یک الگوریتم جدید برای نگهبانی سطوح نامنظم مثلث‌بندی شده

علیرضا زارعی   محمد قدسی

دانشکده‌ی مهندسی کامپیوتر
دانشگاه صنعتی شریف

چکیده

در این مقاله یک الگوریتم جدید برای پوشش سطحی یک ناحیه‌ی مثلث‌بندی شده ارائه شده است که هنگام اجرای الگوریتم از دید سراسری رئوس استفاده می‌شود. در این الگوریتم ابتدا ناحیه‌ی مثلث‌بندی شده با حذف کردن تعدادی از رئوس آن به مجموعه‌ای از چندضلعی‌های ساده تبدیل می‌شود و به هر کدام از رئوس حذف شده یک نگهبان اختصاص می‌یابد. سپس مجموعه‌ی رئوس لازم برای نگهبانی مجموعه‌ی چندضلعی‌های ساده که ناحیه‌ی بیرونی آن‌ها پیوسته است تعیین می‌شود. پس از ارائه‌ی الگوریتم و اثبات درستی آن، نشان می‌دهیم که این الگوریتم در زمان خطی نسبت به تعداد رئوس ناحیه‌ی مثلث‌بندی شده اجرا می‌شود و حد بالای تعداد نگهبان‌های انتخاب شده توسط آن $\lfloor \frac{2n}{3} \rfloor$ می‌باشد. در عین حال اثبات شده است که با این الگوریتم حد بالای تعداد نگهبان‌ها در حالت متوسط $\lfloor \frac{n}{2} \rfloor$ است. همچنین با توجه به ویژگی‌های الگوریتم، برای بهبود کارایی آن و استفاده از دید سراسری رئوس می‌توان با صرف هزینه‌ی زمانی بیشتری برای پیش پردازش تعداد نگهبان‌های انتخاب شده را کاهش داد.

کلیدواژه‌ها: مسائل NP—تمام، مساله‌ی موزه هنر، مجموعه نگهبان، سطوح نامنظم مثلث‌بندی شده، الگوریتم تقریبی.

۱ مقدمه

مساله‌ی موزه هنر برای نخستین بار در سال ۱۹۷۳ توسط ویکتورکلی[1] به این ترتیب بیان شد که«کمترین تعداد نگهبان‌های لازم برای نگهبانی یک موزه هنر چند تاست؟». از این زمان به بعد گونه‌های مختلف این مساله مورد بررسی محققین قرار گرفت. گونه‌ی سه بعدی این مساله تعیین تعداد کمینه نگهبان‌ها برای پوشش یک منطقه‌ی جغرافیایی است. برای مدل‌سازی این سطوح روش‌های مختلفی وجود دارد که یکی از آن‌ها تقریب سطح ناحیه با مجموعه‌ای از مثلث‌های به‌هم چسبیده است. در چنین ناحیه‌ای هر نقطه از سطح به وسیله‌ی سه مؤلفه‌ی طول، عرض و ارتفاع آن مشخص می‌شود و هیچ دو نقطه‌ی متمایزی از آن دارای طول و عرض یکسان نمی‌باشند. این سطوح به سطوح نامنظم مثلث‌بندی شده یا TIN[2] معروف می‌باشند.

در مساله‌ی نگهبانی رأسی برای سطوح نامنظم مثلث‌بندی شده هدف انتخاب کوچکترین زیرمجموعه از رأس‌های ناحیه است که هر نقطه از آن ناحیه از حداقل یکی از آن رئوس قابل دید باشد. یک نقطه توسط نقطه‌ی دیگری از ناحیه قابل دید است هرگاه پاره‌خط متصل کننده‌ی آن‌ها در هیچ نقطه‌ی دیگری آن ناحیه را قطع نکند و بالای ناحیه قرار بگیرد.

این مساله در حالت کلی NP—سخت است و با کاهش مساله SAT به آن اثبات شده‌است[۴]. بنابراین در تمام الگوریتم‌های ارائه شده‌ی فعلی تلاش شده‌است تا در زمان اجرای مناسب، تقریب کوچک‌تر و نزدیک‌تری به جواب بهینه بدست آید. ازطرفی چون محاسبه‌ی سطوح قابل دید از یک نقطه با توجه به وجود پستی و بلندی در این ناحیه‌ها مستلزم پردازش پیچیده و زمان‌بری است، در بیشتر الگوریتم‌های ارائه شده‌ی فعلی گونه‌ی ساده‌تری از مساله مورد توجه قرار گرفته است که در آن سطوح قابل دید از یک رأس فقط شامل آن‌هایی است که آن رأس یکی از رئوس آن باشد. با این ساده‌سازی، مساله تبدیل به یک مساله‌ی دو بعدی می‌شود که در آن ناحیه‌ی مثلث‌بندی شده بوسیله‌ی یک گراف مسطح مدل می‌شود و برای آن مساله‌ی تعیین نگهبان‌ها حل می‌شود.

---
[1] Victor Klee

[2] Triangulated Irregular Network

در مورد مساله‌ی اخیر با استفاده از قضیه‌ی چهار رنگ‌آمیزی گراف‌ها اثبات می‌شود که همواره $\lfloor \frac{n}{2} \rfloor$ نگهبان کافی است و برای برخی از TINها همانند شکل ۱ این تعداد نگهبان ضروری است[۱].

