنند.

یک ساختمان داده‌ی وابسته به حرکت^۲ یک ساختار برای نگهداری یک مشخصه‌ی هندسی هنگام حرکت پیوسته‌ی نقاط است و از دو جزء اصلی تشکیل شده است: (۱) یک مجموعه از روابط هندسی ساده که برای تعیین وضعیت نقاط و اثباتی برای درستی تابع وضعیت هستند. (۲) یک مجموعه از قوانین که با استفاده از آن‌ها، زمانی که ساختار به دلیل حرکت اشیا نامعتبر می‌شود، مجدداً اصلاح و به‌روزرسانی می‌گردد.

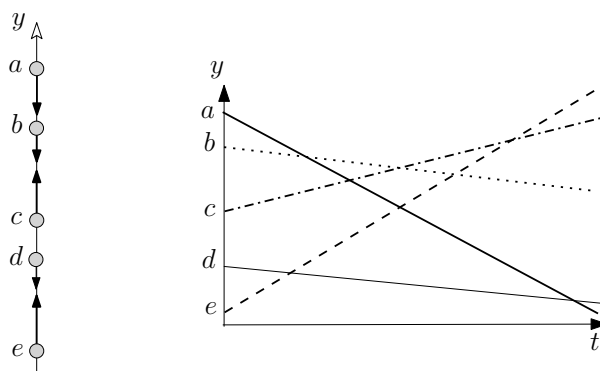
فرض می‌شود که هر شی متحرک، یک طرح پرواز^۳ (مسیر حرکت) تعیین شده دارد که تمام یا بخشی از اطلاعات را در مورد حرکت کنونی شی می‌دهد. طرح پرواز، یا مسیر حرکت هر شی با توابعی از زمان ارزیابی می‌شود. طرح پرواز یک شی می‌تواند تغییر کند. تغییر در یک طرح پرواز می‌تواند به دلیل برهم‌کنش بین اشیای در حال حرکت، محیط و غیره پیش آید. برای مثال وقتی برخوردی بین دو شی در حال حرکت رخ می‌دهد، باید طرح پرواز هر دو شی به‌روزرسانی شود. در ادامه با شرح یک مثال و ارائه‌ی راه‌حل‌های مختلف برای آن علاوه بر این که مفاهیم، تعاریف و معیارهای ارزیابی کیفیت ساختمان داده‌های وابسته به حرکت توضیح داده می‌شود، کارایی ساختارهای وابسته به حرکت هرم و درخت مسابقه برای حل این مسئله مشخص خواهد شد.

۱.۱.۲ یک مثال با شرح و توضیح

مثالی که در این زیربخش بررسی خواهد شد، مسئله‌ی یافتن بالاترین نقطه از بین نقاط در حال حرکت است. در ادامه به معرفی دقیق این مثال پرداخته می‌شود. برای پیوند بهتر مسائلی که مطرح خواهند شد، وضعیت ساده یک بعدی زیر در نظر گرفته می‌شود. مجموعه‌ای از n نقطه، پیوسته در حال حرکت در امتداد محور y هستند. هدف، تعیین نقطه‌ای است که در هر زمان بالاترین نقطه است (بزرگ‌ترین آن‌ها، اگر نقاط را بعنوان اعداد در نظر گرفته شود). اگر دو نقطه با هم برخورد کنند، فرض می‌شود که آن‌ها بدون برهم‌کنش بر یکدیگر از هم عبور می‌کنند. علاوه بر آن، فرض می‌شود که نقاط با شروع از یک پیکربندی اولیه‌ی دلخواه و با سرعت‌های ثابت حرکت می‌کنند، که سرعت ثابت به جهت حرکت ثابت نیز دلالت می‌کند. (سرعت‌های ثابت اما با این امکان که متفاوت از یکدیگر باشند). در ادامه چهار راه‌حل متفاوت برای این مسئله مطرح شده و با توجه به معیارهایی که معرفی خواهند شد کیفیت این راه‌حل‌ها بررسی می‌شود.

^۲Kinetic data structure (KDS)

^۳Flight plan



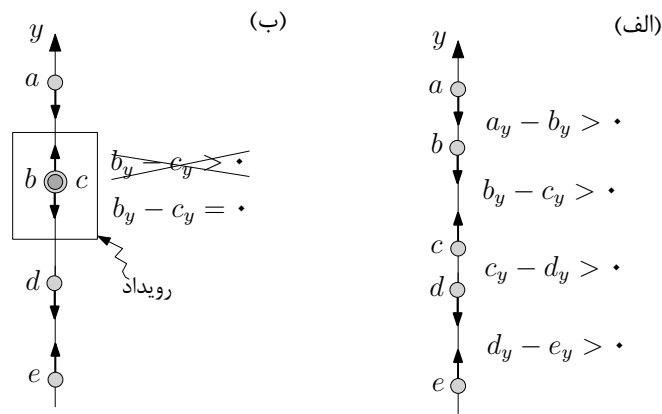
شکل ۱.۲: نمودار ty

راه حل اول

اگر مسیرهای حرکت نقاط در صفحه ty رسم شود (که محور افقی t است و به زمان اشاره می‌کند)، سپس مسئله، تبدیل به مسئله‌ی محاسبه‌ی پوشش بالایی یک مجموعه از n خط مستقیم در صفحه می‌شود (شکل ۱.۲). این پوشش بالایی می‌تواند در زمان $O(n \log n)$ با الگوریتم تقسیم و غلبه که در [۲۹] معرفی شده محاسبه شود. در بدترین حالت، تعداد دفعاتی که در طول حرکت، بالاترین نقطه (تابع وضعیت) تغییر می‌کند، برابر با پیچیدگی (تعداد رئوس) پوشش بالایی خطوط در صفحه‌ی ty است که با توجه به آن چه که در بخش ۲.۱ بیان شد با کاهش مسئله به دنباله‌ی داونپورت-شینزل متناظرش، این پیچیدگی برابر با طول دنباله‌ی $\lambda_3(n)$ می‌شود که با استفاده از قضیه‌ی ۲.۲.۱، $\lambda_3(n) = \Theta(\alpha(n)n)$ است و از آن جایی که رشد تابع $\alpha(n)$ بسیار آهسته و برای مقادیر عملی n کم‌تر از عدد ۴ است، بنابراین می‌توان گفت، پیچیدگی پوشش بالایی این خطوط برابر $\Theta(n)$ است. با این روش می‌توان تابع وضعیت (پوشش بالایی) را در زمان $O(n \log n)$ که تنها یک ضریب لگاریتمی بیش‌تر از حداکثر تعداد تغییرات در آن یعنی $\Theta(n)$ است، محاسبه کرد.

این راه حل رضایت‌بخش نیست؛ زیرا اساس آن معلوم بودن کل حرکت نقاط است و چنانچه طرح پرواز نقطه‌ای با گذشت زمان به دلیل برخورد با نقاط دیگر تغییر کند برای به‌روزرسانی پوشش بالایی، باید دوباره الگوریتم، با زمان $O(n \log n)$ از نو اجرا شود (هزینه‌ای بالا تنها برای یک تغییر). هدف، استراتژی است که به صورت بر-خط^۴ کار می‌کند و می‌تواند به‌روزرسانی طرح‌های پرواز را به‌راحتی و با کم‌ترین هزینه در خود جای دهد. بنابراین این راه حل کنار گذاشته می‌شود و راه حل بعدی بررسی می‌شود.

^۴On-line



شکل ۲.۲: شکل (الف): تاییدیه‌های مربوط به نگهداری ترتیب نقاط متحرک روی محور y ، شکل (ب): بروز یک رویداد بر اثر حرکت نقاط

راه حل دوم

در این روش سعی می‌شود تا ترتیب مرتب شده‌ای از نقاط در امتداد محور y ، به صورت بر-خط نگهداری شود. برای شرح بیش‌تر این روش لازم است مفاهیم زیر تعریف شوند:

۱. **تاییدیه‌ها**^۵: مجموعه‌ای از شرایط و قیدها، که در هر زمان، اثباتی برای درستی تابع وضعیت در آن زمان هستند. تاییدیه‌ها، یک ویژگی کلیدی برای ساختمان داده‌های وابسته به حرکت محسوب می‌شوند. برای نمونه، در راه حل دوم (نگهداری ترتیب مرتب شده‌ای از نقاط روی محور y) مجموعه روابط جبری برای مقایسه‌ی مولفه‌ی y دو نقطه، تاییدیه‌های این روش هستند. اگر مولفه‌ی y هر نقطه‌ی a ، با a_y نشان داده شود و ترتیب نقاط نزولی باشد، آنگاه مجموعه‌ی تاییدیه‌ها، در هر زمان t ، شامل $n - 1$ تاییدیه از نوع $a_y(t) - b_y(t) > 0$ ، برای هر دو نقطه‌ی a و b پشت‌سرهم می‌باشد. شکل ۲.۲ قسمت (الف) را ببینید. بدیهی است که این تاییدیه‌ها اثباتی برای درستی تابع وضعیت در هر زمان t ، که بالاترین نقطه روی محور y در زمان t است، می‌باشند.

۲. **رویداد**^۶: زمانی که یک تاییدیه، نامعتبر می‌شود (تغییر علامت می‌دهد)، یک رویداد رخ می‌دهد. شکل ۲.۲ قسمت (ب) را ببینید. در زمان بروز یک رویداد، ساختمان داده‌ی وابسته به حرکت، باید به‌روزرسانی شود تا همچنان معتبر باقی بماند. تا زمانی که تمام تاییدیه‌ها معتبر است، تابع وضعیت نیز درست و

^۵Certificates

^۶Event

معتبر است. در زمان بروز یک رویداد (نامعتبر شدن یک تاییدیه)، تابع وضعیت ممکن است تغییر کند یا نکند. مثلاً در رویداد شکل ۲.۲ قسمت (ب) تابع وضعیت (بالاترین نقطه) تغییر نمی‌کند. اما اگر در این شکل پس از این رویداد، در رویداد بعدی نقطه‌ی a با نقطه‌ی c جابه‌جا شوند، تابع وضعیت، یعنی بالاترین نقطه تغییر می‌کند. به عبارت دیگر، تمام رویدادها، منجر به به‌روزرسانی و تغییر ساختارها می‌شوند، اما تنها بخشی از آن‌ها هستند که تابع وضعیت را نیز تغییر می‌دهند.

۳. **صف رویداد**^۷: واسطه‌ی بین ساختمان داده‌های وابسته به حرکت و اشیای در حال حرکت، یک صف رویداد سراسری است. متناظر با هر تاییدیه زمان نامعتبر شدن آن (زمان رویداد آن) با استفاده از مسیر حرکت نقاط تعیین و در صف رویداد قرار می‌گیرد. این صف یک صف اولویت است، به طوری که هر رویداد با اولویت زمانی بیش‌تر (رویدادی که زودتر رخ خواهد داد) در ابتدای صف قرار خواهد گرفت. برای شبیه‌سازی یک مسئله‌ی وابسته به حرکت کافی است رویدادها را یکی یکی از صف برداشت و ساختمان داده‌ی وابسته به حرکت را بر اساس آن به‌روزرسانی کرد. لازم به ذکر است که عملیات حذف و درج برای صف رویداد که یک صف اولویت است همواره از مرتبه‌ی $O(\log n)$ است.

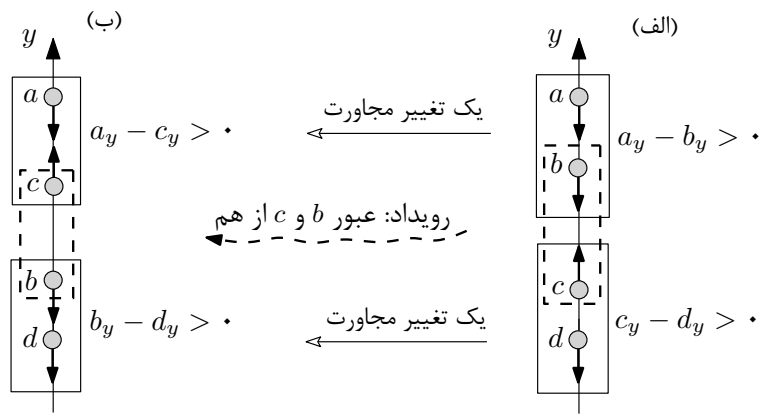
همان‌طور که دیده شد در این راه‌حل، برای هر جفت از نقاط، که اکنون در امتداد محور y پشت‌سرهم هستند، یک رویداد برنامه‌ریزی می‌شود، که برابر با اولین زمانی است که این نقاط از هم عبور می‌کنند. (یا اگر اطلاعات ما در مورد حرکت‌ها کامل نبود، می‌توان بر اساس مدت زمانی، که با اطمینان می‌توان گفت ترتیب مرتبط با نقاط در این مدت تغییر نمی‌کند، یک رویداد را تقریب زد). در ادامه چهار معیار مهم برای ارزیابی کیفیت ساختمان داده‌های وابسته به حرکت معرفی می‌شود و هم‌زمان با آن، راه‌حل دوم نیز بررسی می‌شود تا مشخص شود که این معیارها را دارد یا نه. قبل از آن لازم است توضیح داده شود منظور از "کم باشد" در معرفی معیارهای ارزیابی این است که: نزدیک به مرتبه‌ی $O(Poly \log(n))$ ، یا $O(n^\epsilon)$ (در صورت وجود یک $\epsilon > 0$ و کوچک) باشد. که n تعداد نقاط یا اشیای هندسی مسئله می‌باشد.

۱. **پاسخ‌گویی**^۸: در زمان پردازش هر رویداد، نیاز است تا به‌روزرسانی‌های زیر روی ساختار وابسته به حرکت اعمال گردد:

(الف) تاییدیه‌هایی که نامعتبر شده‌اند و اثبات درستی ساختار را نادرست کرده‌اند حذف شوند، همچنین

^۷Event queue

^۸Responsiveness



شکل ۳.۲: وضعیت (الف): ترتیب مرتب شده نقاط قبل از رویداد، وضعیت (ب): ترتیب مرتب شده نقاط بعد از رویداد

رویدادهای متناظر با این تاییدیه‌ها نیز از صف رویداد حذف گردند.

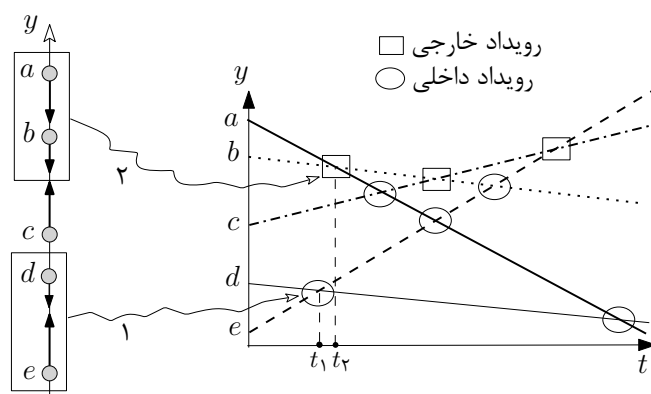
(ب) تاییدیه‌های جدیدی که برای برقراری مجدد درستی ساختار لازم هستند اضافه شوند، همچنین رویدادهای جدید متناظر با این تاییدیه‌ها به صف رویداد اضافه گردند.

یک ساختمان داده‌ی وابسته به حرکت پاسخ‌گو است اگر برای هر رویداد، زمان به‌روزرسانی ساختار کم باشد. در حالت کلی این به‌روزرسانی باید برای تاییدیه‌هایی که درستی ساختار را اثبات می‌کنند و (شاید) برای تابع وضعیت صورت گیرد. به عنوان نمونه، ساختار راه‌حل دوم مسئله‌ی بالاترین نقطه روی محور y (نگهداری ترتیب مرتب شده نقاط) پاسخ‌گوست؛ زیرا زمانی که یک رویداد رخ می‌دهد و دو نقطه‌ی مجاور از هم عبور می‌کنند، تنها دو مجاورت (تاییدیه) قدیمی نامعتبر شده و دو مجاورت (تاییدیه) جدید در لیست مرتب شده‌ی نقاط ایجاد می‌شود که این عملیات از مرتبه‌ی $O(1)$ خواهد بود. همچنین دو رویداد قدیمی متناظر با آن‌ها از صف رویداد حذف و دو رویداد جدید در صف رویداد درج می‌گردد (شکل ۳.۲ را ببینید)، که این عملیات از مرتبه‌ی $O(\log n)$ خواهد بود. بنابراین ساختار هنگام پردازش یک رویداد، می‌تواند در زمان لگاریتمی خودش را به‌روزرسانی کند و پاسخ‌گوست.

۲. کارایی^۹: برای تعریف دومین معیار، لازم است بین رویدادهای خارجی^{۱۰}، یعنی آن‌هایی که تابع

^۹Efficiency

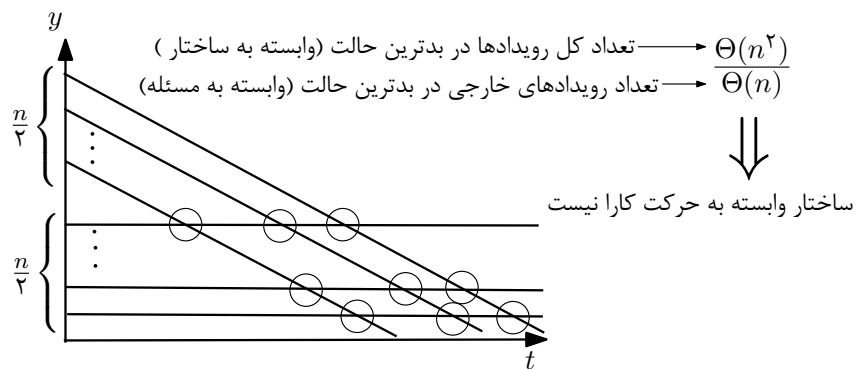
^{۱۰}External events



شکل ۴.۲: رویدادهای داخلی و رویدادهای خارجی برای مسئله بالاترین نقطه از بین نقاط متحرک

وضعیت را تغییر می‌دهند و رویدادهای داخلی^{۱۱}، یعنی آنهایی که به دلیل نیازهای داخلی ساختار وابسته به حرکت پردازش می‌شوند اما تابع وضعیت را تغییر نمی‌دهند، تمایز قائل شد. مثلاً در شکل ۴.۲ رویداد شماره ۱ که در زمان t_1 رخ می‌دهد یک رویداد داخلی و رویداد شماره ۲ که در زمان t_2 رخ می‌دهد یک رویداد خارجی است. از آنجایی که هر ساختار وابسته به حرکت برای تعیین تابع وضعیت ایجاد می‌شود، تنها محاسبه‌ی رویدادهای خارجی مورد نظر است ولی در اکثر موارد در این ساختارها نمی‌توان به‌طور کامل از پردازش رویدادهای داخلی اجتناب کرد. تعداد رویدادهای خارجی، وابسته به مسئله‌ی هندسی است و نه ساختار وابسته به حرکت. بنابراین این تعداد یک کران پایین را برای هر ساختار وابسته به حرکت، به‌منظور نگهداری تابع وضعیت مسئله تعیین می‌کند. در نتیجه ساختارهای وابسته به حرکتی مناسب هستند که تعداد کل رویدادهای پردازش شده توسط آن‌ها در بدترین حالت، نزدیک یا کمی بزرگ‌تر از تعداد رویدادهای خارجی مسئله در بدترین حالت باشد (نسبت کل رویدادها به رویدادهای خارجی کم باشد)، یک ساختار وابسته به حرکت که این ویژگی را داشته باشد کارا است. در ادامه کارایی راه‌حل دوم مسئله‌ی بالاترین نقطه روی محور y (نگهداری ترتیب مرتب شده نقاط روی محور y) بررسی می‌شود. اگر مسیر حرکت نقاط در صفحه‌ی ty شود، که محور t به زمان اشاره می‌کند (یادآوری: فرض شد که نقاط با سرعت ثابت حرکت می‌کنند و مسیر حرکت آن‌ها خطی از زمان می‌شود)، هر تقاطع بین دو خط، معادل یک رویداد است. بنابراین برای محاسبه‌ی رویدادها باید تقاطع‌های بین هر جفت از خطوط در صفحه محاسبه شود، که با استفاده از الگوریتم پیمایش صفحه‌ی

^{۱۱}Internal events



شکل ۵.۲: بررسی کارایی ساختار وابسته به حرکت

بنتلی^{۱۲} و اتمان^{۱۳} [۹] و اصلاح شده‌ی آن توسط برون^{۱۴} [۱۳] به راحتی امکان پذیر است. ساختمان داده‌ی وابسته به حرکت ممکن است مجبور باشد تا $\Theta(n^2)$ رویداد را پردازش کند. برای نمونه فرض کنید نیمی از نقاط، بی حرکت هستند و نیم دیگر از آن‌ها می‌گذرند، در این حالت $\Theta(n^2)$ برخورد بین خطوط وجود دارد (شکل ۵.۲) و الگوریتم باید تمام این برخوردها را که هر کدام معادل یک رویداد است بررسی کند، درحالی که همان طور که در راه حل اول هم بیان شد تنها $\Theta(n)$ تا از آن‌ها رویداد خارجی هستند (روی پوشش بالایی خطوط قرار دارند). بنابراین نسبت تعداد کل رویدادها به تعداد رویدادهای خارجی در این حالت خطی از تعداد نقاط می‌شود و این راه حل کارا نیست. شکل ۵.۲ را ببینید.

۳. **فشرده‌گی**^{۱۵}: اگر بیشترین تعداد رویدادی که در یک مرحله از زمان می‌تواند در صف قرار گیرد اندازه‌ی^{۱۶} یک ساختمان داده‌ی وابسته به حرکت تعریف شود، بنابراین یک ساختمان داده‌ی وابسته به حرکت فشرده است اگر اندازه‌ی آن تقریباً خطی از تعداد اشیای در حال حرکت باشد.

۴. **محل‌ی بودن داده‌ها**^{۱۷}: یک ساختمان داده‌ی وابسته به حرکت **محل‌ی** است، هنگامی که در هر زمان حداکثر تعداد رویداد در صف که وابسته به هر شی است، کوچک باشد. این ویژگی در دسترسی سریع برای

^{۱۲}Bentley

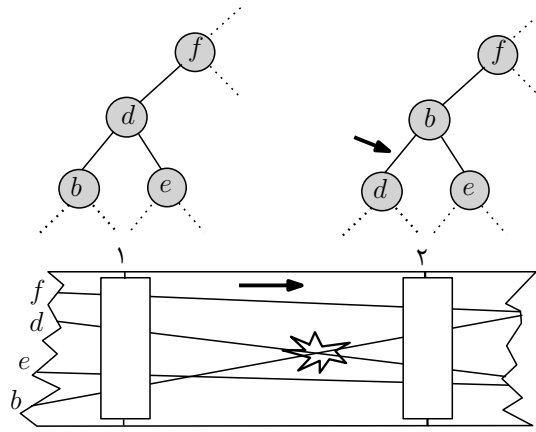
^{۱۳}Ottmann

^{۱۴}Brown

^{۱۵}Compactness

^{۱۶}Size

^{۱۷}Locality



شکل ۶.۲: بروز یک رویداد در یک هرم وابسته به حرکت

به‌روزرسانی طرح‌های پرواز بسیار مهم است.^{۱۸} زیرا هنگام به‌روزرسانی طرح پرواز یک شی، چنان‌چه آن شی در تعداد کمی از رویدادها مشارکت داشته باشد (ساختار محلی باشد)، زمان به‌روزرسانی مولفه‌های مرتبط به آن شی، یک زمان کم و منطقی خواهد بود.

در مورد ساختار وابسته به حرکت راه‌حل دوم می‌توان گفت: از آن جایی که هر نقطه حداکثر در دو تاییدیه (مجاورت) قرار می‌گیرد و رویدادها متناظر با تاییدیه‌ها می‌باشند، بنابراین هر نقطه حداکثر وابسته به دو رویداد در صف می‌شود و ساختار محلی است. همچنین با توجه به این اندازه‌ی صف نیز خطی از تعداد نقاط می‌شود و ساختار فشرده نیز است. بنابراین ساختار وابسته به حرکت راه‌حل دوم پاسخ‌گو، محلی و فشرده است اما کارا نیست. به همین دلیل این راه‌حل کنار گذاشته می‌شود و راه‌حل بعدی مطرح می‌شود.

راه‌حل سوم

در سومین ساختار برای نگهداری بالاترین نقطه از بین نقاط متحرک روی محور y ، که تمام معیارهای ارزیابی کیفیت یک ساختار وابسته به حرکت برای آن برقرار است، نقاط در حال حرکت، در یک هرم نگهداری می‌شوند که ریشه‌ی هرم شامل بالاترین (بزرگ‌ترین) عنصر است. برای وابسته به حرکت کردن یک هرم به‌صورت زیر عمل می‌شود. هر اتصال بین دو گره (ارتباط پدر- فرزندی) در هرم یک تاییدیه دارد، که تضمین می‌کند که نقطه‌ی فرزند در پایین نقطه‌ی پدر قرار دارد و متناظر با هر این‌چنین اتصالی، یک رویداد تعریف می‌شود، که زمان عبور این دو نقطه از هم می‌باشد. برای پردازش یک رویداد و نگهداری یک هرم همچنان معتبر، کافی است که پدر و فرزند آن اتصالی که مربوط به آن رویداد است با هم عوض شوند (شکل ۶.۲)، چنان‌چه سایر

^{۱۸} توجه شود که محلی بودن داده‌ها، به فشردگی آن‌ها نیز اشاره می‌کند. اما، سایر معیارهای ارزیابی کیفیت، مستقل هستند.

نامساوی‌های (اتصال‌های) هرم در آن زمان هنوز معتبر هستند (فرض می‌شود که حرکت نقاط پیوسته است و هیچ دو رویدادی هم‌زمان رخ نمی‌دهند. به عبارت دیگر در این جا هیچ دو اتصالی هم‌زمان نامعتبر نمی‌شوند). زمانی که تعویض دو عنصر در هرم اتفاق می‌افتد، باید حداکثر چهار رابطه‌ی مجاورت (پدر-فرزندی) در هرم تغییر کند، همچنین باید چهار رویداد قدیمی متناظر با این مجاورت‌ها را از صف رویداد حذف و چهار رویداد جدید در صف درج گردد. برای مثال در شکل ۶.۲، پس از رخ دادن رویداد، برای به‌روزرسانی هرم، کافی است b و d در هرم با هم عوض شوند، علاوه بر آن تاییدیه‌ها و رویدادهای مربوط به چهار اتصال df ، de و دو اتصال مربوط به فرزندان گره b باید حذف و چهار تاییدیه و رویداد جدید مجدداً برنامه‌ریزی شوند.

این توضیحات **هرم وابسته به حرکت**^{۱۹} را توصیف می‌کند، که در هر زمان بالاترین عنصر را برای مجموعه‌ی عناصر در حال حرکت نگهداری می‌کند. پاسخ‌گویی، محلی بودن و فشردگی ساختار واضح است. اما هرم وابسته به حرکت در بدترین حالت، چه تعداد رویداد را باید پردازش کند؟ (فرض شد که نقاط با سرعت‌های ثابت حرکت می‌کنند.) شاید تصور شود که تعداد رویدادها بدیهی است و به راحتی محاسبه می‌شوند، اما این‌گونه نیست. برای محاسبه‌ی تعداد رویدادهایی که این ساختار محاسبه می‌کند، آن‌چه که بدیهی است این است که تعداد رویدادها کم‌تر از تعداد برخوردها که از مرتبه‌ی $O(n^2)$ است، خواهد بود؛ زیرا برای برخوردهایی مانند برخورد b و e در شکل ۶.۲ رویدادی تعریف نخواهد شد. برای محاسبه‌ی تعداد رویدادها می‌توان از روش مطرح شده در [۷] استفاده کرد. با این روش، هرم وابسته به حرکت برای مسیرهای حرکت خطی تعداد $O(n \log^2 n)$ رویداد را پردازش می‌کند. بنابراین این ساختار کارا هم هست و تمام چهار معیار ارزیابی کیفیت ساختارهای وابسته به حرکت را داراست.

هرچند که هرم وابسته به حرکت ساختار مناسبی است، اما کارا بودن آن تنها تحت این فرض که مسیر حرکت نقاط تابعی خطی از زمان است بررسی شد. برای این که راه‌حل کلی‌تری (امکان غیر خطی بودن مسیر نقاط نیز باشد) برای نگهداری بزرگ‌ترین عنصر در مسائل وابسته به حرکت مطرح شود، چهارمین ساختار معرفی می‌شود. این ساختار ابتدا برای این مسئله مطرح می‌شود و سپس چهارچوب کلی آن برای استفاده در سایر مسائل وابسته به حرکت مطرح خواهد شد.

^{۱۹}Kinetic heap

راه حل چهارم

قبل از مطرح کردن این راه حل، لازم است یک استراتژی کلی برای وابسته به حرکت کردن بیان شود. این استراتژی به این صورت است که ابتدا یک الگوریتم ایستای مناسب برای مسئله‌ی هندسی اجرا می‌شود. سپس نتایج (خروجی‌های) الگوریتم در ساختارهای مناسبی نگهداری می‌شود. تاییدیه‌هایی برای نمایش وضعیت ساختار تعریف می‌شوند که اثباتی برای درستی تابع وضعیت نیز هستند. در ادامه مرتبط با هر کدام از این تاییدیه‌ها یک رویداد در صف سراسری قرار می‌گیرد و زمانی که در آینده، این تاییدیه نامعتبر خواهد شد (یا ممکن است نامعتبر شود) را نشان می‌دهد. به این فرایند که الگوریتم‌ای که برای داده‌های ایستاست به یک ساختمان داده‌ی وابسته به حرکت تبدیل می‌شود **وابسته به حرکت کردن**^{۲۰} گفته می‌شود. بسیار اهمیت دارد که ساختار وابسته به حرکت ایجاد شده، در زمان بروز یک رویداد، بتواند از یک راه ساده و کارا خروجی الگوریتم را به‌روزرسانی کند و مجدداً مجموعه‌ی تاییدیه‌ها را برای نشان دادن درستی ساختار تنظیم و نگهداری کند. در ادامه این استراتژی کلی با گام‌هایی روی درخت مسابقه که نتایج محاسبات الگوریتم ایستای تعیین بزرگ‌ترین عدد را دربر دارد، نشان داده خواهد شد. لازم به ذکر است که در مسئله‌ی یافتن بالاترین نقطه روی محور y ، مولفه‌ی y هر نقطه یک عدد است و این مسئله در واقع یک مسئله‌ی یافتن بزرگ‌ترین عدد از بین اعداد متغیر است که در ادامه بیان می‌شود.

۱. معرفی الگوریتم ایستای تعیین بزرگ‌ترین عدد و ساختار درخت مسابقه برای نگهداری نتایج

آن: ابتدا یک الگوریتم که بزرگ‌ترین عدد را از بین n عدد (ایستا) محاسبه می‌کند، در نظر بگیرید. الگوریتم، بزرگ‌ترین عدد را به صورت بازگشتی با افزاز اعداد به دو گروه (دلخواه) با اندازه‌ی تقریباً مساوی، محاسبه‌ی بزرگ‌ترین عدد در هر گروه و سپس مقایسه‌ی این دو عدد برای انتخاب عدد بزرگ‌تر محاسبه می‌کند. اگر نگاهی از پایین به بالای این ساختار انداخته شود، این دقیقاً یک مسابقه، برای محاسبه‌ی برنده‌ی نهایی است. در پایان، این الگوریتم $O(n)$ مقایسه را اجرا کرده است که اثبات می‌کند، بزرگ‌ترین عددی که محاسبه شده است به درستی بزرگ‌ترین عدد از بین اعداد است. عملکرد این الگوریتم را می‌توان با یک درخت دودویی نشان داد، به طوری که در هر گره v ، بزرگ‌ترین عدد واقع در زیردرخت با ریشه‌ی v قرار می‌گیرد که حاصل از مقایسه‌ی عدد گره فرزند چپ و عدد گره فرزند راستش می‌باشد. بنابراین هر گره درخت متناظر با یک مقایسه است، (شکل ۷.۲ قسمت (الف) را ببینید) به این درخت، **درخت**

^{۲۰} Kinetization

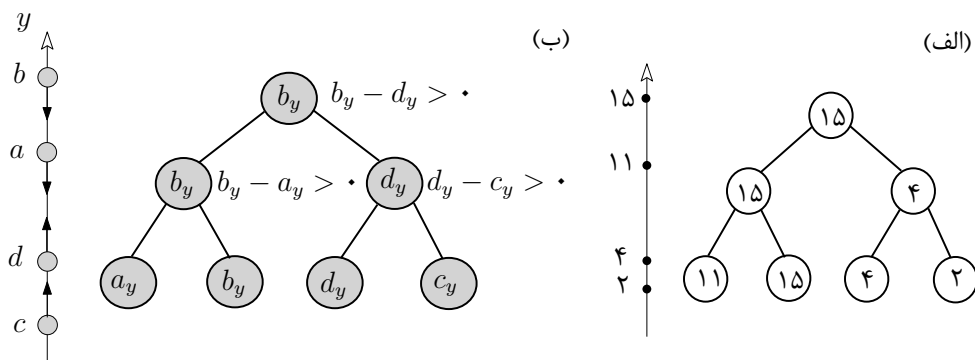
مسابقه^{۲۱} گفته می‌شود.

۲. معرفی تاییدیه‌ها و رویدادها (وابسته به حرکت کردن درخت مسابقه): حال اگر اعداد شروع به تغییر کنند، تا زمانی که مقایسه‌های الگوریتم معتبر باقی می‌مانند، بزرگ‌ترین عدد نیز معتبر باقی می‌ماند، هرچند که ممکن است مقدار آن تغییر کند. بنابراین برای وابسته به حرکت کردن ساختار درخت مسابقه، کافی است برای هر مقایسه‌ای که الگوریتم انجام داده است (این مقایسه، بین برنده‌های دو زیرگروه که در یک سطح معین از درخت مسابقه قرار دارند، انجام شده است)، یعنی هر گره درخت، یک تاییدیه تعریف شود و متناظر با هر تاییدیه نیز یک رویداد در صف رویداد ساختار قرار گیرد. شکل ۷.۲ قسمت (ب) را ببینید. با اجرای این روند، الگوریتم ایستای تعیین بزرگ‌ترین عدد تبدیل به یک ساختار وابسته به حرکت شد که به این ساختار یک **مسابقه‌ی وابسته به حرکت**^{۲۲} گفته می‌شود.

۳. ارزیابی کیفیت مسابقه‌ی وابسته به حرکت: زمانی که یک تاییدیه نامعتبر می‌شود (بروز یک رویداد) بدین معنی است که برنده‌ی مقایسه‌ی متناظر با آن تغییر کرده است. سپس این برنده‌ی جدید به سمت بالای درخت مسابقه نفوذ می‌کند تا زمانی که در یک سطح درخت شکست بخورد یا به ریشه‌ی درخت رسیده و به عنوان بزرگ‌ترین عدد تعیین شود. از آنجایی که درخت مسابقه متوازن است، بنابراین این محاسبات (مربوط به یک رویداد) حداکثر به اندازه‌ی ارتفاع درخت، یعنی $O(\log n)$ بالا می‌رود و از آنجایی که یک تعداد ثابت تغییرات برای تاییدیه‌ها در هر سطح لازم است (یک تعداد ثابت از آن‌ها باید به دلیل نامعتبر شدن حذف و تاییدیه‌های جدید معتبر به جای آن‌ها درج شوند)، بنابراین به‌روزرسانی ساختار هنگام یک رویداد می‌تواند به‌سادگی و در زمان $O(\log n)$ انجام شود و ساختار پاسخ‌گوست. علاوه بر این چون هر عدد حداکثر در $O(\log n)$ (ارتفاع درخت) تاییدیه شرکت دارد، ساختار محلی و فشرده نیز هست. در ادامه معیار کارایی برای ساختار بررسی خواهد شد. اگر اعداد به‌طور یکنواخت تغییر کنند (یعنی نقاط متناظر با این اعداد روی محور y با سرعت‌های ثابت حرکت کنند)، مسابقه‌ی وابسته به حرکت چه تعداد رویداد را باید پردازش کند؟ نکته‌ی کلیدی برای جواب دادن به این سوال تشخیص این است که مسابقه‌ی وابسته به حرکت، یک الگوریتم تقسیم و غلبه را برای محاسبه‌ی پوشش بالایی n خط مستقیم در صفحه ty (مسیرهای حرکت نقاط)، اجرا می‌کند. برای مثال، مقایسه‌هایی که

^{۲۱}Tournament tree

^{۲۲}Kinetic tournament



شکل ۷.۲: شکل (الف): یک درخت مسابقه‌ی ایستا و شکل (ب): یک درخت مسابقه‌ی وابسته به حرکت برای مسئله‌ی بالاترین نقطه از بین نقاط متحرک روی محور y

در طول زمان در بالاترین سطح (ریشه‌ی درخت مسابقه)، برای اعلان برنده‌ی نهایی انجام می‌شود، دقیقاً همان‌هایی هستند که برای ادغام پوشش‌های بالایی دو زیرگروه از خطوط نیاز می‌شوند. به راحتی دیده می‌شود که هزینه‌ی کلی ادغام دو پوشش $O(n)$ است. بنابراین اجرای راه تقسیم و غلبه برای محاسبه‌ی پوشش بالایی، یک بدترین هزینه را دارد که از رابطه‌ی بازگشتی $C(n) = 2C(n/2) + \Theta(n)$ به دست می‌آید و جواب این رابطه‌ی بازگشتی $C(n) = O(n \log n)$ است. تعداد رویدادهای مسابقه‌ی وابسته به حرکت متناسب با تعداد دفعاتی است که هویت یکی از مسابقه دهنده‌ها در یک نود درخت مسابقه تغییر می‌کند. هر این چنین تغییر هویتی، متناظر با یک برخورد در یکی از زیرپوشش‌های محاسبه شده توسط الگوریتم تقسیم و غلبه است، که توسط معادله‌ی بازگشتی $C(n)$ شمرده می‌شود. بنابراین تعداد کل رویدادهای مسابقه‌ی وابسته به حرکت برابر جواب معادله‌ی بازگشتی، یعنی $O(n \log n)$ می‌شود. پس مسابقه‌ی وابسته به حرکت، نگهداری بر خط بزرگ‌ترین عدد از یک مجموعه از اعداد در حال تغییر را به خوبی انجام می‌دهد و علاوه بر آن پاسخ‌گو، کارا، فشرده و محلی است.

