

## الگوریتم‌های ابتکاری برای شبه مثُلث‌بندی مجموعه نقاط تصادفی در صفحه

فهیمه طاهر خانی<sup>\*</sup> منا نقده فروشها<sup>\*</sup> علی نورالله<sup>\*\*</sup>

<sup>\*</sup> کارشناسی ارشد، دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی، تاکستان

<sup>\*\*</sup> استادیار، دانشکده مهندسی برق و کامپیوتر، دانشگاه آزاد اسلامی، قزوین

تاریخ پذیرش: ۱۳۹۲/۰۶/۲۰

تاریخ دریافت: ۱۳۹۲/۰۳/۳۱

### چکیده

شامل بررسی عملکرد الگوریتم‌ها و ارزیابی زمان پردازنده مورد نیاز الگوریتم‌ها، حائز اهمیت می‌باشد.

کلید واژگان: شبه مثُلث‌بندی، پوسته محدب، زنجیره محدب و مقعر، قابلیت رویت، چندضلعی ساده.

### ۱. مقدمه

عنوانین شبه‌مثُلث و شبه‌مثُلث‌بندی در سال ۱۹۹۳ توسط Pocchiola و Vegter مطرح شده‌اند. در اوایل سال ۱۹۹۰ شبه‌مثُلث‌بندی چندضلعی‌ها در هندسه محاسباتی تحت نام مثُلث‌بندی‌های ژئودزیک بیان شده است [۱]. مسیر ژئودزیک کوتاهترین مسیر بین دو نقطه در یک چندضلعی بوده که هیچ یک اضلاع چندضلعی را قطع نمی‌کند. یک شبه‌مثُلث، یک چندضلعی ساده در صفحه دقیقاً با سه رأس محدب می‌باشد که آن‌ها را گوشه می‌نامند و سه زنجیره مقعر از یال‌ها، گوشه‌ها را به هم متصل می‌کند. فرض کنید  $S$  مجموعه‌ای از  $n$  نقطه در صفحه است شبه‌مثُلث‌بندی  $S$  یک تقسیم‌بندی از پوسته‌ی محدب  $S$  به شبه‌مثُلث‌هایی است که مجموعه رئوس آن  $S$  باشد [۲].

در سال ۲۰۰۰ Streinu, [۳] نشان داده که بین استحکام گراف‌ها و شبه‌مثُلث‌بندی‌های کمینه ارتباط قوی وجود دارد علاوه بر این او ثابت کرد که تعداد کمینه یال‌های مورد نیاز در یک شبه‌مثُلث‌بندی نوك‌دار  $3n - 2$  یال است. همچنین با استفاده از تئوری چند وجهی اویلر، تعداد شبه‌مثُلث‌ها در یک شبه‌مثُلث‌بندی کمینه نوك‌دار  $2n - 6$  است که به هیچ ساختاری از مجموعه نقاط بجز تعداد آن بستگی ندارد [۴]. در یک شبه‌مثُلث‌بندی، رأسی که در بین تمام زوایای آن یک زاویه بزرگ‌تر

یافتن الگوریتم‌هایی برای مسائل متنوع مطرح شده در هندسه محاسباتی از جمله مسئله شبه مثُلث‌بندی مجموعه نقاط در صفحه جزو موضوعات علمی است که تاکنون زمینه فکری دانشمندان علم کامپیوتر را به خود اختصاص داده است. شبه مثُلث‌بندی  $S$  که مجموعه‌ای از  $n$  نقطه در صفحه است، افزار پوسته‌ی محدب این مجموعه نقاط از طریق اتصال چندین یال به تعدادی شبه‌مثُلث می‌باشد که همه نقاط را در بر می‌گیرد. برای شبه مثُلث‌بندی معیارهای بهینگی گوناگونی بررسی شده است که اغلب براساس وزن یال‌ها و گوشه‌ها بوده که در آن شبه مثُلث‌بندی مجموعه نقاط با کمترین وزن یال‌ها جزو مسائل باز می‌باشد. به‌طور کلی شبه مثُلث‌بندی کمینه به شبه مثُلث‌بندی اطلاق می‌شود که تعداد شبه‌مثُلث‌های ایجاد شده در آن دقیقاً  $n-2$  شبه‌مثُلث و تعداد کمترین یال‌های مورد نیاز در آن  $3-2n$  یال باشد، همچنین تمامی رئوس یک شبه مثُلث‌بندی کمینه نوك‌دار می‌باشند؛ به این معنی که در بین تمام زوایای وابسته به آن رئوس، یک زاویه بزرگ‌تر از  $\pi$  وجود داشته باشد.

هدف این مقاله ارائه روش‌هایی جدید برای شبه مثُلث‌بندی مجموعه نقاط  $S$  در صفحه است تا بتواند تفکرات الگوریتمی جدیدی را در این زمینه باز کند. این مقاله نشان می‌دهد که ایجاد لایه‌هایی از پوسته محدب برای مجموعه نقاط و شبه مثُلث‌بندی آن‌ها با دو الگوریتم خاص منجر به تولید شبه مثُلث‌بندی کمینه خواهد شد. همچنین الگوریتمی جدید برای ایجاد یک چندضلعی ساده حلزونی شبه مثُلث‌بندی شده ارائه می‌دهد که تولید چندضلعی‌های ساده تصادفی در دو زمینه مهم کاربردی،

مثلث‌بندی مجموعه نقاط تصادفی در صفحه مطرح می‌شوند. در بخش ۵ تحلیل الگوریتم‌های پیشنهادی و نهایتاً در بخش ۶ نتیجه‌گیری ارائه خواهد شد.

## ۲. تعاریف اولیه

چندضلعی ساده یک سطح محدود شده به وسیله یک مجموعه محدود از پاره‌خط‌هایی است که یک خم ساده‌ی بسته را تشکیل می‌دهند. به عبارتی، یک چندضلعی ساده  $P$  روی مجموعه رئوس تصادفی  $S$ ، چندضلعی است که در آن یال‌ها جز در رئوس  $S$  هم‌دیگر را قطع نمی‌کنند.

به چندضلعی ساده‌ای که تمام زوایای داخلی بین رئوس آن چندضلعی از  $\pi$  کمتر باشد، چندضلعی محذب می‌گویند. بنا بر تعریف، مجموعه  $S$  از نقاط در صفحه محذب نامیده می‌شود اگر و فقط اگر به ازای هر دو نقطه  $p, q \in S$ ، پاره خط واصل  $pq$  کاملاً درون  $S$  واقع شود ( $pq \subseteq S$ ). پس می‌توان نتیجه گرفت که یک چندضلعی با هر نوع فرورفتگی نمی‌تواند محذب باشد.