الگوریتم‌های تقریبی ارائه شده برای این مساله الگوریتم‌هایی با مرتبه‌ی زمانی $O(n^2)$ [۲] و $O(n^{\frac{3}{2}})$ [۳] برای قرار دادن حداکثر $\lfloor \frac{n}{2} \rfloor$ نگهبان رأسی می‌باشند که اولی با استفاده از چهاررنگ‌آمیزی گراف مسطح و انتخاب دو رنگ از چهار رنگ استفاده شده انجام می‌شود و دومی براساس یافتن تطابق بیشینه در گراف دوگان مربوط به گراف مسطح می‌باشد. همچنین بخاطر پیچیدگی بالای زمانی و الگوریتمی این روش‌ها الگوریتم‌های ساده‌تری با زمان اجرای کمتر ارائه شده‌است که نگهبان‌های بیشتری را نسبت به حد بالای $\lfloor \frac{n}{2} \rfloor$ بدست می‌آورند. از جمله الگوریتم با زمان اجرای خطی که حداکثر $\lfloor \frac{3n}{5} \rfloor$ نگهبان رأسی را انتخاب می‌کند[۱] و براساس ۵رنگ‌آمیزی گراف مسطح و انتخاب ۳ رنگ از ۵ رنگ بکار رفته استوار است.

در این مقاله یک روش تقریبی جدید برای حل این مساله ارائه می‌شود که برای یک TIN با n رأس قابل اجرا در زمان $O(n)$ می‌باشد و حداکثر تعداد نگهبان‌های انتخاب شده توسط آن $\lfloor \frac{2n}{3} \rfloor$ می‌باشد. مهم‌ترین ویژگی این الگوریتم که آن‌را از گونه‌های ارائه شده‌ی قبلی متمایز می‌کند قابلیت استفاده از دید سراسری رئوس در فرایند افزایشی تخصیص نگهبان‌های ناحیه است. در این روش در صورت تعیین سطوح قابل دید از هر رأس با انجام یک پیش پردازش که با هزینه‌ی $O(n^3)$ قابل انجام است، می‌توان هنگام تخصیص نگهبان‌های ناحیه از این اطلاعات استفاده کرد و تعداد نگهبان‌های مورد نیاز را کم کرد. با این حال در تحلیل حالت متوسط این الگوریتم نشان داده شده است که حتی بدون پیش پردازش و فقط با انجام بررسی‌های بیشتری هنگام تخصیص نگهبان‌ها که باعث افزایش مرتبه‌ی زمانی الگوریتم نیز نمی‌شود، حد بالای تعداد نگهبان‌های انتخاب شده در حالت متوسط $\lfloor \frac{n}{2} \rfloor$ است.

نکته‌ای که هست این است که معمولا تعداد نگهبان‌های مورد نیاز بسیار کمتر از $\lfloor \frac{n}{2} \rfloor$ است و به کمک این الگوریتم می‌توان برای TINهای واقعی به نتایج بهتری نسبت به سایر روش‌ها رسید.

در ادامه‌ی این مقاله در بخش ۲ مساله‌ی نگهبانی مجموعه‌ای از چندضلعی‌های بدون سوراخ بررسی می‌شود و در بخش ۳ روشی برای کاهش یک TIN به مجموعه‌ای از چندضلعی‌های بدون سوراخ ارائه می‌شود و در بخش ۴ الگوریتم جدید و اثبات صحت و تحلیل کارایی آن می‌آید. در بخش ۵ تحلیل حالت متوسط الگوریتم بیان خواهد شد. در آخر نیز مطالب گفته شده نتیجه‌گیری و خلاصه می‌شوند.

۲ نگهبانی رأسی در چندضلعی ساده

هر n ضلعی ساده‌ی مثلث‌بندی شده را می‌توان بوسیله‌ی حداکثر $\lfloor \frac{n}{3} \rfloor$ نگهبان رأسی نگهبانی کرد و برای آن یک الگوریتم خطی وجود دارد[٦]. بدیهی است که همین الگوریتم برای چندضلعی‌های مثلث‌بندی شده‌ی P_1، P_2، $\cdots$ ، P_k که بصورت مجزا از هم قرار دارند قابل اعمال است. همچنین اگر چندضلعی‌های P_i در برخی از نقاط مرزی با هم اشتراک داشته باشند، به‌عبارتی دیگر اگر چندضلعی‌های P_i دارای رأس مشترک باشند، اثبات می‌شود که مشروط بر این‌که وجود رأس مشترک بین چندضلعی‌ها باعث بوجود آمدن سوراخ در صفحه نشود همچنان با حداکثر $\lfloor \frac{n}{3} \rfloor$ نگهبان می‌توان چندضلعی‌ها را نگهبانی کرد و این کار در مرتبه‌ی زمانی $O(n)$ انجام‌پذیر است که در این حالت n تعداد رأس‌های کل چندضلعی‌هاست که رئوس مشترک آن‌ها یک‌بار شمرده شده‌اند.

برای اثبات این مطلب مطابق شکل ۲، می‌توان در محل رأس مشترک بین دو چندضلعی با اضافه کردن یک یال آن دو چند ضلعی را به یک چندضلعی واحد با همان تعداد رأس تبدیل کرد. بنابراین مجموعه‌ی چندضلعی‌های P_1، P_2، $\cdots$ ، P_k که هر P_i با P_{i+1} فقط یک رأس مشترک دارد و P_k با P_1 دارای رأس مشترک نیستند، قابل تبدیل به یک چندضلعی است که دارای همان تعداد رأس است و نگهبانی تمام سطوح آن به معنای نگهبانی تمام سطوح چندضلعی‌های P_1 تا P_k می‌باشد.