۲.۲ مسابقه‌ی وابسته به حرکت

در درخت مسابقه‌ی وابسته به حرکت معرفی شده در بالا، عناصر تشکیل دهنده‌ی درخت مسابقه ثابت هستند و تنها هنگام بروز یک رویداد، تغییر مقدار می‌دهند که این تغییر مقدار با حذف مقدار قبلی عنصر و درج مقدار جدید آن انجام می‌شود. در حالت کلی یک درخت مسابقه‌ی وابسته به حرکت می‌تواند علاوه بر امکان تغییر مقدار عناصر، امکان حذف یک عنصر به طور کامل از ساختار (ناپدید شدن آن) و نیز امکان درج عنصری کاملاً

جدید در ساختار (پدیدار شدن آن) را نیز داشته باشد، البته با این شرط که تعداد عناصر تشکیل دهنده‌ی ساختار در هر زمان از مرتبه‌ی $O(n)$ باقی بماند (n : تعداد عناصر تشکیل دهنده‌ی ساختار در شروع حرکت). برای این که بین این نوع حذف و درج، با حذف و درج مربوط به تغییر مقدار، تمایز ایجاد شود از اصطلاح ناپدید شدن و پدیدار شدن استفاده می‌شود.

یک درخت مسابقه‌ی وابسته به حرکت با n عنصر تشکیل دهنده در شروع حرکت، که امکان پدیدار و ناپدید شدن عناصر در آن وجود دارد را در نظر بگیرید. اگر تعداد تمام عناصر متمایزی که حداقل یک‌بار در این ساختار حضور داشته‌اند با m نشان داده شود، آنگاه لم زیر برای شمارش تعداد کل رویدادها در این ساختار برقرار است.

لم ۱.۲.۲ ([۵]) تعداد کل رویدادهایی که می‌تواند در یک مسابقه‌ی وابسته به حرکتی که امکان پدیدار و ناپدید شدن عناصر را دارد، رخ دهد، برابر با $O(\lambda_{s+2}(m) \log n)$ است. که $\lambda_{s+2}(m)$ به طول بزرگ‌ترین $(m, s+2)$ دنباله‌ی داوونپورت-شینزل اشاره می‌کند (بخش ۲.۱ را ببینید).

اثبات. از آنجایی که هر رویداد با نامعتبر شدن تأییدیه‌ی یک گره (تغییر مقدار آن) رخ می‌دهد، بنابراین تعداد کل رویدادهایی که در یک درخت مسابقه‌ی وابسته به حرکت رخ خواهد داد، متناسب می‌شود با تعداد دفعاتی که گره‌های آن درخت تغییر خواهند کرد. در ادامه این تعداد تغییرات شمارش خواهد شد.

گره v را در درخت مسابقه در نظر بگیرید. اگر تعداد تمام عناصر متمایزی که حداقل یک‌بار در زیردرخت با ریشه‌ی v ، حضور داشته‌اند با n_v نشان داده شود، مقدار بزرگ‌ترین عنصر در این زیردرخت (مقدار گره v) به اندازه‌ی پیچیدگی پوشش بالایی توابع تغییر مقدار عناصرش، یعنی حداکثر $\lambda_{s+2}(n_v)$ بار، تغییر کرده است. توجه شود که چون امکان ناپدید و پدیدار شدن عناصر وجود دارد، پس هر عنصر می‌تواند در یک بازه‌ی (یا بازه‌های) مشخص زمانی در ساختار حضور داشته باشد. این بدان معنی است که پوشش بالایی باید برای یک مجموعه از توابع که به صورت بازه‌ای تعریف شده‌اند، محاسبه گردد. پس با توجه به بخش ۲.۱ به جای λ_s ، از λ_{s+2} استفاده می‌شود.

اگر L_i به مجموعه‌ی گره‌هایی که در سطح i هستند اشاره کند،

$$\sum_{v \in L_i} n_v \leq m.$$

بنابراین تعداد کل تغییرات (رویدادها) در سطح i حداکثر برابر است با

$$\sum_{v \in L_i} \lambda_{s+2}(n_v) \leq \lambda_{s+2}(m).$$

از آن جایی که درخت مسابقه دارای ارتفاع $O(\log n)$ است بنابراین کل تعداد رویدادها برابر با $O(\lambda_{s+2}(m) \log n)$ می‌شود. □

با توجه به تمام مطالب بیان شده، بدیهی است که برای ساختار مسابقه‌ی وابسته به حرکتی که امکان پدیدار و ناپدید شدن عناصر را نداشته باشد (عناصر تشکیل‌دهنده‌ی ساختار در تمام زمان‌ها ثابت و برابر با n است)، لم زیر برقرار خواهد بود.

لم ۲.۲.۲ ([۴۰]) تعداد کل رویدادهایی که می‌تواند در یک مسابقه‌ی وابسته به حرکتی که امکان پدیدار و ناپدید شدن عناصر را ندارد، رخ دهد، برابر با $O(\lambda_s(n) \log n)$ است.

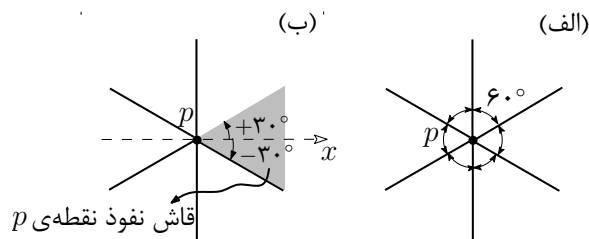
۳.۲ نزدیک‌ترین جفت از بین نقاط در حال حرکت

هرچند که الگوریتم‌های مناسبی برای یافتن نزدیک‌ترین جفت در حالت ایستا (نقاط ثابت و بدون حرکت) وجود دارد [۱۰، ۱۴، ۲۸]، ولی زمانی که این الگوریتم‌ها وابسته به حرکت می‌شوند کارا نیستند. بنابراین در این بخش، ابتدا الگوریتم ایستای جدیدی، برای یافتن نزدیک‌ترین جفت توصیف می‌شود، سپس ساختارهای وابسته به حرکت برای نگهداری خروجی این الگوریتم مطرح می‌شوند و نشان داده می‌شود که این ساختارها می‌توانند زمانی که نقاط در صفحه حرکت می‌کنند با به‌روزرسانی خود همچنان معتبر باقی بمانند و در هر لحظه نزدیک‌ترین جفت را به‌درستی نشان دهند.

۱.۳.۲ الگوریتم ایستای پیمایش صفحه

ایده‌ی اصلی الگوریتم ایستای یافتن نزدیک‌ترین جفت، تقسیم فضای اطراف هر نقطه، به شش قاش 60° است. شکل ۸.۲، قسمت (الف) را ببینید. بدیهی است که در این حالت می‌توان نزدیک‌ترین همسایه در هر قاش را یافت و سپس نزدیک‌ترین از بین آن‌ها، نزدیک‌ترین نقطه به نقطه‌ی مورد بررسی خواهد بود. نشان داده می‌شود که یک تعریف تقریبی از نزدیک‌ترین همسایه در هر قاش، کافی است تا نزدیک‌ترین جفت را تعیین کند. یک تعریف مناسب، اجازه می‌دهد تا همسایه‌ها به‌صورت کارآمد محاسبه شوند و نیز هنگام وابسته به حرکت کردن الگوریتم کمک می‌کند.

^{۲۲}Wedge



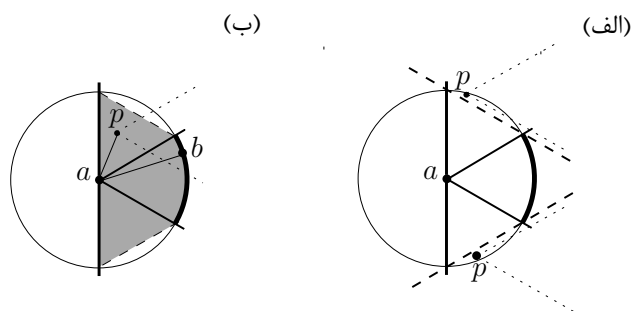
شکل ۸.۲: قاش نفوذ یک نقطه

قاش نفوذ^{۲۴} نقطه‌ی p به این شکل تعریف می‌شود: قاش گسترش یافته از سمت راست p ، که مرزهای بالا و پایین آن، به ترتیب زوایای $+30^\circ$ و -30° با محور x مثبت می‌سازند و با نماد $Dom(p)$ نمایش داده می‌شود (شکل ۸.۲، قسمت (ب)). قاش نفوذ یک نقطه از بالا بسته و از پایین باز است (مرز بالایی خود را شامل می‌شود اما مرز پایینی‌اش را شامل نمی‌شود). برای یافتن نزدیک‌ترین جفت کافی است، برای هر نقطه، نزدیک‌ترین همسایه به آن که در سمت راست‌اش واقع شده است (واقع در سه قاش گسترش یافته از سمت راست) را یافته و سپس از بین تمام این نزدیک‌ترین‌ها، نزدیک‌ترین جفت نهایی را یافت. درستی این روش به این دلیل است که اگر نزدیک‌ترین همسایه به نقطه‌ی دلخواه p ، در سه قاش گسترش یافته از سمت چپ‌اش، یعنی q باشد، آنگاه هرچند در زمان مشاهده‌ی سه قاش سمت راست p ، نقطه‌ی q به‌عنوان نزدیک‌ترین به آن انتخاب نمی‌شود، ولی هنگام مشاهده‌ی q ، نقطه‌ی p در سمت راست q قرار گرفته و به‌عنوان نزدیک‌ترین نقطه به آن انتخاب می‌شود.

بنابراین کافی است از روش پیمایش صفحه‌ای^{۲۵}، که صفحه را از سمت راست به چپ با خط عمودی l پیمایش می‌کند، استفاده کرد. برای توضیحات بیش‌تر ([۲۴]، فصل دو) را ببینید. بنابراین پیمایش روی نقاط از سمت راست صفحه به سمت چپ انجام می‌شود و زمانی که خط عمودی پیمایش صفحه به نقطه‌ی p می‌رسد، برای یافتن نزدیک‌ترین نقطه از سمت راست به نقطه‌ی p ، بررسی سه قاش گسترش یافته از سمت راست آن نقطه کافی است. اما الگوریتم تنها برای قاش‌ای که $Dom(p)$ است اجرا می‌شود و برای دو قاش دیگر که در بالا و پایین $Dom(p)$ هستند، می‌توان با دو بار چرخش $\pm 60^\circ$ صفحه، هر کدام از این دو قاش را در وضعیت $Dom(p)$ قرار داد و الگوریتم را مجدداً اجرا کرد. به‌عبارتی دیگر الگوریتم برای استفاده از این سه قاش سه بار اجرا می‌شود: یک‌بار روی صفحه بدون چرخش، یک‌بار روی صفحه با چرخش $+60^\circ$ و یک‌بار هم روی صفحه

^{۲۴}Dominance wedge

^{۲۵}Plane sweep



شکل ۹.۲: شکل اثبات لم ۱.۳.۲

با چرخش 60° -. در این بخش، $Circ(p, r)$ دایره‌ای با شعاع r و مرکز p و $\|pq\|$ فاصله‌ی بین دو نقطه‌ی p و q را نشان می‌دهد. همچنین اگر p در سمت چپ q باشد (یا در صورتی که مولفه‌ی x آن‌ها یکسان باشد در پایین q باشد)، برای راحتی این ویژگی با $q < p$ نمایش داده می‌شود.

اگر (a, b) نزدیک‌ترین جفت از بین مجموعه نقاط S باشند و ویژگی $a < b$ هم برقرار باشد. بدون از دست دادن کلیت مسئله فرض می‌شود $b \in Dom(a)$ ؛ اگر این برقرار نباشد، صفحه‌ی $60^\circ \pm$ چرخیده شده‌ای در نظر گرفته می‌شود که b در $Dom(a)$ قرار می‌گیرد.

لم ۱.۳.۲ اگر (a, b) نزدیک‌ترین جفت در S و $b \in Dom(a)$ ، آنگاه برای هر نقطه‌ی سومی مانند p که $a < p$ ، نقطه b در $Dom(p)$ واقع نمی‌شود.

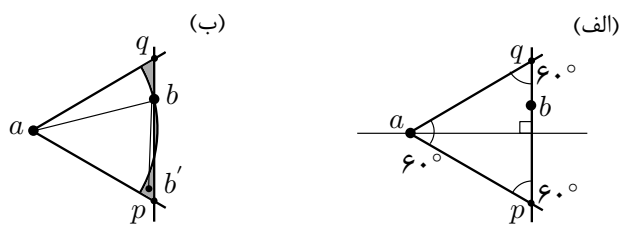
اثبات. نقطه‌ی b می‌تواند هر نقطه از کمان $Circ(a, ab) \cap Dom(a)$ باشد، که این کمان در شکل ۹.۲ مشخص است. اگر فرض خلف گرفته شود که b در $Dom(p)$ واقع شود، از آنجایی که $a < p$ است، پس نقطه‌ی p باید در ناحیه بین خط عمود گذرنده از a و کمان $Circ(a, ab) \cap Dom(a)$ قرار گیرد. علاوه بر آن نقطه‌ی p باید در ناحیه‌ی بین خطوط باز $30^\circ \pm$ گذرنده از بالا و پایین کمان $Circ(a, ab) \cap Dom(a)$ نیز باشد (اگر نقطه p بالای خط $30^\circ +$ یا پایین خط $30^\circ -$ قرار گیرد شرط $b \in Dom(p)$ برقرار نخواهد بود. شکل ۹.۲، قسمت (الف) را ببینید). بنابراین ناحیه‌ای که نقطه p می‌تواند در آن واقع شود (ناحیه‌ی تیره رنگ در شکل ۹.۲، قسمت (ب)) کاملاً در داخل دایره‌ی $Circ(a, ab)$ قرار می‌گیرد. در نتیجه $ap < ab$ و این با نزدیک‌ترین جفت بودن (a, b) متناقض است. \square

لم ۲.۳.۲ اگر (a, b) نزدیک‌ترین جفت در S و $b \in Dom(a)$ ، آنگاه سمت چپ‌ترین نقطه از S در $Dom(a)$ ، نقطه‌ی b است.

اثبات. (فرض خلف) فرض کنید b' سمت چپ‌ترین نقطه‌ی S در $Dom(a)$ باشد و $b' \neq b$ در نظر بگیرید. $r = \|ab\|$ فاصله‌ی نزدیک‌ترین جفت (a, b) از هم باشد. از آن جایی که (a, b) نزدیک‌ترین جفت هستند، پس ab' باید بیش‌تر از r باشد. بنابراین b' باید در ناحیه‌ای که خارج از $Circ(a, r)$ است واقع شود، این ناحیه A نامیده می‌شود. چون b' سمت چپ‌ترین نقطه از S در $Dom(a)$ نسبت به b است، پس باید در مثلث حاصل از اشتراک $Dom(a)$ با نیم صفحه‌ی سمت چپ خط عمود گذرنده از b ، قرار گیرد. این ناحیه B نامیده می‌شود که یک مثلث متساوی‌الاضلاع است. شکل ۱۰.۲، قسمت (الف) را ببینید. ناحیه‌ی C حاصل از اشتراک ناحیه‌ی A و ناحیه‌ی B را در نظر بگیرید. همان ناحیه‌ی تیره رنگ در شکل ۱۰.۲، قسمت (ب). دورترین نقطه از b واقع در ناحیه‌ی C ، نقطه p یا نقطه q است که حاصل برخورد خط عمود گذرنده از b با مرزهای $Dom(a)$ است (زیرا پاره خط‌هایی که از b به مرزهای $Dom(a)$ عمود می‌شوند، در داخل $Circ(a, r)$ قرار می‌گیرند). بدون از دست دادن کلیت مسئله فرض می‌شود که نقطه‌ی p از b دورتر است، یعنی $pb \geq qb$. مثلث abp را در نظر بگیرید. با استفاده از قانون سینوس‌ها،

$$pb/ab = pb/r = \sin(\angle pab)/\sin(\angle apb) = \sin(\angle pab)/\sin 60^\circ$$

چون $\angle pab \leq 60^\circ$ ، پس $pb \leq r$ و چون $bb' \leq pb$ پس نتیجه می‌شود، $bb' \leq r$ و از طرفی چون طبق تعریف، $Dom(a)$ نسبت به مرز پایینی خود باز است، نتیجه می‌شود که $b' \neq p$ ، پس $bb' < r$ ، یعنی (b, b') نزدیک‌ترین جفت هستند و این با نزدیک‌ترین جفت بودن (a, b) متناقض است. \square



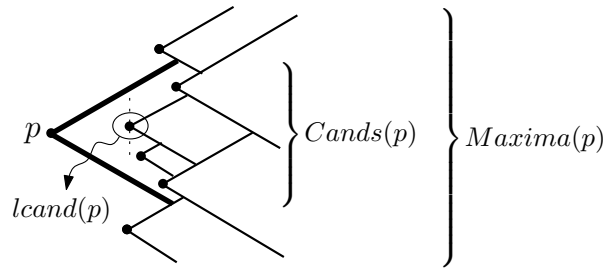
شکل ۱۰.۲: شکل اثبات لم ۲.۳.۲

برای هر نقطه‌ی p ، مجموعه نقاط **ماکسیمای** آن به صورت مجموعه‌ای از نقاط S که روی مرز

$$\bigcup Dom(q)$$

$$q \in S$$

$$p \prec q$$



شکل ۱۱.۲: مجموعه نقاط ماکسیما و کاندیدای نقطه‌ی p و نقطه‌ی $lcand(p)$

هستند، تعریف می‌شود و با $Maxima(p)$ نمایش داده می‌شود. به عبارت دیگر، این مجموعه، تمام نقاط سمت راست p که در قاش نفوذ هیچ نقطه‌ی دیگری در سمت راست p واقع نشده‌اند را شامل می‌شود. مجموعه نقاط کاندید برای نقطه‌ی p به صورت مجموعه‌ی $Maxima(p) \cap Dom(p)$ تعریف می‌شود و با $Cands(p)$ نشان داده می‌شود. سمت چپ‌ترین نقطه‌ی این مجموعه با $lcand(p)$ نمایش داده می‌شود. شکل ۱۱.۲ را ببینید. با توجه به لم‌های ۱.۳.۲ و ۲.۳.۲ می‌توان گفت، هرگاه (a, b) نزدیک‌ترین جفت نقاط باشند و $b \in Dom(a)$ ، آنگاه $b = lcand(a)$ است. از این، نتیجه می‌شود که نزدیک‌ترین جفت به صورت $(a, lcand(a))$ است.

الگوریتم ایستای پیمایش صفحه برای یافتن نزدیک‌ترین جفت:

۱. لیست $Maxima$ شامل نقاط مرتب شده بر حسب y را با تهی مقداردهی اولیه کن. (هرچند که لیست $Maxima$ در شروع اجرای الگوریتم تهی است اما در گام‌های بعدی، در شروع هر تکرار حلقه‌ی زیر برای هر نقطه‌ی p شامل نقاط $Maxima(p)$ خواهد بود.)

۲. برای هر نقطه‌ی $p \in S$ از راست به چپ تکرار کن:

(الف) قرار بده $Cands(p) = Maxima \cap Dom(p)$.

(ب) سمت چپ‌ترین نقطه از $Cands(p)$ را در $lcand(p)$ قرار بده.

(ج) نقاط $Cands(p)$ را از لیست $Maxima$ پاک کن.

(د) نقطه p را در لیست $Maxima$ در محل درستش بر اساس مولفه y درج کن.

این الگوریتم برای سه جهت ذکر شده (صفحه بدون چرخش و صفحه با چرخش‌های $\pm 60^\circ$) سه بار اجرا می‌شود و سه مجموعه تولید می‌کند. هر مجموعه شامل n جفت $(p, lcand(p))$ است (n تعداد نقاط مجموعه S) که یکی از این سه مجموعه، نزدیک‌ترین جفت را دربر می‌گیرد. به عبارتی پس از سه بار اجرای الگوریتم از بین $3n$

جفت به دست آمده نزدیکترین جفت تعیین می شود.

تحلیل مرتبه‌ی زمانی الگوریتم:

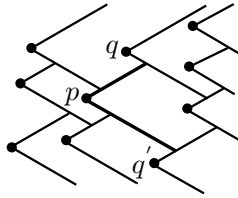
الگوریتم پیمایش صفحه می تواند در زمان $O(n \log n)$ اجرا شود. این الگوریتم نقاط را از سمت راست صفحه به سمت چپ پیمایش می کند. به عبارتی نقاط بر اساس ترتیب \leftarrow پیمایش می شوند. پس مرتب‌سازی نقاط S با ترتیب گفته شده، $O(n \log n)$ زمان می برد. لیست *Maxima* با ساختار درخت دودویی متوازن پیاده‌سازی می شود، به طوری که تمام نقاط متعلق به آن علاوه بر این که در نودهای داخلی درخت دودویی به منظور راهنمای جستجو قرار می گیرند، مرتب شده بر اساس y در برگ‌ها نیز قرار می گیرند. (بدین منظور مقدار هر نود داخلی در سمت راست‌ترین برگ زیردرخت چپ‌اش قرار می گیرد.) بنابراین زمان لگاریتمی (به اندازه‌ی عمق درخت) برای انجام عملیات جستجو، درج و حذف در این ساختار نیاز است [۲۲].

محاسبه‌ی $Cands(p)$ به دو زمان جستجوی دودویی $O(\log n)$ روی ساختار *Maxima* نیاز دارد، زیرا $Cands(p)$ یک زیردنباله‌ی پشت سرهم از نقاط *Maxima* است. پس کافی است با یک جستجو روی ساختار *Maxima* به کمک معادله‌ی خط مرز بالایی $Dom(p)$ ، بالاترین نقطه در مجموعه $Cands(p)$ را یافت و با جستجوی مشابه روی ساختار *Maxima* به کمک معادله‌ی خط مرز پایینی $Dom(p)$ ، پایین‌ترین نقطه در مجموعه‌ی $Cands(p)$ را نیز یافت. سپس تمام نقاط بین این دو نقطه در ساختار *Maxima* را در مجموعه‌ی $Cands(p)$ قرار داد.

پیدا کردن $lcand(p)$ به اندازه‌ی $|Cands(p)|$ زمان می برد، اما چون هر نقطه‌ای از *Maxima* تنها یکبار به عنوان $Cands()$ قرار می گیرد و سپس از ساختار *Maxima* حذف می شود، بنابراین کل زمانی، که برای پیدا کردن سمت چپ‌ترین نقاط کل مجموعه‌های کاندید لازم است برابر $O(n)$ می شود. حذف نقاط $Cands(p)$ از *Maxima* و درج نقطه‌ی p در مکان درستش، برای هر نقطه زمان $O(\log n)$ می برد. بنابراین زمان اجرای کل برابر $O(n \log n)$ می شود.

۲.۳.۲ وابسته به حرکت کردن

زمانی که نقاط مجموعه‌ی S دارای حرکت مداوم هستند، لازم است الگوریتم ایستای پیمایش صفحه، وابسته به حرکت شود. بدین منظور نیاز است تا: (الف) ساختمان داده‌ی مناسبی برای نگهداری نتیجه‌ی عملکرد الگوریتم پیمایش صفحه (خروجی الگوریتم ایستا) تعیین شود. (ب) همچنین به یک مجموعه از تاییدیه‌ها نیاز است تا نشان دهد که آن ساختمان داده برای وضعیت فعلی مجموعه نقاط معتبر است. برای مورد (الف) خروجی



شکل ۱۲.۲: نقاط q و q' مقصدهای نقطه p در نمودار ماکسیما

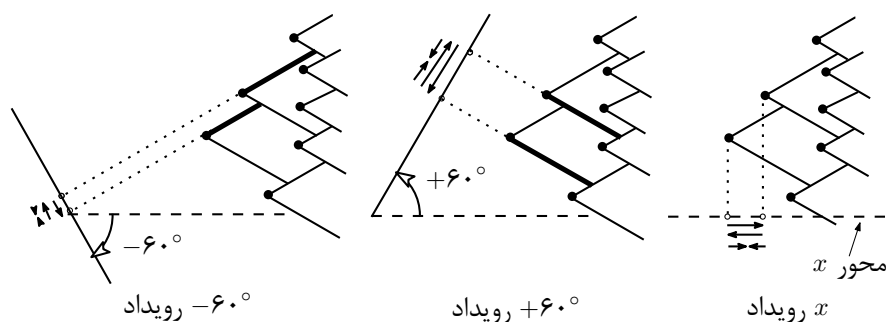
الگوریتم پیمایش صفحه، نمودار ماکسیماست که در ادامه تعریف می‌شود. برای مورد (ب) نیز ۳ نوع تاییدیه معرفی خواهند شد.

برای هر نقطه‌ی p ، بخشی از مرز $Dom(p)$ که خارج از $\bigcup_{q \in Maxima(p)} Dom(q)$ واقع می‌شود را در نظر بگیرید. اجتماع این مرزها برای تمام نقاط نمودار ماکسیما^{۲۶} را تعریف می‌کند. هر نقطه‌ی p از S ، نقطه‌ی انتهایی چپ دو پاره‌خط در نمودار ماکسیما است که این پاره‌خطها از p به مرزهای $Dom(q)$ و $Dom(q')$ ختم شده‌اند، به طوری که q و q' دو نقطه در $Maxima(p)$ هستند و گفته می‌شود که q و q' مقصدهای p در نمودار ماکسیما هستند. شکل ۱۲.۲ را ببینید. به q که پاره‌خط بالایی p به مرز Dom آن ختم شده است مقصد بالایی و به q' که پاره‌خط پایینی p به مرز Dom آن ختم شده است مقصد پایینی گفته می‌شود.

برای وابسته به حرکت کردن الگوریتم، لازم است تاییدیه‌هایی معرفی شوند که موقعیت نقاط در حال حرکت را نسبت به هم بیان کنند و نیز صفی از رویدادها متناظر با این تاییدیه‌ها شکل بگیرد. سه ترتیب مرتب شده به‌عنوان تاییدیه‌ها استفاده می‌شوند: ترتیب نقاط S زمانی که روی محور x تصویر شده‌اند و ترتیب نقاط زمانی که روی خطوطی که زاویه‌ی $\pm 60^\circ$ با محور x می‌سازند، تصویر شده‌اند. هر نقطه در یک ترتیب خاص در دو تاییدیه برای نشان دادن وضعیتش قرار می‌گیرد. یک تاییدیه برای همسایه‌ی چپ و یک تاییدیه برای همسایه‌ی راست. از طرفی برای نقاط، سه ترتیب خاص یا به‌عبارتی دیگر سه لیست متفاوت از تاییدیه‌ها وجود دارد. بنابراین هر نقطه، حداکثر متعلق به شش تاییدیه است. همچنین برای وابسته به حرکت کردن الگوریتم ایستا، به طوری که نزدیکترین جفت را بدهد به یک عنصر دیگر نیز نیاز است، یک درخت مسابقه‌ی وابسته به حرکت تا نتایج حاصل از سه‌بار اجرای الگوریتم را برای تعیین نزدیک‌ترین جفت (کم‌ترین فاصله‌ی متناظر با آن) نگهداری کند. بنابراین تاییدیه‌هایی برای گره‌های داخلی این درخت نیاز خواهد شد. برای توضیحات بیش‌تر در مورد مسابقه‌ی وابسته به حرکت بخش ۲.۲ را ببینید.

زمانی یک رویداد رخ می‌دهد که تصویر دو نقطه روی خطوط $+60^\circ$ یا -60° ، در اثر حرکت به هم برسند

^{۲۶}Maxima diagram



شکل ۱۳.۲: سه نوع رویداد متفاوت

و از هم عبور کنند. (نامعتبر شدن تاییدیه ترتیب آن‌ها) شکل ۱۳.۲ را ببینید. این منجر به تعریف سه نوع رویداد می‌شود: (در نظر بگیرید که در نمودار ماکسیما تمام مرزهای بالایی $Dom()$ ها با هم و نیز تمام مرزهای پایینی $Dom()$ ها با هم موازیند).

۱. x رویداد: زمانی که تصویر دو نقطه روی محور x به هم می‌رسند، و با هم جابه‌جا می‌شوند.

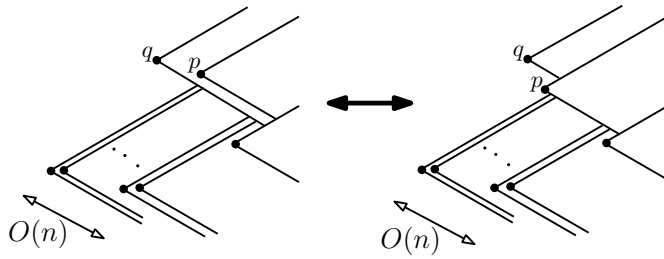
۲. $+60^\circ$ رویداد: تصویر دو نقطه روی خط $+60^\circ$ به هم می‌رسند و با هم جابه‌جا می‌شوند (زمانی که مرزهای پایینی $Dom()$ های دو نقطه به هم می‌رسند و با هم جابه‌جا می‌شوند).

۳. -60° رویداد: تصویر دو نقطه روی خط -60° به هم می‌رسند و با هم جابه‌جا می‌شوند (زمانی که مرزهای بالایی $Dom()$ های دو نقطه به هم می‌رسند و با هم جابه‌جا می‌شوند).

لم ۳.۳.۲ اگر دو شکل مختلف از نقاط S در صفحه دارای سه ترتیب برابر برای نقاط باشند (سه لیست تاییدیه یکسان در دو شکل)، آنگاه برای هر نقطه‌ی p ، $Maxima(p)$ ، $Cands(p)$ و $lcand(p)$ در دو شکل یکسان است.

اثبات. با استقرا روی نقاط S ، از راست به چپ به ترتیب \prec ، درستی لم بررسی می‌شود. برای هر نقطه‌ی p فرض می‌شود که $Maxima(p)$ در دو شکل یکسان است و نقاط در $Maxima(p)$ به ترتیب y مرتب شده‌اند. اثبات می‌شود که $Cands(p)$ و $lcand(p)$ هم در دو شکل یکسان هستند و سرانجام نشان داده می‌شود که $Maxima(p')$ برای p' نقطه‌ای که بعد از p باید پردازش شود ($p' \prec p$)، در دو شکل یکسان است.

فرض می‌شود که $Maxima(p)$ در دو شکل یکسان است. از آنجایی که ترتیب $+60^\circ$ یکسان برای همه نقاط و از جمله نقاط $Maxima(p)$ در دو شکل وجود دارد، نتیجه می‌شود که نقطه p مقصد پایینی یکسان در



شکل ۱۴.۲: عدم پاسخ‌گویی ساختار هنگام پیاده‌سازی صریح نمودار ماکسیما

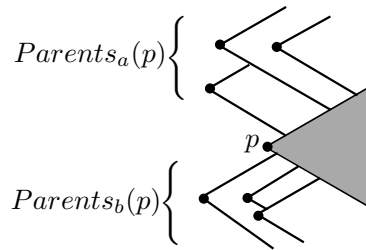
هر دو نسخه $Maxima(p)$ دارد و نیز از آن‌جایی که ترتیب -60° یکسان برای همه‌ی نقاط و از جمله نقاط $Maxima(p)$ در دو شکل وجود دارد، نتیجه می‌شود که نقطه‌ی p مقصد بالایی یکسان هم در هر دو نسخه از $Maxima(p)$ دارد. از آن‌جایی که نقاط متعلق به مجموعه‌ی $Cands(p)$ نقاطی از $Maxima(p)$ هستند که بین دو مقصد نقطه‌ی p واقع شده‌اند. بنابراین $Cands(p)$ در دو شکل یکسان است و چون ترتیب x هم برای همه‌ی نقاط در دو شکل یکسان است، در نتیجه $lcand(p)$ هم در دو شکل یکسان می‌شود. مجموعه‌ی $Maxima(p')$ از مجموعه‌ی $Maxima(p)$ با حذف نقاط $Cands(p)$ ، و سپس درج p بین دو مقصدش به دست می‌آید؛ بنابراین $Maxima(p')$ هم در دو شکل یکسان است. \square

با استفاده از لم بالا می‌توان حرکت پیوسته‌ی نقاط را تبدیل به زمان‌های گسسته کرد. به عبارتی دیگر برای وابسته به حرکت کردن الگوریتم لازم نیست که در هر لحظه ساختارهای الگوریتم را به‌روزرسانی کرد، بلکه تنها با به‌روزرسانی ساختارها در لحظه‌های بروز یک رویداد (نامعتبرشدن یک تاییدیه) می‌توان در هر لحظه تابع وضعیت مسئله (در این‌جا نزدیک‌ترین جفت) برای نقاط در حال حرکت را به‌درستی نگهداری کرد.

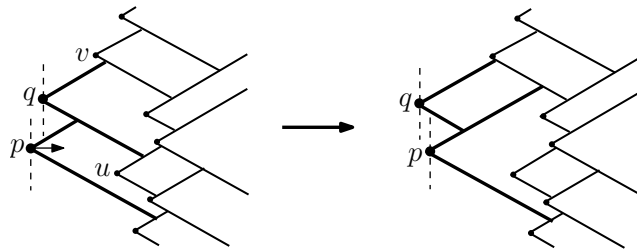
نمودار ماکسیما اگر به‌صورت صریح پیاده‌سازی شود، در زمان یک رویداد می‌تواند تحت یک تعداد خطی از تغییرات قرار گیرد و به‌روزرسانی آن زمان خطی طول بکشد. شکل ۱۴.۲ را ببینید. همان‌طور که در این شکل مشخص شده است با رخ دادن این رویداد $+60^\circ$ ، نمودار ماکسیما باید $O(n)$ تعداد از تغییرات را برای به‌روزرسانی خود اعمال کند. در این حالت ساختمان داده‌ی وابسته به حرکت دیگر پاسخ‌گو نخواهد بود. برای رفع این مشکل، تغییرات در نمودار ماکسیما می‌تواند با یک ساختمان داده‌ی ضمنی که تنها نیاز به $O(\log n)$ زمان برای به‌روزرسانی خود در هر رویداد دارد، پیاده‌سازی شود. بدین منظور، دو ساختمان داده‌ی کمکی برای هر نقطه‌ی $p \in S$ نگهداری می‌شود. یک ساختمان داده برای نگهداری نقاط $Cands(p)$ که با همین نام $Cands(p)$ نیز معرفی خواهد شد و یک ساختمان داده‌ی $Parents(p)$ که در ادامه به‌طور کامل معرفی می‌گردد.

۱. $Cands(p)$ اشتراک $Maxima(p)$ با $Dom(p)$ را دربر می‌گیرد و به صورت یک دنباله‌ی مرتب شده از نقاط بر اساس مولفه‌ی y می‌باشد. (این ترتیب برای نقاط $Cands(p)$ با ترتیبی که برای این نقاط در امتداد جهت‌های $\pm 60^\circ$ وجود دارد یکسان است ولی با ترتیب x نقاط تفاوت دارد.) این دنباله در یک درخت دودویی متوازن ذخیره شده است، که این درخت عملیات معمول جستجو و به‌روزرسانی را پشتیبانی می‌کند. علاوه‌برآن، هر گره درخت یک اشاره‌گر به پدرش در درخت دارد و ریشه‌ی درخت $Cands(p)$ به نقطه‌ی p اشاره می‌کند. بنابراین هر نقطه‌ی $q \in S$ می‌تواند نقطه‌ی $p \in S$ به‌طوری‌که $q \in Cands(p)$ را در زمان $O(\log n)$ پیدا کند. همچنین هر گره داخلی در یک درخت $Cands()$ سمت چپ‌ترین نقطه (به ترتیب x) در زیردرختش را با خود ذخیره می‌کند، بنابراین ریشه‌ی $Cands(p)$ نقطه‌ی $lcand(p)$ را با خود ذخیره خواهد کرد. اشاره‌گرهای پدر می‌توانند به‌عنوان بخشی از درخت استاندارد نگهداری شوند و زمان عملیات به‌روزرسانی در این حالت با زمان عملیات به‌روزرسانی در درخت استاندارد دارای محدوده‌ی یکسان است. همچنین برای فیلدهای سمت چپ‌ترین نیز این روند برقرار است. چنان‌که بخشی از الگوریتم، تضمین می‌کند که هر زمان که ترتیب x نقاط عوض می‌شود، فیلدهای سمت چپ‌ترین نیز به‌درستی نگهداری شوند.