پرکاربردترین ساختار در هندسه ریاضیک پوسته محذب می‌باشد. پوسته محذب مجموعه نقاط  $S$  در صفحه  $(CH(S))$ ، کوچکترین چندضلعی محذب  $P$  است که  $S$  را محصور می‌کند (شکل ۲-۲). اگر  $k$  تعداد لایه‌های پوسته محذب مجموعه نقاط  $S$  باشد هر لایه با  $l$  معرفی می‌شود ( $1 \leq l_c \leq k, k > 1$ ). در یک مجموعه از نقاط  $S$  نقاطی از پوسته محذب که در آن‌ها زاویه داخلی اکیداً کوچک‌تر از  $\pi$  باشد را نقاط رأسی می‌نامند.

سه نقطه  $p_1(x_1, y_1)$ ,  $p_2(x_2, y_2)$  و  $p_3(x_3, y_3)$  مفروض هستند. ماتریس  $A$  این گونه تعریف می‌شود:

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \quad (1)$$

در صورتی که  $\det(A) > 0$  (دترمینان ماتریس  $A$ ) باشد بین سه نقطه  $p_1 p_2 p_3$  رابطه پادساعتگردی وجود دارد و اگر  $\det(A) < 0$  باشد بین این سه نقطه رابطه ساعتگردی وجود دارد و  $\det(A) = 0$  به این معنی است که سه نقطه در یک خط واقع شده‌اند.

دو نقطه در یک چندضلعی نسبت به هم قابل رویت هستند اگر پاره‌خط واصل بین آن‌ها با هیچ پاره‌خط دیگری از همان مجموعه متقطع نباشد (در حالت خاص دو پوسته محذب  $l_1$  و  $l_2$  را در نظر بگیرید بطوریکه  $l_2$  درون  $l_1$  باشد. فرض کنید  $p_1$  و  $p_2$  دو نقطه رأسی به ترتیب در  $l_1$  و  $l_2$  باشند، این دو نقطه را نسبت به هم قابل رویت

از  $\pi$  وجود داشته باشد نوک‌دار نامیده می‌شود. تمامی رئوس یک شبه مثلث‌بندی کمینه نوک‌دار می‌باشند [۵-۶].

شبه‌مثلث‌بندی‌ها به طور قابل ملاحظه‌ای در هندسه محاسباتی مورد توجه قرار گرفته‌اند که عمدها ناشی از کاربردشان در تئوری استحکام، برنامه‌ریزی حرکت بازوی روبات‌ها، مجموعه‌های قابل رویت، شعاع پرتاب، ردیابی تصادم جنبشی و نگهبانی کردن از چندضلعی‌ها می‌باشد [۷-۱۰]. با توجه به اینکه برخی از خواص ترکیبی و هندسی جالب شبه‌مثلث‌بندی‌ها اخیراً کشف شده اما هنوز بسیاری از پرسش‌های اصلی باز در مورد آن‌ها باقی مانده است [۲].

در این مقاله به مسئله شبه‌مثلث‌بندی مجموعه  $S$  با  $n$  نقطه تصادفی در صفحه با استفاده از پوسته‌های محذب لایه‌ای پرداخته و در آن دو روش جدید شبه‌مثلث‌بندی مطرح می‌شود که منجر به تولید شبه‌مثلث‌بندی کمینه خواهد شد. همچنین الگوریتمی جدید برای ایجاد یک چندضلعی ساده حلزونی ارائه شده است که به طور همزمان نیز شبه‌مثلث‌بندی روی آن انجام می‌شود. بطورکلی تولید چندضلعی‌های ساده تصادفی در دو زمینه‌ی مهم کاربردی حائز اهمیت است:

- بررسی عملکرد الگوریتم‌ها

• ارزیابی زمان پردازنده مورد نیاز الگوریتم‌ها.

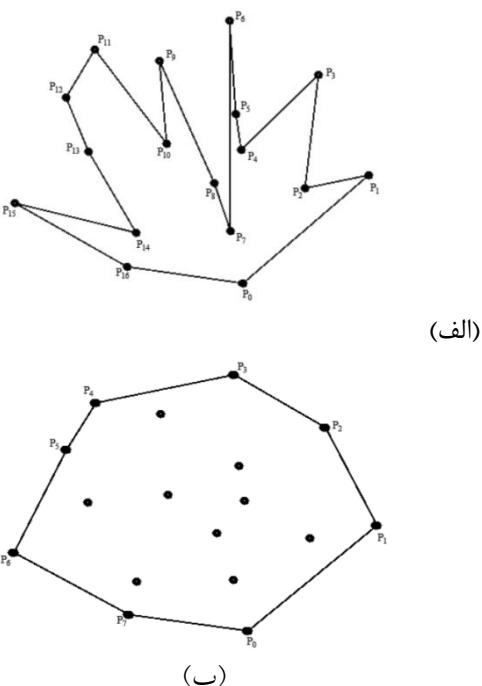
تولید اشیاء هندسی تصادفی مورد توجه خاص محققان قرار گرفته است. به عنوان مثال Epstein تولید مثلث‌بندی تصادفی یکنواخت را مورد مطالعه قرار داده است [۱۱].

مجموعه  $S$  شامل  $n$  نقطه تصادفی  $p_0, p_1, \dots, p_{n-1}$  در صفحه مفروض است. هدف ایجاد یک چندضلعی ساده‌ی تصادفی با توزیع یکنواخت است (اگر  $k$  چندضلعی ساده روی  $S$  وجود داشته باشد احتمال تولید همه این چندضلعی‌ها با یکدیگر یکسان بوده و برابر  $\frac{1}{k}$  است) [۱۱-۱۲]، به گونه‌ای که مجموعه رئوس چندضلعی ایجاد شده شامل همه مجموعه نقاط  $S$  باشد که به طور همزمان نیز شبه‌مثلث‌بندی می‌شود.

دو روش جدید دیگری از شبه‌مثلث‌بندی را نیز ارائه می‌دهیم که نسبت به سه روش دیگر ویژگی خاصی نداشته و فقط به عنوان الگوریتمی جدید برای شبه‌مثلث‌بندی مجموعه نقاط تصادفی در صفحه مطرح شده‌اند.

