

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه یزد

پردیس علوم

پایان نامه

برای دریافت درجه دکتری  
علوم کامپیوتر-محاسبات علمی

عنوان

ارائه‌ی چند الگوریتم برای ساخت پوشش‌های هندسی

استاد راهنما

دکتر محمد فرشی

استادان مشاور

دکتر محمد علی آبام و دکتر مهدیه هاشمی نژاد

پژوهش و نگارش

داود بخشش

تیرماه ۱۳۹۶



کلیه‌ی حقوق مادی و معنوی مترتب بر نتایج مطالعات، ابتکارات و نوآوری‌های ناشی از تحقیق موضوع این پایان‌نامه/رساله متعلق به دانشگاه یزد است و هرگونه استفاده از نتایج علمی و عملی از این پایان‌نامه/رساله برای تولید دانش فنی، ثبت اختراع، ثبت اثر بدیع هنری، همچنین چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس و ارائه مقاله در سمینارها و مجلات علمی از این پایان‌نامه/رساله منوط به موافقت کتبی دانشگاه یزد است.



تقدیم به

پدر و مادر و همسر 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 می‌گوییم 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 آییم. به‌راستی که همه چیز از آن اوست و همه چیز به خواست اوست.

در اینجا لازم است از تمامی افرادی که در نوشتن این رساله به من کمک کردند، تشکر نمایم. اول از همه، از استاد راهنمای گرانقدرم، دکتر محمد فرشی، کمال تشکر و قدردانی را دارم. ایشان، در این مدت چهارساله، به من بسیار راهنمایی و کمک کردند و با صبر، درایت و اندوخته‌های علمی ارزشمندشان همواره پاسخگوی سوالات زیاد بنده بودند. ایده‌های ناب و خلاقانه ایشان، مشوق و راهنمای خیلی خوبی برای بنده در این مدت بود.

من همچنین از پدر و مادرم به خاطر حمایت‌های بی‌دریغشان تشکر می‌کنم. مطمئناً، دعای خیر این عزیزان اگر همراه من نبود، نمی‌توانستم در نوشتن این رساله، موفق شوم. درنهایت، از همسرم به خاطر حمایت‌ها و تشویق‌هایش، بسیار سپاس گزارم.



بسمه تعالی

شناسه: ب/د/3	صور تجلسه دفاعیه پایان نامه دانشجوی دوره دکتری	 مدیریت تحصیلات تکمیلی
<p>جلسه دفاعیه پایان نامه تحصیلی آقای/خانم: داود بخشش رشته/گرایش: علوم کامپیوتر/علوم کامپیوتر-محاسبات علمی تحت عنوان: ارائه‌ی چند الگوریتم برای ساخت پوشش‌های هندسی و تعداد واحد: ۱۸ در تاریخ ۱۳۹۶/۴/۲۸ با حضور اعضای هیأت داوران (به شرح ذیل) تشکیل گردید. پس از ارزیابی توسط هیأت داوران، پایان نامه با نمره به عدد ۱۹.۷۵ به حروف نوزده و هفتاد و پنج صدم و درجه عالی مورد تصویب قرار گرفت.</p>		
<u>امضاء</u>	<u>نام و نام خانوادگی</u>	<u>عنوان</u>
	دکتر محمد فرشی	استاد/استادان راهنما: الف:
	دکتر محمد علی آبام دکتر مهدیه هاشمی نژاد	استاد/استادان مشاور: الف: ب:
	دکتر محمد رضا هوشمند اصل	داور داخل گروه: الف:
	دکتر محمد قدسی دکتر علی محدث خراسانی	داور خارج از گروه: الف: ب:
	نماینده تحصیلات تکمیلی دانشگاه (ناظر) نام و نام خانوادگی: دکتر علی محمد لطیف امضاء:	



## چکیده

در این رساله به ارائه‌ی چند الگوریتم جدید برای ساخت پوشش‌های هندسی و همچنین بهبود برخی الگوریتم‌های موجود و بررسی برخی خواص هندسی پوشش‌های هندسی می‌پردازیم. یک گراف هندسی (اقلیدسی) را یک  $t$ -پوشش (برای  $t > 1$ ) گوئیم اگر بین هر دو رأس گراف، مسیری به طول حداکثر  $t$  برابر فاصله‌ی اقلیدسی آنها وجود داشته باشد.

در این رساله؛ ابتدا گراف یائوی پیوسته را معرفی می‌کنیم که از چرخش یک مخروط به مرکز هر نقطه و اتصال آن نقطه به نقاطی که در یک لحظه از این چرخش، نزدیک‌ترین نقطه به رأس مخروط بوده‌اند، بدست می‌آید. هرچند این گراف با در نظر گرفتن مقدار مناسب برای زاویه‌ی مخروط، یک  $t$ -پوشش است، اما تعداد یال‌های گراف می‌تواند مربعی باشد. اما، نشان می‌دهیم گراف حاصل، در برابر خطاهای ناحیه‌ای تحمل‌پذیر است.

سپس الگوریتم شکاف-حریصانه را که در سال ۱۹۹۷ برای ساخت پوشش‌های هندسی معرفی شد را بررسی و با تغییراتی، زمان اجرا و حافظه‌ی مصرفی آن را کاهش می‌دهیم. همچنین مقدار  $t$  را در نتایج مربوط به  $t$ -پوشش بودن نتیجه‌ی این الگوریتم بهبود می‌دهیم.

در ادامه، به بررسی گراف‌های هندسی خود-گرا و وتر-افزایشی می‌پردازیم. یک گراف هندسی را خود-گرا گوئیم، اگر برای هر دو رأس گراف، مسیری بین آن دو رأس وجود داشته باشد که در حرکت روی آن مسیر، از مبدأ به سمت مقصد، هیچگاه از نقاط پیش رو در مسیر دور نشویم. چنین مسیری را یک مسیر خود-گرا می‌نامیم. گراف هندسی، وتر-افزایشی است اگر برای هر دو رأس، یک مسیر وجود داشته باشد که با شروع از هر انتها به عنوان مبدأ، یک مسیر خود-گرا باشد. در این رساله، نشان می‌دهیم که برای هر مجموعه‌ای از نقاط در صفحه که روی اضلاع یک مثلث حاده قرار گرفته‌اند، یک گراف وتر-افزایشی مسطح با به کار بردن حداکثر سه نقطه‌ی اشتاینری وجود دارد. همچنین، نشان می‌دهیم که برای هر مجموعه‌ای از نقاط در صفحه که روی اضلاع یک مثلث منفرجه یا قائم‌الزاویه قرار گرفته‌اند، یک گراف وتر-افزایشی مسطح وجود دارد.

سپس؛ مفهوم مسیر خود-گرای ضعیف را با جایگزینی دور نشدن از تمام نقاط پیش رو در طول حرکت، با شرط دور نشدن از مقصد، تعریف می‌کنیم و نشان می‌دهیم که برای هر مجموعه‌ای از نقاط در صفحه، یک گراف خود-گرای ضعیف مسطح وجود دارد.

در ادامه، یک الگوریتم تقسیم و غلبه برای ساخت پوشش‌های هندسی ارائه می‌کنیم که یک پوشش هندسی با تعداد یال‌های نزدیک به خطی می‌سازد.

در نهایت به بررسی گراف‌های هندسی مقید به زاویه می‌پردازیم. یک گراف هندسی را مقید به زاویه‌ی  $\theta$  می‌نامیم، اگر زاویه‌ی بین هر دو یال مجاور به یک رأس در گراف، حداقل  $\theta$  باشد. نشان می‌دهیم که برای هر  $\theta > \pi/3$ ، لزوماً یک گراف همبند مقید به زاویه‌ی  $\theta$  برای هر مجموعه از نقاط وجود ندارد. همچنین نشان می‌دهیم برای  $\theta = \pi/3$  و هر  $t > 2/\sqrt{3}$ ، لزوماً یک  $t$ -پوشش مقید به زاویه‌ی  $\theta$  نیز برای هر مجموعه از نقاط وجود ندارد.



# فهرست مطالب

د	فهرست جداول
ه	فهرست تصاویر
ح	فهرست نمادها
۱	۱ پیشگفتار
۲	۱-۱ پوشش هندسی
۵	۲-۱ الگوریتم‌های ساخت $t$ -پوشش
۵	۱-۲-۱ الگوریتم مسیر-حریصانه
۷	۲-۲-۱ الگوریتم شکاف-حریصانه
۹	۳-۲-۱ WSPD-پوشش
۱۱	۴-۲-۱ گراف یائو و $\Theta$ -گراف
۱۲	۵-۲-۱ $t$ -پوشش تحمل‌پذیر ناحیه-خطا
۱۳	۶-۲-۱ گراف‌های خود-گرا
۱۴	۳-۱ کلیات رساله
۱۹	۲ گراف یائوی پیوسته
۲۰	۱-۲ مقدمه
۲۲	۲-۲ گراف یائوی پیوسته برای مخروط‌های باریک
۲۴	۳-۲ گراف $cY(2\pi/3)$ یک $411/6^\circ$ -پوشش
۳۲	۴-۲ زاویه‌های بزرگ‌تر
۳۴	۵-۲ تحمل‌پذیری خطا در $cY(\theta)$
۳۸	۶-۲ خود-گرایی گراف‌های یائوی پیوسته

۴۰ ..... نتیجه‌گیری ۷-۲

### ۳ بهبود پیچیدگی زمانی و حافظه‌ی پوشش شکاف-حریصانه ۴۱

۴۲ ..... مقدمه ۱-۳

۴۴ ..... بهبود پیچیدگی حافظه و زمان ۲-۳

۴۵ ..... ۱-۲-۳ بهبود پیچیدگی حافظه

۵۴ ..... ۲-۲-۳ بهبود پیچیدگی زمانی

۵۶ ..... ۳-۳ بهبود ضریب کشش

۵۸ ..... ۴-۳ الگوریتم‌هایی از مرتبه‌ی ۳

۵۹ ..... ۱-۴-۳ الگوریتم شکاف-حریصانه‌ی تصادفی

۵۹ ..... ۲-۴-۳ الگوریتم شکاف-حریصانه‌ی ممنوع

۶۱ ..... ۳-۴-۳ الگوریتم شکاف-حریصانه‌ی دودویی

۶۱ ..... ۵-۳ نتایج پیاده‌سازی

۶۵ ..... ۶-۳ نتیجه‌گیری

### ۴ گراف‌های هندسی خود-گرا ۶۷

۶۸ ..... ۱-۴ مقدمه

۷۳ ..... ۲-۴ گراف مسطح وتر-افزایشی برای نقاط روی مثلث

۸۳ ..... ۳-۴ گراف‌های خود-گرای ضعیف

۸۴ ..... ۱-۳-۴ تشخیص خود-گرای ضعیف بودن یک گراف

۸۶ ..... ۴-۴ ساخت گراف‌های خود-گرای ضعیف مسطح

۹۵ ..... ۵-۴ پوشش‌های خود-گرای ضعیف

۹۶ ..... ۱-۵-۴ الگوریتمی مشابه الگوریتم گراف یائو

۱۰۰ ..... ۲-۵-۴ یک الگوریتم مبتنی بر WSPD

۱۰۱ ..... ۶-۴ نتیجه‌گیری



۱۰۳	۵ ادغام پوشش‌های هندسی و کاربردهای آن
۱۰۴	۱-۵ مقدمه
۱۰۵	۲-۵ ادغام پوشش‌ها
۱۱۱	۳-۵ کاربردها
۱۱۱	۱-۳-۵ ساخت یک $t$ -پوشش
۱۱۵	۲-۳-۵ ساخت پوشش‌های تحمل‌پذیر ناحیه-خطا
۱۱۷	۴-۵ نتیجه‌گیری
۱۱۹	۶ پوشش‌های هندسی مقید به زاویه
۱۲۰	۱-۶ مقدمه
۱۲۰	۲-۶ نتیجه‌ی اصلی
۱۲۳	۳-۶ نتیجه‌گیری
۱۲۵	نتیجه‌گیری
۱۲۹	واژه‌نامه فارسی به انگلیسی
۱۳۳	واژه‌نامه انگلیسی به فارسی
۱۴۱	منابع و مآخذ
۱۴۷	نمایه

# فهرست جداول

- ۱-۳ زمان اجرای (برحسب ثانیه) Orig و Bina برای مقادیر مختلف  $\theta$  و  $w$  و مجموعه نقاط با ۱۰۰، ۵۰۰، ۱۰۰۰، ۲۰۰۰ و ۵۰۰۰ نقطه. . . . . ۶۳
- ۲-۳ زمان اجرای محاسبه‌ی پوشش حریصانه و پوشش شکاف-حریصانه برای مجموعه نقاط مختلف و پارامترهای متفاوت. در اینجا منظور از T-Gap و T-Greedy به ترتیب زمان ساخت پوشش شکاف-حریصانه و پوشش حریصانه است. . . . . ۶۶

# فهرست تصاویر

- ۱-۱  $t$ -پوشش برای  $t$ های مختلف. . . . . ۳
- ۲-۱ نقطه‌ی  $u$  به نزدیک‌ترین نقطه در هر مخروط وصل می‌شود. . . . . ۱۲
- ۳-۱ مسیر خود-گرا و وتر-افزایشی. . . . . ۱۴
- ۱-۲ تعداد یال‌های گراف یائوی پیوسته می‌تواند مربعی باشد. . . . . ۲۱
- ۲-۲ . . . . . ۲۵
- ۳-۲ ناحیه‌های مجاور به  $a$  و  $b$  در حالت الف). . . . . ۲۷
- ۴-۲ (آ) نقطه‌ی  $v^*$  و زاویه‌ی  $\psi = \text{angle}(v^*a, ac^*)$ . ب) مخروط  $C'_{ab}$  که با چرخاندن  $C_{ab}$  به اندازه‌ی  $\psi$  درجه در جهت خلاف عقربه‌های ساعت، بدست می‌آید. . . . . ۲۹
- ۵-۲  $N_b$  و  $N'_a$  . . . . . ۳۰
- ۶-۲ اگر  $\theta > \pi$ ، آنگاه  $cY(\theta)$  می‌تواند ناهمبند باشد. . . . . ۳۱
- ۷-۲ اثبات قضیه ۲-۴-۳. . . . . ۳۳
- ۸-۲ اثبات قضیه‌ی ۲-۵-۲. . . . . ۳۷
- ۹-۲ یک مجموعه نقطه که  $Y_4$  روی آن خود-گرا نیست اما  $cY(\pi/2)$  روی آن خود-گرا است. یال نقطه‌چین متعلق به  $Y_4$  نیست ولی متعلق به  $cY(\pi/2)$  است. . . . . ۳۸
- ۱۰-۲ اثبات قضیه‌ی ۲-۶-۱. . . . . ۳۹
- ۱-۳ اثبات لم ۳-۳-۱. . . . . ۵۷
- ۲-۳ زمان اجرای Orig، Bina و Forb برای مقادیر مختلف  $\theta$  و  $w$ : (آ)  $\theta = 0.174533$  و  $w = 0.1555$  . . . . . ۶۴
- ۳-۳ زمان اجرای Orig و Bina برای مقادیر مختلف  $\theta$  و  $w$  و مجموعه نقاط با  $100$ ،  $500$ ،  $1000$ ،  $2000$  و  $5000$  نقطه. (آ)  $\theta = 0.34907$  و  $w = 0.1489$  (ب)  $\theta = 0.174533$  و  $w = 0.1555$  . . . . . ۶۴

- ۴-۳ اندازه، درجه،  $weigh^*$  و زمان اجرای پوشش شکاف-حریصانه در مقایسه با پوشش حریصانه به ازای پارامترهای مختلف. (آ): اندازه.  $t = 2$ ,  $\theta = 0.349066$  و  $Weight^*$ . (ب) درجه.  $t = 1.5$ ,  $\theta = 0.174533$  و  $w = 0.72$  (ج)  $Weight^*$ . (د) زمان.  $t = 2$ ,  $\theta = 0.174533$  و  $w = 0.1489$  و  $t = 1.5$  و  $\theta = 0.34907$  و  $w = 0.1555$  . . . . . ۶۵
- ۱-۴ مسیر  $pacdeq$  یک مسیر خود-گرا از  $p$  به  $q$  است و مسیر  $pabq$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  است که خود-گرا نیست. . . . . ۷۰
- ۲-۴ (آ) یک مسیر حریصانه (ب) یک مسیر خود-گرای ضعیف که ضریب کشش آن می تواند زیاد باشد. . . . . ۷۱
- ۳-۴ خطوط  $l_{uv}$ ,  $l_{vu}$  و نیم صفحه  $l_{uv}^+$  . . . . . ۷۴
- ۴-۴ یک مجموعه نقطه‌ی محدب یک طرفه نسبت به خط  $d$ . . . . . ۷۵
- ۵-۴ اثبات قضیه‌ی ۴-۲-۴: حالتی که  $q$  بین  $A$  و  $p'$  است. . . . . ۷۹
- ۶-۴ اثبات قضیه‌ی ۴-۲-۴: حالتی که  $q$  بین  $B$  و  $p'$  است و  $i < j_1$ . . . . . ۷۹
- ۷-۴ اثبات قضیه‌ی ۴-۲-۴: حالتی که  $q$  بین  $B$  و  $p'$  است و  $i \geq j_1$  و  $j_1 \neq k + 1$ . . . . . ۸۰
- ۸-۴ یک گراف وتر-افزایشی تولید شده توسط الگوریتم ACUTETRIANGLESTEINER. توجه کنید که نقاط اشتاینری به شکل مربع‌های کوچک نمایش داده شده‌اند. . . . . ۸۱
- ۹-۴ مجموعه‌ی  $S$  با ۵ نقطه که نشان می‌دهد که گراف  $G(S, E)$  تولید شده توسط NEARESTINDISK یک گراف مسطح نیست. . . . . ۸۸
- ۱۰-۴ اثبات ادعای سوم در لم ۴-۲-۴. (آ): حالتی که  $p \in D_{ac}$ . (ب): حالتی که  $x \in D_{ac}$ . . . . . ۹۱
- ۱۱-۴ اثبات لم ۴-۲-۴. (آ): حالتی که  $p \in D_{ac}$ . (ب): حالتی که  $x \in D_{ac}$  و  $p \notin D_{ac}$ . . . . . ۹۲
- ۱۲-۴ اثبات لم ۴-۲-۳. . . . . ۹۴
- ۱۳-۴ یک مجموعه‌ای از  $n$  نقطه. برای  $k = 4$ ، گراف  $W_k$  روی این مجموعه نقطه دارای  $O(n^2)$  یال است. . . . . ۹۸

- ۱-۵ اثبات قضیه‌ی ۲-۲-۵. . . . . ۱۰۸
- ۲-۵ افراز با  $c = 5$ . . . . . ۱۱۳
- ۳-۵ ادغام یک در یک. . . . . ۱۱۵
- ۴-۵ اثبات قضیه‌ی ۵-۳-۵. . . . . ۱۱۷
- ۱-۶ مجموعه‌ی  $P$  با  $n = 7$ . . . . . ۱۲۱
- ۲-۶ اثبات قضیه‌ی ۱-۲-۶. . . . . ۱۲۲
- ۳-۶ گراف  $G_1$  با ۷ رأس. . . . . ۱۲۳

# فهرست نمادها

۲	مجموعه‌ی اعداد حقیقی $d$ بعدی	$\mathbb{R}^d$
۲	گراف $G$ با مجموعه‌ی رئوس $S$ و مجموعه‌ی یال‌های $E$	$G(S, E)$
۲	فاصله‌ی اقلیدسی بین $p$ و $q$	$ pq $
۴	وزن درخت پوشای کمینه روی مجموعه نقطه‌ی $S$	$wt(MST(S))$
۹	کوچک‌ترین جعبه‌ی احاطه‌کننده‌ی مجموعه‌ی $A$	$R(A)$
۹	تجزیه‌ی زوجی خوش-مجزا	<b>WSPD</b>
۱۱	گراف یائو با پارامتر $k$	$Y_k$
۱۲	مجموعه‌ی حاصل از حذف اعضای $S'$ از $S$	$S \setminus S'$
۱۳	گراف کامل روی مجموعه نقطه‌ی $S$	$K_S$
۱۳	خانواده‌ی همه‌ی ناحیه‌های محدب در صفحه	$C$
۱۴	گراف یائوی پیوسته با پارامتر $\theta$	$cY(\theta)$
۲۶	مثلث با رئوس $a$ ، $b$ و $c$	$\triangle abc$
۷۳	پوشش محدب نقاط $S$	$CH(S)$
۷۳	پاره‌خط واصل بین دو نقطه‌ی $x$ و $y$	$\overline{xy}$
۷۴	خطی که از $v$ می‌گذرد و عمود بر $\overline{uv}$ است	$l_{uv}$
۷۴	نیم‌صفحه‌ی بسته‌ای که مرزش $l_{uv}$ است و $u$ را شامل نمی‌شود	$l_{uv}^+$
۷۷	تهی	<b>nil</b>
۸۲	زاویه‌ی بین پاره‌خط $\overline{CA'}$ و $\overline{A'C'}$	$\angle CA'C'$
۹۰	اولین یال متقاطع با پاره‌خط $\overline{px}$	$first(\overline{px})$

فصل ۱

پیشگفتار

## ۱-۱ پوشش هندسی

فرض کنید که می‌خواهیم چند شهر را با ساخت تعدادی جاده، به هم دیگر متصل کنیم به طوری که وقتی که شخصی بخواهد روی جاده‌هایی که ساخته‌ایم، بین دو شهر رانندگی کند، طول مسیری را که طی می‌کند نسبت به طول مسیر مستقیم (جاده‌ی کاملاً مستقیم بین آن دو شهر) تفاوت زیادی نداشته باشد. در مدل گرافی این مسئله، منظور این است که مجموع طول یال‌هایی که دو رأس (دو شهر) را به هم متصل می‌کنند، دارای یک کران بالایی که تابعی از فاصله‌ی دو نقطه است، باشد. این نوع از مسائل را می‌توان با ساخت یک  $t$ -پوشش<sup>۱</sup> حل کرد.

فرض کنید که  $S$  مجموعه‌ای از نقاط در فضای  $\mathbb{R}^d$  باشد. یک گراف  $G(S, E)$  را یک **گراف هندسی** یا **شبکه‌ی هندسی** گویند، هرگاه هر یال  $e = (p, q) \in E$  بین دو رأس  $p$  و  $q$ ، پاره‌خطی واصل بین  $p$  و  $q$  باشد و وزن آن یال برابر با فاصله‌ی اقلیدسی بین  $p$  و  $q$  باشد که آن را با  $|pq|$  نمایش می‌دهیم. همچنین، طول یک مسیر  $P$  بین دو رأس در  $G$  را مجموع وزن همه‌ی یال‌های روی  $P$  تعریف می‌کنیم و با  $|P|$  نمایش می‌دهیم. حال، یک  $t$ -پوشش هندسی به صورت زیر تعریف می‌شود:

**تعریف ۱-۱-۱** ( $t$ -پوشش هندسی). فرض کنید که  $t > 1$  یک عدد حقیقی باشد. یک گراف هندسی  $G(S, E)$  را یک  $t$ -پوشش هندسی برای  $S$  می‌نامند هرگاه برای هر دو رأس  $p, q \in S$  یک مسیر  $P$  بین  $p$  و  $q$  در  $G$  وجود داشته باشد به طوری که

$$|P| \leq t \cdot |pq|.$$

هر مسیری که دارای این شرط باشد، یک  $t$ -مسیر<sup>۲</sup> بین  $p$  و  $q$  در  $G$  نامیده می‌شود.

برای سادگی، تا انتهای این رساله، به جای  $t$ -پوشش هندسی، از  $t$ -پوشش استفاده می‌کنیم.

**تعریف ۱-۱-۲** (ضریب کشش). برای دو رأس  $p$  و  $q$  در یک گراف هندسی  $G(S, E)$ ، نسبت طول کوتاه‌ترین مسیر بین  $p$  و  $q$  در  $G$ ، به  $|pq|$  را ضریب کشش بین دو رأس  $p$  و  $q$  در  $G$  می‌نامند. بیش‌ترین ضریب کشش بین همه‌ی زوج رئوس گراف  $G$  را **ضریب کشش** گراف  $G$  می‌نامند.

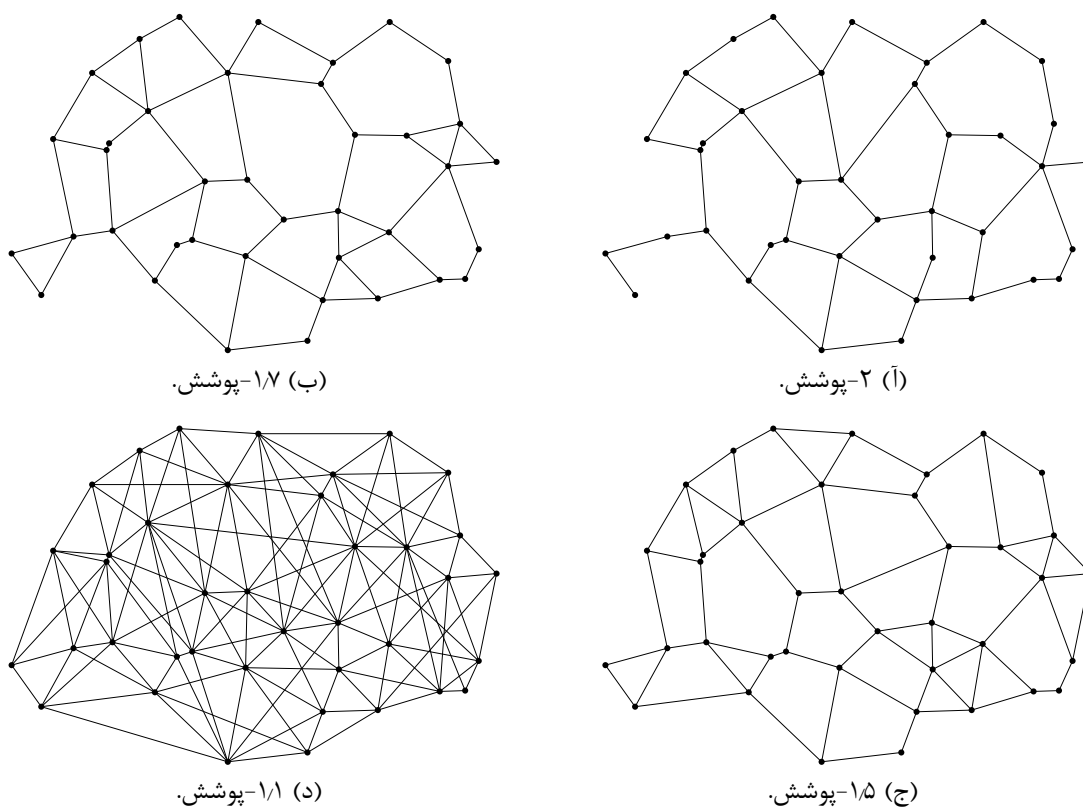
<sup>۱</sup> $t$ -spanner

<sup>۲</sup> $t$ -path



با توجه به تعریف‌های ۱-۱-۱ و ۲-۱-۱، به راحتی می‌توان گفت که ضریب کشش یک گراف هندسی  $G = (S, E)$  برابر است با کوچک‌ترین عدد حقیقی  $t > 1$  که یک  $t$ -پوشش برای  $S$  باشد.

به طور شهودی، خاصیت  $t$ -پوشش در یک گراف هندسی قوی‌تر از خاصیت همبندی در آن گراف است، زیرا در یک گراف هندسی که  $t$ -پوشش است، نه تنها بین هر دو رأس یک مسیر وجود دارد بلکه طول آن مسیر حداکثر  $t$  برابر با فاصله ی اقلیدسی بین آن دو رأس است. پارامتر  $t$  معین می‌کند که یک  $t$ -پوشش چقدر به یک گراف کامل نزدیک است. به عبارتی دیگر، هر چقدر که  $t$  به عدد یک نزدیک شود،  $t$ -پوشش (برای نقاطی که روی یک خط قرار ندارند) نیز به گراف کامل نزدیک می‌شود (شکل ۱-۱ را ببینید).



شکل ۱-۱:  $t$ -پوشش برای  $t$ های مختلف.

برای طراحی یک شبکه برای مجموعه‌ی مشخصی از نقاط، چندین معیار کیفی در نظر گرفته می‌شود. در زیر تعدادی از مهم‌ترین معیارهای کیفی برای ارزیابی شبکه‌های هندسی بیان شده است.

۱. اندازه<sup>۱</sup>: تعداد یال‌های شبکه را اندازه‌ی شبکه می‌نامند. در حالت کلی، ترجیح داده می‌شود که  $t$ -پوشش‌ها تا حد ممکن اندازه‌ی کوچکی (در حالت ایده‌آل، تعداد یال‌ها، از مرتبه‌ی خطی از تعداد رئوس است) داشته باشند.

۲. وزن<sup>۲</sup>: وزن یک گراف هندسی، برابر با مجموع وزن همه‌ی یال‌های آن گراف است. از آنجایی که هر  $t$ -پوشش باید تمام نقاط را به هم وصل کند، وزن آن از پایین، با وزن درخت پوشای کمینه<sup>۳</sup> کران‌دار می‌شود. بنابراین، بهترین وزنی که یک  $t$ -پوشش روی یک مجموعه نقطه‌ی  $S$  می‌تواند داشته باشد، به صورت مجانبی برابر وزن یک درخت پوشای کمینه روی  $S$  است. وزن درخت پوشای کمینه روی  $S$  را با  $wt(MST(S))$  نمایش می‌دهیم. وزن، معیار خوبی برای سنجش هزینه‌ی ساخت برخی شبکه‌ها نظیر شبکه‌ی راه‌ها است. بنابراین، اغلب،  $t$ -پوشش‌هایی با وزن کم مد نظر هستند. اخیراً،  $t$ -پوشش‌هایی با وزن کم در زمینه‌های مختلفی از جمله جستجوی فضای متریک<sup>۴</sup> [۴۵، ۴۶] و انتشار<sup>۵</sup> در شبکه‌های ارتباطی [۳۰]، مورد استفاده قرار گرفته‌اند.

۳. درجه<sup>۶</sup>: بیش‌ترین تعداد یال‌های مجاور به هر رأس در گراف را درجه‌ی گراف می‌نامند و اغلب نیاز است که درجه‌ی گراف با یک ثابت کوچک محدود شود. کراندار بودن درجه‌ی  $t$ -پوشش بیانگر اندازه‌ی کوچک آن است، اما، عکس این مطلب لزوماً درست نیست.

ثابت شده است که برای هر مجموعه‌ای از  $n$  نقطه در  $\mathbb{R}^d$  و برای هر عدد حقیقی  $t > 1$ ، یک  $t$ -پوشش با اندازه‌ی  $O(n)$ ، درجه‌ی  $O(1)$  و وزن  $O(wt(MST(S)))$  وجود دارد [۲۷، ۴۴].

مفهوم،  $t$ -پوشش‌ها یا به عبارتی دیگر، شبکه‌های پوششی هندسی، اولین بار توسط پلگ<sup>۷</sup> و شافر<sup>۸</sup> [۴۸] در حوزه‌ی محاسبات توزیع شده و سپس توسط چو<sup>۹</sup> [۲۳] در حوزه‌ی هندسه‌ی محاسباتی معرفی شد. تاکنون، الگوریتم‌های زیادی برای ساخت پوشش‌های هندسی با خواص مختلف نظیر اندازه‌ی کوچک، درجه‌ی کران‌دار و وزن پایین یا ترکیب برخی از این خواص، ارائه شده

است. در سال ۲۰۰۷، ناراسیمهان<sup>۱۰</sup> و اسمید<sup>۱۱</sup> [۴۴] کتابی با عنوان Geometric Spanner Networks

<sup>۱</sup> size

<sup>۲</sup> weight

<sup>۳</sup> minimum spanning tree

<sup>۴</sup> metric space searching

<sup>۵</sup> broadcasting

<sup>۶</sup> degree

<sup>۷</sup> Peleg

<sup>۸</sup> Schäffer

<sup>۹</sup> Chew

<sup>۱۰</sup> Narasimhan

<sup>۱۱</sup> Smid

منتشر کردند که تقریباً تمام الگوریتم‌های اصلی ساخت شبکه‌های پوششی هندسی را شامل می‌شود. شبکه‌های پوششی هندسی به عنوان یکی از شاخه‌های مهم هندسه محاسباتی، دارای کاربردهای زیادی هستند. از جمله این که در سال ۱۹۹۸، راثو<sup>۱</sup> و اسمیت<sup>۲</sup>، یک الگوریتم تقریبی برای حل مسئله‌ی فروشنده‌ی دوره‌گرد با کمک شبکه‌های پوششی ارائه کردند [۵۰]. در سال ۲۰۰۵، راسل<sup>۳</sup> و گییاس<sup>۴</sup>، با کمک شبکه‌های پوششی هندسی به مطالعه‌ی پروتئین‌ها پرداختند [۵۲]. در سال ۲۰۰۷، ناوارو<sup>۵</sup> و همکارانش روشی مبتنی بر  $t$ -پوشش‌ها برای جستجو در فضای متریک<sup>۶</sup> ارائه کردند [۴۷]. یکی دیگر از کاربردهای جالب شبکه‌های پوششی هندسی، تقریب درخت پوشای کمینه است. فرض کنید  $G$  یک  $t$ -پوشش برای  $S$  باشد. ثابت شده است که درخت پوشای کمینه‌ی گراف  $G$  یک  $t$ -تقریب<sup>۷</sup> برای درخت پوشای کمینه روی  $S$  است [۴۴]. به عبارتی دیگر، وزن درخت پوشای کمینه‌ی  $G$  حداکثر  $t$  برابر وزن درخت پوشای کمینه روی  $S$  است. مسئله‌ی ساخت  $t$ -پوشش‌ها با خصوصیات مختلف نظیر اندازه‌ی خطی، درجه‌ی محدود، وزن کم و یا ترکیبی از این خصوصیات، روی یک مجموعه از نقاط برای یک  $t > 1$ ، تاکنون به طور گسترده مطالعه شده است. در بخش بعدی به طور خلاصه برخی از الگوریتم‌های  $t$ -پوشش‌ها را بررسی می‌کنیم.

## ۲-۱ الگوریتم‌های ساخت $t$ -پوشش

در این بخش، به معرفی چند الگوریتم اساسی جهت ساخت  $t$ -پوشش می‌پردازیم.

### ۱-۲-۱ الگوریتم مسیر-حریصانه

یکی از معروف‌ترین الگوریتم‌های ساخت شبکه‌های پوششی هندسی، الگوریتم مسیر-حریصانه<sup>۸</sup> یا به اختصار حریصانه است (الگوریتم ۱-۲-۱ را ببینید). الگوریتم حریصانه، اولین بار توسط آلزوفر<sup>۹</sup>

<sup>۱</sup>Rao

<sup>۲</sup>Smith

<sup>۳</sup>Russel

<sup>۴</sup>Guibas

<sup>۵</sup>Navarro

<sup>۶</sup>metric

<sup>۷</sup> $t$ -approximate

<sup>۸</sup>path-greedy

<sup>۹</sup>Althöfer

و همکارانش در سال ۱۹۹۳ ارائه شد [۷]. این الگوریتم به این شکل عمل می‌کند که ابتدا تمام زوج نقاط را براساس فاصله‌ی بین آن‌ها (طول زوج نقطه) از کوچک به بزرگ مرتب می‌کند و سپس به ترتیب، آن زوج نقاط را پردازش می‌کند. الگوریتم از گراف با مجموعه یال‌های تهی شروع می‌کند. پردازش به این شکل است که الگوریتم بررسی می‌کند که آیا طول کوتاه‌ترین مسیر بین آن زوج نقطه در گراف ساخته شده (تا آن لحظه از اجرای الگوریتم) بزرگ‌تر از  $t$  برابر طول آن زوج نقطه است یا خیر. اگر بزرگ‌تر بود، آنگاه بین آن زوج نقطه، یالی اضافه می‌کند و آن یال به مجموعه‌ی یال‌های گراف، اضافه می‌شود. به گراف هندسی حاصل از اجرای این الگوریتم، پوشش حریمانه گفته می‌شود.

آنچه که واضح است این است که با کمک یک الگوریتم پیدا کردن کوتاه‌ترین مسیر در یک گراف (مثلاً الگوریتم معروف دایکسترا<sup>۱</sup>)، می‌توانیم الگوریتم حریمانه را پیاده‌سازی کنیم. ثابت شده است که پیاده‌سازی مستقیم الگوریتم حریمانه با کمک الگوریتم دایکسترا، باعث می‌شود که الگوریتم دارای پیچیدگی زمانی  $O(n^3 \log n)$  باشد [۴۴]. در سال ۲۰۱۰، بوز<sup>۲</sup> و همکارانش [۱۷] توانستند پیچیدگی زمانی الگوریتم حریمانه را به  $O(n^2 \log n)$  بهبود دهند. تا جایی که ما اطلاع داریم، این نتیجه، بهترین نتیجه‌ای است که تاکنون روی پیچیدگی زمانی الگوریتم حریمانه ارائه شده است. البته لازم به ذکر است که در سال ۲۰۱۷، آلواینز<sup>۳</sup> و همکارانش [۶] الگوریتمی جهت محاسبه‌ی پوشش حریمانه روی مجموعه‌ای از نقاط داده شده در صفحه که به طور یکنواخت توزیع شده‌اند ارائه کردند که امید ریاضی پیچیدگی زمانی آن  $O(n(\log^2 n) \log \log n)$  است.

تاکنون، کارهای تحقیقاتی زیادی بر روی این الگوریتم انجام شده است. از جمله اینکه در سال ۱۹۹۴، سوارس<sup>۴</sup> [۵۳] ثابت کرد که درجه‌ی پوشش حریمانه، کران‌دار است که این موضوع نشان می‌دهد که تعداد یال‌های پوشش حریمانه خطی است. در سال ۱۹۹۷، داس<sup>۵</sup> و ناراسیمهان [۲۷] ثابت کردند که وزن گراف پوشش حریمانه روی یک مجموعه نقطه‌ی  $S$ ، از مرتبه‌ی  $O(wt(MST(S)))$  است. اگر از نظر پیچیدگی حافظه به الگوریتم حریمانه نگاه کنیم، می‌بینیم که دارای پیچیدگی حافظه از مرتبه‌ی  $O(n^2)$  است. البته در سال ۲۰۱۵، آلواینز و همکارانش

<sup>۱</sup>Dijkstra

<sup>۲</sup>Bose

<sup>۳</sup>Alewijnse

<sup>۴</sup>Soares

<sup>۵</sup>Das

[۵]، پیچیدگی حافظه‌ی الگوریتم حریصانه را با کمک مفهوم تجزیه‌ی زوجی خوش-مجزا<sup>۱</sup> که در بخش‌های بعدی معرفی می‌شود، به  $O(n)$  بهبود دادند.

---

**Algorithm ۱-۲-1: PATHGREEDY( $S, t$ )**

---

```

/* This algorithm takes as input a set  $S$  of  $n$  points in  $\mathbb{R}^d$  and a
   real number  $t > 1$  The algorithm returns a  $t$ -spanner for  $S$ . */
1 Sort the  $\binom{n}{2}$  pairs of distinct points in non-decreasing order of their distances (ties
   are broken arbitrarily), and store them in a list  $L$ ;
2  $E := \emptyset$ ;
3  $G := (S, E)$ ;
4 foreach  $\{p, q\} \in L$  do
5    $\delta :=$ length of a shortest path in  $G$  between  $p$  and  $q$ ;
6   if  $\delta > t|pq|$  then
7      $E := E \cup \{(p, q)\}$ ;
8      $G := (S, E)$ ;
9   end
10 end
11 return  $G = (S, E)$ ;

```

---

## ۲-۲-۱ الگوریتم شکاف-حریصانه

یکی دیگر از الگوریتم‌های ساخت شبکه‌های پوششی هندسی که در کلاس الگوریتم‌های حریصانه قرار دارد، الگوریتم شکاف-حریصانه<sup>۲</sup> است که توسط آریا<sup>۳</sup> و اسمید [۹] معرفی شد (الگوریتم ۲-۲-۱ را ببینید). به گراف حاصل از این الگوریتم، پوشش شکاف-حریصانه گویند. این الگوریتم، مبتنی بر خاصیت شکاف<sup>۴</sup> است که توسط چاندرا<sup>۵</sup> و همکارانش [۲۲] معرفی شد.

فرض کنید که  $w \geq 0$  یک عدد حقیقی و  $E$  مجموعه‌ای از یال‌های جهت‌دار در فضای  $\mathbb{R}^d$  باشد. گوییم مجموعه‌ی  $E$  دارای خاصیت  $w$ -شکاف<sup>۶</sup> است هرگاه برای هر دو یال مجزای  $(p, q)$  و  $(r, s)$  در  $E$  داشته باشیم  $|pr| > \min(|pq|, |rs|)$ . گوییم مجموعه‌ی  $E$  دارای خاصیت  $w$ -شکاف قوی است هرگاه برای هر دو یال مجزای  $(p, q)$  و  $(r, s)$  در  $E$  داشته باشیم  $|pr| > \min(|pq|, |rs|)$  و  $|qs| > \min(|pq|, |rs|)$ .

خاصیت شکاف، خاصیت مفیدی برای تحلیل وزن یک شبکه‌ی پوششی است. چاندرا و

---

<sup>۱</sup>well-separated pair decomposition

<sup>۲</sup>gap-greedy

<sup>۳</sup>Arya

<sup>۴</sup>gap

<sup>۵</sup>Chandra

<sup>۶</sup> $w$ -gap

همکارانش در [۲۲] ثابت کردند که در هر گراف  $G(S, E)$  که  $E$  خاصیت شکاف را داراست، درجه‌ی خروجی هر رأس در  $G$  حداکثر یک و وزن  $G$ ، از مرتبه‌ی  $O(\log n \cdot wt(MST(S)))$  است. آن‌ها همچنین نشان دادند که در هر گراف  $G(S, E)$  که  $E$  خاصیت شکاف قوی را داراست، درجه‌ی ورودی هر رأس در  $G$  نیز حداکثر یک است.

الگوریتم شکاف-حریصانه به این صورت عمل می‌کند که ابتدا همه‌ی زوج نقاط را براساس طولشان از کوچک به بزرگ مرتب می‌کند، سپس این زوج نقاط به ترتیب پردازش می‌شوند. فرض کنید که  $p$  و  $q$  زوج نقطه‌ای باشد که الگوریتم قصد پردازش آن‌ها را دارد. الگوریتم، یال  $(p, q)$  را به مجموعه‌ی یال‌ها اضافه می‌کند اگر و فقط اگر تا آن لحظه از اجرای الگوریتم، هیچ یال  $(r, s)$  در مجموعه‌ی یال‌های گراف ساخته شده، وجود نداشته باشد به طوری که الف)  $(p, q)$  و  $(r, s)$  تقریباً هم‌جهت باشند ب) حداقل یکی از فاصله‌های  $|pq|$  و  $|rs|$  کوچک باشد (برای مشاهده‌ی جزئیات بیش‌تر الگوریتم، بخش ۱-۳ را ببینید). اگر چنین یال  $(r, s)$  وجود داشت، آنگاه یال  $(p, q)$  به مجموعه‌ی یال‌ها اضافه نمی‌شود در غیر اینصورت، اضافه می‌شود. چاندرا و همکارانش در همان مقاله، ثابت کردند که الگوریتم شکاف-حریصانه، یک شبکه‌ی پوششی هندسی با ضریب کشش ثابت و از درجه‌ی کران‌دار تولید می‌کند که می‌توان مجموعه‌ی یال‌ها را به تعداد ثابتی زیرمجموعه افزایش کرد به طوری که هر زیرمجموعه خاصیت شکاف را دارا باشد و در نتیجه وزن آن از مرتبه‌ی  $O(\log n \cdot wt(MST(S)))$  می‌باشد.

الگوریتم شکاف-حریصانه، در پیاده‌سازی مستقیم دارای پیچیدگی زمانی  $O(n^3)$  و پیچیدگی حافظه‌ی  $O(n^2)$  است (الگوریتم ۱-۲-۲ را ببینید). برطبق آنچه که می‌دانیم، هنوز هیچ بهبودی روی زمان و حافظه‌ی این الگوریتم وجود ندارد و این مسئله، همچنان حل نشده است. البته آریا و اسمید [۹]، الگوریتمی مشابه الگوریتم شکاف-حریصانه ارائه کردند که دارای پیچیدگی زمانی  $O(n \log^2 n)$  است و گراف حاصل، خواصش مشابه خواص گراف حاصل از الگوریتم شکاف-حریصانه

است اما لزوماً همان گراف حاصل از الگوریتم شکاف-حریصانه را تولید نمی‌کند.

---

**Algorithm ۱-۲-2:** GAPGREEDY( $S, \theta, w$ )

---

```

/* This algorithm takes as input a set  $S$  of  $n$  points in the
plane, and two real numbers  $\theta$  and  $w$  such that  $0 < \theta < \pi/4$ ,
 $0 \leq w < (\cos \theta - \sin \theta)/2$ . The algorithm returns a directed
 $t$ -spanner  $G = (S, E)$ , for  $t = 1/(\cos \theta - \sin \theta - 2w)$ . */
1 Sort the  $2 \binom{n}{2}$  ordered pairs of distinct points in non-decreasing order of their
distances (ties are broken arbitrarily), and store them in a list  $L$ ;
2  $E := \emptyset$ ;
3 foreach ordered pair  $(p, q) \in L$  do
4    $add := true$ ;
5   foreach edge  $(r, s) \in E$  do
6     /* Let  $\vec{a}$  be a vector from the origin to the point  $a$ . We
define  $angle(pq, rs)$  to be the angle between the vectors
 $\vec{q-p}$  and  $\vec{s-r}$ . */
7     if  $angle(pq, rs) \leq \theta$  then
8        $add := add \wedge (|pr| > w|rs|) \wedge (|qs| > w|rs|)$ ;
9     end
10    if  $add = true$  then
11       $E := E \cup (p, q)$ ;
12    end
13 end
14 return  $G = (S, E)$ ;

```

---

### ۳-۲-۱ WSPD-پوشش

در سال ۱۹۹۵، کالاهان<sup>۱</sup> و کاساراجو<sup>۲</sup> [۱۰] ساختمان داده‌ای به نام تجزیه‌ی زوجی خوش-مجزا (WSPD)، را ارائه کردند. این ساختمان داده، یکی از مهم‌ترین ساختمان داده‌هایی است که می‌توان از آن جهت حل مسائل مجاورتی<sup>۳</sup> استفاده کرد. قبل از اینکه این ساختمان داده را معرفی نماییم، ابتدا مفهوم زوج خوش-مجزا را معرفی می‌کنیم.

فرض کنید که  $A$  یک مجموعه از نقاط در فضای  $\mathbb{R}^d$  و  $R(A)$  کوچک‌ترین جعبه‌ی احاطه‌کننده<sup>۴</sup>ی مجموعه‌ی  $A$  باشد.

**تعریف ۱-۲-۱** (زوج خوش-مجزا). فرض کنید که  $s > 0$  یک عدد حقیقی و  $A$  و  $B$  دو مجموعه‌ی

---

<sup>۱</sup>Callahan  
<sup>۲</sup>Kosaraju

<sup>۳</sup>proximity problems  
<sup>۴</sup>bounding box

متناهی در فضای  $\mathbb{R}^d$  باشند. می‌گوییم که  $A$  و  $B$  نسبت به  $s$ ، خوش-مجزا هستند هرگاه توپ<sup>۱</sup>های مجزای  $d$ -بعدی  $C_A$  و  $C_B$  وجود داشته باشند به طوری که

۱. توپ‌های  $C_A$  و  $C_B$  دارای شعاع‌های یکسانی باشند.

۲. توپ  $C_A$ ، شامل جعبه‌ی احاطه‌گر  $R(A)$  باشد.

۳. توپ  $C_B$ ، شامل جعبه‌ی احاطه‌گر  $R(B)$  باشد.

۴. فاصله‌ی بین  $C_A$  و  $C_B$  بزرگ‌تر یا مساوی  $s$  برابر شعاع  $C_A$  باشد.

**تعریف ۱-۲-۲** (تجزیه‌ی زوجی خوش-مجزا). فرض کنید  $S$  یک مجموعه از  $n$  نقطه در فضای  $\mathbb{R}^d$  باشد. همچنین فرض کنید  $s > 0$  یک عدد حقیقی باشد. یک تجزیه‌ی زوجی خوش-مجزا برای  $S$  نسبت به  $s$ ، یک دنباله به صورت

$$\{A_1, B_1\}, \{A_2, B_2\}, \dots, \{A_m, B_m\}$$

از زیرمجموعه‌های غیرتهی در  $S$  است به طوری که

۱. برای هر  $i$  با  $1 \leq i \leq m$ ،  $A_i$  و  $B_i$  نسبت به  $s$  زوج خوش-مجزا باشند و

۲. برای هر دو نقطه‌ی متمایز  $p$  و  $q$  در  $S$ ، دقیقاً یک اندیس  $i$  با  $1 \leq i \leq m$  وجود داشته باشد

به طوری که  $p \in A_i$  و  $q \in B_i$  یا  $q \in A_i$  و  $p \in B_i$ .

عدد صحیح  $m$  را اندازه‌ی تجزیه‌ی زوجی خوش-مجزا و  $s$  را ضریب تفکیک<sup>۲</sup> می‌نامند.

کالاها و کاساراجو در همان مقاله، ثابت کردند که برای هر مجموعه‌ی  $S$  از  $n$  نقطه در فضای  $\mathbb{R}^d$  و برای یک  $d$  ثابت و یک  $s$  ثابت می‌توان یک تجزیه‌ی زوجی خوش-مجزا با تعداد خطی زوج در زمان  $O(n \log n)$  ساخت.

کالاها و کاساراجو در همان مقاله، نشان دادند که با کمک نتیجه‌ی بالا، می‌توان برای مجموعه‌ی  $S$  یک شبکه‌ی پوششی هندسی با ضریب کشش دلخواه  $t > 1$  و با تعداد خطی یال، در

<sup>۱</sup>Ball

<sup>۲</sup>separation ratio



زمان  $O(n \log n)$  ساخت. روش آن‌ها به این صورت است که ابتدا برای مجموعه‌ی  $S$  یک تجزیه‌ی زوجی خوش-مجزا به صورت  $\{(A_i, B_i)\}_{i=1}^m$  با  $m = O(n)$  و با ضریب تفکیک  $s = \frac{t(t+1)}{t-1}$  می‌سازد. سپس، برای هر زوج  $\{A_i, B_i\}$ ، یک نقطه‌ی دلخواه در  $A_i$  و یک نقطه‌ی دلخواه در  $B_i$  انتخاب می‌کند و در نهایت آن نقاط دلخواه را با یک یال به هم وصل می‌کند. کالاهان و کاساراجو در همان مقاله ثابت کردند که گراف حاصل از الگوریتم که آن را WSPD-پوشش می‌نامیم، یک شبکه‌ی  $t$ -پوشش با تعداد خطی یال است. لازم به ذکر است که حداقل زمان لازم برای ساخت یک  $t$ -پوشش روی یک مجموعه نقطه‌ی  $S$  با  $n$  نقطه در فضای  $\mathbb{R}^d$ ،  $\Omega(n \log n)$  است [۳۲]. واضح است که الگوریتم ساخت WSPD-پوشش از نظر زمان، بهینه است.

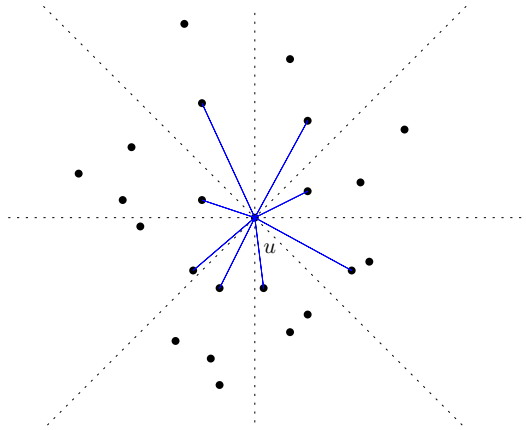
#### ۴-۲-۱ گراف یائو و $\Theta$ -گراف

یکی دیگر از روش‌های ساخت یک  $t$ -پوشش برای یک مجموعه نقطه‌ی داده شده  $S$  و یک عدد ثابت  $t > 1$ ، استفاده از **گراف یائو**<sup>۱</sup> است. گراف یائو اولین بار توسط یائو<sup>۲</sup> [۵۶] برای ساخت درخت پوشای کمینه در فضای اقلیدسی و در ابعاد بالا معرفی شد. برای یک عدد طبیعی  $k \geq 2$ ، گراف یائوی  $Y_k$  برای مجموعه‌ی  $S$  به صورت زیر ساخته می‌شود: در هر نقطه‌ی  $u$  در  $S$ ،  $k$  مخروط به مرکز  $u$  و با دهانه‌ی<sup>۳</sup>  $\frac{2\pi}{k}$  رسم می‌کنیم برای هر نقطه‌ی  $u$  در  $S$  و هر مخروط  $C_u$ ، یک یال بین  $u$  و نزدیک‌ترین نقطه (یکی از نزدیک‌ترین نقاط) به  $u$  در  $C_u$  وصل می‌کنیم (شکل ۲-۱ را ببینید). اگر عدد  $k$  به اندازه‌ی کافی بزرگ انتخاب شود، می‌توان ثابت کرد که گراف یائوی  $Y_k$  یک  $t$ -پوشش است که این  $t$  فقط وابسته به مقدار  $k$  است [۱۸].

در سال ۱۹۸۷، کلارکسون<sup>۴</sup> [۲۴] گرافی مشابه گراف یائو، که  $\Theta$ -گراف نامیده می‌شود، ارائه کرد. تفاوتی که بین ساخت گراف یائو و  $\Theta$ -گراف هست این است که اولاً یک خط دلخواه  $\ell$  را که از مرکز مخروط به سمت داخل مخروط می‌گذرد را در نظر می‌گیریم و دوماً در هر مخروط، رأس  $u$  را به نقطه‌ای در مخروط  $C_u$  وصل می‌کنیم که فاصله‌ی تصویر عمودش روی  $\ell$  تا  $u$  کم‌ترین باشد.

<sup>۱</sup>Yao graph  
<sup>۲</sup>Yao

<sup>۳</sup>aperture  
<sup>۴</sup>Clarkson



شکل ۲-۱: نقطه‌ی  $u$  به نزدیک‌ترین نقطه در هر مخروط وصل می‌شود.

الگوریتمی با زمان اجرای  $O(n \log^{O(1)} n)$  برای ساخت  $\Theta$ -گراف روی مجموعه‌ای از نقاط در فضای  $\mathbb{R}^d$  که  $d > 2$  است، ارائه شده است [۴۴] اما تاکنون برای ساخت گراف یائو روی این مجموعه از نقاط، الگوریتم با زمان اجرای  $O(n \log^{O(1)} n)$  ارائه نشده است.

برای  $\Theta$ -گراف همانند گراف یائو وقتی عدد  $k$  به اندازه‌ی کافی بزرگ انتخاب شود، می‌توان ثابت کرد که  $\Theta$ -گراف، با پارامتر  $k = \frac{2\pi}{\theta}$ ، یک  $t$ -پوشش است که این  $t$  فقط وابسته به مقدار  $k$  است [۴۴]. واضح است که هم گراف یائو و هم  $\Theta$ -گراف هر دو دارای اندازه‌ی خطی هستند، اما می‌توان مثال‌هایی ارائه داد که نشان می‌دهند که درجه‌ی گراف یائو و  $\Theta$ -گراف، کراندار نیست [۴۴].