۲. $Parents(p)$ تمام نقاطی که p یک مقصد برای آن‌ها در نمودار ماکسیما است را ذخیره می‌کند. $Parents(p)$ یک دنباله‌ی مرتب شده از (نقاط متناظر با) یال‌هایی است که از سمت چپ در نمودار ماکسیما با $Dom(p)$ برخورد دارند. این دنباله، طبق ترتیبی که یال‌ها با $Dom(p)$ برخورد دارند، مرتب شده است. این ترتیب نیازی ندارد که متناظر با ترتیب y باشد. دنباله می‌تواند به نقاط بالای p و نقاط پایین p تقسیم شود. که به ترتیب با $Parents_a(p)$ و $Parents_b(p)$ نشان داده می‌شوند. شکل ۱۵.۲ را ببینید. $Dom(p)$ مقصد برای یال‌های پایینی گسترش یافته از تمام عناصر $Parents_a(p)$ ، و همچنین مقصد برای یال‌های بالایی گسترش یافته از تمام عناصر $Parents_b(p)$ است. در هر دو زیردنباله‌ی $Parents_a(p)$ و $Parents_b(p)$ ترتیب نقاط (ترتیبی که یال‌های آن‌ها با $Dom(p)$ برخورد کرده است) با ترتیب x ایشان یکسان است. دنباله‌ی $Parents(p)$ در یک درخت دودویی متوازن ذخیره می‌شود و نیز برای هر گره درخت اشاره‌گری به پدرش قرار داده می‌شود. به‌طوری‌که برای هر دو یال گسترش یافته از نقطه‌ی q در نمودار ماکسیما، می‌توان نقطه‌ی p را به‌طوری‌که $q \in Parents(p)$ در زمان لگاریتمی یافت.



شکل ۱۵.۲: مجموعه نقاط $Parents(p)$ و تقسیم‌بندی آن به دو دسته‌ی $Parents_a(p)$ و $Parents_b(p)$



شکل ۱۶.۲: یک رویداد x و تغییر در مجموعه‌های $Cands$

این‌ها تنها ساختمان داده‌هایی هستند که در وابسته به حرکت کردن الگوریتم استفاده می‌شوند. به‌ویژه، از ساختمان داده $Maxima$ که در الگوریتم ایستا توصیف شد، استفاده نمی‌شود. طرح الگوریتمی زیر نشان می‌دهد که چه‌طور تمام ساختارهای تحت تاثیر واقع شده $Cands()$ ، $Parents()$ و نیز فیلدهای $lcand()$ زمانی که موقعیت x دو نقطه‌ی p و q باهم عوض می‌شود (یک رویداد اتفاق می‌افتد)، به‌روزرسانی شوند. بدون از دست دادن کلیت مسئله، فرض می‌شود که قبل از عوض شدن p و q ، $p \prec q$ (در سمت چپ q است). علاوه‌بر آن، فرض می‌شود که در لحظه‌ی عوض شدن p پایین q است (شبه‌کد زمانی که p بالای q است به‌طور مشابه به‌کار می‌رود). شکل ۱۶.۲ را ببینید.

به‌روزرسانی ساختارها هنگام رخ دادن یک رویداد

۱. اگر $p \in Parents(q)$ ، به‌ویژه متعلق به $Parents_b(q)$ ، آنگاه انجام بده:

(الف) بخشی از $Cands(q)$ که داخل $Dom(p)$ است را جدا کن و آن را به بالای $Cands(p)$ الحاق کن.

(ب) فرض کن که u نقطه‌ای باشد که q متعلق به $Parents_a$ آن باشد. نقطه‌ی q را از $Parents_a(u)$ حذف کن و آن را در $Parents_a(p)$ درج کن.

(ج) دنباله‌ی تغییر یافته‌ی $Cands(q)$ ، از گام (الف) را در نظر بگیر، در صورت غیرتهی بودن این دنباله، v را پایین‌ترین نقطه‌ی آن بگیر، در غیر این صورت v را نقطه‌ای بگیر که q متعلق به $Parents_b$ آن باشد. p را از $Parents_b(q)$ حذف کن و آن را در $Parents_b(v)$ درج کن.

۲. فرض کن p' و q' نقاطی باشند که p متعلق به $Cands(p')$ و q متعلق به $Cands(q')$ باشد. اگر $p' = q'$ آنگاه $lcand(p')$ را با شروع از p و q در درخت $Cands(p')$ به‌روزرسانی کن.

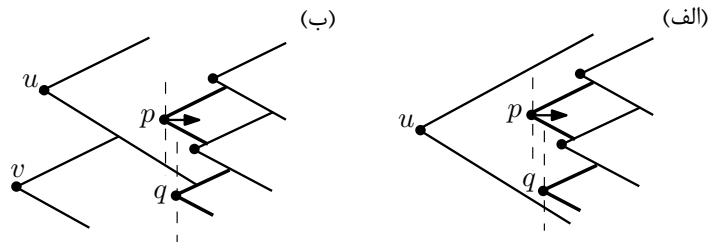
با استفاده از دو ساختمان داده‌ی کمکی $Cands()$ ، $Parents()$ که برای هر نقطه جداگانه تعریف می‌شود، به‌روزرسانی این ساختارها تنها به زمان لگاریتمی (ارتفاع درخت) نیاز دارد و مشکل ذکر شده در هنگام پیاده‌سازی صریح نمودار ماکسیما برطرف می‌شود؛ زیرا کافی است مانند گام ۱. (الف) با عملیات جداسازی^{۲۷} و الحاق^{۲۸} که در زمان لگاریتمی انجام می‌شوند [۲۲]، ساختارها را به‌روزرسانی کرد. از آنجایی که با بروز یک x رویداد قسمت‌هایی از ساختار نامعتبر شده و درستی خود را از دست می‌دهند، در ادامه، با لم زیر اثبات خواهد شد که با اجرای شبه‌کد بالا برای به‌روزرسانی ساختار نامعتبر شده، ساختار به‌درستی به‌روزرسانی خواهد شد و مجدداً مجموعه‌ی تاییدیه‌ها پس از به‌روزرسانی، درستی تابع وضعیت را نشان خواهند داد.

لم ۴.۳.۲ بعد از انجام رویه‌ی ذکر شده برای به‌روزرسانی ساختارهای $Cands()$ ، $Parents()$ ، و فیلدهای $lcand()$ زمانی که ترتیب x دو نقطه با هم عوض می‌شوند، ساختار وابسته به حرکت، نمودار ماکسیما را برای شکل فعلی نقاط در S به‌درستی نمایش می‌دهد و فیلدهای $lcand()$ نیز درست هستند.

اثبات. اگر q یک مقصد برای p باشد، تعویض x آن‌ها (رخ دادن یک x رویداد) نمودار ماکسیما را تغییر می‌دهد. به‌ویژه، نقاط $Maxima(q)$ که متعلق به $Dom(p) \cap Dom(q)$ هستند، از $Cands(q)$ به $Cands(p)$ انتقال داده می‌شوند. نقطه‌ی p یک مقصد جدید می‌گیرد؛ به‌طوری که نقطه‌ی جدید اتصال، بین پاره‌خط از p و مقصد جدیدش، داخل $Dom(q)$ واقع می‌شود. همچنین q ، p را به عنوان یک مقصد می‌گیرد. گام ۱ به این تغییرات دسترسی دارد. تنها یال‌های نمودار ماکسیما که تغییر می‌کنند یال‌هایی هستند که از سمت راست p و q گسترش یافته‌اند-هیچ مقصدی برای نقاط راست یا چپ $\{p, q\}$ تغییر نمی‌کند-بنابراین عملیات گام ۱ برای به‌روزرسانی نمودار ماکسیما کافی است.

^{۲۷}Split

^{۲۸}Join



شکل ۱۷.۲: دو x رویداد بدون نیاز به به‌روزرسانی فیلدهای $Cands()$ و $Parents()$

اگر p و q هیچ کدام مقصدی برای دیگری نباشد، آنگاه نمودار ماکسیما تغییر نمی‌کند و فیلدهای $Cands()$ و $Parents()$ نیاز به به‌روزرسانی ندارند. شکل ۱۷.۲ را ببینید. در این شکل، دو x رویداد که نیازی به به‌روزرسانی فیلدهای $Cands()$ و $Parents()$ ندارند، نشان داده شده‌اند و گام ۱ شبه‌کد برای آن‌ها اجرا نمی‌شود. در هر صورت نمودار ماکسیما تغییر کند یا نه، یک فیلد $lcand()$ ممکن است تغییر کند. اگر یک نقطه‌ی u وجود داشته باشد که $p, q \in Cands(u)$ ، لازم است مطمئن شد که فیلدهای سمت چپ‌ترین، در درخت دودویی $Cands(u)$ به‌روزرسانی می‌شوند، به طوری که هر مقایسه‌ی مرتبط با p و q در آن درخت مجدداً محاسبه می‌گردد؛ این ممکن است باعث تغییر $lcand(u)$ شود که گام ۲ این را برآورده می‌کند. برای نمونه، اجرای گام ۲ برای قسمت (الف) شکل ۱۷.۲، که p و q متعلق به یک مجموعه‌ی $Cands()$ هستند، لازم است.

شبه‌کد زیر به‌روزرسانی ساختار وابسته به حرکت را در زمان بروز یک $+60^\circ$ رویداد برای نقاط p و q پیاده‌سازی می‌کند. بدون از دست دادن کلیت مسئله، فرض می‌شود که p در سمت چپ q است. برای این رویداد دو حالت وجود دارد (شکل ۱۸.۲):

۱. q به $Dom(p)$ وارد شود،

۲. q از $Dom(p)$ خارج گردد.

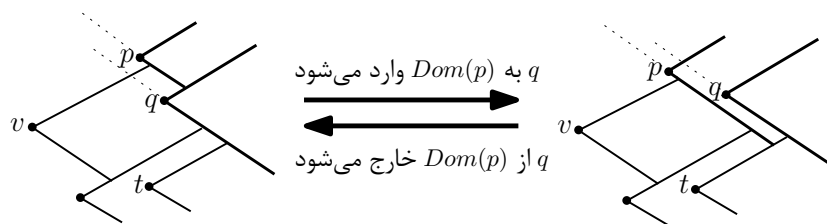
به‌روزرسانی ساختارها هنگام رخ دادن یک $+60^\circ$ رویداد

در مورد اول، q به $Dom(p)$ وارد می‌شود. شکل ۱۸.۲ را ببینید. انجام بده:

۱. اگر $p \in Parents_a(q)$ سپس انجام بده:

(الف) فرض کنید v نقطه‌ای باشد که q متعلق به $Cands$ آن باشد. q را از $Cands(v)$ حذف کن و

در $Cands(p)$ درج کن.



شکل ۱۸.۲: یک 60° + رویداد

(ب) دنباله‌ی $Parents_b(q)$ که در سمت راست p قرار دارد را در نظر بگیر. در صورت غیرتهی بودن آن نقطه‌ی t را سمت چپ‌ترین نقطه از این دنباله در نظر بگیر. در غیر این صورت t را نقطه‌ای بگیر که q متعلق به $Parents_a(t)$ باشد. (با توجه به این که ترتیب x برای نقاط $Parents_b(q)$ برابر است با ترتیبی که یال‌های آن‌ها با $Dom(q)$ برخورد می‌کنند.) p را از $Parents_a(q)$ حذف کن و آن را در $Parents_a(t)$ درج کن.

(ج) زیردنباله‌ای از $Parents_b(q)$ که در سمت چپ t (و در نتیجه در سمت چپ p) واقع شده‌اند را جدا کن و آن‌ها را به پایین $Parents_b(p)$ الحاق کن.

در مورد دوم، q از $Dom(p)$ خارج می‌شود. شکل ۱۸.۲ را ببینید. انجام بده:

۱. اگر $q \in Cands(p)$ آنگاه انجام بده:

(الف) فرض کن t نقطه‌ای باشد که p متعلق به $Parents_a$ آن باشد. نقطه‌ی p را از $Parents_a(t)$ حذف کن و آن را در $Parents_a(q)$ درج کن.

(ب) نقاطی از $Parents_b(p)$ که یال‌های آن‌ها در پایین q با $Dom(p)$ برخورد کرده‌اند را جدا کن و آن‌ها را به بالای $Parents_b(q)$ الحاق کن.

(ج) دنباله‌ی تغییر یافته‌ی $Parents_b(p)$ از گام (ب) را در نظر بگیر. در صورت غیرتهی بودن این دنباله، نقطه‌ی v را سمت راست‌ترین نقطه از آن بگیر. در غیر این صورت v را نقطه‌ای بگیر که p متعلق به $Cands$ آن باشد. q را از $Cands(p)$ حذف کن و آن را در $Cands(v)$ درج کن.

از آن جایی که با بروز یک 60° + رویداد قسمت‌هایی از ساختار نامعتبر شده و درستی خود را از دست می‌دهند، در ادامه، با لم زیر اثبات خواهد شد که با اجرای شبه‌کد بالا برای به‌روزرسانی ساختار نامعتبر شده، ساختار

به درستی به روزرسانی خواهد شد و مجدداً مجموعه‌ی تاییدیه‌ها پس از به روزرسانی، درستی تابع وضعیت را نشان خواهند داد.

لم ۵.۳.۲ بعد از انجام رویه‌ی ذکر شده برای به روزرسانی ساختارهای $Cands()$ ، $Parents()$ و فیلدهای $lcand()$ زمانی که دو نقطه از S در ترتیب $+60^\circ$ با هم عوض می‌شوند، ساختار وابسته به حرکت، نمودار ماکسیما را برای شکل فعلی S به درستی نمایش می‌دهد و فیلدهای $lcand()$ نیز درست هستند.

اثبات. مورد اول که q به $Dom(p)$ وارد می‌شود را در نظر بگیرید. اگر $p \notin Parents_a(q)$ ، سپس $q \notin Maxima(p)$ و تعویض p و q در ترتیب $+60^\circ$ نمودار ماکسیما را تحت تاثیر قرار نمی‌دهد و به روزرسانی هیچ ساختمان داده‌ای هم لازم نیست.

اگر $p \in Parents_a(q)$ باشد، نمودار ماکسیما فقط در همسایگی p و q تغییر می‌کند. تعویض p و q هر دو مقصد p که در سمت راست آن واقع شده‌اند را تغییر نمی‌دهد بلکه تنها مقصد پایینی p نیاز به به روزرسانی دارد که گام ۱. (ب) این را برآورده می‌کند. از بین نقاط سمت چپ p ، تنها آن نقاطی که q به عنوان مقصد بالایی‌اشان است (یعنی، اعضای $Parents_b(q)$) نیاز دارند تا مقصدهایشان را به p تغییر دهند که گام ۱. (ج) این را برآورده می‌کند. مجموعه $Cands()$ برای p (چون q به آن وارد می‌شود) و نیز برای نقطه‌ی v که q از مجموعه‌ی $Cands()$ آن خارج می‌شود، تغییر می‌کند. که گام ۱. (الف) این را برآورده می‌کند. فیلدهای سمت چپ‌ترین درخت‌های دودویی $Cands()$ هم‌زمان با سایر اصلاحات به روزرسانی می‌شوند، به طوری که $lcand(p)$ و $lcand(v)$ درست نگهداری شوند. لیست‌های $Cands()$ و $Parents()$ برای این که ساختار ترکیبی نمودار ماکسیما را مشخص کنند، کافی هستند و چون این ساختارها به درستی نگهداری می‌شوند، بنابراین نمودار ماکسیما هم به درستی نگهداری می‌شود.

در مورد دوم که q از $Dom(p)$ خارج می‌شود، تغییرات نمودار ماکسیما برعکس تغییرات در مورد اول است. برای رویه‌ی به روزرسانی این مورد، کافی است عملیات مربوط به به روزرسانی در مورد اول برعکس شود. بنابراین رویه‌ی به روزرسانی در مورد دوم هم درست است. \square

رویه برای تعویض دو نقطه در ترتیب -60° مشابه با تعویض دو نقطه در ترتیب $+60^\circ$ است. واضح است که هر کدام از اعمال به روزرسانی روی ساختمان داده‌های کمکی $Cands()$ ، $Parents()$ و $lcand()$ دارد، زمان $O(\log n)$ طول می‌کشد؛ زیرا هر کدام تعداد ثابتی عملیات استاندارد روی درخت دودویی متوازن را شامل می‌شوند.

آخرین عنصر ساختمان داده‌ی وابسته به حرکت، یک درخت مسابقه‌ی وابسته به حرکت روی $3n$ فاصله‌ی متناظر با جفت‌های $(p, lcand(p))$ است (این $3n$ تاییدیه را به ساختار وابسته به حرکت اضافه می‌کند). ریشه‌ی درخت مسابقه‌ی وابسته به حرکت در هر لحظه در طول اجرای الگوریتم، نزدیک‌ترین جفت را شامل می‌شود. زمانی که $lcand(p)$ در ساختار $Cands(p)$ تغییر می‌کند، اشاره‌گری از آن به فاصله‌ی متناظرش در درخت مسابقه‌ی وابسته به حرکت وجود دارد که آن فاصله در درخت مسابقه را تغییر می‌دهد و به دنبال آن یک به‌روزرسانی در درخت مسابقه صورت می‌گیرد که ممکن است منجر به تغییر نزدیک‌ترین جفت شود. زمان‌های مربوط به تغییرات در این درخت مسابقه‌ی وابسته به حرکت، همان است که در بخش ۲.۲ گفته شده است. در ادامه، در قضیه‌ی زیر اثبات خواهد شد که ساختار وابسته به حرکت معرفی شده برای نگهداری نزدیک‌ترین جفت از بین مجموعه نقاط متحرک در صفحه، با گذشت زمان، تمام معیارهای ارزیابی کیفیت معرفی شده در ابتدای این فصل را دارد.

قضیه ۶.۳.۲ ساختمان داده‌ی وابسته به حرکت برای مسئله‌ی یافتن نزدیک‌ترین جفت، کارا، پاسخ‌گو، فشرده و محلی است.

اثبات. برای بررسی کارا بودن ساختار لازم است یک کران بالا برای نسبت تعداد کل رویدادها به تعداد رویدادهای خارجی محاسبه شود. بدین منظور یک کران پایین برای تعداد رویدادهای خارجی و یک کران بالا برای تعداد کل رویدادها محاسبه می‌شود.

زمانی که مسیر حرکت n نقطه‌ی مجموعه‌ی S به صورت توابع شبه‌جبری از زمان است، یک کران پایین برای تعداد رویدادهای خارجی تقریباً مربعی می‌شود. n نقطه‌ی در حال حرکت روی یک خط را در نظر بگیرید؛ اگر مکان هر نقطه روی خط، در هر لحظه از زمان، در یک نمودار مکان-زمان در صفحه رسم شود، مسیر حرکت آن نقطه به صورت پارامتری از زمان به دست خواهد آمد. بدیهی است که هر بار دو نقطه روی این خط به هم می‌رسند (یک برخورد)، نزدیک‌ترین جفت کنونی، آن‌ها خواهند بود، از طرفی، این برخورد روی این خط، در واقع معادل با یک تقاطع بین مسیر حرکت رسم شده‌ی این دو نقطه در نمودار مکان-زمان خواهد بود. بنابراین می‌توان گفت که تعداد نزدیک‌ترین جفت‌هایی که تغییر می‌کنند، حداقل برابر تعداد برخوردهای بین مسیرهای حرکت نقاط است. از حداقل استفاده می‌شود؛ زیرا حالت‌هایی هستند که دو نقطه به هم نزدیک می‌شوند و نزدیک‌ترین جفت کنونی می‌شوند ولی باهم برخوردی ندارند.

برای محاسبه‌ی یک کران بالا برای تعداد کل رویدادها لازم است، مجموع تعداد رویدادهای رخ داده مربوط

به سه لیست تاییدیه‌ی ترتیب (سه نوع رویداد معرفی شده) و تعداد رویدادهای مربوط به درخت مسابقه‌ی وابسته به حرکت محاسبه شود.

همان‌طور که دیده شد، سه لیست تاییدیه‌ی ترتیب برای تعیین رویدادهای محور x ، $+60^\circ$ و -60° نیاز است. بنابراین باید سه لیست مرتب شده برای نقاط در حال حرکت نگهداری شوند. همان‌طور که در زیربخش ۱.۱.۲، راه‌حل دوم بیان شد، تعداد رویدادها برای هر لیست مرتب شده‌ی نقاط برابر با تعداد دفعات تغییر ترتیب نقاط در این لیست‌ها، یعنی تعداد برخوردها بین مسیرهای حرکت نقاط است و از آن‌جایی که هر جفت از نقاط، تحت حرکت‌های شبه‌جبری، تنها یک تعداد ثابت بار باهم برخورد دارند (بخش ۱.۱ را ببینید)؛ پس برای هر دو نقطه تعداد ثابت رویداد و تعداد کل رویدادها (برای تمام نقاط) در هر کدام از سه لیست ترتیب $O(n^2)$ خواهد بود.

در بدترین حالت اگر پردازش بخش عمده‌ای از رویدادهای ترتیب، منجر به تغییر فیلد $lcand$ شود، $O(n^2)$ بار تغییر برای فیلدهای $lcand()$ در طول اجرای الگوریتم رخ می‌دهد؛ از آن‌جایی که فاصله‌های متناظر با جفت‌های $(p, lcand(p))$ عناصر درخت مسابقه را می‌سازند پس تعداد به‌روزرسانی این عناصر که متناظر با تغییر فیلدهای $lcand$ است برابر $O(n^2)$ خواهد شد. (یعنی برای استفاده از لم ۱.۲.۲ $m \in O(n^2)$). برای محاسبه‌ی تعداد کل رویدادهای درخت مسابقه‌ی وابسته به حرکت، با استفاده از لم ۱.۲.۲ باید تعداد رأس‌های روی پوشش پایینی توابع تغییرات عناصر درخت مسابقه، یعنی تغییرات فاصله‌های متناظر با جفت‌های $(p, lcand(p))$ محاسبه شود. چون مسیر حرکت نقاط شبه‌جبری از زمان با درجه‌ی حداکثر s فرض شده‌اند، فاصله‌ی بین هر جفت از مسیرها نیز به‌صورت توابع شبه‌جبری از زمان با درجه‌ی حداکثر $2s$ می‌شود (این بدیهی است؛ زیرا مربع فاصله، دارای حداکثر درجه‌ی $2s$ خواهد بود). بنابراین با استفاده از لم ۱.۲.۲، تعداد رویدادهای درخت مسابقه برابر با $O(\lambda_{2s+2}(n^2) \log n)$ می‌شود که به‌راحتی با استفاده از قضیه‌ی ۲.۲.۱ می‌توان گفت که از مرتبه‌ی $O(n^2 \beta(n) \log n)$ است که $\beta(n)$ یک تابع است که رشد آن به‌شدت آهسته و از رشد هر تابعی از جمله توابع لگاریتمی کم‌تر است؛ بنابراین یک کران بالا برای نسبت مربوط به بررسی کارایی حداکثر

$$\leq \frac{O(n^2 \beta(n) \log n) + O(n^2)}{O(n^2)} = O(\beta(n) \log n) \in O(\log^2 n)$$

است. پس ساختمان داده‌ی وابسته به حرکت معرفی شده کارا است.

ساختارهای معرفی شده $Cands()$ و $Parents()$ که برای تعیین و به‌روزرسانی درست فیلد $lcand$ هر نقطه هستند، با درخت‌های دودویی متوازن پیاده‌سازی شدند. از طرفی عملیاتی که برای به‌روزرسانی این

ساختارها هنگام وقوع یک نوع رویداد لازم است، همان‌طور که در شبه‌کده‌های مربوط به هر نوع رویداد دیده شد، شامل یک تعداد ثابت عملیات جداسازی، الحاق، حذف، درج و جستجو است، که هر کدام از این عملیات در زمان $O(\log n)$ انجام می‌شوند. همان‌طور که در بخش ۲.۲ دیده شد، درخت مسابقه‌ی وابسته به حرکت پاسخ‌گو است و دارای زمان پاسخ‌گویی $O(\log^2 n)$ است زیرا در بدترین حالت، پردازش یک رویداد منجر به انجام $O(\log n)$ (ارتفاع درخت) تعداد عملیات روی ساختار درخت و برنامه‌ریزی $O(\log n)$ تعداد رویداد در صف رویداد درخت می‌شود. بنابراین ساختمان داده‌ی وابسته به حرکت معرفی شده پاسخ‌گو است.

آن‌جا تنها $O(n)$ تعداد رویداد در صف رویداد در هر زمان وجود دارد: یک رویداد برای هر $3 - 3n$ تاییدیه‌ی ترتیب و یک رویداد برای هر $O(n)$ تاییدیه‌ی مقایسه که برای هر گره داخلی درخت مسابقه‌ی وابسته به حرکت تعریف می‌شود. بنابراین ساختمان داده‌ی وابسته به حرکت معرفی شده فشرده است.

هر نقطه‌ی p از مجموعه‌ی S ، به حداکثر شش تاییدیه‌ی ترتیب و $O(\log n)$ تاییدیه‌ی مقایسه که برای نگهداری فیلدهای سمت چپ‌ترین در گره‌های داخلی هر درخت $Cands()$ به‌منظور تعیین فیلد $lcand$ آن درخت هستند، تعلق دارد. در ادامه نشان داده می‌شود که هر نقطه حداکثر در $O(\log n)$ تاییدیه در درخت مسابقه‌ی وابسته به حرکت نیز می‌تواند شرکت داشته باشد. با توجه به این که مجموعه‌های $Cands()$ برای نقاط کاملاً مجزا از یکدیگرند (تمام مجموعه‌های $Cands()$ با هم یک افراز از مجموعه نقاط S هستند، یعنی اشتراک هر دو مجموعه‌ی $Cands()$ تهی است)، بنابراین هر نقطه در هر چرخش صفحه تنها در یک مجموعه‌ی $Cands()$ ظاهر می‌شود و اگر در بهترین حالت، سمت چپ‌ترین نقطه در مجموعه‌ی $Cands()$ خود نیز باشد (یعنی $lcand$)، آنگاه می‌توان گفت که هر نقطه، حداکثر متعلق به سه جفت $(p, lcand(p))$ می‌شود (یکی برای هر چرخش). به عبارتی دیگر، هر نقطه‌ی p ، برای $lcand(q)$ حداکثر سه q متفاوت، یکی برای هر چرخش است؛ و از طرفی هر جفت فعال $(p, lcand(p))$ در $O(\log n)$ تعداد رویداد در درخت مسابقه‌ی وابسته به حرکت شرکت دارد. در نتیجه با توجه به تمام مطالب بیان شده، هر نقطه در حداکثر $O(\log n)$ تاییدیه قرار دارد و ساختمان داده‌ی وابسته به حرکت معرفی شده محلی است. \square

فصل ۳

تغییرات توپولوژیکی در مثلث‌بندی دلونی بر

اساس متر L_∞

چیو^۱ در مقاله‌ی [۱۶] نشان می‌دهد که برای نگهداری مثلث‌بندی دلونی یک مجموعه از نقاط در صفحه، بر اساس مترهای L_1 و L_∞ ، هنگامی که نقاط آن با سرعت‌های ثابت حرکت می‌کنند، باید $O(n^2 \cdot \alpha(n))$ رویداد بررسی شود. از آنجایی که از تعمیم این مطلب، برای تحلیل پوشش وابسته به حرکت معرفی شده در فصل ۴ استفاده خواهد شد، چگونگی به‌دست آمدن این تعداد رویداد، در این فصل بیان می‌شود و سپس نتایج حاصل به متر d_\diamond که در زیربخش ۱.۹.۱ معرفی شده است، تعمیم داده می‌شود.

۱.۳ مفاهیم مقدماتی

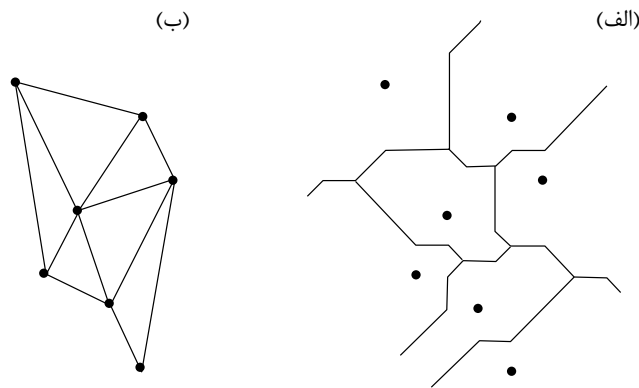
طبق قضیه‌ی ۳.۷.۱، هر مثلث از مثلث‌بندی دلونی مجموعه نقاط P در صفحه یک دایره‌ی محیطی منحصر به فرد دارد که هیچ نقطه‌ی دیگری از P را دربر نمی‌گیرد. یعنی دایره‌ی محیطی هر مثلث تنها سه نقطه را روی مرزش دارد. یک مثلث با دایره‌ی محیطی‌اش را در نظر بگیرید. زمانی که در اثر حرکت نقاط، نقطه‌ای دیگری مجاور به این مثلث، روی مرز این دایره قرار می‌گیرد، یعنی چهار نقطه از دو مثلث مجاور روی مرز یک دایره واقع می‌شوند گفته می‌شود که یک **تغییر توپولوژیکی**^۲ رخ داده است. در این بخش نشان داده می‌شود که تعداد تغییرات توپولوژیکی در مثلث‌بندی دلونی n نقطه در صفحه، بر اساس مترهای L_1 و L_∞ ، زمانی که نقاط با سرعت‌های ثابت حرکت می‌کنند، برابر با $O(n^2 \alpha(n))$ است، که $\alpha(n)$ همان‌طور که در بخش ۲.۱ بیان شد، تابعی است که بسیار آهسته رشد می‌کند و برای مقادیر عملی n ، یک ثابت کوچک است.

مثلث‌بندی دلونی یک مجموعه از نقاط در صفحه، به‌طوری که هر نقطه با سرعتی ثابت حرکت می‌کند را در نظر بگیرید. هرچند که این مثلث‌بندی نیز با حرکت نقاط، به‌طور پیوسته حرکت می‌کند (تغییر می‌کند)، اما تغییرات توپولوژیکی که نیاز به به‌روزرسانی مثلث‌بندی دلونی را ایجاد می‌کنند، تنها در زمان‌های گسسته رخ می‌دهد. سوال این است: چه تعداد تغییرات توپولوژیکی (رویداد) در مثلث‌بندی دلونی، زمانی که نقاط با سرعت‌های ثابت حرکت می‌کنند، رخ می‌دهد؟

برای یک مثلث‌بندی دلونی با نقاط متحرک، دو مثلث مجاور به هم را در نظر بگیرید. به چهار نقطه‌ی تشکیل دهنده‌ی این دو مثلث، یک چهارتایی یا سلول گفته می‌شود. با توجه به مطالب زیربخش ۱.۴.۱ و حرکت نقاط مثلث‌بندی به‌راحتی دیده می‌شود که لحظه‌ای درست قبل از غیرمجاز شدن یال مجاور به دو

^۱Chew

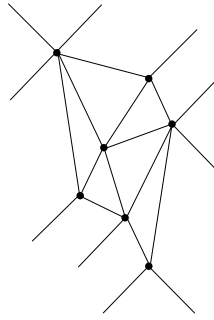
^۲Topological change



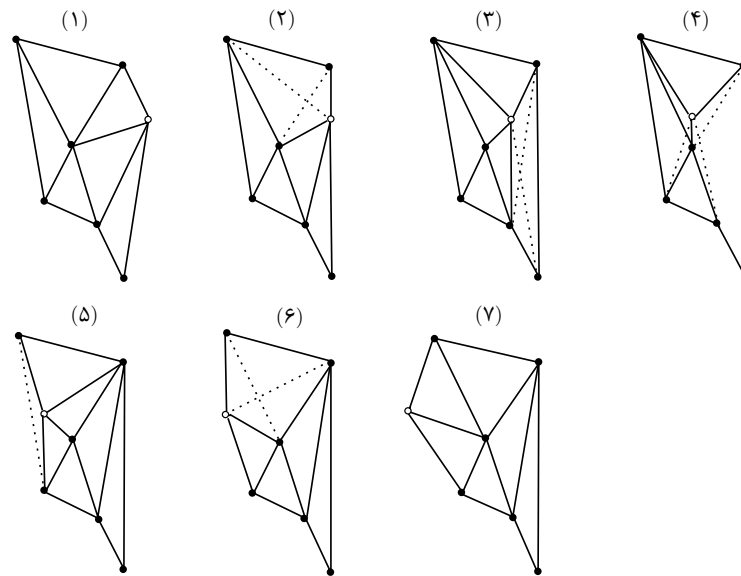
شکل ۱.۳: (الف): نمودار ورنوی و شکل (ب): مثلث‌بندی دلونی متناظر با آن بر اساس متر L_∞

مثلث (قطر یک سلول)، هر چهار نقطه‌ی آن سلول روی یک دایره قرار می‌گیرند (یک تغییر توپولوژیکی رخ می‌دهد). بنابراین برای تشخیص غیرمجاز شدن یک یال کافی است زمان‌هایی که چهار نقطه‌ی یک سلول روی یک دایره قرار می‌گیرند (تغییرات توپولوژیکی) تعیین شوند و به‌روزرسانی مثلث‌بندی در این زمان‌ها صورت گیرد. این به‌روزرسانی با چرخش یال غیرمجاز شده انجام می‌شود. به‌طور خلاصه می‌توان گفت، زمانی که نقاط مثلث‌بندی دلونی حرکت می‌کنند، تمام مجاورت‌ها (یال‌ها) حفظ می‌شوند به‌جز در زمان‌های گسسته‌ای که تغییرات توپولوژیکی رخ می‌دهند و کافی است در این مواقع با چرخش یال ساختار را همچنان حفظ کرد.

همان‌طور که در قضیه ۱.۷.۱ بیان شد، برای هر یال مثلث‌بندی دلونی یک بزرگ‌ترین دایره‌ی خالی وجود دارد، که نقاط انتهایی یال روی مرز آن قرار گیرد. همان‌طور که در بخش ۸.۱ بیان شد، دایره‌ی L_∞ -متریک، یک مربع است و برای L_1 -متریک، دایره‌ی مذکور یک مربع 45° درجه چرخیده شده است. برای این بخش، L_∞ و L_1 متریک هم ارز هستند، چون ماهیت شکل مهم است. بنابراین در ادامه از L_∞ -متریک استفاده می‌شود، چون رسم مربع‌های چرخیده نشده آسان‌تر است. شکل ۱.۳ را برای یک مثال نمودار ورنوی L_∞ و مثلث‌بندی دلونی L_∞ متناظرش ببینید. یک تفاوت بین مثلث‌بندی دلونی L_∞ و مثلث‌بندی دلونی استاندارد L_2 در شکل دیده می‌شود: مثلث‌بندی دلونی L_∞ لزوماً غشای محدب مجموعه نقاط اولیه‌اش را مثلث‌بندی نمی‌کند. برای ساده سازی نمایش، فرض می‌شود که مثلث‌بندی دلونی در حصار یک مربع با نقاط گوشه در (∞, ∞) ، $(-\infty, \infty)$ ، $(-\infty, -\infty)$ و $(\infty, -\infty)$ ساخته می‌شود. شکل ۲.۳ را ببینید. با این فرض: کل صفحه مثلث‌بندی می‌شود؛ بنابراین، می‌توان از سروکار داشتن با موارد خاص‌ای که برای نقاط روی مرز بیرونی مثلث‌بندی دلونی، اتفاق می‌افتد، اجتناب کرد.



شکل ۲.۳: همان مثلث‌بندی دلونی بر اساس متر L_∞ با نقطه‌هایی اضافه شده در بی‌نهایت



شکل ۳.۳: تغییرات در مثلث‌بندی دلونی بر اساس متر L_∞ ، چنان‌چه تنها یک نقطه‌ی آن حرکت کند.

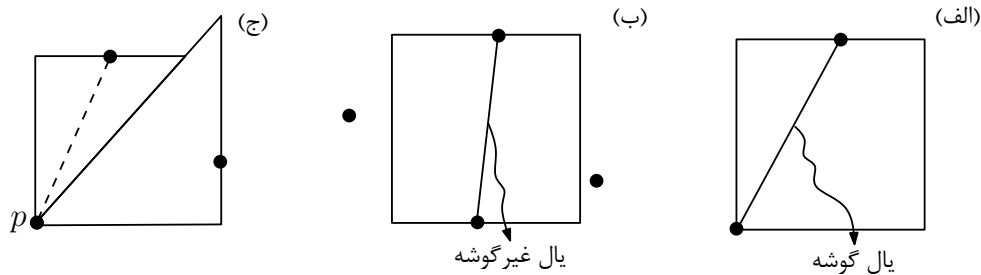
۲.۳ شمارش تعداد تغییرات توپولوژیکی

در این بخش، تعداد تغییرات توپولوژیکی در مثلث‌بندی دلونی زمانی که نقاط اولیه با سرعت ثابت حرکت می‌کنند، تعیین خواهد شد. شکل ۳.۳ تغییرات‌ای که برای یک مثال ساده رخ می‌دهد زمانی که تنها یکی از نقاط اولیه حرکت می‌کنند را نشان می‌دهد. توجه کنید که تغییرات (چرخش یال‌ها) تنها زمانی که چهار نقطه روی یک دایره قرار می‌گیرند رخ می‌دهد (که در این جا دایره L_∞ یک مربع است). به عبارت دیگر، یک تغییر تنها زمانی رخ می‌دهد که یک مربع خالی که چهار نقطه روی مرزش قرار دارد وجود داشته باشد.