ادامه مقاله به این شکل سازماندهی شده است که در بخش ۲ تعاریف اولیه ارائه می‌گردد. در بخش ۳ نحوه ایجاد پوسته‌های محذب لایه‌ای بیان می‌شود. در بخش ۴ الگوریتم‌های پیشنهادی جهت شبه-

بالای پشته قرار گیرد. سپس دو رأس سر پشته به همراه رأس بعدی (رأس  $p_2$ ) واقع شده بر روی مرز چندضلعی ستاره‌ای (شکل) بررسی شده و پاد ساعتگردی یا ساعتگردی این سه رأس متوازی تعیین می‌شود. اگر زاویه پاد ساعتگرد بود رأس نظر را در پشته درج می‌کند در غیر این صورت عنصر سر پشته حذف می‌شود و به همین ترتیب الگوریتم تکرار می‌گردد. در نهایت تمام رئوسی که در پشته قرار دارند همان رئوس واقع شده بر روی پوسته محدب می‌باشند (شکل ۲-ب).



شکل ۲-الف- چندضلعی ستاره‌ای شکل. ب- پوسته محدب

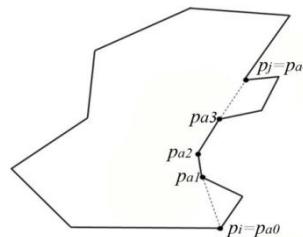
اکنون نقاط روی این پوسته محدب را از مجموعه نقاط خارج می‌کنیم و برای نقاط باقی مانده با انجام مراحل قبل، مجدداً پوسته محدب ایجاد می‌شود و این کار تا زمانی ادامه پیدا می‌کند که به کمتر از سه نقطه برسد به این معنی که یا یک نقطه و یا دو نقطه باقی مانده باشد. بدین ترتیب پوسته‌های محدب لایه‌ای ایجاد می‌شوند.

#### ۴. الگوریتم‌های پیشنهادی شبه مثُلث‌بندی مجموعه نقاط تصادفی در صفحه

فرض کنید تعداد لایه‌های محدب ایجاد شده در بخش قبلی، باشد  $L_j$  یعنی لایه‌ی  $j$  را که  $M = 1, \dots, M$  است و  $P_{i,j}$  رأس  $i$  در لایه  $j$  را می‌باشد.

می‌گوییم هرگاه پاره خط وصل بین آنها با هیچ پاره خط دیگری از همان مجموعه متقاطع نباشد).

فرض کنید  $p_0, p_1, \dots, p_{n-1}$  رئوس یک چندضلعی ساده  $P$  می‌باشند که این رئوس در خلاف جهت حرکت عقربه‌های ساعت قرار گرفته‌اند (شکل ۱).  $\delta(p_i, p_j)$  را کوتاهترین مسیر بین دو رأس  $p_i$  و  $p_j$  از رئوس  $P$  می‌نامیم. مسیر  $\delta(p_i, p_j)$  را زنجیره محدب گویند اگر از رأس  $i$  به رأس  $j$  روی مسیر آن حرکت کیم مسیر راست‌گرد باشد در غیر این صورت مسیر  $\delta(p_i, p_j)$  را زنجیره مقعر می‌نامند.



شکل ۱- زنجیره‌ی محدب و مقعر

#### ۳. ایجاد پوسته‌های محدب لایه‌ای

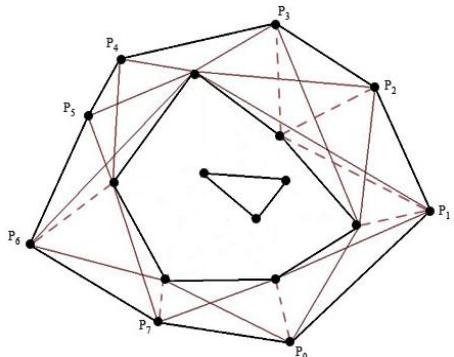
در این بخش به بررسی الگوریتم ایجاد پوسته محدب خواهیم پرداخت. جهت ایجاد پوسته محدب مختصات رئوس و ترتیب اتصال آنها به یکدیگر مورد نیاز است. برای ایجاد پوسته محدب الگوریتم‌های متفاوتی وجود دارد که در این مقاله از الگوریتم گراهام نسخه  $B$  استفاده شده است. جهت محاسبه پوسته محدب ابتدا باید نقاط کرانی را پیدا کرد که در این الگوریتم پایین‌ترین نقطه سمت راست یک نقطه کرانی می‌باشد. مجموعه  $S$  با  $n$  نقطه در صفحه مفروض است. بر اساس الگوریتم اسکن گراهام نسخه  $B$ , مراحل زیر طی می‌شود:

مرحله اول: پایین‌ترین نقطه سمت راست را پیدا کرده و این نقطه را  $p_0$  می‌نامیم.

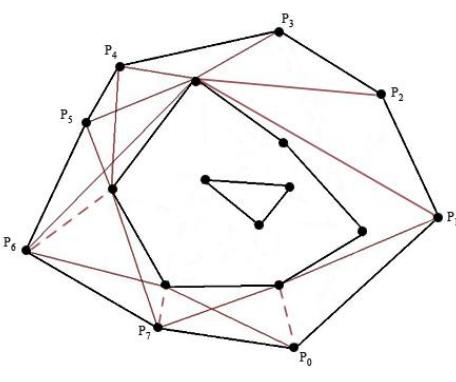
مرحله دوم: مابقی نقاط بر اساس زاویه حول نقطه  $p_0$  مرتب می‌شوند. در صورتی که دو نقطه زاویه یکسانی با  $p_0$  داشته باشند، یعنی در یک راستا باشند نقطه‌ای در نظر گرفته می‌شود که فاصله بیشتری با نقطه  $p_0$  دارد. سپس این نقاط را از  $p_1, p_2, \dots, p_{n-1}$  نام‌گذاری کرده و با اتصال این نقاط به یکدیگر یک چندضلعی ستاره‌ای شکل ایجاد خواهد شد (شکل ۲-الف).

مرحله سوم: پایین‌ترین نقطه (رأس  $p_0$ ) و نقطه بعد از آن (رأس  $p_1$ ) که کمترین زاویه را نسبت به  $p_0$  دارد، قطعاً بر روی پوسته محدب واقع شده‌اند پس این دو رأس در پشته درج می‌شوند بطوریکه  $p_1$  در

رویتی که کمترین و بیشترین زاویه قطبی را نسبت به این رأس دارند را به عنوان دو گوشه دیگر شبهمثلث درنظر می‌گیریم و یال‌های مربوطه به عنوان دو ضلع شبهمثلث از رأس  $P_{i,j}$  به دو رأس مورد نظر رسم می‌شوند.