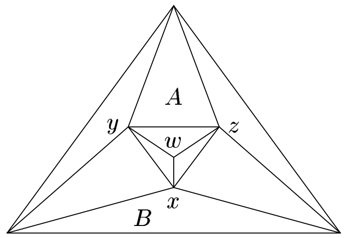
بنابراین می‌توان قضیه‌ی زیر را بیان نمود:

قضیه ۱. هرگاه هر دو چندضلعی از مجموعه‌ی چندضلعی‌های P_1، P_2، $\cdots$ ، P_k که در یک صفحه واقعند دارای حداکثر یک رأس مشترک باشند و ناحیه‌ی بیرون چندضلعی‌ها پیوسته باشد، این چندضلعی‌ها را می‌توان در مرتبه‌ی زمانی $O(n)$ و با حداکثر $\lfloor \frac{n}{3} \rfloor$ نگهبان رأسی، نگهبانی کرد که n تعداد رئوس تمام چندضلعی‌هاست که رئوس مشترک بین چندضلعی‌ها یک‌بار شمرده شده‌اند.

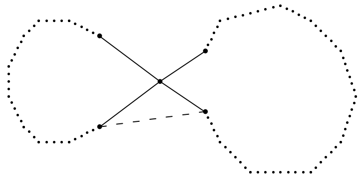
۳ کاهش تین به چندضلعی ساده

ایده‌ی اصلی الگوریتم ارائه شده در این مقاله این است که با انتخاب تعدادی از رئوس TIN به عنوان نگهبان و حذف مثلث‌های قابل دید از آن‌ها، آن را به یک TIN ساده‌تر تبدیل کرد. در این بخش روشی برای تقلیل TIN به چندضلعی‌های ساده‌ی بدون سوراخ ارائه می‌شود که در این روش تا جایی که نیاز است از رئوس TIN حذف می‌شود تا در پایان کار تبدیل به یک یا چند چندضلعی شود که تمام ناحیه‌ی بیرون چندضلعی‌ها پیوسته باشد.

در هنگام انجام عمل کاهش TIN باید همواره دو نکته‌ی زیر برقرار باشد:



شکل ۱: یک TIN با $\lfloor \frac{n}{2} \rfloor$ نگهبان رأسی مورد نیاز



شکل ۲: توسعه‌ی دو چندضلعی مجاور به یک چندضلعی بزرگ‌تر

۱) حذف کردن یک رأس باعث بوجود آمدن سوراخ در صفحه نشود.

۲) بجز در مورد دو نوع TIN که در شکل ۳ نشان داده شده‌اند و از این به بعد به نام‌های TIN نوع «الف» و TIN نوع «ب» خوانده می‌شوند، در مورد بقیه‌ی TINها هنگام حذف یک رأس حداقل چهار مثلث مجاور آن حذف شود.

از آن‌جا که هنگام حذف کردن یک رأس تمام مثلث‌های مجاور آن حذف می‌شوند تمام ناحیه‌ی حذف شده پیوسته است بنابراین اگر هنگام حذف کردن یک رأس، یک رأس مرزی یا یک یال مرزی از TIN حذف شود ناحیه‌ی حذف شده‌ی فعلی با ناحیه‌ی بیرون TIN پیوسته شده و یک ناحیه را تشکیل می‌دهند و در صفحه سوراخ بوجود نخواهد آمد و مورد ۱ برقرار خواهد ماند.

در مورد نکته‌ی دوم نیز چنانچه در شکل ۳ مشخص است در هیچ‌کدام از این TINها حذف کردن یک رأس باعث حذف شدن چهار مثلث نمی‌شود. نشان می‌دهیم که در مورد سایر TINها، یا TIN مورد نظر یک چندضلعی ساده است یا این‌که می‌توان رأسی را پیدا کرد که با حذف کردن آن حداقل چهار مثلث از TIN حذف می‌شود و علاوه بر آن یا خود این رأس روی مرز قرار دارد یا حذف کردن آن باعث حذف شدن یک یال مرزی می‌شود تا در مراحل کاهش مورد ۲ نیز حفظ شده باشد.

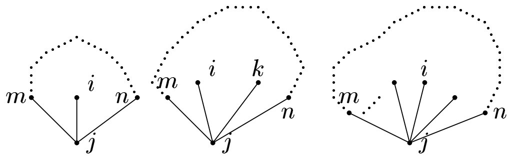
در یک TIN که چندضلعی ساده نباشد حداقل یک رأس داخلی i وجود دارد که روی مرز TIN قرار ندارد. اگر j یک رأس مرزی باشد که بین i و j یال وجود دارد در مورد رأس j بر اساس درجه‌ی آن سه حالت وجود دارد که در شکل ۴ نشان داده شده‌اند.

- در حالت «الف» رأس j دارای حداقل چهار مثلث مجاور خود است که با حذف شدن آن از TIN حذف می‌شوند.
- در حالت «ب» براحتی قابل بررسی است که اگر هرکدام از رأس‌های i و k دارای حداکثر سه مثلث مجاور خود باشند آن‌گاه TIN معادل TIN نشان داده شده در شکل ۳ قسمت «ب» خواهد بود و در غیر این‌صورت حداقل یکی از این رأس‌ها(مثلا k) دارای حداقل چهار مثلث مجاور خود است و با حذف شدن آن، یک یال مرزی (مثلا jn) از TIN حذف می‌شود. پس ناحیه‌ی حذف شده با ناحیه‌ی بیرون TIN پیوسته می‌شود.
- در حالت «پ» نیز براحتی قابل بررسی است که چنانچه هیچ‌کدام از رئوس i، n و m دارای بیشتر از سه مثلث مجاور خود نباشند TIN معادل یکی از TINهای نشان داده شده در شکل ۳ خواهد بود و در غیر این‌صورت حداقل یکی از این سه رأس دارای حداقل چهار مثلث مجاور خود می‌باشد و با حذف شدن آن، یک رأس مرزی از TIN نیز حذف می‌شود و در نتیجه ناحیه‌ی حذف شده با ناحیه‌ی بیرون TIN پیوسته می‌شود.