### ۵-۲-۱ $t$ -پوشش تحمل‌پذیر ناحیه-خطا

یکی از مباحث مهم در ساخت شبکه‌های هندسی، تحمل‌پذیری آن شبکه در برابر حذف تعدادی رأس یا یال از آن شبکه هست. لوکوپولوس<sup>۱</sup> و همکارانش [۳۹] مفهوم  $t$ -پوشش تحمل‌پذیر خطای  $k$ -رأسی<sup>۲</sup> را به صورت زیر تعریف کردند. گراف  $G = (S, E)$  را یک  $t$ -پوشش تحمل‌پذیر خطای  $k$ -رأسی یا به اختصار  $(k, t)$ -VFTS (برای عدد حقیقی  $1 < t$  و عدد صحیح غیرمنفی  $k$ ) گویند هرگاه برای هر زیرمجموعه‌ی  $S'$  از  $S$  با اندازه‌ی حداکثر  $k$ ، گراف  $G \setminus S'$  برای  $S \setminus S'$  یک  $t$ -پوشش باشد. همچنین  $G$  را یک  $t$ -پوشش تحمل‌پذیر خطای  $k$ -یالی<sup>۳</sup> و یا به اختصار  $(k, t)$ -EFTS

<sup>۱</sup>Levcopoulos

<sup>۲</sup> $k$ -vertex fault tolerant  $t$ -spanner

<sup>۳</sup> $k$ -edge fault tolerant  $t$ -spanner

گویند هرگاه برای هر زیرمجموعه‌ی  $E'$  از  $E$  با اندازه‌ی حداکثر  $k$  و برای هر  $p$  و  $q$  در  $S$ ، گراف  $G \setminus E'$  شامل یک مسیر بین  $p$  و  $q$  با طول حداکثر  $t$  برابر طول کوتاه‌ترین مسیر بین  $p$  و  $q$  در گراف  $K_S \setminus E'$  باشد، که  $K_S$  گراف کامل روی  $S$  است. لوکوپولوس و همکارانش در همان مقاله، برای اولین بار سه الگوریتم برای ساخت شبکه‌های  $(k, t)$ -VFTS ارائه کردند. تعدادی کار تحقیقاتی دیگر نیز در همین زمینه انجام شده است [۲۵، ۴۲]. سوالی که ممکن است پرسیده شود این است که آیا می‌توان روی یک مجموعه نقطه‌ی داده شده، یک پوشش هندسی ساخت به طوری که با حذف یک ناحیه از آن، گراف حاصل باز هم یک پوشش هندسی باشد؟

در سال ۲۰۰۹، آبام<sup>۱</sup> و همکارانش [۲] مفهوم پوشش‌های تحمل‌پذیر ناحیه-خط<sup>۲</sup> را برای نقاط در صفحه معرفی کردند. برای یک ناحیه‌ی  $F$  در صفحه، فرض کنید  $G \ominus F$  گراف حاصل از حذف همه‌ی رئوس  $G$  که داخل ناحیه‌ی  $F$  قرار می‌گیرند و حذف همه یال‌های  $G$  که با ناحیه‌ی  $F$  اشتراک دارند، باشد. فرض کنید  $\mathcal{F}$  مجموعه‌ای از ناحیه‌ها در صفحه باشد. می‌گوییم که  $G$  یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{F}$ -خط است اگر برای هر  $F \in \mathcal{F}$ ، طول کوتاه‌ترین مسیر بین هر دو رأس  $p$  و  $q$  در گراف  $G \ominus F$  حداکثر  $t$  برابر طول کوتاه‌ترین مسیر بین  $p$  و  $q$  در گراف  $K_S \ominus F$  باشد که در اینجا  $K_S$  گراف کامل هندسی روی  $S$  است. آبام و همکارانش در همان مقاله، نشان دادند که برای هر مجموعه‌ای از  $n$  نقطه در صفحه و هر خانواده‌ی  $\mathcal{C}$  از ناحیه‌های محدب<sup>۳</sup>، می‌توان یک پوشش تحمل‌پذیر  $\mathcal{C}$ -خط با  $O(n \log n)$  یال را در زمان  $O(n \log^2 n)$  ساخت.

## ۱-۲-۶ گراف‌های خود-گرا

در سال ۲۰۱۳، علمداری<sup>۴</sup> و همکارانش [۳] مفهوم رسم خود-گرا و رسم وتر-افزایشی را معرفی کردند. فرض کنید که  $G$  یک گراف هندسی با مجموعه‌ی رئوس  $S$  باشد. برای هر دو رأس  $s, t \in S$ ، یک مسیر از  $s$  به  $t$  در  $G$  را خود-گرا<sup>۵</sup> گویند، هرگاه برای هر نقطه‌ی  $q$  روی مسیر (نه ضرورتاً یک رأس)، یک نقطه‌ای که به طور پیوسته روی مسیر از  $s$  به  $q$  در حال حرکت است، هرگز در حرکتش

<sup>۱</sup>Abam

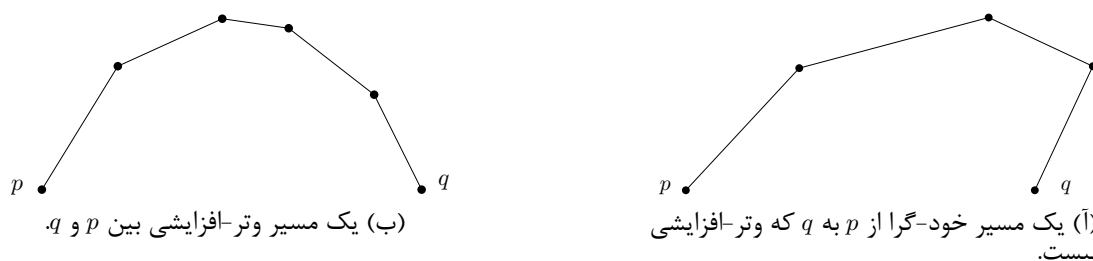
<sup>۲</sup>region-fault tolerant spanners

<sup>۳</sup>convex regions

<sup>۴</sup>Alamdari

<sup>۵</sup>self-approaching

از  $q$  دور نشود. گراف هندسی  $G$  را خود-گرا گویند هرگاه برای هر دو رأس  $s$  و  $t$  در  $G$ ، یک مسیر خود-گرا از  $s$  به  $t$  در  $G$  وجود داشته باشد (شکل ۱-۳ را ببینید). یک مسیر را وتر-افزایشی<sup>۱</sup> گویند اگر آن مسیر در هر دو جهت (مبدأ به مقصد و مقصد به مبدأ) خود-گرا باشد و یک گراف هندسی را وتر-افزایشی گویند هرگاه بین هر دو رأسش یک مسیر وتر-افزایشی وجود داشته باشد (شکل ۱-۳ ب را ببینید).



شکل ۱-۳: مسیر خود-گرا و وتر-افزایشی.

در زمینه‌ی ساخت گراف‌های خود-گرا بر روی یک مجموعه از نقاط در صفحه تعدادی کار تحقیقاتی انجام شده است [۳، ۲۸، ۴۳]. یکی از دلایل توجه زیاد به این نوع گراف‌ها در کارهای تحقیقاتی این است که این نوع گراف‌ها دارای ضریب کشش خوبی هستند. ثابت شده است که ضریب کشش گراف‌های خود-گرا حداکثر  $۵/۳۳۳۲$  [۳۶] و ضریب کشش گراف‌های وتر-افزایشی حداکثر  $۲/۰۹۴$  است [۵۱].

### ۳-۱ کلیات رساله

در این رساله، ما چند مسئله‌ی مرتبط با طراحی و تحلیل شبکه‌های هندسی را در نظر می‌گیریم: در فصل ۲، نوع جدیدی از گراف‌های هندسی به نام گراف یائوی پیوسته را معرفی می‌کنیم. فرض کنید که  $0 < \theta \leq 2\pi$  یک عدد حقیقی باشد. گراف یائوی پیوسته نسبت به زاویه‌ی  $\theta$  که با نماد  $cY(\theta)$  نمایش داده می‌شود روی مجموعه نقطه‌ی  $S$  به صورت زیر ساخته می‌شود: بین هر دو رأس  $p, q \in S$ ، یالی وجود دارد اگر و فقط اگر یک مخروط با دهانه‌ی  $\theta$  به مرکز  $p$  وجود داشته باشد به طوری که  $q$  نزدیک‌ترین نقطه به  $p$  در این مخروط باشد. به عبارتی دیگر، تصور کنید که

<sup>۱</sup>increasing-chord

یک مخروط با دهانه  $\theta$  حول یک نقطه‌ی  $p \in S$  می‌چرخد. بنابراین در گراف یائوی پیوسته،  $p$  همواره توسط یالی به نزدیک‌ترین نقطه‌ی داخل مخروط در حین چرخش متصل می‌شود. در این فصل، ضریب کشش، تحمل‌پذیری خطا و خاصیت خود-گرایی این نوع گراف‌ها را بررسی می‌کنیم. ما نشان می‌دهیم که برای هر  $0 < \theta \leq 2\pi/3$ ، گراف یائوی پیوسته‌ی  $cY(\theta)$  یک  $t$ -پوشش است که  $t$  یک مقدار ثابت می‌باشد، ولی برای  $\pi \leq \theta \leq 2\pi$  این گونه نیست. همچنین، نشان می‌دهیم که برای هر  $0 < \theta \leq \pi$ ، گراف  $cY(\theta)$  همبند است ولی برای هر  $\theta > \pi$ ، لزوماً همبند نیست. همچنین این خصوصیات را در مدل تحمل‌پذیری ناحیه-خطا نیز بررسی می‌کنیم و نشان می‌دهیم که برای هر  $\theta < \pi/3$ ، گراف  $cY(\theta)$  یک  $t$ -پوشش باقی می‌ماند ( $t$  مقدار ثابت و وابسته به  $\theta$  است) حتی اگر ناحیه‌ی محدبی از آن حذف شود. ما همچنین نشان می‌دهیم که  $cY(\theta)$  برای هر  $\theta \leq \pi$  همبند باقی می‌ماند حتی اگر بخش محدبی از آن حذف شود. این فصل، مستخرج از مقاله‌ی [۱۱] است.

در فصل ۳، یک الگوریتم برای ساخت پوشش شکاف-حریصانه با کمک WSPD ارائه می‌کنیم که پیچیدگی زمانی و حافظه‌ی آن به ترتیب  $O(n^3)$  و  $O(n)$  است. اگرچه، پوشش حریصانه یک پوشش با کیفیت عالی است و همچنین وزن آن از دیدگاه نظری بهتر از وزن پوشش شکاف-حریصانه است، اما تا جایی که ما اطلاع داریم هیچ الگوریتمی برای ساخت پوشش حریصانه در زمان  $O(n^2)$  پیش از این ارائه نشده است. در این فصل، نشان می‌دهیم که الگوریتمی برای ساخت پوشش شکاف-حریصانه در زمان  $O(n^2)$  و با بکاربردن  $O(n^2)$  حافظه وجود دارد. همچنین، ضریب کشش پوشش شکاف-حریصانه را بهبود می‌دهیم. در این فصل، همچنین سه الگوریتم با زمان اجرای  $O(n^3)$  برای ساخت پوشش شکاف-حریصانه ارائه می‌کنیم و چندین نتیجه‌ی تجربی روی زمان اجرای این الگوریتم‌ها ارائه می‌کنیم. با توجه به نتایج تجربی، نتیجه می‌گیریم که یکی از این الگوریتم‌ها (الگوریتم ۳-۴-۲) نسبت به الگوریتم‌های دیگری که معرفی می‌شوند، سریع‌تر است. ما همچنین به صورت تجربی خصوصیات هندسی پوشش حریصانه را در مقایسه با پوشش شکاف-حریصانه، مورد بررسی قرار می‌دهیم. این فصل، مستخرج از مقاله‌ی [۱۵] است.

در فصل ۴، مفهوم جدیدی به نام **گراف‌های خود-گرای ضعیف** که حالت تضعیف شده‌ی گراف‌های خود-گرا است را معرفی می‌کنیم. ما یک الگوریتم ارائه می‌کنیم که در زمان چندجمله‌ای

تشخیص می‌دهد که آیا یک گراف هندسی داده شده، خود-گرای ضعیف است یا خیر. همچنین، مسئله‌ی ساخت یک گراف خود-گرای ضعیف مسطح برای هر مجموعه نقطه‌ای در صفحه و مسئله‌ی ساخت  $t$ -پوشش خود-گرای ضعیف (یک  $t$ -پوشش که بین هر دو رأس آن یک  $t$ -مسیر خود-گرای ضعیف وجود دارد) برای هر مجموعه نقطه در صفحه را بررسی می‌کنیم. به طور خاص، نشان می‌دهیم که برای هر عدد حقیقی  $t > 1$  و هر مجموعه نقطه‌ی  $S$  که روی یک مثلث توزیع شده است، یک  $t$ -پوشش وتر-افزایشی با  $O(kn)$  یال که  $k$  فقط وابسته به  $t$  است، وجود دارد. همچنین در این فصل، مسئله‌ی حل نشده‌ای که توسط دهکردی و همکارانش [۴۹] مطرح شد را مورد بحث قرار می‌دهیم. به طور خاص، ما نشان می‌دهیم که یک گراف وتر-افزایشی مسطح برای هر مجموعه نقطه‌ای که روی یک مثلث منفرجه یا قائم‌الزاویه قرار گرفته است، وجود دارد. همچنین، نشان می‌دهیم که برای هر مجموعه نقطه‌ای که روی یک مثلث حاده قرار دارد، یک گراف وتر-افزایشی مسطح با اضافه کردن حداکثر ۳ نقطه‌ی اشتاینری وجود دارد. این فصل، مستخرج از مقاله‌ی [۱۲] است.

در فصل ۵، مسئله‌ی ادغام دو پوشش هندسی را معرفی می‌کنیم و سپس یک الگوریتم برای حل آن ارائه می‌کنیم. تا جایی که ما اطلاع داریم، هنوز هیچ الگوریتم تقسیم و غلبه‌ای برای ساخت یک پوشش هندسی ارائه نشده است. بنابراین، ما برای اولین بار، با استفاده از مفهوم ادغام دو پوشش هندسی، دو الگوریتم تقسیم و غلبه جهت ساخت یک  $t$ -پوشش برای یک مجموعه نقطه‌ی داده شده در صفحه ارائه می‌کنیم. همچنین، یک الگوریتم تقسیم و غلبه برای ساخت یک  $t$ -پوشش تحمل‌پذیر ناحیه-خطا برای یک مجموعه از نقاط داده شده و برای یک‌سری از ناحیه‌های خاص در صفحه، ارائه می‌کنیم. این فصل، مستخرج از مقاله‌ی [۱۴] است.

در فصل ۶، مسئله‌ی ساخت پوشش‌های هندسی مقید به زاویه<sup>۱</sup> را روی یک مجموعه نقطه در صفحه در نظر می‌گیریم. فرض کنید که  $\theta$  یک عدد حقیقی مثبت باشد. یک گراف هندسی  $G(S, E)$  را مقید به زاویه‌ی  $\theta$  گویند هرگاه زاویه‌ی بین هر دو یال مجزای  $\{p, q\}$  و  $\{p, r\}$  در  $E$  حداقل  $\theta$  باشد. برای اولین بار در سال ۲۰۱۲، کارمی<sup>۲</sup> و اسمید [۲۱] مسئله‌ی ساخت شبکه‌ی پوششی هندسی مقید به زاویه را بررسی کردند. آن‌ها نشان دادند که برای هر مجموعه نقطه‌ی  $S$

<sup>۱</sup>angle-constrained

<sup>۲</sup>Carmi

شامل  $n$  نقطه در فضای اقلیدسی  $\mathbb{R}^d$  و برای هر زاویه  $\theta$  با  $0 < \theta < \pi/3$ ، یک  $t$ -پوشش مقید به زاویه  $\theta$  وجود دارد که در زمان  $O(n \log n)$  ساخته می‌شود که  $t$  فقط به  $\theta$  وابسته است. آن‌ها در همان مقاله، مسئله‌ی ساخت شبکه‌ی پوششی هندسی مقید به زاویه  $\theta$  با  $\theta \geq \pi/3$  را به عنوان مسئله‌ی حل نشده مطرح کردند. در این فصل، ما این مسئله‌ی حل نشده را حل می‌کنیم. به طور خاص، نشان می‌دهیم که یک مجموعه‌ی  $P$  شامل  $n$  نقطه در صفحه وجود دارد به طوری که برای هر  $\theta > \pi/3$  هیچ گراف هندسی همبند مقید به زاویه  $\theta$  با مجموعه‌ی رئوس  $P$  وجود ندارد. همچنین نشان می‌دهیم که برای  $\theta = \pi/3$  و برای هر  $t < 2/\sqrt{3}$ ، هیچ  $t$ -پوشش مقید به زاویه  $\theta$  روی  $P$  وجود ندارد. این فصل، مستخرج از مقاله‌ی [۱۳] است.

در فصل ۶-۳، نتیجه‌گیری و مسائل حل نشده را بیان می‌کنیم.





## فصل ۲

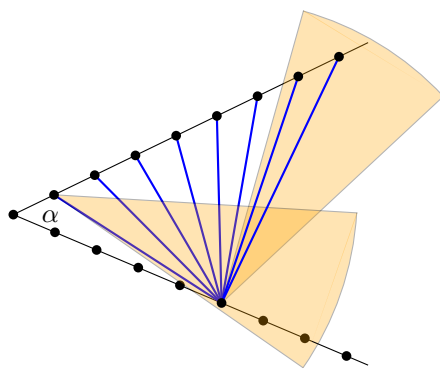
### گراف یائوی پیوسته

در فصل قبل، گراف یائو را تعریف کردیم. مجدداً این تعریف را در اینجا بیان می‌کنیم. فرض کنید که  $S$  مجموعه‌ای شامل  $n$  نقطه در صفحه باشد. یک روش ساده‌ی ساخت  $t$ -پوشش این است که ابتدا صفحه را حول هر نقطه‌ی  $p \in S$  به تعداد ثابتی مخروط<sup>۱</sup> افزاز نماییم (لازم به ذکر است که مرز اولین مخروط در جهت ساعت‌گرد، مماس با محور  $x$ ها است) و سپس یال  $(p, r)$ ، که  $r$  نزدیک‌ترین نقطه به  $p$  در مخروط است را به گرافی که تاکنون ساخته شده، اضافه کنیم. این نوع گراف‌ها، به گراف‌های یائو معروف هستند که مستقلاً توسط فلینچباک<sup>۲</sup> و همکارش [۳۳] و یائو [۵۶] معرفی شدند. ثابت شده است که گراف‌های یائو، گراف‌های هندسی کامل را به خوبی تقریب می‌زنند [۷، ۱۶، ۱۸، ۲۰، ۲۴، ۲۶].

گراف یائو را با  $Y_k$  نمایش می‌دهیم که  $k$  تعداد مخروط‌ها است و هر مخروط دارای دهانه‌ی  $\theta = 2\pi/k$  است. برای اولین بار در سال ۱۹۸۷، کلارکسون [۲۴] نشان داد که  $Y_{12}$  یک  $(1 + \sqrt{3})$ -پوشش است. آلزوفر و همکارانش [۷] نشان دادند که برای هر  $t > 1$ ، یک  $k$  وجود دارد که  $Y_k$  یک  $t$ -پوشش است. برای  $k > 8$ ، بوز و همکارانش [۲۰] نشان دادند که  $Y_k$  یک شبکه‌ی پوششی با ضریب کشش حداکثر  $1/(\cos \theta - \sin \theta)$  است که بعدها بوز و همکارانش ثابت کردند که برای  $k > 6$ ، گراف  $Y_k$ ، یک شبکه‌ی پوششی با ضریب کشش حداکثر  $1/(1 - 2 \sin(\theta/2))$  است [۱۸]. دامیان<sup>۳</sup> و رادونیس<sup>۴</sup> [۲۶] ثابت کردند که ضریب کشش  $Y_6$ ،  $17/64$  است که بعدها توسط باربا<sup>۵</sup> و همکارانش [۱۶] این ضریب کشش به  $5/8$  بهبود داده شد. در همان مقاله، همچنین ضریب کشش  $Y_k$  برای مقادیر  $k$  فرد که  $k \geq 5$  به  $1/(1 - 2 \sin(3\theta/8))$  بهبود داده شد. بوز و همکارانش [۱۸] نشان دادند که  $Y_4$  یک  $663$ -پوشش است. برای  $k < 4$ ، ال ملا<sup>۶</sup> [۲۹] نشان داد که هیچ عدد ثابت  $t$  وجود ندارد که به ازای آن،  $Y_k$  یک  $t$ -پوشش شود.

گراف‌های یائو مبتنی بر این فرض اولیه هستند که همه‌ی نقاط یک چرخش مخروط یکسان با توجه به یک جهت ثابت خارجی را بکار می‌برند. از دیدگاه کاربردی، اگر این نقاط دستگاه‌های

<sup>۱</sup> cone<sup>۲</sup> Flinchbaugh<sup>۳</sup> Damian<sup>۴</sup> Raudonis<sup>۵</sup> Barba<sup>۶</sup> El Molla



شکل ۱-۲: تعداد یال‌های گراف یائوی پیوسته می‌تواند مربعی باشد.

بسیم را نمایش دهند و یال‌ها راه‌های ارتباطی بین آنها را نمایش دهند، آنگاه این نقاط برای اینکه قادر باشند تا مخروط‌هایشان را به صورت یکسان بچرخانند، نیازمند اشتراک گذاری یک دستگاه مختصات یکسان هستند. عدم وجود اطلاعات مختصاتی یکسان، سطحی از مشکلات را به شبکه وارد می‌کند. در این رساله، ما نوع جدیدی از گراف‌های یائو به نام گراف‌های یائوی پیوسته، را از دیدگاه چرخش مخروط‌ها به طور داخلی، بررسی می‌کنیم.

فرض کنید که  $0 < \theta \leq 2\pi$  یک عدد حقیقی باشد. گراف یائوی پیوسته نسبت به زاویه  $\theta$  که با نماد  $cY(\theta)$  نمایش داده می‌شود روی مجموعه نقطه‌ای  $S$  به صورت زیر ساخته می‌شود: بین هر دو رأس  $p, q \in S$  یالی وجود دارد اگر و فقط اگر یک مخروط با دهانه‌ی  $\theta$  به مرکز  $p$  وجود داشته باشد به طوری که  $q$  نزدیک‌ترین نقطه به  $p$  در این مخروط باشد. به عبارتی دیگر، تصور کنید که یک مخروط با دهانه‌ی  $\theta$  حول یک نقطه‌ی  $p \in S$  می‌چرخد. در گراف یائوی پیوسته،  $p$  به نزدیک‌ترین نقطه‌ی داخل مخروط در حین چرخش، متصل می‌شود.

برخلاف گراف یائو، گراف یائوی پیوسته دارای این خاصیت است که به ازای هر  $\theta$  و هر  $\gamma$  که  $cY(\theta) \subseteq cY(\gamma)$ ،  $\theta \geq \gamma$ ، همچنین، برخلاف گراف‌های یائو که همواره تعداد یال‌های آن‌ها خطی است، گراف یائوی پیوسته ممکن است شامل تعدادی یال از مرتبه‌ی مربعی باشد (به طور مثال فرض کنید که نقاط ورودی به صورت یکسان روی دو پاره‌خط متقاطع که با هم زاویه‌ی  $\alpha < \pi$  می‌سازند، توزیع شده باشند. به ازای هر  $\theta < \alpha$ ،  $cY(\theta)$  شامل همه‌ی آن یال‌هایی است که یک سر انتهایی آن‌ها یک نقطه روی یک پاره‌خط و سر دیگر آن یال، نقطه‌ای روی پاره‌خط دیگر است، شکل ۱-۲ را ببینید).

در این فصل، خاصیت پوششی بودن، خاصیت خود-گرایی و خاصیت تحمل‌پذیر بودن ناحیه-خطا را برای گراف‌های یائوی پیوسته مطالعه می‌کنیم. در بخش ۲، نشان می‌دهیم که اگر  $\theta < 2\pi/3$ ، آنگاه ضریب کشش  $cY(\theta)$  برابر  $1/(1 - 2\sin(\theta/4))$  است. خواهیم دید که روشی که در این بخش بکار می‌رود را نمی‌توانیم برای حالت  $\theta = 2\pi/3$  بکار ببریم. در بخش ۳، برای اولین بار، روش جبری جدیدی را معرفی می‌کنیم و به کمک آن نشان خواهیم داد که  $cY(2\pi/3)$  یک  $6/41$ -پوشش است. در بخش ۴، حالتی که  $\theta > 2\pi/3$  است را بررسی می‌کنیم. نشان خواهیم داد که هیچ عدد ثابت  $t > 1$  وجود ندارد که به ازای آن،  $cY(\pi)$  یک  $t$ -پوشش باشد. ما مقدار ضریب کشش در حالتی که  $2\pi/3 < \theta < \pi$  باشد را به عنوان مسئله‌ی حل‌نشده مطرح می‌کنیم. همچنین همبندی گراف  $cY(\theta)$  را بررسی می‌کنیم و نشان خواهیم داد که به ازای هر  $\theta \leq \pi$ ، گراف  $cY(\theta)$  همبند است، اما اگر  $\theta > \pi$ ، آنگاه  $cY(\theta)$  ممکن است ناهمبند باشد. در بخش ۵، تحمل‌پذیری ناحیه-خطا در  $cY(\theta)$  را مطالعه می‌کنیم. در بخش ۶، نشان می‌دهیم که  $cY(\theta)$  خود-گرا نیست و در بخش ۷، این فصل را نتیجه‌گیری می‌کنیم.

توجه کنید که در طول این فصل، فرض می‌کنیم که نقاط در حالت کلی قرار دارند یعنی فاصله‌ها از یک رأس به رئوس دیگر یکتا هستند.

## ۲-۲ گراف یائوی پیوسته برای مخروط‌های باریک

در این بخش، مسئله‌ی محاسبه‌ی ضریب کشش گراف  $cY(\theta)$  برای زمانی که  $\theta < 2\pi/3$  است را در نظر می‌گیریم.

**لم ۲-۲-۱ ([۱۶]).** اگر  $a$ ،  $b$  و  $c$  سه نقطه باشند به طوری که  $|ac| \leq |ab|$  و  $\alpha < \pi$ ،  $\text{angle}(ba, ac) \leq \alpha$  آنگاه

$$|bc| \leq |ab| - (1 - 2\sin(\alpha/2))|ac|,$$

که در اینجا،  $\text{angle}(ba, ac)$  زاویه‌ی بین پاره‌خط‌های  $ba$  و  $ac$  است.

برای دو رأس  $a$  و  $b$  در گراف  $cY(\theta)$ ، فرض می‌کنیم که  $C_{ab}$  مخروطی با دهانه‌ی  $\theta$  و به مرکز

$a$  باشد به طوری که  $b$  روی نیمساز زاویه اش قرار می گیرد.

**قضیه ۲-۲-۲.** اگر  $0 < \theta < 2\pi/3$ ، آنگاه گراف  $cY(\theta)$  دارای ضریب کشش  $1/(1 - 2\sin(\theta/4))$  است.

**اثبات.** برای اثبات قضیه، کافی است نشان دهیم که بین هر دو رأس  $a$  و  $b$  در  $cY(\theta)$  یک مسیر با طول حداکثر  $|ab|/(1 - 2\sin(\theta/4))$  وجود دارد. اثبات مبتنی بر استقرا روی رتبه‌ی فاصله<sup>۱</sup>  $|ab|$  است (منظور از رتبه‌ی فاصله‌ی یک زوج نقطه یعنی اینکه تمام زوج نقاط را بر اساس فاصله‌ی اقلیدسی آن‌ها از کوچک به بزرگ مرتب کنیم. به شماره‌ی هر زوج نقطه در این ترتیب، رتبه‌ی فاصله‌ی آن زوج می‌گوییم).

**پایه‌ی استقرا:** در صورتی که  $a$  و  $b$  نزدیک‌ترین زوج از رئوس باشند، از این رو، یال  $(a, b)$  در هر مخروط با مرکز  $a$  که شامل  $b$  باشد، به  $cY(\theta)$  اضافه می‌شود.

**فرض استقرا:** فرض می‌کنیم که حکم، برای هر دو رأس که فاصله‌ی اقلیدسی آن‌ها از فاصله‌ی اقلیدسی زوج  $(a, b)$ ، کمتر است، برقرار باشد.

**حکم استقرا:** حال، نشان می‌دهیم که بین دو رأس  $a$  و  $b$  یک مسیر به طول  $|ab|/(1 - 2\sin(\theta/4))$  وجود دارد.

اگر یال  $(a, b)$  در گراف باشد، آنگاه بدیهی است که یک مسیر به طول حداکثر  $|ab|/(1 - 2\sin(\theta/4))$  بین  $a$  و  $b$  وجود دارد و اثبات تمام است. پس، فرض کنید که  $(a, b)$

متعلق به  $cY(\theta)$  نباشد. این بدین معناست که رأسی نزدیک‌تر به  $a$  نسبت به  $b$  در هر مخروط با دهانه‌ی  $\theta$  و به مرکز  $a$  که شامل  $b$  باشد، وجود دارد. بنابراین، این موضوع برای مخروط  $C_{ab}$

نیز صدق می‌کند. فرض کنید که  $n_a$  نزدیک‌ترین رأس به  $a$  در  $C_{ab}$  باشد. از آنجایی که  $C_{ab}$  دارای دهانه‌ی  $\theta$  است، بنابراین  $\angle(n_a a, ab)$  حداکثر  $\theta/2$  است. بنابراین، طبق لم ۲-۲-۱ داریم

$|bn_a| \leq |ab| - (1 - 2\sin(\theta/4))|an_a|$ . توجه کنید از آنجایی که  $0 < \theta < 2\pi/3$ ، داریم  $\theta/4 < \pi/6$  که

بدین معناست که  $1 - 2\sin(\theta/4) > 0$  و از این رو،  $|bn_a| < |ab|$ . بنابراین، می‌توانیم فرض استقرا

را برای  $n_a$  و  $b$  بکار ببریم. پس، یک مسیر  $Q$  بین  $n_a$  و  $b$  با طول حداکثر  $|bn_a|/(1 - 2\sin(\theta/4))$

وجود دارد. حال مسیر  $\{(a, n_a)\} \cup Q$  را در نظر بگیرید. به وضوح این مسیر، یک مسیر بین  $a$  و  $b$

<sup>۱</sup>rank of distance

است و طول آن حداکثر برابر است با:

$$|an_a| + \frac{1}{1 - 2 \sin(\theta/4)} |bn_a| \leq |an_a| + \frac{1}{1 - 2 \sin(\theta/4)} (|ab| - (1 - 2 \sin(\theta/4)) |an_a|) =$$

$$|an_a| + \frac{1}{1 - 2 \sin(\theta/4)} |ab| - |an_a| = \frac{1}{1 - 2 \sin(\theta/4)} |ab|.$$

این رابطه، اثبات را کامل می‌کند. ■

## ۲-۳ گراف $cY(2\pi/3)$ یک $6/0$ پوشش

بزرگ‌ترین ریشه‌ی چندجمله‌ای

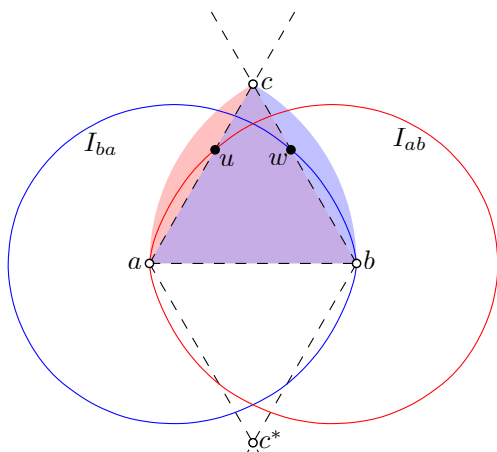
$$p(t) = -25 + 90t - 39t^2 - 246t^3 + 363t^4 + 138t^5 - 589t^6 + 216t^7 + 291t^8 - 204t^9 - 84t^{10} + 6t^{11} + 2t^{12}$$

را در نظر بگیرید. فرض می‌کنیم که  $t \approx 6/0$  بزرگ‌ترین ریشه‌ی  $p(t)$  است. در این بخش، ثابت می‌کنیم که  $cY(2\pi/3)$  یک  $t$ -پوشش است. بدین منظور، نشان می‌دهیم که برای هر دو رأس  $a$  و  $b$  در  $cY(2\pi/3)$ ، یک مسیر از  $a$  به  $b$  به طول حداکثر  $t|ab|$  وجود دارد. در انتهای این بخش، مشخص خواهد شد که این چندجمله‌ای چگونه بدست می‌آید.

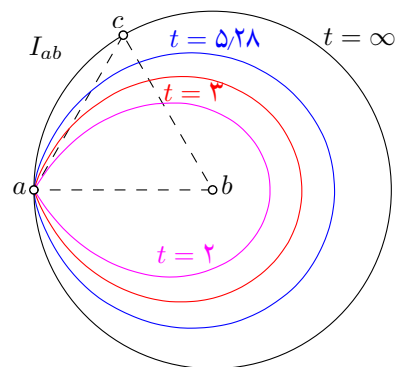
فرض کنید که  $a$  و  $b$  دو رأس دلخواه در  $cY(2\pi/3)$  باشد. اثبات مبتنی بر استقرا روی رتبه‌ی فاصله‌ی  $|ab|$  است. در صورتی که  $a$  و  $b$  نزدیک‌ترین زوج رئوس در  $cY(2\pi/3)$  باشند، به وضوح، یال  $(a, b)$  در هر مخروط به مرکز  $a$  که شامل  $b$  باشد، اضافه می‌شود (توجه کنید که ما از قبل فرض کرده‌ایم که نقاط در حالت کلی قرار دارند). حال، فرض کنید که نتایج برای هر دو رأس که فاصله‌ی آن‌ها کمتر از  $|ab|$  است، برقرار باشد. نشان می‌دهیم که برای دو رأس  $a$  و  $b$  در  $cY(2\pi/3)$ ، یک مسیر از  $a$  به  $b$  به طول حداکثر  $t|ab|$  وجود دارد. بدون از دست دادن کلیت، فرض کنید که  $a = (0, 0)$  و  $b = (1, 0)$ . بنابراین  $|ab| = 1$ . ما مجموعه‌ی استقرایی  $a$  نسبت به  $b$  را با  $I_{ab} = \{p \in \mathbb{R}^2 : |ap| + t|pb| \leq t|ab|\}$  تعریف می‌کنیم (شکل ۲-۲ را ببینید).

اثبات را با یک لم شروع می‌کنیم که از فرض اولیه که نقاط در حالت کلی قرار دارند، نتیجه

می‌شود.



(ب) مجموعه‌های استقرایی  $I_{ba}$  و  $I_{ab}$  نمایش داده شده‌اند. بخش‌های دایره‌ای که  $n_b$  و  $n_a$  می‌توانند قرار گیرند به ترتیب با رنگ آبی روشن و قرمز روشن نمایش داده شده است.



(آ) مجموعه‌ی استقرایی  $I_{ab}$  برای مقادیر مختلف  $t$ .

شکل ۲-۲

لم ۲-۳-۱. مجموعه‌ی استقرایی  $I_{ab}$ ، توسط دیسک  $D$  به مرکز  $b$  و شعاع  $|ab|$  دربرگرفته می‌شود. همچنین، هر نقطه‌ی  $p \neq a$  که روی مرز  $D$  است، خارج از  $I_{ab}$  قرار دارد.

**اثبات.** فرض کنید که  $p \neq a$  یک نقطه در  $I_{ab}$  باشد. از آنجایی که  $|ap| > 0$ ، داریم  $t|pb| < |ap|$ . در نتیجه،  $|ap| + t|pb| \leq t|ab|$ . حتماً داخل دایره به مرکز  $b$  و شعاع  $|ab|$  قرار دارد و روی مرز آن قرار ندارد. ■

همان طور که از قبل ذکر شد  $C_{ab}$  یک مخروط به مرکز  $a$  است که  $b$  روی نیمسازش قرار دارد. فرض کنید که  $n_a$  و  $n_b$  به ترتیب نزدیک‌ترین نقاط به  $a$  و  $b$  در مخروط‌های  $C_{ba}$  و  $C_{ab}$  باشند. هرگاه  $n_a \in I_{ab}$ ، آنگاه یک مسیر از  $a$  به  $b$  به طول حداکثر  $t|ab|$  وجود داشته باشد. درواقع، چون  $n_a \in I_{ab}$  طبق لم ۲-۳-۱ داریم  $|n_a b| < |ab|$ . بنابراین، می‌توانیم فرض استقرا را بکار ببریم و یک مسیر  $Q$  به طول حداکثر  $t|n_a b|$  بین  $n_a$  و  $b$  بدست آوریم. چون  $n_a \in I_{ab}$  بنابراین مسیر  $\{(a, n_a)\} \cup Q$  یک مسیر از  $a$  به  $b$  به طول حداکثر  $|an_a| + t|n_a b| \leq t|ab|$  است که مطلوب می‌باشد. به طور مشابه، مجموعه‌ی  $I_{ba}$  دارای خاصیت استقرایی است.

توجه کنید که اگر  $n_a \in I_{ab}$  یا  $n_b \in I_{ba}$  آنگاه اثبات با توجه به خاصیت استقرایی، تمام است و حکم برقرار می‌باشد. از این رو، فرض می‌کنیم که  $n_a \notin I_{ab}$  و  $n_b \notin I_{ba}$  از آنجایی که

$a = (0, 0)$  و  $b = (1, 0)$  بنابراین مجموعه‌ی نقاط روی مرز  $I_{ab}$ ، در رابطه‌ی زیر صدق می‌کنند:

$$((x^2 + y^2)t^2 + (x^2 + y^2)^2 - 2(2 + (-2 + x)x + y^2)(x^2 + y^2)t^2 = 0, \quad (1-2)$$

که این معادله، یک منحنی درجه‌ی چهار بر اساس  $x$  و  $y$  را تعریف می‌کند. فرض کنید که  $c$  و  $c^*$  نقاط اشتراک مرزهای  $C_{ba}$  و  $C_{ab}$  باشند و فرض کنید که  $c$  بالای  $c^*$  قرار دارد (شکل ۲-۲). از آنجایی که مثلث‌های  $\Delta abc$  و  $\Delta abc^*$  متساوی‌الاضلاع هستند، داریم  $c = (1/2, \sqrt{3}/2)$  و  $c^* = (1/2, -\sqrt{3}/2)$ . نقطه‌ی اشتراک مرز  $I_{ab}$  با پاره‌خط  $ac$  را  $u$  می‌نامیم. بنابراین،  $u$  برابر است با

$$u = \left( \frac{t(t-2)}{2(t^2-1)}, \frac{\sqrt{3}t(t-2)}{2(t^2-1)} \right) \approx (0.3438, 0.5956). \quad (2-2)$$

به طور مشابه، فرض کنید که نقطه‌ی اشتراک مرز  $I_{ba}$  با پاره‌خط  $bc$  را  $w$  می‌نامیم. بنابراین،  $w$  برابر است با

$$w = \left( 1 - \frac{t(t-2)}{2(t^2-1)}, \frac{\sqrt{3}t(t-2)}{2(t^2-1)} \right) \approx (0.6561, 0.5956).$$

سه نقطه‌ی  $x$ ،  $y$  و  $y'$  را به گونه‌ای در نظر می‌گیریم که  $|xy'| = |xy|$ . قطاع مدور<sup>۱</sup> به مرکز  $x$  که بین  $xy$  و  $xy'$  و در جهت خلاف عقربه‌های ساعت، قرار گرفته است را با  $\mathcal{C}(x, y, y')$  نمایش می‌دهیم. حال، دو حالت وجود دارد که آن‌ها را در نظر می‌گیریم:

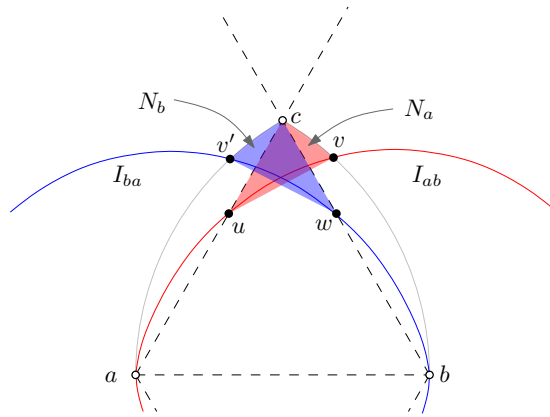
**حالت الف):** فرض کنید که  $n_a$  و  $n_b$  هر دو بالای محور  $x$  قرار دارند. اثبات در حالتی که دو نقطه زیر محور  $x$  است، مشابه است. از آنجایی که  $n_a$  و  $n_b$  به ترتیب در قطاع‌های مدور  $\mathcal{C}(a, b, c)$  و  $\mathcal{C}(b, c, a)$  قرار دارند، بنابراین، داریم  $|n_a n_b| < |ab|$ . پس، می‌توانیم فرض استقرا را روی  $n_a n_b$  بکار ببریم و سپس یک مسیر  $\varphi_{n_a n_b}$  از  $n_a$  به  $n_b$  به طول حداکثر  $t|n_a n_b|$  بدست آوریم. حال، مسیر  $\varphi_{ab} = a n_a \cap \varphi_{n_a n_b} \cup n_b b$  از  $a$  به  $b$  را در نظر بگیرید. نشان می‌دهیم که طول  $\varphi_{ab}$  حداکثر  $t|ab|$  است. بدین منظور، کرانی را روی طول پاره‌خط  $n_a n_b$  پیدا می‌کنیم.

**لم ۲-۳-۲.** در پیکربندی حالت الف) که در شکل ۲-۲ نمایش داده شده است، داریم:

$$|n_a n_b| \leq |uc| = |wc| = |uw|.$$

<sup>۱</sup> circular sector





شکل ۲-۳: ناحیه‌های مجاور به  $a$  و  $b$  در حالت الف).

**اثبات.** همان طور که از قبل ذکر شد،  $n_a$  باید داخل قطاع مدور  $\mathcal{C}(a, b, c)$  قرار گیرد. همچنین، چون فرض کرده‌ایم که  $n_a$  خارج از  $I_{ab}$  قرار دارد، بنابراین  $n_a$  در قسمت  $\mathcal{C}(a, b, c) \setminus I_{ab}$  قرار دارد. فرض کنید که  $N_a$  پوشش محدب<sup>۱</sup>  $\mathcal{C}(a, b, c) \setminus I_{ab}$  باشد و  $v$  نقطه‌ی اشتراک  $I_{ab}$  و کمانی از قطاع مدور  $\mathcal{C}(a, b, c)$  باشد (شکل ۲-۳ را ببینید). به طور مشابه، فرض کنید که  $v'$  نقطه‌ی اشتراک  $I_{ba}$  و کمانی از قطاع مدور  $\mathcal{C}(b, c, a)$  باشد. بنابراین،  $N_a$  توسط پاره‌خط‌های  $ac$ ،  $uv$  و کمان مدور به مرکز  $a$  و به شعاع ۱ که  $v$  را به  $c$  متصل می‌کند، احاطه شده است. به طور مشابه،  $N_b$  را پوشش محدب  $\mathcal{C}(b, c, a) \setminus I_{ba}$  می‌نامیم. چون  $n_a \in N_a$  و  $m_b \in N_b$  بنابراین با محاسبه‌ی بیش‌ترین فاصله‌ی بین یک نقطه در  $N_a$  و یک نقطه در  $N_b$ ، می‌توانیم یک کران بالا روی فاصله‌ی بین  $n_a$  و  $m_b$  بدست آوریم. ما دو نقطه‌ای که این فاصله را بیشینه می‌کند، با  $m_a$  و  $m_b$  نمایش می‌دهیم که  $m_a \in N_a$  و  $m_b \in N_b$  (توجه کنید که ممکن است چند زوج  $(m_a, m_b)$  وجود داشته باشد). از آنجایی که تابع فاصله‌ی اقلیدسی، یک تابع محدب (تابع  $f: X \rightarrow \mathbb{R}$  یک تابع محدب در فضای برداری  $X$  است هرگاه به ازای هر  $x_1$  و  $x_2$  در  $X$  و به ازای هر  $t \in [0, 1]$ ، داشته باشیم

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

است و از آنجایی که هر دو مجموعه‌ی  $N_a$  و  $N_b$  محدب هستند، بنابراین زوجی مانند

$(m_a, m_b)$  وجود دارد به طوری که  $m_a$  روی مرز  $N_a$  و  $m_b$  روی مرز  $N_b$  قرار دارند.

در حقیقت، ادعا می‌کنیم که فقط نیاز است که مرزهای مثلث‌های  $\Delta(u, v, c) \subset N_a$  و

$\Delta(w, c, v') \subset N_b$  را به منظور پیدا کردن  $(m_a, m_b)$ ، در نظر بگیریم. برای اثبات این ادعا، ما<sup>۲</sup>

ایجاد شده توسط  $N_a \setminus \Delta(u, v, c)$  را در نظر بگیرید. برای یک نقطه‌ی  $x$  در این ماه، دورترین نقطه

<sup>۱</sup>convex hull

<sup>۲</sup>lune

از آن که در  $N_b$  قرار دارد و با  $f(x)$  نمایش می‌دهیم را در نظر بگیرید. توجه کنید که دایره به مرکز  $f(x)$  که از  $x$  می‌گذرد، یا  $c$  و یا  $v$  و یا هر دو را شامل نمی‌شود، زیرا، شعاع این دایره کوچک‌تر از شعاع کمان مدور روی مرز  $N_a$  است (شکل ۲-۳ را ببینید). بنابراین، یا  $c$  و یا  $v$  نسبت به  $x$  از  $f(x)$  دورترند و از این رو،  $m_a$  و  $m_b$  نمی‌توانند داخل این ماه قرار گیرند. بنابراین،  $m_a$  روی مرز مثلث  $\Delta(u, v, c)$  قرار دارد. به طور مشابه می‌توان ثابت کرد که  $m_b$  روی مرز مثلث  $\Delta(w, c, v')$  قرار دارد. از آنجایی که مختصات رئوس مثلث‌های  $\Delta(u, v, c)$  و  $\Delta(w, c, v')$  را از قبل می‌دانیم، بنابراین

می‌توانیم به راحتی بررسی کنیم  $|m_a m_b| = |uc| = |cw| = |uv|$ .

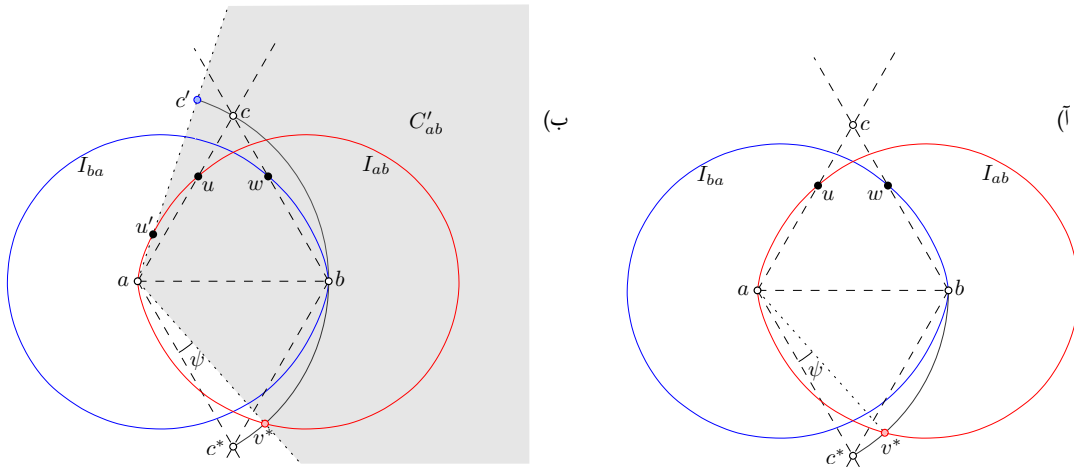
از آنجایی که طول  $|n_a n_b|$  حداکثر  $|uc|$  است و  $|a n_a|$  و  $|b n_b|$  هر دو حداکثر ۱ هستند، بنابراین، با توجه به لم ۲-۳-۲، طول مسیر  $\varphi_{ab} = a n_a \cup \varphi_{n_a n_b} \cup n_b b$  حداکثر  $2 + t|uc|$  است. حال، ثابت می‌کنیم که  $2 + t|uc| \leq t|ab|$  از آنجایی که  $a = (0, 0)$ ،  $b = (1, 0)$ ،  $c = (1/2, \sqrt{3}/2)$  و  $|au| = \frac{t(t-2)}{t^2-1}$  نامساوی  $2 + t|uc| \leq t|ab|$  هم ارز با نامساوی زیر است:

$$2 + t \left( 1 - \frac{t(t-2)}{t^2-1} \right) \leq t.$$

این نامساوی درست است هرگاه  $0 \leq t^3 - 4t^2 + 2$  و  $t > 1$ . از آنجایی که  $t = 670411$  بزرگ‌تر از بزرگ‌ترین ریشه‌ی حقیقی  $2 + 4x^2 - x^3$  است، بنابراین اثبات تمام است. بنابراین، هرگاه پیکربندی حالت الف) را داشته باشیم، می‌توانیم فرض استقررا را بکار ببریم و یک مسیر  $\varphi_{ab}$  از  $a$  به  $b$  به طول حداکثر  $2 + t|uc| \leq t|ab|$  بدست آوریم.

**حالت ب):** اثبات حالت ب) کمی مشکل است اما روش اثبات مانند حالت الف) است. اگر  $n_a$  و  $n_b$  در جهت‌های مخالف همدیگر نسبت به پاره خط  $ab$  قرار گیرند، آنگاه بدون از دست دادن کلیت، می‌توانیم فرض کنیم که  $n_a$  پایین محور  $x$  و  $n_b$  بالای محور  $x$  قرار دارد. توجه کنید که  $c^*$  نقطه‌ی تقاطع مرزهای  $C_{ab}$  و  $C_{ba}$  است که در پایین محور  $x$  قرار دارد.

از آنجایی که  $(a, b)$  یال  $cY(2\pi/3)$  نیست، بنابراین  $n_a$  باید داخل  $\mathcal{C}(a, c^*, b)$  قرار گیرد. فرض کنید که  $v^*$  نقطه‌ی تقاطع مرز  $I_{ab}$  با کمان  $\mathcal{C}(a, c^*, b)$  باشد (شکل ۲-۴ را ببینید). این نقطه‌ی تقاطع همیشه وجود دارد زیرا  $b$  داخل  $I_{ab}$  است و طبق لم ۲-۳-۱،  $c^*$  خارج از  $I_{ab}$  قرار دارد. از آنجایی که کمان  $\mathcal{C}(a, c^*, b)$  بخشی از دایره‌ی  $x^2 + y^2 = 1$  است، از معادله‌ی ۱-۲ داریم:



شکل ۲-۴: (آ) نقطه‌ی  $v^*$  و زاویه‌ی  $\psi = \text{angle}(v^*a, ac^*)$  (ب) مخروط  $C'_{ab}$  که با چرخاندن  $C_{ab}$  به اندازه‌ی  $\psi$  درجه در جهت خلاف عقربه‌های ساعت، بدست می‌آید.

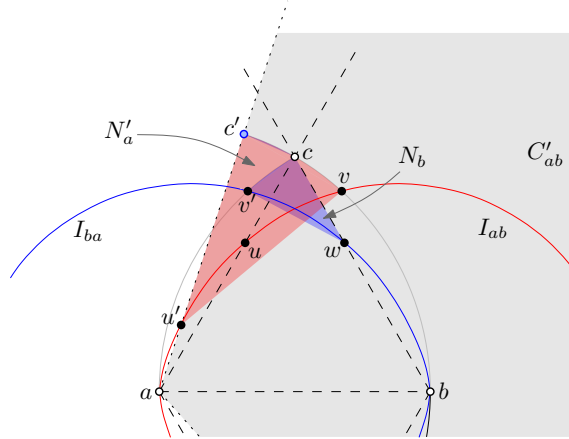
$$v^* = \left( \frac{t^2 + 2t - 1}{2t^2}, -\frac{t-1}{2t^2} \sqrt{(t+1)(3t-1)} \right) \quad (3-2)$$

$$\approx (0.6518, -0.7583).$$

فرض کنید که  $\psi = \text{angle}(v^*a, ac^*)$  (شکل ۲-۴(آ) را ببینید). از آنجایی که  $\psi = \pi/3$  فرض کنیم، بنابراین از معادله‌ی ۳-۲ داریم:

$$\begin{aligned} \tan(\psi) &= \tan(\pi/3 - \text{angle}(ba, av^*)) = \frac{\tan(\pi/3) - \tan(\text{angle}(ba, av^*))}{1 + \tan(\pi/3) \tan(\text{angle}(ba, av^*))} \\ &= \frac{\sqrt{3}(t^2 + 2t - 1) - (t-1)\sqrt{(t+1)(3t-1)}}{t^2 + 2t - 1 + \sqrt{3}(t-1)\sqrt{(t+1)(3t-1)}}. \end{aligned} \quad (4-2)$$

بنابراین  $\tan(\psi) \approx 0.1885$  و در نتیجه  $\psi \approx 10.6800^\circ$ . فرض کنید که مخروط  $C'_{ab}$  (و بدین ترتیب نقطه‌ی  $c'$ )، از چرخش  $C_{ab}$  (نقطه‌ی  $c$ ) حول نقطه‌ی  $a$  و در جهت خلاف عقربه‌های ساعت و به اندازه‌ی  $\psi$  درجه، بدست آید. توجه کنید که  $\mathcal{C}(a, v^*, b) \subset I_{ab}$  (شکل ۲-۴(ب) را ببینید). فرض کنید که  $n'_a$  نزدیک‌ترین نقطه به  $a$  داخل  $C'_{ab}$  باشد. اگر  $n'_a$  داخل  $I_{ab}$  باشد، بنابر فرض استقرا، اثبات تمام است. بنابراین، فرض کنید که  $n'_a \notin I_{ab}$  از آنجایی که  $\mathcal{C}(a, v^*, b) \subset I_{ab}$  نمی‌تواند داخل  $\mathcal{C}(a, v^*, b)$  باشد و از این رو،  $n'_a$  باید بالای محور  $x$  قرار گیرد. فرض کنید که  $N'_a$  پوشش محدب  $\mathcal{C}(a, c', b) \setminus I_{ab}$  باشد. بنابراین،  $n'_a$  باید داخل  $N'_a$  باشد (شکل ۲-۵ را ببینید). همانند حالت الف)  $n_b$  باید داخل  $N_b$  که پوشش محدب  $\mathcal{C}(b, c, a) \setminus I_{ba}$  است، قرار گیرد. فرض کنید که  $u' \in ac$  نقطه‌ی اشتراک مرزهای  $C'_{ab}$  و  $I_{ab}$  باشد (شکل ۲-۵ را ببینید). از معادله‌ی ۴-۲ نتیجه



شکل ۲-۵:  $N_b$  و  $N'_a$

می‌شود که معادله‌ی خطی که از دو نقطه‌ی  $a$  و  $c'$  می‌گذرد، به صورت زیر است:

$$y = \tan(\pi/3 + \psi) x = \frac{\tan(\pi/3) + \tan(\psi)}{1 - \tan(\pi/3) \tan(\psi)} x$$

$$= \frac{\sqrt{3}(t^2 + 2t - 1) + (t - 1)\sqrt{(t+1)(3t-1)}}{-(t^2 + 2t - 1) + \sqrt{3}(t-1)\sqrt{(t+1)(3t-1)}} x.$$

بنابراین، مؤلفه‌ی  $x$  نقطه‌ی  $u'$  برابر است با

$$\frac{1}{4t^2(t^2 - 1)} \left( 5t^4 - 2t^3 + 2t^2 + 2t - 1 \right. \\ \left. - \sqrt{3}(t-1)(t^2 + 4t - 1)\sqrt{(t+1)(3t-1)} \right)$$

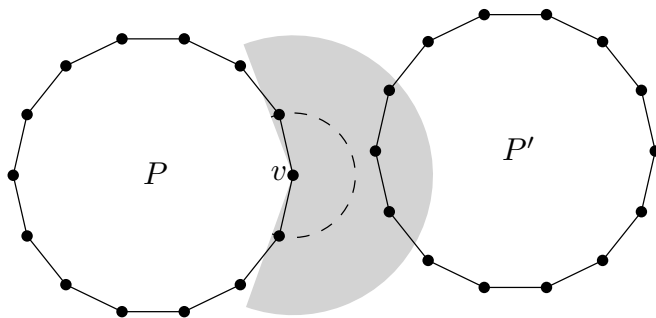
و مؤلفه‌ی  $x$  نقطه‌ی  $c'$  برابر است با

$$\frac{-(t^2 + 2t - 1) + \sqrt{3}(t-1)\sqrt{(t+1)(3t-1)}}{4t^2}.$$

بنابراین،  $u' \approx (0.1124, 0.3207)$  و  $c' \approx (0.3308, 0.9436)$ . مشابه اثبات لم ۲-۳-۲، لم زیر را ثابت می‌کنیم.