دو نوع یال دلونی وجود دارد: یال گوشه^۳ و یال غیر گوشه^۴. قسمت‌های (الف) و (ب) شکل ۴.۳ را ببینید.

^۳ Corner edge

^۴ Non-corner edge



شکل ۴.۳: شکل‌های (الف) و (ب): یک یال گوشه و یک یال غیر گوشه، شکل (ج): نزدیک‌ترین برخورد متناظر با یک یال گوشه برای نقطه‌ی p

اگر یک مربع خالی که از نقاط انتهایی یال می‌گذرد و یک گوشه آن روی یکی از نقاط انتهایی یال است وجود داشته باشد آنگاه آن یال دلونی یک **یال گوشه** نامیده می‌شود. یال‌های دلونی باقیمانده یال غیر گوشه نامیده می‌شوند. توجه کنید که یک مربع خالی که روی یک یال واقع می‌شود می‌تواند بلغزد و تغییر اندازه دهد در حالی که همچنان در تماس با نقاط انتهایی یال باقی بماند. بنابراین، بیش‌تر یال‌ها در یک مثلث‌بندی دلونی L_∞ یال‌های گوشه هستند. آن چنان‌که تمام یال‌ها در مثلث‌بندی دلونی شکل ۲.۳ یال‌های گوشه هستند. به راحتی دیده می‌شود که یک یال غیر گوشه با ضلع‌های مقابل به هم در مربع خالی تماس دارد؛ زیرا اگر یک یال دلونی e با دو ضلع مجاور به هم یا یک ضلع مربع خالی تماس داشته باشد، می‌توان مربع خالی را کوچک کرد تا نشان دهد که e یک یال گوشه است. حتی زمانی که یال e با ضلع‌های مقابل به هم یک مربع خالی تماس دارد، اغلب می‌توان با لغزاندن مربع نشان داد که e یک یال گوشه است. یک یال دلونی یک **یال غیر گوشه** است اگر لغزیدن مربع خالی قبل از این که نقطه‌ی انتهایی یال در گوشه مربع قرار گیرد، توسط نقاط دیگر مثلث‌بندی متوقف شود. قسمت (ب) شکل ۴.۳ را ببینید. هر یال غیر گوشه یک جفت از این نقاط بازدارنده را دارد.

هر تغییر در مثلث‌بندی دلونی، معادل تغییر در یک یال گوشه، یا تغییر در یک یال غیر گوشه است. بنابراین اگر برای تغییرات یال‌ها یک کران تعیین شود، در حقیقت، یک کران برای تعداد کل تغییرات مثلث‌بندی دلونی نیز تعیین شده است. در ادامه، یک کران برای تعداد تغییرات یال‌های گوشه تعیین می‌شود.

۱.۲.۳ شمارش تغییرات یال‌های گوشه

قضیه ۱.۲.۳ برای n نقطه در صفحه، دارای حرکت با سرعت ثابت، $O(n^2 \alpha(n))$ تغییر در مجموعه یال‌های گوشه وجود دارد.

اثبات. نشان داده می‌شود که برای هر نقطه‌ی اولیه‌ی p ، $O(n \alpha(n))$ تغییر در یال‌های گوشه‌اش وجود دارد.

سپس مجموع این تغییرات برای تمام نقاط، کران ذکر شده در قضیه را به دست می آورد. توجه کنید که چهار نوع یال گوشه متناظر با چهار گوشه مربع وجود دارد. بدون از دست دادن کلیت مسئله، یال گوشه‌ی متناظر با گوشه‌ی پایین سمت چپ مربع در نظر گرفته می شود.

در یک زمان ثابت t ، می توان یال گوشه را برای نقطه‌ی اولیه p یافت. ابتدا گوشه‌ی پایین سمت چپ یک مربع خیلی کوچک، روی نقطه‌ی p قرار داده می شود و سپس آن مربع خالی، تا زمانی که با تعدادی نقطه برخورد کند، بزرگ می شود. اولین نقطه‌ای که مربع با آن برخورد می کند (فرض می شود که با نقطه‌ای برخورد می کند) یال گوشه‌ی کنونی برای p را مشخص می کند. در ادامه، این دیدگاه برای یافتن یال گوشه، با استفاده از مثلث‌ها به جای مربع، از نو طرح می شود و مربع به دو مثلث شکافته می شود (دلیل این کار در قسمت‌های بعدی بیان می شود). قسمت (ج) شکل ۴.۳ را ببینید. هر دو مثلث روی نقطه‌ی اولیه p در نظر گرفته می شود، سپس هر کدام از آن‌ها، به طور مستقل بزرگ می شوند تا زمانی که هر کدام با یک نقطه برخورد کنند. نزدیک‌ترین برخورد (از بین حداکثر دو برخورد) یال گوشه برای p را تعیین می کند. در ادامه، تنها یکی از دو مثلث در نظر گرفته می شود، مثلث بالایی. برای هر نقطه‌ی q ($q \neq p$)، یک نسخه از مثلث ایجاد و گوشه‌ی پایین آن، روی p قرار داده می شود و تا زمانی که q را لمس کند بزرگ می شود. یک مقدار به هر نقطه‌ی q اختصاص داده می شود؛ مقدار q اندازه‌ی مثلث حاصل است (برای نمونه، طول وتر مثلث حاصل می تواند به عنوان این مقدار سنجیده شود). توجه کنید که مثلث گسترش یافته با تعداد زیادی از نقاط برخورد نمی کند، بنابراین، مقدار مرتبط با آن‌ها بی نهایت فرض می شود.

با گذشت زمان به دلیل حرکت پیوسته نقاط، مقدار مرتبط با هر q تغییر می کند که این تابعی از زمان را برای هر نقطه‌ی q تعریف می کند. چون هر نقطه دارای حرکت خطی (سرعت ثابت) است پس مقدار مرتبط با آن (برای نمونه طول وتر مثلث) نیز به صورت تابعی خطی از زمان تغییر می کند و از آنجایی که یک نقطه‌ی q می تواند زمانی به مثلث مذکور وارد شده و زمانی دیگر از آن خارج شود (در بازه‌ای از زمان در مثلث حضور داشته باشد)، در نتیجه برای هر نقطه‌ی منحصر به فرد q ، تابع تغییرات مقدار مرتبط با آن، شبیه به یک پاره خط خواهد شد. دلیل تبدیل مربع به دو مثلث و گسترش مثلث‌ها برای یافتن یال گوشه، نیز این بود که تابع تغییرات مقدار مرتبط با هر q در یک مثلث شبیه به یک پاره خط می شود (اگر از مربع استفاده می شد، تابع تغییرات مقدار مرتبط با هر q شبیه به یک چندجمله‌ای می شد). بنابراین، اگر این تابع تغییرات برای هر نقطه‌ی q با گذشت زمان رسم شود، نتیجه به صورت یک مجموعه از حداکثر n پاره خط خواهد شد. یک مجموعه‌ی مشابه از پاره خط‌ها برای مثلث پایینی نیز به دست می آید. از طرفی، همان طور که قبلاً بیان شد، یال گوشه برای p متناظر

با اولین برخوردی است که با بزرگ کردن همزمان دو مثلث (معادل یک مربع) به دست می آید. پس یال گوشه برای p متناظر با پوشش پایینی مجموعه‌ی ترکیبی از پاره‌خط‌های هر دو مثلث است و یک تغییر در یال گوشه، متناظر با یک نقطه‌ی شکست در پوشش پایینی است.

با توجه به مطالب ذکر شده در بخش ۲.۱، اگر هر کدام از توابعی که پیچیدگی پوشش پایینی آن‌ها محاسبه می‌شود تابعی چندجمله‌ای از درجه‌ی s ، تعریف شده روی یک بازه باشند، آنگاه پیچیدگی پوشش پایینی برابر $O(\lambda_{s+2}(n))$ می‌شود. بنابراین در این جا $s = 1$ و طبق قضیه‌ی ۲.۲.۱،

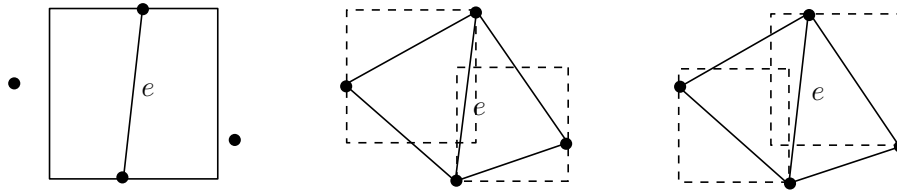
$$O(\lambda_{s+2}(n)) = O(\lambda_3(n)) = O(n\alpha(n)).$$

بنابراین پوشش پایینی برای یک مجموعه از $O(n)$ پاره‌خط، دارای پیچیدگی $O(n\alpha(n))$ است و حداکثر این تعداد تغییرات برای یک یال گوشه‌ی نقطه‌ی p وجود دارد. از آن جایی که هر نقطه‌ی p می‌تواند در چهار گوشه‌ی متمایز یک مربع قرار گیرد (چهار یال گوشه‌ی متمایز داشته باشد)، پس این تعداد تغییرات، چهار برابر می‌شود. پس برای تمام یال‌های گوشه‌ی هر نقطه‌ی p نیز تعداد $O(n\alpha(n))$ تغییر باید بررسی شود. در نتیجه برای کل نقاط (n نقطه)، $O(n^2\alpha(n))$ تغییر باید بررسی شود. \square

۲.۲.۳ شمارش تغییرات یال‌های غیر گوشه

یادآوری می‌شود که هر تغییر در مثلث‌بندی دلونی، متناظر با یک تغییر در یال گوشه، یا یک تغییر در یال غیر گوشه است. حالا که یک کران برای تغییرات یال‌های گوشه به دست آمد، یافتن کران برای تغییرات یال‌های غیر گوشه، مرتبط با آن و آسان است.

ادعا می‌شود که هر یال غیر گوشه‌ی مثلث‌بندی دلونی، در یک سلول متشکل از چهار یال گوشه قرار دارد. این مطلب، از این مشاهده به دست می‌آید که برای یک یال غیر گوشه، مربع خالی‌ای که روی نقاط انتهایی یال واقع می‌شود، به دلیل برخورد با دو نقطه‌ی بازدارنده، نمی‌تواند به خیلی دور بلغزد. اگر مربع در خلاف جهت یکی از این نقاط بازدارنده لغزانده شود، به راحتی دیده می‌شود که مربع می‌تواند کوچک شود به طوری که وجود دو یال گوشه را برای هر نقطه‌ی بازدارنده تایید کند. شکل ۵.۳ را ببینید. در مدت طول عمر یک سلول، تنها $O(1)$ بار، دو قطر آن سلول با یکدیگر جابه‌جا می‌شوند (چرخش یال). این، به این دلیل است که: چون نقاط با سرعت‌های ثابت حرکت می‌کنند (از درجه‌ی یک هستند)، پس تابع تغییرات فاصله‌ی هر دو نقطه نیز حداکثر از درجه‌ی دو خواهد بود. از طرفی یک مجموعه از چهار نقطه‌ی تشکیل دهنده‌ی یک سلول، زمانی روی یک



شکل ۵.۳: قرار گرفتن هر یال غیر گوشه‌ی e ، در یک سلول متشکل از چهار یال گوشه

دایره (مربع) قرار می‌گیرند که فاصله‌ی بین دو نقطه‌ی غیرمجاور برابر با فاصله‌ی بین دو نقطه‌ی انتهایی یال مجاور به دو مثلث در این سلول شود. این زمان، معادل یک نقطه‌ی برخورد بین توابع تغییرات فاصله‌ی این نقاط بر حسب زمان است. از طرفی تعداد نقاط برخورد بین دو تابع فاصله از درجه‌ی دو، حداکثر دو نقطه خواهد بود. بنابراین اگر تمام حالت‌های ممکن نیز برای یک سلول بررسی شود، نقاط آن حداکثر یک تعداد ثابت بار، روی یک دایره قرار خواهند گرفت. از طرفی $O(n)$ سلول در شروع حرکت یعنی در زمان صفر وجود دارد. طبق قضیه‌ی ۱.۲.۳، $O(n^2 \alpha(n))$ سلول جدید نیز با گذشت زمان، به دلیل تغییر یال‌های گوشه به وجود می‌آیند. بنابراین، در مجموع $O(n^2 \alpha(n))$ سلول متفاوت وجود خواهد داشت و همان‌طور که بیان شد برای یال غیر گوشه‌ی هر سلول نیز یک تعداد ثابت بار تغییر رخ خواهد داد. در نتیجه قضیه‌ی زیر به دست می‌آید،

قضیه ۲.۲.۳ یک مجموعه از n نقطه که با سرعت‌های ثابت در صفحه حرکت می‌کنند را در نظر بگیرید. تعداد $O(n^2 \alpha(n))$ تغییر در مثلث‌بندی دلونی این مجموعه نقاط بر اساس متر L_∞ (یا L_1) رخ خواهد داد.

۳.۳ تعمیم نتایج به متر d_\diamond

روشی که مطرح شد، می‌تواند از چندین راه مختلف تعمیم داده شود از جمله:

۱. روش برای فاصله‌های دیگر، بجز L_∞ و L_1 نیز به کار می‌رود. این روش، برای هر تابع فاصله‌ی محدب که شکل محدب آن یک چندضلعی باشد نیز به کار می‌رود. همچنین با استفاده از روش بالا می‌توان نشان داد که برای یک مثلث‌بندی دلونی بر اساس یک تابع فاصله‌ی محدب که شکل محدب آن یک چندضلعی k -وجهی است و مجموعه‌ی n نقطه‌ی سازنده‌ی آن، هر کدام با سرعتی ثابت حرکت می‌کنند، $O(k^2 n^2 \alpha(n))$ تغییر وجود دارد. برای جزئیات بیشتر، مقالات [۱۷، ۱۸]، توسط چپو^۵ و کدم^۶ را

^۵Chew

^۶Kedem

ببینید.

۲. به طور مشابه، کران به دست آمده برای حرکت‌های پیچیده‌تر نیز معتبر باقی می‌ماند. کافی است، حرکت نقاط اولیه به گونه‌ای محدود شده باشد که هیچ چهار نقطه‌ای بیش از یک تعداد ثابت بار، روی یک دایره قرار نگیرند. توجه کنید که منظور از دایره در این جا، دایره‌ی استاندارد نیست، بلکه دایره‌ی متری است که مسئله استفاده می‌کند یا شکل محدبی که تابع فاصله‌ی محدب را تعریف می‌کند.

تابع فاصله‌ی محدب d_\diamond ، که بر اساس شکل لوزی \diamond ، در زیر بخش ۱.۹.۱ معرفی شد و یک متر نیز بود را در نظر بگیرید. با توجه به نکته‌ی ۱، برای مثلث‌بندی دلونی بر اساس متر d_\diamond نیز، زمانی که نقاط آن با سرعت‌های ثابت حرکت کنند، تعداد تغییرات توپولوژیکی برابر با $O(n^2\alpha(n)) = O(n\lambda_{1+2}(n))$ می‌شود و با توجه به نکته‌ی ۲، اگر نقاط دارای حرکت‌های شبه‌جبری از درجه‌ی s باشند، نتایج به طور مشابه برقرار و تعداد تغییرات توپولوژیکی، طبق قضیه‌ی ۲.۲.۱، برابر با $O(n\lambda_{s+2}(n))$ است.

نتیجه ۱.۳.۳ تعداد تغییرات توپولوژیکی برای مثلث‌بندی دلونی بر اساس متر d_\diamond ، زمانی که نقاط دارای حرکت‌های شبه‌جبری از درجه‌ی s هستند، برابر با $O(n\lambda_{s+2}(n))$ است.

فصل ۴

یک پوشش هندسی وابسته به حرکت در صفحه

در این فصل یک $\varepsilon + 1$ -پوشش ساده با اندازه‌ی خطی، برای یک مجموعه از n نقطه در صفحه معرفی می‌شود. این پوشش زمانی که نقاط آن حرکت می‌کنند، می‌تواند به‌صورتی کارا نگهداری شود. فرض می‌شود که مسیر حرکت نقاط با توابع شبه‌جبری از درجه‌ی حداکثر s توصیف می‌شوند (بخش ۱.۱ را ببینید). لازم به ذکر است که مطالب این فصل بر اساس مقاله‌ی آبام^۱ و همکارانش [۲] تنظیم شده است.

۱.۴ کارهای مرتبط

یک $\varepsilon + 1$ -پوشش را در نظر بگیرید. برای جزئیات بیش‌تر در مورد پوشش‌ها می‌توانید بخش ۱۰.۱ را ببینید. هنگامی که نقاط پوشش دارای حرکت پیوسته‌ای از زمان باشند، پوشش به‌طور پیوسته از زمان تغییر می‌کند، اما تنها در لحظه‌های مشخصی نیاز به به‌روزرسانی خواهد داشت، تا همچنان در تمام زمان‌ها یک $\varepsilon + 1$ -پوشش باقی بماند. در فصل ۲ بیان شد که این لحظات رویداد نامیده می‌شوند. از آنجایی که کارایی و پاسخ‌گویی، از مهم‌ترین معیارها برای ارزیابی کیفیت ساختارهای وابسته به حرکت هستند، هدف، طراحی پوششی است که تعداد رویدادها و نیز زمان پاسخ‌گویی آن، کم باشد. همان‌طور که در فصل ۲ بیان شد، در حالت ایده‌ال، تعداد رویدادها باید نزدیک به کم‌ترین تعداد رویدادهایی که برای نگهداری ساختار لازم است باشد. برای پوشش‌ها، این بدین معنی است که تعداد رویدادها باید $O(n^2)$ باشد؛ زیرا مجموعه‌های n تایی از نقاط که به‌صورت خطی در صفحه حرکت می‌کنند وجود دارند، که برای هر $\varepsilon + 1$ -پوشش باید $\Omega(n^2/(1+\varepsilon)^2)$ رویداد پردازش شود [۲۷]. زمان پاسخ‌گویی نیز در حالت ایده‌ال باید چندلگاریتمی باشد.

گائو^۲ و همکارانش در [۲۷] نشان دادند که چگونه می‌شود یک $\varepsilon + 1$ -پوشش از نقاط متحرک را در فضای \mathbb{R}^d نگهداری کرد. پوشش آن‌ها دارای اندازه‌ی $O(n/\varepsilon^d)$ است. اما تعداد رویدادها و زمان پاسخ‌گویی آن، به میزان پراکندگی^۳ مجموعه نقاط پوشش از هم وابسته است که چندان خوشایند نیست. لازم به ذکر است که پراکندگی، که با ρ نشان داده می‌شود، برابر با نسبت بیش‌ترین فاصله‌ی بین دو نقطه در پوشش، به کم‌ترین فاصله‌ی بین دو نقطه در همان پوشش است. پوشش آن‌ها با فرض این که مسیر حرکت نقاط به‌صورت توابع جبری از درجه‌ی کم هستند، دارای $O(n^2 \log \rho)$ رویداد است و زمان مورد نیاز برای به‌روزرسانی آن در رویدادهای مذکور نیز $O(\log \rho / \varepsilon^d)$ است.

^۱Abam

^۲Gao

^۳Spread

چیو^۴ در مقاله‌ی [۱۵] نشان می‌دهد که مثلث‌بندی دلونی یک مجموعه از نقاط در صفحه، بر اساس متر L_1 ، یک پوشش با ضریب کشش $\sqrt{10}$ است. او همچنین در مقاله‌ی [۱۶] نشان می‌دهد که برای نگهداری این مثلث‌بندی، هنگامی که نقاط آن با سرعت‌های ثابت حرکت می‌کنند (مسیرهای حرکت خطی)، باید $O(n^2 \cdot \alpha(n))$ رویداد (تغییر توپولوژیکی) بررسی شود. فصل ۳ را ببینید. مشخص است که اگر ضریب کشش پوشش چیو از $\sqrt{10}$ به $1 + \varepsilon$ بهبود یابد (با حفظ همین کران برای تعداد رویدادها)، با توجه به کران پایین برای تعداد رویدادها در یک $1 + \varepsilon$ -پوشش، این تعداد رویداد برای نگهداری کارای پوشش، تحت حرکت نقاط، تعداد مناسب و قابل قبولی خواهد بود.

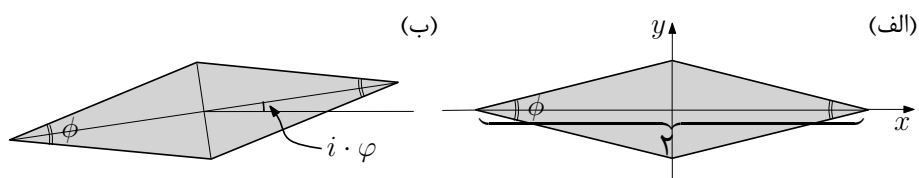
اگر از مثلث‌بندی دلونی در متر اقلیدسی استفاده شود، یک ۲.۴۲-پوشش به دست می‌آید [۳۲]. اخیراً این نتیجه، برای موارد خاصی که نقاط در موقعیت محدب قرار دارند به ۲.۳۳-پوشش بهبود یافته است [۲۳]. هرچند که ضریب کشش پوشش در این متر کم‌تر از متر L_1 است، اما بهترین مرز شناخته شده روی تعداد رویدادها برای مثلث‌بندی دلونی در متر اقلیدسی برابر با $O(n^2 \cdot \lambda_s(n))$ است [۴]، با فرض این که مسیر حرکت نقاط به صورت توابع چندجمله‌ای با حداکثر درجه‌ی s است. این تعداد رویداد چندان مناسب نیست؛ زیرا ساختار کارا نمی‌باشد.

از تمام این مطالب می‌توان این نکته را دریافت که برای ساخت یک $1 + \varepsilon$ -پوشش وابسته به حرکت که کارا نیز باشد، باید به نحوی از تعداد مناسب رویدادها در متر L_1 بهره برد و از طرفی با یک روش مناسب، ضریب کشش پوشش را کاهش داد. بدین منظور، همان‌طور که در این فصل بیان خواهد شد، به جای یک مثلث‌بندی دلونی در متر L_1 ، از اجتماع تعداد مناسبی مثلث‌بندی دلونی در متر دیگری که مشابه به متر L_1 است، استفاده می‌شود. دیده خواهد شد که با این روش، یک $1 + \varepsilon$ -پوشش با اندازه‌ی خطی که کارا و پاسخ‌گو نیز هست، به دست می‌آید. در ادامه این پوشش معرفی خواهد شد.

۲.۴ یک پوشش ایستای مناسب

P را یک مجموعه از n نقطه در صفحه و ε را یک ثابت مثبت داده شده ($0 < \varepsilon < 1$) در نظر بگیرید. در ابتدا الگوریتم جدیدی برای ساخت یک پوشش از P معرفی می‌شود، و سپس ضریب کشش و اندازه‌ی پوشش بررسی می‌شود.

^۴Chew



شکل ۱.۴: شکل (الف): لوزی اولیه \diamond ، شکل (ب): لوزی \diamond که به اندازه $i \cdot \varphi$ چرخیده است

۱.۲.۴ ساخت پوشش

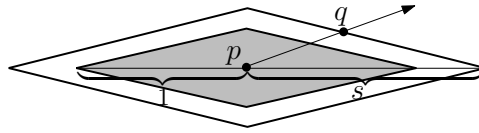
ساخت پوشش کاملاً ساده است. ابتدا یک مجموعه D از $O(1/\varepsilon^2)$ تعداد لوزی باریک تعریف می‌شود، که این لوزی‌ها، نسخه‌های چرخیده شده‌ای از یک لوزی اولیه ثابت \diamond هستند. البته، همان‌طور که بعداً مشخص خواهد شد، لوزی اولیه \diamond منحصر به فرد نخواهد بود. هر لوزی از D یک تابع فاصله‌ی محدب ایجاد می‌کند (زیربخش ۱.۹.۱ را ببینید)، بنابراین می‌توان یک مثلث‌بندی دلونی از مجموعه نقاط P را بر اساس تابع فاصله‌ی محدب متناظر با هر لوزی در D محاسبه کرد (زیربخش ۳.۹.۱ را ببینید). ثابت می‌شود که اجتماع این $O(1/\varepsilon^2)$ مثلث‌بندی دلونی، یک $1 + \varepsilon$ -پوشش برای مجموعه نقاط P می‌باشد. در ادامه، این ایده با دقت بیش‌تری بررسی خواهد شد.

ابتدا لوزی \diamond تعریف می‌شود. ϕ را یک زاویه‌ی ثابت در نظر بگیرید، که اندازه‌ی آن همان‌طور که بعداً در زیربخش ۲.۲.۴ توضیح داده خواهد شد، به‌صورت تابعی از ε تعیین می‌شود تا یک $1 + \varepsilon$ -پوشش را تولید کند. مرکز لوزی \diamond در مبدأ واقع شده است و قطرهای آن موازی با محورهای x و y هستند. سمت چپ‌ترین و سمت راست‌ترین رأس‌های لوزی دارای زاویه‌ی ϕ هستند، و طول قطر افقی برابر با ۲ است. این قطر، که قطر بلندتر لوزی است ^۵ قطر اصلی نامیده می‌شود. قسمت (الف) شکل ۱.۴ را ببینید.

در ادامه، مجموعه D ، به‌صورت مجموعه‌ای از لوزی‌های چرخیده شده، تعریف می‌شود. برای یک عدد صحیح $i > 0$ ، لوزی \diamond_i اشاره می‌کند به لوزی \diamond که حول مبدأ به اندازه‌ی زاویه‌ی $i \cdot \varphi$ ، در خلاف جهت عقربه‌های ساعت چرخیده شده است، که در آن $\varphi < \phi$ مقداری وابسته به ε است که در زیربخش ۲.۲.۴ تعیین خواهد شد. قسمت (ب) شکل ۱.۴ را ببینید. بنابراین مجموعه‌ی $D = \{\diamond_i : 0 \leq i < k\}$ است، به‌طوری‌که $k = 2\pi/\varphi$ می‌باشد.

هر لوزی $\diamond_i \in D$ ، یک تابع فاصله‌ی محدب $d_i : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ را تعریف می‌کند. برای توضیحات بیش‌تر در مورد تابع فاصله‌ی محدب بخش ۵.۱ را ببینید. دو نقطه $p, q \in \mathbb{R}^2$ را در نظر بگیرید. تابع فاصله‌ی

^۵Main diagonal



شکل ۲.۴: اگر لوزی \diamond به اندازه ضریب معیار s بزرگ شود، نقطه‌ی q روی مرز آن قرار می‌گیرد.

محدب لوزی \diamond_i برای این دو نقطه، به این شکل محاسبه می‌شود: ابتدا یک نسخه از لوزی \diamond_i به گونه‌ای انتقال داده می‌شود که مرکز آن روی نقطه‌ی p قرار گیرد. سپس مقدار تابع فاصله‌ی محدب $d_i(p, q)$ نسبت به p برابر است با ضریب معیاری که اگر لوزی به اندازه‌ی آن ضریب، انبساط (انقباض) یابد، نقطه‌ی q روی مرز آن قرار می‌گیرد. شکل ۲.۴ را ببینید. محاسبه‌ی این ضریب معیار ساده است و در بخش ۵.۱ بیان شده است. با توجه به مطالب همان بخش، از آنجایی که لوزی \diamond_i نسبت به مرکزش متقارن است، بنابراین تابع فاصله‌ی محدب d_i یک متر است. توجه کنید که در این جا لوزی \diamond_i ، نقش دایره‌ی این متر را دارد. در ادامه، هر نسخه از لوزی \diamond_i ، که به گونه‌ای انتقال داده می‌شود که مرکز آن روی نقطه‌ی مورد نظر مسئله قرار گیرد و سپس معیار می‌شود (منبسط یا منقبض می‌شود) یک i -لوزی نامیده می‌شود. از آنجایی که d_i یک تابع فاصله‌ی محدب است، می‌توان نمودار ورنوی و مثلث‌بندی دلونی را بر اساس آن ساخت. برای توضیحات بیشتر تر بخش ۹.۱ را ببینید. مثلث‌بندی دلونی مجموعه نقاط P بر اساس تابع فاصله‌ی محدب d_i ، با $DT_i(P)$ نشان داده می‌شود. بنابراین برای هر دو نقطه $p, q \in P$ ، یال (p, q) یک یال در $DT_i(P)$ است اگر و تنها اگر یک i -لوزی وجود داشته باشد، به طوری که نقاط p و q روی مرز آن قرار گیرند و این i -لوزی هیچ نقطه‌ی دیگری از P را دربر نگیرد (نه در داخلش و نه روی مرزش). از آنجایی که پوشش از اجتماع مثلث‌بندی‌های دلونی DT_i بر اساس متر d_i ، برای $0 \leq i < k$ حاصل می‌شود، بنابراین پوشش لوزی-دلونی^۶ مجموعه نقاط P نامیده می‌شود و با نماد $DDS(P)$ نشان داده می‌شود. با توجه به این مطالب (p, q) یک یال در $DDS(P)$ است اگر و تنها اگر i ای وجود داشته باشد که (p, q) یک یال در DT_i باشد.

۲.۲.۴ تحلیل ضریب کشش و اندازه‌ی پوشش

در این زیربخش ثابت خواهد شد که اگر زاویه‌های ϕ و φ به گونه‌ای انتخاب شوند که شرط‌های

$$(\cos \phi - \sin \phi) \geq 1/(1 + \varepsilon) \quad (1-4)$$

^۶Diamond-Delaunay spanner

$$(\sin(\phi/2)/\sin(\varphi/2)) \geq 1 + 2/\varepsilon \quad (2-4)$$

برقرار باشند، آنگاه $DDS(P)$ یک $1 + \varepsilon$ -پوشش برای مجموعه نقاط P خواهد بود. این شرطها می‌توانند با تنظیم زوایای ϕ و φ به صورت $\phi = \arcsin \frac{\varepsilon}{2(1+\varepsilon)}$ و $\varphi = 2 \arcsin \frac{\varepsilon^2}{2(1+\varepsilon)(2+\varepsilon)}$ به دست آیند. همواره فرض می‌شود که $\phi < \varphi$.

مشاهده ۱.۲.۴ اگر $\phi = \arcsin \frac{\varepsilon}{2(1+\varepsilon)}$ برقرار باشد آنگاه $0 < \phi < \frac{\pi}{6}$.

اثبات. از آنجایی که $\frac{\varepsilon}{2(1+\varepsilon)} < \frac{1}{2}$ و تابع \arcsin در بازه $(0, \pi/2)$ یک تابع صعودی است، بنابراین،
 $0 < \phi = \arcsin \frac{\varepsilon}{2(1+\varepsilon)} < \arcsin \frac{1}{2} = \frac{\pi}{6}$.

□

مشاهده ۲.۲.۴ با انتخاب $\phi = \arcsin \frac{\varepsilon}{2(1+\varepsilon)}$ و $\varphi = 2 \arcsin \frac{\varepsilon^2}{2(1+\varepsilon)(2+\varepsilon)}$ دو شرط ۱-۴ و ۲-۴ برقرار است.

اثبات. قسمت اول اثبات: با توجه به مشاهده‌ی ۱.۲.۴، رابطه‌ی $0 < \phi < \frac{\pi}{6}$ برقرار است، بنابراین رابطه‌ی $\cos \phi \geq 1 - \sin \phi$ نیز برقرار خواهد بود و شرط ۱-۴ به این صورت اثبات می‌شود:

$$\phi = \arcsin \frac{\varepsilon}{2(1+\varepsilon)}$$

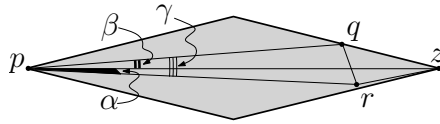
$$\sin \phi = \frac{\varepsilon}{2(1+\varepsilon)}$$

$$2 \sin \phi - 1 = \frac{\varepsilon}{1+\varepsilon} - 1$$

$$1 - 2 \sin \phi = \frac{1}{1+\varepsilon}$$

$$1 - \sin \phi - \sin \phi = \frac{\varepsilon}{1+\varepsilon}$$

$$\cos \phi - \sin \phi \geq \frac{1}{1+\varepsilon}.$$



شکل ۳.۴: شکل مربوط به مشاهده‌ی ۳.۲.۴

قسمت دوم اثبات: از مشاهده‌ی ۱.۲.۴، رابطه‌ی $1 < \cos(\phi/2) < \frac{1}{\epsilon}$ و از قسمت اول اثبات، رابطه‌ی $\sin \phi = \frac{\epsilon}{2(1+\epsilon)}$ برقرار است. با توجه به این روابط و نیز رابطه‌ی مثلثاتی $\sin \phi = 2 \sin(\phi/2) \cos(\phi/2)$ ، شرط ۲-۴ به این صورت اثبات می‌شود:

$$\varphi = 2 \arcsin \frac{\epsilon^2}{4(\epsilon+2)(1+\epsilon)}$$

$$\sin(\varphi/2) = \frac{\epsilon^2}{4(\epsilon+2)(1+\epsilon)}$$

$$\sin(\varphi/2) \cos(\phi/2) \leq \frac{\epsilon^2}{4(\epsilon+2)(1+\epsilon)}$$

$$\frac{1}{\sin(\varphi/2) \cos(\phi/2)} \geq \frac{4(\epsilon+2)(1+\epsilon)}{\epsilon^2}$$

$$\frac{\frac{\epsilon}{2(1+\epsilon)}}{\sin(\varphi/2) \cos(\phi/2)} \geq \frac{2(\epsilon+2)}{\epsilon}$$

$$\frac{\sin \phi}{2 \sin(\varphi/2) \cos(\phi/2)} \geq \frac{\epsilon+2}{\epsilon}$$

$$\frac{\sin(\phi/2)}{\sin(\varphi/2)} \geq 1 + 2/\epsilon.$$

□

لوزی \diamond را یک \circ -لوزی در نظر بگیرید، به طوری که نقطه‌ی p سمت چپ‌ترین و نقطه‌ی z سمت راست‌ترین رأس‌های آن باشند. فرض کنید نقطه‌ی q ، یک نقطه روی یال غیرمجاور به p در این لوزی باشد و نقطه‌ی r نیز نقطه‌ای دلخواه متعلق به \diamond و در سمت راست نقطه‌ی q باشد. شکل ۳.۴ را ببینید.

مشاهده ۳.۲.۴ دو رابطه‌ی زیر در \circ -لوزی با نقاط معرفی شده برقرار است:

$$(الف): \|pz\| \leq \|pr\| \text{ و } (ب): \|qr\| \leq \|qz\|.$$

اثبات. قسمت (الف): مثلث zpr را در نظر بگیرید. برای زاویه‌ی $\angle zpr$ که با α در شکل ۳.۴ نشان داده شده

است، رابطه‌ی $\frac{\pi}{6} < \phi < \alpha \leq \circ$ برقرار است. بنابراین $1 > \cos \alpha > \frac{\sqrt{3}}{4}$. طبق قانون کسینوس‌ها،

$$\|pr\| = \|pz\| \cdot \cos \alpha \leq \|pz\|.$$

قسمت (ب): در دو مثلث qpr و qpz به ترتیب دو زاویه‌ی $\angle qpr$ و $\angle qpz$ را در نظر بگیرید. این دو زاویه به

ترتیب با نمادهای β و γ در شکل ۳.۴ نشان داده شده‌اند. طبق قانون کسینوس‌ها،

$$\|qr\| = \|pq\| \cdot \cos \gamma, \quad \|qz\| = \|pq\| \cdot \cos \beta$$

و رابطه‌ی $\frac{\pi}{6} < \phi < \gamma < \beta \leq \circ$ نیز برقرار است. در نتیجه،

$$\cos \gamma \leq \cos \beta$$

$$\|pq\| \cdot \cos \gamma \leq \|pq\| \cdot \cos \beta$$

$$\|qr\| \leq \|qz\|.$$

□

حالا به راحتی با استفاده از فرض‌های ذکر شده و مشاهده‌های ۱.۲.۴ و ۳.۲.۴ می‌توان لم زیر را اثبات کرد. شکل

۴.۴ را ببینید.