با توجه به مطالب گفته شده می‌توان قضیه‌ی زیر را در مورد کاهش TIN بیان نمود:

قضیه ۲. هر TIN را می‌توان با حذف کردن تعدادی از رئوس آن به مجموعه‌ای از چندضلعی‌های ساده و یا TINهای از نوع «الف» و «ب» تبدیل کرد که در این کاهش تعداد مثلث‌های حذف شده از TIN حداقل چهار برابر تعداد رئوس حذف شده می‌باشد و ناحیه‌ی بیرونی چندضلعی‌ها و TINهای باقی‌مانده پیوسته است.

شکل ۳: TINهای خاص که کاهش آن‌ها باعث حذف بیشتر از سه مثلث از آنها نمی‌شود



شکل ۴: حالات مختلف رأس داخلی i و رأس مرزی j مجاور آن

۴ نگهبانی سطوح نامنظم مثلث‌بندی شده

در این قسمت و با استفاده از مطالب گفته شده در بخش‌های ۲ و ۳ الگوریتمی برای نگهبانی یک TIN دلخواه با n رأس ارائه می‌شود که حد بالای تعداد نگهبان‌های انتخاب شده توسط آن $\lfloor \frac{2n}{3} \rfloor$ و متوسط تعداد نگهبان‌های انتخاب شده $\lfloor \frac{n}{2} \rfloor$ می‌باشد.

این الگوریتم شامل سه مرحله است: در مرحله‌ی اول آن با حذف کردن تعدادی از رئوس TIN مطابق بخش ۳، TIN اولیه به مجموعه‌ای از چندضلعی‌های ساده و TINهای نوع «الف» و «ب» تبدیل می‌شود و به هر کدام از رئوس حذف شده یک نگهبان تخصیص می‌یابد. عمل حذف کردن تا جایی ادامه پیدامی‌کند که هیچ رأس باقی‌مانده‌ای دارای بیش از سه مثلث مجاور خود نباشد.

در مرحله‌ی دوم الگوریتم به هر کدام از رئوس مرکزی TINهای نوع «الف» یک نگهبان تخصیص می‌یابد و مثلث‌های مجاور آن حذف می‌شوند. همچنین در مورد هرکدام از TINهای نوع «ب» یک نگهبان به یکی از رئوس مرزی آن که دارای سه مثلث در مجاورت خود می‌باشد تخصیص می‌یابد. باید این نکته را در نظر داشت که رأسی که هیچ مثلثی در مجاورت آن نباشد که نیاز به نگهبانی داشته باشد خود به خود از مجموعه‌ی رئوس حذف می‌شود.

برای افزایش کارایی الگوریتم در این دو مرحله هنگام انتخاب یک رأس به عنوان نگهبان، تمام مثلث‌هایی که بر اثر حذف آن رأس در مرز ناحیه قرار می‌گیرند بررسی می‌شوند و در صورتی‌که از آن رأس قابل دید باشند از ناحیه‌ی نگهبانی نشده حذف می‌شوند و این کار تا زمانی که تمام مثلث‌هایی که با انتخاب نگهبان جدید در مرز ناحیه قرار گرفته‌اند، از آن رأس قابل دید نباشند تکرار می‌شود. با این روش در حالت متوسط حد بالای تعداد نگهبان‌ها $\lfloor \frac{n}{2} \rfloor$ خواهد شد.

با انجام این عملیات فقط مجموعه‌ای از چندضلعی‌های ساده باقی می‌مانند که در شرط قضیه ۱ صدق می‌کنند که به عنوان مرحله‌ی سوم الگوریتم به کمک همان روش نگهبان‌های این مجموعه چندضلعی‌ها تعیین می‌شود. بنابراین اگر تعداد رئوس این مجموعه چندضلعی‌ها m باشد با حداکثر $\lfloor \frac{m}{3} \rfloor$ نگهبان و در مرتبه‌ی زمانی $O(m)$ می‌توان نگهبان‌های آن‌ها را مشخص کرد.

بدین ترتیب و با انجام این سه مرحله کل TIN نگهبانی خواهد شد. در ادامه‌ی این بخش و در زیربخش‌های بعدی صحت الگوریتم ارائه شده بررسی می‌شود و از نظر تعداد نگهبان‌های انتخاب شده و زمان اجرا تحلیل خواهد شد.

۱.۴ بررسی درستی الگوریتم گفته شده

از آنجا که در مرحله‌ی کاهش فقط مثلث‌هایی از TIN حذف می‌شوند که مجاور یک رأس حذف شده باشند و چون به تمام رأس‌های حذف شده در این مرحله نگهبان اختصاص می‌یابد پس تمام این سطوح نگهبانی می‌شوند. پس از مرحله‌ی کاهش نیز به هر کدام از TINهای نوع «الف» یک نگهبان تخصیص می‌یابد که تمام آن‌را نگهبانی می‌کند و به TINهای نوع «ب» نیز یک نگهبان اختصاص می‌یابد که فقط یک مثلث از آن را نگهبانی نمی‌کند و این مثلث به مجموعه‌ی چندضلعی‌ها اضافه می‌شود.

در بخش ۲ نیز روش نگهبانی این چندضلعی‌های باقی‌مانده از TIN گفته شد و با بکارگیری این روش تمام سطوح TIN تحت پوشش مجموعه‌ی نگهبان‌های انتخاب شده قرار می‌گیرند.

بنابراین قضیه زیر در مورد این الگوریتم برقرار است.

قضیه ۳. مجموعه‌ی نگهبان‌های انتخاب شده توسط الگوریتم

گفته شده برای نگهبانی یک TIN تمام سطوح آن را نگهبانی می‌کند.

۲.۴ حد بالای تعداد نگهبان‌ها

در این قسمت میزان کارایی الگوریتم گفته شده از نظر حد بالای تعداد نگهبان‌های انتخاب شده توسط آن ارزیابی می‌شود. ابتدا دو نکته را در قالب دو لم ساده بیان می‌کنیم:

لم ۱. پس از مرحله‌ی کاهش هیچ دو TIN نوع «الف» دارای رأس مشترک نیستند.