لم ۲-۳-۳. در پیکربندی حالت ب)، فاصله‌ی بین  $n'_a$  و  $n_b$  حداکثر  $|u'c|$  است.

**اثبات.** چون  $m_b \in N_b$  و  $n'_a \in N'_a$ ، بنابراین، با محاسبه‌ی بیش‌ترین فاصله‌ی بین یک نقطه در  $N'_a$  و یک نقطه در  $N_b$ ، می‌توانیم یک کران بالا برای فاصله‌ی بین  $n'_a$  و  $n_b$  بدست آوریم. مشابه اثبات لم ۲-۳-۲، می‌توانیم نشان دهیم که بیش‌ترین فاصله زمانی ایجاد می‌شود که یک نقطه مانند  $m_a$  از مرز  $N'_a$  و یک نقطه مانند  $m_b$  از مرز  $N_b$  انتخاب شود. به راحتی می‌توان درستی این موضوع را بررسی کرد که فاصله‌ی هر نقطه‌ی  $p \in N_b$  از  $u'$  نسبت به فاصله‌ی هر نقطه‌ی  $q \in N'_a$  که  $q \neq p$  از



شکل ۲-۶: اگر  $\theta > \pi$ ، آنگاه  $cY(\theta)$  می‌تواند ناهمبند باشد.

$u'$  بیش‌تر است. بنابراین، کافی است دورترین نقطه‌ی  $N_b$  از  $u'$  را پیدا کنیم. همچنین توجه کنید که دایره به مرکز  $u'$  که از نقطه‌ای روی کمان مدور از  $N_b$  می‌گذرد، شامل  $c$  نمی‌شود. بنابراین، کافی است که دورترین نقطه‌ی روی مرز مثلث  $\Delta(w, c, v')$  از  $N_b \supset \Delta(w, c, v')$  را پیدا کنیم.

با توجه به مختصات دقیق  $u'$  و همچنین مختصات رئوس مثلث  $\Delta(w, c, v')$ ، به راحتی

می‌توانیم بررسی کنیم که  $m_a = u'$  و  $m_b = c$  بنابراین  $|n'_a n_b| \leq |m_a m_b| = |u'c|$ .

با توجه به لم ۲-۳-۳، فاصله‌ی بین  $n'_a$  و  $n_b$  حداکثر برابر  $|u'c| > 1$  است. بنابراین، طبق

فرض استقرا می‌توانیم یک مسیر  $\varphi_{n'_a n_b}$  از  $n'_a$  به  $n_b$  به طول حداکثر  $t|n'_a n_b|$  بدست آوریم.

فرض کنید که  $\varphi_{ab} = n_b b \cup \varphi_{n'_a n_b} \cup a n'_a$  یک مسیر از  $a$  به  $b$  باشد. با توجه به لم ۲-۳-۳،

به وضوح، طول  $\varphi_{ab}$  حداکثر  $2 + \varphi_{n'_a n_b} \geq 2 + t|u'c|$  است.

ما اکنون ثابت می‌کنیم که  $t|ab| \geq 2 + t|u'c|$ . از آنجایی که  $a = (0, 0)$ ،  $b = (1, 0)$  و

$c = (1/2, \sqrt{3}/2)$ ، بنابراین با استفاده از مختصات  $u'$ ، به این نتیجه می‌رسیم که  $t|ab| \geq 2 + t|u'c|$

هرگاه  $p(t) \geq 0$  و  $t \approx 6.411$  (توجه کنید که  $t \approx 6.411$  بزرگ‌ترین ریشه‌ی حقیقی چندجمله‌ای

$p(t)$  است). بنابراین، هرگاه پیکربندی حالت (ب) رخ دهد، می‌توانیم با کمک استقرا یک مسیر  $\varphi_{ab}$

از  $a$  به  $b$  به طول حداکثر  $t|ab| \geq 2 + t|u'c|$  بدست آوریم.

به طور خلاصه، برای هر دو رأس  $a$  و  $b$  در  $cY(2\pi/3)$  و برای ثابت  $t \approx 6.411$ ، یک مسیر

از  $a$  به  $b$  در  $cY(2\pi/3)$  به طول حداکثر  $t|ab|$  وجود دارد. بنابراین قضیه‌ی زیر را بدست می‌آوریم.

**قضیه ۲-۳-۴.** گراف  $cY(\theta)$  یک  $t$ -پوشش است به طوری که اگر  $\theta = 2\pi/3$ ، آنگاه  $t \leq 6.411$

و اگر  $\theta < 2\pi/3$ ، آنگاه  $t \leq \min \left\{ 6.411, \frac{1}{1 - 2 \sin(\theta/4)} \right\}$ .

## ۴-۲ زاویه‌های بزرگ‌تر

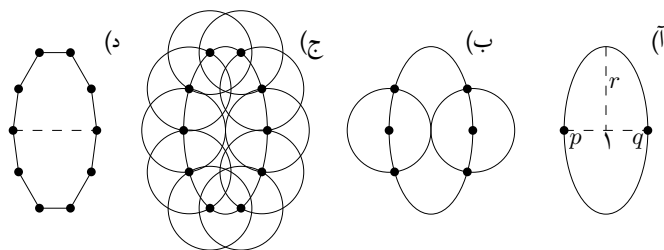
قضیه ۴-۳-۲، یک کران بالا برای ضریب کشش  $cY(\theta)$  برای مقادیر  $\theta \leq 2\pi/3$  فراهم می‌کند. اما برای مقادیر  $\theta > 2\pi/3$  چه اتفاقی می‌افتد؟ نتایجی که در ادامه می‌آید نشان می‌دهد که اگر  $\theta$  خیلی بزرگ باشد، آنگاه  $cY(\theta)$  می‌تواند ناهمبند باشد.

**قضیه ۴-۲-۱.** برای هر  $\theta > \pi$ ، مجموعه‌ای از نقاط وجود دارد به طوری که  $cY(\theta)$  روی آن ناهمبند است.

**اثبات.** چون  $\theta > \pi$ ، بنابراین  $\epsilon > 0$  وجود دارد به طوری که  $\theta = \pi + \epsilon$ . فرض کنید که  $P$  یک چندضلعی منتظم باشد به طوری که زاویه‌های داخلی آن حداقل  $\pi - \epsilon/2$  رادیان باشد.  $P'$  را یک کپی از  $P$  در نظر بگیرید. حال،  $P$  و  $P'$  را طوری در صفحه قرار می‌دهیم که فاصله‌ی بین آن‌ها بیش‌تر از فاصله‌ی بین دو رأس متوالی روی  $P$  باشد (شکل ۴-۲ را ببینید). یک رأس  $v$  روی  $P$  را در نظر بگیرید. زاویه‌ی خارجی  $v$ ، حداکثر  $2\pi - (\pi - \epsilon/2) = \pi + \epsilon/2$  رادیان است. از آنجایی که این زاویه کم‌تر از  $\theta$  است، هر مخروط به مرکز  $v$  یکی از همسایه‌های  $v$  روی  $P$  را شامل می‌شود. از این رو، از آنجایی که فاصله‌ی بین  $P$  و  $P'$  بیش‌تر از فاصله‌ی  $v$  تا همسایه‌هایش است، در  $cY(\theta)$ ، هرگز به رأسی در  $P'$  وصل نخواهد شد. از آنجایی که ما  $v$  را به صورت کاملاً دلخواه انتخاب کردیم و  $P'$  یک کپی از  $P$  است، در نتیجه، هیچ یالی در  $cY(\theta)$  را به  $P'$  متصل نخواهد کرد. ■

**قضیه ۴-۲-۲.** برای هر مجموعه نقطه‌ی  $S$  و هر  $\theta \leq \pi$ ، گراف یائوی پیوسته‌ی  $cY(\theta)$  روی  $S$  همبند است.

**اثبات.** از آنجایی که برای هر  $\theta \leq \pi$ ، داریم  $cY(\pi) \subseteq cY(\theta)$ ، کافی است قضیه را برای  $\theta = \pi$  ثابت کنیم. فرض کنید که  $S_r$  مجموعه‌ی همه‌ی نقاط در  $S$  باشد به طوری که مختصات  $x$  آن‌ها بیش‌ترین است. حال، مجموعه‌ی  $S' = S \setminus S_r$  را در نظر بگیرید. فرض کنید که  $S' \neq \emptyset$  و  $v$  یک نقطه‌ی دلخواه در  $S'$  باشد (توجه کنید که در حالتی که  $S' = \emptyset$  است، همه‌ی نقاط  $S$  روی یک خط افقی قرار دارند که مشابه آنچه که در پاراگراف بعدی آمده است، می‌توان ثابت کرد که  $cY(\theta)$



شکل ۷-۲: اثبات قضیه ۲-۴-۳.

همبند است). ثابت می‌کنیم که بین  $v$  و یک نقطه در  $S_r$  یک مسیری در  $cY(\pi)$  وجود دارد. از آنجایی که  $v \in S'$ ، بنابراین، همواره یک مخروط  $C$  به مرکز  $v$  و با دهانه‌ی  $\pi$  وجود دارد که شامل یک نقطه در  $S_r$  باشد. حال، فرض کنید که  $q$  نزدیک‌ترین نقطه به  $v$  در  $C$  باشد. به وضوح،  $(v, q)$  یالی در  $cY(\pi)$  است. از آنجایی که تعداد نقاط  $S$  متناهی است، بنابراین با تکرار روند بالا، بین  $v$  و یک نقطه در  $S_r$  یک مسیری وجود دارد.

حال، فرض کنید که تعداد اعضای  $S_r$ ، حداقل ۲ باشد. به راحتی می‌توان نشان داد که بین هر دو رأس در  $S_r$ ، یک مسیر مستقیم وجود دارد که فقط کافیست مخروط را در هر رأس  $S_r$  اندکی بچرخانید تا شامل فقط یک رأس در  $S_r$  باشد. حال، از آنجایی که بین هر رأس در  $S'$  و یک رأس در  $S_r$  یک مسیری هست، پس با اتصال مسیرها، بین هر دو رأس در  $S$  یک مسیری در  $cY(\pi)$  است و در نتیجه  $cY(\pi)$  همبند است. ■

در ادامه، نشان می‌دهیم که هیچ عدد ثابت  $t$  وجود ندارد که  $cY(\pi)$  یک  $t$ -پوشش باشد.

**قضیه ۲-۴-۳.** به ازای هر عدد ثابت  $t > 1$ ، مجموعه‌ای از نقاط در صفحه وجود دارد به طوری که گراف یائوی پیوسته‌ی  $cY(\pi)$  روی آن یک  $t$ -پوشش نیست.

**اثبات.** مجموعه‌ی نقاطی را که می‌خواهیم بسازیم،  $S$  می‌نامیم. فرض کنید که  $S$  مجموعه‌ای از دو نقطه‌ی  $p$  و  $q$  باشد به طوری که  $|pq| = 1$  باشد. می‌خواهیم نقاطی را به  $S$  اضافه کنیم به طوری که طول کوتاه‌ترین مسیر بین  $p$  و  $q$  در گراف  $cY(\pi)$  روی مجموعه‌ی رؤس  $S$ ، به اندازه‌ی دلخواه زیاد شود. روش ساخت این مجموعه در شکل ۷-۲ نمایش داده شده است. فرض کنید که  $C$  دایره‌ای به قطر  $|pq|$  باشد که از  $p$  و  $q$  می‌گذرد. ما این دایره را به صورت عمودی و با مضربی از  $2r$  که  $r \geq 1$  یک عدد حقیقی است، همانند یک کش، می‌کشیم (شکل ۷-۲ آ را ببینید). شکل

حاصل یک بیضی می‌باشد. ابتدا چهار نقطه، هر یک به فاصله‌ی  $1/2$  از  $p$  یا  $q$  را روی این بیضی قرار می‌دهیم (شکل ۲-۷ را ببینید). سپس، نقاط دیگری را به این شکل اضافه می‌کنیم که ما این نقاط را در فاصله‌ی  $1/2$  از نقاط موجود قرار می‌دهیم و این کار را ادامه می‌دهیم تا وقتی که هر دو زنجیره‌ی نقاط، به همدیگر برسند (شکل ۲-۷ ج را ببینید).

حال، یک رأس  $v$  از مجموعه‌ی جدید  $S$  را در نظر بگیرید. فرض کنید که  $H$  نیم‌صفحه‌ای است که از  $v$  می‌گذرد و شامل تعدادی نقطه در  $S$  است. بر طبق ساختی که از  $S$  ارائه کردیم به وضوح، فاصله‌ی  $v$  تا همسایه‌هایش روی بیضی نسبت به فاصله‌ی  $v$  تا نقاط دیگر که در  $H$  هستند، کمتر است (به غیر از انتهای زنجیره). بنابراین، در ساخت  $cY(\pi)$  هیچ قطری در این بیضی ایجاد نمی‌شود. در نتیجه،  $cY(\pi)$  تشکیل یک چندضلعی محدب می‌دهد که در واقع همان پوشش محدب  $S$  است (شکل ۲-۷ د را ببینید).

حال اگر  $r$  را افزایش دهیم، تعداد رئوس روی هر زنجیره افزایش می‌یابد. وقتی که هر یک از زنجیره‌ها دارای  $k$  رأس است، طول کوتاه‌ترین مسیر بین  $p$  و  $q$  حداقل  $k/2 = 2k/2$  خواهد بود. از آنجایی که فاصله‌ی بین  $p$  و  $q$  ثابت می‌ماند، می‌توانیم  $r$  را به اندازه‌ی دلخواه، بزرگ انتخاب کنیم، بنابراین، هیچ مقدار ثابت  $t$  وجود ندارد که به ازای آن،  $cY(\pi)$  یک  $t$ -پوشش باشد. ■

## ۲-۵ تحمل‌پذیری خطا در $cY(\theta)$

فرض کنید که  $C$  خانواده‌ی همه‌ی ناحیه‌های محدب در صفحه باشد. در این بخش، نشان می‌دهیم که برای هر  $0 < \theta < \pi/3$ ، گراف یائوی پیوسته‌ی  $cY(\theta)$  یک  $t$ -پوشش تحمل‌پذیر  $C$ -خطا برای  $t \geq 1/(1 - 2 \sin(\theta/2))$  است. علاوه بر این، نشان می‌دهیم که برای هر  $\theta \leq \pi$  و هر ناحیه‌ی محدب  $C$ ، گراف  $cY(\theta) \ominus C$  همبند است اگر و فقط اگر گراف  $K_S \ominus C$  همبند باشد. ما به لم زیر نیاز داریم.

لم ۲-۵-۱ ([۲]). گراف هندسی  $G$  روی  $S$  یک  $t$ -پوشش تحمل‌پذیر  $C$ -خطا است اگر و فقط اگر یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}$ -خطا باشد که در آن  $C$  خانواده‌ی همه‌ی ناحیه‌های محدب در صفحه



و  $\mathcal{H}$  خانواده‌ی همه‌ی نیم‌صفحه‌ها است.

حال، قضیه‌ی زیر را ثابت می‌کنیم.

**قضیه ۲-۵-۲.** فرض کنید که  $\theta$  و  $t$  دو عدد حقیقی باشند به طوری که  $0 < \theta < \pi/3$  و  $t \geq 1/(1 - 2 \sin(\theta/2))$ . برای هر مجموعه نقطه‌ی  $S$  در صفحه، گراف  $cY(\theta)$  یک  $t$ -پوشش تحمل‌پذیر  $C$ -خطا است که  $C$  خانواده‌ی همه‌ی ناحیه‌های محدب در صفحه است.

**اثبات.** طبق لم ۲-۵-۱، کفایت نشان دهیم که  $cY(\theta)$  یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}$ -خطاست که  $\mathcal{H}$  خانواده‌ی همه‌ی نیم‌صفحه‌ها است. فرض کنید که  $h$  یک نیم‌صفحه‌ی دلخواه در  $\mathcal{H}$  باشد. باید نشان دهیم که برای هر زوج نقطه‌ی  $p, q \in S$  که خارج از  $h$  قرار دارند، یک  $t$ -مسیر بین  $p$  و  $q$  در  $cY(\theta) \ominus h$  وجود دارد. فرض کنید که  $p, q \in S$  دو نقطه خارج از  $h$  باشند. اثبات بر اساس استقرا روی رتبه‌ی فاصله‌ی اقلیدسی  $|pq|$  است.

**پایه‌ی استقرا:** در صورتی که  $p$  و  $q$  نزدیک‌ترین زوج نقاط در  $S \ominus h$  باشند، بنابراین، یال  $(p, q)$  باید در  $cY(\theta) \ominus h$  باشد (زیرا طبق فرض اولیه‌ی ما، نقاط در حالت کلی قرار دارند یعنی فاصله‌ی بین یک رأس تا رئوس دیگر یکتا است).

**فرض استقرا:** فرض می‌کنیم  $(p, q)$  یک زوج نقطه باشد و برای هر  $a, b \in S \ominus h$  که  $|ab| < |pq|$ ، یک  $t$ -مسیر بین  $a$  و  $b$  در  $cY(\theta) \ominus h$  وجود دارد.

**حکم استقرا:** نشان می‌دهیم که یک  $t$ -مسیر بین  $p$  و  $q$  در  $cY(\theta) \ominus h$  وجود دارد. بدون از دست دادن کلیت، فرض کنید که  $p$  به  $h$  نزدیک‌تر از  $q$  به  $h$  باشد (توجه کنید که ما از ابتدا فرض کرده‌ایم که نقاط در حالت کلی قرار دارند). از آنجایی که  $p$  و  $q$  خارج از  $h$  قرار دارند، بنابراین یک  $\theta$ -مخروط (یک مخروط با دهانه‌ی  $\theta$ ) مانند  $C_p$  به مرکز  $p$  وجود دارد به طوری که  $q \in C_p$  و  $C_p$  نیم‌صفحه‌ی  $h$  را قطع نکند (شکل ۲-۸ را ببینید). فرض کنید که  $r$  نزدیک‌ترین نقطه به  $p$  داخل مخروط  $C_p$  باشد. از آنجایی که  $0 < \theta < \pi/3$ ، داریم  $1 - 2 \sin(\theta/2) > 0$  و از آنجایی که  $|pr| \leq |pq|$ ، طبق لم ۲-۲-۱، داریم  $|rq| < |pq|$ . بنابراین، می‌توانیم فرض استقرا را روی  $(r, q)$  بکار ببریم و یک  $t$ -مسیر  $Q$  از  $r$  به  $q$  در  $cY(\theta) \ominus h$  بدست آوریم. حال، مسیر  $P := \{(p, r)\} \cup Q$  را در نظر بگیرید. به

وضوح،  $P$  یک مسیر از  $p$  به  $q$  در  $h \ominus cY(\theta)$  است. با توجه به لم ۲-۲-۱، داریم:

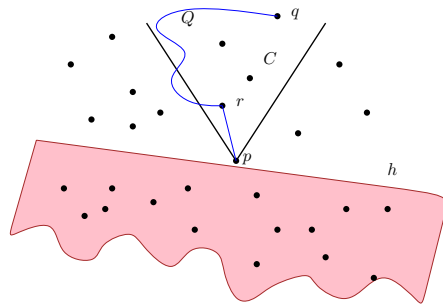
$$\begin{aligned} |P| &= |pr| + |Q| \leq |pr| + t|rq| \leq |pr| + t(|pq| - (1 - 2\sin(\theta/2))|pr|) \\ &= t|pq| + (1 - t(1 - 2\sin(\theta/2)))|pr| \leq t|pq|. \end{aligned}$$

نامساوی آخر در رابطه‌ی بالا، از رابطه‌ی  $t \geq 1/(1 - 2\sin(\theta/2))$  نتیجه می‌شود. بنابراین،  $P$  یک  $t$ -مسیر از  $p$  به  $q$  در  $h \ominus cY(\theta)$  است. ■

در ادامه، ما همبندی  $C \ominus cY(\theta)$  که  $C$  یک ناحیه‌ی محدب در صفحه است را بررسی می‌کنیم. از آنجایی که برای هر  $\theta < \pi/3$ ، گراف  $cY(\theta)$  یک پوشش تحمل‌پذیر ناحیه-خطا است، بنابراین  $C \ominus cY(\theta)$  همبند است اگر و فقط اگر  $K_S \ominus C$  همبند باشد. در ادامه، نشان می‌دهیم که برای  $\pi/3 \leq \theta \leq \pi$ ، گراف  $C \ominus cY(\theta)$  همبند است. ابتدا، لم زیر را ثابت می‌کنیم.

**لم ۲-۵-۳.** برای هر نیم‌صفحه‌ی  $h$  و هر زاویه‌ی  $0 < \theta \leq \pi$ ، گراف  $h \ominus cY(\theta)$  همبند است.

**اثبات.** فرض کنید که  $p$  دورترین رأس از  $h$  که خارج از  $h$  قرار دارد، باشد. نشان می‌دهیم که از هر رأس خارج از  $h$  یک مسیر به  $p$  وجود دارد. با اتصال مسیرها از رئوس مختلف، یک مسیر بین هر زوج از رئوس خارج از  $h$  بدست می‌آوریم که این لم را ثابت می‌کند. حال، فرض کنید که  $v$  یک رأس خارج از  $h$  باشد. اگر  $v$  دورترین رأس از  $h$  نباشد، آنگاه فرض کنیم که  $L$  یک خط موازی با مرز  $h$  باشد که از  $v$  می‌گذرد. از آنجایی که رئوسی وجود دارند که فاصله‌ی آن‌ها از  $h$  نسبت به فاصله‌ی  $v$  تا  $h$  بیش‌تر است، بنابراین، نیم‌صفحه‌ای مانند  $g$  که مرز آن  $L$  باشد و شامل  $h$  نباشد وجود دارد که حتماً شامل رأسی در  $h \ominus cY(\theta)$  باشد. بنابراین،  $g$  شامل یک  $\theta$ -مخروط غیرتهی به مرکز  $v$  است. فرض کنید که  $u$  نزدیک‌ترین رأس به  $v$  در این  $\theta$ -مخروط باشد. به وضوح،  $(u, v)$  یک یال در  $h \ominus cY(\theta)$  است. حال، با حرکت از  $v$  به سمت  $u$  و تکرار این فرایند، از دور می‌شویم تا زمانی که به یک رأس مانند  $v'$  که در دورترین فاصله از  $h$  قرار دارد، برسیم. در این لحظه، توجه کنید که همه‌ی رئوسی که در دورترین فاصله از  $h$  قرار دارند باید روی یک خط  $\ell$  موازی با  $h$  قرار گیرند. فرض کنید که  $v' \neq t$ . بنابراین، مخروط به مرکز  $v'$  که  $h$  را قطع نمی‌کند و همچنین یکی از مرزهایش منطبق بر خط  $\ell$  و در جهت  $p$  است را در نظر بگیرید. این مخروط را اندکی بچرخانید



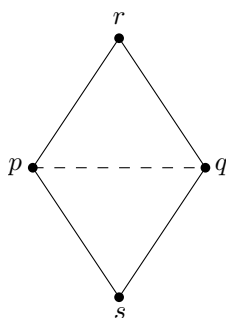
شکل ۲-۸: اثبات قضیه ۲-۵-۲.

تا  $p$  را شامل شود ولی شامل هیچ رأسی در  $h$  نباشد. نزدیکترین رأس در این مخروط به  $w'$  رأس بعد از  $v'$  روی  $\ell$  و در جهت  $p$  است. ما این رأس را  $w$  می‌نامیم. با حرکت از  $v'$  به سمت  $w$  و تکرار این فرایند، در نهایت به  $p$  می‌رسیم. این نشان می‌دهد که  $cY(\theta) \ominus h$  همبند است. ■

**قضیه ۲-۵-۴.** برای هر ناحیه‌ی محدب  $C$  و هر زاویه‌ی  $\theta \leq \pi$ ، گراف  $cY(\theta) \ominus C$  همبند است اگر و فقط اگر  $K_S \ominus C$  همبند باشد که  $K_S$  گراف کامل هندسی روی  $S$  است.

**اثبات.** فرض کنید که  $C$  یک ناحیه‌ی محدب دلخواه در صفحه باشد. از آنجایی که  $cY(\theta)$  زیرگرافی از  $K_S$  است، بنابراین  $cY(\theta) \ominus C$  زیرگرافی از  $K_S \ominus C$  است. بنابراین، اثبات یک طرف قضیه واضح است: از همبندی  $cY(\theta) \ominus C$  می‌توان همبندی  $K_S \ominus C$  را نتیجه گرفت. حال، طرف دیگر قضیه را ثابت می‌کنیم. بدین منظور، نشان می‌دهیم که بین هر زوج رأسی که در  $K_S \ominus C$  توسط یک یال بهم وصل هستند، یک مسیر در  $cY(\theta) \ominus C$  وجود دارد. اتصال این مسیرها، بین هر دو رأسی که در  $K_S \ominus C$  توسط یک مسیر به هم متصل هستند، یک مسیر در  $cY(\theta) \ominus C$  ایجاد می‌کند و اثبات را کامل می‌کند.

فرض کنید که  $(u, v)$  یک یال در  $K_S \ominus C$  باشد. بنابراین یال  $(u, v)$  ناحیه‌ی  $C$  را قطع نمی‌کند. از آنجایی که هر دو چندضلعی محدب نامتقاطع، می‌توانند توسط یک خط، مجزا شوند، بنابراین یک نیم‌صفحه‌ی  $h$  وجود دارد که شامل  $C$  باشد ولی  $u$  و  $v$  در آن نیم‌صفحه نباشد. با توجه به لم ۲-۵-۳،  $cY(\theta) \ominus h$  همبند است، بنابراین یک مسیر از  $u$  به  $v$  در  $cY(\theta)$  وجود دارد که به طور کامل خارج از  $h$  قرار دارد. از آنجایی که  $h$  شامل  $C$  است، این مسیر در  $cY(\theta) \ominus C$  نیز وجود دارد. بنابراین، یک مسیر بین هر دو رأس انتهایی هر یال  $K_S \ominus C$  در  $cY(\theta) \ominus C$  وجود دارد که



شکل ۲-۹: یک مجموعه نقطه که  $Y_4$  روی آن خود-گرا نیست اما  $cY(\pi/2)$  روی آن خود-گرا است. یال نقطه‌چین متعلق به  $Y_4$  نیست ولی متعلق به  $cY(\pi/2)$  است.

■

این قضیه را ثابت می‌کند.

## ۲-۶ خود-گرایی گراف‌های یائوی پیوسته

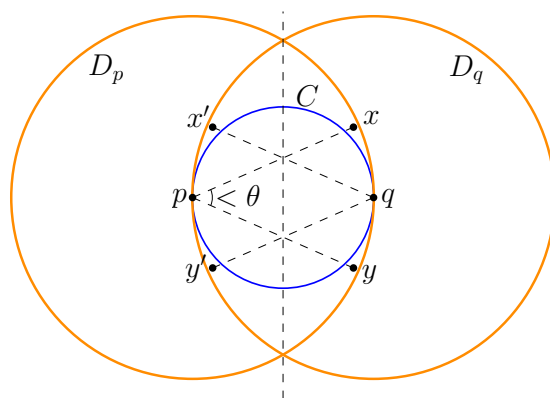
ممکن است این طور تصور شود که گراف‌های یائو، خود-گرا هستند. اما مثال‌های نقضی برای این موضوع وجود دارد. حال، سؤالی که ممکن است مطرح شود این است که آیا یک  $\theta$  وجود دارد که  $cY(\theta)$  روی هر مجموعه از نقطه در صفحه، خود-گرا یا وتر-افزایشی باشد؟ شکل ۲-۹ مثالی از مجموعه‌ای از چهار نقطه‌ی  $\{p, q, r, s\}$  را نشان می‌دهد که در آن گراف یائوی  $Y_4$  خود-گرا نیست اما  $cY(\pi/2)$  خود-گرا است. توجه کنید که در این شکل، این چهار نقطه، رئوس یک متوازی‌الاضلاع هستند که فقط اندکی جابجا شده‌اند که هیچ دو فاصله‌ای یکی نباشد. همه‌ی چهار یال متوازی‌الاضلاع متعلق به  $cY(\pi/2)$  و  $Y_4$  هستند اما قطر کوچک این متوازی‌الاضلاع یعنی  $(p, q)$ ، متعلق به  $cY(\pi/2)$  است ولی متعلق به  $Y_4$  نیست. اما عدم وجود یال  $(p, q)$  در  $Y_4$  باعث می‌شود که هر مسیر بین  $p$  و  $q$  در  $Y_4$ ، خود-گرا نباشد. این نشان می‌دهد که  $Y_4$  خود-گرا نیست و به راحتی می‌توان بررسی کرد که  $cY(\pi/2)$  روی این مجموعه نقطه، خود-گرا است. حال، در ادامه نشان می‌دهیم که گراف  $cY(\theta)$ ، همیشه خود-گرا نیست.

قضیه ۲-۶-۱. برای هر  $\theta > 0$ ، مجموعه‌ای از نقاط وجود دارد به طوری که  $cY(\theta)$  روی آن خود-گرا نیست.

**اثبات.** ما قضیه را برای  $0 < \theta \leq 2\pi/3$  ثابت می‌کنیم. از آنجایی که اگر  $\theta \geq 2\pi/3$ ، آنگاه  $cY(\theta) \subseteq cY(2\pi/3)$ ، بنابراین، قضیه برای هر  $\theta > 0$ ، نتیجه می‌شود.

برای ساخت مجموعه نقطه، دو نقطه‌ی  $p = (0, 0)$  و  $q = (1, 0)$  را در نظر بگیرید. فرض کنید که  $C$  دایره‌ای به قطر یک باشد که از  $p$  و  $q$  عبور می‌کند و  $D_p$  و  $D_q$  دایره‌هایی به ترتیب به مرکز  $p$  و  $q$  و به شعاع یک باشند (شکل ۲-۱۰ را ببینید). همچنین، فرض کنید که  $x$  و  $y$  دو نقطه، خارج از  $C$  و داخل ماه  $p$  و  $q$  باشند به طوری که  $\angle(xp, pq) < \theta/2$  و  $\angle(yp, pq) < \theta/2$  باشد و  $x'$  و  $y'$  تصویر آینه‌ای  $x$  و  $y$  نسبت به عمود منصف  $(p, q)$  باشند.

حال،  $cY(\theta)$  را روی این مجموعه نقطه در نظر بگیرید. از آنجایی که  $\angle(xp, py) = \angle(x'q, qy')$ ،  $cY(\theta)$  شامل یال  $(p, q)$  نمی‌باشد، زیرا هر  $\theta$ -مخروط به مرکز  $p$  که شامل  $q$  باشد، حداقل یکی از نقاط  $x$  و  $y$  را که به  $p$  در مقایسه با  $q$  نزدیک‌تر هستند، در بر می‌گیرد. به طور مشابه، هر  $\theta$ -مخروط به مرکز  $q$  که شامل  $p$  باشد باید حداقل یکی از  $x'$  و  $y'$  را که به  $q$  نسبت به  $p$  نزدیک‌تر هستند، در بر بگیرد. به علاوه، براساس قضیه‌ی تالس<sup>۱</sup>، هیچ یک از یال‌های  $(p, x)$ ،  $(p, y)$ ،  $(p, x')$  و  $(p, y')$  نمی‌توانند بخشی از مسیر خود-گرا از  $p$  به  $q$  باشند. از آنجایی که این یال‌ها تنها یال‌هایی هستند که در  $cY(\theta)$  وجود دارند، بنابراین در  $cY(\theta)$  هیچ مسیر خود-گرا از  $p$  به  $q$  و یا بالعکس وجود ندارد. این نشان می‌دهد که  $cY(\theta)$  خود-گرا نیست. ■



شکل ۲-۱۰: اثبات قضیه‌ی ۲-۶-۱.

<sup>۱</sup>Thales

## ۷-۲ نتیجه‌گیری

ما نوع جدیدی از گراف‌های هندسی به نام گراف یائوی پیوسته را معرفی کردیم و ضریب کشش، تحمل‌پذیری خطا و خاصیت خود-گرایی آن‌ها را بررسی کردیم. ما نشان دادیم که برای هر  $0 < \theta \leq 2\pi/3$ ، گراف یائوی پیوسته‌ی  $cY(\theta)$  یک  $t$ -پوشش است که در آن  $t$  یک مقدار ثابت می‌باشد، ولی برای  $\pi \leq \theta \leq 2\pi$  این‌گونه نیست. همچنین نشان دادیم که برای هر  $0 < \theta \leq \pi$ ، گراف  $cY(\theta)$  همبند است ولی برای هر  $\theta > \pi$  لزوماً همبند نیست. ما همچنین این خصوصیات را در مدل تحمل‌پذیری ناحیه-خطا نیز بررسی کردیم و نشان دادیم که برای هر  $\theta < \pi/3$ ، گراف  $cY(\theta)$  یک  $t$ -پوشش باقی می‌ماند ( $t$  مقدار ثابت و وابسته به  $\theta$  است) حتی اگر ناحیه‌ی محدبی از آن حذف شود. همچنین نشان دادیم که  $cY(\theta)$  برای هر  $\theta \leq \pi$  همبند باقی می‌ماند حتی اگر بخش محدبی از آن حذف شود.

در زیر دو سؤال به عنوان مسئله‌ی حل نشده، مطرح می‌کنیم.

۱. آیا  $cY(\theta)$  برای  $2\pi/3 < \theta < \pi$  یک  $t$ -پوشش که  $t$  مقدار ثابتی باشد، است؟

۲. آیا  $cY(\theta)$  برای  $\pi/3 \leq \theta \leq \pi$  یک  $t$ -پوشش تحمل‌پذیر  $C$ -خطا که  $t$  یک مقدار ثابتی باشد،

است؟

## فصل ۳

بهبود پیچیدگی زمانی و حافظه‌ی پوشش شکاف-حریصانه

یکی از الگوریتم‌های حریصانه‌ی ساخت پوشش، الگوریتم شکاف-حریصانه است که توسط آریا و همکارانش [۹] معرفی شد. این الگوریتم، مبتنی بر خاصیت شکاف<sup>۱</sup> است که توسط چاندرا<sup>۲</sup> و همکارانش [۲۲] معرفی شد. فرض کنید که  $w \geq 0$  یک عدد حقیقی باشد و  $E$  مجموعه‌ی یال‌های جهت‌دار در  $\mathbb{R}^d$  باشد. گفته می‌شود که  $E$  خاصیت  $w$ -شکاف دارد اگر برای هر دو یال مجزای (جهت‌دار)  $(p, q)$  و  $(r, s)$  در  $E$  داشته باشیم  $|pr| > w \times \min(|pq|, |rs|)$  و همچنین گفته می‌شود که  $E$  خاصیت  $w$ -شکاف قوی دارد اگر برای هر دو یال مجزای (جهت‌دار)  $(p, q)$  و  $(r, s)$  در  $E$  داشته باشیم  $|qs| > w \times \min(|pq|, |rs|)$  و  $|pr| > w \times \min(|pq|, |rs|)$ .

خاصیت شکاف از این جهت حائز اهمیت است که می‌توان از آن برای پیدا کردن یک کران بالا روی وزن یک گراف هندسی استفاده کرد. چاندرا و همکارانش [۲۲] نشان دادند که اگر مجموعه‌ی یالی یک گراف هندسی جهت‌دار  $G(S, E)$  خاصیت شکاف داشته باشد، آنگاه داریم  $wt(E) = O(\log n) \cdot wt(MST(S))$  آن‌ها همچنین ثابت کردند که اگر مجموعه‌ی یالی  $G$  خاصیت  $w$ -شکاف قوی داشته باشد، آنگاه برای هر رأس  $v$  در  $G$ ، درجه‌ی ورودی و درجه‌ی خروجی رأس  $v$  حداکثر یک است.

در سال ۱۹۹۷، آریا و همکارانش [۹] الگوریتم شکاف-حریصانه را جهت ساخت یک پوشش روی هر مجموعه‌ای از نقاط در صفحه معرفی کردند که به گراف حاصل از این الگوریتم، پوشش شکاف-حریصانه می‌گویند (الگوریتم ۱-۱-۳ را ببینید [۴۴]). آن‌ها نشان دادند که پوشش شکاف حریصانه دارای ضریب کشش ثابت و کوچک است. همچنین آن‌ها نشان دادند که مجموعه‌ی یالی پوشش شکاف-حریصانه می‌تواند به تعداد ثابتی زیرمجموعه افراز شود به طوری که یال‌های هر افراز، خاصیت شکاف-حریصانه‌ی قوی داشته باشند. از این رو، با بکاربردن نتایج چاندرا و همکارانش، می‌توان نشان داد که هر رأس از پوشش شکاف-حریصانه‌ی تولید شده توسط الگوریتم  $GAPGREEDY(S, \theta, w)$  (الگوریتم ۱-۱-۳ را ببینید) دارای درجه‌ی حداکثر  $\lceil 2\pi/\theta \rceil$  است [۲۲].

همچنین، اگر  $w > 0$ ، آنگاه وزن پوشش شکاف-حریصانه کم‌تر از  $\lceil 2\pi/\theta \rceil (1 + 2/w) \log n$  برابر

<sup>۱</sup>gap property

<sup>۲</sup>Chandra



است. فرض کنید که  $\theta$  و  $w$  دو عدد حقیقی باشند به طوری که  $0 < \theta < \pi/4$  و  $0 \leq w < (\cos \theta - \sin \theta)/2$ . می توان ثابت کرد که گراف تولید شده توسط الگوریتم  $\text{GAPGREEDY}(S, \theta, w)$  یک  $t$ -پوشش است به طوری که  $t = 1/(\cos \theta - \sin \theta - 2w)$  [۹]. به وضوح، پیچیدگی زمانی و حافظه‌ی الگوریتم  $\text{GAPGREEDY}(S, \theta, w)$  به ترتیب  $O(n^3)$  و  $O(n^2)$  است. بر طبق اطلاعاتی که تاکنون داریم، هیچ الگوریتمی با پیچیدگی زمانی  $o(n^3)$  برای ساخت پوشش شکاف-حریصانه وجود ندارد. آریا و اسمید [۹]، الگوریتمی به نام الگوریتم شکاف-حریصانه‌ی اصلاح شده را ارائه کردند که برای هر مجموعه‌ای از  $n$  نقطه در صفحه، یک شبکه‌ی پوششی هندسی می‌سازد که از دیدگاه نظری خصوصیاتش مشابه خصوصیات پوشش شکاف-حریصانه است و دارای پیچیدگی زمانی و حافظه‌ی (هر دو)  $O(n \log^2 n)$  است. ما تأکید می‌کنیم که خروجی الگوریتم شکاف-حریصانه‌ی اصلاح شده با خروجی الگوریتم شکاف-حریصانه‌ی اصلی، لزوماً یکی نیست.

---

**Algorithm ۳-۱-1:**  $\text{GAPGREEDY}(S, \theta, w)$

---

```

/* This algorithm takes as input a set  $S$  of  $n$  points in the
plane, and two real numbers  $\theta$  and  $w$  such that  $0 < \theta < \pi/4$ ,
 $0 \leq w < (\cos \theta - \sin \theta)/2$ . The algorithm returns a directed
 $t$ -spanner  $G = (S, E)$ , for  $t = 1/(\cos \theta - \sin \theta - 2w)$ . */
1 Sort the  $2 \binom{n}{2}$  ordered pairs of distinct points in non-decreasing order of their
distances (ties are broken arbitrarily), and store them in a list  $L$ ;
2  $E := \emptyset$ ;
3 foreach ordered pair  $(p, q) \in L$  do
4    $\text{add} := \text{true}$ ;
5   foreach edge  $(r, s) \in E$  do
6     /* Let  $\vec{a}$  be a vector from the origin to the point  $a$ . We
define  $\text{angle}(pq, rs)$  to be the angle between the vectors
 $\vec{q-p}$  and  $\vec{s-r}$ . */
7     if  $\text{angle}(pq, rs) \leq \theta$  then
8        $\text{add} := \text{add} \wedge (|pr| > w|rs|) \wedge (|qs| > w|rs|)$ ;
9     end
10  if  $\text{add} = \text{true}$  then
11     $E := E \cup (p, q)$ ;
12  end
13 end
14 return  $G = (S, E)$ ;

```

---

پوشش حریصانه را از فصل اول به خاطر آورید. اگر چه وزن پوشش حریصانه از دیدگاه نظری نسبت به وزن پوشش شکاف-حریصانه بهتر است، اما بر اساس آخرین اطلاعاتی که داریم،

تاکنون هیچ الگوریتمی با زمان اجرای  $O(n^2 \log n)$  برای ساخت پوشش حریصانه ارائه نشده است. در این فصل، یک الگوریتم برای ساخت پوشش شکاف-حریصانه ارائه می‌کنیم که به  $O(n^2)$  زمان و  $O(n^2)$  حافظه، نیاز دارد. بر اساس آخرین اطلاعاتی که داریم، این الگوریتم، اولین الگوریتم ساخت پوشش شکاف-حریصانه است که به  $O(n^3)$  زمان نیاز دارد. همچنین، با اعمال تغییراتی روی این الگوریتم، الگوریتم دیگری برای ساخت پوشش شکاف-حریصانه ارائه می‌کنیم که به  $O(n^3)$  زمان و به  $O(n)$  حافظه نیاز دارد. ما در این فصل، ضریب کشش پوشش شکاف-حریصانه را از  $1/(\cos \theta - \sin \theta - 2w)$  به  $1/(1 - 2 \sin(\theta/2) - 2w)$  بهبود می‌دهیم که این عمل، باعث می‌شود که محدودیت زاویه برای الگوریتم از حد بالای ۴۵ درجه به حد بالای ۶۰ درجه بهبود داده شود. خواهیم دید که به راحتی می‌توانیم نتایج را به ابعاد بالاتر تعمیم دهیم. در این فصل، همچنین سه الگوریتم با زمان اجرای  $O(n^3)$ ، جهت ساخت پوشش شکاف-حریصانه ارائه می‌کنیم و زمان اجرای این الگوریتم‌ها را در عمل با زمان اجرای الگوریتم ۳-۱-۱ مقایسه می‌کنیم. در ابتدای این تحقیق، امیدوار بودیم که مجموعه نقاطی در صفحه وجود داشته باشد که زمان ساخت پوشش حریصانه برای آنها نسبت به پوشش شکاف-حریصانه بیش‌تر باشد (با فرض یکی بودن خواص نظری هر دو پوشش)، اما در نهایت، نتایج تجربی، این موضوع را تأیید نکرد.

### ۲-۳ بهبود پیچیدگی حافظه و زمان

در این بخش، ابتدا الگوریتمی جهت ساخت پوشش شکاف-حریصانه ارائه می‌کنیم که دارای پیچیدگی از زمانی مرتبه‌ی سه و پیچیدگی حافظه‌ی خطی است. در انتها، تغییراتی را روی این الگوریتم انجام می‌دهیم که پوشش شکاف-حریصانه را در زمانی از مرتبه‌ی دو و با کمک حافظه‌ای از مرتبه‌ی دو، می‌سازد.

### ۱-۲-۳ بهبود پیچیدگی حافظه

الگوریتم اول، مبتنی بر مفهوم تجزیه‌ی زوجی خوش-مجزا (WSPD) است (بخش ۱-۲-۳ را ببینید). الگوریتمی که ما در اینجا ارائه می‌کنیم مشابه الگوریتمی است که توسط آلواینز و همکارانش [۵]، جهت ساخت پوشش حریصانه با حافظه‌ی خطی ارائه شد. در ابتدا، به طور خلاصه، الگوریتم را توصیف می‌کنیم. الگوریتم از این خاصیت استفاده می‌کند که پوشش شکاف-حریصانه در هر زوج WSPD، شامل تعداد ثابتی یال است (این خاصیت را بعداً نشان خواهیم داد). بنابراین، الگوریتم ابتدا یک WSPD با اندازه‌ی خطی و با بکاربردن یک حافظه‌ی خطی، می‌سازد. سپس الگوریتم سعی می‌کند که برای هر زوج WSPD، یال‌هایی از پوشش شکاف-حریصانه را پیدا کند که داخل آن زوج قرار دارند. بدین منظور، الگوریتم، همه‌ی زوج‌های WSPD را بر اساس فاصله‌ی بین جعبه‌های احاطه‌گر آن‌ها، از کوچک به بزرگ مرتب می‌کند. سپس، برای هر زوج WSPD، زوج نقاطی را در آن زوج WSPD انتخاب می‌کند که کم‌ترین فاصله‌ی اقلیدسی را نسبت به خاصیتی که بعداً معرفی می‌شود، داشته باشد. ما این زوج نقطه را زوج نقطه‌ی انتخابی برای آن زوج WSPD می‌نامیم. حال، فرض کنید که  $(u, v)$  نزدیک‌ترین زوج نقطه در بین تمام زوج نقاط انتخابی در مرحله‌ی قبل باشد. نشان خواهیم داد که  $(u, v)$  و  $(v, u)$ ، یال‌های پوشش شکاف-حریصانه هستند که الگوریتم آن‌ها را به مجموعه‌ی یالی که تاکنون محاسبه شده، اضافه می‌کند. پس از اضافه کردن این یال‌ها به مجموعه‌ی یالی، الگوریتم باید زوج نقاط انتخابی را برای هر زوج WSPD که تحت تأثیر اضافه شدن یال‌هایی که اخیراً اضافه شده است، قرار گرفته اند، بروزرسانی کند. عمل اصلی الگوریتم، این است که بروزرسانی را فقط برای تعداد خاصی از زوج‌های WSPD انجام می‌دهد و نه برای همه‌ی زوج‌ها. لازم به ذکر است که از آنجایی که WSPD نیاز به حافظه‌ی خطی دارد، بنابراین، الگوریتم از این خاصیت، برای بهبود حافظه‌ی مصرفی استفاده می‌کند.

قبل از اینکه به جزئیات الگوریتم بپردازیم، تعدادی لم مهم که بعداً لازم می‌شود را بیان می‌کنیم. در ادامه، فرض می‌کنیم که  $t > 1$ ، یک عدد حقیقی ثابت است.

فرض کنید که  $S$  مجموعه‌ای از  $n$  نقطه در صفحه باشد. همچنین، فرض کنید که  $\binom{n}{2}$  زوج

نقطه را به صورت غیر-نزولی و بر اساس فاصله‌ی اقلیدسی مرتب کرده‌ایم و داخل یک لیست مانند  $L$  ذخیره کرده‌ایم به طوری که برای هر  $i$  که  $0 \leq i \leq \binom{n}{2} - 1$  داریم: اگر  $L[2i] = (p, q)$ ، آنگاه  $L[2i+1] = (q, p)$ ، که در اینجا  $L[i]$  یعنی نهمین درایه‌ی لیست  $L$  که برای سادگی، فرض می‌کنیم که در هر درایه‌ی  $L$  یک زوج از نقاط را ذخیره کرده‌ایم.

در ادامه، فرض می‌کنیم که در گام اول الگوریتم ۱-۱-۳، لیست  $L$  به صورتی که در بالا توضیح داده شد، ذخیره شده است. حال، فرض کنید که  $\theta$  و  $w$  دو عدد حقیقی باشند به طوری که  $0 < \theta < \pi/4$  و  $0 \leq w < (\cos \theta - \sin \theta)/2$  و پوشش شکاف-حریصانه‌ی تولید شده توسط الگوریتم ۱-۱-۳ با ورودی‌های  $\theta$  و  $w$  باشد. حال، لم زیر را داریم:

**لم ۱-۲-۳.**  $(q, p)$  یالی در پوشش شکاف-حریصانه است اگر و فقط اگر  $(p, q)$  در آن وجود داشته باشد.

**اثبات.** فرض کنید که  $i$  یک عدد صحیح باشد به طوری که  $0 \leq i \leq \binom{n}{2} - 1$  و  $L[2i] = (p, q)$  برای اثبات لم، کفایت نشان دهیم که  $L[2i+1] \in E$  اگر و فقط اگر  $L[2i] \in E$ . در اینجا فقط ثابت می‌کنیم که اگر  $L[2i] \in E$ ، آنگاه  $L[2i+1] \in E$ . سمت دیگر قضیه، به طور مشابه اثبات می‌شود. اثبات از طریق استقرا روی  $i$  است.

**پایه‌ی استقرا:** برای  $i = 0$ . با توجه به الگوریتم ۱-۱-۳، به راحتی می‌توان دید که  $L[0] \in E$  و  $L[1] \in E$ .

**فرض استقرا:** فرض کنید که نتایج برای هر  $k < i$  برقرار باشد.

**حکم استقرا:** برای عدد صحیح  $0 < i$  به طوری که  $L[i] = (p, q)$  و  $L[2i+1] = (q, p)$  فرض می‌کنیم که  $L[2i] \in E$ . طبق فرض استقرا، برای هر  $l < i$  داریم: اگر  $L[2l] \in E$ ، آنگاه  $L[2l+1] \in E$ . فرض کنید که  $E_i = \left\{ L[2l], L[2l+1] \mid l < i \text{ و } L[2l] \in E \right\}$ . برای اینکه نشان دهیم  $(q, p) \in E$ ، کفایت نشان دهیم که برای هر  $(r, s) \in E$  داریم:

$$\begin{cases} ۱. \text{ angle}(qp, rs) > \theta, \text{ یا} \\ ۲. \text{ angle}(qp, rs) \leq \theta \text{ و } |qr| > w \times |rs| \text{ و } |ps| > w \times |rs|. \end{cases} \quad (۱-۳)$$

توجه کنید که  $\text{angle}(pq, qp) = \pi > \theta$  حال، فرض کنید که  $(r, s)$  یک عضو دلخواه در  $E_i$  باشد. طبق تعریف  $E_i$ ، از آنجایی که  $(r, s) \in E_i$ ، بنابراین  $(s, r) \in E_i$ . حال، از آنجایی که  $(p, q) \in E$  و طبق تعریف  $E_i$ ، برای یال  $(s, r)$  داریم:

$$\begin{cases} ۱. \text{angle}(pq, sr) > \theta, \text{ یا,} \\ ۲. \text{angle}(pq, sr) \leq \theta \text{ و } |ps| > w \times |rs| \text{ و } |qr| > w \times |rs|. \end{cases} \quad (۲-۳)$$

به راحتی می‌توان فهمید که  $\text{angle}(pq, sr) = \text{angle}(qp, rs)$  از این رو، طبق معادله‌ی

۲-۳، به راحتی معادله‌ی ۱-۳ را نتیجه می‌گیریم. ■

برای دو عدد حقیقی  $\theta$  و  $w$  که  $0 < \theta < \pi/4$  و  $0 \leq w < (\cos \theta - \sin \theta)/2$ ، تعریف می‌کنیم:

$$s := \max \left( \frac{4}{\sin(\theta/2)}, \frac{2}{w} \right). \quad (۳-۳)$$

به وضوح،  $2/s \leq w$  و  $4/s \leq \sin(\theta/2)$ .

لم ۳-۲-۲ ([۴۴]). فرض کنید که  $s > 0$  یک عدد حقیقی و  $A$  و  $B$  دو مجموعه‌ی متناهی از نقاط باشند که نسبت به پارامتر  $s$  خوش-مجزا هستند. اگر  $p$  و  $p'$  دو نقطه در  $A$  و  $q$  و  $q'$  دو نقطه در  $B$  باشند. آنگاه

$$|pp'| \leq (2/s)|pq|, \quad ۱.$$

$$|p'q'| \leq (1 + 4/s)|pq|. \quad ۲.$$

$$\sin(\text{angle}(pq, p'q')) \leq 4/s. \quad ۳.$$

حال، فرض کنید که  $\theta$  و  $w$  دو عدد حقیقی دلخواه باشند به طوری که  $0 < \theta < \pi/4$  و

$0 \leq w < (\cos \theta - \sin \theta)/2$  و  $s = \max \left( \frac{4}{\sin(\theta/2)}, \frac{2}{w} \right)$ . فرض کنید که  $S \subseteq \mathbb{R}^2$  مجموعه‌ای از

$n$  نقطه در صفحه و  $\{(A_i, B_i)\}_{i=1}^m$  یک WSPD برای  $S$  نسبت به پارامتر  $s$  باشد. علاوه بر این،

برای یک مجموعه‌ی  $C$  از نقاط، فرض کنید که  $R(C)$  جعبه‌ی احاطه‌گر  $C$  باشد. طول یک زوج

خوش-مجزای  $(A_i, B_i)$ ، را فاصله‌ی بین مراکز دایره‌های حول  $R(A_i)$  و  $R(B_i)$  تعریف می‌کنیم و

با نماد  $\ell(A_i, B_i)$  نمایش می‌دهیم.

لم ۳-۲-۳ ([۵]). فرض کنید که  $\gamma$  و  $\ell$  دو عدد حقیقی مثبت باشند و  $(A_i, B_i)$  یک زوج خوش-مجزا نسبت به پارامتر  $s$  در  $WSPD$  و  $\ell(A_i, B_i) = \ell$  باشد. تعداد زوج‌های خوش-مجزای  $(A'_i, B'_i)$  که طولشان در بازه‌ی  $[\ell/2, 2\ell]$  باشد و فاصله‌ی حداقل یکی از  $R(A'_i)$  و  $R(B'_i)$  تا  $R(A_i)$  یا  $R(B_i)$  کم‌تر یا مساوی  $\gamma\ell$  باشد، برابر با  $O(s^d(1 + \gamma s)^d)$  است.

فرض کنید که  $G(S, E)$  پوشش شکاف-حریصانه‌ی تولید شده توسط الگوریتم ۳-۱-۱ با توجه به پارامترهای  $\theta$  و  $w$  باشد. حال، لم زیر را داریم:

لم ۴-۲-۳. برای هر  $i$  که  $1 \leq i \leq m$ ، پوشش شکاف-حریصانه یا دقیقاً شامل دو یال  $(p, q)$  و  $(q, p)$  است به طوری که  $(p, q) \in (A_i \times B_i) \cup (B_i \times A_i)$  و یا شامل هیچ یالی در  $(A_i \times B_i) \cup (B_i \times A_i)$  نیست.

**اثبات.** ابتدا، ادعا می‌کنیم که  $|E \cap (A_i \times B_i)| \leq 1$  و  $|E \cap (B_i \times A_i)| \leq 1$ . ما حالت  $|E \cap (A_i \times B_i)| \leq 1$  را اثبات می‌کنیم و حالت دیگر به صورت مشابه اثبات می‌شود. فرض کنید که  $|E \cap (A_i \times B_i)| \geq 2$ . از این رو، پوشش شکاف-حریصانه شامل دو یال  $e = (x, y)$  و  $e' = (x', y')$  است که  $x, x' \in A_i$  و  $y, y' \in B_i$  بدون از دست دادن کلیت، فرض کنید که  $|xy| \leq |x'y'|$ ، بنابراین  $e$  قبل از  $e'$  توسط الگوریتم ۳-۱-۱ در نظر گرفته می‌شود. بر طبق لم ۳-۲-۲، داریم  $\sin(\text{angle}(x'y', xy)) \leq 4/s$ . همچنین، از آنجایی که  $s = \max\left(\frac{4}{\sin(\theta/2)}, \frac{2}{w}\right)$  داریم  $4/s \leq \sin(\theta/2)$  و بنابراین، از آنجایی که  $\text{angle}(x'y', xy) \leq \theta/2 \leq \theta$  داریم  $0 < \theta < \pi/4$ .