(الف)

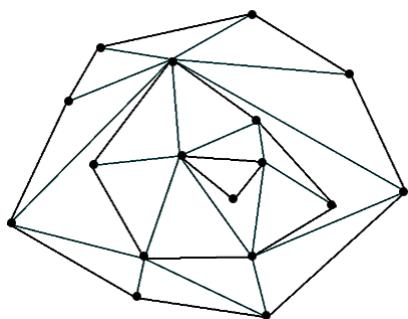


(ب)

شکل ۳-الف- تعیین نقاط قابل رویت برای هر رأس از لایه  $L_j$  در مرحله اول،

ب- ثابت شدن دو رأس قابل رویت  $P_1$  و تعیین مجدد نقاط قابل رویت دیگر

رئوس از لایه  $L_j$



شکل ۴- خروجی نهایی الگوریتم بیشترین نقاط قابل رویت

براساس شکل (۵-الف) مجموع فاصله اقلیدوسی دو دورترین رأس قابل رویت نسبت به رأس  $P_5$  از لایه  $L_j$  کمترین مقدار را داشته، بنابراین در این مرحله رئوس قابل رویت رأس  $P_5$  ثابت شده (مطابق با

#### ۴.۱. الگوریتم پیشنهادی بیشترین نقاط قابل رویت

در این الگوریتم هر رأس  $P_{i,j}$  رأسی است که بیشترین نقاط قابل رویت از لایه  $L_{j+1}$  را دارد. هر رأس  $P_{i,j}$  تعدادی از رئوس لایه  $L_{j+1}$  را رویت می‌کند. براساس این الگوریتم رأسی که بیشترین نقاط قابل رویت از لایه  $L_{j+1}$  را داشته باشد؛ به عنوان یک گوشه از شبهمثلث انتخاب شده و دو رأس قابل رویت که کمترین و بیشترین زاویه قطبی را نسبت به این رأس دارند را به عنوان دو گوشه دیگر شبهمثلث درنظر می‌گیریم و یال‌های مربوطه به عنوان دو ضلع شبهمثلث از رأس  $P_{i,j}$  به دو رأس مورد نظر رسم می‌شوند.

زنگیره‌ی معمعری که بر روی لایه  $L_{j+1}$  این دو رأس قابل رویت را به یکدیگر وصل می‌کند؛ به عنوان ضلع سوم شبهمثلث محسوب می‌شود. این الگوریتم را برای دیگر رئوس باقی‌مانده از لایه  $L_j$  تکرار می‌کنیم با این شرط که نقاط بین دو گوشه‌ی انتهایی ضلع مقعر شبهمثلث بددست آمده در مرحله قبل به عنوان نقاط قابل رویت مرحله بعد درنظر گرفته نشوند.

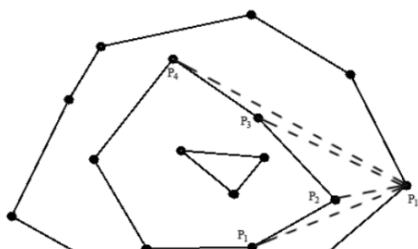
همان‌طور که در شکل (۳-الف) مشاهده می‌شود رئوسی که بیش از دو رأس قابل رویت دارند (رئوس قابل رویت بین دو رأسی که کمترین و بیشترین زاویه قطبی را داشته)؛ با استفاده از خط‌چین نمایان شده‌اند. از بین تمامی رئوس لایه  $L_j$  رأس  $P_1$  بیشترین نقاط قابل رویت را دارد (چهار نقطه قابل رویت)؛ بنابراین در این مرحله دو رأس قابل رویت  $P_1$  که کمترین و بیشترین زاویه قطبی را دارند ثابت شده (شکل ۳-ب) و مجدداً الگوریتم برای دیگر رئوس تکرار می‌شود. در نهایت خروجی الگوریتم مطابق با شکل (۴) خواهد بود. با اعمال این الگوریتم بر روی تمامی لایه‌ها شبهمثلث‌بندی مجموعه نقاط داده شده در صفحه ایجاد می‌شود.

#### ۴.۲. الگوریتم پیشنهادی کوتاهترین طول

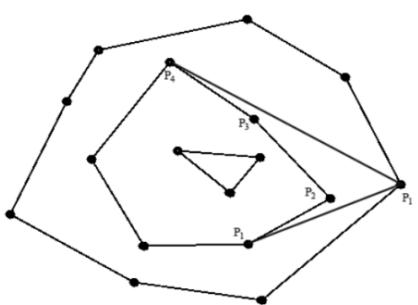
براساس این الگوریتم برای تمامی رئوس  $P_{i,j}$  از لایه  $L_j$  بطری جداگانه رئوس قابل رویتشان از لایه  $L_{j+1}$  تعیین می‌شود. سپس دو رأس قابل رویتی که کمترین و بیشترین زاویه قطبی را نسبت به رئوس  $P_{i,j}$  دارند درنظر گرفته شده و مجموع فاصله اقلیدوسی بین تک‌تک رئوس  $P_{i,j}$  و دو رأس قابل رویت‌شان محاسبه می‌شود (شکل ۵-الف). از بین تمامی رئوس  $P_{i,j}$  از لایه  $L_j$ ، رأسی که کوتاهترین طول را داشته باشد به عنوان گوشه‌ای از شبهمثلث انتخاب شده و دو رأس قابل

در مرحله قبل به عنوان نقاط قابل رویت مرحله بعد در نظر گرفته نشوند. با اعمال این الگوریتم بر روی تمامی لایه‌ها شبه مثلاً بندی مجموعه نقاط داده شده در صفحه ایجاد می‌شود. با توجه به اینکه این الگوریتم براساس کوتاهترین طول عمل می‌کند جزو الگوریتم‌های شبه مثلاً بندی مجموعه نقاط با کمترین هزینه در نظر گرفته می‌شود.

**۴.۲. الگوریتم‌های پیشنهادی شبه مثلاً بندی کمینه نوک دار**  
 برای انجام الگوریتم ابتدا از بین تمام رئوس قابل رویت هر رأس  $P_{i,j}$ , دو رأس قابل رویتی که کمترین و بیشترین زاویه قطبی را نسبت به این رأس دارند به عنوان دو دورترین رأس قابل رویت منظور می‌گردند. بدین ترتیب برای تمامی رئوس لایه  $L_j$  نسبت به لایه  $L_{j+1}$ , دو رأس قابل رویت وجود خواهد داشت. به عنوان مثال همان‌طورهای طور شکل (الف) نشان داده شده است رأس  $p_1$  از لایه محدب  $L_j$  چهار رأس  $p_4, p_3, p_2, p_1$  و  $p_4$  را از لایه محدب  $L_{j+1}$  رویت می‌کند که  $p_1$  و  $p_4$  به عنوان دو دورترین رأس قابل رویت انتخاب می‌شوند (شکل ۷-ب).