لم ۴.۲.۴ ([۲]) فرض کنید که \diamond یک \circ -لوزی باشد. اگر p سمت چپ‌ترین رأس \diamond و q یک نقطه روی مرز

\diamond به طوری که زاویه‌ای که \overline{pq} با قطر اصلی \diamond می‌سازد، حداکثر $\varphi/2$ باشد. آنگاه برای هر نقطه‌ی $r \in \diamond$

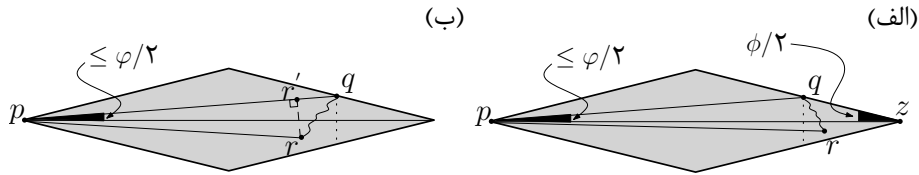
رابطه‌ی زیر برقرار است،

$$\|pr\| + (1 + \varepsilon) \cdot \|rq\| \leq (1 + \varepsilon) \cdot \|pq\|.$$

اثبات. حالت اول: r در سمت راست q واقع شود. فرض کنید z سمت راست‌ترین رأس \diamond باشد. در

مشاهده‌ی ۳.۲.۴ بررسی شد، که روابط $\|pr\| \leq \|pz\|$ و $\|qr\| \leq \|qz\|$ برقرار هستند، همچنین طبق

فرض لم، $\angle qpz \leq \varphi/2$. شکل ۴.۴ قسمت (الف) را ببینید. بنابراین،



شکل ۴.۴: شکل‌های (الف) و (ب): به ترتیب برای اثبات حالت‌های اول و دوم لم ۴.۲.۴

$$\begin{aligned}
 \|pr\| + (1 + \varepsilon) \cdot \|rq\| &\leq \|pz\| + (1 + \varepsilon) \cdot \|zq\| \\
 &\leq \|pq\| + \|zq\| + (1 + \varepsilon) \cdot \|zq\| \\
 &\leq \|pq\| \cdot \left(1 + \frac{\|zq\|}{\|pq\|} + (1 + \varepsilon) \cdot \frac{\|zq\|}{\|pq\|} \right) \\
 &\leq \|pq\| \cdot \left(1 + \frac{\sin(\varphi/2)}{\sin(\phi/2)} + (1 + \varepsilon) \cdot \frac{\sin(\varphi/2)}{\sin(\phi/2)} \right) \\
 &= \|pq\| \cdot \left(1 + (2 + \varepsilon) \cdot \frac{\sin(\varphi/2)}{\sin(\phi/2)} \right) \\
 &\leq \|pq\| \cdot \left(1 + \frac{2 + \varepsilon}{1 + 2/\varepsilon} \right) \\
 &= (1 + \varepsilon) \cdot \|pq\|.
 \end{aligned}$$

حالت دوم: r در سمت راست q واقع نشود. در این صورت یک مخروط با رأس p ، و زاویه‌ی باز ϕ ، که q و r را دربر می‌گیرد و r نزدیک‌تر به p است، تا به q وجود خواهد داشت. شکل ۴.۴ قسمت (ب) را ببینید. r' را تصویر عمود r روی \overline{pq} در نظر بگیرید. از آن جایی که $\|pq\|$ ، بزرگ‌ترین یال در مثلث pqr است،

$$\|rq\| \leq \|rr'\| + \|r'q\| = \|rr'\| + (\|pq\| - \|pr'\|) \leq (\|pq\| - (\cos \angle qpr - \sin \angle qpr) \|pr\|).$$

از آن جایی که، $\angle qpr \leq \phi$ و $\phi < \frac{\pi}{6}$ برقرار است، بنابراین،

$$\|pr\| + (1 + \varepsilon) \cdot \|rq\| \leq \|pr\| + (1 + \varepsilon)(\|pq\| - (\cos \phi - \sin \phi) \|pr\|) \leq (1 + \varepsilon) \cdot \|pq\|.$$

□

یک $(1 + \varepsilon)$ -مسیر از یک نقطه‌ی $p \in P$ تا یک نقطه‌ی $q \in P$ ، به‌عنوان یک مسیر از p تا q در $DDS(P)$ ، با

طول حداکثر $(1 + \varepsilon) \cdot \|pq\|$ تعریف می‌شود. لم ۴.۲.۴، بیان می‌کند که از الحاق یال pr و یک $(1 + \varepsilon)$ -مسیر از r به q ، یک $(1 + \varepsilon)$ -مسیر از p به q به دست می‌آید. از این مطلب، استفاده خواهد شد تا با استقرا نشان داده شود که بین هر جفت از نقاط، یک $(1 + \varepsilon)$ -مسیر وجود دارد.

مشاهده ۵.۲.۴ p و q را یک جفت نقطه در صفحه و $\alpha(p, q)$ را زاویه‌ای بگیرید که \vec{pq} با محور مثبت x می‌سازد. اگر i -لوزی‌ای $(i \in \{0, \dots, k-1\})$ را که سمت چپ‌ترین رأس آن، q یک نقطه روی یال‌اش، و مقدار $|\alpha(p, q) - i \cdot \varphi|$ برای آن حداقل شود، در نظر بگیرید، آنگاه خواص زیر را دارد:

$$\text{الف)} \quad |\alpha(p, q) - i \cdot \varphi| \leq \varphi/2$$

ب) q حتماً روی یک یال غیر مجاور به p در i -لوزی قرار دارد.

اثبات. قسمت الف): زمانی که گفته می‌شود، مقدار $|\alpha(p, q) - i \cdot \varphi|$ برای i -لوزی، حداقل می‌شود، منظور این است که زاویه‌ی بین \vec{pq} و قطر اصلی i -لوزی، حداقل می‌شود. از طرفی زاویه‌ای که قطر اصلی یک i -لوزی با محور مثبت x می‌سازد، هر مقدار پیوسته‌ای نمی‌تواند داشته باشد. بلکه، بسته به انتخاب زاویه‌ی φ ، در ابتدای ساخت پوشش، مقادیر آن گسسته، و به صورت ضرایبی از φ می‌باشد. بنابراین، هر زاویه‌ی دلخواهی، بین دو مقدار گسسته‌ی متوالی از ضرایب φ قرار می‌گیرد. در نتیجه، برای زاویه‌ی $\alpha(p, q)$ ، یک j متعلق به مجموعه‌ی $\{0, 1, \dots, k-1\}$ وجود دارد، به طوری که رابطه‌ی $j \cdot \varphi \leq \alpha(p, q) \leq (j+1) \cdot \varphi$ برقرار است. این بدان معنی است که پاره‌خط \vec{pq} ، بین قطر اصلی $(j+1)$ -لوزی و قطر اصلی j -لوزی قرار گرفته است. بنابراین، اگر \vec{pq} نیم‌ساز این دو قطر باشد، آنگاه، $(j+1)$ -لوزی یا j -لوزی، یکی به دلخواه، به عنوان i -لوزی انتخاب می‌شود و چون زاویه‌ی بین دو قطر متوالی، φ است، بنابراین در این حالت، $|\alpha(p, q) - i \cdot \varphi| = \varphi/2$. در غیر این صورت بدیهی است که به یکی از این دو نزدیک‌تر و از دیگری دورتر است. آن لوزی که قطرش به آن نزدیک‌تر است، به عنوان i -لوزی انتخاب می‌شود، بنابراین، $|\alpha(p, q) - i \cdot \varphi| < \varphi/2$.

قسمت ب): (فرض خلف) فرض کنید که q روی یال مجاور به p ، در i -لوزی قرار داشته باشد. بنابراین همواره، $|\alpha(p, q)| = \phi/2 + i \cdot \varphi$. در نتیجه، $|\alpha(p, q) - i \cdot \varphi| = \phi/2$. که این، در تناقض با قسمت الف همین مشاهده و فرض اولیه‌ی $\phi < \varphi$ در ابتدای بخش است، که بیان می‌کنند: $|\alpha(p, q) - i \cdot \varphi| \leq \varphi/2 < \phi/2$.

□

قضیه ۶.۲.۴ ([۲]) پوشش لوزی-دلونی $DDS(P)$ یک $(1 + \varepsilon)$ -پوشش برای P با اندازه‌ی $O(n/\varepsilon^2)$ است.

اثبات. به منظور تعیین یک کران، برای ضریب کشش $DDS(P)$ باید نشان داده شود که برای هر جفت نقاط $(p, q) \in P \times P$ ، یک $(1 + \varepsilon)$ -مسیر از p به q در $DDS(P)$ وجود دارد. این، با استقرا روی فاصله‌ی اقلیدسی p و q ، یعنی $\|pq\|$ اثبات می‌شود. برای یک جفت (p, q) ، زاویه‌ای که \vec{pq} با محور مثبت x می‌سازد با $\alpha(p, q)$ نشان داده می‌شود.

پایه‌ی استقرا: فرض کنید (p, q) نزدیک‌ترین جفت از لحاظ فاصله‌ی اقلیدسی در P باشند. نقطه‌ی i متعلق به $\{0, \dots, k-1\}$ ، به طوری که با آن، مقدار $|\alpha(p, q) - i \cdot \varphi|$ حداقل شود، در نظر بگیرید. بدون از دست دادن کلیت مسئله، $i = 0$ فرض می‌شود. بنابراین \diamond را یک 0 -لوزی با سمت چپ‌ترین رأس p و q روی مرز در نظر بگیرید. با توجه به قسمت (ب) از مشاهده‌ی ۵.۲.۴، q حتماً روی یک یال غیرمجاور به p در \diamond واقع شده است، بنابراین با استفاده از قانون سینوس‌ها، به ازای هر نقطه‌ی $r \in \diamond$ ، رابطه‌ی $\|qr\| < \|pq\|$ برقرار است. از آنجایی که (p, q) نزدیک‌ترین جفت است، بنابراین \diamond نمی‌تواند هیچ نقطه‌ی دیگری از P را در برگیرد. در نتیجه، (p, q) یک یال در $DT_0(P)$ است. این، پایه‌ی استقرا را اثبات می‌کند.

گام استقرا: یک جفت (p, q) ، که نزدیک‌ترین جفت نیست را در نظر بگیرید. فرض استقرا بیان می‌کند که برای هر جفت (s, t) در P که $\|st\| < \|pq\|$ ، یک مسیر در $DDS(P)$ ، با طول حداکثر $(1 + \varepsilon) \cdot \|st\|$ وجود دارد. چنانچه در پایه‌ی استقرا مطرح شد، بدون از دست دادن کلیت مسئله فرض می‌شود که مقدار $|\alpha(p, q) - i \cdot \varphi|$ ، به ازای $i = 0$ حداقل می‌شود.

\diamond را یک 0 -لوزی کوچک، به طوری که سمت چپ‌ترین رأسش p باشد، و هیچ نقطه‌ی دیگری از P را شامل نشود، بگیرید. \diamond بزرگ می‌شود، به گونه‌ای که p همچنان سمت چپ‌ترین رأس آن باشد، و \diamond ، یک 0 -لوزی باقی بماند. این بزرگ کردن ادامه می‌یابد تا زمانی که \diamond برای اولین بار به تعدادی (یک یا بیش‌تر) نقطه‌ی $r \in P$ برخورد کند. با این کار، یک 0 -لوزی با p و r روی مرزش، به طوری که هیچ نقطه‌ی دیگری از P را دربر نمی‌گیرد، ایجاد می‌شود. این بدین معنی است که (p, r) ، یک یال در $DT_0(P)$ است، و در نتیجه، یک یال در $DDS(P)$ نیز است. (اگر \diamond هم‌زمان با چندین نقطه، برخورد کند، سپس p باید یک یال به حداقل، یکی از آن‌ها داشته باشد، و بحث بر اساس آن یال ادامه پیدا خواهد کرد.) اگر $r = q$ مسئله در همین مرحله اثبات می‌شود. در غیر این صورت بحث ادامه می‌یابد.

بزرگ کردن \diamond ادامه می‌یابد، تا زمانی که \diamond با q برخورد کند. توجه کنید که در این وضعیت با توجه به قسمت (ب) از مشاهده‌ی ۵.۲.۴، q روی یک یال غیرمجاور به p در \diamond واقع می‌شود، و r در داخل \diamond قرار می‌گیرد ($r \in \diamond$). بنابراین $\|rq\| < \|pq\|$. در این جا می‌توان فرض استقرا را به کار برد و نتیجه گرفت

که یک $(1 + \varepsilon)$ -مسیر از r به q در $DD\mathcal{S}(P)$ وجود دارد. علاوه بر آن یک یال از p به r در $DD\mathcal{S}(P)$ وجود دارد. بنابراین مسیر بین p و q شامل یال \overline{pr} و یک $1 + \varepsilon$ -مسیر از r به q است و از آنجایی که مقدار $|\alpha(p, q) - i \cdot \varphi|$ برای $i = 0$ حداقل است، طبق قسمت (الف) از مشاهده ۵.۲.۴، بردار \vec{pq} یک زاویه با اندازه حداکثر $\varphi/2$ با قطر اصلی می‌سازد، در نتیجه می‌توان لم ۴.۲.۴ را به کار برد و نتیجه گرفت که یک $1 + \varepsilon$ -مسیر بین دو نقطه‌ی p و q وجود دارد. این اثبات می‌کند که ضریب کشش پوشش $1 + \varepsilon$ است.

با توجه به قضیه ۱.۶.۱ برای تعداد یال‌های نمودار ورنوی و مطالب بخش ۷.۱ برای نحوه‌ی ساخت مثلث‌بندی دلونی از نمودار ورنوی، هر $DT_i(P)$ نیز حداکثر $6 - 3n$ یال دارد. بنابراین اندازه‌ی هر مثلث‌بندی دلونی از مرتبه $O(n)$ می‌شود. برای محاسبه‌ی تعداد این مثلث‌بندی‌های دلونی، که برابر با $|\mathcal{D}|$ است، با توجه به رابطه‌ی

$$\varphi = \frac{2\pi}{|\mathcal{D}|} = 2 \arcsin \frac{\varepsilon^2}{4(1+\varepsilon)(2+\varepsilon)}$$

کافی است بسط مکلاورن تابع \arcsin یعنی،

$$\arcsin x = x + \frac{1}{6}x^3 + \frac{3}{40}x^5 + \dots$$

را در نظر بگیرید، از آنجایی که $1 > x = \frac{\varepsilon^2}{4(1+\varepsilon)(2+\varepsilon)} > 0$ بنابراین بزرگ‌ترین جمله‌ی این بسط، برابر جمله‌ی اول، یعنی همان x می‌شود،

$$\frac{2\pi}{|\mathcal{D}|} = 2 \arcsin \frac{\varepsilon^2}{4(1+\varepsilon)(2+\varepsilon)}$$

$$\frac{\pi}{|\mathcal{D}|} > \frac{\varepsilon^2}{4(1+\varepsilon)(2+\varepsilon)}$$

$$|\mathcal{D}| < 4\pi \cdot (1 + 2/\varepsilon^2 + 3/\varepsilon)$$

$$|\mathcal{D}| \in O(1/\varepsilon^2).$$

در نتیجه، اندازه‌ی پوشش برابر با مجموع اندازه‌های این مثلث‌بندی‌های دلونی $DT_i(P)$ می‌شود. یعنی،

$$|DD\mathcal{S}(P)| = |\mathcal{D}| \cdot O(n) = O(n/\varepsilon^2).$$

□

۳.۴ نگهداری پوشش تحت حرکت نقاط

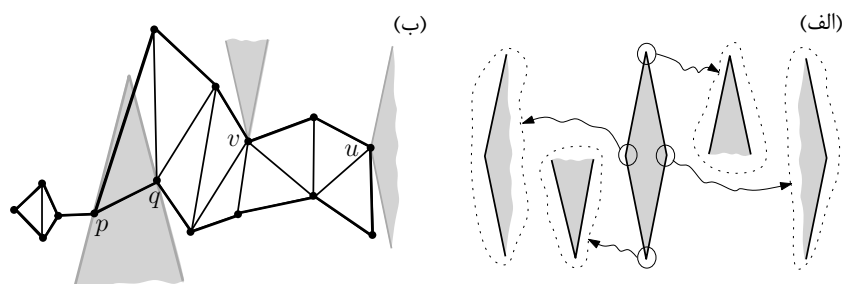
در این بخش نشان داده می‌شود که چگونه پوشش ایستای معرفی شده در بخش قبلی، زمانی که نقاط P به‌طور پیوسته حرکت می‌کنند نگهداری شود. از آنجایی که پوشش لوزی-دلونی اجتماع مثلث‌بندی‌های دلونی بود، برای نگهداری پوشش زمانی که نقاط حرکت می‌کنند، کافی است مثلث‌بندی‌های دلونی (به‌طور جداگانه) نگهداری شوند. بر اساس آن چه که توسط کاراولاس و گیباس [۳۰] مطالعه شده است، نگهداری مثلث‌بندی‌های دلونی تحت حرکت نقاط کاملاً ساده است؛ زیرا برای به‌روزرسانی مثلث‌بندی زمانی که یک یال آن غیرمجاز می‌شود (بروز یک رویداد)، تنها اصلاح تاییدیه‌های مربوط به آن یال (به‌روزرسانی محلی) برای معتبرماندن کل مثلث‌بندی (به‌روزرسانی سراسری) کافی است. این حقیقت که مثلث‌بندی‌های دلونی نسبت به تابع فاصله‌ی محدب d_i تعریف شده است خیلی تفاوتی نمی‌کند. بنابراین برای نگهداری یک مثلث‌بندی دلونی، کافی است تمام یال‌های مثلث‌بندی و تاییدیه‌های متناظر با آن‌ها را در ساختار مناسبی نگهداری کرد و در صورت غیرمجاز شدن یالی، اصلاحات لازم را انجام داد.

برای کامل شدن بحث در ادامه توصیف می‌شود که نگهداری چه‌طور انجام می‌شود و جزئیات بیش‌تر روی ساختار مثلث‌بندی دلونی در L_1 -متریک از نقاط متحرک در بخش ۲.۳ بیان شده است. برای سادگی فرض می‌شود که نقاط در وضعیت عمومی هستند و موارد خاص^۷، به‌ویژه، این که چهار نقطه از P روی مرز یک i -لوزی یکسان قرار گیرند، تنها در زمان رویدادها رخ می‌دهند و هیچ‌گاه در فاصله‌ی زمانی بین دو رویداد متوالی، چنین مواردی رخ نخواهد داد.

همان‌طور که در زیربخش ۳.۹.۱، بیان شد خواص مثلث‌بندی دلونی در متر d_\diamond ، تقریباً مشابه با متر اقلیدسی بود، تنها در بیان ویژگی‌های مثلث‌بندی از دایره‌ی این متر، یعنی لوزی \diamond استفاده شد (جدول ۱-۱ را ببینید). متر d_i با یک انتقال ساده هم‌ارز با متر d_\diamond است، بنابراین به‌راحتی می‌توان تمام ویژگی‌های مثلث‌بندی دلونی در متر d_\diamond را، با جایگزینی i -لوزی به‌جای لوزی \diamond ، برای متر d_i بیان کرد. در ادامه به معرفی این ویژگی‌ها پرداخته می‌شود.

چنانچه در بخش ۴.۱ بیان شد، زمانی که از متر اقلیدسی استفاده می‌شود، مثلث‌بندی دلونی یک مجموعه از نقاط P برابر با یک مثلث‌بندی از غشای محدب P می‌شود. اما برای $DT_i(P)$ نمایش غشا متفاوت است و می‌تواند غیرمحدب باشد (نتیجه‌ی ۴.۹.۱). در ادامه این غشا دقیق‌تر تعریف می‌شود.

^۷Degeneracies

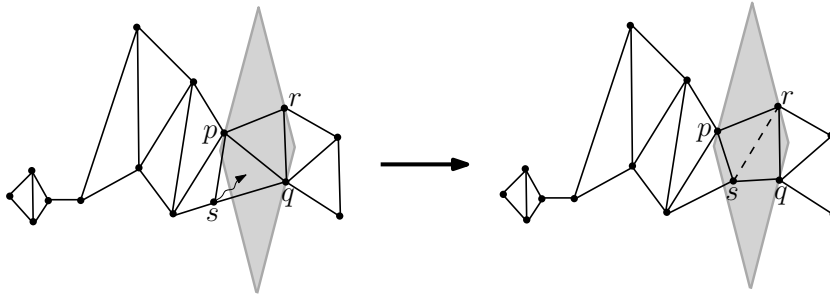


شکل ۵.۴: (الف): چهار i -مخروط متفاوت برای یک i -لوزی، شکل (ب): (p, q) یک یال غشا و u و v رؤس غشا

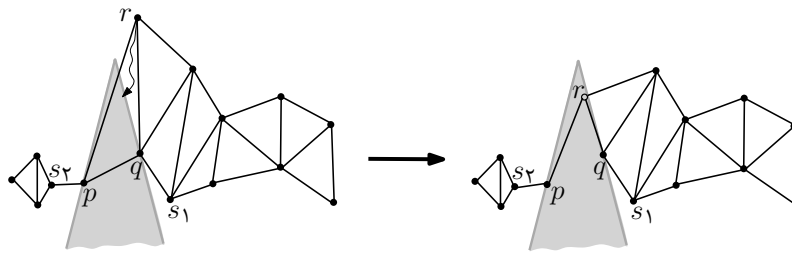
همان‌طور که در زیربخش ۳.۹.۱ بیان شد، به‌طور مشابه، هر رأس v از یک i -لوزی \diamond یک مخروط با رأس v را، با گسترش دو یال مجاور به v به دو نیم‌خط تعریف می‌کند. چنین مخروط‌هایی i -مخروط‌ها نامیده می‌شوند. شکل ۵.۴ قسمت (الف) را ببینید. یک نقطه‌ی $p \in P$ یک رأس d_i -غشا P است اگر و تنها اگر یک i -مخروط با رأس p وجود داشته باشد که هیچ نقطه‌ی دیگری از P را دربر نگیرد و یک جفت نقاط (p, q) یک یال غشا را شکل می‌دهند اگر و تنها اگر یک i -مخروط با p و q روی مرزش وجود داشته باشد که هیچ نقطه‌ی دیگری از P را دربر نگیرد. شکل ۵.۴ قسمت (ب) را ببینید. d_i -غشا P همبند است و نیز هر رأس آن روی وجه خارجی است. بنابراین d_i -غشا، صفحه را به یک وجه نامحدود و یک مجموعه از وجه‌های محدود افزاز می‌کند. مثلث‌بندی دلونی $DT_i(P)$ شامل d_i -غشا و یک مثلث‌بندی از وجه‌های محدود این غشا خواهد بود که مثلث‌بندی وجه‌های محدود با یال‌های داخلی شکل می‌گیرد.

حالا فرض کنید که نقاط شروع به حرکت کردن می‌کنند. چه‌زمانی مثلث‌بندی دلونی $DT_i(P)$ تغییر می‌کند؟ از آن‌جایی که دو نوع یال در مثلث‌بندی وجود دارد، برای تعیین چنین تغییراتی دو نوع تاییدیه، یکی برای یال‌های غشا و یکی برای یال‌های داخلی $DT_i(P)$ تعریف می‌شود. در ادامه این دو نوع تاییدیه معرفی می‌شوند و در مورد آن‌چه که در زمان نامعتبر شدن این تاییدیه‌ها نیاز می‌شود، توضیح داده می‌شود.

تاییدیه‌های یال داخلی: در بین رویدادها، هر یال داخلی (p, q) مجاور به دو مثلث است. نقاط r و s را دو رأس دیگر این دو مثلث در نظر بگیرید. در نتیجه برای یال داخلی (p, q) دو تاییدیه $NotInDiamond$ تعیین می‌شود که یکی تایید می‌کند که s خارج از i -لوزی منحصر به فردی است که از p, q و r می‌گذرد قرار دارد و دیگری نیز به‌طور مشابه، تایید می‌کند که r خارج از i -لوزی منحصر به فردی است که از p, q و s می‌گذرد قرار دارد. شکل ۶.۴ را ببینید. زمانی که یک چنین تاییدیه‌ای نامعتبر می‌شود یک چرخش یال انجام



شکل ۶.۴: نامعتبر شدن تاییدیهی *NotInDiamond* رأس s



شکل ۷.۴: نامعتبر شدن تاییدیهی *NotInCone* رأس r ، مربوط به یال غشای (p, q)

می‌شود. برای نمونه در هنگام نامعتبر شدن تاییدیهی *NotInDiamond* رأس s ، یال (p, q) را با یال (s, r) جابه‌جا می‌شود.

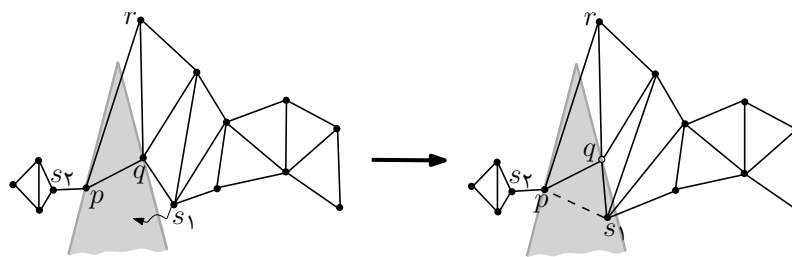
تاییدیه‌های یال غشا: هر یال غشا (p, q) مجاور به حداکثر یک مثلث و نیز مجاور به حداکثر چهار یال غشای دیگر است. نقطه‌ی r را رأس سوم این مثلث (در صورتی که وجود داشته باشد) و s_1, \dots, s_k را نقاط انتهایی یال‌های غشای مجاور به یال (p, q) در نظر بگیرید، به طوری که $k \leq 4$ به تعداد یال‌های غشای مجاور اشاره می‌کند. (این امکان هست که r یکی از s_i ها باشد. یعنی مثلثی مجاور به یال (p, q) وجود نداشته باشد و $k \leq 5$) بنابراین حداکثر پنج تاییدیهی *NotInCone* نیاز است که تایید می‌کنند که r و s_1, \dots, s_k خارج از i -مخروط منحصر به فردی هستند که از p و q می‌گذرد و هیچ نقطه‌ی دیگری از P را دربر نمی‌گیرد. زمانی که یک تاییدیهی *NotInCone* نامعتبر می‌شود (بروز یک رویداد)، دو حالت وجود دارد:

۱. تاییدیهی *NotInCone* برای رأس r (رأس سوم مثلثی که یال (p, q) یک ضلع آن است) نامعتبر

شود. (شکل ۷.۴)

۲. تاییدیهی *NotInCone* برای رأس s_i به طوری که $1 \leq i \leq 4$ (رأس مجاور به p یا q از طریق یک

یال غشا) نامعتبر شود. (شکل ۸.۴)



شکل ۸.۴: نامعتبر شدن تاییدیه *NotInCone* رأس s_1 ، مربوط به یال غشای (p, q)

با توجه به این که غشای مثلث‌بندی را می‌توان با لیست حلقوی از رأس‌های مثلث‌بندی نگهداری کرد، برای به‌روزرسانی مثلث‌بندی به ترتیب برای حالت‌های ۱ و ۲ عمل‌های زیر انجام می‌شود:

۱. رأس r به‌عنوان یک رأس غشا، در مکان متناظرش در لیست رئوس غشای مثلث‌بندی درج می‌شود و

یال (p, q) از مثلث‌بندی حذف می‌شود. (شکل ۷.۴)

۲. که یک رأس غشا بود از مکان متناظرش در لیست رئوس غشای مثلث‌بندی حذف می‌شود و یال

در مثلث‌بندی درج می‌شود. (شکل ۸.۴)

۴.۴ تحلیل پوشش وابسته به حرکت

در فصل ۳ نشان داده شد، که تعداد تغییرات توپولوژیکی در مثلث‌بندی دلونی بر اساس متر L_∞ (یا L_1)، برای نقاطی که با سرعت‌های ثابت حرکت می‌کنند (حرکت خطی) برابر با $O(n^2 \alpha(n))$ است. در بخش ۳.۳ بیان شد که همین کران برای مثلث‌بندی دلونی بر اساس تابع فاصله‌ی محدب d_\diamond نیز برقرار است. (درحقیقت، d_\diamond بعد از به‌کار بردن یک انتقال مناسب هم‌ارز با d_i است.) سپس بیان شد زمانی که نقاط دارای مسیرهای حرکت شبه‌جبری از درجه‌ی حداکثر s هستند، نیز کران برای تعداد رویدادها برابر $O(n \lambda_{s+2}(n))$ می‌شود (نتیجه‌ی ۱.۳.۳). بنابراین پوشش لوزی-دلونی نیز که اجتماع تعداد $O(1/\epsilon^2)$ مثلث‌بندی لوزی-دلونی است، دارای تعداد رویدادهایی از مرتبه $O((n/\epsilon^2) \cdot \lambda_{s+2}(n))$ می‌شود.

لازم است برای هر تاییدیه، زمان نامعتبر شدن آن (زمان رویداد متناظر با آن) در یک صف از رویدادها با اولویت زمانی، ذخیره شود. پیاده‌سازی صف اولویت برای رویدادها می‌تواند با یک هرم بیشینه^۱ انجام شود، به‌طوری‌که رویدادی که اولویت زمانی بیشتری دارد (زودتر از تمام رویدادهای دیگر رخ می‌دهد) در ریشه‌ی

^۱Max heap

هرم قرار می‌گیرد. بنابراین پردازش هر رویداد (که به حذف آن رویداد از هرم و درج رویداد جدیدی در آن نیاز دارد)، به طوری که بتوان با یک ترتیب درست از لحاظ زمانی به سایر رویدادها در آینده نیز دسترسی داشت، از مرتبه‌ی $O(\log n)$ خواهد بود. همچنین بررسی شد که، هر زمان که تاییدیه‌ی یک یال داخلی یا تاییدیه‌ی یک یال غشا نامعتبر می‌شود می‌توان پوشش را در زمان $O(1)$ (یک تعداد ثابت عملیات) به‌روزرسانی کرد. در ادامه، قضیه‌ی زیر نتیجه می‌شود.

قضیه ۱.۴.۴ ([۲]) پوشش لوزی-دلونی وابسته به حرکت یک مجموعه از n نقطه در صفحه می‌تواند هر رویداد را در زمان $O(1)$ ، (به‌علاوه‌ی زمان $O(\log n)$ برای به‌روزرسانی صف رویداد) پردازش کند. همچنین تعداد رویدادهای این پوشش، با فرض این که نقاط دارای مسیرهای حرکت شبه‌جبری از درجه‌ی حداکثر s هستند، برابر با $O(n\lambda_{s+2}(n))$ است.

از ترکیب این نتیجه با قضیه‌ی ۶.۲.۴، نتیجه‌ی اصلی زیر حاصل می‌شود.

نتیجه ۲.۴.۴ ([۲]) مجموعه‌ی P از n نقطه‌ی متحرک در صفحه را در نظر بگیرید. فرض کنید ε یک پارامتر مثبت داده شده باشد. آنگاه می‌توان یک $\varepsilon + 1$ -پوشش وابسته به حرکت برای P نگهداری کرد، به طوری که ویژگی‌های زیر را داشته باشد،

۱. تعداد یال‌های پوشش در تمام زمان‌ها $O(n/\varepsilon^2)$ باشد؛

۲. هزینه‌ی به‌روزرسانی ساختار برای هر رویداد برابر با $O(1)$ باشد، علاوه بر این، به زمان $O(\log n)$ نیز برای به‌روزرسانی صف رویداد نیاز است؛

۳. با فرض این که نقاط دارای مسیرهای حرکت شبه‌جبری از درجه‌ی حداکثر s هستند، تعداد رویدادها برابر با $O(n\lambda_{s+2}(n))$ باشد.

در فصل ۲، چهار معیار برای ارزیابی کیفیت ساختارهای وابسته به حرکت معرفی شده‌اند: پاسخ‌گویی، کارایی، فشردگی و محلی بودن داده‌ها. طبق ویژگی ۲ از نتیجه‌ی ۲.۴.۴، پوشش وابسته به حرکت پاسخ‌گوست و با توجه به آن‌چه که توسط گائو و همکارانش در [۲۷] مطالعه شده است، تعداد رویدادهای یک پوشش وابسته به حرکت دارای کران پایین $\Omega(n^2)$ است، در نتیجه نسبت تعداد کل رویدادها به تعداد رویدادهای خارجی در بدترین وضعیت لگاریتمی خواهد شد، $\beta(n)$ یک تابع است که رشد آن به‌شدت آهسته است و از رشد هر تابعی

از جمله توابع لگاریتمی کم‌تر است.)

$$\frac{O(n\lambda_{s+2}(n))}{\Omega(n^2)} = \frac{O(n^2\beta(n))}{\Omega(n^2)} = O(\beta(n)) \in O(\log n).$$

بنابراین پوشش لوزی-دلونی وابسته به حرکت، کارا نیز است. هرچند که پوشش دو معیار مهم‌تر ارزیابی کیفیت را داراست (پاسخ‌گویی و کارایی)، اما محلی نیست؛ زیرا یک نقطه‌ی متحرک می‌تواند به $\Omega(n)$ تاییدیه مرتبط باشد [۳۰]. لازم به ذکر است که پوشش معیار فشردگی را نیز دارد؛ زیرا اندازه‌ی صف رویداد برابر با $O(n/\varepsilon^2)$ است.

فصل ۵

یک پوشش هندسی وابسته به حرکت در فضای

\mathbb{R}^d

در این فصل یک $\varepsilon + 1$ -پوشش وابسته به حرکت برای مجموعه‌ی n نقطه در \mathbb{R}^d معرفی می‌شود. این پوشش زمانی که نقاط آن حرکت می‌کنند، می‌تواند به صورتی کارا نگهداری شود. فرض می‌شود که مسیر حرکت نقاط با توابع شبه‌جبری از درجه‌ی محدود توصیف می‌شوند (بخش ۱۰.۱ را ببینید). لازم به ذکر است که مطالب این فصل بر اساس مقاله‌ی آبام^۱ و دی برگ^۲ [۱] تنظیم شده است.

۱.۵ مقدمه

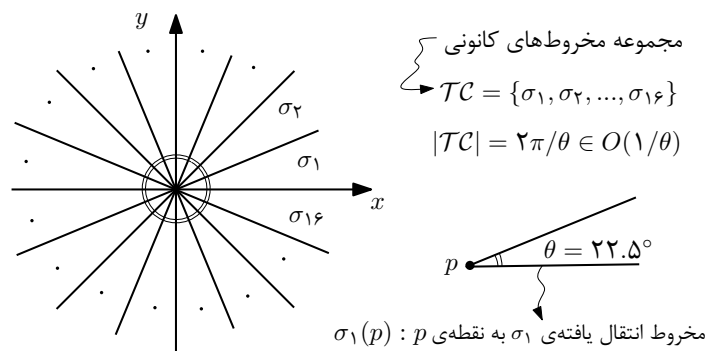
شاید اولین راهی که برای ساخت پوشش در فضای \mathbb{R}^d به ذهن برسد، این باشد که پوشش معرفی شده در فصل ۴ برای صفحه، به فضای \mathbb{R}^d تعمیم داده شود، اما لازمی این کار استفاده از مثلث‌بندی دلونی در ابعاد بالاتر است که در این صورت دارای تعداد مربعی یال خواهد بود و دیگر یک پوشش با اندازه‌ی خطی به دست نخواهد آمد. علاوه بر این پوشش معرفی شده در فصل ۴، معیار محلی بودن داده‌ها را هم ندارد و در بدترین وضعیت، تغییر طرح پرواز یک نقطه به $O(n)$ زمان نیاز خواهد داشت. بنابراین در این فصل از یک دیدگاه دیگر، پوشش جدیدی در فضای \mathbb{R}^d ($d \geq 2$) معرفی خواهد شد که علاوه بر داشتن ویژگی‌های مطلوب مورد نظر، محلی نیز هست و وابستگی آن به ε هم کم‌تر است. مثلاً برای $d = 2$ ، وابستگی آن به ε از مرتبه‌ی $O(1/\varepsilon)$ است در صورتی که برای پوشش فصل ۴ از مرتبه‌ی $O(1/\varepsilon^2)$ است.

قبل از معرفی دیدگاه جدید برای ساخت پوشش، لازم است نکاتی بیان شوند. با ساختار t -پوشش‌ها در بخش ۱۰.۱ آشنا شدید. نکته‌ی قابل توجه برای ساخت پوشش‌ها این است که می‌توان یک t -پوشش را از یک مجموعه نقاط ساخت یا این که آن را از یک گراف ساخت ([۲۶]، فصل ۱). زمانی که یک t -پوشش از یک گراف ساخته می‌شود، رأس‌های آن همان رأس‌های گراف است اما یال‌های آن زیرمجموعه‌ای از یال‌های آن گراف خواهند بود. در این فصل t -پوشش از یک گراف اقلیدسی کامل \mathcal{G} ساخته می‌شود که در ادامه معرفی خواهد شد.

همان‌طور که در بخش ۱۰.۱ بیان شد، یک گراف وزن‌دار $\mathcal{G} = (P, E)$ یک گراف اقلیدسی است اگر رأس‌های آن، یعنی مجموعه‌ی P ، زیرمجموعه‌ای از فضای اقلیدسی \mathbb{R}^d باشند و وزن هر یال از E برابر با فاصله‌ی اقلیدسی بین دو نقطه‌ی انتهایی آن یال باشد. در این فصل گراف اقلیدسی \mathcal{G} یک گراف اقلیدسی کامل

^۱Abam

^۲de Berg



شکل ۱.۵: مخروط‌های کانونی در صفحه و برای $\theta = 22.5^\circ$

در نظر گرفته می‌شود. یعنی به ازای هر دو نقطه از P ، یک یال در E وجود خواهد داشت. یک t -پوشش از \mathcal{G} ، یک زیرگراف $\mathcal{S} = (P, E_S)$ از \mathcal{G} است، به طوری که برای هر دو نقطه‌ی $p, q \in P$ رابطه‌ی $d_S(p, q) \leq t \cdot \|pq\|$ برقرار است، که $d_S(p, q)$ به کوتاه‌ترین مسیر بین p و q در \mathcal{S} اشاره می‌کند.