از طرف دیگر، بر طبق لم ۳-۲-۲، داریم  $|xx'| \leq (2/s)|xy|$ . همچنین، از آنجایی که  $s = \max\left(\frac{4}{\sin(\theta/2)}, \frac{2}{w}\right)$  داریم  $2/s \leq w$ . با ترکیب این دو معادله، داریم  $|xx'| \leq w|xy|$ . از این رو، بر طبق الگوریتم ۳-۱-۱، از آنجایی که الگوریتم،  $e'$  را بعد از  $e$  بررسی می‌کند، هرگز  $e'$  به  $E$  اضافه نمی‌شود، که این یک تناقض است و در نتیجه، ادعای ما ثابت می‌شود. بنابراین، نتیجه می‌گیریم که  $|E \cap ((A_i \times B_i) \cup (B_i \times A_i))| \leq 2$ .

بر طبق لم ۳-۲-۱، داریم  $|E \cap ((A_i \times B_i) \cup (B_i \times A_i))| \neq 1$  که این، لم را ثابت می‌کند. ■

از لم ۴-۲-۳، نتیجه‌ی زیر بدست می‌آید:

نتیجه ۳-۲-۵. پوشش شکاف-حریصانه حداکثر شامل  $O(s^d n)$  یال است.

فرض کنید که  $E$  مجموعه‌ای از یال‌ها روی مجموعه نقطه‌ی  $S$  باشد. برای هر  $i$  با  $1 \leq i \leq m$ ، فرض کنید که  $C_i$  مجموعه‌ی همه‌ی زوج نقاط  $(u, v)$  که  $(u, v) \in (A_i \times B_i) \cup (B_i \times A_i)$  باشد به طوری که برای هر  $(x, y) \in E$  داریم:

$$1. \text{angle}(uv, xy) > \theta, \text{ یا}$$

$$2. \text{angle}(uv, xy) \leq \theta \text{ و } |ux| > w \times \min(|uv|, |xy|) \text{ و } |vy| > w \times \min(|uv|, |xy|)$$

حال، فرض کنید که  $(u_i, v_i) \in C_i$  یک زوج نقطه با کم‌ترین فاصله‌ی اقلیدسی باشد. ما چنین زوجی را زوج انتخابی  $(A_i, B_i)$  نسبت به  $E$  می‌نامیم.

لم ۳-۲-۶. برای هر  $1 \leq i \leq m$ ، زوج انتخابی  $(A_i, B_i)$  می‌تواند در زمان  $O(|A_i||B_i||E|)$  و با بکار بردن  $O(|E|)$  حافظه، محاسبه شود.

**اثبات.** فرض کنید که  $\text{SELECTEDPAIR}(i)$  تابعی باشد که یک عدد صحیح  $1 \leq i \leq m$  را از ورودی می‌گیرد و سپس با بررسی همه‌ی زوج‌های  $(p, q) \in A_i \times B_i$  و شرط مربوط به یالها که در بالا ذکر شد، زوج انتخابی  $(A_i, B_i)$  را پیدا می‌کند و بر می‌گرداند. به وضوح، پیچیدگی زمانی این تابع  $O(|A_i||B_i||E|)$  و پیچیدگی حافظه‌ی آن  $O(|E|)$  است. ■

حال، به جزئیات الگوریتم اول (ساخت پوشش شکاف-حریصانه با حافظه‌ی خطی) می‌پردازیم. این الگوریتم، مشابه الگوریتمی است که برای ساخت پوشش حریصانه با حافظه‌ی خطی ارائه شده است [۵]. الگوریتمی که می‌خواهیم ارائه کنیم دارای سه تابع است:  $\text{SELECTEDPAIR}$ ،  $\text{ADDTOQUEUE}$  (الگوریتم ۳-۲-۱) و  $\text{LINEARSPACEGAPGREEDYSPANNER}$  (الگوریتم ۳-۲-۲). الگوریتم ۳-۲-۲، تابع اصلی می‌باشد.

در ادامه، وقتی می‌گوییم "الگوریتم" منظورمان الگوریتم ۳-۲-۲ است. الگوریتم، با یک مجموعه‌ی تهی  $E$  و یک صف خالی  $Q$  شروع می‌شود. در ابتدا، الگوریتم یک  $\text{WSPD}$   $\{(A_i, B_i)\}_{i=1}^m$  برای مجموعه نقطه‌ی  $S$  با پارامتر  $s = \max\left(\frac{4}{\sin(\theta/2)}, \frac{2}{w}\right)$  می‌سازد و سپس زوج‌های  $(A_i, B_i)$  را به صورت صعودی و بر اساس  $\min(A_i, B_i)$  مرتب می‌کند (برای هر دو مجموعه‌ی  $A_i$  و  $B_i$ ،

$\min(A_i, B_i)$  طول کم‌ترین فاصله‌ی بین دو دایره که حول جعبه‌های احاطه‌گر  $A_i$  و  $B_i$  قرار دارند، است.  $\max(A_i, B_i)$  به طور مشابه تعریف می‌شود. سپس الگوریتم، در شروع کار، تابع  $\text{AddToQueue}(i)$  را با  $i = 1$  فراخوانی می‌کند. پس از فراخوانی این تابع، زوج‌های انتخابی زوج‌های خوش-مجزای  $(A_i, B_i)$  براساس  $\min(A_i, B_i)$  در صف اولویت  $Q$  ذخیره می‌شوند. توجه کنید که در هر فراخوانی این تابع، اضافه کردن به صف تا وقتی ادامه می‌یابد که  $\min(A_i, B_i) \leq \min(Q)$  یا  $Q$  تهی باشد.

---

**Algorithm ۳-۲-1: AddToQueue( $Q, i$ )**

---

**Input:** Queue  $Q$  and integer  $i$  with  $1 \leq i \leq m$ .

```

1 while  $i \leq m$ , and either  $Q$  is empty or  $\min(A_i, B_i) \leq \min(Q)$  do
2    $x := \text{SELECTEDPAIR}(i)$ ;
3   if  $x$  is not nil, but a directed pair  $(u, v)$  then
4     Add  $(u, v)$  to  $Q$  with key  $|uv|$ , and associate this entry with  $(A_i, B_i)$ ;
5   end
6    $i = i + 1$ ;
7 end

```

---

پس از برگشت از تابع  $\text{AddToQueue}$  به تابع اصلی، الگوریتم، کوچک‌ترین مقدار  $Q$  (خط ۹ در الگوریتم ۳-۲-۲ را ببینید) را استخراج می‌کند. فرض کنید که زوج نقطه‌ی مطابق با کوچک‌ترین مقدار  $Q$  باشد که از  $Q$  استخراج می‌شود. واضح است که  $(u, v)$  یک یال در پوشش شکاف-حریصانه است. همچنین، طبق لم ۳-۲-۴،  $(v, u)$  یال دیگری از پوشش شکاف-حریصانه است. بنابراین، الگوریتم این دو یال را به مجموعه‌ی یالی  $E$  که تاکنون محاسبه شده است، اضافه می‌کند. توجه کنید که، پس از استخراج، الگوریتم، درایه‌ی متناظر با کوچک‌ترین عضو در  $Q$  را حذف می‌کند.

حال، الگوریتم باید  $Q$  را بروزرسانی کند. بروزرسانی باید با دقت انجام شود (خطوط ۱۱-۱۴ در الگوریتم ۳-۲-۲ را ببینید). به طور کلی، وقتی که یال‌های  $(u, v)$  و  $(v, u)$  به  $E$  اضافه می‌شوند، الگوریتم، همه‌ی درایه‌های  $Q$  را بروزرسانی نمی‌کند و فقط تعداد خاصی از درایه‌های  $Q$  را که زوج‌های انتخابی آن‌ها تحت تأثیر اضافه شدن یال‌های  $(u, v)$  و  $(v, u)$  قرار گرفته‌اند، بروزرسانی می‌شوند. توجه کنید که در خط ۱ از الگوریتم ۳-۲-۱، ممکن است که وقتی حلقه‌ی **while** پایان می‌یابد، مقدار  $i$  کم‌تر از  $m$  باشد. از این رو، همه‌ی زوج‌های خوش-مجزا که اندیس آن‌ها بزرگ‌تر



---

**Algorithm ۳-۲-2: LINEARSPACEGAPGREEDYSPANNER( $S, \theta, w$ )**

---

```
/* This algorithm takes as input a set  $S$  of  $n$  points in the
plane, and two real numbers  $\theta$  and  $w$  such that  $0 < \theta < \pi/4$ ,
 $0 \leq w < (\cos \theta - \sin \theta)/2$ . The algorithm returns a directed
 $t$ -spanner  $G = (S, E)$ , for  $t = 1/(\cos \theta - \sin \theta - 2w)$ . */
1  $s := \max\left(\frac{4}{\sin(\theta/2)}, \frac{2}{w}\right)$ ;
2  $\{(A_i, B_i)\}_{i=1}^m := \text{WSPD of } S \text{ with respect to } s$ ;
3 Sort increasingly the pairs  $\{(A_i, B_i)\}_{i=1}^m$  according to  $\min(A_i, B_i)$ ;
4  $E := \emptyset$ ;
5  $Q := \emptyset$ ; /*  $Q$  is a priority queue. */
6  $i := 1$ ;
7 ADDTOQUEUE( $Q, i$ );
8 while  $Q$  is not empty do
9   Extract the minimum from  $Q$  and denote it by  $(u, v)$ . Also, suppose that
    $(u, v) \in (A_k, B_k)$ ;
10   $E := E \cup \{(u, v), (v, u)\}$ ;
11  foreach pair  $(A_j, B_j)$  with an entry in  $Q$  for which either bounding box is at
   most  $w \times \max(A_j, B_j)$  away from either  $u$  or  $v$  do
12     $x := \text{SELECTEDPAIR}(j)$ ;
13    if  $x$  is a pair  $(u', v')$  then
14      Update the entry associated with  $\{A_j, B_j\}$  in  $Q$  to include  $(u', v')$  and
      change its key to  $|u'v'|$ ;
15    else
16      /*  $x$  is nil */
17      Remove from  $Q$  the entry associated with  $(A_j, B_j)$ ;
18    end
19  end
20  ADDTOQUEUE( $Q, i$ );
21 end
22 return the graph  $G = (S, E)$ ;
```

---

یا مساوی  $i$  هستند، توسط **ADDTOQUEUE** در نظر گرفته نمی‌شوند. بنابراین، در الگوریتم ۳-۲-۲، پس از اضافه شدن آن دو یال به  $E$  و بروزرسانی  $Q$ ، هنوز نیاز داریم که تابع **ADDTOQUEUE** را با پارامترهای  $i$  و  $Q$  (ما فرض کرده‌ایم که  $i$  یک متغیر سراسری است) فراخوانی کنیم.

**قضیه ۳-۲-۷.** خروجی دو الگوریتم **LINEARSPACEGAPGREEDYSPANNER** و **GAPGREEDY** با ورودی یکسان، یکسان است.

**اثبات.** می‌دانیم که در الگوریتم **GAPGREEDY** اگر یک یال  $(u, v)$  به مجموعه‌ی یالی  $E$  اضافه شود، آنگاه  $(u, v)$  نزدیک‌ترین زوج نقطه‌ای است که با هر یال  $(x, y)$  در مجموعه‌ی یالی  $E$  که تاکنون محاسبه شده است، روابط زیر را دارد:

۱.  $\text{angle}(uv, xy) > \theta$  یا،

۲.  $\text{angle}(uv, xy) \leq \theta$  و  $|ux| > w \times \min(|uv|, |xy|)$  و  $|vy| > w \times \min(|uv|, |xy|)$ .

ما نشان می‌دهیم که مجموعه‌ی یالی پوشش شکاف-حریصانه دقیقاً ترکیبی از همین یال‌ها است. به وضوح، اگر ما تابع  $\text{SELECTEDPAIR}(i)$  را روی هر زوج خوش-مجزا فراخوانی کنیم و سپس نزدیک‌ترین زوج که با  $(u, v)$  نمایش داده می‌شود را محاسبه کنیم، آنگاه آن زوج نقطه، نزدیک‌ترین زوج نقطه‌ای است که با هر یال  $(x, y)$  متعلق به مجموعه‌ی یالی  $E$  که تاکنون محاسبه شده است، روابط ۱ و ۲ که در بالا ذکر شد، را دارا است. واقعاً، الگوریتم  $\text{LINEARSPACEGAPGREEDYSPANNER}$  همین کار را انجام می‌دهد فقط اینکه آن، تابع  $\text{SELECTEDPAIR}(i)$  را برای همه‌ی زوج‌های  $(A_i, B_i)$  محاسبه نمی‌کند بلکه فقط برای زوج‌های خاص محاسبه می‌کند. ■

لم ۳-۲-۸. برای هر زوج خوش-مجزای  $(A_r, B_r)$  ( $1 \leq r \leq m$ )، تعداد دفعاتی که تابع  $\text{SELECTEDPAIR}(r)$  فراخوانی می‌شود حداکثر  $1 + c_{sw'}$  است که  $w' = \frac{3}{4}w$  و  $c_{sw'} = O(s^d(1+w's)^d)$ .

اثبات. روش اثبات، مشابه اثبات لم ۹ در [۵] است. برای هر عدد صحیح  $r$  که  $1 \leq r \leq m$ ، تابع  $\text{SELECTEDPAIR}(r)$  یک بار در تابع  $\text{ADDTOQUEUE}$  فراخوانی می‌شود. به علاوه، تابع  $\text{SELECTEDPAIR}(r)$  ممکن است که در خط ۱۲ در الگوریتم ۳-۲-۲ نیز فراخوانی شود.

فرض کنید که زوج خوش-مجزای  $(A_k, B_k)$  باعث فراخوانی تابع  $\text{SELECTEDPAIR}(r)$  شود. مشابه اثبات لم ۹ در [۵]، می‌توانیم ثابت کنیم که  $\ell(A_k, B_k) \in [\ell(A_r, B_r)/2, 2\ell(A_r, B_r)]$ . از طرف دیگر، با توجه به شرط خط ۱۱ در الگوریتم ۳-۲-۲ و همچنین از آنجایی که  $\max(A_r, B_r) \leq R(A_r)$  یا  $R(B_r)$ ،  $\frac{3}{4}\ell(A_r, B_r)$  (مقاله [۵] را ببینید)، فاصله‌ی حداقل یکی از  $R(A_k)$  و  $R(B_k)$  تا  $R(A_r)$  یا  $R(B_r)$  حداکثر  $w'\ell(A_r, B_r)$  است. از این رو، با توجه به لم ۳-۲-۳، نتیجه می‌گیریم که تعداد زوج‌های خوش-مجزای  $(A_k, B_k)$  که باعث فراخوانی  $\text{SELECTEDPAIR}(r)$  می‌شوند حداکثر برابر  $c_{sw'}$  است. ■

فرض کنید که در خط ۲ از الگوریتم ۳-۲-۲،  $\{(A_i, B_i)\}_{i=1}^m$  یک WSPD با اندازه‌ی  $m = O(s^d n)$  باشد که در زمان  $O(n \log n + s^d n)$  و با بکار بردن  $O(s^d n)$  حافظه محاسبه می‌شود.

حال، قضیه‌ی زیر را داریم:

**قضیه ۳-۲-۹.** فرض کنید که  $\theta$  و  $w$  دو عدد حقیقی باشند به طوری که  $0 < \theta < \pi/4$  و  $0 \leq w < (\cos \theta - \sin \theta)/2$  و همچنین فرض کنید که  $s = \max\left(\frac{4}{\sin(\theta/2)}, \frac{2}{w}\right)$ . الگوریتم `LINEARSPACEGAPGREEDYSPANNER`، پوشش شکاف-حریصانه را در زمان  $O(s^{2d}(1 + \frac{3}{4}ws)^d n^3)$  و با صرف  $O(s^d n)$  حافظه، محاسبه می‌کند.

**اثبات.** ما می‌توانیم به سادگی و بدون اثرگذاری روی زمان اجرای کل، همه‌ی زوج‌های خوش-مجزا را با صرف  $O(m^2)$  زمان، پیش‌پردازش کنیم و به این درخواست جواب دهیم: یک زوج  $(u, v) \in (A_k, B_k)$  داده شده است، زوج‌های خوش مجزای  $(A_j, B_j)$  را پیدا کنیم که فاصله‌ی  $R(A_j)$  یا  $R(B_j)$  تا  $u$  یا  $v$  حداکثر  $w \times \max(A_j, B_j)$  باشد (خط ۱۱ در الگوریتم ۳-۲-۲ را ببینید). توجه کنید که برای هر زوج خوش-مجزا، حداکثر به تعداد  $c_{sw}$  از این نوع زوج‌های خوش-مجزا وجود دارد (لم ۳-۲-۳ را ببینید). از این رو، پیش‌پردازش نیاز به یک حافظه‌ی خطی دارد. با بکار بردن نتیجه‌ی ۳-۲-۵، لم‌های ۳-۲-۶ و ۳-۲-۸، همه‌ی فراخوانی‌های `SELECTEDPAIR(i)` به

$$O\left(\sum_{i=1}^m (1 + c_{sw'}) s^d |A_i| |B_i| n\right),$$

زمان نیاز دارد و حافظه‌ی لازم برای آن  $O(s^d n)$  است (توجه کنید که می‌توانیم حافظه‌ها را در فراخوانی‌ها، مجدداً بکار ببریم). از آنجایی که  $w' = \frac{3}{4}w$  و  $c_{sw'} = O(s^d(1 + w's)^d)$  داریم:

$$\begin{aligned} O\left(\sum_{i=1}^m (1 + c_{sw'}) s^d |A_i| |B_i| n\right) &= O\left((1 + c_{sw'}) s^d n \sum_{i=1}^m |A_i| |B_i|\right) \\ &= O\left((1 + c_{sw'}) s^d n \binom{n}{2}\right) \\ &= O(s^{2d}(1 + \frac{3}{4}ws)^d n^3). \end{aligned}$$

توجه کنید که طبق تعریف WSPD، داریم  $\sum_{i=1}^m |A_i| |B_i| = \binom{n}{2}$ . لازم به ذکر است که زمان

گام‌های دیگر الگوریتم کمتر از زمان فراخوانی `SELECTEDPAIR(i)` است و همچنین می‌توانیم آن‌ها را در یک حافظه‌ی خطی انجام دهیم، لذا این، اثبات را کامل می‌کند. ■

### ۲-۲-۳ بهبود پیچیدگی زمانی

در این بخش، نشان می‌دهیم که چگونه پوشش شکاف-حریصانه را در زمان  $O(n^2)$  و با بکاربردن  $O(n^2)$  حافظه بسازیم. در بخش قبل، در تابع `LINEARSPACEGAPGREEDYSPANNER` و `ADDTQUEUE`، از تابع `SELECTEDPAIR` استفاده کردیم و دیدیم که زمان اجرای کل الگوریتم به زمان اجرای `SELECTEDPAIR` بسیار وابسته است. برای بهبود زمان کل اجرای الگوریتم، تغییراتی را روی تابع `SELECTEDPAIR` ایجاد می‌کنیم. ایده‌ای که در اینجا بکار می‌بریم به شرح زیر است.

در بخش قبل دیدیم که تابع `LINEARSPACEGAPGREEDYSPANNER` بعد از اضافه کردن دو یال به مجموعه‌ی یال‌ها، زوج انتخابی یک زوج خوش-مجزا را با مقایسه کردن همه‌ی زوج نقاط در آن زوج خوش-مجزا با همه‌ی یال‌هایی که تاکنون محاسبه شده‌اند، بروزرسانی می‌کند. حال، در اینجا، ما به هر زوج خوش-مجزا لیستی را اختصاص می‌دهیم که شامل تمام یال‌هایی است که احتمال دارند یالی برای پوشش شکاف-حریصانه باشند و از این رو، بروزرسانی زوج انتخابی هر زوج خوش-مجزا با مقایسه‌ی همه‌ی اعضای لیست متناظر، با دو یالی که اخیراً به مجموعه‌ی یالی اضافه شده است، انجام می‌شود. به طور دقیق‌تر، برای هر  $1 \leq i \leq m$ ، به درایه‌ی  $Q$ ، متناظر با زوج  $(A_i, B_i)$ ، لیست  $N_i$  را اختصاص می‌دهیم که این لیست شامل همه‌ی زوج‌های  $(x, y)$  با  $(x, y) \in (A_i \times B_i) \cup (B_i \times A_i)$  است به طوری که هر  $(x, y)$  می‌تواند یالی در پوشش شکاف-حریصانه است. بنابراین، پس از اضافه کردن یال‌های  $(u, v)$  و  $(v, u)$  (خط ۱۰ در الگوریتم ۲-۲-۳ را ببینید)، درایه‌هایی از  $Q$  را بروزرسانی می‌کنیم که شامل لیست  $N_j$  هستند و  $j$  در شرایط خط ۱۱ در الگوریتم ۲-۲-۳ صدق می‌کند (توجه کنید که علاوه بر لیست  $N_j$ ، کلید متناظر با زوج  $(A_j, B_j)$  در درایه‌ی  $Q$  بروزرسانی می‌شود). بدین منظور، تابع `SELECTEDPAIR` را با تابع زیر که به `IMPROVEDSELECTEDPAIR` نام‌گذاری شده است، جایگزین می‌کنیم (الگوریتم ۳-۲-۳ را ببینید). توجه کنید که در شروع الگوریتم اصلی، برای هر  $1 \leq i \leq m$ ،  $N_i$  را به صورت  $N_i := (A_i \times B_i) \cup (B_i \times A_i)$  در نظر می‌گیریم و آن‌ها را در صف  $Q$  ذخیره می‌کنیم.

توجه کنید که در گام بروزرسانی، نیازی نیست که اعضای  $N_i$  را با همه‌ی یال‌های  $E$  که از

---

**Algorithm ۳-۲-3: IMPROVEDSELECTEDPAIR( $i, u, v$ )**

---

```
1 minspair:=nil and let |minspair| :=  $\infty$ ;  
2  $F := \{(u, v), (v, u)\}$ ;  
3 foreach pair  $(p, q) \in N_i$  do  
4   flag:=CHECKPAIR( $p, q, F$ );  
5   if flag = true then  
6     if  $|pq| < |minspair|$  then  
7       | minspair:=( $p, q$ );  
8     end  
9   end  
10  else  
11  |  $N_i := N_i - \{(p, q)\}$ ;  
12  end  
13 end  
14 Update the entry associated with  $\{A_i, B_i\}$  in  $Q$  to include  $N_i$ ;  
15 return minspair;
```

---

قبل محاسبه شده‌اند، مقایسه کنیم زیرا این کار در گام‌های قبلی انجام می‌شود. به سادگی می‌توان دید که برای هر  $i$  با  $1 \leq i \leq m$ ، رابطه‌ی  $|N_i| \leq |(A_i \times B_i) \cup (B_i \times A_i)|$  برقرار است. از این رو، تابع IMPROVEDSELECTEDPAIR( $i, u, v$ ) نیاز به  $O(|A_i||B_i|)$  زمان دارد. همچنین، مشابه تابع SELECTEDPAIR( $i$ )، تابع IMPROVEDSELECTEDPAIR( $i, u, v$ ) حداکثر  $1 + c_{sw'}$  بار فراخوانی می‌شود، بنابراین، پیچیدگی زمانی الگوریتم ۳-۲-۲ پس از اعمال تغییراتی که ذکر شد، برابر است با

$$O\left(\sum_{i=1}^m (1 + c_{sw'})|A_i||B_i|\right) = O\left((1 + c_{sw'})\binom{n}{2}\right) = O(s^d(1 + \frac{3}{4}ws)^d n^2).$$

از آنجایی که برای ذخیره کردن هر لیست  $N_i$ ، به  $O(|A_i||B_i|)$  حافظه نیاز داریم، به وضوح، کل حافظه‌ای که برای ساخت پوشش شکاف-حریصانه پس از اعمال تغییرات بالا، نیاز است برابر  $O(n^2)$  است. بنابراین، با توجه به توضیحات بالا، داریم:

**قضیه ۳-۲-۱۰.** پوشش شکاف-حریصانه را می‌توان در زمان  $O(n^2)$  و با کمک  $O(n^2)$  حافظه ساخت.

### ۳-۳ بهبود ضریب کشش

در سال ۱۹۹۷، آریا و اسمید [۹]، ثابت کردند که پوشش شکاف-حریصانه دارای ضریب کشش  $1/(\cos(\theta) - \sin(\theta) - 2w)$  است. در اینجا، ما این ضریب کشش را به  $1/(1 - 2\sin(\theta/2) - 2w)$  بهبود می‌دهیم. تا جایی که ما اطلاع داریم، تاکنون این ضریب کشش بهبود داده نشده است. حال، مشابه لم‌های ۶.۴.۱ و ۷.۱.۱ در [۴۴]، لم زیر را ثابت می‌کنیم:

لم ۳-۳-۱. فرض کنید که  $\theta$ ،  $t$  و  $w$  اعداد حقیقی باشند به طوری که  $0 < \theta < \pi/3$ ،  $0 \leq w < (1 - 2\sin(\theta/2))/2$  و اگر  $p, q, r$  و  $s$  نقاطی در  $\mathbb{R}^d$  باشند به طوری که

$$1. \quad p \neq q, r \neq s$$

$$2. \quad \text{angle}(pq, rs) \leq \theta$$

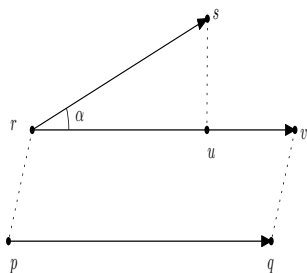
$$3. \quad |rs| \leq |pq| \text{ و}$$

$$4. \quad |pr| \leq w|rs|$$

آنگاه،  $|pr| < |pq|$ ،  $|sq| < |pq|$  و  $t|pr| + |rs| + t|sq| \leq t|pq|$

**اثبات.** ابتدا، ثابت می‌کنیم که  $|pr| < |pq|$ . از آنجایی که  $|pq| < \sqrt{2}|pq|$  و بر طبق شرط ۳، داریم  $|rs| < \sqrt{2}|pq|$ . همچنین، از آنجایی که  $|pr| \leq w|rs|$  (شرط ۴) و  $w < 1/2$ ، داریم  $|pr| < |rs|/2$ . حال، با ترکیب نامساوی‌های  $|rs| < \sqrt{2}|pq|$  و  $|pr| < |rs|/2$ ، داریم  $|pr| < \sqrt{2}|pq|/2 < |pq|$ . حال، ثابت می‌کنیم که  $|sq| < |pq|$ . فرض کنید که  $(r, v)$  یک بردار هم‌جهت با بردار  $(p, q)$  باشد به طوری که  $|rv| = |pq|$ . فرض کنید که  $u$  تصویر متعامد  $s$  روی بردار  $(r, v)$  باشد (شکل ۳-۱ را ببینید). توجه کنید که از آنجایی که  $|rs| \leq |pq|$ ، همیشه نقطه‌ی  $u$  وجود دارد. به وضوح، داریم  $|pr| = |vq|$ .

فرض کنید که  $\alpha = \text{angle}(pq, rs) \leq \theta$ . با استفاده از نامساوی مثلثی و با بکاربردن لم



شکل ۱-۳: اثبات لم ۱-۳-۳.

۱-۲-۲، داریم:

$$\begin{aligned}
 |sq| &\leq |sv| + |vq| \\
 &\leq |rv| - (1 - 2 \sin(\alpha/2))|rs| + |vq| \quad (\text{بر اساس لم ۱-۲-۲}) \\
 &= |pq| - (1 - 2 \sin(\alpha/2))|rs| + |pr| \\
 &\leq |pq| - (1 - 2 \sin(\alpha/2))|rs| + w|rs| \quad (\text{بر اساس شرط ۴}) \\
 &\leq |pq| - (1 - 2 \sin(\theta/2))|rs| + w|rs| \quad (\alpha \leq \theta \text{ که آنجایی که}) \\
 &= |pq| - (1 - 2 \sin(\theta/2) - w)|rs| \\
 &< |pq| \quad (w < (1 - 2 \sin(\theta/2))/2 \text{ که آنجایی که}).
 \end{aligned}$$

حال، ثابت می‌کنیم که  $t|pr| + |rs| + t|sq| \leq t|pq|$  بر اساس فرمول‌های بالا داریم:

$$\begin{aligned}
 t|pr| + |rs| + t|sq| &\leq t|pr| + |rs| + t(|pq| - (1 - 2 \sin(\theta/2) - w)|rs|) \\
 &\leq tw|rs| + |rs| + t(|pq| - (1 - 2 \sin(\theta/2) - w)|rs|) \\
 &= t|pq| + (1 - t(1 - 2 \sin(\theta/2) - 2w))|rs| \\
 &\leq t|pq| \quad (t \geq 1/(1 - 2 \sin(\theta/2) - 2w) \text{ که آنجایی که}).
 \end{aligned}$$

روابط بالا، اثبات را کامل می‌کنند. ■

لم ۲-۳-۳. فرض کنید که  $\theta$ ،  $t$  و  $w$  اعداد حقیقی باشند به طوری که  $0 < \theta < \pi/3$ ،  $0 \leq w < (1 - 2 \sin(\theta/2))/2$  و  $t \geq 1/(1 - 2 \sin(\theta/2) - 2w)$ . فرض کنید که  $S$  مجموعه‌ای از  $n$  نقطه در صفحه و  $G(S, E)$  یک گراف جهت‌دار باشد به طوری که شرایط زیر برقرار باشد: برای

هر دو نقطه‌ی متمایز  $p$  و  $q$  در  $S$ ، یک یال  $(r, s) \in E$  وجود دارد به طوری که

$$1. \text{angle}(pq, rs) \leq \theta$$

$$2. |rs| \leq |pq| \text{ و}$$

$$3. |pr| \leq w|rs| \text{ یا } |qs| \leq w|rs|.$$

در این صورت، گراف  $G$  یک  $t$ -پوشش برای  $S$  است.

**اثبات.** روش اثبات، مشابه اثبات لم ۷.۲.۱ در [۴۴] است. فقط لازم است که لم ۳-۳-۱ اعمال شود. ■

قضیه‌ی زیر نشان می‌دهد که ضریب کشش پوشش شکاف-حریصانه حداکثر برابر  $1/(1 - 2 \sin(\theta/2) - 2w)$  است.

**قضیه ۳-۳-۳.** فرض کنید که  $\theta$ ،  $t$  و  $w$  اعداد حقیقی باشند به طوری که  $0 < \theta < \pi/3$ ،  $0 \leq w < (1 - 2 \sin(\theta/2))/2$  و  $t \geq 1/(1 - 2 \sin(\theta/2) - 2w)$ . فرض کنید که  $S$  مجموعه‌ای از  $n$  نقطه در صفحه باشد. گراف  $G(S, E)$  تولید شده توسط الگوریتم  $GAPGREEDY(S, \theta, w)$ ، یک  $t$ -پوشش برای  $S$  است که  $t = 1/(1 - 2 \sin(\theta/2) - 2w)$ .

**اثبات.** روش اثبات، مشابه اثبات لم ۷.۲.۱ در [۴۴] است فقط لازم است که لم ۳-۳-۲ اعمال شود. ■

### ۳-۴ الگوریتم‌هایی از مرتبه‌ی ۳

در بخش ۳-۱، الگوریتم شکاف-حریصانه (الگوریتم ۳-۱-۱) را ارائه کردیم. در این بخش، سه الگوریتم دیگر جهت ساخت پوشش شکاف-حریصانه پیشنهاد می‌کنیم. اگر چه این الگوریتم‌ها دارای پیچیدگی زمانی  $O(n^3)$  و پیچیدگی حافظه‌ی  $O(n^2)$  هستند، اما در عمل، رفتارهای متفاوتی دارند. توجه کنید که همه‌ی الگوریتم‌هایی که در اینجا معرفی می‌کنیم، گراف یکسانی را به عنوان



خروجی تولید می‌کنند، بنابراین نیازی به مقایسه‌ی این الگوریتم‌ها از نظر خواص نظری گراف تولید شده نیست. در اینجا فقط زمان ساخت گراف تولید شده را در این الگوریتم‌ها بررسی می‌کنیم. در ادامه، این سه الگوریتم را توصیف می‌کنیم. در بخش ۳-۵، نتایج تجربی روی اجرای این الگوریتم‌ها را ارائه خواهیم کرد. ما این الگوریتم‌ها را الگوریتم شکاف-حریصانه‌ی تصادفی، الگوریتم شکاف-حریصانه‌ی ممنوع و الگوریتم شکاف-حریصانه‌ی دودویی می‌نامیم.

### ۳-۴-۱ الگوریتم شکاف-حریصانه‌ی تصادفی

خطوط ۸-۵ در الگوریتم ۳-۱-۱ را در نظر بگیرید. در این خطوط، الگوریتم، در بدترین حالت، زوج  $(p, q)$  را با تمام یال‌هایی که تاکنون محاسبه شده‌اند، مقایسه می‌کند. از این رو، ترتیب یال‌های  $(r, s)$  که با  $(p, q)$  مقایسه می‌شوند اصلاً مهم نیست. این عمل باعث می‌شود که بتوانیم یال‌های  $(r, s)$  را به صورت تصادفی با  $(p, q)$  مقایسه کنیم. از این رو، در الگوریتم شکاف-حریصانه‌ی تصادفی، ابتدا یک جایگشت تصادفی روی اعضای  $E$  ایجاد می‌کنیم و سپس یال‌های  $(r, s) \in E$  را با  $(p, q)$  مقایسه می‌کنیم. قبل از پیاده‌سازی این الگوریتم، حدس زده می‌شد که این تغییرات روی الگوریتم، زمان اجرا را برای نقاطی که به صورت یکنواخت توزیع شده‌اند، بهبود دهد، اما بعد از پیاده‌سازی، مشاهده شد که این حدس نادرست است و حتی نتیجه از نظر زمان اجرا بدتر می‌شود (نتایج بخش ۳-۵ را ببینید).

### ۳-۴-۲ الگوریتم شکاف-حریصانه‌ی ممنوع

فرض کنید که  $S$  مجموعه‌ای از  $n$  نقطه در صفحه باشد و  $w > 0$  یک عدد حقیقی باشد. برای هر زوج  $(r, s)$ ، فرض کنید که  $D_{rs}$  مجموعه‌ی همه‌ی نقاط  $p \in S$  باشد به طوری که  $|pr| \leq w|rs|$ . برطبق الگوریتم ۳-۱-۱، به وضوح، اگر برای یک  $(p, q)$  داشته باشیم  $|pr| \leq w|rs|$  یا  $|qs| \leq w|rs|$  و  $\angle(pq, rs) \leq \theta$ ، زوج  $(p, q)$  به  $E$  اضافه نمی‌شود. در الگوریتم شکاف-حریصانه‌ی ممنوع، ما همیشه کوتاه‌ترین زوج نقطه‌ی  $(r, s) \in L$  را به  $E$  به عنوان یال اضافه می‌کنیم ( $L$  لیست مرتب

شده‌ی همه‌ی زوج نقاط است). سپس، مجموعه‌ی  $D_{rs}$  و  $D_{sr}$  را محاسبه می‌کنیم. سپس، برای هر  $(p, q) \in D_{rs}$ ، اگر نقطه‌ی  $q \in S$  و  $p \in S$  وجود داشته باشد که  $\text{angle}(pq, rs) \leq \theta$  آنگاه زوج  $(p, q)$  را از  $L$  حذف می‌کنیم. برای جزئیات بیشتر، الگوریتم ۳-۴-۱ را ببینید. قابل توجه است که زوج  $(p, q)$  ممکن است حداقل دو بار در طول اجرای الگوریتم شناسایی شود. ما امید داشتیم که این تغییرات، الگوریتم را خیلی سریع‌تر کند (حداقل در عمل)، اما نتایج تجربی نشان می‌دهند که این حدس، نادرست است (نتایج بخش ۳-۵ را ببینید).

---

**Algorithm ۳-۴-1: FORBIDGAPGREEDY( $S, \theta, w$ )**

---

```

/* This algorithm takes as input a set  $S$  of  $n$  points in the
   plane, and two real numbers  $\theta$  and  $w$  such that  $0 < \theta < \pi/4$ ,
    $0 \leq w < (\cos \theta - \sin \theta)/2$ . The algorithm returns a directed
    $t$ -spanner  $G = (S, E)$ , for  $t = 1/(\cos \theta - \sin \theta - 2w)$ . */
1 Sort the  $2\binom{n}{2}$  ordered pairs of distinct points in non-decreasing order of their
  distances (ties are broken arbitrarily), and store them in a list  $L$ ;
2  $E := \emptyset$ ;
3 while  $L \neq \emptyset$  do
4   Choose the shortest edge  $(r, s) \in L$ ;
5    $E := E \cup \{(r, s)\}$ ;
6   Find all points  $p \in S$  such that  $|pr| \leq w|rs|$ , and store them in  $D_{rs}$ ;
7   Find all points  $q \in S$  such that  $|qs| \leq w|rs|$ , and store them in  $D_{sr}$ ;
8   foreach point  $p \in D_{rs}$  do
9     foreach point  $q \in S$  do
10      if  $\text{angle}(pq, rs) \leq \theta$  then
11         $L := L - \{(p, q)\}$ ;
12      end
13    end
14  end
15  foreach point  $q \in D_{sr}$  do
16    foreach point  $p \in S$  do
17      if  $\text{angle}(pq, rs) \leq \theta$  then
18         $L := L - \{(p, q)\}$ ;
19      end
20    end
21  end
22 end
23 return  $G = (S, E)$ ;

```

---

### ۳-۴-۳ الگوریتم شکاف-حریصانه‌ی دودویی

خط ۷ در الگوریتم ۳-۱-۱، فقط همه‌ی یال‌های  $(r, s) \in E$  را در نظر می‌گیرد که  $angle(pq, rs) \leq \theta$  است. برای اینکه الگوریتم، تشخیص دهد که کدام یال  $(r, s) \in E$  در شرط  $angle(pq, rs)$  صدق می‌کند، همه‌ی اعضای  $E$  را یکی یکی بررسی می‌کند. به وضوح، بررسی کردن همه‌ی اعضای  $E$  روش خوب و منطقی نیست. از این رو، ما الگوریتم شکاف-حریصانه‌ی دودویی را پیشنهاد می‌کنیم. این الگوریتم، در طول اجرا، همیشه یال‌های  $(r, s) \in E$  را براساس زاویه‌ی بین بردار  $\vec{r}$  و  $\vec{s}$  و محور  $x$ ها و به صورت صعودی مرتب نگه می‌دارد. بنابراین، وقتی الگوریتم می‌خواهد زوج  $(p, q)$  را بررسی کند، از یک الگوریتم جستجوی دودویی استفاده می‌کند تا تشخیص دهد که کدام زوج  $(r, s) \in E$  در شرط  $angle(pq, rs) \leq \theta$  صدق می‌کند. برای جزئیات بیشتر، الگوریتم ۳-۴-۲ را ببینید.

اگر چه زمان اجرای الگوریتم شکاف-حریصانه‌ی دودویی هنوز  $O(n^3)$  است، اما نتایج تجربی نشان می‌دهند که زمان اجرا در مقایسه با الگوریتم ۳-۱-۱ به طرز قابل توجهی بهبود پیدا کرده است.

### ۳-۵ نتایج پیاده‌سازی

در اینجا، با تعدادی جدول و شکل، رفتارهای سه الگوریتمی که در بخش قبل معرفی شدند را در مقایسه با الگوریتم ۳-۱-۱ نشان می‌دهیم. همچنین با استفاده از تعدادی جدول و شکل‌های دیگر، خصوصیات نظری پوشش شکاف-حریصانه را با پوشش حریصانه مقایسه می‌کنیم. برنامه‌هایی که ما نوشته‌ایم، روی یک کامپیوتر با پردازنده‌ی Intel® Core™i3-4160 CPU @ 3.60GHz × 4، ۸ گیگابایت حافظه‌ی RAM، سیستم عامل ابونتو<sup>۱</sup> ۱۶.۰۴ و کامپایلر ++g نسخه‌ی ۴.۹.۲ اجرا شده‌اند. لازم به ذکر است که برای تعداد نقاط زیاد، برنامه‌ها را اجرا نکردیم زیرا زمان اجرای الگوریتم‌ها بسیار زیاد می‌شد.

<sup>۱</sup>Ubuntu

---

**Algorithm ۳-۴-2: BINARYGAPGREEDY( $S, \theta, w$ )**

---

```
/* This algorithm takes as input a set  $S$  of  $n$  points in the
   plane, and two real numbers  $\theta$  and  $w$  such that  $0 < \theta < \pi/4$ ,
    $0 \leq w < (\cos \theta - \sin \theta)/2$ . The algorithm returns a directed
    $t$ -spanner  $G = (S, E)$ , for  $t = 1/(\cos \theta - \sin \theta - 2w)$ . */
1 Sort the  $2\binom{n}{2}$  ordered pairs of distinct points in non-decreasing order of their
   distances (ties are broken arbitrarily), and store them in a list  $L$ ;
2  $E := \emptyset$ ;
3 foreach ordered pair  $(p, q) \in L$  do
4    $add := true$ ;
5   Sort the pairs  $(r, s)$  in  $E$  according to the angle between the vector  $\overrightarrow{s-r}$  and the
    $x$ -axis;
6   Use the binary search algorithm to find the largest index  $i$  such that for each
    $j < i$  we have  $angle(pq, r_j s_j) \leq \theta$ , where  $E[j] := (r_j, s_j)$ ;
7   for  $(j = 0; j < i; j++)$  do
8      $add := add \wedge (|pr_j| > w|r_j s_j|) \wedge (|qs_j| > w|r_j s_j|)$ ;
9   end
10  if  $add = true$  then
11     $E := E \cup (p, q)$ ;
12  end
13 end
14 return  $G = (S, E)$ ;
```

---

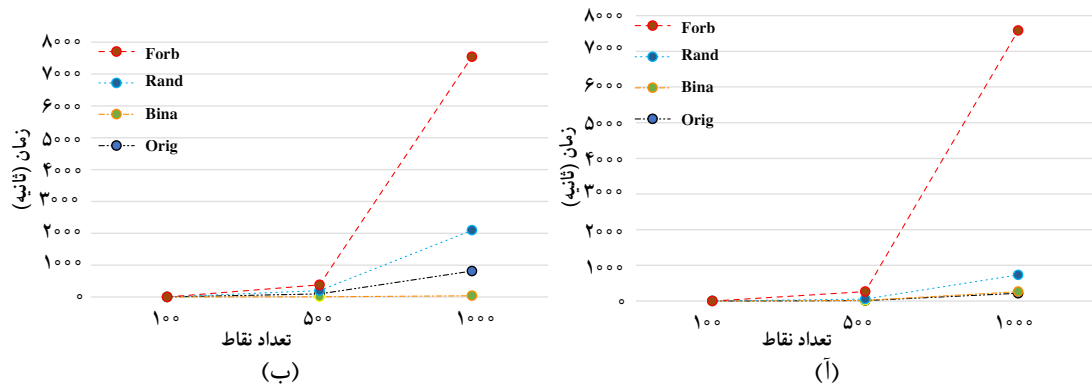
نتایج تجربی روی مجموعه نقاط با حداکثر ۵۰۰۰ نقطه که به صورت یکنواخت در صفحه توزیع شده‌اند، بدست آمده‌اند. برای هر مجموعه از  $n$  نقطه، مختصات هر نقطه در بازه  $[0, \sqrt{n}]$  است. برای هر عدد  $n$  که ما در نظر گرفتیم، الگوریتم‌ها روی ۱۰ مجموعه نقطه‌ی متفاوت و با پارامترهای ورودی مختلف اجرا شدند و مقدار کمینه، متوسط و بیشینه‌ی هر پارامتر (زمان، تعداد یال، درجه و ...) ذخیره شدند. توجه کنید که ما در شکل‌ها فقط از مقدار متوسط پارامترها استفاده می‌کنیم.

برخی از نتایج در جدول‌های ۱-۳ و ۲-۳ برای نقاط و پارامترهای مختلف نشان داده شده است. زمان در همه‌ی جدول‌ها و اشکال برحسب ثانیه است. برای سادگی، الگوریتم‌های ۱-۱-۳، شکاف حریصانه‌ی تصادفی، شکاف-حریصانه‌ی دودویی و شکاف-حریصانه‌ی ممنوع را به ترتیب با Orig، Rand، Bina و Forb نشان می‌دهیم.

شکل ۲-۳، زمان اجرای الگوریتم‌های Orig، Rand، Bina و Forb را برای  $n = 100, 500, 1000$  و مقادیر مختلف  $\theta$  و  $w$  نشان می‌دهد. از آنجایی که زمان اجرای Rand و Forb در مقایسه با Orig

جدول ۳-۱: زمان اجرای (برحسب ثانیه) Bina و Orig برای مقادیر مختلف  $\theta$  و  $w$  و مجموعه نقاط با  $1000$ ،  $500$ ،  $100$ ،  $2000$  و  $5000$  نقطه.

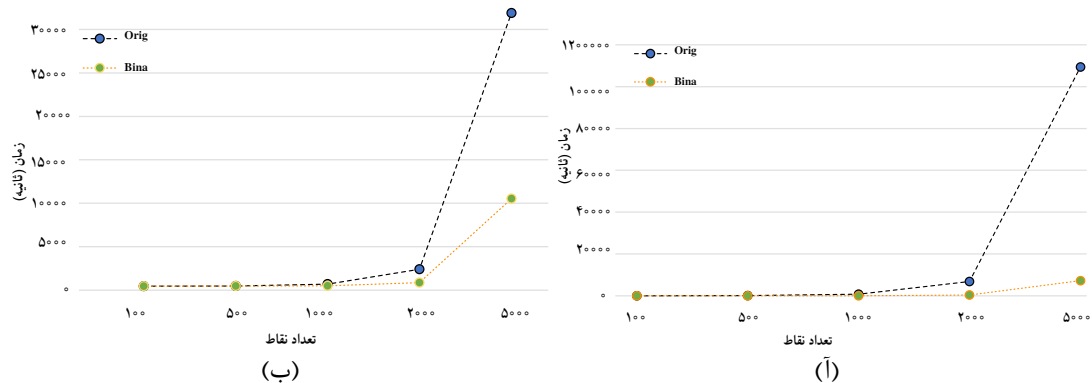
Bina			Orig			$w$	$\theta$	$t$	$n$
Max	Avg	Min	Max	Avg	Min				
۰٫۴۱۵۴	۰٫۳۳۶۷۲	۰٫۳۲۱۳۸	۰٫۲۰۰۴۵۲	۰٫۱۹۲۱۶۸	۰٫۱۸۹۰۵۲	۰٫۱۵۵۵	۰٫۱۷۴۵۳۳	۲	۱۰۰
۰٫۴۰۵۹۹	۰٫۳۹۹۷۷	۰٫۳۹۲۰۴	۰٫۱۱۱۵۰۱	۰٫۱۰۸۵۵۶	۰٫۱۰۴۶۸۵	۰٫۰۴۸۸	۰٫۳۴۹۰۶۶	۲	۱۰۰
۰٫۳۱۷۹۸	۰٫۲۷۰۹۷	۰٫۲۵۲۷۹	۰٫۳۲۵۸۵۹	۰٫۳۰۴۹۷۶	۰٫۲۹۲۲۹۵	۰٫۲۰۴۵	۰٫۰۸۷۲۶۷	۲	۱۰۰
۰٫۲۶۵۵۲	۰٫۲۵۱۰۷	۰٫۲۳۹۳۸	۰٫۵۷۹۰۳۴	۰٫۵۶۱۸۸۴	۰٫۵۴۱۷	۰٫۱۴۸۹	۰٫۳۴۹۰۷	۱٫۵	۱۰۰
۰٫۳۱۰۷	۰٫۲۹۴۴۵	۰٫۲۸۲۲۲	۰٫۳۴۲۱۰۹	۰٫۳۳۰۵۶۶	۰٫۳۱۴۹۷۵	۰٫۱۲۱۱	۰٫۰۸۷۲۶۷	۱٫۵	۱۰۰
۰٫۳۵۸۱۲	۰٫۳۴۳۱۷	۰٫۳۲۸۵۶	۰٫۲۰۲۰۴۸	۰٫۱۹۷۵۹۵	۰٫۱۹۱۴۸۶	۰٫۰۷۲	۰٫۱۷۴۵۳۳	۱٫۵	۱۰۰
۰٫۳۱۹۹۱	۰٫۲۹۸۷۹	۰٫۲۸۲۰۳	۰٫۶۷۱۳۴۶	۰٫۶۳۷۷	۰٫۶۱۴۰۹۶	۰٫۰۶۵۵	۰٫۳۴۹۰۷	۱٫۲	۱۰۰
۰٫۳۴۶۶۳	۰٫۳۱۴۳۷	۰٫۲۸۴۲۲	۰٫۴۳۲۲۲۴	۰٫۴۱۲۹۴۳	۰٫۳۹۵۱۸۵	۰٫۰۴۷۲	۰٫۰۶۹۸۱۳	۱٫۲	۱۰۰
۰٫۳۴۹۷	۰٫۳۳۳۵۹	۰٫۳۲۶۰۶	۰٫۲۴۶۸۹۹	۰٫۲۴۱۰۰۳	۰٫۲۳۷۸۸۷	۰٫۰۰۸۸	۰٫۱۳۹۶۲۶	۱٫۲	۱۰۰
۵٫۱۹۶۳۳	۳٫۰۹۴۷۳	۵٫۰۸۶۶۸	۲۶۹۲۰۸	۱۶۰۲۸	۲۶۵۷۸۱	۰٫۱۵۵۵	۰٫۱۷۴۵۳۳	۲	۵۰۰
۶٫۱۴۳۷۴	۶٫۰۸۵۵۴	۶٫۰۴۷۳۳	۱۴۸۸۳۸	۱۴۶۳۹۴	۱۴۴۶۱۳	۰٫۰۴۸۸	۰٫۳۴۹۰۶۶	۲	۵۰۰
۴٫۲۰۰۶۲	۴٫۰۶۰۱۶	۳٫۹۵۴۳۷	۴۵۵۲۵۴	۴۴۸۸۶۱	۴۴۲۹۴۱	۰٫۲۰۴۵	۰٫۰۸۷۲۶۷	۲	۵۰۰
۳٫۸۸۴۲۴	۳٫۷۵۰۹۵	۳٫۶۱۳۷۷	۱۰۰۰۶۹۷	۹۷۹۶۴۴	۹۶۰۰۱۷	۰٫۱۴۸۹	۰٫۳۴۹۰۷	۱٫۵	۵۰۰
۴٫۸۸۴۱۷	۴٫۷۳۸۰۷	۴٫۶۰۴۶۲	۵۱۰۰۶۶۶	۵۰۳۳۰۳	۴۹۰۷۰۰۵	۰٫۱۲۱۱	۰٫۰۸۷۲۶۷	۱٫۵	۵۰۰
۵٫۸۸۵۸۴	۵٫۷۸۲۶۸	۵٫۶۵۲۷۶	۲۸۲۰۹۵	۲۸۱۲۶۸	۲۸۰۱۹۶	۰٫۰۷۲	۰٫۱۷۴۵۳۳	۱٫۵	۵۰۰
۵٫۰۶۵۸۷	۴٫۸۷۹۸۴	۴٫۴۸۱۱۵	۱۱۵۸۵۴	۱۱۵۱۵۴	۱۱۴۱۴۲	۰٫۰۶۵۵	۰٫۳۴۹۰۷	۱٫۲	۵۰۰
۵٫۲۶۰۴۲	۵٫۱۵۲۳۷	۴٫۹۰۶۵۱	۶۶۲۲۳۷	۶۵۳۴۲۳	۶۴۵۸۷۷	۰٫۰۴۷۲	۰٫۰۶۹۸۱۳	۱٫۲	۵۰۰
۵٫۹۱۲۰۵	۵٫۸۷۸۸۳	۵٫۸۵۱۸۹	۳۵۷۰۵۲	۳۵۴۰۳۵	۳۵۲۰۲۶	۰٫۰۰۸۸	۰٫۱۳۹۶۲۶	۱٫۲	۵۰۰
۴۵۰۳۶	۴۳۶۱۹۷	۴۱۹۷۶۴	۲۱۹۳۰۹	۲۱۸۴۰۹	۲۱۷۸۲۹	۰٫۱۵۵۵	۰٫۱۷۴۵۳۳	۲	۱۰۰۰
۵۴۰۵۰۲	۲۱۵۹۸۸	۵۳۹۴۳۶	۱۱۷۱۳	۴۶۸۴۵۳	۱۱۷۰۹۶	۰٫۰۴۸۸	۰٫۳۴۹۰۶۶	۲	۱۰۰۰
۴۷۷۵۵۱	۴۶۰۸۳۷	۴۴۲۵۰۵	۳۸۱۴۳۶	۳۷۳۲۴۴	۳۶۶۳۰۸	۰٫۲۰۴۵	۰٫۰۸۷۲۶۷	۲	۱۰۰۰
۳۸۶۵۷۳	۳۷۷۱۴۴	۳۶۶۴۵۳	۸۲۴۶۸۶	۸۱۴۰۹۹	۸۰۳۷۶	۰٫۱۴۸۹	۰٫۳۴۹۰۶۶	۱٫۵	۱۰۰۰
۴۵۶۵۰۳	۴۱۹۳۰۲	۳۹۳۰۰۹	۴۱۸۶۰۸	۴۱۴۸۴	۴۱۱۸۵۳	۰٫۱۲۱۱	۰٫۰۸۷۲۶۶۵	۱٫۵	۱۰۰۰
۵۰۰۸۱۷۷	۴۹۷۰۵۳	۴۸۸۳۰۹	۲۲۸۲۳۶	۲۲۷۲۲۵	۲۲۵۳۸۳	۰٫۰۷۲	۰٫۱۷۴۵۳۳	۱٫۵	۱۰۰۰
۵۷۱۸۵۶	۵۶۳۴۲۶	۵۵۵۵۴	۹۹۳۳۶۷	۹۸۵۸۱۷	۹۷۹۳۰۷	۰٫۰۶۵۵	۰٫۳۴۹۰۶۶	۱٫۲	۱۰۰۰
۵۴۶۴۳۴	۵۱۹۰۸۱	۴۷۸۰۲۱	۵۴۶۶۴۵	۵۴۲۵۹	۵۴۰۰۳۵	۰٫۰۴۷۲	۰٫۰۶۹۸۱۳۲	۱٫۲	۱۰۰۰
۵۳۳۲۶۷	۵۱۵۸۴۷	۴۹۴۰۰۶	۲۹۰۲۵۸	۲۸۹۴۲۴	۲۸۷۹۷۹	۰٫۰۰۸۸	۰٫۱۳۹۶۲۶	۱٫۲	۱۰۰۰
۳۷۳۷۶۶	۳۵۴۷۰۱	۳۴۶۴۹۴	۱۷۵۹۶۵	۱۷۴۹۴	۱۷۳۹۴۵	۰٫۱۵۵۵	۰٫۱۷۴۵۳۳	۲	۲۰۰۰
۴۶۳۶۲	۴۵۵۹۹۹	۴۴۸۹۷۲	۹۹۷۱۶	۹۸۵۴۳۴	۹۷۳۶۴	۰٫۰۴۸۸	۰٫۳۴۹۰۶۶	۲	۲۰۰۰
۳۴۵۳۳	۳۴۰۷۶۳	۳۳۲۲۱۱	۳۰۵۱۵۳	۳۰۱۷۰۶	۲۹۸۵۹۸	۰٫۲۰۴۵	۰٫۰۸۷۲۶۷	۲	۲۰۰۰
۳۹۵۰۲۴	۳۸۶۳۴	۳۷۷۴۰۴	۶۸۳۶۳۲	۶۸۰۱۸۷	۶۷۳۸۶	۰٫۱۴۸۹	۰٫۳۴۹۰۷	۱٫۵	۲۰۰۰
۴۵۷۲۳۵	۴۵۱۳۵۴	۴۴۱۵۴۹	۳۳۳۹۹۳	۳۳۱۲۰۶	۳۲۹۳۸۳	۰٫۱۲۱۱	۰٫۰۸۷۲۶۷	۱٫۵	۲۰۰۰
۴۳۴۶۵۲	۴۲۵۱۷	۴۲۰۴۵۳۴	۱۸۶۵۶۲	۱۸۵۴۷	۱۸۴۴۸۹	۰٫۰۷۲	۰٫۱۷۴۵۳۳	۱٫۵	۲۰۰۰
۵۴۱۱۰۸	۵۳۷۶۹۸	۵۲۸۱۵۹	۸۲۹۹۴۸	۸۲۵۳۷۵	۸۲۳۳۹۵	۰٫۰۶۵۵	۰٫۳۴۹۰۷	۱٫۲	۲۰۰۰
۵۷۹۱۴۲	۵۶۷۴۹۲	۵۵۷۸۱۱	۴۵۴۴۸۷	۴۵۱۵۳	۴۴۹۵	۰٫۰۴۷۲	۰٫۰۶۹۸۱۳	۱٫۲	۲۰۰۰
۵۳۵۷۵۶	۵۳۰۴۰۳	۵۲۶۳۲۳	۲۳۹۳۸۲	۲۳۸۱۶۷	۲۳۷۳۴۳	۰٫۰۰۸۸	۰٫۱۳۹۶۲۶	۱٫۲	۲۰۰۰
۹۵۱۷۱۱	۹۰۰۸۰۸	۸۶۶۸۱۸	۲۸۳۲۸۱	۲۸۱۱۹۹	۲۸۰۳۲۶	۰٫۱۵۵۵	۰٫۱۷۴۵۳۳	۲	۵۰۰۰
۱۰۳۵۱۳	۱۰۰۹۶۱	۹۹۶۵۳	۱۵۱۹۳۳	۱۵۱۴۶۲	۱۵۰۹۳۱	۰٫۰۴۸۸	۰٫۳۴۹۰۶۶	۲	۵۰۰۰
۷۸۱۳۷۱	۷۶۹۷۶	۷۴۳۶۵۸	۴۸۸۵۷۶	۴۸۵۲۳۴	۴۸۲۰۹۵	۰٫۲۰۴۵	۰٫۰۸۷۲۶۷	۲	۵۰۰۰
۷۴۱۴۲۴	۷۲۶۷۷۳	۷۱۲۲۶۶	۱۰۹۷۳۷	۱۰۹۳۳۳	۱۰۸۸۲۴	۰٫۱۴۸۹	۰٫۳۴۹۰۷	۱٫۵	۵۰۰۰
۹۶۷۹۳۶	۹۵۴۴۱۳	۹۲۷۷۲۸	۵۴۹۱۹	۵۴۸۱۷۶	۵۴۷۲۷۵	۰٫۱۲۱۱	۰٫۰۸۷۲۶۷	۱٫۵	۵۰۰۰
۱۱۰۰۷۵	۱۰۸۵۸۹	۱۰۷۰۳۹	۲۹۹۳۱۹	۲۹۷۶۳۲	۲۹۵۶۶۲	۰٫۰۷۲	۰٫۱۷۴۵۳۳	۱٫۵	۵۰۰۰
۱۰۳۴۱۲	۱۰۱۰۱۸	۹۸۴۳۲۱	۱۳۴۳۶۲	۱۳۴۲۲۶	۱۳۳۹۸۷	۰٫۰۶۵۵	۰٫۳۴۹۰۷	۱٫۲	۵۰۰۰
۱۱۶۶۴۳	۱۱۶۲۰۴	۱۱۵۶۷۷	۷۲۹۶۲۴	۷۲۸۲۸۸	۷۲۷۰۹۹	۰٫۰۴۷۲	۰٫۰۶۹۸۱۳	۱٫۲	۵۰۰۰



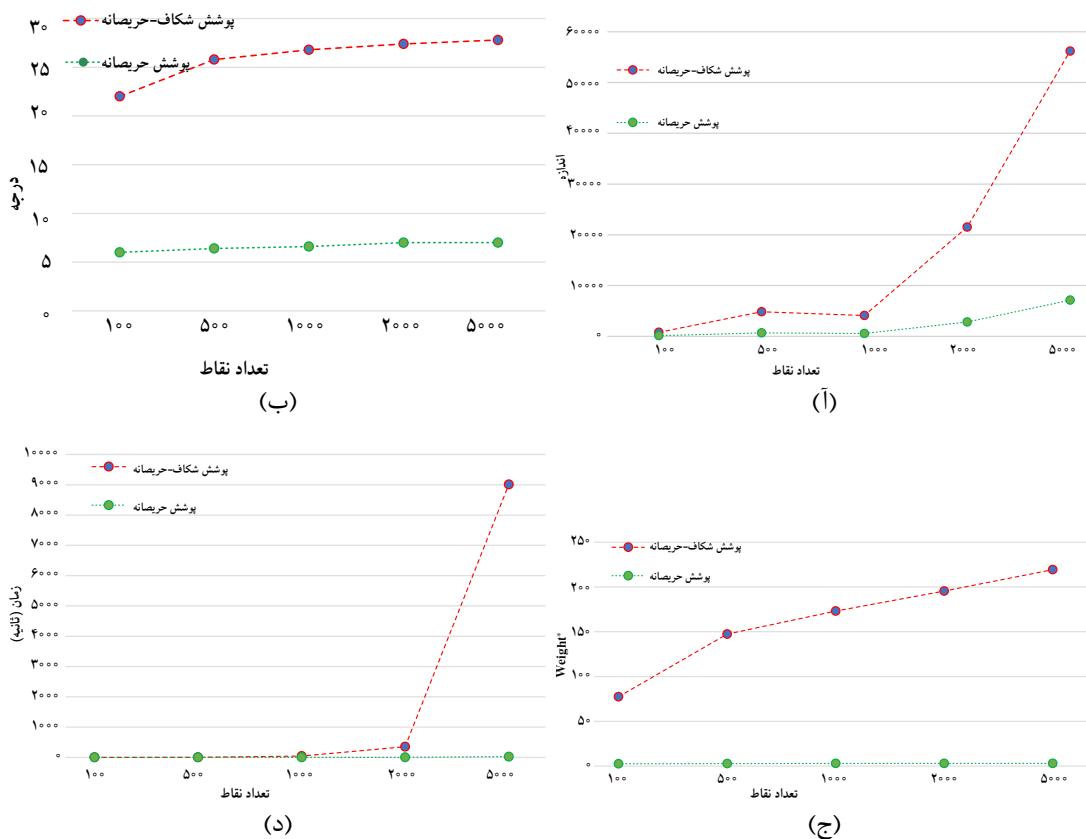
شکل ۳-۲: زمان اجرای Orig و Bina و Forb برای مقادیر مختلف  $\theta$  و  $w$ . (آ):  $\theta = 0.174533$  و  $w = 0.1555$  (ب):  $\theta = 0.139626$  و  $w = 0.088$

و Bina خیلی زیاد است، بنابراین برای  $n = 2000, 5000$  ما فقط کدهای مربوط به Orig و Bina را اجرا کردیم. شکل ۳-۳، زمان اجرای الگوریتم‌های Orig و Bina را برای  $n = 100, 500, 1000, 2000, 5000$  و برای پارامترهای مختلف نشان می‌دهد. به وضوح، نتایج نشان می‌دهند که الگوریتم ۳-۴-۲ بسیار سریع‌تر از الگوریتم‌های دیگر است.