(الف)

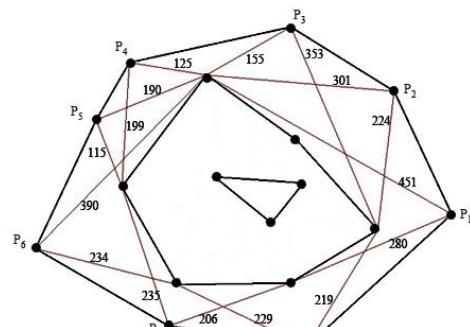


(ب)

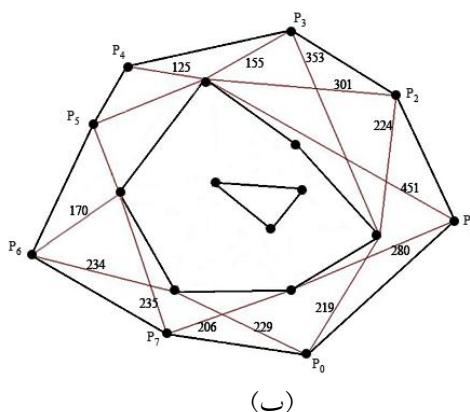
شکل ۷- الف- رئوس قابل رویت رأس  $p_1$  از لایه  $L_j$ . ب- دو دورترین رأس قابل رویت رأس  $p_1$

همان‌طور که بیان شد، تمامی رئوس در پوسته‌های محدب لایه‌ای نسبت به لایه بعدی خود دو دورترین رأس قابل رویت دارند. در این بخش برای ایجاد شبه مثلاً بندی باید یکی از این دو رأس انتخاب شوند

شکل ۵- ب) و مجدداً الگوریتم برای دیگر رئوس تکرار می‌شود. در نهایت خروجی الگوریتم مطابق با شکل (۶) خواهد بود:

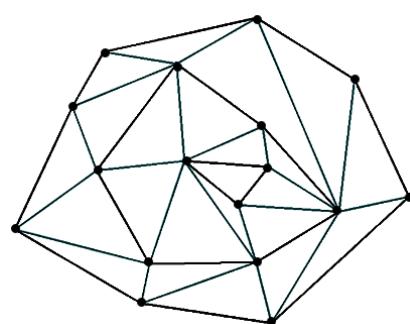


(الف)



(ب)

شکل ۵- الف- تعیین نقاط قابل رویت برای هر رأس از لایه  $L_j$  در مرحله اول بر اساس فاصله اقلیدوی، ب- ثابت شدن دو رأس قابل رویت  $P_5$  و تعیین مجدد نقاط قابل رویت دیگر رئوس از لایه  $L_j$

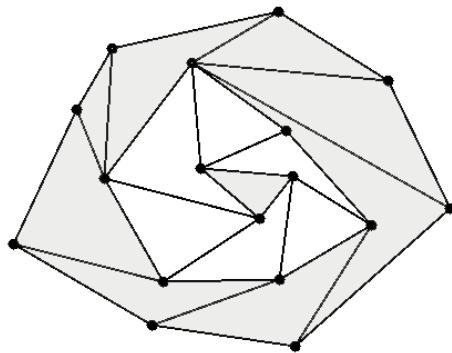


شکل ۶- خروجی نهایی الگوریتم کوتاهترین طول

در این الگوریتم نیز همانند الگوریتم قبلی زنجیره‌ی مقعری که بر روی لایه‌ی محدب  $L_{j+1}$  این دو رأس قابل رویت را به یکدیگر وصل می‌کند؛ به عنوان ضلع سوم شبه مثلاً بندی محاسبه می‌شود. این الگوریتم را مجدداً برای دیگر رئوس باقی‌مانده از لایه‌ی  $L_j$  تکرار می‌کنیم با این شرط که نقاط بین دو گوشی‌ای ضلع مقعر شبه مثلاً بندی بدست آمده

رأس مورد نظر از لایه  $L$  و دو رأس قابل رویت آن رابطه پادساعت-گردی برقرار باشد. بدین ترتیب که، رأس مورد نظر در لایه زام را درنظر گرفته و دو رأس قابل رویت را  $P_{s,j+1}$  و  $P_{k,j+1}$  می‌نامیم. چرخش از رأس مورد نظر به دو رأس قابل رویت را بررسی کرده اگر  $(P_{i,j}P_{s,j+1}P_{k,j+1})$  پادساعت-گرد باشد پاره خط بین  $j$  و  $i$  متصل می‌شود در صورتی که  $(P_{i,j}P_{k,j+1}P_{s,j+1})$  پادساعت-گرد باشد پاره خط ساختگردد برقرار باشد. پاره خط بین  $j$  و  $i$  متصل می‌شود. با اتصال این خطوط در هر لایه رئوس قابل رویت پادساعت-گرد تولید می‌شوند (شکل ۸-ب).

چنانچه یکی از دو روش مطرح شده و یا ترکیبی از دو روش بطوریکه برای هر یک از لایه‌ها، تنها یکی از دو روش استفاده شود) را بر روی پوسته‌های محدب لایه‌ای اعمال کنیم لایه‌های محدب مطابق شکل ۹ شبهمثربندی می‌شوند.



شکل ۹- خروجی الگوریتم به روش پادساعت-گرد

۴.۴. الگوریتم ایجاد چندضلعی ساده حلزونی شبهمثربندی شده الگوریتم پیشنهادی شامل دو مرحله است که با اعمال این دو مرحله بر روی لایه‌های محدب، با ایجاد شبهمثربندی‌های بی‌درپی در نهایت چندضلعی حلزونی شکل شبهمثربندی شده حاصل می‌شود.

مرحله اول الگوریتم:

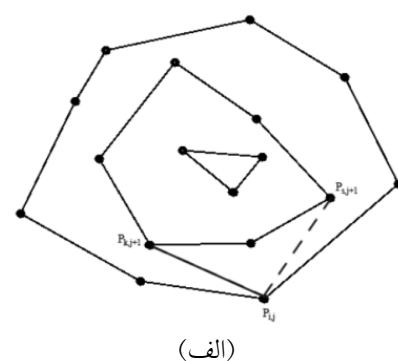
گام اول- ۱  $\leftarrow$  زرا در نظر گرفته و روی  $L$  (بیرونی‌ترین لایه محدب) یک نقطه را به عنوان نقطه شروع انتخاب نموده و آن را  $p_{1,j}$  می‌نامیم.