اثبات. زیرا در غیراین‌صورت آن رأس مشترک دارای بیش از سه مثلث در مجاورت خود خواهد بود و در مرحله‌ی کاهش با تخصیص یک نگهبان به این رأس از TIN حذف می‌شود. □

لم ۲. چنانچه پس از مرحله‌ی کاهش یک TIN نوع «الف» با یک TIN نوع «ب» دارای رأس مشترک باشد در مرحله‌ی تخصیص نگهبان به TINهای نوع «الف» و «ب» تمام مثلث‌های مجاور این رأس نگهبانی می‌شوند و در نتیجه آن رأس از TIN حذف می‌شود.

اثبات. با توجه به نحوه‌ی تخصیص نگهبان به TINهای نوع «ب» و محدودیت تعداد مثلث‌های رأس‌های مرزی در TINهای نوع «الف» اثبات آن بدیهی است. □

با در نظر گرفتن این دو نکته به وضعیت رأس‌های مرزی TINهای نوع «الف» پس از تخصیص نگهبان به آن‌ها می‌پردازیم. اگر k تعداد TINهای نوع «الف» باشد و i تعداد مثلث‌های باقی‌مانده از TIN اولیه پس از تخصیص نگهبان به TINهای نوع «الف» و «ب» باشد که حداقل یکی از رئوس آن‌ها جزء رئوس مرزی یک TIN نوع «الف» بوده است، اگر $i \leq k$ باشد همواره لم زیر برقرار است:

لم ۳. تعداد رأس‌های مرزی در TINهای نوع «الف» که بعد از تخصیص نگهبان به TINهای نوع «الف» و «ب» دارای هیچ مثلثی در مجاورت خود نمی‌باشند بزرگ‌تر یا مساوی با $3k - 3i$ است و این رأس‌ها از TIN حذف می‌شوند.

اثبات. با توجه به این‌که هر TIN نوع «الف» دارای سه رأس مرزی است و این TINها دارای هیچ رأس مشترکی نمی‌باشند پس تعداد این رئوس $3k$ می‌باشد و از طرفی هر مثلث از مجموعه‌ی i مثلث حداکثر با سه رأس از این رأس‌ها مجاورت دارد. پس حداقل $3k - 3i$ رأس باقی‌مانده مجاور با هیچ مثلث دیگری نمی‌باشند و می‌توان آنها را بدون تخصیص نگهبان از مجموعه‌ی رئوس حذف کرد. □

با استفاده از این مقدمات می‌توان قضیه زیر را در مورد حد بالای تعداد نگهبان‌ها در الگوریتم گفته شده بیان کرد:

قضیه ۴. حد بالای تعداد نگهبان‌ها در الگوریتم گفته شده برای یک TIN با n رأس $\lfloor \frac{2n}{3} \rfloor$ می‌باشد.

اثبات. فرض می‌کنیم:

m : تعداد رئوس حذف شده در مرحله‌ی کاهش الگوریتم،

j : تعداد TINهای نوع «ب» پس از مرحله‌ی کاهش الگوریتم،

k : تعداد TINهای نوع «الف» پس از مرحله‌ی کاهش الگوریتم و

i : تعداد مثلث‌های باقی‌مانده از TIN اولیه پس از تخصیص نگهبان به TINهای نوع «الف» و «ب» باشد که حداقل یک رأس از آن‌ها جزء رئوس مرزی یک TIN نوع «الف» بوده است.

در هر TIN با n رأس و t رأس مرزی $2n - t - 2$ مثلث وجود دارد[5] و در این الگوریتم به ازای هر کدام از رئوس m حداقل چهار مثلث از تعداد مثلث‌های TIN کم می‌شود و هر کدام از TINهای نوع «الف» دارای سه مثلث و هرکدام از TINهای نوع «ب» دارای چهار مثلث می‌باشند و هیچ‌کدام از این مثلث‌ها با هم مشترک نمی‌باشند و همچنین با مجموعه‌ی مثلث‌های i نیز مشترک نیستند. پس می‌توان نامساوی زیر را نوشت:

$$\begin{aligned} 4m + 4j + 3k + i &\leq 2n - t - 2 \\ \Rightarrow 4m + 4j + 3k + i &\leq 2n \end{aligned} \qquad (1)$$

حال قضیه را در دو حالت بررسی می‌کنیم:

۱) $i \leq k$: در این حالت طبق لم ۳ حداقل $3k - 3i$ رأس بدون تخصیص نگهبان از مجموعه‌ی رئوس حذف می‌شوند. بنابراین کل تعداد نگهبان‌های تخصیص داده شده از رابطه‌ی زیر بدست می‌آید:

$$\text{تعداد نگهبان‌ها} \leq m + j + k + \lfloor \tfrac{n-(m+j+k)-(3k-3i)}{3} \rfloor \qquad (2)$$

و از طرفی از رابطه‌ی (۱) می‌توان نتیجه گرفت که $m + j \leq \frac{2n-3k-i}{4}$ و با جای‌گذاری در رابطه‌ی (۲) رابطه‌ی زیر بدست می‌آید:

$$\begin{aligned} \text{تعداد نگهبان‌ها} &\leq \tfrac{2n-3k-i}{4} + k + \lfloor \tfrac{n-(\frac{2n-3k-i}{4}+k)-(3k-3i)}{3} \rfloor \\ \Rightarrow &\leq \tfrac{6n-9n-3i+12k+4n-2n+3k+i-16k+12i}{12} \\ \Rightarrow &\leq \tfrac{8n-10k+10i}{12} \\ \Rightarrow &\leq \tfrac{2n}{3} + \tfrac{5}{6}(i-k) \end{aligned}$$

و با توجه به این‌که طبق فرض $i \leq k$ است پس تعداد نگهبان‌ها حداکثر $\lfloor \frac{2n}{3} \rfloor$ خواهد بود.