فرض کنید که  $weight^*$ ، نسبت وزن گراف هندسی  $G$  به  $wt(MST(S))$  باشد. شکل ۳-۴، اندازه، درجه،  $weight^*$  و زمان اجرای پوشش شکاف-حریصانه را در مقایسه با پوشش حریصانه نشان می‌دهد. توجه کنید که برای محاسبه‌ی پوشش حریصانه، از الگوریتم FG-GREEDY که توسط بوز و همکارانش [۱۷] ارائه شد، استفاده کردیم. نتایج نشان می‌دهند که در عمل، پوشش حریصانه از نظر اندازه، درجه،  $weight^*$  و زمان ساخت، از پوشش شکاف-حریصانه بهتر است.



شکل ۳-۳: زمان اجرای Orig و Bina برای مقادیر مختلف  $\theta$  و  $w$  و مجموعه نقاط با  $100, 500, 1000, 2000$  و  $5000$  نقطه. (آ):  $\theta = 0.034907$  و  $w = 0.1489$  (ب):  $\theta = 0.174533$  و  $w = 0.1555$



شکل ۳-۴: اندازه، درجه،  $weigh^*$  و زمان اجرای پوشش شکاف-حریصانه در مقایسه با پوشش حریصانه به ازای پارامترهای مختلف. (آ): اندازه.  $t = 2$  و  $\theta = 0.349066$  و  $w = 0.488$ . (ب) درجه.  $t = 1.5$  و  $\theta = 0.174533$  و  $w = 0.072$ . (ج)  $Weight^*$ .  $t = 1.5$  و  $\theta = 0.34907$  و  $w = 0.1489$ . (د) زمان.  $t = 2$  و  $\theta = 0.174533$  و  $w = 0.1555$ .

### ۶-۳ نتیجه‌گیری

در این فصل، یک الگوریتم برای ساخت پوشش شکاف-حریصانه با کمک WSPD ارائه کردیم که پیچیدگی زمانی و حافظه‌ی آن به ترتیب  $O(n^3)$  و  $O(n)$  است. اگرچه، پوشش حریصانه یک پوشش با کیفیت عالی است و همچنین وزن آن از دیدگاه نظری، بهتر از وزن پوشش شکاف-حریصانه است، اما تا جایی که اطلاع داریم هیچ الگوریتمی برای ساخت پوشش حریصانه در زمان  $O(n^2)$  وجود ندارد. در این فصل، نشان دادیم که یک الگوریتمی برای ساخت پوشش شکاف-حریصانه در زمان  $O(n^2)$  و با بکاربردن  $O(n^2)$  حافظه وجود دارد. همچنین، ضریب کشش پوشش شکاف-حریصانه را بهبود دادیم. در این فصل، همچنین سه الگوریتم با زمان اجرای  $O(n^3)$  برای ساخت پوشش شکاف-حریصانه و تعدادی نتایج تجربی روی زمان اجرای این الگوریتم‌ها ارائه کردیم. با توجه

جدول ۳-۲: زمان اجرای محاسبه‌ی پوشش حریصانه و پوشش شکاف-حریصانه برای مجموعه نقاط مختلف و پارامترهای متفاوت. در اینجا منظور از T-Gap و T-Greedy به ترتیب زمان ساخت پوشش شکاف-حریصانه و پوشش حریصانه است.

	T-Greedy			T-Gap			$w$	$\theta$	$t$	$n$
	Max	Avg	Min	Max	Avg	Min				
۳۴۲۰۲۷	۲۹۷۴۶۸	۲۸۶۳۲۱	۳۷۳۷۶۶	۳۵۴۷۰۱	۳۴۶۴۹۴	۰/۱۵۵۵	۰/۱۷۴۵۳۳	۲	۲۰۰۰	
۲۹۶۶۶۶	۲۹۴۰۰۹	۲۹۱۳۷۶	۴۶۳۶۲	۴۵۵۹۹۹	۴۴۸۹۷۲	۰/۰۴۸۸	۰/۳۴۹۰۶۶	۲	۲۰۰۰	
۲۹۶۸۲۴	۲۹۲۸۰۷	۲۸۹۰۹۱	۳۴۵۳۳	۳۴۰۷۶۳	۳۳۲۲۱۱	۰/۲۰۴۵	۰/۰۸۷۲۶۷	۲	۲۰۰۰	
۳۸۸۶۴۲	۳۴۶۳۳۱	۳۳۱۱۱۱	۳۹۵۰۲۴	۳۸۶۳۴	۳۷۷۴۰۴	۰/۱۴۸۹	۰/۰۳۴۹۰۷	۱/۵	۲۰۰۰	
۳۶۰۰۸۲	۳۴۴۰۷۹	۳۳۰۶۳۴	۴۵۷۲۳۵	۴۵۱۳۵۴	۴۴۱۵۴۹	۰/۱۲۱۱	۰/۰۸۷۲۶۷	۱/۵	۲۰۰۰	
۴۵۱۷۶۷	۳۴۱۵۴۱	۳۳۲۰۶۲	۴۳۴۶۵۲	۴۲۵۱۷	۴۲۰۴۵۳	۰/۰۷۲	۰/۱۷۴۵۳۳	۱/۵	۲۰۰۰	
۶۲۱۴۸۶	۴۹۶۷۱۲	۴۵۶۸۲۷	۵۴۱۰۸	۵۳۷۶۹۸	۵۲۸۱۵۹	۰/۰۶۵۵	۰/۰۳۴۹۰۷	۱/۲	۲۰۰۰	
۴۹۲۴۳۴	۴۷۸۷۰۴	۴۵۴۰۰۵	۵۷۹۱۴۲	۵۶۷۴۹۲	۵۵۷۸۱۱	۰/۰۴۷۲	۰/۰۶۹۸۱۳	۱/۲	۲۰۰۰	
۵۱۸۱۴۷	۴۸۳۸۴	۴۵۶۴۳۷	۵۳۵۷۵۶	۵۳۰۴۰۳	۵۲۶۳۲۳	۰/۰۰۸۸	۰/۱۳۹۶۲۶	۱/۲	۲۰۰۰	
۲۸۶۷۶۷	۲۴۸۳۲۴	۲۳۹۴۲۲	۹۵۱۷۱۱	۹۰۰۸۰۸	۸۶۶۸۱۸	۰/۱۵۵۵	۰/۱۷۴۵۳۳	۲	۵۰۰۰	
۲۴۹۴۸۷	۲۴۳۸	۲۴۰۲۳۲	۱۰۳۵۱۳	۱۰۰۹۶۱	۹۹۶۵۳	۰/۰۴۸۸	۰/۳۴۹۰۶۶	۲	۵۰۰۰	
۲۷۷۸۹۷	۲۴۵۲۰۵	۲۳۷۱۶۶	۷۸۱۳۷۱	۷۶۹۷۶	۷۴۳۶۵۸	۰/۲۰۴۵	۰/۰۸۷۲۶۷	۲	۵۰۰۰	
۳۱۳۸۲	۲۸۴۹۲۳	۲۷۵۷۱۲	۷۴۱۴۲۴	۷۲۶۷۷۳	۷۱۲۲۶۶	۰/۱۴۸۹	۰/۰۳۴۹۰۷	۱/۵	۵۰۰۰	
۳۰۰۵۴۴	۲۸۰۹۶۲	۲۷۳۷۸	۹۶۷۹۳۶	۹۵۴۴۱۳	۹۲۷۷۲۸	۰/۱۲۱۱	۰/۰۸۷۲۶۷	۱/۵	۵۰۰۰	
۲۸۹۱۳۵	۲۷۹۰۳۲	۲۷۳۲۵۷	۱۱۰۰۷۵	۱۰۸۵۸۹	۱۰۷۰۳۹	۰/۰۷۲	۰/۱۷۴۵۳۳	۱/۵	۵۰۰۰	
۴۴۳۳۳۶	۴۱۳۰۸۷	۴۰۴۱۹۶	۱۰۳۴۱۲	۱۰۱۰۱۸	۹۸۴۳۲۱	۰/۰۶۵۵	۰/۰۳۴۹۰۷	۱/۲	۵۰۰۰	
۴۷۲۰۰۲	۴۲۱۷۱۱	۳۹۷۳۷۲	۱۱۶۶۴۳	۱۱۶۲۰۴	۱۱۵۶۷۷	۰/۰۴۷۲	۰/۰۶۹۸۱۳	۱/۲	۵۰۰۰	

به نتایج تجربی، ما نتیجه می‌گیریم که الگوریتم Bina (الگوریتم ۳-۴-۲) نسبت به الگوریتم‌های دیگری که در بخش ۳-۴ ارائه شدند، سریع‌تر است.

ما همچنین به صورت تجربی خصوصیات نظری پوشش حریصانه را در مقایسه با پوشش شکاف-حریصانه مورد بررسی قرار دادیم. لازم به ذکر است که بر طبق اطلاعاتی که ما داریم، تا قبل از این، پوشش شکاف-حریصانه با پوشش حریصانه با یکدیگر در عمل مقایسه نشده بودند. قابل توجه است که الگوریتم‌های ساخت پوشش حریصانه، تاکنون در عمل با یکدیگر مقایسه شده‌اند [۳۱، ۶، ۵]، اما در این مقالات، پوشش شکاف-حریصانه مد نظر قرار نگرفته است. ما این فصل را با بیان یک مسئله‌ی حل نشده خاتمه می‌دهیم:

آیا به ازای مقادیر خاصی از  $\theta$  و  $w$ ، پوشش حریصانه می‌تواند زیرگرافی از پوشش شکاف-حریصانه باشد؟



## فصل ۴

### گراف‌های هندسی خود-گرا

فرض کنید که  $G(S, E)$  یک گراف باشد. گویند  $G$  یک گراف مجاورت<sup>۱</sup> است هرگاه رئوس گراف به عنوان اشیاء هندسی باشند و دو رأس زمانی با یک یال به هم متصل هستند که اشیاء متناظر آنها با یکدیگر همسایه باشند (این همسایگی براساس نحوه‌ی ارتباطی که بین اشیاء است، تعریف می‌شود). یک گراف با مجموعه رئوس  $S$  در صفحه را یک گراف مجاورت هندسی گویند اگر هر دو نقطه‌ی  $p, q \in S$  که در برخی از شرایط همسایگی صدق می‌کنند، توسط پاره‌خط  $\overline{pq}$  به هم متصل باشند. از جمله گراف‌های مجاورت، گراف‌های یائو و گراف‌های گابریل<sup>۲</sup> را می‌توان نام برد.

در سال‌های اخیر، گراف‌های مجاورت در زمینه‌ی رسم گراف<sup>۳</sup> و هندسه‌ی محاسباتی بسیار مورد توجه قرار گرفته‌اند [۳۸، ۴۱، ۵۵]. گراف‌های حریمانه<sup>۴</sup> نوع خاصی از گراف‌های مجاورت هستند. یک مسیر هندسی (جهت‌دار)  $(v_1, v_2, \dots, v_n)$  را حریمانه گویند هرگاه فاصله‌ی اقلیدسی بین رأس  $v_i$  ( $1 \leq i \leq n$ ) و رأس مقصد  $v_n$  با افزایش  $i$  کاهش یابد. یک گراف هندسی  $G$  را حریمانه گویند هرگاه برای هر دو رأس  $p$  و  $q$  در  $G$ ، یک مسیر حریمانه از  $p$  به  $q$  در  $G$  وجود داشته باشد. کارهای تحقیقاتی زیادی در زمینه‌ی گراف‌های حریمانه انجام شده است [۸، ۱۹، ۳۴، ۳۵، ۳۷، ۵۴].

نوع دیگری از گراف‌های حریمانه، گراف‌های خود-گرا و گراف‌های وتر-افزایشی هستند که توسط علمداری و همکارانش معرفی شدند [۴]. فرض کنید که  $G$  گرافی هندسی با مجموعه رئوس  $S$  باشد. برای هر دو رأس  $s, t \in S$ ، یک مسیر از  $s$  به  $t$  در  $G$  را خود-گرا گویند، هرگاه برای هر نقطه‌ی  $q$  روی مسیر (نه لزوماً یک رأس)، یک نقطه‌ای که به طور پیوسته روی مسیر از  $s$  به سمت  $q$  در حال حرکت است، هرگز در مسیر حرکتش از  $q$  دور نشود. گراف  $G$  را خود-گرا گویند هرگاه برای هر دو رأس  $s$  و  $t$  در  $G$ ، یک مسیر خود-گرا از  $s$  به  $t$  در  $G$  وجود داشته باشد. یک مسیر را وتر-افزایشی گویند اگر آن مسیر در دو جهت (مبدأ به مقصد و بالعکس) خود-گرا باشد و یک گراف را وتر-افزایشی گویند هرگاه بین هر دو رأسش یک مسیر وتر-افزایشی وجود داشته باشد. به

<sup>۱</sup>proximity graph  
<sup>۲</sup>Gabriel

<sup>۳</sup>graph drawing  
<sup>۴</sup>greedy graphs

وضوح، یک گراف هندسی وتر-افزایشی، یک گراف خود-گرا نیز هست اما عکس این مطلب همیشه درست نیست.

یکی از دلایل استفاده‌ی عمده‌ی گراف‌های خود-گرا در کارهای تحقیقاتی، ضریب کشش خوب آن‌ها است. ثابت شده است که ضریب کشش گراف‌های خود-گرا حداکثر  $5.3332$  [۳۶] و گراف‌های وتر-افزایشی حداکثر  $2.94$  [۵۱] است.

در زمینه‌ی ساخت گراف‌های خود-گرا بر روی یک مجموعه از نقاط در صفحه چند کار تحقیقاتی انجام شده است. علمداری و همکارانش [۴] نشان دادند که در زمان چندجمله‌ای می‌توان تشخیص داد که آیا یک مسیر در  $\mathbb{R}^2$  یا  $\mathbb{R}^3$  خود-گرا است یا خیر. علاوه بر این، آن‌ها ثابت کردند که مسئله‌ی وجود یا عدم وجود مسیر خود-گرا بین دو رأس داده شده در یک گراف هندسی در  $\mathbb{R}^d$  ( $d \geq 3$ )،  $NP$ -سخت است. با استفاده از این مسئله، به راحتی می‌توان  $NP$ -سخت بودن مسئله‌ی اینکه آیا یک گراف هندسی داده شده در  $\mathbb{R}^d$  ( $d \geq 3$ )، خود-گرا است یا خیر، را نتیجه گرفت. آن‌ها حالت  $d = 2$  را به عنوان مسئله‌ی حل‌نشده مطرح کردند. همچنین، آن‌ها نشان دادند که برای هر مجموعه‌ی  $S$  از  $n$  نقطه در صفحه، می‌توان یک گراف هندسی وتر-افزایشی با اضافه کردن  $O(n)$  نقطه‌ی اشتاینری و  $O(n)$  یال ساخت. دهکردی<sup>۱</sup> و همکارانش [۴۹] نشان دادند که

الف) برای هر مجموعه‌ای از  $n$  نقطه در صفحه، یک گراف مسطح خود-گرا با اضافه کردن  $O(n)$  نقطه‌ی اشتاینری وجود دارد.

ب) برای هر مجموعه‌ای از  $n$  نقطه در حالت محدب<sup>۲</sup> (یعنی همه‌ی نقاط دقیقاً روی پوشش محدبشان قرار می‌گیرند) در صفحه، یک گراف هندسی وتر-افزایشی با  $O(n \log n)$  یال وجود دارد (بدون اضافه کردن نقطه‌ی اشتاینری).

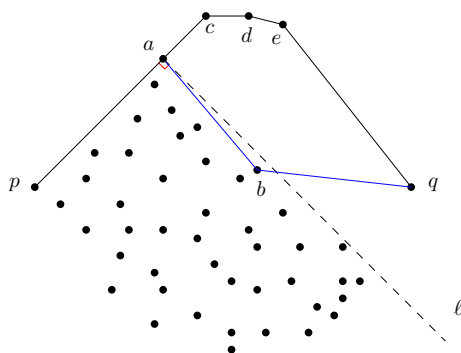
در این فصل، مفهوم گراف‌های هندسی خود-گرای ضعیف و وتر-افزایشی ضعیف را معرفی می‌کنیم. یک مسیر هندسی از  $r$  به  $s$  را یک مسیر خود-گرای ضعیف می‌نامیم، هرگاه در صورتی که نقطه‌ای مانند  $p$  (نه لزوماً رأس) روی مسیر باشد و از  $r$  به  $s$  حرکت کند، فاصله‌ی اقلیدسی بین  $p$  و  $s$  به طور پیوسته کاهش یابد یا ثابت بماند. ما یک گراف هندسی را خود-گرای ضعیف می‌نامیم

<sup>۱</sup>Dehkordi

<sup>۲</sup>convex position

هرگاه بین هر دو رأس  $r$  و  $s$  در آن گراف، یک مسیر خود-گرای ضعیف از  $r$  به  $s$  وجود داشته باشد. می‌گوییم یک مسیر هندسی بین  $r$  و  $s$  وتر-افزایشی ضعیف است هرگاه مسیری بین  $r$  و  $s$  وجود داشته باشد به طوری که آن مسیر هم خود-گرای ضعیف از  $r$  به  $s$  باشد و هم خود-گرای ضعیف از  $s$  به  $r$  باشد. می‌گوییم یک گراف هندسی، وتر-افزایشی ضعیف است هرگاه بین هر دو رأس آن، یک مسیر وتر-افزایشی ضعیف وجود داشته باشد. یکی از تفاوت‌های مسیر خود-گرای ضعیف و مسیر خود-گرا در این است که هر زیرمسیر از یک مسیر خود-گرا، نیز خود-گرا است اما در مسیر خود-گرای ضعیف این چنین نیست و فقط هر زیرمسیری که رأس انتهایی آن با رأس انتهایی مسیر اصلی یکی باشد، خود-گرای ضعیف است.

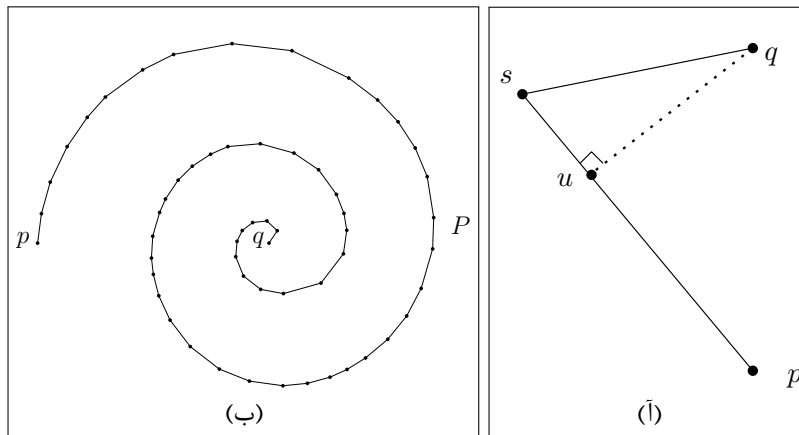
یکی از معایب گراف حریصانه این است که اگرچه هر رأس در طول مسیر حریصانه، به مقصد نزدیک‌تر می‌شود اما این مطلب برای نقاط میانی روی مسیر صدق نمی‌کند (شکل ۴-۲ آ را ببینید). از این رو، جهت رفع این عیب، گراف‌های خود-گرا می‌توانند جایگزین خوبی باشند. زیرا در گراف‌های خود-گرا، طبق تعریف، علاوه بر رئوس روی مسیر، نقاط میانی روی مسیر هم به نقاط بعدی روی مسیر در حال نزدیک شدن هستند. البته، در عمل شاید واقعاً نیاز به این نباشد که نقاط روی مسیر، به نقاط بعد از خود، در حال نزدیک شدن باشند بلکه همین که نقاط روی مسیر (نه فقط رئوس) به مقصد نزدیک شوند کافی باشد. از این رو، گراف‌های خود-گرای ضعیف می‌توانند مفید واقع شوند (شکل ۴-۱ را ببینید).



شکل ۴-۱: مسیر  $pacdeq$  یک مسیر خود-گرا از  $p$  به  $q$  است و مسیر  $pabq$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  است که خود-گرا نیست.

توجه کنید که تعریف مسیر خود-گرای ضعیف، تعریف ضعیف‌تری از مسیر خود-گرا و تعریف قوی‌تری از مسیر حریصانه است. لازم به ذکر است که، برخلاف گراف‌های خود-گرا، ضریب کشش

گراف‌های خود-گرای ضعیف ممکن است نامتناهی باشند (شکل ۴-۲ب را ببینید). اگر چه گراف‌های



شکل ۴-۲: (آ) یک مسیر حریصانه (ب) یک مسیر خود-گرای ضعیف که ضریب کشش آن می‌تواند زیاد باشد.

هندسی خود-گرای ضعیف، سبب بهبود ضریب کشش گراف‌های خود-گرا نمی‌شوند، اما این نوع گراف‌ها می‌توانند از جهات دیگر مفید واقع شوند. به عنوان مثال، برای یک مجموعه نقطه‌ای داده شده در صفحه، می‌توان یک گراف هندسی خود-گرای ضعیف مسطح ساخت (این یکی از نتایج این فصل خواهد بود که بعداً مفصل بیان خواهد شد) در حالی که تا جایی که ما اطلاع داریم، مسئله‌ی ساخت گراف هندسی خود-گرای مسطح برای مجموعه‌ای از نقاط در صفحه، هنوز حل نشده است. به علاوه، این مسئله که آیا برای هر مجموعه از نقاط در صفحه، یک گراف هندسی خود-گرای مسطح وجود دارد یا خیر، نیز هنوز حل نشده است. به عنوان مثالی دیگر، این مسئله که آیا یک گراف هندسی داده شده در فضای  $\mathbb{R}^d$  ( $d \geq 1$ )، خود-گرای ضعیف است یا خیر، را می‌توان در زمان چندجمله‌ای پاسخ داد (یکی دیگر از نتایج این فصل) در حالی که چنین مسئله‌ای در بحث گراف‌های خود-گرا برای حالت  $d > 2$ ،  $NP$ -سخت است و برای  $d = 2$  هنوز حل نشده است.

همان طور که قبلاً اشاره شد، ضریب کشش هر گراف هندسی خود-گرا حداکثر  $5/3332$  است [۳۶]. سؤالی که ممکن است پیش آید این است که برای یک مجموعه نقطه‌ای داده شده در صفحه و یک عدد حقیقی  $1 < t < 5/3332$ ، آیا یک گراف هندسی خود-گرا با ضریب کشش حداکثر  $t$  روی  $S$  وجود دارد؟ برای پاسخ به این سؤال، آبام و همکارانش [۱] مفهوم  $t$ -پوشش خود-گرا را معرفی کردند. یک گراف هندسی  $G$  را  $t$ -پوشش خود-گرا (ضعیف) گویند هرگاه بین هر دو نقطه‌ی  $p$  و  $q$  در  $S$ ، یک مسیر از  $p$  به  $q$  وجود داشته باشد به طوری که این مسیر هم یک  $t$ -مسیر بین  $p$  و  $q$

هم یک مسیر خود-گرا (ضعیف) از  $p$  به  $q$  باشد. به طور مشابه یک  $t$ -پوشش وتر-افزایشی (ضعیف) تعریف می‌شود.

**نتایج این فصل.** در این فصل، ما نتایج زیر را ارائه می‌دهیم. در ادامه، فرض کنید که  $t > 1$  یک عدد حقیقی ثابت باشد.

۱. برای هر مجموعه نقطه در صفحه که روی مرز یک مثلث منفرجه (مثلثی که حداقل یک زاویه‌ی بزرگتر از  $90^\circ$  درجه دارد) یا قائم‌الزاویه قرار گرفته‌اند، یک گراف هندسی وتر-افزایشی مسطح وجود دارد.

۲. برای هر مجموعه نقطه در صفحه که روی مرز یک مثلث حاده (مثلثی که تمام زاویه‌های آن کمتر از  $90^\circ$  درجه است) قرار گرفته‌اند، یک گراف هندسی وتر-افزایشی مسطح با بکار بردن حداکثر سه نقطه‌ی اشتاینری وجود دارد.

۳. در زمان چندجمله‌ای می‌توان تشخیص داد که آیا یک گراف هندسی داده شده در  $\mathbb{R}^d$  خود-گرای ضعیف است یا خیر.

۴. برای هر مجموعه نقطه در صفحه، یک گراف هندسی خود-گرای ضعیف مسطح وجود دارد.

۵. ما دو الگوریتم برای ساخت یک  $t$ -پوشش خود-گرای ضعیف روی هر مجموعه نقطه ارائه می‌کنیم. اولین الگوریتم، مشابه ساخت گراف یائو است در حالی که الگوریتم دوم مبتنی بر مفهوم تجزیه زوجی خوش-مجزا (WSPD) است.

۶. برای هر مجموعه نقطه در صفحه که روی مرز یک مثلث توزیع شده‌اند، یک  $t$ -پوشش وتر-افزایشی و یک  $t$ -پوشش خود-گرای ضعیف وجود دارد که هر دو دارای  $O(kn)$  یال هستند و  $k$  فقط وابسته به  $t$  است.

لازم به ذکر است که نتایج ۱ و ۲ از این جهت مورد توجه هستند که دهکردی و همکارانش [۴۹] جهت حل مسئله‌ی وجود گراف مسطح وتر-افزایشی برای هر مجموعه نقطه، ابتدا فرض کردند که نقاط روی یک مثلث قرار می‌گیرند، اما آن‌ها راه حلی برای مسئله در این حالت خاص ارائه نکردند.

## ۲-۴ گراف مسطح وتر-افزایشی برای نقاط روی مثلث

در این بخش، برای یک مجموعه نقطه‌ای داده شده در صفحه که روی یک مثلث قرار گرفته‌اند، یک گراف هندسی وتر-افزایشی مسطح می‌سازیم. توجه کنید که در طول این فصل، منظورمان از گراف، گراف هندسی است.

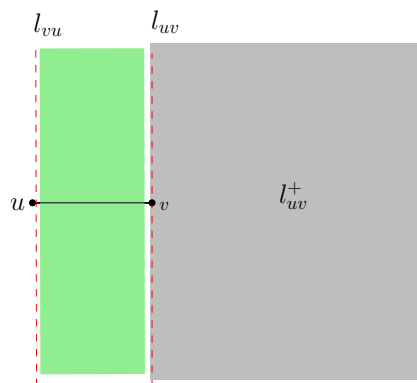
گراف کامل روی هر مجموعه نقطه‌ای در صفحه، به وضوح یک گراف وتر-افزایشی است. در ادامه، نشان می‌دهیم که برای هر مجموعه نقطه با حداقل چهار نقطه در صفحه، یک گراف غیرکامل وتر-افزایشی وجود دارد.

**گزاره ۱-۲-۴.** برای هر مجموعه‌ی  $S$  از  $n > 3$  نقطه در صفحه، همواره یک گراف غیرکامل وتر-افزایشی وجود دارد.

**اثبات.** ما ادعا می‌کنیم که مجموعه‌ی  $S$  شامل سه نقطه‌ی متمایز  $x$ ،  $y$  و  $z$  است به طوری که یکی از زاویه‌های داخلی مثلث  $\Delta_{xyz}$  حداقل  $\pi/2$  است. فرض کنید که  $c$  تعداد نقاط روی  $CH(S)$  باشد. ابتدا، ادعا را برای  $c > 3$  ثابت می‌کنیم. از آنجایی که هر چندضلعی با  $c > 3$  رأس، دارای حداقل یک زاویه‌ی داخلی بزرگ‌تر یا مساوی  $\pi/2$  است، بنابراین ادعا برقرار است. حال، فرض کنید که  $c = 3$ ،  $w$  و  $v$  نقاطی روی  $CH(S)$  باشند. همچنین فرض کنید که  $p \in S$  رأسی داخل  $CH(S)$  باشد. مثلث‌های  $\Delta_{puv}$ ،  $\Delta_{pvw}$  و  $\Delta_{pww}$  را در نظر بگیرید. از آنجایی که مجموع زاویه‌های حول  $p$  برابر  $2\pi$  است، بنابراین یک مثلث در بین مثلث‌های  $\Delta_{puv}$ ،  $\Delta_{pvw}$  و  $\Delta_{pww}$  وجود دارد که شامل یک زاویه‌ی داخلی بزرگ‌تر یا مساوی  $\pi/2$  است. از این رو، ادعا ثابت می‌شود.

حال، فرض کنید که  $x$ ،  $y$  و  $z$  سه نقطه در  $S$  باشند به طوری که زاویه‌ی بین پاره‌خط‌های  $\overline{xy}$  و  $\overline{yz}$  حداقل  $\pi/2$  باشد (طبق ادعای بالا، چنین نقاط  $x$ ،  $y$  و  $z$  وجود دارند). فرض کنید که  $K_S$  گراف کامل روی  $S$  باشد و فرض کنید که  $G$  گراف حاصل از حذف یال  $(x, z)$  از  $K_S$  باشد. به وضوح، مسیر  $(x, y) \cup (y, z)$  یک مسیر وتر-افزایشی بین  $x$  و  $z$  در  $G$  است و بنابراین  $G$  وتر-افزایشی است. این، اثبات را کامل می‌کند. ■

لازم به ذکر است که هر گراف غیرکامل روی رئوس یک مثلث متساوی‌الاضلاع، لزوماً یک



شکل ۳-۴: خطوط  $l_{vu}$  و نیم‌صفحه‌ی  $l_{uv}^+$

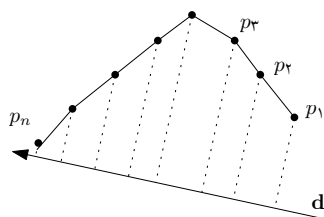
گراف وتر-افزایشی نیست.

در ادامه، لم مفیدی را که توسط علمداری و همکارانش [۳] اثبات شده است، بیان می‌کنیم که بعداً لازم می‌شود. در اینجا، ما از نمادهایی که در مقاله‌ی [۳] تعریف شده‌اند، استفاده می‌کنیم. فرض کنید که  $u$  و  $v$  دو نقطه‌ی متمایز باشند و  $\overline{uv}$  پاره‌خط بین  $u$  و  $v$  باشد. فرض کنید که  $l_{uv}$  خطی باشد که از  $v$  می‌گذرد و عمود بر  $\overline{uv}$  است. فرض کنید که  $l_{uv}^+$  نیم‌صفحه‌ی بسته‌ای باشد که مرز  $l_{uv}$  است و  $u$  را شامل نمی‌شود (شکل ۳-۴ را ببینید).

لم ۴-۲-۲ ([۳]). فرض کنید که  $P = (v_1, v_2, \dots, v_n)$  یک مسیر هندسی در  $\mathbb{R}^2$  باشد.  $P$  خود-گرا است اگر و فقط اگر برای هر  $1 < i < j \leq n$ ، نقطه‌ی  $v_j$  در  $l_{v_{i-1}v_i}^+$  قرار گیرد. به عبارتی دیگر،  $P$  خود-گرا است اگر و فقط اگر برای هر  $1 < i \leq n$ ، پوشش محدب  $\{v_i, v_{i+1}, \dots, v_n\}$  در  $l_{v_{i-1}v_i}^+$  قرار گیرد.

تعاریف و لم بالا را می‌توان برای ابعاد بالاتر تعمیم داد. کافی است که نیم‌صفحه‌های  $l_{v_{i-1}v_i}^+$  را با نیم‌فضاهایی که توسط نیم‌صفحاتی عمود بر  $\overline{v_{i-1}v_i}$  محدود شده‌اند، جایگزین کرد. دهکردی و همکارانش [۴۹] مسئله‌ی حل‌نشده زیر را مطرح کردند: آیا برای نقاطی که روی مرز یک مثلث حاده قرار دارند، یک گراف وتر-افزایشی مسطح وجود دارد؟ آن‌ها هیچ اشاره‌ای به حالتی که مثلث منفرجه یا قائم‌الزاویه است، نکردند. لازم به ذکر است که پوشش محدب نقاطی که روی اضلاع یک چندضلعی منتظم توزیع شده‌اند، ضرورتاً یک گراف وتر-افزایشی نیست. به عنوان مثال، می‌توان به راحتی این موضوع را برای یک مثلث حاده یا منفرجه بررسی کرد. ما نشان می‌دهیم که برای





شکل ۴-۴: یک مجموعه نقطه‌ی محدب یک‌طرفه نسبت به خط  $d$ .

هر مجموعه نقطه در صفحه که روی مرز مثلث منفرجه یا قائم‌الزاویه قرار دارند، یک گراف مسطح وتر-افزایشی وجود دارد. برای حالتی که مثلث حاده است، هنوز نمی‌دانیم که چگونه می‌توان این مسئله را حل کرد ولی در عوض، می‌توانیم نشان دهیم که برای نقاطی که روی مرز یک مثلث حاده قرار دارند یک گراف وتر-افزایشی مسطح با بکاربردن حداکثر سه نقطه‌ی اشتاینری وجود دارد. در ادامه، ابتدا نتیجه‌ای که دهکردی و همکارانش بدست آورده‌اند را بیان می‌کنیم [۴۹] که بعداً به آن نیاز داریم.

فرض کنید که  $S$  مجموعه‌ای از  $n$  نقطه در صفحه و در حالت محدب باشد و  $d$  یک خط مستقیم جهت‌دار باشد که بر هیچ خطی که از دو نقطه در  $S$  می‌گذرد، عمود نباشد. ترتیب نقاط در  $S$  را همان ترتیب تصویر نقاط روی خط  $d$  در نظر می‌گیریم. کوچک‌ترین و بزرگ‌ترین نقطه‌ی  $S$  با توجه به  $d$  به ترتیب برابر با اولین و آخرین نقطه در ترتیب تعریف شده روی  $S$  است. دهکردی و همکارانش [۴۹] مجموعه‌ی محدب  $S$  را مجموعه‌ی یک‌طرفه نسبت به  $d$  نامیدند هرگاه کوچک‌ترین و بزرگ‌ترین نقاط در  $S$  نسبت به  $d$  بر روی  $CH(S)$  پشت سر هم باشند (شکل ۴-۴ را ببینید). یک مجموعه‌ی محدب  $S$  را مجموعه‌ی محدب یک‌طرفه می‌گویند هرگاه خطی مانند  $d$  وجود داشته باشد به طوری که  $S$  نسبت به آن یک‌طرفه باشد.

لم ۴-۲-۳ ([۴۹]). فرض کنید که  $S$  یک مجموعه‌ی محدب یک‌طرفه با  $n$  نقطه باشد. یک گراف وتر-افزایشی مسطح روی  $S$  با  $2n - 3$  یال وجود دارد.

ساخت گرافی که در لم بالا بیان شد به صورت زیر است: ما این گراف را  $G$  می‌نامیم. بدون از دست دادن کلیت، فرض کنید که  $S$  نسبت به جهت مثبت محور  $x$ ، که با  $x$  نمایش می‌دهیم، یک‌طرفه است و  $S := \{s_1, s_2, \dots, s_n\}$  به طوری که نقاط بر اساس ترتیب تصویر آن‌ها روی  $x$

مرتب شده‌اند. ساخت  $G$  به صورت استقرایی است. اگر  $n = 2$ ، آنگاه  $G$  تنها شامل یال  $(s_1, s_2)$  است. حال، فرض کنید که  $n > 2$  و  $s_j$  نقطه‌ای با بیش‌ترین مختصات  $y$  در  $S$  باشد. ابتدا گراف  $G'$  که یک گراف مسطح وتر-افزایشی روی  $R = S \setminus \{s_j\}$  است را بدست می‌آوریم و سپس مجموعه‌ی یالی  $G'$  را به  $G$  اضافه می‌کنیم. توجه کنید که  $R$  یک مجموعه‌ی محدب یک‌طرفه نسبت به  $x$  است. در پایان، یال‌های  $(s_{j-1}, s_j)$  و  $(s_j, s_{j+1})$  را به  $G$  اضافه می‌کنیم.

در ادامه، مسئله‌ی ساخت گراف وتر-افزایشی مسطح برای مجموعه نقاطی که روی یک مثلث قرار دارند را در نظر می‌گیریم. فرض کنید که  $\Delta_{ABC}$  یک مثلث با رئوس  $A$ ،  $B$  و  $C$  باشد. بدون از دست دادن کلیت، فرض می‌کنیم که ضلع  $\overline{AB}$  افقی است و  $A$  در سمت چپ  $B$  در صفحه قرار دارد و  $S$  مجموعه‌ای از  $n$  نقطه در صفحه است که روی مثلث  $\Delta_{ABC}$  قرار دارد. ابتدا فرض می‌کنیم که  $\Delta_{ABC}$  یک مثلث منفرجه یا قائم‌الزاویه است. بدون از دست دادن کلیت، فرض می‌کنیم که زاویه‌ی بین اضلاع  $\overline{AC}$  و  $\overline{BC}$  حداقل  $\pi/2$  باشد.

در ادامه، ایده‌ی الگوریتم ساخت گراف وتر-افزایشی مسطح  $G(S, E)$  که با OBTUSETRIANGLE نام‌گذاری شده است، را بیان می‌کنیم. فرض کنید که  $x'$  تصویر نقطه‌ی  $x$  روی  $\overline{AB}$  باشد. در ابتدا، همه‌ی نقاط روی  $\overline{AC}$  را روی  $\overline{AB}$  تصویر می‌کنیم. سپس، هر نقطه‌ی  $p$  در  $\overline{AC}$  را به یک نقطه در  $S \cap \overline{AC}'$  که نزدیک‌ترین نقطه به یک نقطه در سمت راست  $p'$  است، توسط یالی متصل می‌کنیم. همچنین، برای هر دو نقطه‌ی متوالی  $u, v \in S \cap \overline{AC}$  (ما در اینجا فرض می‌کنیم که  $v$  نسبت به  $u$  به  $C$  نزدیک‌تر است)، اگر  $S \cap \overline{u'v'} \neq \emptyset$ ، آنگاه  $v$  را به یک نقطه در  $S \cap \overline{u'v'}$  که نزدیک‌ترین به  $v'$  است وصل می‌کنیم. ما همچنین روش بالا را برای نقاط روی ضلع  $\overline{BC}$  اعمال می‌کنیم فقط لازم است که کلمه‌ی "راست" را با کلمه‌ی "چپ" و  $\overline{AC}'$  را با  $\overline{BC}'$  جایگزین کنیم. با انجام عملیات بالا، ممکن است که یک مجموعه نقطه‌ی محدب یک‌طرفه  $S' \subseteq S$  نسبت به خطی که از  $\overline{AB}$  می‌گذرد، به وجود آید که در صورت به وجود آمدن چنین مجموعه‌ای، الگوریتمی که در اثبات لم ۴-۲-۳ بیان شد را بر روی مجموعه‌ی  $S'$  اعمال می‌کنیم (برای مشاهده‌ی جزئیات، الگوریتم ۴-۲-۱ را ببینید). در الگوریتم ۴-۲-۱، ما از نمادهایی استفاده کرده‌ایم که در ادامه آن‌ها را تعریف می‌کنیم. فرض کنید که  $v_1, \dots, v_k$  نقاط داخل  $S \cap \overline{AC}$  باشند که به صورت صعودی و براساس فاصله‌شان تا  $A$

مرتب شده‌اند. به طور مشابه، فرض کنید که  $S \cap \overline{BC} = \{u_1, \dots, u_l\}$  (نقاط بر اساس فاصله‌شان تا  $B$  به صورت صعودی مرتب شده‌اند) و  $S \cap \overline{AB} = \{w_1, \dots, w_m\}$  (نقاط بر اساس فاصله‌شان تا  $A$  به صورت صعودی مرتب شده‌اند). ما مسیر حاصل از اتصال  $v_a$  به  $v_b$  با بکار بردن فقط نقاط روی  $\overline{AC}$  و نقاط بین  $v_a$  و  $v_b$  را مسیر مستقیم بین  $v_a$  و  $v_b$  روی  $\overline{AC}$  می‌نامیم و آن را با  $Dpath_{AC}(v_a, v_b)$  نمایش می‌دهیم. فرض کنید  $\overline{XY}$  پاره‌خطی با نقاط انتهایی  $X$  و  $Y$  باشد،  $x$  یک نقطه در  $S$  و  $x'$  تصویر عمود آن روی خطی که از  $\overline{XY}$  می‌گذرد، باشند. ما  $AX(x, \overline{XY})$  را نزدیک‌ترین نقطه در  $S \cap \overline{XY}$  به  $x'$  به طوری که بین  $x'$  و  $X$  باشد، تعریف می‌کنیم. اگر چنین نقطه‌ای وجود نداشته باشد،  $AX(x, \overline{XY}) = \text{nil}$ . توجه کنید که طبق تعریف بالا،  $AX(x, \overline{YX})$  نزدیک‌ترین نقطه در  $S \cap \overline{XY}$  به  $x'$  است به طوری که بین  $x'$  و  $Y$  قرار دارد. همچنین، فرض کنید که  $j_1$  کوچک‌ترین عدد صحیحی باشد به طوری که هیچ نقطه‌ای در  $S$  بین  $v'_{j_1}$  و  $C'$  وجود نداشته باشد. در اینجا  $C'$  تصویر عمود  $C$  روی  $\overline{AB}$  است. اگر چنین  $j_1$  وجود نداشته باشد، فرض می‌کنیم  $j_1 = k + 1$ . همچنین فرض کنید که  $j_2$  کوچک‌ترین عدد صحیحی باشد که هیچ نقطه‌ای در  $S$  بین  $u'_{j_2}$  و  $C'$  وجود نداشته باشد. اگر چنین  $j_2$  وجود نداشت، فرض می‌کنیم  $j_2 = l + 1$ . قضیه‌ی زیر درستی الگوریتم `OBTUSETRIANGLE` را نشان می‌دهد.

**قضیه ۴-۲-۴.** گراف  $G(S, E)$  تولید شده توسط الگوریتم `OBTUSETRIANGLE`، یک گراف وتر-افزایشی مسطح روی  $S$  است.

**اثبات.** بر طبق ساخت گراف  $G$  و همچنین طبق لم ۴-۲-۳ به راحتی می‌توان فهمید که  $G$  مسطح است. حال، ثابت می‌کنیم که  $G$  یک گراف وتر-افزایشی است. کافی است نشان دهیم که برای هر دو نقطه‌ی متمایز  $p, q \in S$ ، یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  وجود دارد. اگر  $(p, q)$  یالی در  $G$  باشد، آنگاه به وضوح، یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  وجود دارد. بنابراین، فرض می‌کنیم که  $(p, q) \notin E$ .

در ادامه، برای اثبات اینکه یک مسیر، وتر-افزایشی است، لم ۴-۲-۲ را بکار می‌بریم. واضح است که اگر  $p$  و  $q$  روی یک ضلع یکسانی از مثلث قرار گرفته باشند، آنگاه یک مسیر وتر-افزایشی بین  $p$  و  $q$  وجود دارد که همان مسیر مستقیم بین  $p$  و  $q$  روی آن ضلع است.

---

**Algorithm ۴-۲-1: OBTUSETRIANGLE( $S$ )**

---

**input:** A set  $S$  of  $n$  points in the plane placed on the boundary of the obtuse triangle  $\Delta_{ABC}$ .  
**output:** An increasing-chord plane graph, denoted by  $G(S, E)$ , spanning  $S$ .

```
1  $E := \emptyset$ ;  
2 for  $i = 1$  to  $l-1$  do  
3   |  $E := E \cup (u_i, u_{i+1})$ ;  
4 end  
5 for  $i = 1$  to  $k-1$  do  
6   |  $E := E \cup (v_i, v_{i+1})$ ;  
7 end  
8 for  $i = 1$  to  $m-1$  do  
9   |  $E := E \cup (w_i, w_{i+1})$ ;  
10 end  
11 if  $v_k \neq u_l$  then  
12   |  $E := E \cup (v_k, u_l)$ ;  
13 end  
14 for  $i = 1$  to  $j_1 - 1$  do  
15   |  $E := E \cup (v_i, AX(v_i, \overline{BA}))$ ;  
16   | if  $AX(v_i, \overline{AB})$  is between  $v'_{i-1}$  and  $v'_i$  then  
17     |  $E := E \cup (v_i, AX(v_i, \overline{AB}))$ ;  
18   | end  
19 end  
20 if  $j_2 = l + 1$  then  
21   | for  $i = j_1$  to  $k$  do  
22     |  $E := E \cup (v_i, AX(v_i, \overline{BA}))$ ;  
23   | end  
24 end  
25 for  $i = 1$  to  $j_2 - 1$  do  
26   |  $E := E \cup (u_i, AX(u_i, \overline{AB}))$ ;  
27   | if  $AX(u_i, \overline{BA})$  is between  $u'_{i-1}$  and  $u'_i$  then  
28     |  $E := E \cup (u_i, AX(u_i, \overline{BA}))$ ;  
29   | end  
30 end  
31 if  $j_1 = k + 1$  then  
32   | for  $i = j_2$  to  $l$  do  
33     |  $E := E \cup (u_i, AX(u_i, \overline{AB}))$ ;  
34   | end  
35 end  
36 if  $j_1 \neq k + 1$  and  $j_2 \neq l + 1$  then  
37   |  $x := AX(v_{j_1}, \overline{BA})$ ;  
38   |  $y := AX(u_{j_2}, \overline{AB})$ ;  
39   |  $E = E \cup E'$ ;  
40 end  
41 return  $G(S, E)$ ;
```

---

اگر  $p$  روی  $\overline{AC}$  و  $q$  روی  $\overline{BC}$  باشد، آنگاه طبق لم ۴-۲-۲ مسیر

$$Dpath_{AC}(p, v_k) \cup (v_k, u_l) \cup Dpath_{BC}(u_l, q)$$

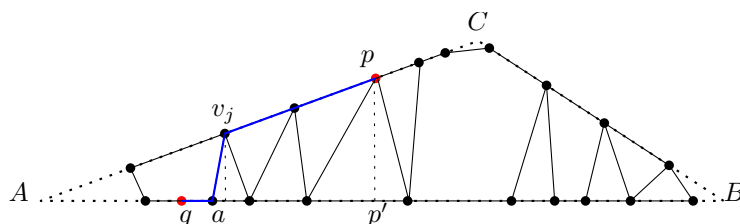
یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  است، زیرا زاویه‌های  $\angle Av_k u_l$  و  $\angle v_k u_l B$  بزرگ‌تر از  $\pi/2$  هستند.

در حالت بعدی، فرض کنید که  $p$  روی  $\overline{AC}$  و  $q$  روی  $\overline{AB}$  باشد و  $i$  یک عدد صحیح باشد به طوری که  $v_i := p$  اگر  $q$  بین  $A$  و  $p'$  قرار گیرد، آنگاه بر طبق ساخت گراف  $G$ ، یک عدد صحیح مانند  $j$  وجود دارد به طوری که  $q$  بین  $A$  و  $AX(v_j, \overline{AB})$  باشد و  $(v_j, AX(v_j, \overline{AB})) \in E$  (شکل ۴-۵ را

ببینید). فرض کنید که  $j$  کوچک‌ترین عدد صحیح با خاصیت بالا باشد. اگر  $a := AX(v_j, \overline{AB})$  آنگاه زاویه‌های  $\angle pv_ja$  و  $\angle v_jaq$  حداقل  $\pi/2$  هستند و بنابراین، طبق لم ۴-۲-۲، مسیر

$$Dpath_{AC}(p, v_j) \cup (v_j, AX(v_j, \overline{AB})) \cup Dpath_{AB}(AX(v_j, \overline{AB}), q)$$

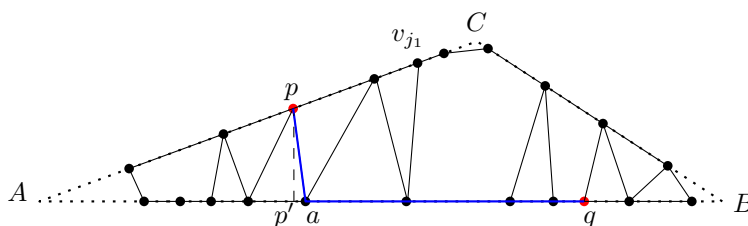
یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  است.