۲.۵ یک پوشش ایستای مناسب

فرض کنید P یک مجموعه از n نقطه در \mathbb{R}^d و ε یک ثابت مثبت داده شده باشد. در ابتدا الگوریتم جدیدی برای ساخت یک پوشش از P معرفی می‌شود و سپس روش نگهداری این پوشش در حالتی که نقاط حرکت می‌کنند بیان می‌شود.

۱.۲.۵ ساخت پوشش بر اساس دیدگاه θ -گراف

ساخت پوشش بر اساس دیدگاه θ -گراف می‌باشد [۲۱، ۳۱]. این دیدگاه به این شکل مطرح می‌شود: فرض می‌شود θ یک ثابت کوچک مناسب (وابسته به ε) است و θ -مخروط به این شکل تعریف می‌شود: ناحیه‌ی اشتراک d نیم‌فضای غیرموازی، به طوری که زاویه‌ی هر دو اشعه‌ی صادرشده از رأس مخروط و در داخل آن، حداکثر θ باشد. مجموعه‌ی TC را یک مجموعه از θ -مخروط‌های مجزا در نظر بگیرید، به طوری که رأس همه‌ی آنها مبدأ باشد و با هم، کل فضای \mathbb{R}^d را بپوشانند (ناحیه‌ی اشتراک آنها تنها می‌تواند مرز مخروط‌ها باشد). به مخروط‌های متعلق به TC مخروط‌های کانونی^۳ گفته می‌شود (به‌عنوان نمونه، می‌توانید در شکل ۱.۵ این

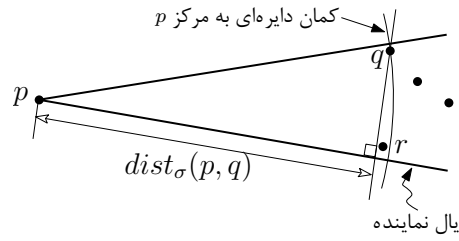
^۳Canonical cones

مخروطها را در صفحه و برای $\theta = 22/5^\circ$ ببینید). ثابت می‌شود که $|TC| \in O(1/\theta^{d-1})$ است (یعنی تعداد مخروط‌های این مجموعه از مرتبه $O(1/\theta^{d-1})$ است)، این اثبات در فصل ۵ کتاب [۳۷] آمده است. برای نمونه، زمانی که $d = 2$ است (برای نقاط در صفحه)، مخروط‌های کانونی می‌توانند با رسم $O(1/\theta)$ اشعه که از مبدأ صادر می‌شوند و زاویه‌ی بین هر دو اشعه‌ی متوالی حداکثر θ است، به‌دست آیند. شکل ۱.۵ را ببینید. برای یک مخروط $\sigma \in TC$ و یک نقطه‌ی $p \in \mathbb{R}^d$ ، نسخه‌ی انتقال یافته‌ی σ که رأس آن روی نقطه‌ی p منطبق شده است، با نماد $\sigma(p)$ نشان داده می‌شود. شکل ۱.۵ را ببینید.

θ -گراف برای مجموعه نقاط P ، با اضافه کردن حداکثر $|TC|$ یال برای هر نقطه‌ی $p \in P$ ساخته می‌شود. بدین صورت که، برای هر مخروط $\sigma \in TC$ ، نقطه‌ی p به نقطه‌ی $q \in P \cap \sigma(p)$ ، که نزدیک‌ترین نقطه به p است، متصل می‌شود (اگر q منحصر به فرد نبود، به دلخواه به یکی از آن‌ها وصل می‌شود). منظور از نزدیک‌ترین نقطه، نزدیک‌ترین از لحاظ متر اقلیدسی نیست؛ زیرا پیدا کردن نقطه‌ای از داخل $\sigma(p)$ که نزدیک‌ترین نقطه (اقلیدسی) به p باشد هزینه‌بر است (برای نمونه، برای نقاط در صفحه، نیاز به مرتب‌سازی دو بعدی، بر اساس مولفه‌ی x و y است تا نزدیک‌ترین نقطه به p مشخص شود) و علاوه بر این، نگهداری آن نیز، زمانی که مسئله، وابسته به حرکت می‌شود سخت است. بنابراین در تعریف نزدیک‌ترین نقطه، در هنگام ساخت θ -گراف، تغییر اندکی ایجاد می‌شود تا این مشکلات حل شوند. بدین منظور، برای هر $\sigma \in TC$ ، یکی از یال‌هایش به عنوان **یال نماینده**^۴ انتخاب می‌شود و $dist_\sigma(p, q)$ به عنوان فاصله‌ی p تا تصویر عمودی q روی یال نماینده‌ی $\sigma(p)$ تعریف می‌شود. اکنون به‌جای اتصال p به یک نقطه‌ی $q \in \sigma(p)$ که نزدیک‌ترین نقطه به p از لحاظ فاصله‌ی اقلیدسی است، p به یک نقطه‌ی q به‌طوری که $dist_\sigma(p, q)$ کم‌ترین مقدار را داشته باشد، وصل می‌شود. شکل ۲.۵ را ببینید. در این بخش هر جا که از نزدیک‌ترین نقطه به p در $\sigma(p)$ صحبت می‌شود منظور یک چنین نقطه‌ای است.

در این روش اگر زاویه‌ی θ به‌گونه‌ای انتخاب شود که $\cos \theta - \sin \theta \geq 1/(1 + \varepsilon)$ و $0 < \theta < 2\pi/9$ ، آنگاه θ -گراف حاصل، یک $1 + \varepsilon$ -پوشش می‌شود، فصل ۵ کتاب [۳۷] را ببینید. (همچنین [۳۱، ۲۱] را ببینید). بنابراین با توجه به این شرط‌ها، مشخص است که زاویه‌ی θ و ثابت ε رابطه‌ی مستقیم با هم خواهند داشت و هرچه ثابت ε کوچک‌تر انتخاب شود ($\varepsilon \rightarrow 0$)، آنگاه زاویه‌ی θ نیز کوچک‌تر می‌شود ($\theta \rightarrow 0$). در این وضعیت $1 + \varepsilon \sim 1 + \theta$ است. در ادامه لم زیر بیان می‌شود تا بعداً با استفاده از آن ثابت شود که θ -گراف با این روش، یک $1 + \varepsilon$ -پوشش را تولید می‌کند.

^۴Representative edge



شکل ۲.۵: نقطه‌ی q از لحاظ فاصله $dist_\sigma$ نزدیک‌ترین به نقطه p است.

لم ۱.۲.۵ ([۱]) فرض کنید که \mathcal{TC} یک مجموعه از θ -مخروط‌ها و رابطه‌ی $\cos 2\theta - \sin 2\theta \geq 1/(1 + \varepsilon)$ برقرار باشد. همچنین فرض کنید که $\sigma \in \mathcal{TC}$ یک مخروط و q و r دو نقطه متعلق به $\sigma(p)$ باشند، به طوری که $dist_\sigma(p, r) \leq dist_\sigma(p, q)$ باشد، آنگاه

$$\|pr\| + (1 + \varepsilon) \cdot \|rq\| \leq (1 + \varepsilon) \cdot \|pq\|.$$

اثبات. اثبات برای دو حالت در نظر گرفته می‌شود:

الف) $\|pq\| \geq \|pr\|$: شکل ۳.۵ قسمت (الف) را ببینید. در مثلث pqr ، زاویه‌ی qpr کوچک‌تر از θ است و نیز یک زاویه‌ی کوچک است، بنابراین از این مطلب و نیز از فرض $\|pq\| \geq \|pr\|$ ، این نتیجه حاصل می‌شود که یال (p, q) بزرگ‌ترین یال این مثلث است. بنابراین $\|pq\| \geq \|qr\|$ و رابطه‌ی

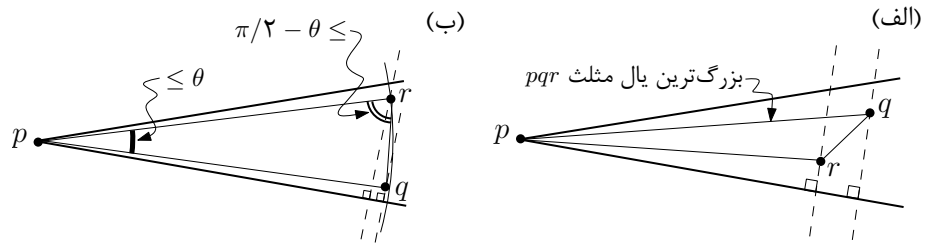
$$\cos \angle qpr - \sin \angle qpr = \frac{\|pr\|}{\|pq\|} - \frac{\|qr\|}{\|pq\|} < 1$$

برقرار است. با توجه به این رابطه و همچنین رابطه‌ی نامساوی مثلثی در مثلث pqr نتیجه‌ی زیر حاصل می‌شود:

$$\begin{aligned} \|rq\| &\leq \|pq\| - \|pr\| \\ &\leq \|pq\| - (\cos \angle qpr - \sin \angle qpr) \|pr\|. \end{aligned}$$

بنابراین،

$$\begin{aligned} \|pr\| + (1 + \varepsilon) \cdot \|rq\| &\leq \|pr\| + (1 + \varepsilon) \cdot (\|pq\| - (\cos \theta - \sin \theta) \|pr\|) \\ &\leq (1 + \varepsilon) \cdot \|pq\|, \end{aligned}$$



شکل ۳.۵: شکل مربوط به اثبات لم ۱.۲.۵.

زیرا

$$\cos \theta - \sin \theta \geq \cos 2\theta - \sin 2\theta \geq 1/(1 + \varepsilon).$$

(ب) $\|pq\| < \|pr\|$: از آنجایی که $dist_\sigma(p, r) \leq dist_\sigma(p, q)$ رابطه‌ی

$$\angle prq > \pi/2 - \angle qpr \quad (1-5)$$

برقرار است. (شکل ۳.۵ قسمت (ب) را برای نمایش این حالت در صفحه ببینید.) علاوه بر این، از

آنجایی که پاره‌خط‌های pq و pr در داخل $\sigma(p)$ هستند، رابطه‌ی

$$0 \leq \angle qpr \leq \theta \quad (2-5)$$

برقرار خواهد شد. حال با توجه به رابطه‌ی ۲-۵، می‌توان با جای‌گذاری $\angle qpr$ در رابطه‌ی ۱-۵، رابطه‌ی

$$\pi/2 - \theta \leq \angle prq \leq \pi/2 \quad (3-5)$$

را به‌دست آورد. از رابطه‌ی ۲-۵، رابطه‌ی $\sin \angle qpr \leq \sin \theta$ و از رابطه‌ی ۳-۵ نیز رابطه‌ی $\sin \angle prq \geq$

$\cos \theta$ نتیجه می‌شود. پس،

$$\|rq\|/\|pq\| = \sin \angle qpr / \sin \angle prq \leq \sin \theta / \cos \theta$$

برقرار است. بنابراین،

$$\begin{aligned} \|pr\| + (1 + \varepsilon) \cdot \|rq\| &\leq \|pq\| + \|rq\| + (1 + \varepsilon) \cdot \|rq\| \\ &= (1 + (2 + \varepsilon)(\|rq\|/\|pq\|))\|pq\| \\ &\leq (1 + (2 + \varepsilon)(\sin \theta / \cos \theta))\|pq\|. \end{aligned}$$

برای اثبات لم کافی است نشان داده شود: $(1 + (2 + \varepsilon)(\sin \theta / \cos \theta)) \leq 1 + \varepsilon$. این نامساوی می‌تواند به سادگی به $(\cos \theta - \sin \theta) / (\cos \theta + \sin \theta) \geq 1 / (1 + \varepsilon)$ کاهش یابد. به این صورت:

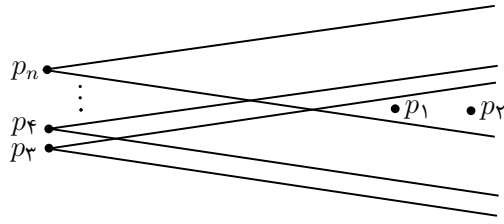
$$\begin{aligned}
 1 + (2 + \varepsilon) \left(\frac{\sin \theta}{\cos \theta} \right) \leq 1 + \varepsilon &\sim \frac{\cos \theta + (2 + \varepsilon) \sin \theta}{\cos \theta} \leq 1 + \varepsilon \\
 &\sim \cos \theta + (2 + \varepsilon) \sin \theta \leq (1 + \varepsilon) \cos \theta \\
 &\sim \cos \theta + \sin \theta + (1 + \varepsilon) \sin \theta \leq (1 + \varepsilon) \cos \theta \\
 &\sim \cos \theta + \sin \theta \leq (1 + \varepsilon)(\cos \theta - \sin \theta) \\
 &\sim \frac{\cos \theta - \sin \theta}{\cos \theta + \sin \theta} \geq \frac{1}{1 + \varepsilon}.
 \end{aligned}$$

با توجه به فرض $\cos 2\theta - \sin 2\theta \geq 1 / (1 + \varepsilon)$ روابط زیر برقرار خواهد بود:

$$\begin{aligned}
 (\cos \theta - \sin \theta) / (\cos \theta + \sin \theta) &= (\cos^2 \theta - \sin^2 \theta) / (\cos \theta + \sin \theta)^2 \\
 &= (\cos 2\theta) / (1 + \sin 2\theta) \\
 &\geq (\cos 2\theta - \sin 2\theta) / (1 + \sin 2\theta - \sin 2\theta) \\
 &\geq \cos 2\theta - \sin 2\theta \\
 &\geq 1 / (1 + \varepsilon)
 \end{aligned}$$

□

لم ۱.۲.۵ به این اشاره می‌کند که با الحاق یال pr به یک $1 + \varepsilon$ -مسیر از r به q ، یک $1 + \varepsilon$ -مسیر از p به q به دست می‌آید. (یادآوری: یک $1 + \varepsilon$ -مسیر از p به q ، یک مسیر با طول حداکثر $(1 + \varepsilon) \cdot \|pq\|$ می‌باشد.) این لم مطرح شد تا در ادامه، به کمک آن و با روش استقرا، نشان داده شود که اگر هر نقطه‌ی p به نزدیک‌ترین



شکل ۴.۵: درجه‌ی نقطه‌ی p_1 در θ -گراف، $n - 1$ است.

نقطه در هر کدام از مخروط‌هایش متصل شود، یک $1 + \varepsilon$ -پوشش به دست می‌آید. از این جا به بعد، یک زاویه‌ی θ تحت فرض‌های $\theta = \Omega(\varepsilon)$ و $\cos 2\theta - \sin 2\theta \geq 1/(1 + \varepsilon)$ تعیین می‌شود، همچنین یک مجموعه‌ی TC از θ -مخروط‌ها با اندازه‌ی $|TC| = O(1/\theta^{d-1}) = O(1/\varepsilon^{d-1})$ نیز تعیین می‌گردد.

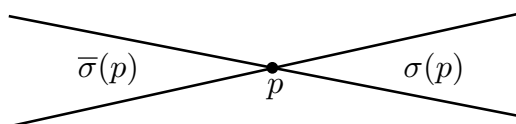
یکی از معایب روش θ -گراف این است که درجه رؤس می‌تواند خیلی زیاد باشد. برای نمونه، نقطه‌ی p_1 در شکل ۴.۵، نزدیک‌ترین نقطه به رأس‌های $n - 2$ مخروط نشان داده شده در شکل است و درجه‌ی آن حداقل $n - 2$ است. هنگامی که $1 + \varepsilon$ -پوشش، با روش θ -گراف به دست آید، این موضوع (درجه‌ی زیاد)، به‌ویژه هنگام وابسته به حرکت پوشش (زمانی که نقاط حرکت می‌کنند)، باعث بروز این مشکل می‌شود: زمانی که p_1 و p_2 در شکل ۴.۵ ترتیب‌شان را عوض می‌کنند (تصویر این نقاط روی یال نماینده‌ی مخروط‌هایشان با هم جابه‌جا می‌شود)، یعنی یک رویداد رخ می‌دهد، لازم است برای پردازش این رویداد، یال‌های $p_i p_1$ برای $3 \leq i \leq n$ با یال‌های جایگزین شوند. بنابراین به زمان $\Omega(n)$ برای پاسخ به یک چنین رویدادی نیاز است و این ساختار وابسته به حرکت پاسخ‌گو نخواهد بود. برای رفع این مشکل، انواعی از θ -گراف وجود دارد، از جمله θ -گراف مرتب‌شده^۵ [۱۲]، که در آن‌ها، درجه‌ی هر نقطه کم است، اما نگهداری پوشش‌های حاصل از آن‌ها هنگام حرکت نقاط سخت است. در ادامه توضیح داده می‌شود که چطور می‌توان با اصلاح دیدگاه θ -گراف، از یک راه جدید یک پوشش با حداکثر درجه‌ی $O(\log^d n)$ را به دست آورد.

۲.۲.۵ ساخت پوشش از CSPD‌ها

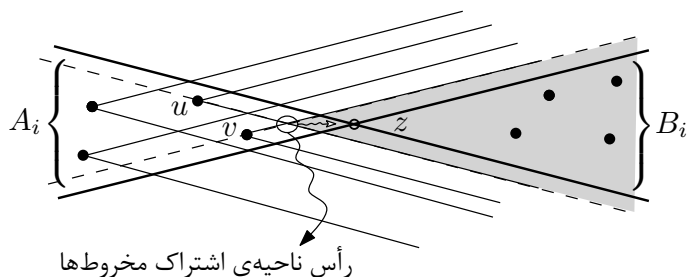
ابتدا لازم است مفهوم تجزیه‌ی جفت‌های مخروط-مجزا^۶ تعریف شود. قبل از آن لازم است مخروط مزدوج $\sigma(p)$ ، که با نماد $\bar{\sigma}(p)$ نشان داده می‌شود، به‌عنوان تصویر $\sigma(p)$ نسبت به نقطه‌ی p تعریف شود. شکل ۵.۵ را

^۵Ordered θ -graph

^۶Cone-Separated Pairs Decomposition



شکل ۵.۵: $\bar{\sigma}(p)$: مخروط مزدوج $\sigma(p)$



شکل ۶.۵: انتخاب نقطه‌ی z

ببینید.

تعریف ۲.۲.۵ ([۱]) فرض کنید P یک مجموعه از نقاط در \mathbb{R}^d و $\sigma \in TC$ یک مخروط باشد. یک تجزیه‌ی جفت مخروط-مجزا، که به اختصار با $CSPD$ نمایش داده می‌شود، برای P نسبت به σ ، یک مجموعه‌ی $\Psi_\sigma = \{(A_1, B_1), \dots, (A_m, B_m)\}$ است به طوری که A_i ها و B_i ها، زیرمجموعه‌هایی از P هستند و Ψ_σ دارای این خواص است:

۱. برای هر دو نقطه‌ی $p, q \in P$ به طوری که $q \in \sigma(p)$ است، یک جفت منحصر به فرد $(A_i, B_i) \in \Psi_\sigma$ وجود دارد، به طوری که $p \in A_i$ و $q \in B_i$ است.
۲. برای هر جفت $(A_i, B_i) \in \Psi_\sigma$ و برای هر دو نقطه‌ی $p \in A_i$ و $q \in B_i$ ، رابطه‌ی $q \in \sigma(p)$ و در نتیجه رابطه‌ی $p \in \bar{\sigma}(q)$ برقرار است.

شرط ۲ می‌تواند به صورت شهودی به این شکل بیان شود: برای هر i ، باید یک نقطه‌ی $z \in \mathbb{R}^d$ وجود داشته باشد، به طوری که روابط $A_i \subset \bar{\sigma}(z)$ و $B_i \subset \sigma(z)$ برقرار باشند. برای نمونه، z می‌تواند به این شکل انتخاب شود: تمام مخروط‌های انتقال یافته‌ی $\sigma(p)$ که p هر رأس متعلق به A_i است را در نظر بگیرید. ناحیه‌ی اشتراک این مخروط‌ها نیز یک مخروط جدید خواهد بود، که رأس این مخروط را می‌توان به عنوان نقطه‌ی z انتخاب

کرد. شکل ۶.۵ را ببینید. در این شکل، نقطه‌ی z می‌تواند رأس ناحیه‌ی تیره رنگ (ناحیه‌ی اشتراک مخروط‌ها) باشد، ولی برای این که نقاطی مانند u و v روی مرز قرار نگیرند می‌توان این رأس را کمی جابه‌جا کرد و نقطه‌ی جدید را z گرفت. در ادامه بیان خواهد شد که چطور می‌توان با استفاده از روش‌های جستجوی محدوده^۷ یک $CSPD$ به دست آورد، به طوری که هر نقطه‌ی $p \in P$ تنها در $O(\log^d n)$ جفت (A_i, B_i) ظاهر شود.

یادآوری می‌شود که مخروط‌های $\sigma(p_i)$ برای $1 \leq i \leq n$ ، نسخه‌های انتقال یافته‌ای از یک مخروط کانونی $\sigma \in TC$ هستند. در ادامه از ساختار **درخت محدوده**^۸ چندبعدی استفاده خواهد شد، برای اطلاعات بیش‌تر در مورد این ساختار می‌توانید فصل ۵ از کتاب [۲۴] را ببینید. فرض کنید که \mathcal{T}_σ ، یک درخت محدوده‌ی چندبعدی باشد که برای گزارش تمام نقاطی از P که در یک چنین مخروط‌های انتقال یافته‌ای $(\sigma(p_i))$ واقع شده‌اند، ساخته شده باشد. فرض کنید $B(v)$ ، **زیرمجموعه‌ی کانونی**^۹ گره v در سطح d درخت محدوده باشد، که به این شکل تعریف می‌شود: زیرمجموعه‌ای از نقاط P ، که در زیردرخت با ریشه‌ی v ذخیره شده‌اند. آنگاه نقاط داخل یک مخروط پرسش^{۱۰} شده، اجتماع $O(\log^d n)$ زیرمجموعه‌ی کانونی مجزای $B(v)$ خواهد بود. علاوه بر این، هر نقطه‌ی $p \in P$ می‌تواند در $O(\log^d n)$ زیرمجموعه‌ی کانونی $B(v)$ واقع شود.

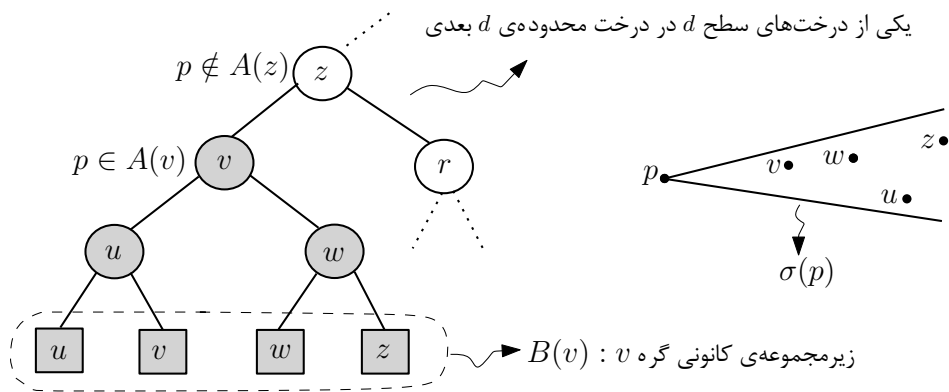
برای ساخت یک تجزیه‌ی Ψ_σ از روی درخت محدوده‌ی \mathcal{T}_σ ، به این طریق عمل می‌شود: برای هر مخروط $\sigma(p_i)$ به طوری که $1 \leq i \leq n$ ، یک پرسش روی درخت \mathcal{T}_σ اجرا می‌شود و همزمان با آن به همراه هر گره v در سطح d درخت \mathcal{T}_σ ، مجموعه‌ی $A(v)$ آن ذخیره می‌شود، که $A(v)$ به این شکل تعریف می‌شود: زیرمجموعه‌ای از نقاط $p_i \in P$ ، که هنگام پرسش با مخروط $\sigma(p_i)$ مجموعه‌ی $B(v)$ به عنوان یکی از زیرمجموعه‌های کانونی‌اش انتخاب شده باشد. به عبارتی دیگر، تمام نقاط ذخیره شده در زیردرخت با ریشه‌ی v ، در داخل مخروط پرسش شده واقع شوند، شکل ۷.۵ را ببینید. در این شکل، نقطه‌ی p متعلق به $A(v)$ است چون هنگام پرسش با $\sigma(p)$ ، $B(v)$ در داخل مخروط قرار گرفته و انتخاب می‌شود. اما p متعلق به $A(z)$ نیست چون نقطه‌ی $r \in B(z)$ در داخل $\sigma(p)$ قرار ندارد و $B(z)$ انتخاب نمی‌شود. با توجه به این مطالب، برای هر گره v در سطح d درخت \mathcal{T}_σ ، یک جفت $(A(v), B(v))$ ذخیره می‌شود که به راحتی در تعریف ۲.۲.۵، بیان شده برای Ψ_σ صدق می‌کند. بنابراین به این طریق جفت‌های (A_i, B_i) برای $CSPD$ به دست می‌آیند. لم زیر به

^۷Range searching techniques

^۸Range tree

^۹Canonical subset

^{۱۰}Query



شکل ۷.۵: تعیین مجموعه $A(v)$ هر گره v

بیان این موضوع می‌پردازد.

لم ۳.۲.۵ ([۱]) برای هر مجموعه‌ی P از n نقطه در \mathbb{R}^d و هر مخروط $\sigma \in TC$ ، یک تجزیه‌ی جفت مخروط-مجزای $(CSPD)$ $\Psi_\sigma = (A_1, B_1), \dots, (A_m, B_m)$ وجود دارد، به طوری که هر نقطه‌ی $p \in P$ تنها در جفت (A_i, B_i) ظاهر می‌شود.

بنابراین با توجه به مطالب بیان شده، می‌توان برای هر مخروط از TC ، یک درخت محدوده ساخت و سپس از روی جفت‌های ذخیره شده در هر گره در سطح آخر این درخت‌ها، $CSPD$ را برای تمام این مخروط‌های TC به دست آورد. حال این سوال باقی می‌ماند که چگونه از $CSPD$ ‌ها می‌توان پوشش مورد نظر را ساخت. در ادامه به این سوال پاسخ داده خواهد شد. برای ساخت پوشش در این مرحله، لازم است تا یال‌های مناسبی به $CSPD$ ‌ها اضافه شوند. بنابراین ابتدا با تعریف زیر، مجموعه‌ی مناسبی از یال‌ها برای ساخت پوشش معرفی می‌شوند و سپس با ارائه‌ی روش مناسبی، یال‌هایی با این تعریف، به ساختار اضافه می‌شوند.

تعریف ۴.۲.۵ ([۱]) فرض کنید $\mathcal{S} = (P, E_{\mathcal{S}})$ یک گراف اقلیدسی روی P و (A_i, B_i) یک جفت در تجزیه‌ی جفت مخروط-مجزای Ψ_σ برای P باشد. جفت (A_i, B_i) در گراف \mathcal{S} خوش اتصال^{۱۱} است اگر برای هر دو نقطه‌ی $p \in A_i$ و $q \in B_i$ یکی از این شرایط برقرار باشد:

(i) یک یال $(p, r) \in E_{\mathcal{S}}$ وجود داشته باشد به طوری که روابط $r \in \sigma(p)$ و $dist_\sigma(p, r) \leq dist_\sigma(p, q)$ برقرار باشند،

^{۱۱}Well connected

(ii) یک یال $(q, r) \in E_S$ وجود داشته باشد به طوری که روابط $r \in \bar{\sigma}(q)$ و $dist_{\bar{\sigma}}(q, r) \leq dist_{\bar{\sigma}}(q, p)$ برقرار باشند.

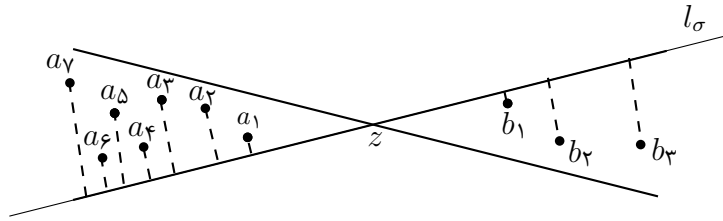
با توجه به این تعریف، در لم زیر بیان خواهد شد که اگر یک گراف اقلیدسی خوش اتصال باشد، آنگاه این گراف یک $\varepsilon + 1$ -پوشش است.

لم ۵.۲.۵ ([۱]) فرض کنید P یک مجموعه از نقاط در \mathbb{R}^d و $\{\Psi_{\sigma} : \sigma \in TC\}$ یک مجموعه از $CSPD$ ها باشد. همچنین فرض کنید $S = (P, E_S)$ یک گراف اقلیدسی روی P باشد، به طوری که هر جفت (A_i, B_i) از هر Ψ_{σ} ، در گراف S خوش اتصال باشد. آنگاه S ، یک $\varepsilon + 1$ -پوشش برای P است.

اثبات. برای اثبات لم، کافی است ثابت شود که بین هر دو نقطه p و q دلخواه در S یک $\varepsilon + 1$ -مسیر وجود دارد. این اثبات با استقرا روی فاصله $\|pq\|$ انجام می‌شود. (با دقت بیشتر، برای هر جفت (p, q) فاصله p و q محاسبه می‌شود، سپس جفت‌ها بر اساس این فاصله‌ها، مرتب می‌شوند و از استقرا روی این ترتیب استفاده می‌شود.)

پایه‌ی استقرا: فرض کنید (p, q) نزدیک‌ترین جفت باشد، همچنین فرض کنید $\sigma \in TC$ مخروطی باشد که $q \in \sigma(p)$ است. در ابتدا این ادعا که: هیچ نقطه‌ی $r \in \sigma(p)$ وجود ندارد به طوری که $dist_{\sigma}(p, r) < dist_{\sigma}(p, q)$ باشد، اثبات می‌شود. بدین منظور، فرض خلف گرفته می‌شود که چنین نقطه‌ای وجود داشته باشد. سپس $\angle qrp > \angle rpq$ برقرار است شکل ۳.۵ قسمت (ب) را ببینید. توجه کنید که زاویه‌ی θ کوچک است. این به رابطه‌ی $\|rq\| < \|pq\|$ اشاره می‌کند، متناقض با نزدیک‌ترین جفت بودن (p, q) و ادعا اثبات می‌شود. در ادامه فرض کنید $(A_i, B_i) \in \Psi_{\sigma}$ باشد، به طوری که $p \in A_i$ و $q \in B_i$ باشد؛ از آنجایی که Ψ_{σ} یک تجزیه‌ی جفت مخروط-مجزاست، یک چنین جفتی وجود دارد. با ادعایی که اثبات شد، مشخص است که p و q در تصویر عمودی $A_i \cup B_i$ روی خط گذرنده از یال نماینده‌ی σ مجاور هستند. از آنجایی که (A_i, B_i) خوش اتصال است، بنابراین طبق تعریف ۴.۲.۵، یال (p, q) یک یال متعلق به E_S است. در نتیجه، پایه‌ی استقرا اثبات می‌شود.

حکم استقرا: حالا دو نقطه‌ی دلخواه $p, q \in P$ را در نظر بگیرید. فرض کنید $\sigma \in TC$ مخروطی است که $q \in \sigma(p)$ باشد. همچنین فرض کنید $(A_i, B_i) \in \Psi_{\sigma}$ ، جفتی است که $p \in A_i$ و $q \in B_i$ است. از آنجایی که (A_i, B_i) خوش اتصال است. بنابراین طبق تعریف ۴.۲.۵، باید برای دو نقطه‌ی p و q شرط (i) یا شرط (ii) برقرار باشد، می‌توان بدون از دست دادن کلیت مسئله فرض کرد که شرط (i) برقرار است

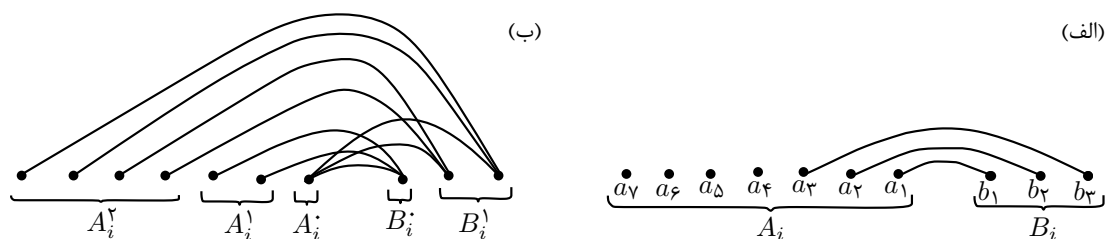


شکل ۸.۵: A_i و B_i توسط مخروط‌های $\sigma(z)$ و $\bar{\sigma}(z)$ از هم مجزا می‌شوند.

(حالت دیگر یعنی زمانی که شرط (ii) برقرار است، مشابه با همین حالت اثبات می‌شود)، بنابراین یک نقطه‌ی $r \in P \cap \sigma(p)$ وجود دارد به طوری که $dist_\sigma(p, r) \leq dist_\sigma(p, q)$ و $(p, r) \in E_S$ است. سپس می‌توان به کمک اثبات لم ۱.۲.۵ دو حالت ممکن را در نظر گرفت و در هر دو حالت رابطه‌ی $\|rq\| > \|pq\|$ را به راحتی به دست آورد (اثبات لم ۱.۲.۵ را ببینید): در حالت (الف) اثبات نتیجه می‌شود که یال (p, q) بزرگ‌ترین یال مثلث rpq است و بنابراین $\|pq\| > \|rq\|$ برقرار است. در حالت (ب) اثبات نیز نتیجه می‌شود که $\|rq\|/\|pq\| = \sin \angle qpr / \sin \angle prq \leq \sin \theta / \cos \theta < 1$ که این به معنی $\|pq\| > \|rq\|$ است. در نتیجه رابطه‌ی $\|pq\| > \|rq\|$ برقرار است و با استفاده از فرض استقرا، یک ε -مسیر در S بین q و r وجود دارد و با استفاده از لم ۱.۲.۵، با اضافه کردن یال (p, r) به این مسیر، یک ε -مسیر بین p و q نیز در S حاصل می‌شود. \square

تنها این مطلب از ساخت پوشش ایستا باقی می‌ماند که چگونه یک گراف با تعداد کم یال به دست آورد، به طوری که در آن جفت‌های $(A_i, B_i) \in \Psi_\sigma$ خوش اتصال باشند. در ادامه توضیح داده می‌شود که چگونه این کار امکان‌پذیر است.

همان‌طور که قبلاً بیان شد، با توجه به شرط دوم تعریف ۲.۲.۵ برای تجزیه‌ی جفت‌های مخروط-مجزا، باید برای هر جفت (A_i, B_i) ، نقطه‌ای مانند $z \in \mathbb{R}^d$ باشد، به طوری که روابط $A_i \subset \bar{\sigma}(z)$ و $B_i \subset \sigma(z)$ برقرار باشند. فرض کنید l_σ یک خط گذرنده از z و موازی با یال نماینده‌ی مخروط σ باشد. شکل ۸.۵ را ببینید. تمام نقاط $A_i \cup B_i$ را روی l_σ تصویر کنید. سپس تصویر نقاط متعلق به A_i ، در یک طرف z و تصویر نقاط متعلق به B_i در طرف دیگر z واقع می‌شوند. نقاط A_i با a_1, \dots, a_k و نقاط B_i با b_1, \dots, b_l هر دو به ترتیب افزایش فاصله تا z برچسب‌گذاری می‌شوند. یک راه ساده برای این‌که مطمئن شد (A_i, B_i) خوش اتصال خواهد شد این است که: یال (a_i, b_i) برای هر $1 \leq i \leq \min(k, l)$ به گراف اضافه شود، چنان‌چه در شکل



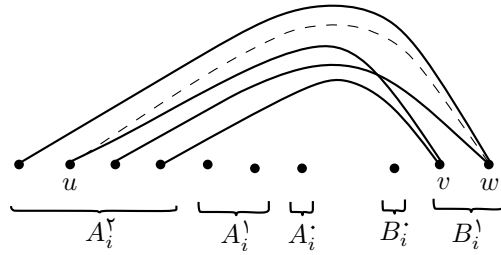
شکل ۹.۵: (الف): ساده‌ترین راه برای خوش اتصال کردن، شکل (ب): تقسیم‌بندی نقاط به گروه‌های لگاریتمی برای خوش اتصال کردن

۹.۵ قسمت (الف) نمایش داده شده است. متاسفانه، نگهداری چنین مجموعه‌ای از یال‌ها زمانی که نقاط حرکت می‌کنند، هزینه‌بر است؛ زیرا زمانی که یک نقطه‌ی جدید وارد یک مخروط می‌شود، ممکن است تعداد زیادی از یال‌ها نیاز به تغییر داشته باشند. برای نمونه، در نظر بگیرید که A_i و B_i هر کدام شامل $O(n)$ نقطه باشند و بر اثر حرکت نقاط، نقطه‌ای به A_i وارد شده و برچسب آن، هنگام مرتب‌سازی جدید نقاط، a_1 باشد. در این صورت $O(n)$ یال نیاز به به‌روزرسانی دارند و کران پایین برای به‌روزرسانی نقاط در این وضعیت $\Omega(n)$ خواهد بود (پاسخ‌گو نبودن ساختار). بنابراین، در ادامه روش جدیدی برای خوش اتصال کردن جفت (A_i, B_i) مطرح خواهد شد، که این مشکل را ندارد. مجموعه‌ی A_i به تعداد لگاریتمی گروه A_i^h, \dots, A_i^o افزاز می‌شود، به طوری که $A_i^j := \{a_{2^j}, \dots, a_{2^{j+1}-1}\}$ باشد. شکل ۹.۵ قسمت (ب) را ببینید. گفته می‌شود که یک گروه A_i^j ، یک گروه پر^{۱۲} است، اگر شامل دقیقاً 2^j نقطه باشد. توجه کنید که تمام گروه‌ها پر هستند، به جز گروه آخر A_i^h ، که امکان دارد پر نباشد. به طور مشابه، B_i نیز به گروه‌های B_i^j افزاز می‌شود. در ادامه مجموعه‌ی $E(A_i, B_i)$ از یال‌های اتصال دهنده‌ی نقاط در A_i به نقاط در B_i ، تعریف می‌شود.