گام دوم- در جهت پادساعت-گرد دو نقطه دیگر را به ترتیب انتخاب و آنها را  $p_{2,j}$  و  $p_{3,j}$  نامگذاری می‌نامیم. این سه نقطه را به عنوان رئوس شبهمثربندی درنظر گرفته و پاره خط‌های  $p_{1,j}p_{2,j}$  و  $p_{2,j}p_{3,j}$  را ایجاد می‌کنیم.

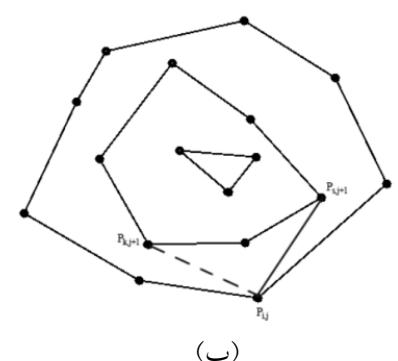
که در ادامه دو روش جدید برای انتخاب یکی از دو رأس قابل رویت جهت شبهمثربندی مطرح می‌شود.

#### ۴.۳.۱. الگوریتم انتخاب رئوس قابل رویت ساعت-گرد

در روش انتخاب رئوس قابل رویت ساعت-گرد، از بین دو دورترین رأس قابل رویت تعیین شده، رأسی انتخاب می‌شود که بین رأس مورد نظر از لایه  $L$  و دو رأس قابل رویت آن رابطه ساعتگردی برقرار باشد. بنابراین رأس مورد نظر در لایه زام را  $P_{i,j}$  در نظر گرفته و دو رأس قابل رویت را  $P_{s,j+1}$  (اندیس نزدیک‌ترین رأس قابل رویت) و  $P_{k,j+1}$  (اندیس دورترین رأس قابل رویت) می‌نامیم. چرخش از رأس مورد نظر به دو رأس قابل رویت را بررسی کرده اگر  $(P_{i,j}P_{s,j+1}P_{k,j+1})$  ساعت-گرد باشد پاره خط بین  $j$  و  $i$  و  $P_{s,j+1}$  متصل می‌شود. با اتصال این خطوط در هر لایه رئوس قابل رویت ساعت-گرد تولید می‌شوند (شکل ۸-الف)



(الف)

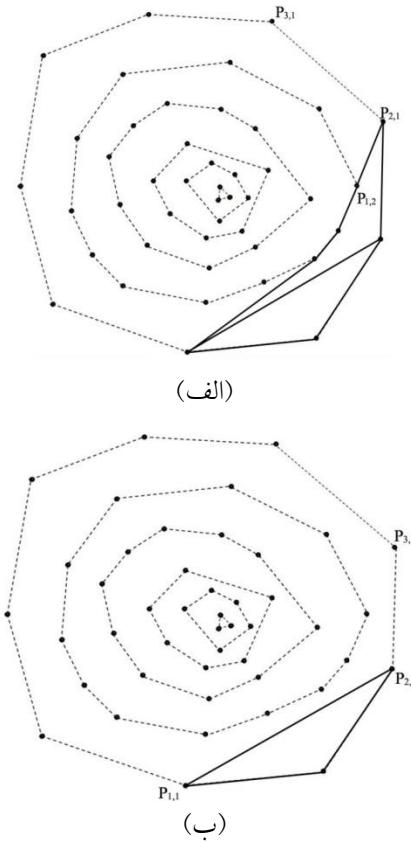


(ب)

شکل ۸- الف- انتخاب رأس قابل رویت ساعت-گرد برای رأس  $P_{i,j}$ . ب- انتخاب رأس قابل رویت پادساعت-گرد برای رأس  $P_{i,j}$

#### ۴.۳.۲. الگوریتم انتخاب رئوس قابل رویت پادساعت-گرد

در روش انتخاب رئوس قابل رویت پادساعت-گرد، از بین دو دورترین رأس قابل رویت تعیین شده، رأسی انتخاب می‌شود که بین



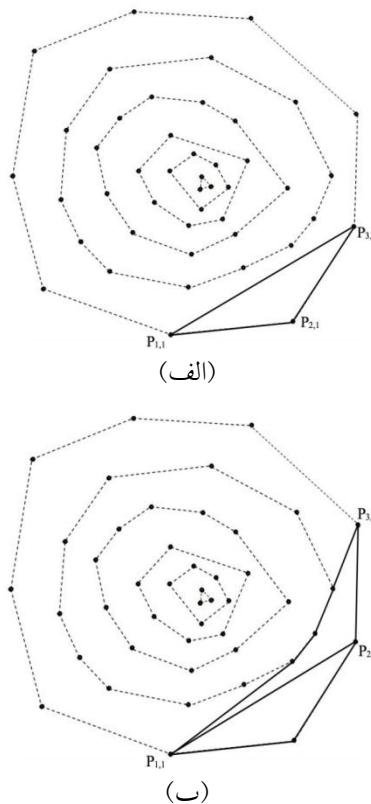
شکل ۱۱-الف- تغییر موقعیت مکانی رئوس مثلث در صورت وجود تقاطع با لایه  $L_{j+1}$   
شکل ۱۱-ب- تغییر موقعیت مکانی رئوس مثلث در صورت عدم تقاطع با لایه  $L_{j+1}$

#### مرحله دوم الگوریتم:

با ملاقات آخرین نقطه روی لایه  $L_j$ ، در این مرحله از بین نقاط باقیمانده روی لایه  $L_{j+1}$  که در ایجاد زنجیره مقرر در مرحله اول الگوریتم استفاده نشده‌اند، دورترین نقطه قابل رویت نسبت به آخرین نقطه روی لایه  $L_j$  یعنی رأس  $p_{3,j}$  را پیدا کرده آن را  $v$  می‌نامیم و پاره خط واصل بین آن دو رسم می‌کنیم. همچنین از رأس  $p_{1,j}$  به نقطه قابل رویت روی لایه  $L_{j+1}$  حرکت کرده و به ترتیب پاره خط‌های واصل بین نقاط موجود در این مسیر را رسم می‌نماییم.

بعد از اتمام این دو مرحله، با در نظر گرفتن  $j + 1 \leftarrow j \leftarrow z$  وارد لایه  $L_j$  بعدی شده و نقطه مجاور به دورترین نقطه قابل رویت (در مرحله دوم از الگوریتم) در جهت ساعت‌گرد را به عنوان نقطه شروع این لایه درنظر می‌گیریم و از گام دوم مرحله اول، الگوریتم را با نقاط باقیمانده (بجز دورترین نقطه قابل رویت) در این لایه، ادامه می‌دهیم و مراحل را تا  $M < Z$  تکرار می‌کنیم (شکل ۱۲-الف). بدین ترتیب با تکرار مراحل الگوریتم پیشنهادی تا  $M < Z$  چندضلعی ساده حلزونی شبه مثلث‌بندی شده ایجاد می‌گردد (شکل ۱۲-ب).