۲) $i > k$: در این حالت می‌توان رابطه‌ی زیر را در مورد تعداد نگهبان‌ها نوشت:

$$\text{تعداد نگهبان‌ها} \leq m + j + k + \lfloor \tfrac{n-(m+j+k)}{3} \rfloor \qquad (3)$$

و از طرفی از رابطه‌ی (۱) و با توجه به این‌که طبق فرض $i > k$ است نتیجه می‌شود که $m + j + k \leq \frac{۲n}{۴}$ و با جای‌گذاری در رابطه‌ی (۳) رابطه‌ی زیر بدست می‌آید:

$$\begin{aligned} \text{تعداد نگهبان‌ها} &\leq \frac{۲n}{۴} + \lfloor \frac{n - (\frac{۲n}{۴})}{۳} \rfloor \\ \Rightarrow &\leq \frac{۶n + ۴n - ۲n}{۱۲} \\ \Rightarrow &\leq \frac{۲n}{۳} \end{aligned}$$

بنابراین در این حالت نیز قضیه برقرار است.

□

۳.۴ تحلیل زمان اجرای الگوریتم

در این‌جا فرض می‌کنیم که ساختمان داده‌ی استفاده شده برای نگهداری TIN یک ساختار DCEL[۳] است که مشخصات دقیق آن در [۵] آمده است. در این ساختار پیمایش تمام یال‌های محدود کننده‌ی وجوه در زمان خطی امکان‌پذیر است.

در مرحله‌ی کاهش الگوریتم و با توجه به این‌که هر رأس فقط یک‌بار نیازمند پیمایش و بررسی است و نیز این‌که حذف یک رأس و بازسازی ساختار DCEL برای هر رأس و یال فقط یک‌بار اتفاق می‌افتد، پس کل زمان اجرای مرحله‌ی کاهش الگوریتم نسبت خطی با تعداد یال‌ها و رئوس TIN دارد. همچنین چون هر مثلث حداکثر سه بار(از هر کدام از اضلاع آن) ممکن است در مرز ناحیه قرار بگیرد، کل زمان صرف شده برای بررسی این‌که آیا مثلثی که بر اثر انتخاب یک نگهبان جدید در مرز ناحیه قرار گرفته‌است از آن رأس قابل دید است نسبت به تعداد مثلث‌ها خطی است. از طرفی تعداد یال‌ها، وجوه و رأس‌های یک TIN نیز مطابق فرمول اویلر با هم نسبت خطی دارند[۵]. پس زمان اجرای این مرحله نسبت به تعداد رئوس TIN خطی است.

مرحله‌ی دوم الگوریتم یعنی شناسایی و تخصیص نگهبان به TINهای نوع «الف» و «ب» نیز با توجه به وجود ساختار DCEL و خطی بودن تعداد اجزاء در زمان خطی نسبت به تعداد رئوس آن‌ها انجام‌پذیر است. همچنین مرحله‌ی آخر الگوریتم یعنی نگهبانی چندضلعی‌های ساده‌ی باقی‌مانده مطابق قضیه‌ی ۱ در زمان خطی انجام می‌گیرد.

بنابراین در مورد زمان اجرای این الگوریتم قضیه زیر قابل بیان است:

قضیه ۵. الگوریتم ارائه شده قابل اجرا در زمان خطی نسبت به تعداد رئوس TIN می‌باشد.

۵ تحلیل حالت متوسط الگوریتم

همان‌طوری که گفته شد پس از پایان مرحله اول قسمت‌های نگهبانی نشده و باقی‌مانده از ناحیه‌ی اولیه به‌صورت ترکیبات مختلفی از چندضلعی‌های ساده و TINهای نوع «الف» و «ب» خواهند بود. ابتدا نشان می‌دهیم که اگر تعداد رأس‌های این ناحیه‌های باقی‌مانده k باشد در حالت متوسط حداکثر $\lfloor \frac{n}{۳} \rfloor$ نگهبان نیاز خواهند داشت. در مورد چندضلعی‌های ساده این مطلب در بخش ۲ نشان داده شده است. پس صحت این مطلب فقط در مورد TINهای نوع «الف» و «ب» باید نشان داده شود.

لازم است توجه شود که در تحلیل حالت متوسطی که در ادامه می‌آید احتمال دیدن یک مثلث غیر مجاور و همسایه‌ی مرتبه‌ی دوم یک رأس[۴] از آن رأس با احتمال ندیدن آن برابر فرض شده است که فقط به ارتفاع رئوس بستگی خواهد داشت و در حالت کلی چنین فرضی دور از واقعیت نخواهد بود.

هنگام تخصیص یک نگهبان به رأس مرکزی یک TIN نوع «الف»، مطابق شکل ۵ قسمت «الف»، هرکدام از رأس‌های مرزی آن که عضو هیچ ناحیه‌ی باقی‌مانده‌ای نباشد، یا مثلث متصل به آن توسط نگهبان تخصیص داده شده قابل دید باشد آن رأس دیگر نیازی به نگهبان ندارد زیرا هیچ مثلث نگهبانی نشده‌ای مجاور آن قرار ندارد. پس امید ریاضی تعداد رأس‌هایی که با تخصیص این نگهبان از مجموعه‌ی رئوس باقی‌مانده‌ی ناحیه‌ی اصلی حذف می‌شوند برابر با $۳(\frac{۱}{۲} + \frac{۱}{۴}) + ۱$ است که بزرگ‌تر از ۳ می‌باشد.