۱. برای هر گروه A_i^j ، با $j > 0$ ، یک مجموعه از یال‌ها مطابق این روش درج می‌شود: فرض کنید $k = |B_i^{j-1}|$ باشد. $2k$ نقطه‌ی ابتدایی A_i^j ، یعنی نقاط $a_{2^j}, \dots, a_{2^j+2k-1}$ را در نظر بگیرید. هر کدام از این $2k$ نقطه به یکی از نقاط B_i^{j-1} به گونه‌ای وصل می‌شود که هر نقطه از B_i^{j-1} تنها دو اتصال (یال) را دریافت کند.

۲. برای هر گروه B_i^j ، با $j > 0$ ، یک مجموعه از یال‌ها مطابق این روش درج می‌شود: هر کدام از $2|A_i^{j-1}|$ نقطه‌ی ابتدایی در B_i^j به یک نقطه در A_i^{j-1} به گونه‌ای وصل می‌شود که هر نقطه از A_i^{j-1} تنها دو

^{۱۲}Full



شکل ۱۰.۵: امکان رسم یال (u, w) به جای یال (u, v)

اتصال (یال) را دریافت کند.

۳. تنها نقطه‌ی A_i^j به تنها نقطه‌ی B_i^j وصل می‌شود.

هرچند که برای خوش اتصال کردن یک جفت (A_i, B_i) بسیار اهمیت دارد که نقاط متعلق به A_i و B_i ، هر کدام بر اساس فاصله‌ی $dist_\sigma$ مرتب شوند و سپس به گروه‌های لگاریتمی افزاز گردند، اما نکته‌ی قابل توجه در این روش این است که، هنگام درج یک یال از گروه A_i^j ، به گروه B_i^{j-1} (یا برعکس از گروه B_i^j ، به گروه A_i^{j-1})، این که این یال کدام دو نقطه از این گروه‌ها را بهم متصل می‌کند، اهمیتی ندارد. برای نمونه در شکل ۱۰.۵، فرقی ندارد که نقطه‌ی u متعلق به گروه A_i^j ، به نقطه‌ی v یا نقطه‌ی w از گروه B_i^j وصل شود. این موضوع برای نگهداری خاصیت خوش اتصال بودن، زمانی که نقاط حرکت می‌کنند، بسیار اهمیت دارد؛ زیرا هنگام درج یک نقطه در یک گروه، در صورت پر بودن آن گروه می‌توان به راحتی، یال‌های متصل به آخرین نقطه‌ی این گروه را، یعنی p ، به نقطه‌ی تازه درج شده در این گروه (که بر اساس ترتیب $dist_\sigma$ در جای مناسب در گروه درج می‌شود) داد و نقطه‌ی p را به گروه بعدی منتقل کرد.

لم ۶.۲.۵ ([۱]) مجموعه یال‌های $E(A_i, B_i)$ ، هر نقطه در A_i را به حداکثر سه نقطه در B_i متصل می‌کنند و برعکس. علاوه بر این جفت (A_i, B_i) با یال‌های $E(A_i, B_i)$ خوش اتصال است.

اثبات. به طور مستقیم از روش ساخت مجموعه یال‌ها، مشخص است که هر نقطه به حداکثر سه نقطه‌ی دیگری متصل می‌شود و ادعای اول لم اثبات شده است. برای این که اثبات شود (A_i, B_i) خوش اتصال است، یک جفت نقطه‌ی p و q به طوری که $p \in A_i^j$ و $q \in B_i^{j'}$ را در نظر بگیرید. در ادامه نشان داده خواهد شد که یا شرط (i): یک یال $(p, r) \in E(A_i, B_i)$ وجود دارد به طوری که $r \in \sigma(p)$ و $dist_\sigma(p, r) \leq dist_\sigma(p, q)$ یا شرط (ii): یک یال $(q, r) \in E(A_i, B_i)$ وجود دارد به طوری که $r \in \bar{\sigma}(q)$ و $dist_\sigma(q, r) \leq dist_\sigma(q, p)$.

برقرار است. بدون از دست دادن کلیت مسئله فرض می‌شود که $j \leq j'$ است. اگر $j = j' = 0$ باشد، آنگاه اثبات انجام شده است؛ زیرا یک یال که تنها نقطه‌ی A_i° (که p است) را به تنها نقطه‌ی B_i° (که q است) وصل می‌کند، وجود دارد. اگر $j > 0$ باشد، آنگاه B_i^{j-1} باید وجود داشته باشد و پر نیز باشد. بنابراین $E(A_i, B_i)$ ، یک یال (p, r) برای $p \in B_i^{j-1}$ و $r \in B_i^{j'}$ را دربر می‌گیرد. از آنجایی که $q \in B_i^{j'}$ و $j' \geq j > j - 1$ است، در نتیجه $dist_\sigma(p, r) \leq dist_\sigma(p, q)$ است و اثبات انجام می‌شود. \square

با به کار بردن این رویه برای هر جفت (A_i, B_i) در هر Ψ_σ ، یک گراف $\mathcal{S} = (P, E_S)$ ، که یک $\varepsilon + 1$ -پوشش است، به دست خواهد آمد. از آنجایی که در هر Ψ_σ ، هر نقطه در حداکثر $O(\log^d n)$ جفت (A_i, B_i) حضور دارد و هنگام خوش اتصال کردن گراف \mathcal{S} ، به هر نقطه در هر جفت حداکثر سه یال متصل می‌شود، بنابراین به هر نقطه در هر Ψ_σ حداکثر $O(\log^d n)$ یال متصل می‌شود. در نتیجه، چون n نقطه و $O((1/\varepsilon^{d-1}))$ تجزیه‌ی جفت مخروط مجزای Ψ_σ وجود دارد، تعداد یال‌ها در \mathcal{S} (اندازه‌ی پوشش) از مرتبه‌ی $O((n/\varepsilon^{d-1}) \log^d n)$ خواهد شد، هرچند که در ادامه نشان داده خواهد شد که چگونه این را به مرتبه‌ی خطی کاهش داد.

۳.۲.۵ یک پوشش با اندازه‌ی خطی

در این زیربخش نشان داده خواهد شد که چطور یک پوشش $\mathcal{S}^* = (P, E_{\mathcal{S}^*})$ با یک تعداد خطی یال، با هرس^{۱۳} کردن تعدادی از یال‌های $\mathcal{S} = (P, E_S)$ به دست خواهد آمد. یک یال $(p, q) \in E_S$ و یک مخروط $\sigma \in \mathcal{TC}$ را در نظر بگیرید به طوری که $q \in \sigma(p)$ و $p \in \bar{\sigma}(q)$ باشد. هدف ایجاد $E_{\mathcal{S}^*}$ از E_S است. بدین منظور، یال $(p, q) \in E_S$ به $E_{\mathcal{S}^*}$ اضافه می‌شود اگر و تنها اگر یکی از شرط‌های (i) یا (ii) برقرار باشد:

(i) از بین تمام نقاط $r \in \sigma(p)$ که (p, r) یک یال متعلق به E_S است، q نقطه‌ای است که از همه به p نزدیک‌تر باشد،

(ii) از بین تمام نقاط $r \in \bar{\sigma}(q)$ که (q, r) یک یال متعلق به E_S است، p نقطه‌ای است که از همه به q نزدیک‌تر باشد.

به راحتی دیده می‌شود که هر جفت مخروط-مجزای (A_i, B_i) در \mathcal{S}^* نیز هنوز خوش اتصال است. علاوه بر آن گراف \mathcal{S}^* ، تعداد $O(n/\varepsilon^{d-1})$ یال دارد، زیرا هر یال می‌تواند به یک ترکیب منحصر به فرد، از یک نقطه در P و یک مخروط در \mathcal{TC} شارژ شود.

^{۱۳}Pruning

لم ۷.۲.۵ [۱۱] در گراف $\mathcal{S}^* = (P, E_{\mathcal{S}^*})$ ، هر جفت مخروط-مجزا خوش اتصال است. علاوه بر آن \mathcal{S}^* ، تعداد $O(n/\varepsilon^{d-1})$ یال دارد و هر نقطه از درجه $O(\log^d n)$ است.

۳.۵ نگهداری پوشش تحت حرکت نقاط

به طور خلاصه، پوشش ایستای معرفی شده به این شکل ساخته شد: ابتدا یک تجزیه جفت‌های مخروط-مجزای Ψ_σ برای هر مخروط $\sigma \in TC$ با استفاده از یک درخت محدوده‌ی \mathcal{T}_σ محاسبه شد. سپس یک گراف $\mathcal{S} = (P, E_{\mathcal{S}})$ با خوش اتصال کردن هر جفت مخروط-مجزای (A_i, B_i) در \mathcal{S} تولید شد و سرانجام یک پوشش $\mathcal{S}^* = (P, E_{\mathcal{S}^*})$ با اندازه‌ی خطی، به کمک هرس کردن تعدادی یال از \mathcal{S} به دست آمد. یعنی، برای هر مخروط σ و هر نقطه‌ی p ، تنها یک یال $(p, q) \in E_{\mathcal{S}}$ به مجموعه یال‌های $E_{\mathcal{S}^*}$ اضافه می‌شد، که دارای این ویژگی بود: q از بین تمام نقاط داخل $\sigma(p)$ ، که در \mathcal{S} به p متصل هستند، از همه به p نزدیک‌تر است.

برای وابسته به حرکت کردن این پوشش، کارهای زیر باید انجام شوند:

۱. $CSPD$ ‌های Ψ_σ زمانی که نقاط حرکت می‌کنند، نگهداری شوند.

۲. برای هر مخروط σ و هر نقطه‌ی p ، نزدیک‌ترین نقطه به p از بین تمام نقاط داخل $\sigma(p)$ که به p در \mathcal{S}

متصل می‌شوند، نگهداری شود.

۱.۳.۵ درخت‌های محدوده بر اساس رتبه

یک مخروط $\sigma \in TC$ را در نظر بگیرید. از آنجایی که ساخت Ψ_σ بر اساس درخت محدوده‌ی \mathcal{T}_σ است. بنابراین برای نگهداری Ψ_σ زمانی که نقاط حرکت می‌کنند، نیاز است تا ساختار \mathcal{T}_σ وابسته به حرکت شود. باش^{۱۴} و همکارانش در [۸] توصیف می‌کنند که چطور یک درخت محدوده در تنظیمات وابسته به حرکت نگهداری شود، به این صورت که: هر زمان ترتیب دو نقطه در یکی از مختصات با هم عوض می‌شود، آن نقاط از ساختار حذف می‌شوند و دوباره با مقادیر جدیدشان درج می‌گردند. در نتیجه، به یک درخت محدوده‌ی پویا نیاز است. متاسفانه درخت‌های محدوده‌ی پویای موجود که از روش ساخت مجدد سراسری یا محلی [۳۶]، یا روش ویلارد^{۱۵} و لوکر^{۱۶}

^{۱۴}Basch

^{۱۵}Willard

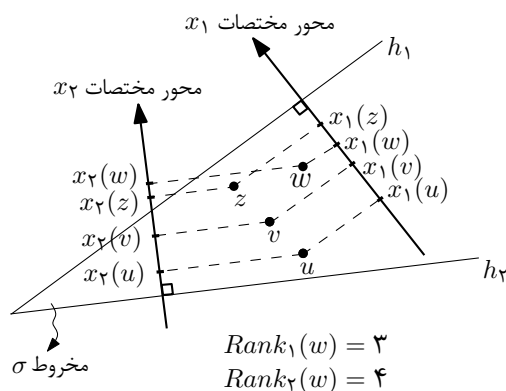
^{۱۶}Lueker

[۳۹] استفاده می‌کنند، برای نگهداری وابسته به حرکت پوشش معرفی شده مناسب نیستند؛ زیرا این روش‌ها یا کران‌های سرشکن شده را به دست می‌آورند یا به عملیات جداسازی و ادغام نیاز دارند (که برای $CSPD$ ‌های معرفی شده سخت است). بنابراین، دیدگاه متفاوتی که بر اساس رتبه‌ی نقاط می‌باشد و برای طراحی BSP ‌های وابسته به حرکت در [۲۵] و درخت‌های kd وابسته به حرکت در [۳] استفاده شده است، در این‌جا نیز معرفی و استفاده خواهد شد.

همان‌طور که بیان شد، ساختارهای پویا به دلیل عملیاتی که برای به‌روزرسانی خود نیاز دارند، از جمله عملیات متوازن سازی، ادغام، جداسازی و غیره برای نگهداری درخت محدوده مناسب نیستند. برای صرف نظر کردن از این عملیات به یک ساختار ایستا نیاز است تا بدنه‌ی ساختار از شروع تا پایان حرکت نقاط ثابت باقی بماند و تنها تغییر مقدار بدهد. بدین منظور ایده‌ی اصلی، تعریف یک درخت اسکلت^{۱۷} است که وابسته به موقعیت نقاط نیست. از آن‌جایی که ساختار اسکلت به موقعیت نقاط وابسته نیست، ایستاست و برای نگهداری اسکلت زمانی که نقاط حرکت می‌کنند، به هیچ عمل متوازن‌سازی نیاز نمی‌شود؛ زیرا تمام گره‌ها و برگ‌ها ثابت هستند و تنها مقادیر مرتبط به آن‌ها می‌تواند تغییر کند یا حتی تهی شود. (یک اخطار: برای صرفه‌جویی در مصرف حافظه، قسمت‌های معینی از اسکلت هرس می‌شوند.) این که هر نقطه در کدام زیرمجموعه‌ی کانونی درخت اسکلت ذخیره شود، به رتبه‌ی مختصات آن نقطه بستگی خواهد داشت. بنابراین، رویدادها تنها زمانی رخ می‌دهند که ترتیب دو نقطه‌ی p و q در طول یکی از محورهای مختصات عوض شود. زمانی که این اتفاق می‌افتد، رتبه‌ی p و q در آن مختصات باهم جابه‌جا می‌شود (رتبه‌های نقاط دیگر تحت تاثیر قرار نمی‌گیرد). در این وضعیت، برای به‌روزرسانی درخت، تنها نیاز می‌شود که p و q حذف شوند و با رتبه‌ی جدیدشان دوباره درج گردند. این تغییرات تنها $O(\log^d n)$ زیرمجموعه‌ی کانونی را تغییر می‌دهد. سپس باید یال‌های پوشش تعریف شده برای این زیرمجموعه‌های کانونی به‌روزرسانی شوند. در ادامه این ایده با جزئیات بیشتری بیان خواهد شد.

فرض کنید h_1, \dots, h_d صفحاتی باشند که مخروط σ را تعریف می‌کنند. برای هر h_i ، محور مختصات x_i به‌عنوان خط عمود بر h_i و $x_i(p)$ به‌عنوان مولفه‌ی x_i نقطه‌ی p تعریف می‌شود. شکل ۱۱.۵ را ببینید. برای هر نقطه‌ی $p \in P$ ، رتبه‌ی $x_i(p)$ در مجموعه‌ی مرتب شده‌ی $\{x_i(q) : q \in P\}$ (یعنی چندمین عضو این مجموعه است)، به‌عنوان $Rank_i(p)$ تعریف می‌شود. شکل ۱۱.۵ را ببینید. برای ساده‌تر شدن بحث، برای هر دو نقطه‌ی p و q فرض می‌شود که $x_i(p) \neq x_i(q)$ برای تمام i ‌ها برقرار است. (البته مختصات دو نقطه می‌تواند به‌طور موقت، زمانی که دو نقطه ترتیب‌شان را باهم عوض می‌کنند، مساوی شود. اما منظور مساوی

^{۱۷}Skeleton tree



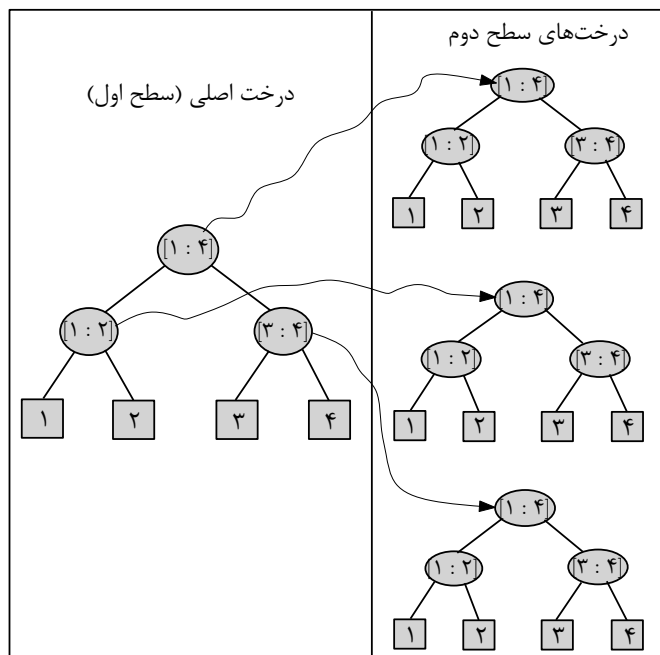
شکل ۱۱.۵: رتبه‌ی نقطه‌ی w در هر مختصات

نبودن مولفه‌ی دو نقطه در بازه‌های زمانی بین چنین رویدادهایی است.)

اسکلت یک درخت محدوده بر اساس رتبه^{۱۸}، که با نماد T_σ^{skel} نشان داده خواهد شد، به این شکل تعریف می‌شود: فرض کنید $n = |P|$ باشد. (در تنظیمات وابسته به حرکت، n ثابت است.) اسکلت یک درخت محدوده بر اساس رتبه‌ی یک بعدی، یک درخت دودویی متوازن روی n برگ است، که برگ‌ها متناظر با رتبه‌های ۱ تا n از چپ به راست هستند. هر گره داخلی v متناظر با یک محدوده از رتبه‌ها است (رتبه‌های تمام برگ‌های واقع در زیردرخت با ریشه‌ی v). اسکلت یک درخت محدوده بر اساس رتبه‌ی d بعدی به این شکل است: یک درخت اصلی، که یک درخت دودویی متوازن روی n برگ است و هر گره داخلی v یک ساختار T_v مرتبط با خودش دارد که این ساختار، اسکلت یک درخت محدوده بر اساس رتبه‌ی $d - 1$ بعدی است. در شکل ۱۲.۵ می‌توانید اسکلت یک درخت محدوده بر اساس رتبه‌ی دو بعدی برای ۴ نقطه را ببینید. توجه کنید که هر درخت در هر سطح، یک درخت دودویی متوازن روی n رتبه از ۱ تا n است، مستقل از این‌که چند نقطه از P در درخت واقع شده است. بنابراین اندازه‌ی اسکلت یک درخت محدوده بر اساس رتبه‌ی d بعدی $\Theta(n^d)$ است. درحالی‌که درخت محدوده بر اساس رتبه‌ی d بعدی که در ادامه استفاده خواهد شد، تنها از $O(n \log^d n)$ فضا استفاده می‌کند؛ زیرا چنان‌چه بعداً توضیح داده می‌شود، اسکلت این ساختار هرس خواهد شد. توجه کنید که قرار نیست اسکلت نگهداری شود یا به‌طور صریح ساخته شود، بلکه تنها از آن استفاده می‌شود تا به کمک آن بتوان درخت محدوده بر اساس رتبه را، از یک راه ساده‌تر تعریف کرد.

در ادامه توضیح داده می‌شود که نقاط P چگونه در T_σ^{skel} ذخیره می‌شوند. همچنین چگونه مجموعه‌های

^{۱۸}Rank-based range tree

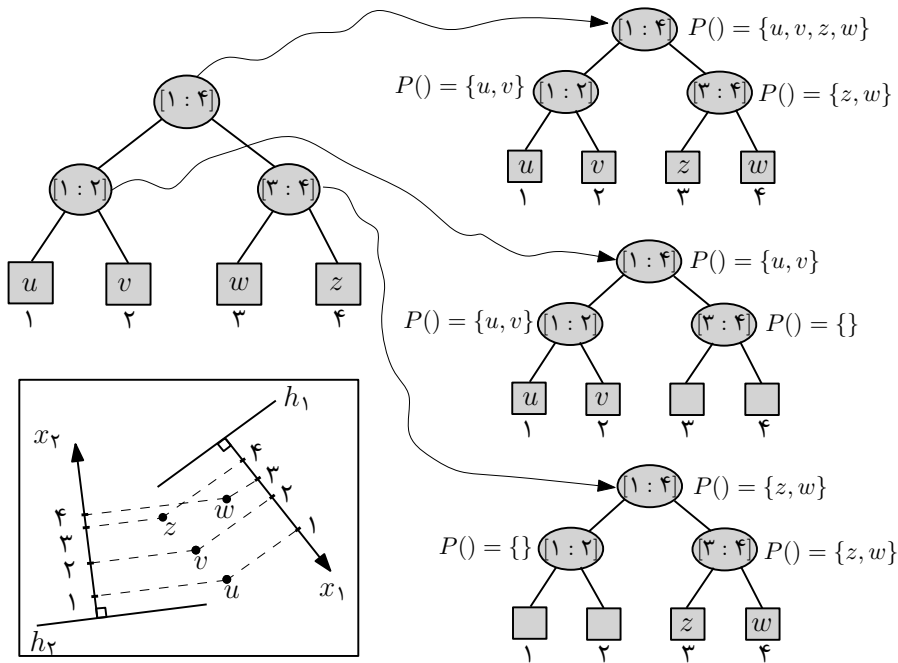


شکل ۱۲.۵: اسکلت یک درخت محدوده بر اساس رتبه‌ی دو بعدی برای ۴ نقطه

$P(v)$ و $R(v)$ به دست می‌آیند و جفت‌های $(P(v), R(v))$ تجزیه‌ی جفت‌های مخروط-مجزای Ψ_σ را ایجاد می‌کنند.

نقاط P در T_σ^{skel} از راه معمول درج می‌شوند. بنابراین برای درج یک نقطه‌ی $p \in P$ در درخت اصلی T_σ^{skel} ، مسیر جستجو تا برگ متناظر با $Rank_1(p)$ طی می‌شود و برای هر گره v روی این مسیر جستجو، به صورت بازگشتی p در درخت‌های مرتبط در سطوح بعدی، یعنی T_v ها درج می‌شود (با استفاده از رتبه‌های $(Rank_2(p), \dots, Rank_d(p))$). برای یک گره v در هر سطح از درخت، $P(v)$ به زیرمجموعه‌ی کانونی v ، یعنی مجموعه نقاطی که مسیر جستجوی‌شان از گره v می‌گذرد، اشاره می‌کند. مجموعه‌های $P(v)$ به صورت صریح برای گره‌های درخت‌های سطح d ذخیره می‌شوند (یعنی درخت‌هایی که برای d امین مختصات تعریف می‌شوند). می‌توانید این مطالب را برای چهار نقطه در صفحه، در شکل ۱۳.۵ ببینید. هر نقطه‌ی $p \in P$ در $O(\log^d n)$ زیرمجموعه‌ی کانونی در سطح d ، که در $O(\log^{d-1} n)$ درخت سطح d توزیع شده‌اند، ذخیره می‌شود.

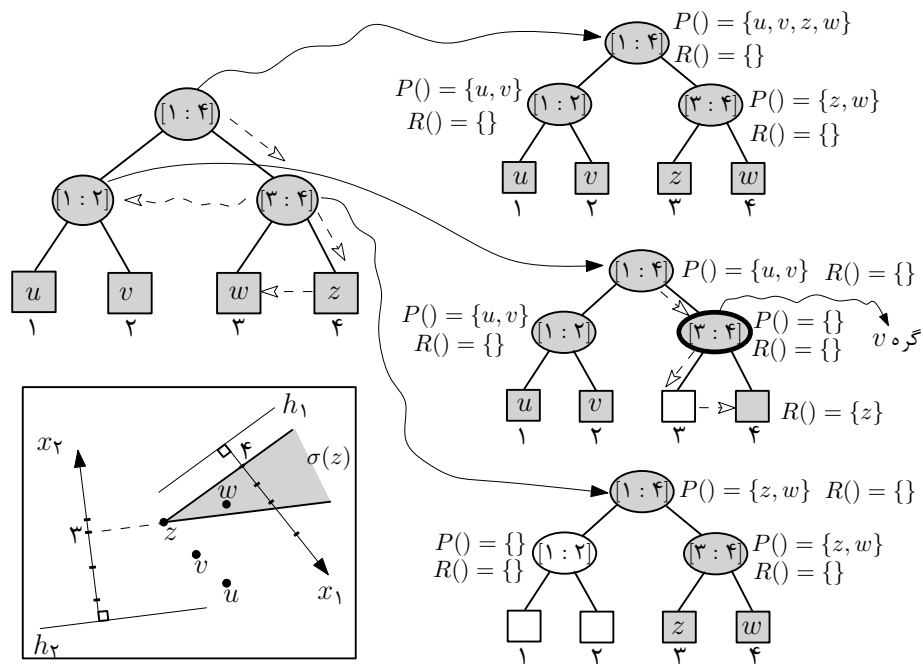
بیان شد که چگونه مجموعه‌های $P(v)$ به دست می‌آیند. بنابراین برای به دست آوردن جفت‌های $(P(v), R(v))$ در تجزیه‌ی جفت‌های مخروط-مجزای Ψ_σ ، لازم است در این مرحله توضیح داده شود که مجموعه‌های $R(v)$ چگونه به دست می‌آیند. بدین منظور، برای هر نقطه‌ی $p \in P$ باید یک پرسش از محدوده‌ی $\sigma(p)$ در اسکلت T_σ^{skel} انجام شود. با توجه به این که رتبه‌ی مرزهای $\sigma(p)$ ، یعنی h_1, \dots, h_d ، همان رتبه‌های نقطه‌ی p ، یعنی



شکل ۱۳.۵: ذخیره‌ی مجموعه‌های $P(v)$ برای گره‌های سطح آخر اسکلت درخت محدوده بر اساس رتبه

$Rank_1(p), \dots, Rank_d(p)$ است، پرسش از محدوده‌ی $\sigma(p)$ ، همان جستجوی نقطه‌ی p است، با این تفاوت که در هر سطح i این روند باید انجام شود: اگر داخل مخروط $\sigma(p)$ ، رتبه‌های کم‌تر از رتبه‌ی $Rank_i(p)$ را شامل می‌شود، هنگام جستجو در درخت‌های این سطح اگر مسیر جستجو به فرزند راست یک گره رفت، آنگاه درخت مرتبط فرزند چپ در سطح بعدی با رتبه‌ی $Rank_{i+1}(p)$ مورد جستجو قرار می‌گیرد. اما اگر مسیر جستجو به فرزند چپ یک گره رفت، لازم نیست کاری انجام شود و جستجو به روال معمول ادامه می‌یابد. اگر داخل مخروط $\sigma(p)$ ، رتبه‌های بیش‌تر از رتبه‌ی $Rank_i(p)$ را شامل شود، اگر مسیر جستجو به فرزند چپ یک گره رفت، آنگاه درخت مرتبط فرزند راست همان گره در سطح بعدی مورد جستجو قرار می‌گیرد. در غیر این صورت مشابه با حالت قبلی عمل می‌شود. شکل ۱۴.۵ را ببینید.

بنابراین، با این روش، $O(\log n)$ گره درخت اصلی که محدوده‌هایشان با هم، محدوده‌ی x_1 از $\sigma(p)$ را می‌پوشاند، انتخاب می‌شوند (با دقت بیش‌تر محدوده‌ی $Rank_1(p)$) و به صورت بازگشتی جستجو در درخت‌های مرتبط با این گره‌ها، از روش بیان شده، ادامه پیدا می‌کند. در نتیجه، همه‌ی زیرمجموعه‌های کانونی $P(v)$ گره‌های انتخاب شده در درخت‌های سطح d ، دقیقاً همان نقاطی که در داخل محدوده‌ی $\sigma(p)$ هستند را شامل خواهند شد. برای یک گره v در هر سطح، $R(v)$ زیرمجموعه‌ای از نقاط $p_i \in P$ است که هنگام پرسش از



شکل ۱۴.۵: ذخیره‌ی مجموعه‌های $R(v)$ برای گره‌های سطح آخر اسکلت درخت محدوده بر اساس رتبه

محدوده‌ی آن‌ها، یعنی پرسش از $\sigma(p_i)$ ، آن گره را انتخاب کرده‌اند. مجموعه‌های $R(v)$ برای گره‌های سطح d به صورت صریح با گره‌هایشان ذخیره می‌شوند. شکل ۱۴.۵ را ببینید. برای نمونه، در این شکل برای جستجوی $\sigma(z)$ باید جستجو با رتبه‌های مرزهای این مخروط یعنی ۴ در سطح اول و ۳ در سطح دوم درخت صورت گیرد. مسیر این جستجو با خط‌چین در شکل مشخص شده است. البته در این شکل، مجموعه‌های $R(v)$ برای برگ‌ها، به جز یک برگ (فرزند راست گره v) تهی بوده است و برای شلوغ نشدن شکل نشان داده نشده است. چنان‌چه قبلاً توضیح داده شد، جفت‌های $(P(v), R(v))$ برای گره‌های سطح d ، جفت‌های تجزیه‌ی جفت مخروط-مجزای Ψ_σ را ایجاد می‌کنند. اندازه‌ی کل تمام مجموعه‌های $P(v)$ و $R(v)$ ، $O(n \log^d n)$ است. هرچند که اندازه‌ی اسکلت هنوز $\Theta(n^d)$ است. در ادامه نشان داده خواهد شد که چطور با هرس کردن T_σ^{skel} ، اندازه‌اش به $O(n \log^d n)$ کاهش یابد.

یک گره v در هر سطح از T_σ^{skel} را یک گره فعال^{۱۹} گویند اگر یکی از شرایط زیر را داشته باشد:

۱. مجموعه‌ی $P(v)$ غیرتهی باشد. (توجه کنید که $P(v)$ هنگامی غیرتهی است که یک گره v' نسل گره

v وجود داشته باشد که $P(v')$ غیرتهی باشد).

^{۱۹}Active

۲. مجموعه‌ی $R(v)$ غیرتهی باشد، یا یک گره v' ، نسل گره v وجود داشته باشد، به طوری که $R(v')$ غیرتهی باشد.

گره‌ای که فعال نیست، گره غیرفعال^{۲۰} نامیده می‌شود. در شکل ۱۴.۵ گره‌های غیرفعال با رنگ سفید مشخص شده‌اند و دارای $P(v)$ و $R(v)$ تهی است. تاکید می‌شود که گره v که در شکل ۱۴.۵ پررنگ شده است با وجود این که دارای $P(v)$ و $R(v)$ تهی است، فعال است؛ زیرا طبق مورد ۲ در تعریف گره‌های فعال، فرزندش v' (فرزند راستش) دارای $R(v')$ غیرتهی است. با هرس کردن تمام گره‌های غیرفعال $T_\sigma^{skel}(P)$ ، درخت محدوده بر اساس رتبه‌ی T_σ^{rb} (ساختاری که مورد نیاز بود) به دست خواهد آمد. ساختن درخت محدوده بر اساس رتبه در زمان $O(n \log^d n)$ چندان سخت نیست، به این صورت که لازم نیست از همان ابتدا کل اسکلت را ساخت، بلکه تنها با ایجاد گره‌های فعال، زمانی که نقاط و محدوده‌ها درج می‌شوند، می‌توان ساختن را انجام داد که منجر به لم زیر می‌شود.

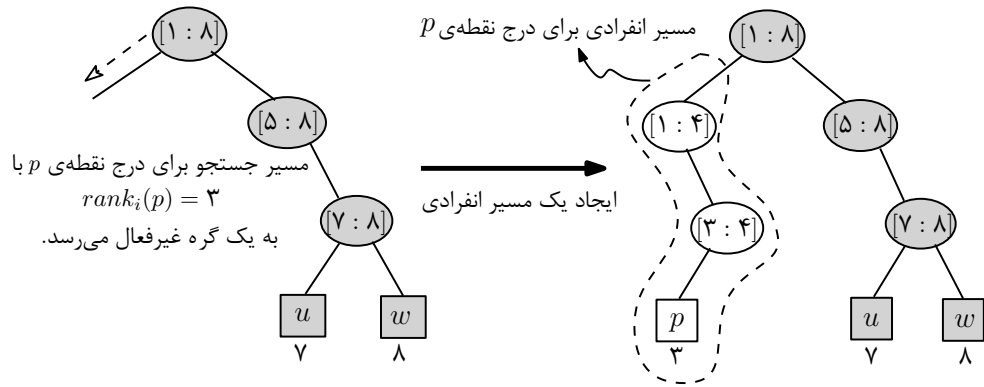
لم ۱۳.۵ ([۱]) برای هر مجموعه‌ی P از n نقطه در \mathbb{R}^d و هر مخروط کانونی $\sigma \in TC$ ، درخت محدوده بر اساس رتبه‌ی T_σ^{rb} از $O(n \log^d n)$ حافظه استفاده می‌کند و می‌تواند در زمان $O(n \log^d n)$ ساخته شود. علاوه بر این، اندازه‌ی تجزیه‌ی جفت مخروط-مجزای Ψ_σ که بر اساس T_σ^{rb} به دست می‌آید، $O(n \log^d n)$ است و هر نقطه‌ی $p \in P$ در $O(\log^d n)$ جفت از Ψ_σ ظاهر می‌شود.

۲.۳.۵ نگهداری وابسته به حرکت T_σ^{rb} و گراف S

درخت محدوده بر اساس رتبه‌ی T_σ^{rb} و این که چگونه تجزیه‌ی جفت مخروط-مجزای Ψ_σ از آن به دست می‌آید، توصیف شد. این سوال باقی می‌ماند که چگونه T_σ^{rb} و Ψ_σ (و نیز گراف $S = (P, E_S)$ ساخته شده روی Ψ_σ) هنگامی که نقاط حرکت می‌کنند، نگهداری شوند.

ساختار ترکیبی T_σ^{rb} ، تنها به رتبه‌های نقاط روی محورهای x_1, \dots, x_d وابسته است و تا زمانی که ترتیب نقاط روی تمام محورها بدون تغییر باقی بماند، این ساختار نیز تغییر نخواهد کرد و نیازی به به‌روزرسانی ندارد. بنابراین کافی است برای هر محور x_i ، نقاط P در یک آرایه‌ی $A_i[1..n]$ که مرتب شده بر اساس ترتیب x_i است، نگهداری شوند. بنابراین محل $A_i[j]$ ، نقطه‌ی p به طوری که $Rank_i(p) = j$ را شامل خواهد شد. فرض کنید که دو نقطه‌ی p و q ترتیب x_i ‌شان با هم عوض شود، برای پردازش یک چنین رویدادی، p و q از ساختار

^{۲۰}Inactive



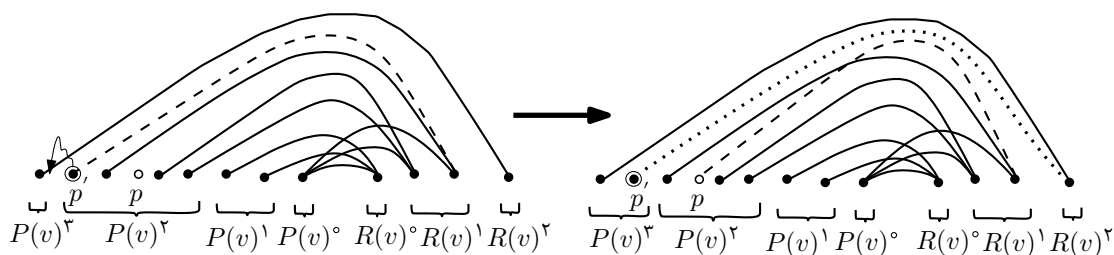
شکل ۱۵.۵: ایجاد یک مسیر انفرادی تا برگ

حذف می‌شوند و دوباره با رتبه‌های جدیدشان در ساختار درج می‌گردند. توجه کنید که یک تغییر رتبه برای p ، یک تغییر در $\sigma(p)$ را نیز به دنبال خواهد داشت. با دقت بیشتر، اگر رتبه‌ی $Rank_i(p)$ تغییر کند (رتبه مولفه‌ی i ام نقطه‌ی p)، این بدان معنی است که رتبه‌ی h_i که یکی از صفحات محدودکننده‌ی $\sigma(p)$ است، تغییر کرده است. همین موضوع برای q نیز درست است. بنابراین نیاز خواهد بود تا $\sigma(p)$ و $\sigma(q)$ نیز حذف شوند و دوباره درج گردند.