شکل ۴-۵: اثبات قضیه‌ی ۴-۲-۴: حالتی که  $q$  بین  $A$  و  $p'$  است.

حال، فرض می‌کنیم که  $q$  بین  $B$  و  $p'$  باشد (شکل ۴-۶ را ببینید). ما دو حالت را در نظر

می‌گیریم: (۱)  $i < j_1$  و (۲)  $i \geq j_1$ .

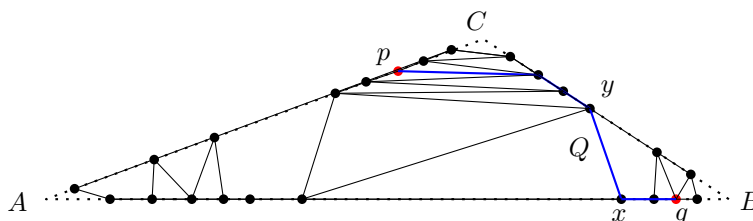


شکل ۴-۶: اثبات قضیه‌ی ۴-۲-۴: حالتی که  $q$  بین  $B$  و  $p'$  است و  $i < j_1$ .

**حالت ۱:**  $i < j_1$  از آنجایی که زاویه‌ی  $\angle paq$  که  $a := AX(p, \overline{BA})$  حداقل  $\pi/2$  است، بنابراین به وضوح، طبق لم ۴-۲-۲،  $(p, AX(p, \overline{BA})) \cup Dpath_{AB}(AX(p, \overline{BA}), q)$  یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  است.

**حالت ۲:**  $i \geq j_1$  در این حالت داریم  $j_1 \neq k+1$ . حال، اگر  $j_2 = l+1$ ، آنگاه بر طبق ساخت گراف  $G$ ، به وضوح مسیر  $(p, AX(p, \overline{BA})) \cup Dpath_{AB}(AX(p, \overline{BA}), q)$  یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  است. حال، فرض کنید  $j_2 \neq l+1$  (شکل ۴-۷ را ببینید) و گراف  $G'$  گرافی باشد که در خط ۳۶ از الگوریتم OBTUSETRIANGLE ساخته می‌شود (الگوریتم ۴-۲-۱ را ببینید). از آنجایی که  $G'$  یک گراف وتر-افزایشی روی  $S'$  و زیرگرافی از  $G$  است، بنابراین، یک مسیر وتر-افزایشی  $Q$  بین

$p$  و  $x$  در  $G$  وجود دارد به طوری که  $x = AX(v_j, \overline{BA})$  فرض کنید  $y$  آخرین رأس روی مسیر  $Q$  قبل از  $x$  باشد. بر طبق ساخت گراف  $G'$ ، به وضوح  $y$  روی  $\overline{BC}$  است و زاویه  $\angle yxq$  حداقل  $\pi/2$  است. از این رو طبق لم ۴-۲-۲، به وضوح مسیر  $Q \cup Dpath_{AB}(x, q)$  یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  است. با دلایل مشابه، می‌توانیم ثابت کنیم که هرگاه  $p$  روی  $\overline{BC}$  و  $q$  روی  $\overline{AB}$  باشد،



شکل ۴-۷: اثبات قضیه ۴-۲-۴: حالتی که  $q$  بین  $p'$  و  $B$  است و  $j_1 \geq j_2 + 1$  و  $j_1 \neq l + 1$ .

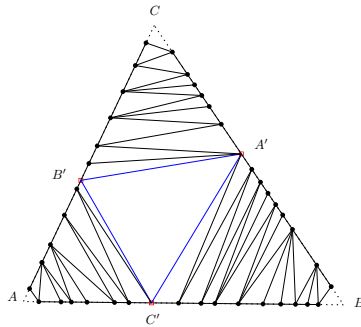
■ آنگاه یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  وجود دارد. این، اثبات را کامل می‌کند.

حال، فرض کنید  $S$  مجموعه‌ای از  $n$  نقطه در صفحه باشد که روی مثلث  $\Delta_{ABC}$  قرار گرفته‌اند. ما یک گراف مسطح وتر-افزایشی روی  $S$  با اضافه کردن حداکثر سه نقطه‌ی اشتاینری می‌سازیم و الگوریتم ساخت این گراف را ACUTETRIANGLESTEINER می‌نامیم. ایده‌ای که در این الگوریتم نهفته است، به شرح ذیل می‌باشد. ما با یک مجموعه‌ی یالی تهی  $E$  شروع می‌کنیم. ابتدا سه نقطه‌ی اشتاینری  $A'$ ،  $B'$  و  $C'$  را به  $S$  اضافه می‌کنیم که  $A'$ ،  $B'$  و  $C'$  به ترتیب تصاویر عمود  $A$ ،  $B$  و  $C$  روی پاره‌خط‌های  $\overline{BC}$ ،  $\overline{AC}$  و  $\overline{AB}$  هستند (شکل ۴-۸ را ببینید). حال، فرض کنید که  $S_1 := S \cup \{A', B', C'\}$  و  $P_1$ ،  $P_2$  و  $P_3$  به ترتیب مجموعه‌ی همه‌ی نقاط در  $S_1$  روی مثلث‌های  $\Delta_{A'B'C}$ ،  $\Delta_{B'C'A}$  و  $\Delta_{A'C'B}$  باشند. لازم به ذکر است که مجموعه‌های  $P_1$ ،  $P_2$  و  $P_3$  به ترتیب مجموعه‌های محدب یک‌طرفه نسبت به خطوطی که از  $A$  و  $B$ ، از  $B$  و  $C$  و از  $A$  و  $C$  می‌گذرند، هستند. الگوریتم، یال‌های  $(A', B')$ ،  $(B', C')$  و  $(C', A')$  را به  $E$  اضافه می‌کند و سپس الگوریتمی که در لم ۴-۲-۳ بیان شد، روی  $P_1$ ،  $P_2$  و  $P_3$  اجرا می‌شود و یال‌های حاصل به مجموعه‌ی  $E$  اضافه می‌شود.

در شکل ۴-۸، گراف وتر-افزایشی مسطح تولید شده توسط الگوریتم ACUTETRIANGLESTEINER

روی یک مجموعه نقطه، نشان داده شده است (برای جزئیات بیشتر، الگوریتم ۴-۲-۲ را ببینید).

قضیه‌ی زیر، درستی این الگوریتم را نشان می‌دهد.



شکل ۴-۸: یک گراف وتر-افزایشی تولید شده توسط الگوریتم ACUTETRIANGLESTEINER. توجه کنید که نقاط اشتاینری به شکل مربع‌های کوچک نمایش داده شده‌اند.

---

**Algorithm ۴-۲-2:** ACUTETRIANGLESTEINER( $S$ )

---

**input:** A set  $S$  of  $n$  points in the plane placed on the boundary of an acute triangle  $\Delta_{ABC}$ .

**output:** An increasing-chord plane graph  $G(S, E)$  with at most three Steiner points.

- 1  $E := \emptyset$ ;
  - 2  $A' :=$  orthogonal projection of  $A$  on  $\overline{BC}$ ;
  - 3  $B' :=$  orthogonal projection of  $B$  on  $\overline{AC}$ ;
  - 4  $C' :=$  orthogonal projection of  $C$  on  $\overline{AB}$ ;
  - 5  $S_1 := S \cup \{A', B', C'\}$ ;
  - 6  $P_1 :=$  the set of all points in  $S_1$  which are on the boundary of the triangle  $\Delta_{A'B'C'}$ ;
  - 7  $P_2 :=$  the set of all points in  $S_1$  which are on the boundary of the triangle  $\Delta_{B'C'A}$ ;
  - 8  $P_3 :=$  the set of all points in  $S_1$  which are on the boundary of the triangle  $\Delta_{A'C'B}$ ;
  - 9 Construct the increasing-chord plane graphs  $G_1 = (P_1, E_1)$ ,  $G_2 = (P_2, E_2)$  and  $G_3 = (P_3, E_3)$  using Lemma ۴-۲3-;
  - 10  $E = \{(A', B'), (B', C'), (C', A')\} \cup E_1 \cup E_2 \cup E_3$ ;
  - 11 **return**  $G$ ;
- 

**قضیه ۴-۲-۵.** گراف  $G(S, E)$  تولید شده توسط الگوریتم ACUTETRIANGLESTEINER، یک گراف وتر-افزایشی مسطح روی  $S$ ، با بکار بردن حداکثر سه نقطه‌ی اشتاینری، است. تعداد یال‌های این گراف بین  $۲n - ۳$  و  $۲n + ۳$  است.

**اثبات.** بر اساس ساخت گراف  $G$ ، به راحتی می‌توان دریافت که گراف  $G$  مسطح است. کران پایین و بالا روی تعداد یال‌های  $G$  به سادگی از لم ۴-۲-۳ نتیجه می‌شود. حال، ثابت می‌کنیم که  $G$  یک گراف وتر-افزایشی است. فرض کنید  $p$  و  $q$  دو نقطه‌ی دلخواه متمایز در  $S$  باشند. ما نشان می‌دهیم که یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G$  وجود دارد. اگر  $(p, q) \in E$ ، آنگاه به وضوح یک مسیر وتر-افزایشی بین  $p$  و  $q$  وجود دارد.

حال، فرض می‌کنیم که  $(p, q) \notin E$ . اگر هر دو نقطه‌ی  $p$  و  $q$  متعلق به یک مجموعه‌ی یکسان

$P_i$  که  $i = 1, 2, 3$  باشند، آنگاه از آنجایی که زیرگراف متناظر با هر  $P_i$  یک گراف وتر-افزایشی است، یک مسیر وتر-افزایشی بین  $p$  و  $q$  در  $G_i$  و در نتیجه در  $G$ ، وجود دارد. حال، فرض کنید که هر دو نقطه‌ی  $p$  و  $q$  متعلق به مجموعه‌ی یکسان  $P_i$  نباشند. ما قضیه را برای حالتی که  $p \in P_1$  و  $q \in P_2$  باشد، ثابت می‌کنیم. حالت‌های دیگر به طور مشابه اثبات می‌شوند.

فرض کنید که  $p \in P_1$  و  $q \in P_2$ . اگر هر دو نقطه‌ی  $p$  و  $q$  روی یک ضلع از  $\Delta_{ABC}$  باشند، آنگاه مسیر مستقیم بین  $p$  و  $q$  روی آن ضلع، یک مسیر وتر-افزایشی است (لازم به ذکر است که هر گراف تولید شده با بکار بردن اثبات لم ۴-۲-۳، یک گراف خارج مسطح<sup>۱</sup> است، بنابراین، همه‌ی نقاط متوالی  $P_i$  روی یک ضلع یکسان از مثلث، توسط یک یال به هم متصل هستند). حال، فرض کنید که  $p$  و  $q$  روی ضلع یکسانی از  $\Delta_{ABC}$  قرار ندارند. ما سه حالت زیر را در نظر می‌گیریم:

**حالت ۱:** روی  $p$   $\overline{AC}$  و روی  $q$   $\overline{AB}$  باشد. از آنجایی که  $p, B' \in P_1$ ، بنابراین مسیر مستقیم  $Dpath_{AC}(p, B')$  یک مسیر وتر-افزایشی بین  $p$  و  $B'$  در  $G$  است. همچنین از آنجایی که  $q, B' \in P_2$ ، بنابراین یک مسیر وتر-افزایشی  $Q_2$  بین  $B'$  و  $q$  در  $G$  وجود دارد. بر اساس نحوه‌ی انتخاب  $B'$ ، به ازای هر دو نقطه‌ی  $v_i, v_{i+1} \in P_1$ ، همه‌ی نقاط  $P_2$  در  $l_{v_{i+1}v_i}^+$  قرار می‌گیرند. از این رو، بر طبق لم ۴-۲-۲، مسیر  $Dpath_{AC}(p, B') \cup Q_2$  یک مسیر خود-گرا از  $p$  به  $q$  در  $G$  است. به طور مشابه، مسیر  $Dpath_{AC}(p, B') \cup Q_2$  یک مسیر خود-گرا از  $q$  به  $p$  در  $G$  است. بنابراین،  $Dpath_{AC}(p, B') \cup Q_2$  یک مسیر وتر-افزایشی بین  $p$  و  $q$  است.

**حالت ۲:** روی  $p$   $\overline{BC}$  و روی  $q$   $\overline{AB}$  باشد. ثابت می‌کنیم که مسیر

$$Dpath_{BC}(p, A') \cup \{(A', C')\} \cup Dpath_{AB}(C', q)$$

یک مسیر وتر-افزایشی بین  $p$  و  $q$  است. به راحتی می‌توان فهمید که زاویه‌های داخلی  $\angle CA'C'$  و  $\angle AC'A'$  در چهارضلعی با رئوس  $C', A', C$  و  $A$  بزرگ‌تر از  $\pi/2$  هستند. همچنین بر اساس انتخاب  $A'$  و  $C'$ ، به ازای هر دو نقطه‌ی  $u_i, u_{i+1} \in P_1$ ، همه‌ی نقاطی از  $P_2$  که روی  $\overline{AB}$  هستند، در  $l_{u_{i+1}u_i}^+$  قرار می‌گیرند و همچنین به ازای هر دو نقطه‌ی  $w_i, w_{i+1} \in P_2$ ، همه‌ی نقاطی از  $P_1$  که روی  $\overline{BC}$

<sup>۱</sup>outerplanar



هستند، در  $l_{w_i w_{i+1}}^+$  قرار می‌گیرند. از این رو، بر طبق لم ۴-۲-۲، مسیر

$$Dpath_{BC}(p, A') \cup \{(A', C')\} \cup Dpath_{AB}(C', q)$$

یک مسیر وتر-افزایشی بین  $p$  و  $q$  است.

■ **حالت ۳:** روی  $p$  روی  $\overline{BC}$  و روی  $q$  روی  $\overline{AC}$  باشد. اثبات این حالت مشابه حالت اول است.

لازم به ذکر است که اگرچه نتوانستیم یک گراف وتر-افزایشی مسطح و بدون اضافه کردن نقاط اشتتاینری روی  $S$  بسازیم، اما در عوض نشان دادیم که می‌توان این کار را با اضافه کردن حداکثر سه نقطه‌ی اشتتاینری انجام داد. علاوه بر این، در بخش ۴-۵، نشان خواهیم داد که برای هر عدد حقیقی  $t > 1$ ، یک  $t$ -پوشش وتر-افزایشی روی  $S$  (نه لزوماً مسطح) با  $O(kn)$  یال وجود دارد که در اینجا  $k$  فقط به  $t$  وابسته است.

### ۳-۴ گراف‌های خود-گرای ضعیف

در این بخش، الگوریتمی ارائه می‌کنیم که تشخیص می‌دهد آیا یک گراف داده شده در صفحه، خود-گرای ضعیف است یا خیر. علاوه بر این، یک الگوریتم جهت ساخت یک گراف خود-گرای ضعیف مسطح روی یک مجموعه نقطه‌ی داده شده در صفحه ارائه می‌کنیم. ما می‌توانیم به سادگی این نتایج را به ابعاد بالاتر تعمیم دهیم.

فرض کنید  $G$  یک گراف با مجموعه رئوس  $S$  در صفحه باشد و  $r$  و  $s$  دو رأس در  $S$  باشند.

یک مسیر خود-گرای ضعیف از  $r$  به  $s$  را به صورت زیر تعریف می‌کنیم:

**تعریف ۴-۳-۱.** (مسیر خود-گرای ضعیف). یک مسیر از  $r$  به  $s$  در  $G$  را خود-گرای ضعیف می‌نامیم اگر هنگامی که یک نقطه‌ی  $p$  روی آن مسیر از  $r$  به  $s$  در حال حرکت باشد، فاصله‌ی اقلیدسی بین  $p$  و  $s$  به طور پیوسته کاهش یابد یا ثابت بماند. جهت سادگی، فرض می‌کنیم از هر نقطه به خودش یک مسیر خود-گرای ضعیف وجود دارد.

گراف  $G(S, E)$  را خود-گرای ضعیف می‌نامیم هرگاه برای هر دو رأس  $r, s \in S$ ، یک مسیر

خود-گرای ضعیف از  $r$  به  $s$  در  $G$  وجود داشته باشد. به علاوه،  $G$  را وتر-افزایشی ضعیف می‌نامیم

هرگاه برای هر دو رأس  $r, s \in S$ ، یک مسیر بین  $r$  و  $s$  در  $G$  وجود داشته باشد به طوری که این مسیر هم خود-گرای ضعیف از  $r$  به  $s$  و هم خود-گرای ضعیف از  $s$  به  $r$  باشد.

#### ۴-۳-۱ تشخیص خود-گرای ضعیف بودن یک گراف

فرض کنید  $G$  یک گراف با مجموعه رئوس  $S$  باشد که  $S$  شامل  $n$  نقطه در صفحه است. ما یک الگوریتم با زمان چندجمله‌ای ارائه می‌دهیم که تشخیص می‌دهد آیا  $G$  خود-گرای ضعیف است یا خیر. لازم به ذکر است که مسئله‌ی تشخیص خود-گرایی یک گراف هندسی در صفحه، هنوز حل نشده است.

برای هر دو نقطه‌ی  $p$  و  $q$  در صفحه، دیسک بسته‌ای که اندازه‌ی قطر آن برابر با  $|pq|$  است و از نقاط  $p$  و  $q$  عبور می‌کند را با  $D_{pq}$  نمایش می‌دهیم. حال، داریم:

**مشاهده ۴-۳-۱.** فرض کنید  $G(S, E)$  یک گراف در صفحه و  $p$  و  $q$  دو رأس در  $S$  باشند. یک مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $G$  وجود دارد اگر و فقط اگر یک نقطه‌ی  $r \in S \cap D_{pq}$  وجود داشته باشد به طوری که  $(p, r) \in E$  و یک مسیر خود-گرای ضعیف از  $r$  به  $q$  در  $G$  وجود داشته باشد.

**اثبات.** ابتدا فرض کنید که یک مسیر خود-گرای ضعیف از  $p$  به  $q$  که آن را با  $P$  نمایش می‌دهیم، در  $G$  وجود داشته باشد. فرض کنید  $r \in S$  رأس بعد از  $p$  روی مسیر  $P$  باشد. اگر  $r = q$ ، آنگاه طبق تعریف ۴-۳-۱ حکم برقرار است. حال، فرض کنید  $r \neq q$ . از آنجایی که  $P$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  است بنابراین، طبق تعریف مسیر خود-گرای ضعیف، به وضوح زیرمسیری از  $P$  که از  $r$  شروع و به  $q$  ختم می‌شود، یک مسیر خود-گرای ضعیف از  $r$  به  $q$  است. حال، نشان می‌دهیم که  $r \in D_{pq}$ . به عنوان فرض خلف، فرض کنید که  $r \notin D_{pq}$ . بنابراین زاویه‌ی بین  $\overline{pr}$  و  $\overline{rq}$  کم‌تر از  $\pi/2$  است. از آنجایی که  $P$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  است، به راحتی می‌توان فهمید که زاویه‌ی بین  $\overline{pr}$  و  $\overline{rq}$  باید حداقل  $\pi/2$  باشد که این، یک تناقض است.

حال، فرض می‌کنیم که نقطه‌ی  $r \in S \cap D_{pq}$  وجود دارد به طوری که  $(p, r) \in E$  و مسیر

خود-گرای ضعیف از  $r$  به  $q$  در  $G$  وجود دارد که ما این مسیر را با  $Q$  نمایش می‌دهیم. طبق تعریف مسیر خود-گرای ضعیف، به راحتی می‌توان فهمید که مسیر  $P := \{(p, r)\} \cup Q$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  است. ■

حال، الگوریتم TESTWEAKLYSELFAPPROACHING (الگوریتم ۴-۳-۱ را ببینید) که یک گراف  $G(S, E)$  را به عنوان ورودی می‌گیرد و تشخیص می‌دهد که آیا  $G$  خود-گرای ضعیف است یا خیر، را توصیف می‌کنیم. در ابتدا، الگوریتم، همه‌ی زوج نقاط را به صورت صعودی و بر اساس فاصله‌ی اقلیدسی‌شان مرتب می‌کند و حاصل مرتب‌سازی را در یک لیست  $L$  ذخیره می‌کند. سپس برای هر زوج نقطه‌ی مرتب  $(p, q) \in L$ ، الگوریتم، عملیات زیر را انجام می‌دهد: ابتدا دیسک  $D_{pq}$  را پیدا می‌کند و سپس برای رأس  $p$  بررسی می‌کند که آیا رأسی مانند  $r \in S \cap D_{pq}$  وجود دارد به طوری که  $(p, r) \in E$ . اگر چنین رأسی وجود نداشته باشد، آنگاه الگوریتم جمله‌ی " $G$  خود-گرای ضعیف نیست" را در خروجی چاپ می‌کند. اگر برای همه‌ی زوج نقاط مرتب  $(p, q) \in L$ ، چنین رأس  $r$  موجود باشد، آنگاه الگوریتم، جمله‌ی " $G$  خود-گرای ضعیف است" را در خروجی چاپ می‌کند.

---

**Algorithm ۴-۳-1: TESTWEAKLYSELFAPPROACHING( $S, E$ )**

---

**input:** A graph  $G(S, E)$ .

**output:** Decide whether  $G$  is a weakly self-approaching graph.

- 1 Sort the  $2 \binom{n}{2}$  ordered pairs of distinct points in nondecreasing order of their distances (breaking ties arbitrarily), and store them in list  $L$ ;
  - 2 **foreach**  $(p, q) \in L$  **do** /\* process the elements of  $L$  in order. \*/
  - 3     Find the disk  $D_{pq}$ ;
  - 4     **if** there is no edge  $(p, r) \in E$  such that  $r \in S \cap D_{pq}$  **then**
  - 5         **return** " $G$  is not weakly self-approaching";
  - 6     **end**
  - 7 **end**
  - 8 **return** " $G$  is weakly self-approaching";
- 

**قضیه ۴-۳-۲.** الگوریتم TESTWEAKLYSELFAPPROACHING به درستی کار می‌کند و زمان اجرایش برابر با  $O(n^2 \log n + mn)$  است که  $n$  و  $m$  به ترتیب تعداد یال‌ها و رئوس گراف ورودی هستند.

**اثبات.** در ابتدا، ما زمان اجرا را اثبات می‌کنیم. به وضوح در خط ۴ الگوریتم، اگر فقط همه‌ی نقاط  $r$  که مجاور نقطه‌ی  $p$  در  $G$  هستند را بررسی کنیم، آنگاه برای نقطه‌ی ثابت  $p$ ، الگوریتم برای پردازش همه‌ی زوج نقاط  $(v, p) \in L$  که  $v \in S$ ،  $O(m)$  زمان صرف می‌کند. از این رو، خطوط ۲

تا  $\gamma$  الگوریتم، به  $O(mn)$  زمان نیاز دارند. از آنجایی که خط ۱ الگوریتم به  $O(n^2 \log n)$  زمان نیاز دارد، بنابراین، زمان کل اجرای الگوریتم برابر با  $O(n^2 \log n + mn)$  است.

حال، ثابت می‌کنیم که الگوریتم درست کار می‌کند. فرض کنید که  $G(S, E)$  گرافی است که خود-گرای ضعیف نیست. نشان می‌دهیم که الگوریتم جمله‌ی " $G$  خود-گرای ضعیف نیست" را در خروجی چاپ می‌کند. فرض کنید که  $(p, q)$  یک زوج نقطه با کم‌ترین فاصله‌ی اقلیدسی باشد به طوری که هیچ مسیر خود-گرای ضعیفی از  $p$  به  $q$  در  $G$  وجود ندارد. ما ابتدا ادعا می‌کنیم که هیچ نقطه‌ی  $r \in S \cap D_{pq}$  وجود ندارد به طوری که  $(p, r) \in E$ . به عنوان فرض خلف، فرض کنید که چنین نقطه‌ی  $r$  وجود دارد. بر اساس روش انتخاب  $(p, q)$  و از آنجایی که  $|rq| < |pq|$ ، یک مسیر خود-گرای ضعیف مانند  $Q$  از  $r$  به  $q$  در  $G$  وجود دارد. با توجه به مشاهده‌ی ۱-۳-۴، مسیر  $\{(p, r)\} \cup Q$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  است که این یک تناقض است. از این رو، ادعا ثابت می‌شود.

حال، لحظه‌ای را که الگوریتم زوج نقطه‌ی  $(p, q)$  را پردازش می‌کند، در نظر بگیرید. بر اساس ادعای بالا، هیچ یال  $(p, r) \in E$  که  $r \in D_{pq}$  وجود ندارد. از این رو، الگوریتم در خروجی، جمله‌ی " $G$  خود-گرای ضعیف نیست" را چاپ می‌کند و الگوریتم تمام می‌شود.

حال، فرض کنید که  $G$  یک گراف خود-گرای ضعیف است. بنابراین، برای هر زوج نقطه‌ی  $(p, q)$ ، یک مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $G$  موجود است. از این رو، بر اساس مشاهده‌ی ۱-۳-۴، در هر دیسک  $D_{pq}$ ، یک نقطه‌ی  $r \in S$  وجود دارد به طوری که  $(p, r) \in E$ . بنابراین، بر طبق الگوریتم، به سادگی می‌توان پی برد که الگوریتم، جمله‌ی " $G$  خود-گرای ضعیف است" را چاپ می‌کند. ■

#### ۴-۴ ساخت گراف‌های خود-گرای ضعیف مسطح

در این بخش، الگوریتمی ارائه می‌دهیم که یک گراف خود-گرای ضعیف مسطح را برای هر مجموعه‌ی  $S$  شامل  $n$  نقطه در صفحه، می‌سازد. در ادامه، فرض می‌کنیم که همه‌ی نقاط در حالت کلی هستند

که این بدین معنا است که هیچ سه نقطه‌ای روی یک خط مستقیم قرار ندارند.

در ابتدا، یک الگوریتم به نام NEARESTINDISK (الگوریتم ۴-۴-۱ را ببینید) را ارائه می‌کنیم که یک گراف خود-گرای ضعیف روی  $S$  می‌سازد. نشان خواهیم داد که این الگوریتم، مسطح بودن گراف خروجی را تضمین نمی‌کند. پس از بیان این الگوریتم، یک الگوریتم، مشابه NEARESTINDISK را ارائه می‌کنیم که یک گراف خود-گرای ضعیف مسطح روی  $S$  می‌سازد. در ادامه، الگوریتم NEARESTINDISK را توصیف می‌کنیم. فرض کنید که  $nearest(p, q)$  تابعی باشد که دو رأس  $p, q \in S$  را به عنوان ورودی می‌گیرد و نزدیک‌ترین نقطه به  $p$  در  $S \setminus \{p\} \cap D_{pq}$  را به عنوان خروجی برمی‌گرداند.

---

**Algorithm ۴-۴-1: NEARESTINDISK( $S$ )**

---

**input:** A set  $S$  of  $n$  points in the plane.  
**output:** A weakly self-approaching graph  $G(S, E)$ .

```

1  $E := \emptyset$ ;
2 foreach  $p \in S$  do
3   foreach  $q \in S$  do
4      $r := nearest(p, q)$ ;
5      $E := E \cup \{(p, r)\}$ ;
6   end
7 end
8 return  $G(S, E)$ ;
```

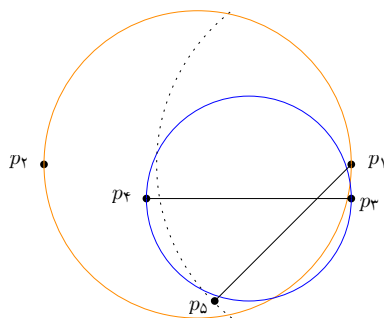
---

توجه کنید که اگر  $D_{pq}$  هیچ نقطه‌ای به جز  $p$  و  $q$  نداشته باشد، آنگاه  $nearest(p, q)$  نقطه‌ی  $p$  را برمی‌گرداند. الگوریتم به ازای هر نقطه‌ی  $q \in S$ ، هر نقطه‌ی  $p \in S$  را به  $nearest(p, q)$  وصل می‌کند.

**قضیه ۴-۴-۱.** گراف  $G(S, E)$ ، تولید شده توسط NEARESTINDISK، یک گراف خود-گرای ضعیف است.

**اثبات.** فرض کنید که  $p$  و  $q$  دو نقطه‌ی دلخواه و متمایز در  $S$  باشند. ثابت می‌کنیم که یک مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $G$  وجود دارد. اثبات از طریق استقرا روی رتبه‌ی فاصله‌ی اقلیدسی  $|pq|$  است.

**پایه‌ی استقرا:** در صورتی که  $(p, q)$  نزدیک‌ترین زوج نقطه در  $S$  باشد، به وضوح، خروجی تابع



شکل ۹-۴: مجموعه‌ی  $S$  با ۵ نقطه که نشان می‌دهد که گراف  $G(S, E)$  تولید شده توسط NEARESTINDISK یک گراف مسطح نیست.

$nearset(p, q)$  برابر با  $q$  است. از این رو، با توجه به الگوریتم،  $(p, q) \in E$  است. بنابراین، یک مسیر خود-گرایی ضعیف از  $p$  به  $q$  در  $G$  وجود دارد که همان یال  $(p, q)$  است.

**فرض استقرا:** فرض کنید که برای هر دو نقطه‌ی  $x$  و  $y$  در  $S$  که  $|xy| < |pq|$ ، یک مسیر خود-گرایی ضعیف از  $x$  به  $y$  در  $G$  وجود دارد.

**حکم استقرا:** ثابت می‌کنیم که یک مسیر خود-گرایی ضعیف از  $p$  به  $q$  در  $G$  وجود دارد. دیسک  $D_{pq}$  را در نظر بگیرید. اگر  $nearest(p, q) = q$ ، آنگاه  $E$  یال  $(p, q)$  را شامل می‌شود و اثبات تمام است. حال، فرض کنید که  $r$  یک رأس در  $S$  باشد به طوری که  $r \neq q$  و  $r = nearest(p, q)$ . توجه به الگوریتم،  $E$  شامل یال  $(p, r)$  است. همچنین از آنجایی که  $|rq| < |pq|$ ، طبق فرض استقرا، یک مسیر خود-گرایی ضعیف از  $r$  به  $q$  در  $G$  وجود دارد که ما این مسیر را با  $Q$  نمایش می‌دهیم. با توجه به مشاهده‌ی ۱-۳-۴،  $\{(p, r)\} \cup Q$  یک مسیر خود-گرایی ضعیف از  $p$  به  $q$  در  $G$  است. ■

در ادامه، با ذکر یک مثال نشان می‌دهیم که گراف  $G$  تولید شده توسط NEARESTINDISK لزوماً یک گراف مسطح نیست. مجموعه‌ی  $S := \{p_1, p_2, p_3, p_4, p_5\}$  را در نظر بگیرید که در آن داریم:

$p_1 = (304, 336)$ ،  $p_2 = (160, 336)$ ،  $p_3 = (304, 320)$ ،  $p_4 = (208, 320)$ ، و  $p_5 = (240, 272)$

(شکل ۹-۴ را ببینید). به راحتی می‌توان دریافت که  $p_3, p_5 \in D_{p_1 p_2}$ ،  $p_4 \notin D_{p_1 p_2}$  و  $|p_1 p_5| < |p_1 p_4|$  و  $(p_1, p_5)$  یال  $nearest(p_1, p_2)$  است. از این رو، الگوریتم NEARESTINDISK روی مجموعه نقطه‌ی  $S$ ، یال  $(p_1, p_5)$  را به  $E$  اضافه می‌کند. از طرف دیگر، دیسک  $D_{p_3 p_4}$  فقط شامل  $p_3$  و  $p_4$  است. بنابراین، الگوریتم، یال  $(p_3, p_4)$  را به  $E$  اضافه می‌کند. به وضوح، یال‌های  $(p_1, p_5)$  و  $(p_3, p_4)$  یکدیگر را قطع می‌کنند. بنابراین، گراف  $G(S, E)$  تولید شده توسط الگوریتم ۱-۴-۴ مسطح نیست.

حال، می‌توان مجموعه نقطه‌ی  $S$  را به یک مجموعه نقطه با تعداد نقاط دلخواه توسعه داد به طوری که گراف تولید شده توسط NEARESTINDISK مسطح نباشد. بدین منظور، می‌توان هر نقطه‌ای که به اندازه‌ی کافی از ۵ نقطه‌ی اصلی در  $S$  دور است، را به  $S$  اضافه کرد، زیرا اضافه کردن نقاطی که دور هستند تغییری در یال‌های  $(p_1, p_5)$  و  $(p_3, p_4)$  ایجاد نمی‌کند.

حال، ما الگوریتم اصلی را که با GREEDYWEAKLYSELFAPPROACH نمایش می‌دهیم، بیان می‌کنیم. (الگوریتم ۲-۴-۴ را ببینید). این الگوریتم، برای هر مجموعه از نقاط در صفحه یک گراف خود-گرای ضعیف مسطح می‌سازد. فرض کنید  $S$  مجموعه‌ای از  $n$  نقطه در صفحه باشد. الگوریتم، با یک مجموعه‌ی تهی  $E$  شروع می‌شود. در ابتدا، الگوریتم، همه‌ی زوج نقاط مرتب  $(p, q)$  که  $p, q \in S$  هستند را بر اساس فاصله‌ی اقلیدسی‌شان مرتب می‌کند. سپس همه‌ی زوج نقاط را به ترتیب پردازش می‌کند. حال، لحظه‌ای را که الگوریتم می‌خواهد زوج  $(p, q)$  را پردازش کند، در نظر بگیرید. همچنین، مجموعه‌ی  $E$  را در این لحظه در نظر بگیرید. فرض کنید که  $E_{pq}$  مجموعه‌ی همه‌ی یال‌های  $(p, r) \in E$  باشد به طوری که  $r \in D_{pq}$  و  $S_{pq}$  مجموعه‌ی همه‌ی نقاط  $r \in S \setminus \{p\} \cap D_{pq}$  باشد به طوری که پاره‌خط  $\overline{pr}$  هیچ یالی در  $E$  را قطع نکند.

---

**Algorithm ۴-۴-2: GREEDYWEAKLYSELFAPPROACH( $S$ )**

---

**input:** A set  $S$  of  $n$  points in the plane.  
**output:** A weakly self-approaching plane graph.

- 1 Sort the  $2\binom{n}{2}$  ordered pairs of distinct points in non-decreasing order of their distances (ties are broken arbitrarily), and store them in a list  $L$ ;
- 2  $E := \emptyset$ ;
- 3 **foreach** ordered pair  $(p, q) \in L$  **do**
- 4     Compute  $E_{pq}$  and  $S_{pq}$ ;
- 5     **if**  $E_{pq} = \emptyset$  **then**
- 6          $r :=$  the nearest point to  $p$  in  $S_{pq}$ ;
- 7          $E := E \cup \{(p, r)\}$ ;
- 8         /\* Note that by Lemma ۴-۴3-, if  $E_{pq} = \emptyset$ , then  $S_{pq} \neq \emptyset$ . \*/
- 9     **end**
- 10 **return**  $G(S, E)$ ;

---

الگوریتم هنگامی که زوج  $(p, q)$  را در نظر می‌گیرد، بررسی می‌کند که آیا  $E_{pq}$  تهی است یا خیر. اگر  $E_{pq} = \emptyset$ ، آنگاه یال  $(p, r)$  که  $r$  نزدیک‌ترین نقطه در  $S_{pq}$  به  $p$  است را به مجموعه‌ی یال‌ها اضافه می‌کند؛ در غیر این صورت، الگوریتم زوج نقطه‌ی بعدی را پردازش می‌کند. در لم‌های ۲-۴-۴

و ۳-۴-۴، لحظه‌ای را که الگوریتم (الگوریتم ۴-۴-۲ را ببینید) می‌خواهد زوج  $(p, q)$  را پردازش کند، در نظر می‌گیریم. علاوه بر این، فرض می‌کنیم گراف  $G$  که تاکنون محاسبه شده، مسطح است. حال، فرض کنید که  $x$  یک نقطه در  $D_{pq} \setminus \{p\} \cap S$  باشد به طوری که یک یال در  $E$  وجود دارد که پاره‌خط  $\overline{px}$  را قطع کند. از بین یال‌هایی که پاره‌خط  $\overline{px}$  را قطع می‌کند، فرض کنید  $(r_x, s_x) \in E$  یالی باشد که محل تقاطع آن با  $\overline{px}$  به  $p$  از همه نزدیک‌تر است. ما چنین یالی را اولین یال متقاطع با  $\overline{px}$  می‌نامیم و با نماد  $first(\overline{px})$  نمایش می‌دهیم. حال، ما مشاهده و لم‌های زیر را داریم.

**مشاهده ۴-۴-۱.** فرض کنید که  $C_{pq}$  یک دیسک بسته با قطر  $\overline{pq}$  و  $\overline{ab}$  پاره‌خطی باشد که  $\overline{pq}$  را قطع می‌کند و  $a, b \notin C_{pq}$ . هر دیسک بسته‌ای که  $a$  و  $b$  روی مرزش قرار گرفته باشد، حتماً شامل  $p$  یا  $q$  است.

**لم ۴-۴-۲.** اگر  $E_{pq} = \emptyset$  و  $S_{pq} = \emptyset$  باشد، آنگاه برای هر  $x \in D_{pq} \setminus \{p\} \cap S$  داریم  $r_x \in D_{pq}$  یا  $s_x \in D_{pq}$  که در اینجا  $first(\overline{px}) = (r_x, s_x)$  است.

**اثبات.** به عنوان فرض خلف، فرض کنید که یک نقطه‌ی  $x \in D_{pq} \setminus \{p\} \cap S$  وجود دارد به طوری که  $first(\overline{px}) = (r_x, s_x)$  و  $r_x, s_x \notin D_{pq}$  برای سادگی، فرض می‌کنیم که  $a := r_x$  و  $b := s_x$ . فرض کنید که  $c \in S$  نقطه‌ای باشد که یال  $(a, b)$  برای اولین بار هنگامی که الگوریتم، زوج  $(a, c)$  را پردازش می‌کند به  $E$  اضافه می‌شود. به وضوح، در هنگام پردازش  $(a, c)$ ، داریم  $E_{ac} = \emptyset$ . در اینجا، فرض می‌کنیم که نقطه‌ی  $a$  بالای  $\overline{px}$  و  $b$  پایین  $\overline{px}$  است. برای موقعیت‌های دیگر  $a$  و  $b$  می‌توانیم به طور مشابه لم را ثابت کنیم. برای اثبات لم، در ابتدا سه ادعای زیر را ثابت می‌کنیم:

**\* ادعای اول:**  $p \in D_{ac}$  یا  $q \in D_{ac}$ . فرض کنید که  $b' \neq a$  محل تقاطع مرز دیسک  $D_{ac}$

و خطی که از  $\overline{ab}$  می‌گذرد، باشد. از آنجایی که  $\overline{ab'}$  پاره‌خط  $\overline{pq}$  را قطع می‌کند و همچنین

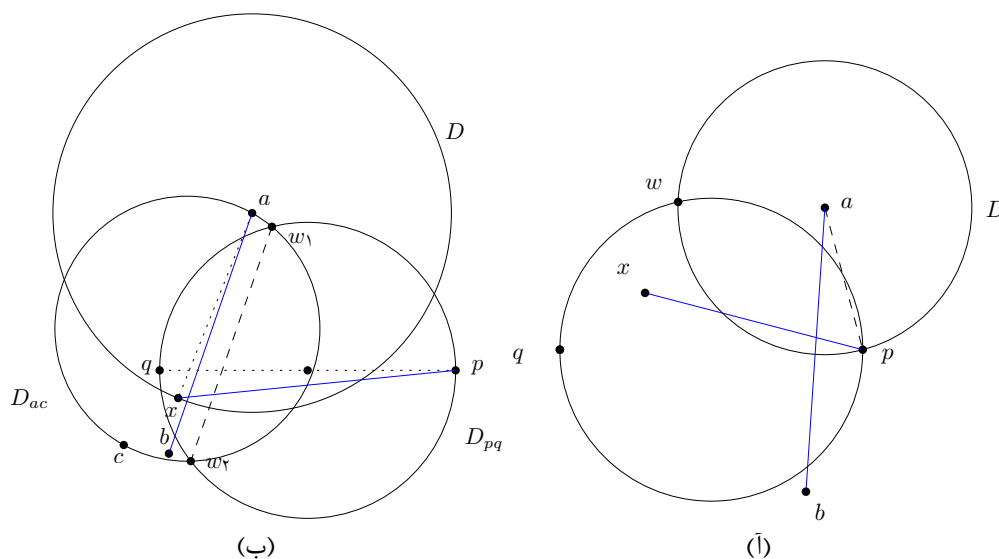
$$a, b \notin D_{pq}, \text{ بنابراین طبق مشاهده ۴-۴-۱، داریم } p \in D_{ac} \text{ یا } q \in D_{ac}.$$

**\* ادعای دوم:**  $p \in D_{ac}$  یا  $x \in D_{ac}$  بر اساس ادعای اول، داریم  $p \in D_{ac}$  یا  $q \in D_{ac}$ . اگر

$q \in D_{ac}$ ، آنگاه ادعا برقرار است. فرض کنید که  $p \notin D_{ac}$  و  $q \in D_{ac}$ . حال، ما از نقطه‌ی  $a$

روی مرز  $D_{ac}$ ، مرز این دیسک را پیمایش می‌کنیم. از آنجایی که  $p \notin D_{ac}$  و  $a, b \notin D_{pq}$





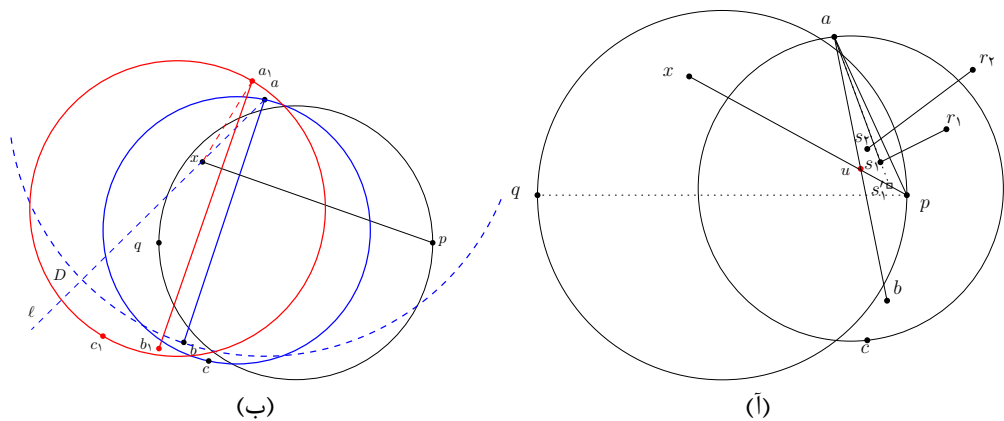
شکل ۱۰-۴: اثبات ادعای سوم در لم ۴-۴-۲. (آ): حالتی که  $p \in D_{ac}$ . (ب): حالتی که  $x \in D_{ac}$ .

بنابراین در پیمایش، با دو نقطه برخورد می‌کنیم که این دو نقطه، محل تقاطع  $D_{pq}$  و  $D_{ac}$  هستند. بنابراین از آنجایی که  $D_{pq}$  و  $D_{ac}$  همدیگر را در حداکثر دو نقطه قطع می‌کنند و از طرفی  $x$  در سمت چپ  $\overline{ab'}$  (در ادعای اول تعریف شده است) قرار دارد و همچنین  $q \in D_{ac}$ ، بنابراین، به وضوح  $x \in D_{ac}$  است.

\* **ادعای سوم:** اگر  $p \in D_{ac}$  آنگاه  $|ap| < |ab|$  و اگر  $x \in D_{ac}$  آنگاه  $|ax| < |ab|$ .

فرض کنید که  $p \in D_{ac}$  و  $D$  دایره‌ای به مرکز  $a$  و به شعاع  $|ap|$  باشد (شکل ۱۰-۴ را ببینید). واضح است که  $D$ ، دیسک  $D_{pq}$  را در دو نقطه قطع می‌کند که یکی از آن‌ها نقطه‌ی  $p$  است. فرض کنید که  $w$  نقطه‌ی دیگر تقاطع  $D$  و  $D_{pq}$  باشد. از آنجایی که  $b \notin D_{pq}$  و  $\overline{ab}$  پاره‌خط  $\overline{px}$  را قطع می‌کند، بنابراین به وضوح داریم  $b \notin D$ . از این رو،  $|ap| < |ab|$ .

حال، فرض کنید که  $x \in D_{ac}$  و  $\ell$  خطی باشد که از  $a$  و  $x$  می‌گذرد و همچنین فرض کنید که  $D$  دایره‌ای به مرکز  $a$  و به شعاع  $|ax|$  و  $w$  محل تقاطع  $D$  و  $D_{pq}$  در سمت راست  $\ell$  باشد (شکل ۱۰-۴ را ببینید). به وضوح، هر نقطه‌ی  $u$  روی  $D$  که بین  $x$  و  $w$  قرار دارد در  $D_{pq}$  است. حال، از آنجایی که  $\overline{ab}$  پاره‌خط  $\overline{px}$  را قطع می‌کند، به وضوح نقطه‌ی  $b$  در سمت راست  $\ell$  قرار دارد. همچنین، طبق فرض داریم  $b \notin D_{pq}$ . از این رو، باید داشته باشیم  $b \notin D$ .



شکل ۴-۱۱: اثبات لم ۴-۴-۲. (آ): حالتی که  $p \in D_{ac}$ . (ب): حالتی که  $x \in D_{ac}$  و  $p \notin D_{ac}$

بنابراین  $|ax| < |ab|$ .

حال، با کمک سه ادعای بالا، لم را ثابت می‌کنیم. بر طبق ادعای دوم، داریم  $p \in D_{ac}$  یا  $x \in D_{ac}$  ابتدا فرض کنید که  $p \in D_{ac}$  بر طبق ادعای سوم، داریم  $|ap| < |ab|$ . از آنجایی که  $|ap| < |ab|$  و همچنین بر طبق روش انتخاب نقطه‌ی  $b$  در الگوریتم ۴-۴-۲ و در طول پردازش  $(a, c)$ ، پاره‌خط  $\overline{ap}$  باید یالی در  $E$  را که تاکنون محاسبه شده است، قطع کند. فرض کنید که  $(r_1, s_1)$  چنین یالی باشد (شکل ۴-۱۱ آ را ببینید). از آنجایی که فرض کردیم که فاصله‌ی اقلیدسی بین  $p$  و  $u$  کم‌ترین است و همچنین از آنجایی که  $G$  در این لحظه، مسطح است، بنابراین نقاط انتهایی یال  $(r_1, s_1)$  باید داخل مثلث  $\Delta_{apu}$  باشند. بدون از دست دادن کلیت، فرض کنید که  $\Delta_{apu}$  شامل  $s_1$  است. به راحتی می‌توان پی برد که  $|as_1| < |ab|$ .

حال، پاره‌خط  $\overline{as_1}$  را در نظر بگیرید. مجدداً، بر اساس روش انتخاب نقطه‌ی  $b$ ، باید یک یال  $(r_2, s_2)$  در  $E$  باشد به طوری که پاره‌خط  $\overline{as_1}$  را قطع کند و نقاط انتهایی  $(r_2, s_2)$  (در اینجا  $s_2$ )، داخل مثلث  $\Delta_{as_1u}$  باشند. در اینجا،  $s'_1$  محل تقاطع یال  $(p, x)$  و خطی است که از نقاط  $a$  و  $s_1$  می‌گذرد. به روش مشابه، می‌توان پی برد که  $|as_2| < |ab|$ . با ادامه‌ی این فرآیند، می‌توان نتیجه گرفت که باید یک نقطه‌ی  $w \in S$  روی یال  $(a, b)$  وجود داشته باشد به طوری که  $w \neq a, b$  و یا نقطه‌ی  $w$  روی پاره‌خط  $\overline{pa}$  باشد به طوری که  $w \neq p, x$  اما این مطلب، فرض اینکه نقاط  $S$  در حالت کلی هستند را نقض می‌کند.

حال، فرض کنید که  $x \in D_{ac}$  و  $p \notin D_{ac}$  بر اساس ادعای سوم، داریم  $|ax| < |ab|$ . بر طبق

الگوریتم ۴-۴-۲، از آنجایی که  $|ax| < |ab|$ ، به راحتی می‌توان پی برد که یال  $(a_1, b_1)$  پاره‌خط  $\overline{ax}$  را قطع می‌کند. فرض کنید که  $D$  یک دیسک به مرکز  $a$  و به شعاع  $|a|$  باشد. همان‌طور که می‌بینید هیچ یک از نقاط  $a_1$  و  $b_1$  در  $D_{ac} \cap D$  نیستند (شکل ۴-۱۱ ب را ببینید) زیرا در غیر این صورت، یال  $(a, b)$  به مجموعه‌ی یالی  $E$  در هنگام پردازش زوج  $(a, c)$  اضافه نمی‌شود و به جای آن،  $(a, a_1)$  یا  $(a, b_1)$  به  $E$  اضافه می‌شود. به وضوح، نقطه‌ی  $x$  در سمت چپ  $(a_1, b_1)$  قرار دارد. حال، فرض کنید که یال  $(a_1, b_1)$  در طول پردازش یک زوج نقطه‌ای که با  $(a_1, c_1)$  نمایش می‌دهیم، به مجموعه‌ی یال‌ها اضافه شود. به راحتی می‌توان پی برد که  $x \in D_{a_1 c_1}$  و  $|a_1 x| < |a_1 b_1|$ . به طور مشابه، می‌توانیم بحث بالا را روی یال  $(a_1, b_1)$ ، جهت پیدا کردن یال  $(a_2, b_2)$  که پاره‌خط  $\overline{a_1 x}$  را قطع می‌کند، ادامه دهیم و به همین ترتیب یال  $(a_3, b_3)$  که پاره‌خط  $\overline{a_2 x}$  را قطع می‌کند را پیدا می‌کنیم. حال، فرض کنید که  $\alpha_0$  زاویه‌ی بین یال  $(a, b)$  و پاره‌خط  $\overline{ax}$  و  $\alpha_i$  زاویه‌ی بین یال  $(a_i, b_i)$  و پاره‌خط  $\overline{a_i x}$  باشد. به راحتی می‌توان پی برد که دنباله‌ی  $\alpha_0, \alpha_1, \alpha_2, \dots$  یک دنباله‌ی نزولی است. حال، فرض کنید که  $\alpha_j$  کم‌ترین مقدار این دنباله باشد. با دلایل مشابه آنچه که در بالا بحث شد، می‌توانیم یال  $(a_{j+1}, b_{j+1})$  را پیدا کنیم که پاره‌خط  $\overline{a_j x}$  را قطع کند و  $\alpha_j > \alpha_{j+1}$  که این با شیوه‌ی انتخاب  $\alpha_j$  در تناقض است. از این رو، حکم ثابت است. ■

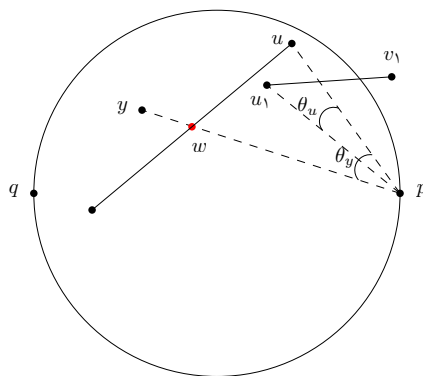
لم ۴-۴-۳. اگر  $E_{pq} = \emptyset$ ، آنگاه  $S_{pq} \neq \emptyset$ .

اثبات. به عنوان فرض خلف، فرض کنید که  $S_{pq} = \emptyset$ . طبق تعریف  $S_{pq}$ ، برای هر نقطه‌ی  $x \in S \setminus \{p\} \cap D_{pq}$  یک یال در گراف  $G$  که تاکنون محاسبه شده است، وجود دارد که  $\overline{px}$  آن را قطع می‌کند. برای هر  $x \in S \setminus \{p\} \cap D_{pq}$ ،  $\theta_x$  را به صورت زیر تعریف می‌کنیم:

$$\theta_x := \begin{cases} \angle(\overline{px}, \overline{pr_x}) & \text{اگر } s_x \notin D_{pq} \text{ و } r_x \in D_{pq} \\ \angle(\overline{px}, \overline{ps_x}) & \text{اگر } s_x \in D_{pq} \text{ و } r_x \notin D_{pq} \\ \min\{\angle(\overline{px}, \overline{pr_x}), \angle(\overline{px}, \overline{ps_x})\} & \text{اگر } r_x, s_x \in D_{pq} \end{cases}$$

که  $(r_x, s_x) := \text{first}(\overline{px})$  و  $\angle(\overline{ab}, \overline{ac})$  زاویه‌ی بین پاره‌خط‌های  $\overline{ab}$  و  $\overline{ac}$  است. توجه کنید که طبق لم ۴-۴-۲، داریم  $r_x \in D_{pq}$  یا  $s_x \in D_{pq}$ .

حال، فرض کنید که  $y$  یک نقطه در  $S \setminus \{p\} \cap D_{pq}$  باشد به طوری که  $\theta_y$  کم‌ترین باشد (شکل ۴-۱۲ را ببینید) و  $(u, v) := \text{first}(\overline{py})$ . طبق لم ۴-۴-۲، داریم  $u \in D_{pq}$  یا  $v \in D_{pq}$  بدون از



شکل ۴-۱۲: اثبات لم ۴-۴-۳.

دست دادن کلیت، فرض می‌کنیم که  $u \in D_{pq}$  حال، فرض کنید که  $first(\overline{pu}) := (u_1, v_1)$ . بر طبق لم ۴-۴-۲، داریم  $u_1 \in D_{pq}$  یا  $v_1 \in D_{pq}$ . از طرف دیگر، طبق تعریف  $first(\overline{pu})$  و از آنجایی که هیچ تقاطعی بین یال‌هایی که از قبل محاسبه شده‌اند، نیست (به جز نقاط انتهایی)، بنابراین یکی از نقاط  $u_1$  و  $v_1$  باید داخل مثلث  $\Delta_{puw}$  باشد که در اینجا،  $w$  نقطه‌ی تقاطع  $\overline{py}$  و  $first(py)$  است. بدون از دست دادن کلیت، فرض کنید که  $u_1$  چنین نقطه‌ای باشد. از آنجایی که  $u_1$  داخل مثلث  $\Delta_{puw}$  است، بنابراین، به وضوح داریم  $\theta_u < \theta_y$  که این با روش انتخاب  $y$  در تناقض است. از این رو،  $S_{pq} \neq \emptyset$ .

**قضیه ۴-۴-۴.** گراف  $G(S, E)$  تولید شده توسط الگوریتم GREEDYWEAKLYSELFAPPROACH یک گراف خود-گرای ضعیف مسطح است.

**اثبات.** مشابه اثبات قضیه‌ی ۴-۴-۱، ثابت می‌شود که  $G$  خود-گرای ضعیف است و اثبات مسطح بودن  $G$  از لم ۴-۴-۳ نتیجه می‌شود.