گام سوم- در این گام جهت ایجاد پاره خط واصل بین دو رأس  $p_{1,j}$  و  $p_{3,j}$ ، در صورت عدم وجود تقاطع با لایه  $L_{j+1}$ ، پاره خط مذکور رسم می‌شود (شکل ۱۰-الف)، در غیر اینصورت باید از رأس  $p_{1,j}$  به رأس  $p_{3,j}$ ، نقاطی از لایه  $L_{j+1}$  که با این دو رأس زنجیره مقرر می-سازند انتخاب و رسم شوند. در این مرحله از الگوریتم یک شبه مثلث ایجاد شده است (شکل ۱۰-ب).



شکل ۱۰-الف- عدم تقاطع با لایه  $L_{j+1}$ . ب- تقاطع با لایه  $L_{j+1}$

گام چهارم- برای ایجاد شبه مثلث بعدی باید موقعیت مکانی رئوس  $p_{1,j}$  و  $p_{2,j}$  و  $p_{3,j}$  با درنظر گرفتن شرایط زیر تغییر یابند:  
(الف) موقعیت مکانی نقطه  $p_{1,j}$  در صورت تقاطع پاره خط  $p_{1,j}p_{3,j}$  با لایه  $L_{j+1}$  تغییر کرده و به نقطه مجاور رأس  $p_{3,j}$  روی زنجیره مقرر تغییر مکان می‌یابد (شکل ۱۱-الف).

(ب) رأس  $p_{2,j}$  به موقعیت مکانی رأس  $p_{3,j}$  فعلی منتقل شده و نقطه مجاور آن در جهت پادساعت‌گرد روی لایه  $L_j$  را رأس  $p_{3,j}$  نامیده و پاره خط  $p_{2,j}p_{3,j}$  را ایجاد می‌کنیم (شکل ۱۱-الف و ۱۱-ب).

**Algorithm Random Polygon Generation**

```

j←1
i←1
P1,j ← p1,j
while j < M do
    while i < nj do
        i ← i + 1
        P2,j ← pi,j
        Draw Line (P1,j, P2,j)
        i ← i + 1
        P3,j ← pi,j
        Draw Line (P2,j, P3,j)
        if (P1,j, P3,j) ∩ lj+1 ≠ Ø then
            find the reflex chain on lj+1 between P1,j, P3,j and draw it
            P4,j ← last element on the reflex chain
        else
            Draw Line (P1,j, P3,j)
            i ← i + 1
            P2,j ← p3,j
            P3,j ← p1,j
        end if
    end while
    v ← find last point which is visible from P3,j among all remain
    points in lj+1
    Draw Line (P3,j, v)
    Draw a chain from P1,j to v in clockwise direction
    i ← index of the nearest point to v in clockwise direction
    j ← j + 1
end while
end of algorithm

```

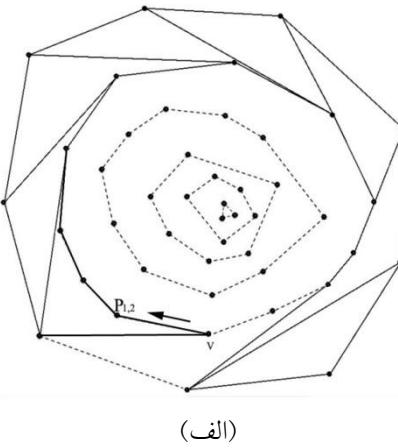
**اثبات درستی الگوریتم**

در گام اول الگوریتم چون تعداد لایه‌های محدب برای هر مجموعه از نقاط،  $1 \leq M$  است (لذا وجود نقطه  $p_{1,j}$  بدینهی است).

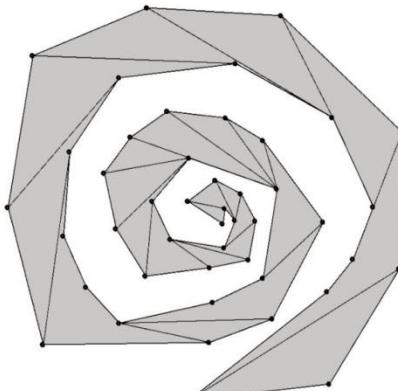
لم ۱. در گام سوم الگوریتم بر روی لایه‌ی  $L_j$  مثلث  $\Delta ABC$  مفروض است اگر رأس  $C$  از طریق رأس  $A$  قابل رویت نباشد نقاطی جهت ایجاد زنجیره مقعر از لایه‌ی  $L_{j+1}$  وجود دارد.

اثبات. اگر رأس  $C$  از رأس  $A$  قابل رویت نباشد به این معنی است که نقاطی از لایه‌ی  $L_{j+1}$  در  $\Delta ABC$  واقع شده است. از بین این نقاط نقطه‌ای که دارای کمترین زاویه قطبی نسبت به رأس  $A$  است را پیدا کرده و آن را  $Q$  می‌نامیم. بدینهی است که  $AQC$  چپ‌گرد است و لذا بر روی زنجیره مقعر قرار خواهد گرفت از این رو وجود زنجیره مقعر قابل اثبات خواهد بود.

با شروع الگوریتم بر روی لایه‌ی  $L_j$  و قوع اولین تقاطع با لایه‌ی  $L_{j+1}$  و ایجاد زنجیره مقعر، اولین نقطه در جهت پادساعت‌گرد را  $Q_1$  و آخرین نقطه را  $Q_n$  (با ملاقات آخرین نقطه روی لایه‌ی  $L_j$  در مرحله دوم الگوریتم) می‌نامیم. اگر  $Q_n$  از طریق لایه‌ی  $L_{j+1}$  به  $Q_1$  متصل باشد به این معنی است که نقاط ایز لایه‌ی  $L_{j+1}$  به پایان رسیده‌اند (تمامی نقاط در ایجاد زنجیره مقعر استفاده شده‌اند) و لذا الگوریتم را باید از لایه‌ی  $L_{j+2}$  ادامه دهیم در غیر اینصورت از بین نقاط باقی‌مانده بر روی لایه‌ی  $L_{j+1}$  نقطه با کمترین زاویه قطبی نسبت به آخرین نقطه بر روی لایه‌ی  $L_j$  را پیدا کرده و آن را  $v$  می‌نامیم. بدینهی است که  $v$  از این نقاط آخرین نقطه بر روی لایه‌ی  $L_j$  قابل رویت می‌باشد.