این حذف‌ها و درج‌های $p, q, \sigma(p)$ و $\sigma(q)$ هیچ چیزی را برای نقاط و مخروط‌های دیگر تغییر نمی‌دهد؛ زیرا رتبه‌ی آن‌ها تحت تاثیر تعویض رتبه‌ی p و q قرار نمی‌گیرد. بنابراین درخت محدوده بر اساس رتبه، تنها برای $O(\log^d n)$ گره که نقاط p و q را دربردارند، تغییر خواهد کرد. درج‌ها (و حذف‌ها) دو گام را شامل می‌شوند: تعیین گره‌هایی که تحت تاثیر قرار گرفته‌اند و به‌روزرسانی مجموعه یال‌های $(E(P(v), R(v)))$ که برای جفت‌های مخروط-مجزای چنین گره‌هایی در سطح d ایجاد شده‌اند.

۱. **یافتن گره‌های تحت تاثیر واقع شده:** این گام نسبتاً ساده است. ابتدا درج یک نقطه‌ی p را در نظر بگیرید. نقطه‌ی p در T_σ^{rb} جستجو می‌شود و به مجموعه‌های $P(v)$ برای تمام گره‌های روی مسیر جستجو در درخت‌های سطح d اضافه می‌شود. این به‌روزرسانی، تنها زمانی که در مسیر جستجو به پردازش گره‌ای که هنوز وجود ندارد (غیرفعال است) می‌رسد، کمی پیچیده می‌شود. در چنین مواردی به راحتی می‌توان با ایجاد یک گره جدید که $\{p\}$ را به‌عنوان زیرمجموعه‌ی کانونی خود دارد، به‌روزرسانی را انجام داد. بدین منظور، یک **مسیر انفرادی**^{۲۱} از گره جدید ایجاد می‌شود تا به یک برگ برای p برسد

^{۲۱} Single path



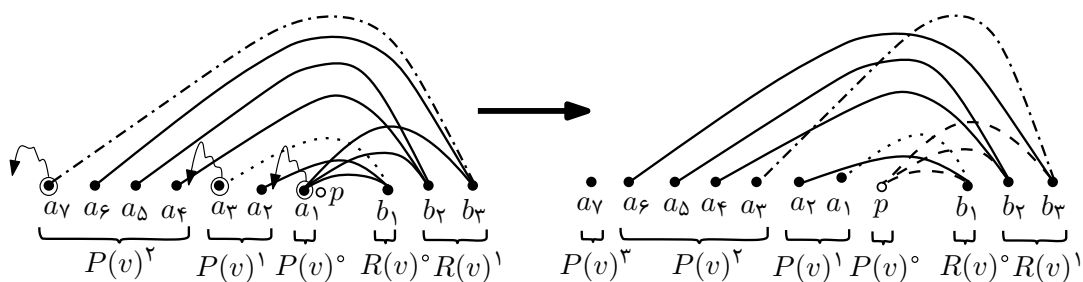
شکل ۱۶.۵: درج نقطه‌ی p در گروه $P(v)^2$.

(شکل ۱۵.۵ را ببینید) و همچنین ساختارهای مرتبط با گره‌های این مسیر جدید نیز ایجاد می‌شوند، که آن‌ها هم مسیرهای انفرادی (با ساختارهای مرتبط) خواهند بود (به صورت بازگشتی در تمام سطوح این ساختارها ایجاد می‌شوند).

درج یک محدوده‌ی $\sigma(p)$ نیز از یک راه مشابه انجام می‌شود. جستجو با این محدوده در T_{σ}^{rb} انجام می‌شود تا زمانی که تمام گره‌های v ، که $\sigma(p) \in R(v)$ می‌باشد، تعیین شوند. هر جا که ضروری بود، گره‌های جدیدی ایجاد می‌شوند. حذف نقاط و محدوده‌ها نیز از راه مشابه انجام می‌شود (ممکن است بر اثر حذف، گره‌هایی غیرفعال شوند).

۲. به روزرسانی مجموعه‌های $E(P(v), R(v))$: گره v را یک گره در سطح d در نظر بگیرید، به طوری که $P(v)$ و $R(v)$ به دلیل حذف یا درج یک نقطه یا محدوده، چنان‌چه قبلاً توضیح داده شد، تغییر کنند. بدیهی است که این تغییر منجر به تغییر در مجموعه یال‌های $E(P(v), R(v))$ نیز خواهد شد. در ادامه توضیح داده می‌شود که چطور یک نقطه در $P(v)$ درج شود به گونه‌ای که این یال‌ها نیز به درستی به روزرسانی شده و خاصیت خوش اتصال بودن خود را حفظ کنند. درج‌ها در $R(v)$ و حذف‌ها نیز به طور مشابه انجام می‌شوند.

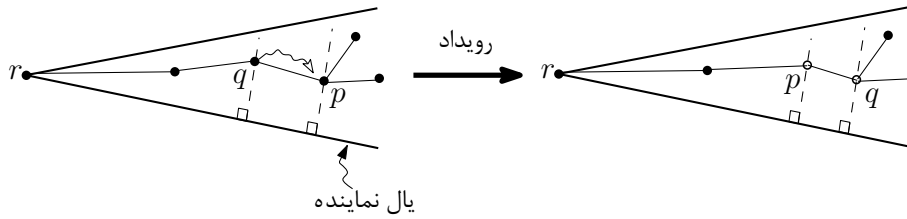
تنها هنگامی که هر دو مجموعه‌ی $P(v)$ و $R(v)$ غیرتهی باشند، لازم است به روزرسانی انجام شود. یادآوری می‌شود که برای ایجاد مجموعه یال‌های $E(P(v), R(v))$ برای جفت مخروط-مجزای $(P(v), R(v))$ ، مجموعه‌های $P(v)$ و $R(v)$ به تعداد لگاریتمی گروه‌های $P(v)^j$ و $R(v)^j$ افزای می‌شوند. سپس از یک گروه $P(v)^j$ ، یال‌هایی به $R(v)^{j-1}$ اضافه می‌شوند، به گونه‌ای که هر $q \in R(v)^{j-1}$ تنها دو یال دریافت می‌کند (و برعکس).



شکل ۱۷.۵: درج نقطه‌ی p در گروه $P(v)^0$.

برای درج p به $P(v)$ ، ابتدا گروه $P(v)^j$ ای که p متعلق به آن است، تعیین می‌شود. برای این که این کار راحت انجام شود، $P(v)$ در یک ترتیب مرتب شده در یک درخت نگهداری می‌شود (این کار می‌تواند بدون تحمیل هزینه‌های اضافی انجام شود). اگر $P(v)^j$ پر باشد، سپس آخرین عنصر آن، p' ، باید به گروه بعدی یعنی $P(v)^{j+1}$ انتقال یابد. شکل ۱۶.۵ را ببینید. در این وضعیت p تمام یال‌های متصل به p' را دریافت می‌کند و به صورت بازگشتی p' در $P(v)^{j+1}$ درج می‌شود. درج بازگشتی p' در $P(v)^{j+1}$ ، ممکن است باعث شود تا آخرین نقطه‌ی p'' از $P(v)^{j+1}$ به $P(v)^{j+2}$ درج شود و همین‌طور تا آخر. شکل ۱۷.۵ را ببینید. بازگشت تا زمانی که یک نقطه در یک گروه غیر پر درج شود ادامه می‌یابد (البته زمانی که از همان ابتدا آخرین گروه $P(v)^j$ پر باشد، باید یک گروه جدید را ایجاد و نقطه را در آن درج کرد. شکل ۱۷.۵ را ببینید).

زمانی که یک نقطه‌ی p در یک گروه غیر پر $P(v)^j$ درج می‌شود، در گروه $R(v)^{j-1}$ (در صورتی که این گروه وجود داشته باشد) نقطه‌ای وجود دارد که هنوز از $P(v)^j$ دو یال ورودی دریافت نکرده است. یک یال به p از چنین نقطه‌ای داده می‌شود و به‌روزرسانی انجام می‌شود. (یک مورد خاص وجود دارد: زمانی که $R(v)^{j-1}$ آخرین گروه از بین $R(v)^i$ ‌هاست، می‌تواند غیر پر یا حتی تهی باشد. در این وضعیت، تنها زمانی که $P(v)^j$ (با نقطه‌ی p) شامل حداکثر $|R(v)^{j-1}| - 2$ نقطه باشد، می‌توان یالی از p به $R(v)^{j-1}$ وصل کرد.) برای این که بتوان به‌طور سریع چنین نقطه‌ای را یافت، یک لیست از تمام نقاط متعلق به $R(v)^i$ که هنوز دو یال ورودی خود را دریافت نکرده‌اند، نگهداری می‌شود. (به‌طور مشابه برای $P(v)^i$ نیز، هنگام درج یک نقطه در $R(v)$ ، یک لیست نگهداری می‌شود.) این اطلاعات اضافی به یک فضای خطی به اندازه‌ی $P(v)$ نیاز دارد و نگهداری آن‌ها نیز به سادگی امکان‌پذیر است.



شکل ۱۸.۵: رویداد: جابه‌جایی تصویر دو نقطه‌ی p و q باهم روی یال نماینده‌ی مخروط

کل روند بیان شده برای درج نقطه‌ی p در $P(v)$ می‌تواند در زمان $O(\log n)$ انجام شود.

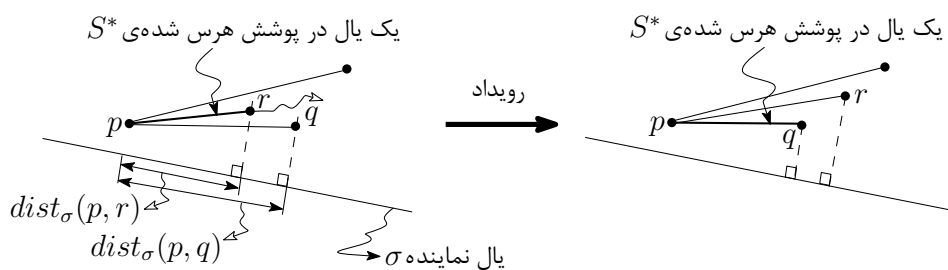
رویدادهای دیگر: نوع دیگری از رویداد وجود دارد که باعث تغییر در مجموعه یال‌های $E(P(v), R(v))$ می‌شود و آن زمانی است که ترتیب دو نقطه از مجموعه‌ی $P(v)$ یا $R(v)$ نسبت به یال نماینده‌ی σ تغییر می‌کند. توجه کنید که این رویداد، به هیچ تغییری در درخت T_σ^{rb} اشاره نمی‌کند؛ زیرا یال نماینده‌ی σ متناظر با هیچ‌کدام از محورهایی که T_σ^{rb} بر اساس آن‌ها تعریف می‌شود، نیست. بنابراین برای تشخیص این نوع رویدادها، کافی است برای هر مخروط σ ، یک آرایه‌ی $A_\sigma[1..n]$ شامل تمام نقاط P ، مرتب شده طبق تصویرشان روی یال نماینده‌ی σ نگهداری شود. سپس زمانی که یک تعویض بین دو نقطه‌ی p و q در این آرایه اتفاق می‌افتد، بررسی می‌شود که آیا هیچ مجموعه‌ی $P(v)$ یا $R(v)$ هست که هر دو نقطه‌ی p و q را شامل شود. اگر چنین مجموعه‌ای وجود داشت، آنگاه تنها کاری که باید انجام شود این است که یال‌های این نقاط در $E(P(v), R(v))$ باهم تعویض شوند. شکل ۱۸.۵ را ببینید.

۳.۳.۵ نگهداری پوشش هرس شده‌ی $S^* = (P, E_{S^*})$

تا این‌جا توصیف شد که چطور درخت محدوده بر اساس رتبه‌ی T_σ^{rb} و نیز گراف $\mathcal{S} = (P, E_S)$ ، هنگام حرکت نقاط نگهداری شوند. اما چنان‌چه قبلاً دیده شد، برای داشتن پوششی با اندازه‌ی خطی، طی روندی پوشش هرس شده‌ی $S^* = (P, E_{S^*})$ به‌دست آمد. بنابراین لازم است تا در ادامه بیان شود که چگونه این پوشش هرس شده^{۲۲}، هنگام حرکت نقاط نگهداری شود.

یادآوری می‌شود که S^* به این روش به‌دست می‌آید: برای هر مخروط σ و هر نقطه‌ی p ، یال $(p, q) \in S$ ، به‌طوری که q نزدیک‌ترین نقطه به p از بین تمام نقاط داخل σ و مجاور به p در S باشد، به عنوان یال S^*

^{۲۲}Pruned spanner



شکل ۱۹.۵: رویداد: جابه‌جایی تصویر دو نقطه‌ی r و q با هم روی یال نماینده‌ی مخروط

انتخاب می‌شود.

برای نگهداری این یال‌ها به صورت وابسته به حرکت، مشاهده می‌شود که یال انتخاب شده برای یک مخروط σ و یک نقطه‌ی p ، تنها زمانی تغییر می‌کند که ترتیب دو نقطه نسبت به یال نماینده‌ی σ تغییر کند (یکی از این نقاط، نقطه‌ی فعلی انتخاب شده برای p است و نقطه‌ی دیگر یک نقطه‌ی جدید می‌باشد) شکل ۱۹.۵ را ببینید. بنابراین هر زمان که دو نقطه‌ی q و r ترتیب‌شان را با هم عوض می‌کنند، نیاز است تا برای تمام نقاط p به طوری که (p, r) و (p, q) یال‌هایی متعلق به E_S هستند، بررسی شود که آیا نیاز به به‌روزرسانی یال انتخاب شده‌ی p هست یا نه (منظور از یال انتخاب شده‌ی p ، یالی است که هنگام هرس کردن پوشش S ، برای نقطه‌ی p ، با توجه به $dist_\sigma$ انتخاب شده است). شکل ۱۹.۵ را ببینید. برای نمونه در رویداد این شکل، یال انتخاب شده برای p هنگام هرس کردن پوشش، نیاز به به‌روزرسانی خواهد داشت. از آن جایی که بیش‌ترین درجه در S ، $O(\log^d n)$ است، این به‌روزرسانی می‌تواند در زمان $O(\log^d n)$ انجام شود؛ زیرا لازم است برای تمام یال‌های متصل به نقطه‌ی r یا q ، بررسی انجام شود. مثلاً اگر r برای این بررسی انتخاب شود، باید برای تمام یال‌های متصل به آن بررسی شود که انتهای دیگر یال چه نقطه‌ای است و آیا این نقطه به q نیز یال دارد. اگر نقطه‌ای به q هم یال داشت، همان نقطه‌ی p معرفی شده در بالاست.

۴.۵ تحلیل پوشش وابسته به حرکت

در این بخش برای ساختار وابسته به حرکت معرفی شده، که برای نگهداری یک $\varepsilon + 1$ -پوشش در \mathbb{R}^d است، چهار معیار ارزیابی کیفیت ساختارهای وابسته به حرکت، یعنی پاسخ‌گویی، کارایی، فشردگی و محلی بودن داده‌ها بررسی می‌شود.

در حالت کلی پوشش وابسته به حرکت معرفی شده، از $d+1$ لیست مرتب شده برای هر مخروط $\sigma \in TC$

به منظور نگهداری پوشش \mathcal{S}^* نیاز دارد، که d لیست مرتب شده‌ی آن برای نگهداری ترتیب نقاط در d سطح (بعد) درخت T_σ^{rb} ، و یک لیست هم برای نگهداری ترتیب نقاط نسبت به یال نماینده‌ی σ می‌باشد. بنابراین تحت این فرض که نقاط دارای مسیرهای حرکت شبه‌جبری از درجه‌ی محدود هستند (بخش ۱.۱ را ببینید)، تعداد رویدادهایی که باید پردازش شوند $O(n^2/\varepsilon^{d-1})$ است؛ زیرا همان‌طور که در زیربخش ۱.۱.۲، راه‌حل دوم بیان شد، تعداد رویدادها برای هر لیست مرتب شده‌ی نقاط برابر است با تعداد دفعات تغییر ترتیب نقاط، یعنی تعداد برخوردها بین مسیرهای حرکت نقاط و از مرتبه‌ی $O(n^2)$ است و این ساختار به $O(1/\varepsilon^{d-1})$ لیست نیاز دارد. با توجه به آن‌چه که توسط گائو و همکارانش در [۲۷] مطالعه شده است، تعداد رویدادهای یک پوشش وابسته به حرکت دارای کران پایین $\Omega(n^2)$ است، در نتیجه نسبت تعداد کل رویدادها به تعداد رویدادهای خارجی در بدترین وضعیت $O(1/\varepsilon^{d-1})$ خواهد شد و ساختار کاراست.

هر رویداد می‌تواند در زمان $O(\log^{d+1} n)$ پردازش شود؛ زیرا برای درج یا حذف یک نقطه در درخت T_σ^{rb} ، باید حداکثر به تعداد $O(\log^d n)$ حذف یا درج در این درخت انجام داد. از طرفی هر نقطه در سطح d این درخت نیز حداکثر در $O(\log^d n)$ سطح از درخت‌های این سطح حضور دارد، بنابراین در حداکثر $O(\log^d n)$ جفت $(P(v), R(v))$ نیز حضور خواهد داشت، در نتیجه با حذف یا درج هر نقطه در ساختار، مجموعه یال‌های $E(P(v), R(v))$ نیز نیاز به به‌روزرسانی خواهند داشت. قبلاً بیان شد که این به‌روزرسانی برای هر مجموعه یال $E(P(v), R(v))$ ، به زمان $O(\log n)$ نیاز دارد و از آن‌جایی که، حداکثر $O(\log^d n)$ مجموعه یال نیاز به به‌روزرسانی دارند، پس هر رویداد می‌تواند در زمان $O(\log^{d+1} n)$ پردازش شود و ساختار پاسخ‌گو نیز هست. علاوه بر آن هر نقطه حداکثر در دو تاییدیه در هر لیست مرتب شده شرکت دارد؛ زیرا تاییدیه‌ها از نوع مجاورت هستند و هر نقطه حداکثر دو مجاورت دارد. بنابراین در مجموع، هر نقطه در $O(1/\varepsilon^{d-1})$ (تعداد لیست‌ها) تاییدیه مشارکت دارد. پس ساختار محلی است. در نتیجه، به‌روزرسانی یک طرح پرواز تنها به $O(\log n/\varepsilon^{d-1})$ زمان نیاز دارد. (ضریب لگاریتمی برای به‌روزرسانی صف رویداد می‌باشد.) همچنین از محلی بودن داده‌ها نتیجه می‌شود که ساختار فشرده نیز هست.

قضیه ۱.۴.۵ [۱] برای هر $\varepsilon > 0$ و برای هر مجموعه‌ی P از n نقطه در \mathbb{R}^d ، یک $1 + \varepsilon$ -پوشش وابسته به حرکت وجود دارد که دارای این خواص است: اندازه‌ی آن $O(n/\varepsilon^{d-1})$ و بیش‌ترین درجه‌ی آن $O(\log^d n)$ است. علاوه بر آن هر نقطه در $O(1/\varepsilon^{d-1})$ تاییدیه واقع خواهد شد و تحت این فرض که مسیرهای حرکت نقاط شبه‌جبری از درجه‌ی محدود هستند، تعداد رویدادهایی که باید پردازش شوند $O(n^2/\varepsilon^{d-1})$ است. هر رویداد می‌تواند در زمان $O(\log^{d+1} n)$ ، با استفاده از یک ساختمان داده‌ی کمکی که به $O((n/\varepsilon^{d-1}) \log^d n)$

حافظه نیاز دارد، پردازش شود. به روزرسانی یک طرح پرواز نیز تنها به $O(\log n/\varepsilon^{d-1})$ زمان نیاز دارد.

فصل ۶

نتیجه‌گیری

هدف از این پایان‌نامه، مطالعه‌ی ساختارهای وابسته به حرکت، پوشش‌های هندسی و ترکیب این دو مفهوم برای داشتن پوشش‌های وابسته به حرکتی کارا بود. اهمیت مطالعه‌ی این نوع پوشش‌ها به خاطر کاربردهایی است که شبکه‌های متحرک (از جمله شبکه‌های همراه، بی‌سیم و غیره) در دنیای امروز ما دارند.

در این پایان‌نامه پوشش‌های وابسته به حرکت جدیدی در صفحه و نیز در فضای \mathbb{R}^d تحت این فرض که نقاط پوشش دارای مسیرهای حرکت شبه‌جبری از درجه‌ی محدود هستند، معرفی شدند. این پوشش‌ها دارای اندازه‌ی خطی از تعداد نقاط هستند و علاوه بر دارا بودن مهم‌ترین معیارهای ارزیابی کیفیت ساختارهای وابسته به حرکت، یعنی کارایی و پاسخ‌گویی، این ویژگی بارز را نیز دارند که اولین $\varepsilon + 1$ -پوشش‌هایی هستند که تعداد رویدادها و زمان پاسخ‌گویی آن‌ها، به میزان پراکندگی نقاط از هم وابسته نیست.

هرچند که دو پوشش معرفی شده در فصل ۴ و ۵ دارای خواص ذکر شده در بالا هستند، اما پوشش ارائه شده در فصل ۵ برتری‌هایی نسبت به پوشش ارائه شده در فصل ۴ دارد، از جمله این‌که: جامع‌تر است و برای نقاط در فضای \mathbb{R}^d ($d \geq 2$) است، ساختار وابسته به حرکت پوشش محلی است و وابستگی آن به ε نیز کم‌تر است.

این پوشش‌ها مهم‌ترین معیارها و ویژگی‌های مورد نظر را دارند، اما متأسفانه، وزن آن‌ها می‌تواند از وزن درخت پوشای کمینه‌ی مجموعه نقاط P ، که با نماد $wt(MST(P))$ نشان داده می‌شود، خیلی بیش‌تر شود. بنابراین معرفی یک $\varepsilon + 1$ -پوشش وابسته به حرکت کارا که وزن کل آن از مرتبه‌ی $O(wt(MST(P)))$ باشد، به عنوان مسئله‌ای برای کارهای تحقیقاتی آینده باقی می‌ماند.

واژه‌نامه فارسی به انگلیسی

Join	الحاق
Size	اندازه
On-line	بر-خط
Largest empty circle	بزرگ‌ترین دایره‌ی خالی
Responsiveness	پاسخ‌گویی
Full	پر
Spread	پراکندگی
Query	پرسش
Spanner	پوشش
Upper envelope	پوشش بالایی
Lower envelope	پوشش پایینی
Diamond-Delaunay spanner	پوشش لوزی-دلونی
Kinetic spanner	پوشش وابسته به حرکت
Pruned spanner	پوشش هرس شده
Geometric spanner	پوشش هندسی
Plane sweep	پیمایش صفحه
Ackermann function	تابع آکرمان
Convex distance function	تابع فاصله‌ی محدب
Configuration function	تابع وضعیت
Dilation	تاخیر
Certificate	تاییدیه
Cone-Separated Pairs Decomposition	تجزیه‌ی جفت‌های مخروط-مجزا
Lexicographically order	ترتیب الفبایی
Topological change	تغییر توپولوژیکی
Maximal subdivision	تقسیم‌بندی ماکسیمال
Split	جداسازی
Edge flip	چرخش یال
Pseudo algebraic motions	حرکت‌های شبه‌جبری
Well connected	خوش اتصال
Degree	درجه
Skeleton tree	درخت اسکلت
Range tree	درخت محدوده
Rank-based range tree	درخت محدوده بر اساس رتبه

Tournament tree	درخت مسابقه
Davenport-Schinzel sequence	دنباله‌ی داونپورت-شینزل
Hull vertex	رأس غشا
Range searching techniques	روش‌های جستجوی محدوده
Event	رویداد
External event	رویداد خارجی
Internal event	رویداد داخلی
Canonical subset	زیرمجموعه‌ی کانونی
Kinetic data structure	ساختمان داده‌ی وابسته به حرکت
Geometric network	شبکه‌ی هندسی
Event queue	صف رویداد
Stretch factor	ضریب کشش
Scale factor	ضریب معیار
Flight plan	طرح پرواز
Convex hull	غشای محدب
Inactive	غیرفعال
Compactness	فشردگی
Active	فعال
Dominance wedge	قاش نفوذ
Main diagonal	قطر اصلی
Efficiency	کارایی
Euclidean graph	گراف اقلیدسی
Delaunay graph	گراف دلونی
Dual graph	گراف دوگان
Expanding waves	گسترش موج‌ها
Triangulation	مثلث‌بندی
Delaunay Triangulation	مثلث‌بندی دلونی
Legal triangulation	مثلث‌بندی مجاز
Convex	محدب
Locality	محلی بودن داده‌ها
Canonical cones	مخروط‌های کانونی
Kinetic tournament	مسابقه‌ی وابسته به حرکت
Path	مسیر
Single path	مسیر انفرادی
Bisector	منصف

Closest pair	نزدیک‌ترین جفت
Voronoi diagram	نمودار ورنوی
Kinetization	وابسته به حرکت کردن
Weight	وزن
General Position	وضعیت عمومی
Pruning	هرس
Max heap	هرم بیشینه
Kinetic heap	هرم وابسته به حرکت
Homeomorphism	همومورفیسم
Homeomorphic	همومورفیک
Interior edge	یال داخلی
Hull edge	یال غشا
Non-corner edge	یال غیر گوشه
Illegal edge	یال غیرمجاز
Corner edge	یال گوشه
Representative edge	یال نماینده
Non-crossing edges	یال‌های نامتقاطع

واژه‌نامه انگلیسی به فارسی

Ackermann function	تابع آکرمان
Active	فعال
Bisector	منصف
Canonical subset	زیرمجموعه‌ی کانونی
Canonical cones	مخروط‌های کانونی
Certificate	تاییدیه
Closest pair	نزدیک‌ترین جفت
Configuration function	تابع وضعیت
Compactness	فشردگی
Cone-Separated Pairs Decomposition	تجزیه‌ی جفت‌های مخروط-مجزا
Convex	محدب
Convex distance function	تابع فاصله‌ی محدب
Convex hull	غشای محدب
Corner edge	یال گوشه
Davenport-Schinzel sequence	دنباله‌ی داوونپورت-شینزل
Degree	درجه
Delaunay graph	گراف دلونی
Delaunay Triangulation	مثلث‌بندی دلونی
Diamond-Delaunay spanner	پوشش لوزی-دلونی
Dilation	تاخیر
Dominance wedge	قاش نفوذ
Dual graph	گراف دوگان
Edge flip	چرخش یال
Efficiency	کارایی
Euclidean graph	گراف اقلیدسی
Event	رویداد
Event queue	صف رویداد
Expanding wave	گسترش موج
External event	رویداد خارجی
Flight plan	طرح پرواز
Full	پر
General Position	وضعیت عمومی
Geometric network	شبکه‌ی هندسی

Geometric spanner	پوشش هندسی
Homeomorphic	همومورفیک
Homeomorphism	همومورفیسم
Hull edge	یال غشا
Hull vertex	رأس غشا
Illegal edge	یال غیرمجاز
Inactive	غیرفعال
Interior edge	یال داخلی
Internal event	رویداد داخلی
Join	الحاق
Kinetic data structure	ساختمان داده‌ی وابسته به حرکت
Kinetic heap	هرم وابسته به حرکت
Kinetic spanner	پوشش وابسته به حرکت
Kinetic tournament	مسابقه‌ی وابسته به حرکت
Kinetization	وابسته به حرکت کردن
Largest empty circle	بزرگ‌ترین دایره‌ی خالی
Legal triangulation	مثلث‌بندی مجاز
Lexicographically order	ترتیب الفبایی
Locality	محل‌ی بودن داده‌ها
Lower envelope	پوشش پایینی
Main diagonal	قطر اصلی
Max heap	هرم بیشینه
Maximal subdivision	تقسیم‌بندی ماکسیمال
Non-corner edge	یال غیر گوشه
Non-crossing edges	یال‌های نامتقاطع
On-line	بر-خط
Path	مسیر
Plane sweep	پیمایش صفحه
Pruned spanner	پوشش هرس شده
Pruning	هرس
Pseudo algebraic motions	حرکت‌های شبه‌جبری
Query	پرسش
Range searching techniques	روش‌های جستجوی محدوده
Range tree	درخت محدوده
Rank-based range tree	درخت محدوده بر اساس رتبه

Representative edge	یال نماینده
Responsiveness	پاسخ‌گویی
Scale factor	ضریب معیار
Single path	مسیر انفرادی
Size	اندازه
Skeleton tree	درخت اسکلت
Spanner	پوشش
Split	جداسازی
Spread	پراکندگی
Stretch factor	ضریب کشش
Topological change	تغییر توپولوژیکی
Tournament tree	درخت مسابقه
Triangulation	مثلث‌بندی
Upper envelope	پوشش بالایی
Voronoi diagram	نمودار ورنوی
Weight	وزن
Well connected	خوش اتصال

مراجع

- [1] Abam, M. A., and de Berg, M. Kinetic spanners in \mathbb{R}^d . *Discrete and Computational Geometry* 45 (2011), 723–736.
- [2] Abam, M. A., de Berg, M., and Gudmundsson, J. A simple and efficient kinetic spanner. *Computational Geometry: Theory and Applications* 43 (2010), 251–256.
- [3] Abam, M. A., de Berg, M., and Speckmann, B. Kinetic kd-trees and longest-side kd-trees. In *Proceedings of ACM Symposium on Computational Geometry* (2007), pp. 364–372.
- [4] Albers, G., Guibas, L., Mitchell, J., and Roos, T. Voronoi diagrams of moving points. *International Journal on Computational Geometry and Applications* 8, 3 (1998), 365–380.
- [5] Basch, J. *Kinetic Data Structures*. Ph.D. thesis, Department of Computer Science, Stanford University, 1999.
- [6] Basch, J., Guibas, L. J., and Hershberger, J. Data structures for mobile data. *Journal of Algorithms* 31 (1999), 1–28.
- [7] Basch, J., Guibas, L. J., and Ramkumar, G. D. Sweeping lines and line segments with a heap. In *Proceedings of the 13th Annual ACM Symposium on Computational Geometry* (1997), pp. 469–471.
- [8] Basch, J., Guibas, L. J., and Zhang, L. Proximity problems on moving points. In *Proceedings of ACM Symposium on Computational Geometry* (1997), pp. 344–351.
- [9] Bentley, J. L., and Ottmann, T. A. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers* c-28 (1979), 643–647.

- [10] Bespamyatnikh, S. N. An optimal algorithm for closest pair maintenance. In *Proceedings of the 11th Annual ACM Symposium on Computational Geometry* (1995), pp. 152–161.
- [11] Bose, P., Carmi, P., Collette, S., and Smid, M. On the stretch factor of convex delaunay graphs. *Journal of Computational Geometry* 1, 1 (2010), 41–56.
- [12] Bose, P., Gudmundsson, J., and Morin, P. Ordered theta graphs. *Computational Geometry: Theory and Applications* 28 (2004), 11–18.
- [13] Brown, K. Q. Comments on "Algorithms for reporting and counting geometric intersections". *IEEE Transactions on Computers* c-30 (1981), 147–148.
- [14] Callahan, P. B., and Kosaraju, S. R. Algorithms for dynamic closest-pair and n-body potential fields. In *Proceedings of the 6th ACM–SIAM Symposium on Discrete Algorithms* (1995), pp. 263–272.
- [15] Chew, L. P. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences* 39, 2 (1989), 205–219.
- [16] Chew, L. P. Near-quadratic bounds for the L_1 Voronoi diagram of moving points. *Computational Geometry: Theory and Applications* 7 (1997), 73–80.
- [17] Chew, L. P., and Kedem, K. Placing the largest similar copy of a convex polygon among polygonal obstacles. In *Proceedings of 5th Symposium on Computational Geometry* (1989), ACM New York, pp. 167–174.
- [18] Chew, L. P., and Kedem, K. A convex polygon among polygonal obstacles: placement and high-clearance motion. *Computational Geometry: Theory and Applications* 3 (1993), 59–89.
- [19] Chew, L. P., and (Scot) Drysdale III, R. L. Voronoi diagrams based on convex distance functions. In *Proceedings of the 1st ACM Symposium on Computational Geometry* (1985), pp. 235–244.
- [20] Christensen, J. S. Experimental study of kinetic geometric t-spanner algorithms. Master thesis, Department of Computer Science, Aarhus University, 2010.
- [21] Clarkson, K. Approximation algorithms for shortest path motion planning. In *Proceedings of ACM Symposium on Theory of Computing* (1987), pp. 56–65.

- [22] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms*, 3rd ed. The MIT Press, Cambridge, Massachusetts, 2009.
- [23] Cui, S., Kanj, I. A., and Xia, G. On the stretch factor of delaunay triangulations of points in convex position. *Computational Geometry* 44, 2 (2011), 104–109.
- [24] de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer-Verlag, Berlin, Germany, 2008.
- [25] de Berg, M., Comba, J., and Guibas, L. J. A segment-tree based kinetic BSP. In *Proceedings of ACM Symposium on Computational Geometry* (2001), pp. 134–140.
- [26] Farshi, M. *A Theoretical and Experimental Study of Geometric Networks*. Ph.D. thesis, Faculty of Mathematics and Computer Science, Eindhoven University of Technology, 2008.
- [27] Gao, J., Guibas, L. J., and Nguyen, A. Deformable spanners and applications. *Computational Geometry: Theory and Applications* 35, 1–2 (2006), 2–19.
- [28] Golin, M., Raman, R., Schwarz, C., and Smid, M. Randomized data structures for the dynamic closest-pair problem. In *Proceedings of the 4th ACM–SIAM Symposium on Discrete Algorithms* (1993), pp. 301–310.
- [29] Hershberger, J. Finding the upper envelope of n line segments in $O(n \log n)$ time. *Information Processing Letters* 33 (1989), 169–174.
- [30] Karavelas, M., and Guibas, L. Static and kinetic geometric spanners with applications. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms* (2001), pp. 168–176.
- [31] Keil, J. Approximating the complete euclidean graph. In *In Proceedings of the Scandinavian Workshop Algorithm Theory*, vol. 318 of *Lecture Notes in Computer Science*. Springer, Berlin, 1988, pp. 208–213.
- [32] Keil, J., and Gutwin, C. Classes of graphs which approximate the complete Euclidean graph. *Discrete and Computational Geometry* 7 (1992), 13–28.
- [33] Klein, R., and Wood, D. Voronoi diagrams based on general metrics in the plane. In *In Proceedings of the 5th Symposium on Theoretical Aspects of Computer Science*, vol. 294 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1988, pp. 281–291.

- [34] Lee, D. T. Two-dimensional Voronoi diagrams in the L_p metric. *JACM* 27 (1980), 604–618.
- [35] Ma, L. *Bisectors and Voronoi Diagrams for Convex Distance Functions*. Ph.D. thesis, Department of Computer Science, Fern University Hagen, Germany, 2000.
- [36] Mehlhorn, K. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*. Springer, Berlin, EATCS Monographs, 1984.
- [37] Narasimhan, G., and Smid, M. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- [38] Sharir, M., and Agarwal, P. K. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.
- [39] Willard, D., and Lueker, G. Adding range restriction capability to dynamic data structures. *ACM* 32, 3 (1985), 597–617.
- [40] Zhang, L. *Kinetic Maintenance of Proximity Structures*. Ph.D. thesis, Department of Computer Science, Stanford University, 2000.

Abstract

In this thesis, we study $1 + \varepsilon$ -spanners for a set of n points in the plane and in d -dimensional Euclidean space that can be maintained efficiently as the points move. The kinetic spanner in the plane has size $O(n/\varepsilon^2)$. Assuming the trajectories of the points can be described by polynomials whose degrees are at most s , the number of events is $O(n^2\beta(n))$ ($\beta(n)$ grows slower than logarithmic functions), and at each event the spanner can be updated in $O(1)$ time. The kinetic spanner in \mathbb{R}^d has size $O(n/\varepsilon^{d-1})$ and maximum degree $O(\log^d n)$. Assuming that the trajectories of the points can be described by bounded-degree polynomials, the number of events is $O(n^2/\varepsilon^{d-1})$, and using a supporting data structure of size $O((n/\varepsilon^{d-1}) \log^d n)$, we can handle events in time $O(\log^{d+1} n)$. Moreover, the spanner can be updated in time $O(\log n/\varepsilon^{d-1})$ if the flight plan of a point changes. These spanners are the first kinetic spanners whose performance does not depend on the spread of the point set.

Yazd University

Faculty of Mathematics

Department of Computer Science

Thesis submitted

for the degree of Master of Science

Title:

Geometric Spanners for Mobile Points

Supervisor:

Dr. Mohammad Farshi

Advisor:

Dr. Mohammad Reza Hooshmand Asl

By:

Zorieh Soltani Gerdefaramarzi

October, 2011