توجه کنید که پیاده‌سازی معمولی الگوریتم ۴-۴-۲، دارای پیچیدگی زمانی  $O(n^4)$  و پیچیدگی حافظه‌ی  $O(n^2)$  است که  $n$  تعداد نقاط است. در ادامه، نوع دیگری از پیاده‌سازی الگوریتم ۴-۴-۲ را ارائه می‌کنیم که پیچیدگی زمانی آن  $O(n^3)$  است. این پیاده‌سازی جدید از یک ماتریس صفر و یک  $W$  جهت تشخیص اینکه آیا یک پاره‌خط، یالی را در مجموعه‌ی یالی قطع می‌کند یا خیر، استفاده می‌کند. این ماتریس در طول اجرای الگوریتم، بروزرسانی می‌شود. حال، ماتریس  $W$  را در لحظه‌ای از اجرای الگوریتم در نظر بگیرید. اگر برای زوج  $(x, y)$  داشته باشیم  $W(x, y) = 0$

آنگاه این بدین معناست که پاره خط  $\overline{xy}$  هیچ یالی در گرافی که تاکنون محاسبه شده است را قطع نمی‌کند، زیرا در غیر این صورت، داریم  $W(x, y) = 1$ . به وضوح، با استفاده از ماتریس  $W$ ، در زمان  $O(1)$  می‌توانیم تشخیص دهیم که آیا پاره خط  $\overline{xy}$  با یک یالی در مجموعه‌ی یال‌ها تقاطع دارد یا خیر. از این رو، برای هر زوج  $(p, q)$  می‌توانیم  $S_{pq}$  را در زمان  $O(n)$  محاسبه کنیم. توجه کنید که ما باید ماتریس  $W$  را بعد از اضافه کردن یال  $(p, r)$  (خط  $\gamma$  از الگوریتم ۴-۴-۲ را ببینید) به مجموعه‌ی یال‌ها، بروزرسانی کنیم. بروزرسانی می‌تواند با تغییر همه‌ی درایه‌های  $W(x, y)$  و  $W(y, x)$  به عدد یک و به ازای هر  $x$  و  $y$  که در سمت چپ و راست پاره خط  $\overline{pr}$  هستند، انجام شود. به وضوح، بروزرسانی  $W$  بعد از اضافه کردن هر یال، در بدترین حالت، به  $O(n^2)$  زمان نیاز دارد. از آنجایی که گراف خروجی مسطح است، ماتریس  $W$  به اندازه‌ی  $O(n)$  مرتبه بروزرسانی می‌شود. از این رو، زمان اجرای کل برابر با  $O(n^3)$  است. لازم به ذکر است که در این پیاده‌سازی، پیچیدگی فضای حافظه‌ی الگوریتم، به وضوح برابر با  $O(n^2)$  است.

#### ۴-۵ پوشش‌های خود-گرای ضعیف

در این بخش، مفهوم  $t$ -پوشش‌های خود-گرای ضعیف را معرفی می‌کنیم و سپس دو الگوریتم جهت ساخت یک  $t$ -پوشش خود-گرای ضعیف برای یک مجموعه نقطه‌ی داده شده در صفحه و یک عدد حقیقی  $t > 1$  ارائه می‌کنیم. الگوریتم اول شبیه به ساخت گراف یائو است و الگوریتم دوم مبتنی بر مفهوم WSPD است. برای الگوریتم دوم، ما سعی کردیم که کران بالای خوبی روی تعداد یال‌های گراف تولید شده، پیدا کنیم اما موفق نشدیم. در نتیجه، این مسئله را به صورت یک مسئله‌ی حل‌نشده رها کردیم. اما برای الگوریتم اول، ما یک مجموعه نقطه‌ی  $S$  در صفحه ارائه می‌کنیم که نشان می‌دهد که تعداد یال‌های گراف تولید شده روی  $S$  از مرتبه‌ی دو است. اگرچه تعداد یال‌های گراف تولید شده توسط الگوریتم اول ممکن است از مرتبه‌ی دو باشد، اما نشان می‌دهیم که برای تعدادی از مجموعه نقاط خاص یعنی نقاط روی مرز یک مثلث و نقاط روی یک  $n$ -ضلعی منتظم، الگوریتم اول، یک  $t$ -پوشش خود-گرای ضعیف با تعداد یال خطی تولید می‌کند. در نهایت، نشان

می‌دهیم که برای هر عدد  $t > 1$  و برای هر مجموعه نقطه‌ی  $S$  که روی مرز یک مثلث قرار گرفته است، یک  $t$ -پوشش وتر-افزایشی با تعداد یال خطی روی  $S$  وجود دارد.

فرض کنید  $G$  یک گراف با مجموعه‌ی رئوس  $S$  و  $t > 1$  یک عدد حقیقی باشد. گراف  $G$  را یک  $t$ -پوشش خود-گرا می‌نامند اگر برای هر زوج  $(p, q)$  که  $p, q \in S$ ، یک مسیر خود-گرا از  $p$  به  $q$  به طول حداکثر  $t|pq|$  در  $G$  وجود داشته باشد. ما چنین مسیری را یک  $t$ -مسیر خود-گرا از  $p$  به  $q$  می‌نامیم. همچنین،  $G$  را یک  $t$ -پوشش وتر-افزایشی می‌نامند اگر برای هر زوج  $(p, q)$  که  $p, q \in S$  یک مسیر وتر-افزایشی بین  $p$  و  $q$  به طول حداکثر  $t|pq|$  وجود داشته باشد.

ما یک  $t$ -پوشش خود-گرای ضعیف را مشابه تعریف  $t$ -پوشش خود-گرا با تبدیل عبارت "خود-گرا" به "خود-گرای ضعیف" تعریف می‌کنیم. به طور مشابه، می‌توانیم  $t$ -پوشش وتر-افزایشی ضعیف را نیز تعریف کنیم. ما یک  $t$ -مسیر از  $p$  به  $q$  که خود-گرای ضعیف است را یک  $t$ -مسیر خود-گرای ضعیف از  $p$  به  $q$  می‌نامیم.

قبل از اینکه به توصیف الگوریتم‌ها بپردازیم، گزاره‌ی زیر را ثابت می‌کنیم.

**گزاره ۴-۵-۱.** یک مجموعه نقطه‌ی  $S$  شامل  $n$  نقطه در صفحه وجود دارد به طوری که بیش‌ترین درجه‌ی رئوس هر گراف خود-گرا (ضعیف) روی  $S$  نسبت به  $n$  خطی است.

**اثبات.** فرض کنید که  $S$  مجموعه‌ای از  $n$  نقطه در صفحه باشد به طوری که  $n - 1$  نقطه از آن، روی محیط یک دایره قرار گرفته‌اند و یک نقطه‌ی دیگر به نام  $c$  در مرکز این دایره است. به راحتی می‌توان پی برد که برای  $p \in S$  که  $p \neq c$  فقط شامل نقاط  $c$  و  $p$  است. بنابراین، هر گراف خود-گرای (ضعیف)  $G$  روی  $S$  باید شامل یال  $(c, p)$  باشد. از این رو درجه‌ی  $c$  در  $G$  برابر با  $n - 1$  است. این، اثبات را کامل می‌کند. ■

#### ۴-۵-۱ الگوریتمی مشابه الگوریتم گراف یائو

فرض کنید  $S$  مجموعه‌ای از  $n$  نقطه در صفحه و  $k \geq 2$  یک عدد صحیح باشد. مشابه تعریف گراف یائو، ما گراف  $W_k$  را روی مجموعه‌ی  $S$  به صورت زیر تعریف می‌کنیم. در هر نقطه‌ی  $u \in S$

ما صفحه را به  $k$  مخروط به مرکز  $u$  و دهانه‌ی  $\frac{\sqrt{\pi}}{k}$  افراز می‌کنیم. برای هر  $u \in S$  و هر مخروط  $C_u$ ، کارهای زیر را انجام می‌دهیم: فرض کنید  $S_{C,u} = S \cap C_u$ . ما نقطه‌ی  $u$  را به همه‌ی نقاط  $p \in S_{C,u}$  که  $D_{up} \cap S_{C,u} = \{u, p\}$ ، توسط یالی متصل می‌کنیم. به راحتی می‌توان پی برد که  $Y_k$  زیرگرافی از  $W_k$  است. از این رو، ضریب کشش گراف  $W_k$  حداکثر  $\frac{1}{1-\sqrt{\sin(\theta/2)}}$  است که در اینجا داریم  $\theta = \frac{\sqrt{\pi}}{k}$ . در ادامه نشان می‌دهیم که  $W_k$  یک گراف خود-گرای ضعیف است.

**قضیه ۴-۵-۲.** فرض کنید  $k \geq 2$  یک عدد صحیح باشد. گراف  $W_k$  یک گراف خود-گرای ضعیف است.

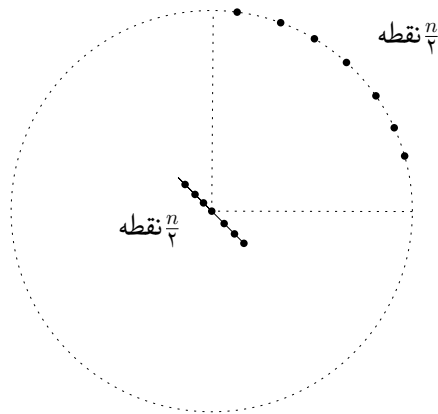
**اثبات.** نشان می‌دهیم که برای هر زوج  $p, q \in S$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $W_k$  وجود دارد. اثبات از طریق استقرا روی رتبه‌ی فاصله‌ی اقلیدسی  $|pq|$  است.

**پایه‌ی استقرا:** فرض کنید که زوج  $(p, q)$  نزدیک‌ترین زوج نقطه باشد. به وضوح،  $W_k$  شامل یال  $(p, q)$  است. از این رو، یک مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $W_k$  وجود دارد که این مسیر، همان یال  $(p, q)$  است.

**فرض استقرا:** فرض کنید که برای هر زوج  $x, y \in S$  با شرط  $|xy| < |pq|$ ، یک مسیر خود-گرای ضعیف از  $x$  به  $y$  در  $W_k$  باشد.

**حکم استقرا:** نشان می‌دهیم که یک مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $W_k$  وجود دارد. فرض کنید که  $C_p$  یک مخروط به مرکز  $p$  و شامل  $q$  باشد. اگر  $(p, q)$  یالی در  $W_k$  باشد، آنگاه به وضوح یک مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $W_k$  وجود دارد. حال، فرض کنید که  $(p, q)$  یالی در  $W_k$  نباشد. بنابراین، بر طبق ساخت  $W_k$ ،  $D_{pq} \cap S_{C,p}$  شامل حداقل یک نقطه در  $S \setminus \{p, q\}$  است. فرض کنید  $r$  نزدیک‌ترین نقطه در  $D_{pq} \cap S_{C,p}$  به  $p$  باشد. طبق ساخت  $W_k$ ،  $(p, r)$  یک یال در  $W_k$  است. از طرفی دیگر، از آنجایی که  $|rq| < |pq|$ ، بر طبق فرض استقرا، یک مسیر خود-گرای ضعیف از  $r$  به  $q$  در  $W_k$  وجود دارد که ما آن را  $Q$  می‌نامیم. طبق مشاهده‌ی ۴-۳-۱، مسیر  $P := \{(p, r)\} \cup Q$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $W_k$  است. ■

توجه کنید که شکل ۴-۱۳ یک مجموعه نقطه برای  $k = 4$  را نشان می‌دهد که در آن تعداد یال‌های  $W_k$  از مرتبه‌ی دو می‌باشد.



شکل ۴-۱۳: یک مجموعه‌ای از  $n$  نقطه. برای  $k = 4$ ، گراف  $W_k$  روی این مجموعه نقطه دارای  $O(n^2)$  یال است.

حال، فرض کنید که  $c > 0$  یک عدد حقیقی ثابت و  $S$  مجموعه‌ای از  $n$  نقطه در صفحه با خاصیت زیر باشد: برای هر نقطه‌ی  $u \in S$  و هر مخروط  $C_u$  در ساخت  $W_k$ ، نقطه‌ی  $u$  به حداکثر  $c$  نقطه در  $C_u$  وصل می‌شود. ما چنین خاصیتی را خاصیت  $c$ -ثابت می‌نامیم. به وضوح گراف  $W_k$  روی هر مجموعه‌ی  $S$  شامل  $n$  نقطه با خاصیت  $c$ -ثابت، دارای  $ckn$  یال است.

در ادامه، دو مجموعه نقطه با خاصیت  $c$ -ثابت برای یک مقدار ثابت  $c > 0$ ، ارائه می‌کنیم. فرض کنید  $S$  مجموعه‌ای از  $n$  نقطه روی مرز یک مثلث  $\Delta_{ABC}$  و  $S'$  مجموعه‌ای از  $n$  نقطه باشد که روی رأس‌های یک  $n$ -ضلعی منتظم قرار گرفته است. به راحتی می‌توان پی برد که  $S$  یک مجموعه با خاصیت  $5$ -ثابت و  $S'$  یک مجموعه با خاصیت  $2$ -ثابت است.

خلاصه‌ی نتایج بالا در قضیه‌ی زیر آمده است.

**قضیه ۴-۵-۳.** فرض کنید  $k \geq 2$  یک عدد صحیح،  $\theta = \frac{\gamma\pi}{k}$  و  $S$  مجموعه‌ای از  $n$  نقطه روی مرز یک مثلث باشد. گراف  $W_k$  روی  $S$  یک  $t$ -پوشش خود-گرای ضعیف به ازای  $t = \frac{1}{1 - \gamma \sin(\theta/\gamma)}$  است. همچنین  $W_k$  شامل حداکثر  $O(kn)$  یال است. برای هر مجموعه‌ای از  $n$  نقطه که روی رأس‌های یک  $n$ -ضلعی منتظم قرار گرفته‌اند، نتایج یکسان به طور مشابه بدست می‌آید.

فرض کنید  $S$  مجموعه‌ای از  $n$  نقطه روی مرز یک مثلث  $\Delta_{ABC}$  و  $k \geq 2$  یک عدد صحیح باشد. لازم به ذکر است که  $W_k$  روی  $S$ ، یک گراف خود-گرای ضعیف است، اما لزوماً یک گراف خود-گرا نیست. در ادامه، برای هر عدد حقیقی  $t > 1$ ، نشان می‌دهیم که چگونه یک  $t$ -پوشش وتر-افزایشی روی  $S$  با تعداد یال‌های خطی بسازیم. روش ساخت، مشابه ساخت گراف  $W_k$  است. ما



فرض می‌کنیم که مثلث  $\Delta_{ABC}$  یک مثلث حاده است. برای حالتی که مثلث، منفرجه یا قائم‌الزاویه است با اندکی تغییرات، نتایج یکسانی بدست می‌آید.

یک ضلع  $\overline{XY}$  از مثلث، توسط یک نقطه‌ی  $p$  در مخروط  $C_p$  قابل مشاهده است هرگاه  $\overline{XY} \cap C_p \neq \emptyset$ . فرض کنید  $p$  یک نقطه در  $S \cap \overline{AB}$  و ضلع دیگری غیر از  $\overline{AB}$  از مثلث باشد که توسط  $p$  در مخروط  $C_p$  قابل مشاهده است. همچنین، فرض کنید  $p'$  تصویر عمود  $p$  روی  $\overline{XY}$  باشد. ما مجموعه‌ی  $\overline{XY}_p$  را به صورت زیر تعریف می‌کنیم: فرض کنید که  $x_1 \in S \cap \overline{p'X}$  نزدیک‌ترین نقطه به  $p'$  در  $C_p$  باشد. اگر چنین نقطه‌ی  $x_1$  وجود نداشته باشد، آنگاه فرض می‌کنیم  $x_1 = \text{nil}$ . فرض کنید  $y_1 \in S \cap \overline{p'Y}$  نزدیک‌ترین نقطه به  $p'$  در  $C_p$  باشد. اگر چنین نقطه‌ی  $y_1$  وجود نداشته باشد، آنگاه فرض می‌کنیم که  $y_1 = \text{nil}$ . حال، هر کدام از  $x_1$  و  $y_1$  که  $\text{nil}$  نیستند را به  $\overline{XY}_p$  اضافه می‌کنیم.

الگوریتم ساخت به صورت زیر است. در هر نقطه‌ی  $p \in S$ ،  $k$  مخروط ( $k \geq 2$ ) به مرکز  $p$  و با دهانه‌ی  $\frac{\gamma\pi}{k}$  رسم می‌کنیم. سپس، برای هر نقطه‌ی  $p \in S$  و هر مخروط  $C_p$ ، عملیات زیر را انجام می‌دهیم: بدون از دست دادن کلیت، فرض کنید که  $p \in \overline{AB}$  برای هر  $\overline{XY} \neq \overline{AB}$  که توسط  $p$  در  $C_p$  قابل مشاهده است، ما یک یال بین  $p$  و هر نقطه در  $\overline{XY}_p$  رسم می‌کنیم. همچنین فرض کنید که  $p \neq z$  نزدیک‌ترین نقطه به  $p$  در  $S \cap \overline{AB}$  و در  $C_p$  باشد. اگر چنین نقطه‌ی  $z$  وجود نداشته باشد، آنگاه فرض می‌کنیم  $z = \text{nil}$ . حال، اگر  $z \neq \text{nil}$ ، آنگاه یال  $(p, z)$  را به  $E$  اضافه می‌کنیم. فرض کنید که  $T_k$  گراف تولید شده توسط الگوریتم بالا باشد. مشابه استدلالی که نشان دادیم  $W_k$  یک  $t$ -پوشش خود-گرای ضعیف به ازای  $t = \frac{1}{1-\gamma \sin(\theta/\gamma)}$  است، می‌توانیم به راحتی ثابت کنیم که  $T_k$  یک  $t$ -پوشش وتر-افزایشی با  $t = \frac{1}{1-\gamma \sin(\theta/\gamma)}$  و با  $O(kn)$  یال است.

**قضیه ۴-۵-۴.** فرض کنید  $k \geq 2$  یک عدد صحیح،  $\theta = \frac{\gamma\pi}{k}$  و  $S$  مجموعه‌ای از  $n$  نقطه روی مرز یک مثلث باشد. گراف  $T_k$  روی  $S$  یک  $t$ -پوشش وتر-افزایشی به ازای  $t = \frac{1}{1-\gamma \sin(\theta/\gamma)}$  است. همچنین، گراف  $T_k$  شامل حداکثر  $O(kn)$  یال است.

#### ۴-۵-۲ یک الگوریتم مبتنی بر WSPD

در این بخش، یک  $t$ -پوشش خود-گرای ضعیف روی یک مجموعه نقطه‌ی داده شده‌ی  $S$  در صفحه، با بکار بردن WSPD می‌سازیم. ما الگوریتم ساخت را WSPDWEAKLYSELFAPPROACHSPANNER می‌نامیم (الگوریتم ۴-۵-۱ را ببینید). فرض کنید  $s = \frac{2t+4}{t-1}$  و  $\{(A_i, B_i)\}_{i=1}^m$  یک WSPD برای  $S$  نسبت به پارامتر  $s$  باشد. ما با یک مجموعه‌ی تهی  $E$  شروع می‌کنیم. برای هر زوج  $(A_i, B_i)$  که  $1 \leq i \leq m$ ، الگوریتم، همه‌ی زوج نقاط  $(p, q)$  که  $p \in A_i$  و  $q \in B_i$  یا  $q \in A_i$  و  $p \in B_i$  را به صورت صعودی و بر اساس فاصله‌ی اقلیدسی  $|pq|$  مرتب می‌کند. سپس، الگوریتم، زوج نقاط را به ترتیب پردازش می‌کند. لحظه‌ای را در نظر بگیرید که الگوریتم می‌خواهد زوج  $(p, q)$  را پردازش کند. بدون از دست دادن کلیت، فرض کنید که  $p \in A_i$  و  $q \in B_i$ . الگوریتم بررسی می‌کند که آیا یک نقطه‌ی  $r \in B_i \cap D_{pq}$  که  $(p, r) \in E$  وجود دارد یا خیر. اگر چنین نقطه‌ی  $r$  وجود داشته باشد، آنگاه زوج نقطه‌ی بعدی را پردازش می‌کند. اما اگر چنین نقطه‌ی  $r$  وجود نداشته باشد، آنگاه الگوریتم یک نقطه‌ی دلخواه  $r \in B_i \setminus \{p\} \cap D_{pq}$  را انتخاب می‌کند و یال  $(p, r)$  را به  $E$  اضافه می‌کند.

---

#### Algorithm ۴-۵-1: WSPDWEAKLYSELFAPPROACHSPANNER( $S, t$ )

---

**input:** A set  $S$  of  $n$  points in the plane and a real number  $t > 1$ .

**output:** A weakly self-approaching  $t$ -spanner.

```

1  $s := \frac{2t+4}{t-1}$ ;
2 Let  $\{(A_i, B_i)\}_{i=1}^m$  be the WSPD of  $S$  with respect to  $s$ ;
3  $E := \emptyset$ ;
4 for  $i = 1$  to  $m$  do
5   Sort increasingly all ordered pairs of points  $(p, q)$  with  $p \in A_i$  and  $q \in B_i$  or
    $p \in B_i$  and  $q \in A_i$  according to their Euclidean distance  $|pq|$ . Let  $L$  be the
   sorted list;
6   foreach ordered pair  $(p, q) \in L$  do
7     /* suppose that  $p \in A_i$  and  $q \in B_i$  */
8     if there is no point  $r \in B_i \cap D_{pq}$  with  $(p, r) \in E$  then
9       Choose an arbitrary point  $r \in B_i \setminus \{p\} \cap D_{pq}$ ;
10       $E := E \cup \{(p, r)\}$ ;
11    end
12  end
13 return  $G(S, E)$ ;
```

---

قضیه ۴-۵-۵. گراف  $G(S, E)$  تولید شده توسط الگوریتم WSPDWEAKLYSELFAPPROACHSPANNER

یک  $t$ -پوشش خود-گرای ضعیف روی  $S$  است.

**اثبات.** ادعا می‌کنیم که برای هر دو نقطه‌ی  $p$  و  $q$  در  $S$ ، یک  $t$ -مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $G$  وجود دارد. اثبات از طریق استقرا روی رتبه‌ی فاصله‌ی اقلیدسی  $|pq|$  است. فرض کنید که  $\{A_i, B_i\}_{i=1}^m$  یک WSPD برای  $S$  باشد که در الگوریتم از آن استفاده شده است.

**پایه‌ی استقرا:** فرض کنید که  $(p, q)$  نزدیک‌ترین زوج نقطه در  $S$  باشد. به راحتی می‌توان پی برد که الگوریتم، یال  $(p, q)$  را به  $E$  اضافه می‌کند و بنابراین، ادعا برقرار است.

**فرض استقرا:** فرض کنید که برای هر دو نقطه‌ی  $u$  و  $v$  با شرط  $|uv| < |pq|$ ، ادعا برقرار باشد.

**حکم استقرا:** ثابت می‌کنیم که یک  $t$ -مسیر خود-گرای ضعیف از  $p$  به  $q$  در  $G$  وجود دارد. فرض کنید که  $\{A_i, B_i\}$  ( $1 \leq i \leq m$ ) یک زوج WSPD باشد به طوری که  $p \in A_i$  و  $q \in B_i$  یا  $p \in B_i$  و  $q \in A_i$ . بدون از دست دادن کلیت، فرض کنید که  $p \in A_i$  و  $q \in B_i$ . بر طبق الگوریتم، یک نقطه‌ی  $r \in B_i \cap D_{pq}$  وجود دارد به طوری که  $(p, r) \in E$ . اگر  $r = q$ ، آنگاه ادعا برقرار است. حال، فرض کنید که  $r \neq q$ . بنابراین داریم  $|rq| < |pq|$ . حال، با توجه به فرض استقرا، یک  $t$ -مسیر خود-گرای ضعیف از  $r$  به  $q$  وجود دارد. چنین مسیری را با  $Q$  نمایش می‌دهیم. حال، مسیر  $P := \{(p, r)\} \cup Q$  را در نظر بگیرید. با توجه مشاهده‌ی ۴-۳-۱، واضح است که  $P$  یک مسیر خود-گرای ضعیف از  $p$  به  $q$  است. علاوه بر این، طبق لم ۳-۲-۲، داریم:

$$|P| = |pr| + |Q| \leq |pr| + t|rq| \leq \left(1 + \frac{4}{s}\right)|pq| + \frac{2t}{s}|pq| = t|pq|.$$

■

رابطه‌ی بالا، ادعا را اثبات می‌کند.

## ۶-۴ نتیجه‌گیری

در این فصل، نشان داده شد که یک گراف وتر-افزایشی مسطح برای هر مجموعه نقطه‌ای که روی مرز یک مثلث منفرجه یا قائم‌الزاویه قرار گرفته است، وجود دارد. همچنین نشان داده شد که برای هر مجموعه نقطه‌ای که روی مرز یک مثلث حاده قرار دارد، یک گراف وتر-افزایشی مسطح با اضافه کردن حداکثر ۳ نقطه‌ی اشتاینری وجود دارد. علاوه بر این، الگوریتمی ارائه شد که در

زمان چندجمله‌ای تشخیص می‌دهد که آیا یک گراف هندسی داده شده، خود-گرای ضعیف است یا خیر. همچنین مسئله‌ی ساخت یک گراف خود-گرای ضعیف مسطح و مسئله‌ی ساخت  $t$ -پوشش خود-گرای ضعیف برای هر مجموعه نقطه در صفحه بررسی شد و نشان داده شد که برای هر عدد حقیقی  $t > 1$  و هر مجموعه نقطه‌ی  $S$  که روی مرز یک مثلث قرار گرفته است، یک  $t$ -پوشش وتر-افزایشی با  $O(kn)$  یال وجود دارد که  $k$  فقط وابسته به  $t$  است.

ما مسائل زیر را به عنوان مسائل حل نشده مطرح می‌کنیم:

۱. آیا برای هر مجموعه‌ی  $S$  از نقاط در صفحه گراف خود-گرای مسطحی وجود دارد که رئوس آن  $S$  باشد

۲. آیا الگوریتمی وجود دارد که در زمان  $o(n^3)$  یک گراف خود-گرای ضعیف مسطح برای هر مجموعه نقطه در صفحه بسازد؟

۳. آیا برای هر مجموعه‌ی  $S$  از نقاط در صفحه یک  $t$ -پوشش خود-گرا (ضعیف) با تعداد یال کم وجود دارد که رئوس آن  $S$  باشد؟

۴. آیا یک الگوریتم وجود دارد که در زمان خطی تشخیص دهد که آیا یک درخت هندسی داده شده، خود-گرای ضعیف است یا خیر؟

## فصل ۵

### ادغام پوشش‌های هندسی و کاربردهای آن

در حوزه‌ی شبکه‌های پوششی هندسی، معمولاً با مسئله‌ی زیر رو به رو می‌شویم: برای یک عدد حقیقی  $t > 1$  و یک مجموعه‌ی داده شده چگونه می‌توان یک  $t$ -پوشش با اندازه‌ی کوچک، روی آن مجموعه و در زمان بهینه ساخت؟ تاکنون مسئله‌ی ساخت یک  $t$ -پوشش با اندازه‌ی کوچک و در زمان کارا، به طور گسترده مطالعه شده است. برای مشاهده‌ی الگوریتم‌های اصلی ساخت پوشش‌ها، به کتابی که توسط ناراسیمهان و اسمید [۴۴] نوشته شده است، می‌توان مراجعه کرد. حال، فرض کنید  $G_1$  و  $G_2$  دو  $t$ -پوشش به ترتیب برای مجموعه نقاط  $P_1$  و  $P_2$  (در فضای اقلیدسی) باشند. سؤالی که ممکن است پیش آید این است که چگونه می‌توان یک  $t$ -پوشش با اندازه‌ی کوچک روی  $P_1 \cup P_2$  با فقط اتصال  $G_1$  به  $G_2$  و با بکار بردن تعداد کمی یال بین آن‌ها، ساخت؟ این سؤال ممکن است در دنیای واقعی نیز به وجود آید. اتصال دو کشور به وسیله‌ی ساخت تعدادی جاده بین آن‌ها و یا اتصال دو شبکه‌ی آب به وسیله‌ی ایجاد تعدادی خط لوله‌ی آب، مثال‌هایی از این مسئله هستند. فرض کنید که  $t > 1$  یک عدد حقیقی دلخواه باشد. ما مسئله‌ی ادغام پوشش‌ها را به صورت زیر تعریف می‌کنیم:

**تعریف ۱-۱-۵** (مسئله‌ی ادغام پوشش‌ها). فرض کنید دو  $t$ -پوشش  $G_1(P_1, E_1)$  و  $G_2(P_2, E_2)$  در صفحه داریم به طوری که  $P_1 \cap P_2 = \emptyset$  باشد. چگونه می‌توان یک  $t$ -پوشش  $G$  روی مجموعه نقطه‌ی  $P_1 \cup P_2$  با اضافه کردن تعداد کمی یال و فقط بین  $G_1$  و  $G_2$  ساخت؟

تأکید می‌کنیم که در مسئله‌ی ادغام پوشش‌ها، گراف  $G$  باید فقط به وسیله‌ی اضافه کردن تعدادی یال بین  $G_1$  و  $G_2$  ساخته شود. زیرا در غیر این صورت، می‌توان با کمک برخی از الگوریتم‌های معروف مانند الگوریتم حریصانه، گراف یائو و  $\Theta$ -گراف، روی  $P_1 \cup P_2$  یک  $t$ -پوشش با اندازه‌ی کوچک ساخت. لازم به ذکر است که می‌توان  $G_1$  و  $G_2$  را با اضافه کردن همه‌ی یال‌های بین  $G_1$  و  $G_2$  ادغام کرد که در این صورت به یال‌های زیادی نیاز است.

در طول این فصل، فرض می‌کنیم که  $t > 1$  یک عدد حقیقی ثابت و  $n_1$  و  $n_2$  به ترتیب تعداد اعضای  $P_1$  و  $P_2$  باشند و بدون از دست دادن کلیت، فرض می‌کنیم  $n_1 \leq n_2$ . در این فصل، یک

الگوریتم جهت حل مسئله‌ی ادغام پوشش‌ها پیشنهاد می‌کنیم که به  $O(n_1 \log n_1 + (n_1 + n_2) \log n_2)$  زمان نیاز دارد. علاوه بر این، با بکار بردن این الگوریتم پیشنهادی، ما تعدادی الگوریتم تقسیم و غلبه، جهت ساخت یک  $t$ -پوشش و یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا برای یک مجموعه نقطه‌ی داده شده در صفحه، ارائه می‌کنیم که در اینجا  $\mathcal{H}_k$  خانواده‌ی همه‌ی نیم‌صفحاتی است که مرزشان موازی با یک خط در یک مجموعه‌ی متناهی از خطوط خاص است (در بخش‌های بعدی، ما  $\mathcal{H}_k$  را به صورت دقیق تعریف می‌کنیم). توجه کنید که الگوریتم‌های تقسیم و غلبه‌ی پیشنهادی ما برای ساخت یک  $t$ -پوشش، به طور مجانبی بهینه نیستند و در نتیجه با الگوریتم‌های معروف مانند  $\Theta$ -گراف و غیره، قابل قیاس نیستند. از این رو، هدف ما از پیشنهاد این الگوریتم‌ها، فقط بیان کاربردهایی از مسئله‌ی ادغام پوشش‌ها است. علاوه بر این، نشان می‌دهیم که الگوریتم تقسیم و غلبه‌ی پیشنهادی، می‌تواند جهت ساخت یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا بر روی مجموعه‌ای از  $n$  نقطه در صفحه با  $O(n \log n)$  یال و در زمان  $O(n \log^2 n)$ ، بکار برده شود. لازم به ذکر است که آبام و همکارانش [۲] با بکار بردن مفهوم تجزیه‌ی زوجی نیم-مجزا<sup>۱</sup> (SSPD)، یک الگوریتم، جهت ساخت یک  $t$ -پوشش تحمل‌پذیر  $C$ -خطا با  $O(n \log n)$  یال و در زمان  $O(n \log^2 n)$  برای مجموعه‌ای از  $n$  نقطه در صفحه ارائه کردند که در اینجا  $C$  خانواده‌ی همه ناحیه‌های محدب در صفحه است. از آنجایی که در الگوریتم پیشنهادی ما، نیاز به محاسبه‌ی SSPD برای یک مجموعه نقطه نیست، به نظر می‌رسد که برای ناحیه‌های  $\mathcal{H}_k$ ، الگوریتم پیشنهادی ما، در مقایسه با الگوریتم آبام و همکارانش، ساده‌تر است.

## ۲-۵ ادغام پوشش‌ها

در ادامه، یک الگوریتم که آن را MERGESPANNERS می‌نامیم، جهت حل مسئله‌ی ادغام پوشش‌ها ارائه می‌کنیم. این الگوریتم، مبتنی بر الگوریتم  $\Theta$ -گراف است. توجه کنید که از آنجایی که الگوریتم  $\Theta$ -گراف می‌تواند به ابعاد بالاتر تعمیم داده شوند [۴۴]، بنابراین نتایج این بخش می‌توانند به سادگی به ابعاد بالاتر تعمیم داده شوند. از این رو، ما فقط روی فضای  $\mathbb{R}^2$  متمرکز می‌شویم.

<sup>۱</sup>semi-separated pair decomposition

فرض کنید که  $k > 1$  یک عدد صحیح باشد. ما صفحه را به  $k$  مخروط با دهانه‌ی  $\frac{\sqrt{\pi}}{k}$  و به مرکز مبدأ مختصات افراز می‌کنیم و مجموعه‌ی این  $k$  مخروط را با  $\mathcal{C}_k$  نمایش می‌دهیم. برای هر مخروط  $C \in \mathcal{C}_k$ ، فرض کنید که  $\ell_C$  یک خط ثابت دلخواه باشد که از مبدأ می‌گذرد و داخل  $C$  قرار دارد. همچنین برای هر نقطه‌ی  $p$  در صفحه، فرض کنید که  $C_p := C + p$  مخروط حاصل از انتقال مرکز  $C$  به  $p$  باشد. به طور مشابه، داریم  $\ell_{C_p} := \ell_C + p$  ایده‌ای که در الگوریتم حل مسئله‌ی ادغام پوشش‌ها نهفته است، به صورت زیر می‌باشد: برای هر مخروط  $C \in \mathcal{C}_k$  و برای هر رأس  $p$  در  $G_1$ ،  $p$  را به یک نقطه‌ی  $r \in C_p \cap P_2$  متصل می‌کنیم به طوری که تصویر نقطه‌ی  $r$  روی  $\ell_{C_p}$  نزدیک‌ترین تصویر به  $p$  باشد. برای مشاهده‌ی جزئیات بیشتر الگوریتم، الگوریتم ۵-۲-۲ را ببینید.

---

**Algorithm ۵-۲-2:** MERGESPANNERS( $G_1, G_2, t$ )

---

**input:**  $t$ -spanners  $G_1(P_1, E_1)$  and  $G_2(P_2, E_2)$ , and suppose that  $|P_1| \leq |P_2|$ .

**output:**  $t$ -spanner  $G(P_1 \cup P_2, E)$  which is obtained by merge of  $G_1$  and  $G_2$ .

---

- 1 Select an angle  $0 < \theta < \pi/4$  such that  $\cos \theta - \sin \theta \geq 1/t$ ;
  - 2 Partition the plane into  $k = \frac{2\pi}{\theta}$  cones. We denote the set of cones that partition the plane by  $\mathcal{C}_k$ ;
  - 3  $E := E_1 \cup E_2$ ;
  - 4 **foreach** cone  $C$  in  $\mathcal{C}_k$  **do**
  - 5     **foreach**  $p$  in  $P_1$  **do**
  - 6          $C_p := C + p$ ;
  - 7          $r :=$  a point of  $C_p \cap P_2$  whose orthogonal projection onto  $\ell_{C_p}$  is closest to  $p$ ;  
        /\* let  $\ell_C$  be a fixed ray that emanates from the origin and that is contained in  $C$  and  $\ell_{C_p} := \ell_C + p$ . The ray  $\ell_C$  can be chosen arbitrarily. \*/
  - 8          $E := E \cup \{(p, r)\}$ ;
  - 9         /\*  $(p, r)$  is an edge. \*/
  - 5     **end**
  - 4 **end**
  - 10 **end**
  - 11 **return**  $G(P_1 \cup P_2, E)$ ;
- 

در ادامه، درستی الگوریتم MERGESPANNERS را ثابت می‌کنیم. در ابتدا، لم هندسی زیر را

بیان می‌کنیم.

لم ۵-۲-۱ ([۴۴]). فرض کنید که  $k \geq 8$  یک عدد صحیح،  $\theta = \frac{\sqrt{\pi}}{k}$ ،  $p$  و  $q$  دو نقطه‌ی متمایز در صفحه و  $C$  یک مخروط در  $\mathcal{C}_k$  باشند به طوری که  $q \in C_p$  باشد. اگر  $r$  یک نقطه در  $C_p$  باشد به طوری که فاصله‌ی تصویر عمود  $r$  روی خط  $\ell_{C_p}$  تا نقطه‌ی  $p$  کم‌تر یا مساوی فاصله‌ی تصویر عمود

$$q \text{ روی خط } \ell_{C_p} \text{ تا نقطه‌ی } p \text{ باشد، آنگاه } |rq| \leq |pq| - (\cos \theta - \sin \theta)|pr|.$$



**قضیه ۲-۲-۵.** اگر  $G_1$  و  $G_2$  به ترتیب دو  $t$ -پوشش برای  $P_1$  و  $P_2$  باشند، آنگاه گراف  $G$  تولید شده توسط الگوریتم  $\text{MERGESPANNERS}(G_1, G_2, t)$ ، یک  $t$ -پوشش برای  $P_1 \cup P_2$  است و فقط با اضافه کردن حداکثر  $kn_1$  یال بین  $G_1$  و  $G_2$  ساخته می‌شود.

**اثبات.** بر طبق الگوریتم ۲-۲-۵، واضح است که  $G$  فقط با اضافه کردن حداکثر  $kn_1$  یال بین  $G_1$  و  $G_2$  ساخته می‌شود. حال، ثابت می‌کنیم که  $G$  یک  $t$ -پوشش برای  $P_1 \cup P_2$  است. فرض کنید که  $p$  و  $q$  دو نقطه‌ی متمایز در  $P_1 \cup P_2$  باشند. ادعا می‌کنیم که یک  $t$ -مسیر بین  $p$  و  $q$  در  $G$  وجود دارد.

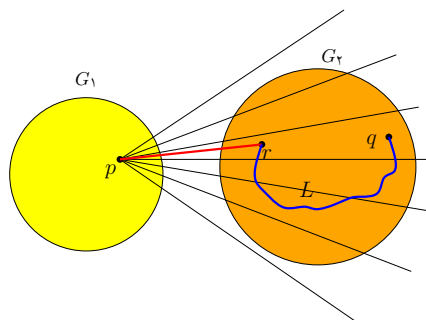
اگر  $p, q \in P_1$  و یا  $p, q \in P_2$  باشند، از آنجایی که  $G_1$  و  $G_2$  به ترتیب  $t$ -پوشش‌هایی برای  $P_1$  و  $P_2$  هستند، بنابراین، ادعا برقرار است. حال، بدون از دست دادن کلیت، فرض کنید که  $p \in P_1$  و  $q \in P_2$  و  $C$  یک مخروط در  $C_k$  باشد به طوری که  $q \in C_p$  باشد. توجه کنید که بر طبق الگوریتم، داریم  $k = \frac{\sqrt{t}}{\theta}$  و  $0 < \theta < \pi/4$  به طوری که  $\cos \theta - \sin \theta \geq 1/t$ . فرض کنید که الگوریتم، هنگامی که نقطه‌ی  $p$  و مخروط  $C_p$  را پردازش می‌کند، یال  $(p, r)$  را به  $G$  اضافه کند. اگر  $r = q$ ، بنابراین، به وضوح، ادعا برقرار است. حال، فرض کنید که  $r \neq q$ . از آنجایی که  $G_2$  یک  $t$ -پوشش برای  $P_2$  است، بنابراین، یک  $t$ -مسیر  $L$  بین  $r$  و  $q$  در  $G_2$  وجود دارد (شکل ۱-۵ را ببینید). حال، مسیر  $Q := \{(p, r)\} \cup L$  که یک مسیر بین  $p$  و  $q$  در  $G$  است را در نظر بگیرید. با استفاده از لم ۱-۲-۵، داریم:

$$\begin{aligned} |Q| &= |pr| + |L| \leq |pr| + t|rq| \\ &\leq |pr| + t(|pq| - (\cos \theta - \sin \theta)|pr|) \\ &= t|pq| - (t(\cos \theta - \sin \theta) - 1)|pr| \\ &\leq t|pq| \quad (\text{از آنجایی که } \cos \theta - \sin \theta \geq 1/t). \end{aligned}$$

■ از این رو،  $Q$  یک  $t$ -مسیر بین  $p$  و  $q$  در  $G$  است. این اثبات را کامل می‌کند.

برای پیاده‌سازی کارای الگوریتم  $\text{MERGESPANNERS}$ ، ما ایده‌ای که در الگوریتم بهینه‌ی  $\Theta$ -گراف استفاده می‌شود را بکار می‌بریم [۴۴]. در ادامه، ما یک الگوریتم جاروی صفحه<sup>۱</sup> برای پیاده‌سازی

<sup>۱</sup>plane sweep



شکل ۵-۱: اثبات قضیه‌ی ۵-۲-۲.

بهینه‌ی MERGESPANNERS ارائه می‌کنیم و از نمادها و اصطلاحاتی که در فصل ۴ کتاب [۴۴] آمده است، استفاده می‌کنیم.

مشابه پیاده‌سازی  $\Theta$ -گراف، ما مسئله‌ی Find-LeftMost-In-Translated-HalfPlane را که در ادامه آمده است، در نظر می‌گیریم.

**مسئله.** (Find-LeftMost-In-Translated-HalfPlane) [۴۴]. فرض کنید که  $h$  یک خط غیر عمودی ثابت باشد که از مبدأ مختصات می‌گذرد. می‌خواهیم یک مجموعه‌ی  $S$  شامل  $n$  نقطه را در ساختمان داده‌ای که عملیات زیر را پشتیبانی می‌کند، ذخیره کنیم:

\*  $\text{MinAbove}(p)$ : نقطه‌ی  $p \in S$  داده شده است، یک نقطه با کم‌ترین مختصات  $x$  در بین همه‌ی نقاط  $S$  که بالای  $h + p$  (خط  $h$  پس از انتقال از مبدأ مختصات به نقطه‌ی  $p$ ) است، محاسبه شود.

\*  $\text{Insert}(p)$ : نقطه‌ی دلخواه  $p \in \mathbb{R}^2$  را در  $S$  درج کند.

\*  $\text{Delete}(p)$ : نقطه‌ی  $p$  را از  $S$  حذف کند.

برای حل مسئله‌ی بالا، ناراسیمهان و اسمید [۴۴] یک درخت جستجوی دودویی  $T$  را بکار بردند. برای مشاهده‌ی جزئیات این مسئله و همچنین درخت  $T$ ، به کتاب [۴۴] مراجعه کنید.

فرض کنید که  $D$  یک خط جهت‌دار و عمود بر خط  $h$  باشد که از مبدأ مختصات می‌گذرد. جهت  $D$  را به صورت زیر تعریف می‌کنیم:  $D$  به سمت نیم‌صفحه‌ای اشاره می‌کند که شامل همه‌ی نقاط  $\mathbb{R}^2$  که بالای  $h$  قرار دارند، است. ترتیب القایی توسط  $D$  روی نقاط  $S$  به صورت زیر تعریف می‌شود: فرض کنید که  $p$  و  $q$  دو نقطه در  $S$  باشند و به ترتیب  $p'$  و  $q'$  تصویر عمود آن‌ها روی  $D$

باشند. اگر  $p$  کوچک‌تر یا مساوی  $q$  باشد، آنگاه می‌گوییم جهت بردار  $\overrightarrow{pq}$  با جهت  $D$  یکی است که آن را با نماد  $p \leq_D q$  نمایش می‌دهیم.

---

**Algorithm ۵-۲-4:** PLANE SWEEP MERGING( $G_1, G_2, t$ )

---

**input:** geometric  $t$ -spanners  $G_1(P_1, E_1)$  and  $G_2(P_2, E_2)$ , and  $t > 1$  (suppose that  $|P_1| \leq |P_2|$ ).

**output:** the geometric  $t$ -spanner  $G(P_1 \cup P_2, E)$ .

- 1 Select an angle  $0 < \theta < \pi/4$  such that  $\cos \theta - \sin \theta \geq 1/t$ ;
- 2  $k := 2\pi/\theta$ ;
- 3  $E := E_1 \cup E_2$ ;
- 4 **foreach** cone  $C \in \mathcal{C}_k$  **do**
  - 5 */\* Consider the lines  $h_1, h_2, D_1, D_2$  correspond to cone  $C$ . \*/*  
Sort the points of  $P_1$  and  $P_2$  according to the order induced by the directed line  $D_1$  and  $D_2$ , respectively. Assume that the lists  $L_1$  and  $L_2$  contain the sorted points of  $P_1$  and  $P_2$ , respectively;
  - 6 Initialize an empty data structure  $T$  for solving problem FIND-LEFTMOST-IN-TRANSLATED-HALFPLANE using  $h := h_2$ ;
  - 7  $\ell := n_2$ ;
  - 8 **for**  $i = n_1$  **down to** 1 **do**
    - 9 **while**  $L_1[i] \leq_{D_1} L_2[\ell]$  **do**
      - 10 | Insert point  $L_2[\ell]$  into  $T$ ;
      - 11 |  $\ell := \ell - 1$ ;
    - 12 **end**
    - 13 **if**  $\ell > 0$  **then**
      - 14 |  $\ell := \ell + 1$ ;
    - 15 **end**
    - 16 */\* Note that the points  $L_2[\ell], L_2[\ell + 1], \dots, L_2[n_2]$  are exactly the points of  $P_2$  that are below the line  $h_1 + L_1[i]$ . \*/*  
Find the point  $r$  in  $T$  that is above the line  $h_2 + L_1[i]$  and whose  $x$ -coordinate is minimum; that is, it answers the query MINABOVE( $L_1[i]$ );
    - 17  $E := E \cup \{(L_1[i], r)\}$ ;
    - 18 **end**
  - 19 **end**
  - 20 **return**  $E$ ;

---

لم ۵-۲-۳ (لم ۴.۱.۹، [۴۴]). درخت جستجوی دودویی  $T$ ، مسئله *Find-LeftMost-In-Translated-HalfPlane*

را حل می‌کند. این درخت، هر یک از عملیات *MinAbove*، *Insert* و *Delete* را در زمان  $O(\log n)$  انجام می‌دهد. پیچیدگی حافظه‌ی این ساختمان داده، برابر با  $O(n)$  است و در زمان  $O(n \log n)$  ساخته می‌شود.

فرض کنید که  $C$  یک مخروط ثابت در  $\mathcal{C}_k$  و  $h_1$  و  $h_2$  دو خط باشند که به ترتیب خطوط مرزی بالا و پایین مخروط  $C$  را تشکیل می‌دهند. همچنین، بدون از دست دادن کلیت، فرض کنید که نیم‌خط  $\ell_C$  بر محور  $x$ ها منطبق باشد و  $D_1$  و  $D_2$  خطوطی باشند که هر دو از مبدأ می‌گذرند و

به ترتیب بر  $h_1$  و  $h_2$  عمود هستند. ما جهت  $D_1$  و جهت  $D_2$  را به صورت زیر تعریف می‌کنیم:  $D_1$  به سمت نیم‌صفحه‌ای اشاره می‌کند که شامل همه‌ی نقاط  $\mathbb{R}^2$  که پایین  $h_1$  قرار دارند، است و  $D_2$  به سمت نیم‌صفحه‌ای اشاره می‌کند که شامل همه‌ی نقاط  $\mathbb{R}^2$  که بالای  $h_2$  قرار دارند، است.

برای پیاده‌سازی کارای الگوریتم MERGESPANNERS، الگوریتمی به نام PLANESWEEP MERGING را پیشنهاد می‌کنیم (الگوریتم ۴-۲-۵ را ببینید) که برای هر نقطه‌ی  $p \in P_1$  و هر مخروط  $C \in \mathcal{C}_k$ ، همه‌ی یال‌های  $(p, r)$  را محاسبه می‌کند به طوری که  $r \in C_p \cap P_2$  و تصویر عمود  $r$  روی  $\ell_C$  به  $p$  نزدیک‌ترین باشد.

فرض کنید که  $G_1(P_1, E_1)$  و  $G_2(P_2, E_2)$  دو  $t$ -پوشش هندسی باشند به طوری که  $|P_1| \leq |P_2|$ . برای سادگی، فرض کنید که  $n_1 = |P_1|$  و  $n_2 = |P_2|$ ، حال، داریم:

**قضیه ۴-۲-۵.** زمان اجرای الگوریتم PLANESWEEP MERGING برابر  $O(n_1 \log n_1 + (n_1 + n_2) \log n_2)$  است.

**اثبات.** فرض کنید که  $\ell_i, k_i$  و  $|T_i|$  به ترتیب مقادیر  $\ell$ ، تعداد نقاطی از  $L_2$  (در الگوریتم ۴-۲-۵-۴، لیست  $L_2$  معرفی شده است) که در  $T$  درج می‌شوند و تعداد گره‌های درخت  $T$  در تکرار  $i$ ام الگوریتم باشند. به وضوح، خطوط ۵ و ۶ در الگوریتم، به ترتیب به  $O(n_1 \log n_1 + n_2 \log n_2)$  و  $O(1)$  زمان نیاز دارند. اگر ما از قبل، نقاط  $L_2$  را براساس ترتیب القایی توسط  $D_1$  مرتب کنیم، آنگاه با بکاربردن الگوریتم جستجوی دودویی و لم ۳-۲-۵، خطوط ۹، ۱۰ و ۱۱ در الگوریتم می‌توانند در زمان  $O(\log \ell_i) + \sum_{j=1}^{k_i} O(\log(|T_i| + j)) = O(\log \ell_i) + O((k_i + 1) \log(k_i + 1 + |T_i|))$  اجرا شوند. علاوه بر این، با استفاده از لم ۳-۲-۵، خط ۱۶ به  $O(\log(|T_i|))$  زمان نیاز دارد. حال، فرض کنید که  $Time(n_1, n_2)$  پیچیدگی زمانی الگوریتم PLANESWEEP MERGING باشد. از این رو، داریم

$$Time(n_1, n_2) = O(n_1 \log n_1 + n_2 \log n_2) + \sum_{i=1}^{n_1} O(\log \ell_i) + \sum_{i=1}^{n_1} O((k_i + 1) \log(k_i + 1 + |T_i|)) + \sum_{i=1}^{n_1} O(\log(|T_i|)).$$

از آنجایی که  $|T_i| \leq n_2$  و  $l_i \leq n_2$  بنابراین داریم:

$$\begin{aligned} \text{Time}(n_1, n_2) &= O(n_1 \log n_1) + O(n_2 \log n_2) + O(n_1 \log n_2) + \\ &O((n_1 + n_2) \log n_2) + O(n_1 \log n_2) \\ &= O(n_1 \log n_1 + (n_1 + n_2) \log n_2). \end{aligned}$$

■

روابط بالا، اثبات را کامل می‌کنند.

### ۳-۵ کاربردها

در این بخش، ما دو کاربرد از مسئله‌ی ادغام پوشش‌ها را بیان می‌کنیم. در ابتدا، یک الگوریتم تقسیم و غلبه برای ساخت یک  $t$ -پوشش برای یک مجموعه نقطه‌ی داده شده در صفحه ارائه می‌کنیم. سپس، یک الگوریتم تقسیم و غلبه برای ساخت یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا برای یک مجموعه نقطه‌ی داده شده در صفحه، ارائه می‌کنیم که در اینجا  $\mathcal{H}_k$  خانواده‌ی همه‌ی نیم‌صفحاتی است که برای هر نیم‌صفحه‌ی  $h \in \mathcal{H}_k$ ، یک مخروط  $C \in \mathcal{C}_k$  وجود دارد که مرز  $h$  موازی یکی از مرزهای  $C$  است.

### ۱-۳-۵ ساخت یک $t$ -پوشش

در این بخش، ما یک الگوریتم تقسیم و غلبه جهت ساخت یک  $t$ -پوشش برای یک مجموعه نقطه‌ی داده شده در صفحه ارائه می‌کنیم. این الگوریتم به صورت زیر است: فرض کنید که  $c > 0$  یک عدد طبیعی و  $S$  مجموعه‌ای از  $n$  نقطه در صفحه باشد. ابتدا  $S$  را به یک شکل دلخواه، به تعدادی زیرمجموعه که هر یک دارای حداکثر  $c$  عضو است، افراز می‌کنیم. سپس، برای هر زیرمجموعه، یک گراف کامل می‌سازیم. سپس، با بکاربردن الگوریتم PLANE SWEEP MERGING (الگوریتم ۲-۵-۴) را ببینید)، این گراف‌های کامل را به صورت تکراری ادغام می‌کنیم. روش ادغام این گراف‌ها مهم است. در ادامه، دو روش جهت ادغام این گراف‌ها معرفی می‌کنیم. ما این دو روش را ادغام متوالی

و ادغام یک در یک می‌نامیم. بسته به نوع ادغامی که استفاده می‌کنیم، زمان اجرا و اندازه‌ی گراف خروجی حاصل از ادغام، متفاوت است.

---

**Algorithm ۵-۳-3: PARTITION( $S, c$ )**

---

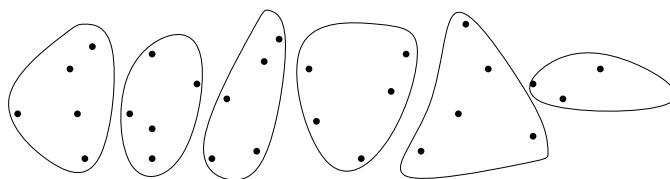
```

1 Sort the points of  $S$  increasingly according to their  $x$ -coordinate. Let  $L$  be the sorted list;
2 for  $h := 1$  to  $\lfloor \frac{n}{c} \rfloor$  do
3   for  $i = 1$  to  $c$  do
4     Create new subset  $S_h$ ;
5     Add  $L[(h - 1) \times c + i]$  into  $S_h$ ;
6   end
7 end
8 if  $n$  is not divisible by  $c$  then
9   Create a new subset  $S_h$  and add the remaining points of  $S$  into  $S_h$ ;
10 end
11 return all subsets  $S_h$ ;

```

---

بدون از دست دادن کلیت، فرض می‌کنیم که نقاط در حالت کلی هستند، بدین معنی که هیچ دو نقطه‌ای، مختصات  $x$  یکسان ندارند. افراز مجموعه نقطه‌ی  $S$  به صورت زیر انجام می‌شود. ابتدا، نقاط  $S$  را براساس مختصات  $x$  و به طور صعودی مرتب می‌کنیم. سپس، این نقاط مرتب شده را به دسته‌های  $c$  تایی تقسیم می‌کنیم (ممکن است که تعداد عناصر  $S$  مضربی از  $c$  نباشند در این صورت آخرین زیرمجموعه‌ای که ساخته می‌شود، کم‌تر از  $c$  عضو دارد) و به این ترتیب، نقاط  $S$  به زیرمجموعه‌هایی با حداکثر  $c$  عضو، افراز می‌شوند (الگوریتم ۵-۳-۳ را ببینید). شکل ۵-۲ یک مجموعه‌ی  $S$  را نشان می‌دهد که با پارامتر  $c = 5$  افراز شده است. توجه کنید که پیچیدگی زمانی الگوریتم PARTITION (الگوریتم ۵-۳-۳) برابر  $O(n \log n)$  است. حال، فرض کنید که  $S_1, S_2, \dots, S_f$  یک افراز برای  $S$  با روشی که در بالا توضیح داده شد، باشد که در اینجا  $f$  تعداد همه‌ی زیرمجموعه‌ها است ( $f \in O(n/c)$ ) و تعداد عناصر مجموعه‌ی  $S_i$  حداکثر  $c$  است. همچنین، فرض کنید که  $G_i$  گراف کامل روی  $S_i$  باشد. به وضوح،  $G_i$  یک  $t$ -پوشش برای  $S_i$  است و تعداد یال‌هایش برابر با  $O(c^2)$  می‌باشد و می‌تواند در زمان  $O(c^2)$  ساخته شود. در ادامه، درباره‌ی دو روش ادغام کردن، صحبت خواهیم کرد.



شکل ۵-۲: افراز با  $c = 5$ .