هنگام تخصیص نگهبان به یک رأس مرزی از یک TIN نوع «ب»، مطابق شکل ۵ قسمت «ب»، در صورتی که این نگهبان(a) تنها مثلث باقی‌مانده از TIN یعنی A را ببیند دو رأس b و c از مجموعه‌ی رئوس باقی‌مانده قابل حذف می‌باشند. همچنین در صورتی که مثلث‌های E و F وجود نداشته باشند یا از نقطه‌ی a قابل دید باشند رئوس e و f قابل حذف می‌باشند. پس امید ریاضی تعداد رأس‌های حذف شدنی در این حالت برابر با $۲(\frac{۱}{۲} + \frac{۱}{۴}) + ۲(\frac{۱}{۲}) + ۱$ است که این مقدار نیز بزرگ‌تر از ۳ است.

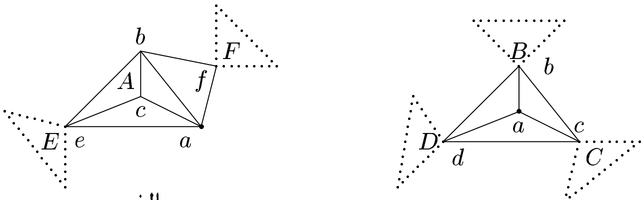
پس برای نگهبانی ناحیه‌های باقی‌مانده پس از مرحله‌ی اول الگوریتم در حالت متوسط به $\lfloor \frac{k}{۳} \rfloor$ نگهبان نیاز خواهد بود که k تعداد رئوس ناحیه‌های باقی‌مانده می‌باشد.

حال نشان داده می‌شود که در حالت متوسط امید ریاضی تعداد مثلث‌های پوشش داده شده توسط هر نگهبان انتخاب شده در مرحله اول الگوریتم و هنگام تخصیص نگهبان به رئوس با درجه‌ی بیشتر از ۳ بیشتر از ٦ می‌باشد.

مطابق فرمول اویلر متوسط درجه‌ی هر رأس ٦ می‌باشد. بنابراین مطابق شکل ٦، اگر رأس i که در این مرحله به‌عنوان نگهبان انتخاب شده است را در نظر بگیریم بر حسب اینکه قبل از آن یکی از رئوس مجاور آن مانند j به عنوان نگهبان انتخاب شده باشد یا نه دو حالت بوجود می‌آید که درحالت اول ۴ مثلث در مجاورت i تحت پوشش آن قرار می‌گیرند و در حالت دوم این تعداد ٦ می‌باشد.

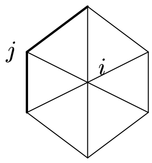
[۳] Doubly-Connected Edge List

[۴] مثلث‌های همسایه مرتبه دوم یک رأس آنهایی هستند که دارای یال مشترک با حداقل یکی از مثلث‌های مجاور آن رأس باشند.



الف الف

شکل ۵: تحلیل TINهای باقی‌مانده پس از مرحله اول الگوریتم



شکل ٦: وضعیت رأس i در حالت متوسط

از طرفی با در نظر گرفتن ارتفاع رئوس، ممکن است یک رأس مثلث‌های غیر مجاور خود را نیز پوشش دهد. برای تحلیل این وضعیت حالت‌های مختلفی برای رأس i و مثلث‌های مجاور آن وجود دارد که آنچه که برای محاسبات این مقاله لازم بوده است این است که اثبات شود در حالت متوسط حداقل دو مثلث غیرمجاور و همسایه‌ی مرتبه‌ی دوم i وجود دارد که در حالت متوسط و بصورت احتمالی، i یکی از آنها را می‌بیند و این مساله با در نظر گرفتن حالات مختلف مثلث‌های همسایه مرتبه دوم نشان داده شده است: فقط در یک حالت i هیچ مثلث مجاور مرتبه‌ی دوم ندارد و در مقابل در یک حالت نیز ۴ مثلث همسایه مرتبه دوم خواهد داشت که در شکل ۷ نشان داده شده‌اند. همچنین به‌ازای هرکدام از حالاتی که i فقط دارای یک مثلث مجاور مرتبه دوم باشد یک حالت وجود دارد که i دارای ۳ مثلث مجاور مرتبه‌ی دوم است و دو نمونه از آن‌ها در شکل ۷ نشان داده شده است.

پس در حالت متوسط رأس i دارای حداقل دو مثلث مجاور مرتبه دوم می‌باشد که بصورت احتمالی و با این فرض که احتمال دیدن و ندیدن یک مثلث مجاور مرتبه دوم برابر باشد امید ریاضی تعداد مثلث‌های مجاور مرتبه دوم قابل دید توسط رأس i، یک خواهد بود و همان‌طوری‌که گفته شد برحسب اینکه قبل از رأس i، رأس j به عنوان نگهبان انتخاب شده باشد یا نه، تعداد کل مثلث‌های قابل پوشش توسط i، ۵ یا ۷ خواهد بود و با فرض مساوی بودن احتمال بروز این دو حالت تعداد متوسط مثلث‌های پوشش داده شده توسط یک نگهبان ٦ خواهد بود.