(الف)



(ب)

شکل ۱۲- الف- ورود به لایه  $L_{j+1}$ . ب- خروجی الگوریتم ایجاد چندضلعی ساده حلزونی شبه‌مثلث‌بندی شده

قضیه ۱. الگوریتم ابتکاری ارائه شده دو برابر تعداد نقاط موجود بر روی بیرونی ترین لایه محدب، چندضلعی ساده تصادفی تولید می‌کند. اثبات.

از آنجایی که نقطه شروع الگوریتم هر یک از نقاط چندضلعی محدب بیرونی ترین لایه می‌تواند باشد و جهت اجرا نیز ساعتگرد یا پادساعتگرد انتخاب گردد، لذا دو برابر تعداد نقاط موجود بر روی بیرونی ترین لایه محدب، چندضلعی ساده تصادفی تولید خواهد شد. شیوه کد مربوط به الگوریتم ایجاد چندضلعی ساده حلزونی شبه-مثلث‌بندی شده به صورت زیر می‌باشد:

مسائلی چون موزه هنری می‌باشند و لذا الگوریتم‌هایی برای تولید چندضلعی بسیار در این کار مؤثرند.

### مراجع

- [1]. G. Rote, F. Santos, and I. Streinu, “Pseudo-Triangulation – a Survey,” *Contemporary Mathematics*, 453: 343-410, American Mathematical Society, 2008.
- [2]. O. Aichholzer, F. Aurenhammer, H. Krasser, and B. Speckmann, “Convexity minimizes pseudo-triangulations,” *Computational Geometry* 28, pp.3-10, 2004.
- [3]. I. Streinu, “A combinatorial approach to planar non-colliding robot arm motion planning,” In : Proc. 41st Annu.IEEE Sympos. Foundat. Comput.Sci. (FOCS'00), , pp.443-453, 2000.
- [4]. S. Gerdjikov, and A. Wolff, “Decomposing a simple polygon into pseudo-triangles and convex polygons,” *Computational Geometry* 41, pp.21-30, 2008.
- [5]. M. Ben-Ner, A. Schulz, and A. Sheffer, “On numbers of pseudo-triangulations,” arXiv: 1210. 7126v1 [cs.CG] 26 oct 2012.
- [6]. O. Aichholzer, T. Hackl, and B. Vogtenhuber, “Compatible pointed pseudo-triangulations,” 22<sup>nd</sup> Canadian Conference on Computational Geometry, 2010.
- [7]. M. Pocchiola, and G. Vegter, “Topologically sweeping visibility complexes via pseudo-triangulations,” *Discrete Compute.Geom.* 16, pp.419-453, 1996.
- [8]. M.T. Goodrich, and R. Tamassia, “Dynamic ray shooting and shortest paths in planar subdivisions via balanced geodesic triangulations,” *J. Algorithms* 23 (1), pp.51-73, 1997.
- [9]. D.G. Kirkpatrick, J. Snoeyink, and B. Speckmann, “kinetic collision detection for simple polygons,” *Internat. J. Comput. Geom. Appl.* 12(1-2), pp. 3-27, 2002.
- [10]. B. Speckmann, and C.D. Tóth, “Allocating vertex  $\pi$ -guard in simple polygons via pseudo-triangulations,” *Discrete Comput. Geom.* 33 (2), pp.345-364, 2005.
- [11]. T. Aure, and M. Held, “Heuristic for generation of random polygons,” 8<sup>th</sup> Canadian Conference On Computational Geometry (CCCG), Ottawa, Canada, pp.38-44,1996.
- [12]. D. Dailey, and D. Whitfield, “Constructing random polygons,” SIGITE'08, USA, pp.119-124, 2008.
- [13]. M. D. Berg, Computational Geometry: Algorithms and Applications, 3<sup>rd</sup> edition, published by Springer-Verlag, 2008.
- [14]. B. Chazelle, “On the convex layers of a planar Set,” *IEEE Tran. Information Theory*, Vol. IT-31, No. 4, pp. 509-517, 1985.

### ۵. تحلیل الگوریتم‌های پیشنهادی شبه مثُلث‌بندی مجموعه نقاط

ایجاد گراف قابل دید دارای مرتبه زمانی ( $O(nlogn)$  بوده) [۱۳] و مرتبه زمانی ایجاد پوسته‌های محدب لایه‌ای ( $O(nlogn)$  می‌باشد) [۱۴]. در بخش ۴ با اجرای هر کدام از الگوریتم‌های پیشنهادی جهت ایجاد پاره خط واصل بین دو نقطه، موضوع قابل رویت بودن دو نقطه باید بررسی شود که با ایجاد گراف قابل دید مرتبه زمانی آن ( $O(nlogn)$ ) است. با توجه به مسطح بودن گراف شبه مثُلث‌بندی برای ایجاد کل شبه مثُلث‌ها حداقل  $O(n)$  زمان صرف می‌شود، لذا در مجموع مرتبه زمانی این الگوریتم‌ها ( $O(nlogn)$  می‌باشد).

### ۶. نتیجه‌گیری

در این مقاله روش‌های جدیدی جهت شبه مثُلث‌بندی مجموعه نقاط  $S$  در صفحه مطرح شد. روند کار به گونه‌ای بود که ابتدا برای مجموعه نقاط  $S$  در صفحه پوسته‌های محدب لایه‌ای ایجاد شده و سپس با اعمال یکی از روش‌ها و یا در مواردی ترکیبی از آن آنها بر روی لایه‌ها، شبه مثُلث‌بندی انجام گردید. بررسی‌ها نشان داد که دو روش از شبه مثُلث‌بندی‌های انجام شده کمینه بوده، به این معنی که تعداد شبه مثُلث‌های تولید شده  $2 - n$  شبه مثُلث و تعداد یال‌های مورد نیاز در آن کمترین مقدار ممکن یعنی  $3 - 2n$  می‌باشد. همچنین در این مقاله الگوریتمی جدید برای ایجاد یک چندضلعی ساده حلزونی شبکه مثُلث‌بندی شده از مجموعه نقاط تصادفی  $S$  در صفحه ارائه شد. که با استفاده از این الگوریتم پیشنهادی به ترتیب از بیرونی ترین لایه محدب تا درونی ترین لایه با ایجاد شبه مثُلث‌های پی دری، چندضلعی حلزونی شبه مثُلث‌بندی شده ایجاد گردید. ایجاد چندضلعی‌های ساده از روی مجموعه نقاط تصادفی دارای کاربردهایی نظیر بررسی الگوریتم‌های ابتکاری در