## ادغام متوالی

در این نوع ادغام، ابتدا دو گراف را ادغام می‌کنیم. سپس، در هر گام، گراف‌های دیگر را با گراف حاصل از مرحله قبل، ادغام می‌کنیم. به طور خاص، ادغام متوالی، به صورت زیر است: فرض کنید که  $G_i^*$  گراف حاصل از ادغام گراف‌های  $G_1, G_2, \dots, G_i$  با بکار بردن ادغام متوالی باشد. با استفاده از الگوریتم PLANE SWEEP MERGING (الگوریتم ۵-۲-۴)، گراف  $G_{i+1}$  را با  $G_i^*$  ادغام می‌کنیم. ما گراف حاصل از ادغام متوالی را  $G(S, E)$  می‌نامیم. طبق قضیه ۵-۲-۲، به وضوح، گراف  $G$  یک  $t$ -پوشش برای  $S$  است. در ادامه، تعداد یال‌های  $G$  و پیچیدگی زمانی ساخت  $G$  را که با  $T(n)$  نمایش می‌دهیم، محاسبه می‌کنیم. با استفاده از قضیه ۵-۲-۲ داریم:

$$|E| = \sum_{i=1}^f O(c^2) + \sum_{i=1}^{f-1} O(kc) = O((c+k)n), \quad (1-5)$$

و با استفاده از قضیه ۵-۲-۴، داریم

$$\begin{aligned} T(n) &= \sum_{i=1}^f O(c^2) + \sum_{i=1}^{f-1} O\left(c \log c + (i+1)c \log(ic)\right) \\ &= O(fc^2) + O(cf^2 \log f) = O(cn + c^{-1}n^2 \log n). \end{aligned}$$

حال، فرض کنید که  $t > 1$  یک عدد حقیقی ثابت باشد. اگر در معادله (۵-۱)،  $k$  یک عدد صحیح ثابت باشد، داریم:

**قضیه ۵-۳-۱.** با روش تقسیم و غلبه، می‌توان یک  $t$ -پوشش روی  $S$  ساخت. اگر از ادغام متوالی استفاده شود، تعداد یال‌های این  $t$ -پوشش برابر با  $O(cn)$  است و در زمان  $O(cn + c^{-1}n^2 \log n)$  ساخته می‌شود.

در قضیه ۵-۳-۱، اگر  $c = \lfloor \log n \rfloor$ ، آنگاه نتیجه‌ی زیر را داریم:

نتیجه ۵-۳-۲. با روش تقسیم و غلبه، می‌توان یک  $t$ -پوشش برای  $S$  با  $O(n \log n)$  یال و در زمان  $O(n^2)$  ساخت.

اگر در قضیه‌ی ۵-۳-۱،  $c$  یک عدد ثابت باشد، آنگاه داریم:

نتیجه ۵-۳-۳. با کمک روش تقسیم و غلبه، می‌توان یک  $t$ -پوشش برای  $S$  با  $O(n)$  یال و در زمان  $O(n^2 \log n)$  ساخت.

### ادغام یک در یک

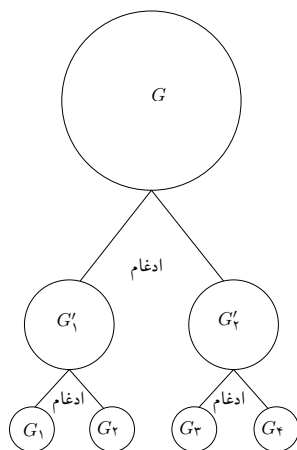
ادغام یک در یک که مشابه ادغام زیرلیست‌ها در مرتب‌سازی ادغامی است، به صورت زیر می‌باشد: با بکار بردن الگوریتم PLANE SWEEP MERGING، به طور تکراری  $G_i$ ها را دو به دو با هم ادغام می‌کنیم تا زیرگراف‌های جدید تولید شوند. این کار را روی گراف‌های جدید تکرار می‌کنیم تا زمانی که فقط یک زیرگراف باقی بماند (شکل ۵-۳ را ببینید). بر طبق قضیه‌ی ۵-۲-۲، گراف حاصل، یک  $t$ -پوشش برای  $S$  است. ما گراف حاصل را با  $G(S, E)$  نمایش می‌دهیم. بر طبق قضیه‌ی ۵-۲-۲، داریم:

$$\begin{aligned} |E| &= \sum_{i=1}^f O(c^2) + \sum_{i=1}^{\log f} O\left(\frac{f}{2^i} (k 2^{i-1} c)\right) \\ &= O(fc^2) + O(kfc \log f) = O(cn + kn \log n) \\ &= O(kn \log n), \end{aligned}$$

و بر طبق قضیه‌ی ۵-۲-۴، پیچیدگی زمانی ساخت گراف  $G$  که با  $T(n)$  نمایش داده می‌شود برابر است با

$$\begin{aligned} T(n) &= \sum_{i=1}^f O(c^2) + \sum_{i=1}^{\log f} O\left(\frac{3f}{2^i} 2^{i-1} c \log(2^{i-1} c)\right) \\ &= O(fc^2) + O\left(\frac{3fc}{2} \times \frac{\log f (\log f - 1)}{2}\right) \\ &= O(fc^2) + O(cf \log^2 f) = O(cn + n \log^2 n) \\ &= O(n \log^2 n). \end{aligned}$$





شکل ۵-۳: ادغام یک در یک.

حال، داریم :

**قضیه ۵-۳-۴.** به کمک روش تقسیم و غلبه، یک  $t$ -پوشش برای  $S$  وجود دارد که تعداد یال‌های آن برابر  $O(n \log n)$  و زمان ساخت آن  $O(n \log^2 n)$  است.

توجه کنید که این نتیجه، پیچیدگی زمانی ساخت  $G$  را در مقایسه با زمانی که از ادغام متوالی استفاده می‌شود، بهبود می‌دهد (نتیجه‌ی ۵-۳-۲ را ببینید).

### ۵-۳-۲ ساخت پوشش‌های تحمل‌پذیر ناحیه-خطا

فرض کنید که  $k \geq 2$  یک عدد صحیح،  $\theta = \frac{2\pi}{k}$  و  $\mathcal{H}_k$  خانواده‌ی همه‌ی نیم‌صفحاتی باشد به طوری که برای هر نیم‌صفحه‌ی  $h \in \mathcal{H}_k$  یک مخروط  $C \in \mathcal{C}_k$  وجود دارد که مرز  $h$  موازی یکی از مرزهای  $C$  است. در این بخش، مسئله‌ی ساخت یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا را بر روی یک مجموعه نقطه‌ی داده شده در صفحه، در نظر می‌گیریم. ابتدا، قضیه‌ی زیر را ثابت می‌کنیم.

در ادامه فرض کنید که  $t$  یک عدد حقیقی باشد به طوری که  $t \geq 1/(\cos \theta - \sin \theta)$  حال،

داریم:

**قضیه ۵-۳-۵.** اگر  $G_1(P_1, E_1)$  و  $G_2(P_2, E_2)$  دو  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا باشند. آنگاه، گراف  $MergeSpanners(G_1, G_2, t) \cup MergeSpanners(G_2, G_1, t)$  یک  $t$ -پوشش تحمل‌پذیر

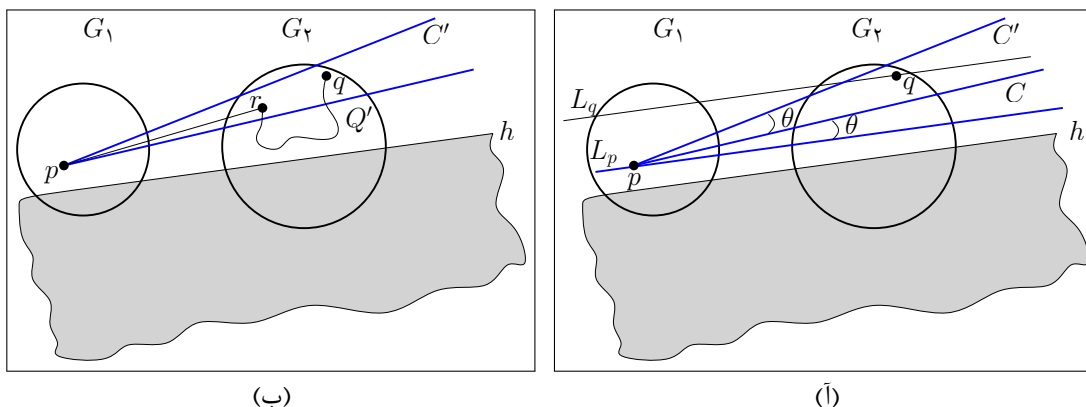
$\mathcal{H}_k$ -خطا برای  $P_1 \cup P_2$  است.

**اثبات.** فرض کنید که  $h$  یک نیم‌صفحه‌ی دلخواه در  $\mathcal{H}_k$  باشد. نشان می‌دهیم که برای هر  $p, q \in P_1 \cup P_2$  که خارج از  $h$  قرار دارند، یک  $t$ -مسیر از  $p$  به  $q$  در  $G \ominus h$  وجود دارد. اگر  $p, q \in P_1$  یا  $p, q \in P_2$ ، بنابراین از آنجایی که  $G_1$  و  $G_2$ ،  $t$ -پوشش‌های تحمل‌پذیر  $\mathcal{H}_k$ -خطا هستند، بنابراین، قضیه برقرار است. حال، بدون از دست دادن کلیت، فرض کنید که  $p \in P_1$  و  $q \in P_2$  و همچنین فرض کنید که  $L$  یکی از مرزهای یک مخروط به نام  $C$  در  $C_k$  باشد به طوری که موازی مرز  $h$  باشد. حال،  $L_p$  و  $L_q$  ( $L_q = L + q$  و  $L_p = L + p$ ) را در نظر بگیرید. از آنجایی که مرز  $h$  موازی  $L_p$  و  $L_q$  نیز است، می‌توانیم بدون از دست دادن کلیت، فرض کنیم که نقطه‌ی  $q$  بالای خط  $L_p$  است (شکل ۵-۴ را ببینید). این فرض، ایجاب می‌کند که یک مخروط  $C' \in C_k$  وجود داشته باشد به طوری که  $q \in C'_p$  و مرز  $h$ ، مخروط  $C'$  را قطع نکند. حال، فرض کنید که نقطه‌ی  $r$  نقطه‌ای است که الگوریتم  $\text{MERGESPANNERS}(G_1, G_2, t)$ ،  $(p, r)$  را هنگام پردازش نقطه‌ی  $p$  و مخروط  $C'_p$  به گراف اضافه می‌کند. اگر  $r = q$ ، بنابراین، به وضوح، قضیه برقرار است. حال، اگر  $r \neq q$ ، آنگاه از آنجایی که  $G_2$  یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا است، بنابراین یک  $t$ -مسیر  $Q'$  بین  $r$  و  $q$  در  $G_2 \ominus h$  وجود دارد (شکل ۵-۴ ب را ببینید). حال، مسیر  $Q := \{(p, r)\} \cup Q'$  در  $G \ominus h$  که  $p$  را به  $q$  متصل می‌کند را در نظر بگیرید. با استفاده از لم ۵-۲-۱، داریم:

$$\begin{aligned} |Q| &= |pr| + |Q'| \leq |pr| + t|rq| \\ &\leq |pr| + t(|pq| - (\cos \theta - \sin \theta)|pr|) \\ &= t|pq| - (t(\cos \theta - \sin \theta) - 1)|pr| \\ &\leq t|pq|. \end{aligned}$$

■ از این رو،  $Q$  یک  $t$ -مسیر بین  $p$  و  $q$  در  $G \ominus h$  است. حال، با کمک قضیه‌های ۵-۳-۴ و ۵-۳-۵ و از آنجایی که هر گراف کامل یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا است، داریم:

**نتیجه ۵-۳-۶.** به کمک روش تقسیم و غلبه، می‌توان یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا روی



شکل ۴-۵: اثبات قضیه‌ی ۵-۳-۵.

یک مجموعه‌ی داده شده از  $n$  نقطه در صفحه، ساخت. این  $t$ -پوشش، شامل  $O(n \log n)$  یال و زمان ساخت آن  $O(n \log^2 n)$  است.

#### ۴-۵ نتیجه‌گیری

در این فصل، مسئله‌ی ادغام دو پوشش هندسی را معرفی کردیم و سپس یک الگوریتم با پیچیدگی زمانی  $O(n_1 \log n_1 + (n_1 + n_2) \log n_2)$  برای حل آن ارائه کردیم که در اینجا  $n_1$  و  $n_2$  تعداد رئوس گراف‌های ورودی هستند. علاوه بر این، با بکار بردن این الگوریتم، دو الگوریتم تقسیم و غلبه جهت ساخت یک  $t$ -پوشش برای یک مجموعه نقطه‌ی داده شده در صفحه ارائه کردیم. همچنین، ما یک الگوریتم تقسیم و غلبه برای ساخت یک  $t$ -پوشش تحمل‌پذیر  $\mathcal{H}_k$ -خطا برای یک مجموعه نقطه‌ی داده شده در صفحه ارائه کردیم که تعداد یال‌های گراف خروجی برابر  $O(n \log n)$  و پیچیدگی زمانی آن برابر  $O(n \log^2 n)$  است. با توجه به این که نتایج مبتنی بر مفهوم  $\Theta$ -گراف است، به راحتی می‌توان نتایج را به ابعاد بالاتر تعمیم داد.



## فصل ۶

### پوشش‌های هندسی مقید به زاویه

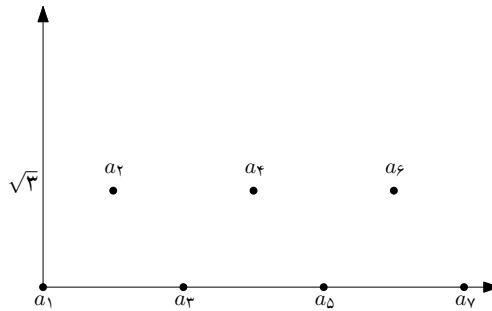
فرض کنید که  $\theta$  یک عدد حقیقی مثبت باشد. یک گراف هندسی  $G(S, E)$  را مقید به زاویه  $\theta$  گویند هرگاه زاویه بین هر دو یال مجزای  $\{p, q\}$  و  $\{p, r\}$  در  $E$  که با نماد  $\text{angle}(pq, pr)$  نمایش داده می‌شود، حداقل  $\theta$  باشد. در سال ۲۰۰۸، لی<sup>۱</sup> در کتابش [۴۰] بیان کرد که یک شبکه‌ی بی‌سیم، اگر یک شبکه‌ی پوششی هندسی مقید به زاویه باشد، شبکه‌ی خوبی است زیرا این خاصیت باعث کاهش تداخلات سیگنالی و همچنین در مواقعی که از آنتن‌های چرخشی استفاده می‌شود باعث کاهش میزان برق مصرفی می‌شود و همچنین وجود چنین خاصیتی در یک شبکه‌ی بی‌سیم، وجود مسیرهای کوتاه بین هر آنتن را تضمین می‌کند. برای اولین بار در سال ۲۰۱۲، کارمی و اسمید، مسئله‌ی ساخت شبکه‌ی پوششی هندسی مقید به زاویه را بررسی کردند [۲۱]. آن‌ها نشان دادند که برای هر مجموعه نقطه‌ی  $S$  از  $n$  نقطه در فضای اقلیدسی  $\mathbb{R}^d$  و برای هر زاویه  $\theta$  با  $0 < \theta < \pi/3$ ، یک  $t$ -پوشش مقید به زاویه  $\theta$  وجود دارد که در زمان  $O(n \log n)$  ساخته می‌شود که  $t$  فقط به  $\theta$  وابسته است. آن‌ها در همان مقاله، مسئله‌ی ساخت شبکه‌ی پوششی هندسی مقید به زاویه  $\theta$  با  $\theta \geq \pi/3$  را به عنوان مسئله‌ی حل‌نشده مطرح کردند. در این فصل، ما به حل این مسئله می‌پردازیم. نتایج این فصل در مقاله‌ی [۱۳] به چاپ رسیده است.

## ۲-۶ نتیجه‌ی اصلی

در این بخش، ما نشان می‌دهیم که یک مجموعه‌ی  $P$  شامل  $n$  نقطه در صفحه وجود دارد به طوری که برای هر  $\theta > \pi/3$ ، هیچ گراف هندسی همبند مقید به زاویه  $\theta$  با مجموعه‌ی رئوس  $P$  وجود ندارد. همچنین، نشان می‌دهیم که برای  $\theta = \pi/3$  و برای هر  $t < 2/\sqrt{3}$ ، هیچ  $t$ -پوشش مقید به زاویه  $\theta$  روی  $P$  وجود ندارد. نتایج این بخش به راحتی قابل تعمیم به ابعاد بالاتر است.

فرض کنید که  $P := \{a_1, a_2, \dots, a_n\}$  مجموعه‌ای از  $n$  نقطه در صفحه باشد به طوری که  $a_1 = (0, 0)$ ،  $a_{2i+1} = (2i, 0)$  و  $a_{2i} = (2i-1, \sqrt{3})$  باشند (شکل ۱-۶ را ببینید). حال، دو قضیه‌ی

<sup>۱</sup>Li

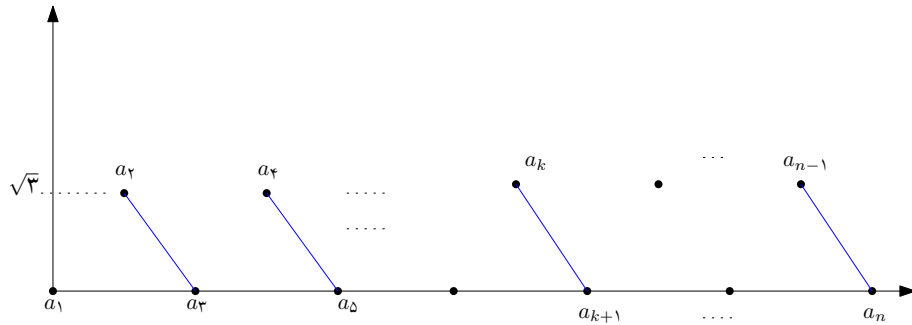


شکل ۶-۱: مجموعه‌ی  $P$  با  $n = 7$ .

زیر را اثبات می‌کنیم.

**قضیه ۶-۲-۱.** به ازای هر  $n > 2$  و هر  $\theta > \pi/3$ ، همه‌ی گراف‌های هندسی مقید به زاویه‌ی  $\theta$  با مجموعه رئوس  $P$ ، ناهمبند هستند.

**اثبات.** فرض کنید که  $\theta$  یک زاویه‌ی دلخواه بزرگ‌تر از  $\pi/3$  و  $G$  یک گراف هندسی مقید به زاویه‌ی  $\theta$  روی  $P$  باشد. ثابت می‌کنیم که  $G$  ناهمبند است. به عنوان فرض خلف، فرض کنید که  $G$  همبند باشد. جهت راحتی در اثبات، ما فرض می‌کنیم که  $n$  عددی فرد باشد. حالتی که  $n$  زوج است به صورت مشابه اثبات می‌شود. ما ادعا می‌کنیم که یال‌های  $(a_{2i}, a_{2i+1})$  به ازای هر  $1 \leq i \leq (n-1)/2$  متعلق به  $G$  هستند. دو رأس  $a_2$  و  $a_3$  را در نظر بگیرید. از آنجایی که فرض کرده‌ایم که  $G$  همبند است، بنابراین یک مسیر  $Q$  بین  $a_2$  و  $a_3$  در  $G$  وجود دارد. فرض کنید که طول یالی  $Q$  (تعداد یال روی مسیر  $Q$ ) بیش‌تر از ۱ باشد. حال، از روی مسیر  $Q$  و از رأس  $a_2$  به سمت رأس  $a_3$  حرکت کنید. از آنجایی که  $G$  مقید به زاویه‌ی  $\theta$  با شرط  $\theta > \pi/3$  است، بنابراین امکان ندارد که روی مسیر  $Q$  با یک یال که یک سر انتهایی آن یک رأس  $a_j$  و یک سر انتهایی دیگر آن یک رأس  $a_k$  با  $k < j$  باشد، مواجه شویم، زیرا در غیر این صورت، دو یال همجوار در  $G$  وجود دارد به طوری که زاویه‌ی بین آن‌ها حداکثر  $\pi/3$  است. از این رو، هیچ مسیری با طول یالی حداقل ۲ بین  $a_2$  و  $a_3$  در  $G$  وجود ندارد. بنابراین، طول یالی  $Q$  یک است که بدین معناست که  $(a_2, a_3)$  یالی در  $G$  است. به طور مشابه، می‌توان نشان داد که  $G$  شامل یال‌های  $(a_{2i}, a_{2i+1})$  با  $i = 2, 3, \dots, (n-1)/2$  است (شکل ۶-۲ را ببینید). حال، دو رأس  $a_1$  و  $a_2$  را در نظر بگیرید. با توجه به توضیحات بالا و از آنجایی که  $G$  مقید به زاویه‌ی  $\theta$  با  $\theta > \pi/3$  است، نتیجه می‌گیریم که



شکل ۲-۶: اثبات قضیه ۱-۲-۶

$G$  نمی‌تواند شامل یال  $(a_1, a_2)$  باشد. از طرفی دیگر، هیچ مسیری با طول یالی حداقل ۲ در  $G$  بین  $a_1$  و  $a_2$  نمی‌تواند وجود داشته باشد زیرا در غیر این صورت،  $G$  شامل یک یال با یک سر انتهایی  $a_2$  خواهد بود. چون  $G$  شامل  $(a_2, a_3)$  است، بنابراین  $G$  شامل دو یال همجوار است که زاویه‌ی بین آن‌ها حداکثر  $\pi/3$  است که این، با فرض اولیه‌ی ما تناقض دارد. بنابراین، هیچ مسیری بین  $a_1$  و  $a_2$  در  $G$  وجود ندارد که این، با همبند بودن  $G$  در تناقض است. ■

**قضیه ۲-۲-۶.** برای  $\theta = \pi/3$  و هر  $t > 2/\sqrt{3}$ ، هیچ  $t$ -پوشش مقید به زاویه‌ی  $\theta$  روی  $P$  وجود ندارد.

**اثبات.** ما قضیه را برای  $n$ ‌های فرد اثبات می‌کنیم. حالتی که  $n$  زوج است به صورت مشابه اثبات می‌شود. فرض کنید که  $G_1(P, E_1)$  یک گراف هندسی روی  $P$  باشد به طوری که

$$E = \bigcup_{i,j=1,2,\dots,\frac{n-1}{2}, i < j} \{(a_{2i-1}, a_{2j-1}), (a_{2i}, a_{2j})\} \cup \bigcup_{i=1}^{\frac{n-1}{2}} \{(a_{2i-1}, a_{2i}), (a_{2i}, a_{2i+1})\}$$

(شکل ۳-۶ را ببینید). به راحتی می‌توان ثابت کرد که ضریب کشش بین رئوس  $a_1$  و  $a_{2i}$  در گراف  $G_1$  برابر است با

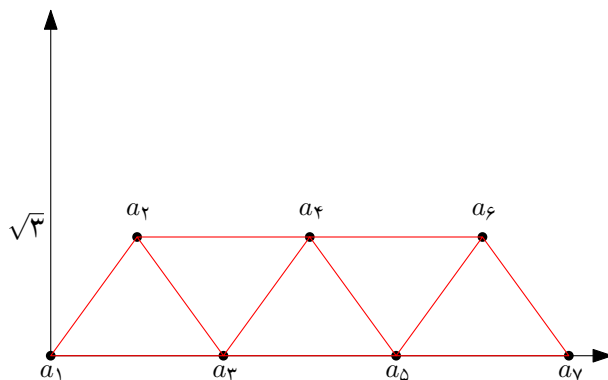
$$\frac{2i}{\sqrt{(2i-1)^2 + 3}}$$

از آنجایی که بیش‌ترین مقدار این ضریب کشش نسبت به متغیر  $i$  برابر است با  $2/\sqrt{3}$ ، بنابراین به راحتی ثابت می‌شود که ضریب کشش  $G_1$  برابر  $2/\sqrt{3}$  است.

حال، فرض کنید که  $G$  یک گراف هندسی همبند مقید به زاویه‌ی  $\theta = \pi/3$  روی  $P$  باشد.

اگر  $G$  شامل یک یال  $(a_{2i}, a_{2j-1})$  با  $i \neq j$  باشد، آنگاه به راحتی می‌توان پی برد که  $G$  باید شامل





شکل ۳-۶: گراف  $G_1$  با ۷ رأس

دو یال  $(x, y)$  و  $(y, z)$  باشد به طوری که  $|xy| = |yz| = |zx| = 2$  و مسیر  $(x, y) \cup (y, z)$  کوتاه‌ترین مسیر بین  $x$  و  $z$  در  $G$  باشد. بنابراین، ضریب کشش بین  $x$  و  $z$  در  $G$  برابر ۲ است. از این رو، ضریب کشش  $G$  در این حالت حداقل ۲ است.

حال، اگر  $G$  شامل هیچ یالی به صورت  $(a_{2i}, a_{2j-1})$  با شرط  $i \neq j$  نباشد، آنگاه به وضوح،  $G$  زیرگرافی از  $G_1$  است. در این صورت، از آنجایی که ضریب کشش  $G_1$  دقیقاً  $2/\sqrt{3}$  است، ضریب کشش  $G$  حداقل  $2/\sqrt{3}$  می‌باشد.

بنابر آنچه در بالا ذکر شد، ما به سادگی نتیجه می‌گیریم که برای هر  $t < 2/\sqrt{3}$ ، هیچ  $t$ -پوشش مقید به زاویه  $\theta = \pi/3$  روی  $P$  وجود ندارد. ■

### ۳-۶ نتیجه‌گیری

ما در اینجا مسئله‌ی حل‌نشده‌ای که توسط کارمی و اسمید [۲۱] معرفی شد را پاسخ دادیم. ما نشان دادیم که یک مجموعه‌ی  $P$  شامل  $n$  نقطه در صفحه وجود دارد به طوری که برای هر  $\theta > \pi/3$ ، هیچ گراف هندسی همبند مقید به زاویه  $\theta$  با مجموعه رئوس  $P$  وجود ندارد. همچنین، نشان دادیم که برای  $\theta = \pi/3$  و برای هر  $t < 2/\sqrt{3}$ ، هیچ  $t$ -پوشش مقید به زاویه  $\theta$  روی  $P$  وجود ندارد.

ما این فصل را با سه مسئله‌ی حل‌نشده که در ادامه آمده است، به پایان می‌رسانیم:

فرض کنید که  $S$  مجموعه‌ای از  $n$  نقطه در صفحه و  $t > 1$  یک عدد حقیقی باشد.

۱. فرض کنید  $\theta > 0$  یک عدد حقیقی دلخواه باشد. آیا یک  $t$ -پوشش مقید به زاویه  $\theta$  روی  $S$  وجود دارد؟ مسئله‌ی بالا یک مسئله‌ی تصمیم‌گیری است. ما حدس می‌زنیم که این مسئله، یک مسئله‌ی  $NP$ -سخت باشد.
۲. آیا یک الگوریتم با زمان چندجمله‌ای برای ساخت یک  $t$ -پوشش روی  $S$  با اندازه‌ی خطی که کم‌ترین زاویه‌ی بین یال‌های مجاور، بیش‌ترین باشد، وجود دارد؟
۳. فرض کنید که نقاط  $S$  به صورت یکنواخت در صفحه توزیع شده باشند. آیا یک الگوریتم با زمان چندجمله‌ای برای ساخت یک  $t$ -پوشش روی  $S$  وجود دارد به طوری که اندازه‌ی آن خطی باشد و امید ریاضی زاویه‌ی بین هر دو یال مجاورش حداقل  $\theta$  باشد؟
۴. آیا گراف‌های یائو،  $\Theta$ -گراف و یائوی پیوسته، به ازای مقادیری از  $\theta$ ، پوشش‌های مقید به زاویه‌ی  $\theta$  هستند؟

نتیجه گیری

در این بخش، ما به نتیجه‌گیری رساله می‌پردازیم و سپس تمامی مسائل حل‌نشده‌ای که در این رساله بیان شدند را مجدداً بیان خواهیم کرد.

در این رساله، چند مسئله‌ی مرتبط با طراحی و تحلیل شبکه‌های هندسی را در نظر گرفتیم: در فصل ۲، نوع جدیدی از گراف‌های هندسی به نام گراف یائوی پیوسته معرفی شد. نشان دادیم که برای هر  $0 < \theta \leq 2\pi/3$ ، گراف یائوی پیوسته‌ی  $cY(\theta)$  یک  $t$ -پوشش است که  $t$  یک مقدار ثابت می‌باشد، ولی برای  $2\pi/3 \leq \theta \leq \pi$  این‌گونه نیست. همچنین نشان دادیم که برای هر  $0 < \theta \leq \pi$ ، گراف  $cY(\theta)$  همبند است ولی برای هر  $\theta > \pi$  لزوماً همبند نیست. همچنین این خصوصیات را در مدل تحمل‌پذیری ناحیه-خط نیز بررسی کردیم و نشان دادیم که برای هر  $\theta < \pi/3$ ، گراف  $cY(\theta)$  یک  $t$ -پوشش باقی می‌ماند ( $t$  مقدار ثابت و وابسته به  $\theta$  است) حتی اگر ناحیه محدبی از آن حذف شود. همچنین نشان دادیم که  $cY(\theta)$  برای هر  $\theta \leq \pi$  همبند باقی می‌ماند حتی اگر بخش محدبی از آن حذف شود.

در فصل ۳، یک الگوریتم برای ساخت پوشش شکاف-حریصانه با کمک WSPD ارائه کردیم که پیچیدگی زمانی و حافظه‌ی آن به ترتیب  $O(n^3)$  و  $O(n)$  است. در این فصل، نشان دادیم که یک الگوریتمی برای ساخت پوشش شکاف-حریصانه در زمان  $O(n^2)$  و با بکاربردن  $O(n^2)$  حافظه وجود دارد. همچنین، ضریب کشش پوشش شکاف-حریصانه را بهبود دادیم. در این فصل، همچنین ما سه الگوریتم با زمان اجرای  $O(n^3)$  برای ساخت پوشش شکاف-حریصانه ارائه کردیم و تعدادی نتایج تجربی روی زمان اجرای این الگوریتم‌ها ارائه دادیم. با توجه به نتایج تجربی، نتیجه می‌گیریم که یکی از این الگوریتم‌ها (الگوریتم ۳-۴-۲) نسبت به الگوریتم‌های دیگری که معرفی می‌شوند، سریع‌تر است. همچنین به صورت تجربی خصوصیات هندسی پوشش حریصانه را در مقایسه با پوشش شکاف-حریصانه، مورد بررسی قرار دادیم.

در فصل ۴، مفهوم جدیدی به نام گراف‌های خود-گرای ضعیف که حالت تضعیف شده‌ی گراف‌های خود-گرا است را معرفی کردیم. یک الگوریتم ارائه کردیم که در زمان چندجمله‌ای تشخیص می‌دهد که آیا یک گراف هندسی داده شده، خود-گرای ضعیف است یا خیر. همچنین، مسئله‌ی ساخت یک گراف خود-گرای ضعیف مسطح برای هر مجموعه نقطه‌ای در صفحه را بررسی

کردیم. همچنین، مسئله‌ی ساخت  $t$ -پوشش خود-گرای ضعیف (یک  $t$ -پوشش که بین هر دو رأس آن یک  $t$ -مسیر خود-گرای ضعیف وجود دارد) برای هر مجموعه نقطه در صفحه را در نظر گرفتیم. به طور خاص، نشان دادیم که برای هر عدد حقیقی  $t > 1$  و هر مجموعه نقطه‌ی  $S$  که روی مرز یک مثلث قرار گرفته‌اند، یک  $t$ -پوشش وتر-افزایشی با  $O(kn)$  یال که  $k$  فقط وابسته به  $t$  است، وجود دارد. همچنین در این فصل، مسئله‌ی حل‌نشده‌ای که توسط دهکردی و همکارانش [۴۹] مطرح شد را مورد بحث قرار دادیم. به طور خاص، نشان دادیم که یک گراف وتر-افزایشی مسطح برای هر مجموعه نقطه‌ای که روی مرز یک مثلث منفرجه یا قائم‌الزاویه قرار گرفته‌اند، وجود دارد. همچنین، نشان دادیم که برای هر مجموعه نقطه‌ای که روی مرز یک مثلث حاده قرار دارد، یک گراف وتر-افزایشی مسطح با اضافه کردن حداکثر سه نقطه‌ی اشتاینری وجود دارد.

در فصل ۵، مسئله‌ی ادغام دو پوشش هندسی را معرفی کردیم و سپس یک الگوریتم برای حل آن ارائه کردیم. تا جایی که اطلاع داریم، هنوز هیچ الگوریتم تقسیم و غلبه‌ای برای ساخت یک پوشش هندسی ارائه نشده است. بنابراین، ما برای اولین بار، با استفاده از مفهوم ادغام دو پوشش هندسی، دو الگوریتم تقسیم و غلبه جهت ساخت یک  $t$ -پوشش برای یک مجموعه نقطه‌ی داده شده در صفحه ارائه کردیم. همچنین، یک الگوریتم تقسیم و غلبه برای ساخت یک  $t$ -پوشش تحمل‌پذیر ناحیه-خطا برای یک مجموعه از نقاط داده شده و برای یک‌سری از ناحیه‌های خاص در صفحه ارائه کردیم.

در فصل ۶، ما مسئله‌ی حل‌نشده‌ای که توسط کارمی و اسمید [۲۱] معرفی شد را پاسخ دادیم. نشان دادیم که یک مجموعه نقطه‌ی  $P$  شامل  $n$  نقطه در صفحه وجود دارد به طوری که برای هر  $\theta > \pi/3$ ، هیچ گراف هندسی همبند مقید به زاویه‌ی  $\theta$  با مجموعه رئوس  $P$  وجود ندارد. همچنین، نشان دادیم که برای  $\theta = \pi/3$  و برای هر  $t < 2/\sqrt{3}$ ، هیچ  $t$ -پوشش مقید به زاویه‌ی  $\theta$  روی  $P$  وجود ندارد.

مسائل حل‌نشده‌ای که در این رساله، بیان شدند، به صورت زیر هستند:

۱. آیا  $cY(\theta)$  برای  $2\pi/3 < \theta < \pi$  یک  $t$ -پوشش که  $t$  مقدار ثابتی باشد، است؟

۲. آیا  $cY(\theta)$  برای  $\pi/3 \leq \theta \leq \pi$  یک  $t$ -پوشش تحمل‌پذیر  $C$ -خطا که  $t$  یک مقدار ثابتی باشد،

است؟

۳. آیا به ازای مقادیر خاصی از  $\theta$  و  $w$ ، پوشش حریصانه می‌تواند زیرگرافی از پوشش شکاف-حریصانه باشد؟

۴. آیا برای هر مجموعه‌ی  $S$  از نقاط در صفحه گراف خود-گرای مسطحی وجود دارد که رئوس آن  $S$  باشد

۵. آیا الگوریتمی وجود دارد که در زمان  $o(n^3)$  یک گراف خود-گرای ضعیف مسطح برای هر مجموعه نقطه در صفحه بسازد؟

۶. آیا برای هر مجموعه‌ی  $S$  از نقاط در صفحه یک  $t$ -پوشش خود-گرا (ضعیف) با تعداد یال کم وجود دارد که رئوس آن  $S$  باشد؟

۷. آیا یک الگوریتم وجود دارد که در زمان خطی تشخیص دهد که آیا یک درخت هندسی داده شده، خود-گرای ضعیف است یا خیر؟

۸. فرض کنید  $\theta > 0$  یک عدد حقیقی دلخواه باشد. آیا یک  $t$ -پوشش مقید به زاویه‌ی  $\theta$  روی  $S$  وجود دارد؟ مسئله‌ی بالا یک مسئله‌ی تصمیم‌گیری است. ما حدس می‌زنیم که این مسئله، یک مسئله‌ی  $NP$ -سخت باشد.

۹. آیا یک الگوریتم با زمان چندجمله‌ای برای ساخت یک  $t$ -پوشش روی  $S$  با اندازه‌ی خطی که کم‌ترین زاویه‌ی بین یال‌های مجاور، بیش‌ترین باشد، وجود دارد؟

۱۰. فرض کنید که نقاط  $S$  به صورت یکنواخت در صفحه توزیع شده باشند. آیا یک الگوریتم با زمان چندجمله‌ای برای ساخت یک  $t$ -پوشش روی  $S$  وجود دارد به طوری که اندازه‌ی آن خطی باشد و امید ریاضی زاویه‌ی بین هر دو یال مجاورش حداقل  $\theta$  باشد؟ هندسی داده شده، خود-گرای ضعیف است یا خیر؟

۱۱. آیا گراف‌های یائو،  $\Theta$ -گراف و یائوی پیوسته، به ازای مقادیری از  $\theta$ ، پوشش‌های مقید به زاویه‌ی  $\theta$  هستند؟

# واژه‌نامه فارسی به انگلیسی

ا

Ubuntu . . . . . ابونتو  
broadcasting . . . . . انتشار  
Size . . . . . اندازه

پ

convex hull . . . . . پوشش محدب  
region-fault tolerant spanners . . . . . پوشش‌های تحمل‌پذیر ناحیه-خطا

ت

Thales . . . . . تالس  
well-separated pair decomposition . . . . . تجزیه‌ی زوجی خوش-مجزا  
semi-separated pair decomposition . . . . . تجزیه‌ی زوجی نیم-مجزا  
Ball . . . . . توپ

## ج

plane sweep ..... جاروی صفحه  
metric space searching ..... جستجوی فضای متریک  
bounding box ..... جعبه‌ی احاطه کننده

## ح

convex position ..... حالت محدب

## خ

outerplanar ..... خارج مسطح  
gap property ..... خاصیت شکاف  
self-approaching ..... خود-گرا

## د

Dijkstra ..... دایکسترا  
Degree ..... درجه  
minimum spanning tree ..... درخت پوشای کمینه  
aperture ..... دهانه‌ی



## ر

rank of distance ..... رتبه‌ی فاصله  
graph drawing ..... رسم گراف

## ش

gap ..... شکاف  
gap-greedy ..... شکاف-حریصانه

## ض

separation ratio ..... ضریب تفکیک

## ق

circular sector ..... قطاع مدور

## گ

proximity graph ..... گراف مجاورت  
Yao graph ..... گراف یائو  
greedy graphs ..... گراف‌های حریصانه

## م

lune	ماه
metric	متریک
cone	مخروط
proximity problems	مسائل مجاورتی
path-greedy	مسیر-حریصانه
angle-constrained	مقید به زاویه

## ن

convex regions	ناحیه‌های محدب
----------------	----------------

## و

increasing-chord	وتر-افزایشی
weight	وزن

## T

$t$ -spanner	$t$ -پوشش
$t$ -approximate	$t$ -تقریب
$t$ -path	$t$ -مسیر

W

$w$ -gap .....  $w$ -شکاف



# واژه‌نامه انگلیسی به فارسی

## A

angle-constrained ..... مقید به زاویه  
aperture ..... دهانه‌ی

## B

Ball ..... توپ  
bounding box ..... جعبه‌ی احاطه کننده  
broadcasting ..... انتشار

## C

circular sector ..... قطاع مدور  
cone ..... مخروط  
convex hull ..... پوشش محدب  
convex position ..... حالت محدب  
convex regions ..... ناحیه‌های محدب

## D

Degree ..... درجه  
Dijkstra ..... دایکسترا

## G

gap ..... شکاف  
gap property ..... خاصیت شکاف  
gap-greedy ..... شکاف-حریصانه  
graph drawing ..... رسم گراف  
greedy graphs ..... گراف‌های حریصانه

## I

increasing-chord ..... وتر-افزایشی

## L

lune ..... ماه

## M

metric ..... متریک  
metric space searching ..... جستجوی فضای متریک  
minimum spanning tree ..... درخت پوشای کمینه

## O

outerplanar ..... خارج مسطح

## P

path-greedy ..... مسیر-حریصانه  
plane sweep ..... جاروی صفحه  
proximity graph ..... گراف مجاورت  
proximity problems ..... مسائل مجاورتی

## R

rank of distance ..... رتبه‌ی فاصله  
region-fault tolerant spanners ..... پوشش‌های تحمل‌پذیر ناحیه-خطا

## S

self-approaching	خود-گرا
semi-separated pair decomposition	تجزیه‌ی زوجی نیم-مجزا
separation ratio	ضریب تفکیک
Size	اندازه

## T

$t$ -approximate	$t$ -تقریب
Thales	تالس
$t$ -path	$t$ -مسیر
$t$ -spanner	$t$ -پوشش

## U

Ubuntu	ابونتو
--------	--------

## W

weight	وزن
well-separated pair decomposition	تجزیه‌ی زوجی خوش-مجزا
$w$ -gap	$w$ -شکاف



## Y

Yao graph ..... گراف یائو



# منابع و مآخذ

- [1] A. Abam, M., Farshi, M., and Gudmundsson, J. Personal communication.
- [2] Abam, M. A., de Berg, M., Farshi, M., and Gudmundsson, J. Region-fault tolerant geometric spanners. *Discrete & Computational Geometry* 41, 4 (2009), 556–582.
- [3] Alamdari, S., Chan, T. M., Grant, E., Lubiw, A., and Pathak, V. Self-approaching graphs. in *International Symposium on Graph Drawing* (2012), Springer, pp. 260–271.
- [4] Alamdari, S., M. Chan, T., Grant, E., Lubiw, A., and Pathak, V. Self-approaching graphs. in *International Symposium on Graph Drawing* (2013), Springer, pp. 260–271.
- [5] Alewijnse, S. P. A., Bouts, Q. W., ten Brink, A. P., and Buchin, K. Computing the greedy spanner in linear space. *Algorithmica* 73, 3 (2015), 589–606.
- [6] Alewijnse, S. P. A., Bouts, Q. W., ten Brink, A. P., and Buchin, K. Distribution-sensitive construction of the greedy spanner. *Algorithmica* 78, 1 (2017), 209–231.
- [7] Althöfer, I., Das, G., Dobkin, D., Joseph, D., and Soares, J. On sparse spanners of weighted graphs. *Discrete & Computational Geometry* 9, 1 (1993), 81–100.
- [8] Angelini, P., Frati, F., and Grilli, L. An algorithm to construct greedy drawings of triangulations. *Journal of Graph Algorithms and Applications* 14, 1 (2010), 19–51.
- [9] Arya, S., and Smid, M. Efficient construction of a bounded-degree spanner with low weight. *Algorithmica* 17, 1 (1997), 33–54.
- [10] B. Callahan, P., and R. Kosaraju, S. A decomposition of multidimensional point sets with applications to  $k$ -nearest-neighbors and  $n$ -body potential fields. *Journal of the ACM (JACM)* 42, 1 (1995), 67–90.

- [11] Bakhshesh, D., Barba, L., Bose, P., De Carufel, J.-L., Damian, M., Fagerberg, R., Farshi, M., van Renssen, A., Taslakian, P., and Verdonshot, S. Continuous Yao graphs. *Computational Geometry: Theory and Applications* (to appear).
- [12] Bakhshesh, D., and Farshi, M. (weakly) self-approaching geometric graphs and spanners. *Submitted to Computational Geometry: Theory and Applications*.
- [13] Bakhshesh, D., and Farshi, M. Angle-constrained spanners with angle at least  $\pi/3$ . *Information Processing Letters* 120 (2017), 44 – 46.
- [14] Bakhshesh, D., and Farshi, M. An algorithm of merging of geometric spanners networks and its applications. *Submitted to Journal of New Researches in Mathematics* (In Persian).
- [15] Bakhshesh, D., and Farshi, M. Improving space and time complexity of the gap-greedy spanner algorithm. *CSI Journal on Computer Science and Engineering* (to appear).
- [16] Barba, L., Bose, P., Damian, M., Fagerberg, R., Keng, W. L., O'Rourke, J., van Renssen, A., Taslakian, P., Verdonshot, S., and Xia, G. New and improved spanning ratios for Yao graphs. in *Proceedings of the 30th Symposium on Computational Geometry* (2014), SoCG'14.
- [17] Bose, P., Carmi, P., Farshi, M., Maheshwari, A., and Smid, M. Computing the greedy spanner in near-quadratic time. *Algorithmica* 58, 3 (2010), 711–729.
- [18] Bose, P., Damian, M., Douïeb, K., O'Rourke, J., Seamone, B., Smid, M., and Wührer, S.  $\pi/2$ -angle Yao graphs are spanners. *International Journal of Computational Geometry & Applications* 22, 1 (2012), 61–82.
- [19] Bose, P., Fagerberg, R., van Renssen, A., and Verdonshot, S. Competitive routing in the half- $\theta_6$ -graph. in *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms* (2012), SIAM, pp. 1319–1328.
- [20] Bose, P., Maheshwari, A., Narasimhan, G., Smid, M., and Zeh, N. Approximating geometric bottleneck shortest paths. *Computational Geometry: Theory and Applications* 29, 3 (2004), 233–249.
- [21] Carmi, P., and Smid, M. An optimal algorithm for computing angle-constrained spanners. *Journal of Computational Geometry* 3, 1 (2012), 196–221.

- [22] Chandra, B., Das, G., Narasimhan, G., and Soares, J. New sparseness results on graph spanners. *International Journal of Computational Geometry & Applications* 05 (1995), 125–144.
- [23] Chew, P. There is a planar graph almost as good as the complete graph. in *Proceedings of the second SOCG* (1986), ACM, pp. 169–177.
- [24] Clarkson, K. L. Approximation algorithms for shortest path motion planning. in *Proceedings of the 19th ACM Symposium on the Theory of Computing (STOC 1987)* (1987), pp. 56–65.
- [25] Czumaj, A., and Zhao, H. Fault-tolerant geometric spanners. *Discrete & Computational Geometry* 32, 2 (2004), 207–230.
- [26] Damian, M., and Raudonis, K. Yao graphs span theta graphs. *Discrete Mathematics, Algorithms and Applications* 4, 02 (2012), 1250024.
- [27] Das, G., and Narasimhan, G. A fast algorithm for constructing sparse euclidean spanners. *International Journal of Computational Geometry & Applications* 07, 04 (1997), 297–315.
- [28] Dehkordi, H. R., Frati, F., and Gudmundsson, J. Increasing-chord graphs on point sets. *Journal of Graph Algorithms and Applications* 19, 2 (2015), 761–778.
- [29] El Molla, N. M. Yao spanners for wireless ad hoc networks. Master’s thesis, Villanova University, 2009.
- [30] Farley, A. M., Proskurowski, A., Zappala, D., and Windisch, K. Spanners and message distribution in networks. *Discrete Applied Mathematics* 137, 2 (2004), 159 – 171.
- [31] Farshi, M., and Gudmundsson, J. Experimental study of geometric  $t$ -spanners: A running time comparison. in *Experimental Algorithms: 6th International Workshop, WEA 2007, Rome, Italy, June 6-8, 2007. Proceedings* (Berlin, Heidelberg, 2007), C. Demetrescu, ed., Springer Berlin Heidelberg, pp. 270–284.
- [32] Farshi, M., and Poureidi, A. A lower bound for computing geometric spanners. *Computational Geometry: Theory and Applications* 53 (2016), 21 – 26.

- [33] Flinchbaugh, B. E., and Jones, L. K. Strong connectivity in directional nearest-neighbor graphs. *SIAM Journal on Algebraic and Discrete Methods* 2, 4 (1981), 461–463.
- [34] H. Papadimitriou, C., and Ratajczak, D. On a conjecture related to geometric routing. *Theoretical Computer Science* 344, 1 (2005), 3–14.
- [35] He, X., and Zhang, H. On succinct convex greedy drawing of 3-connected plane graphs. in *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms* (2011), SIAM, pp. 1477–1486.
- [36] Icking, C., Klein, R., and Langetepe, E. Self-approaching curves. in *Mathematical Proceedings of the Cambridge Philosophical Society* (1999), volume 125, Cambridge University Press, pp. 441–453.
- [37] Leighton, T., and Moitra, A. Some results on greedy embeddings in metric spaces. *Discrete & Computational Geometry* 44, 3 (2010), 686–705.
- [38] Lenhart, W., and Liotta, G. Proximity drawings of outerplanar graphs. in *Symposium on Graph Drawing, GD '96 Berkeley, California, USA, September 18–20, 1996 Proceedings* (1997), Springer, pp. 286–302.
- [39] Levkopoulos, C., Narasimhan, G., and Smid, M. Improved algorithms for constructing fault-tolerant spanners. *Algorithmica* 32, 1 (2002), 144–156.
- [40] Li, X.-Y. *Wireless Ad Hoc and Sensor Networks*. Cambridge University Press, Cambridge, UK, 2008.
- [41] Liotta, G., and Di Battista, G. Computing proximity drawings of trees in the 3-dimensional space. in *Algorithms and Data Structures*. Springer, 1995, pp. 239–250.
- [42] Lukovszki, T. *New results on geometric spanners and their applications*. Ph.D. thesis, Heinz Nixdorf Institute and Department of Mathematics and Computer Science, Paderborn University, Paderborn, Germany, 1999.
- [43] Mastakas, K., and Symvonis, A. On the construction of increasing-chord graphs on convex point sets. in *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on* (2015), IEEE, pp. 1–6.

- [44] Narasimhan, G., and Smid, M. *Geometric spanner networks*. Cambridge University Press, 2007.
- [45] Navarro, G., and Paredes, R. Practical construction of metric  $t$ -spanners. in *Algorithm Engineering and Experiments* (2003), volume 3, pp. 69–81.
- [46] Navarro, G., Paredes, R., and Chávez, E.  $t$ -spanners as a data structure for metric space searching. in *International Symposium on String Processing and Information Retrieval* (2002), Springer Berlin Heidelberg, pp. 298–309.
- [47] Navarro, G., Paredes, R., and Chávez, E.  $t$ -spanners for metric space searching. *Data & Knowledge Engineering* 63, 3 (2007), 820–854.
- [48] Peleg, D., and Schäffer, A. A. Graph spanners. *Journal of graph theory* 13, 1 (1989), 99–116.
- [49] R. Dehkordi, H., Frati, F., and Gudmundsson, J. Increasing-chord graphs on point sets. in *Graph Drawing* (2014), Springer, pp. 464–475.
- [50] Rao, S. B., and Smith, W. D. Approximating geometrical graphs via “spanners” and “banyans”. in *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (1998), ACM, pp. 540–550.
- [51] Rote, G. Curves with increasing chords. in *Mathematical Proceedings of the Cambridge Philosophical Society* (1994), volume 115, Cambridge University Press, pp. 1–12.
- [52] Russel, D., and Guibas, L. J. Exploring protein folding trajectories using geometric spanners. in *Pacific Symposium on Biocomputing* (2005), World Scientific, pp. 42–53.
- [53] Soares, J. Approximating euclidean distances by small degree graphs. *Discrete & Computational Geometry* 11, 1 (1994), 213–233.
- [54] T. Goodrich, M., and Strash, D. Succinct greedy geometric routing in the Euclidean plane. in *Algorithms and Computation: 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*. Springer, 2009, pp. 781–791.
- [55] W. Matula, D., and R. Sokal, R. Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical analysis* 12, 3 (1980), 205–222.

- [56] Yao, A. C. C. On constructing minimum spanning trees in  $k$ -dimensional spaces and related problems. *SIAM Journal on Computing* 11, 4 (1982), 721–736.



# نمایه

درجه، ۴	۱
درخت پوشای کمینه، ۴	ابونتو، ۶۱
دهانه‌ی، ۱۱	انتشار، ۴
	اندازه، ۴
ر	پ
رتبه‌ی فاصله، ۲۳	پوشش محدب، ۲۷
رسم گراف، ۶۸	پوشش‌های تحمل‌پذیر ناحیه-خطا، ۱۳
ش	ت
شکاف، ۷	تالس، ۳۹
شکاف-حریصانه، ۷	تجزیه‌ی زوجی خوش-مجزا، ۷
ض	تجزیه‌ی زوجی نیم-مجزا، ۱۰۵
ضریب تفکیک، ۱۰	توپ، ۱۰
ق	ج
قطاع مدور، ۲۶	جاروی صفحه، ۱۰۷
گ	جستجوی فضای متریک، ۴
گراف مجاورت، ۶۸	جعبه‌ی احاطه‌کننده، ۹
گراف یائو، ۱۱	ح
گراف‌های حریصانه، ۶۸	حالت محدب، ۶۹
م	خ
ماه، ۲۷	خارج مسطح، ۸۲
متریک، ۵	خاصیت شکاف، ۴۲
مخروط، ۲۰	خود-گرا، ۱۳
مسائل مجاورتی، ۹	د
مسیر حریصانه، ۵	دایکسترا، ۶

مقید به زاویه، ۱۶

**ن**

ناحیه‌های محدب، ۱۳

**و**

وتر-افزایشی، ۱۴

وزن، ۴

**T**

*t*-پوشش، ۲

*t*-تقریب، ۵

*t*-مسیر، ۲

**W**

*w*-شکاف، ۷



## Abstract

In this thesis, we propose some algorithms for constructing geometric spanners and improve some of previous algorithms. We also study some geometric properties of the geometric spanners. A geometric (Euclidean) graph is called  $t$ -spanner, if for every two vertices of the graph, there exists a path in the graph of length at most  $t$  times the Euclidean distance between them.

In this thesis, we first introduce the continuous Yao graphs. A continuous Yao graph is obtained by rotating a cone around every point and connecting it to the closest point in the cone during the rotation. Although by considering the cones with a suitable angular diameter the continuous Yao graphs are  $t$ -spanners, the number of the edges of the graph can be quadratic. But, we show that the continuous Yao graphs are region-fault tolerant  $t$ -spanner.

Then, we consider the gap-greedy algorithm that introduced in 1997 for constructing geometric spanners, and reduce the time and space complexity of the gap-greedy algorithm. Also, we improve the value of  $t$  in the results related to the  $t$ -spanner property of the gap-greedy algorithm.

Later, we consider the self-approaching graphs and the increasing-chord graphs. A geometric graph is called self-approaching, if for every two vertices of the graph, there exists a path between them such that by moving on the path, from source to destination, we continuously get closer to the forward points on the path. We call such a path a self-approaching path. A geometric graph is called increasing-chord if for every two vertices, there exists a path between them that is self-approaching from any of the two endpoints to another endpoint. In this thesis, we show that there exists an increasing-chord plane geometric graph for every point set whose points are on the boundary of an acute triangle, using at most three Steiner points. Furthermore, we show that there exists an increasing-chord plane geometric graph for every point set whose points are on the boundary of an obtuse or right triangle.

Then, we introduce weakly self-approaching graphs by replacing the condition getting closer to the forward points by getting closer to the destination, and we show that for every point set in the plane, there exists a weakly self-approaching plane graph.

Later, we propose a divide and conquer algorithm to construct a geometric spanner with near linear number of edges.

Finally, we consider the angle-constrained geometric graphs. For any  $\theta > 0$ , a geometric graph is called  $\theta$ -angle constrained if the angle between any two edges incident to a vertex is at least  $\theta$ . We show that for every  $\theta > \pi/3$ , a connected  $\theta$ -angle constrained graph does not exist, necessarily. Moreover, we show that there exists a point set such that for  $\theta = \pi/3$  and every  $t > 2/\sqrt{3}$ , there is no  $\theta$ -angle constrained  $t$ -spanner on the point set.



**Yazd University**  
**Faculty of Sciences**

A Dissertation Submitted in Partial Fulfillment of the  
Requirements for the Ph.D. in Computer Science

Title:

**Proposing Some Algorithms for Constructing  
Geometric Spanners**

Supervisor

**Dr. Mohammad Farshi**

Advisors

**Dr. Mohammad Ali Abam, Dr. Mahdieh Hasheminezhad**

By

**Davood Bakhshesh**

**July 2017**