اگر m تعداد نگهبان‌های انتخاب شده در مرحله اول و t تعداد مثلث‌های باقی‌مانده برای مرحله دوم الگوریتم برای یک ناحیه n رأسی باشد می‌توان روابط زیر را نوشت:

$$6m + t \leq 2n \Rightarrow m \leq \frac{2n-t}{6} \qquad (۴)$$
$$0 \leq t \leq n - m \qquad (۵)$$

از ترکیب دو رابطه‌ی فوق حد بالای مقدار m در حالت‌های کمترین و بیشترین مقدار t بصورت زیر خواهد بود:

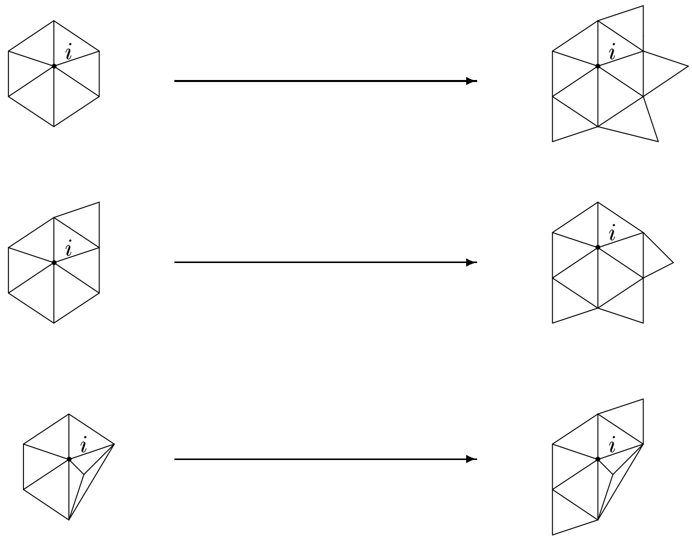
$$t = 0: \quad m \leq \frac{2n}{6} = \frac{n}{3} \qquad (٦)$$
$$t = n - m: \quad m \leq \frac{2n-(n-m)}{6} = \frac{n}{5} \qquad (۷)$$

که در مورد اول همان $\lfloor \frac{n}{3} \rfloor$ نگهبان کل ناحیه را نگهبانی می‌کنند و در حالت دوم حد بالای تعداد کل نگهبان‌های انتخاب شده از رابطه‌ی زیر بدست می‌آید:

$$\text{تعداد نگهبان‌ها} \leq m + \frac{n-m}{3}$$

که با در نظر گرفتن مقدار m از رابطه‌ی ۷، حد بالای تعداد نگهبان‌های بدست آمده $\frac{7n}{15}$ خواهد شد که در هر دو حالت متوسط تعداد نگهبان‌های مورد نیاز کوچکتر از $\lfloor \frac{n}{3} \rfloor$ می‌باشد. بنابراین انتظار می‌رود که کارایی الگوریتم در حالت متوسط بهتر و نزدیک به حد پایین تعداد نگهبان‌های مورد نیاز باشد.

در این تحلیل مثلث‌های غیر مجاور فقط تا یک مرحله‌ی بعد در نظر گرفته شده‌اند و در صورتی‌که بتوان تمام مثلث‌های غیر مجاور را مورد ارزیابی قرار داد می‌توان به نتایج بهتری رسید. همچنین برای بهبود بیشتر الگوریتم می‌توان قبل از شروع الگوریتم و به عنوان پیش پردازش به ازای هر رأس لیست مثلث‌هایی که توسط آن رأس به طور کامل پوشش داده می‌شوند تهیه کرد و در هنگام اجرای الگوریتم و هنگام انتخاب یک رأس به عنوان نگهبان تمام مثلث‌های پوشش داده شده توسط آن نشانه‌گذاری شوند تا هنگامی که در مرز ناحیه قرار می‌گیرند از ناحیه نگهبانی نشده



شکل ۷: برخی از وضعیت‌های مثلث‌های مجاور مرتبه‌ی دوم رأس i

حذف شوند. مشخص است که استفاده از این روش نیازمند صرف هزینه‌ی بیشتری برای انجام پیش‌پردازش است که باعث افزایش مرتبه‌ی زمانی الگوریتم تا $O(n^{3})$ می‌شود.

۶ جمع‌بندی

در این مقاله یک الگوریتم جدید برای نگهبانی سطوح نامنظم مثلث‌بندی شده ارائه شده است که در مرتبه‌ی زمانی $O(n)$ برای نگهبانی یک TIN با n رأس حداکثر $\lfloor \frac{2n}{3} \rfloor$ نگهبان تعیین می‌کند و در حالت متوسط تعداد نگهبان‌های انتخاب شده کمتر از $\lfloor \frac{n}{2} \rfloor$ خواهد بود. یکی از نکات مثبت این الگوریتم این است که در هنگام اجرا و در مرحله‌ی کاهش الگوریتم می‌توان از دید مجموعه‌ی نگهبان بدست آمده‌ی فعلی استفاده کرد و در صورتی‌که این مجموعه سطوحی را که در حال حاضر مورد ارزیابی قرار می‌گیرند را می‌بینند آن سطوح را نیز به عنوان نگهبانی شده علامت‌گذاری کرد و از تخصیص نگهبان دیگری به آن‌ها اجتناب کرد. و به‌این ترتیب علاوه بر دید محلی رئوس که فقط شامل سطوح مجاور آن‌هاست از دید سراسری رئوس نیز استفاده می‌شود.

مراجع

[1] P. Bose, T. Shermer, G. Toussaint, B. Zhu, *Guarding polyhedral terrains*, Computational Geometry-Theory and Applications, 7:3:173-185, 1997.

[2] N. Robertson, D. P. Sanders, P. Saymour, R. Thomas, *A new proof of the four-color theorm*, Electronic Research Announcements of the American Mathematical Society, 2:1, 1996.

[3] P. Bose, D. Kirkpatrick, Z. Li, *Efficient algorithms for guarding or illuminating the surface of a polyhedral terrain*, Proc. of the Eight Canadian Conference on Computational Geometry, 1996.

[4] R. Cole, M. Sharir, *Visibility Problems for Polyhedral Terrains*, Journal of Symbolic Computation 7(1989) PP 11-10, 1981.

[5] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, *Computational Geometry Algorithms and Applications*, Springer pp. 45-60 ,1997.

[6] B. Chazelle, *Triangulating a simple polygon in linear time*, Descrete and Computational Geometry, 6, pp. 485-524, 1991.