اعمال قید قابلیت رؤیت بر مسألهٔ کوتاهترین فاصلهٔ پیوندی

Visibility Constraint on Minimum Link-Distance Problem

محمد قدسی ghodsi@sharif.edu

بهزاد زارع مؤیدی zmoayedi@ce.sharif.edu

دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف

چکیده

در این مقاله الگوریتمی ارائه خواهد شد که قید قابلیت رؤیت را بر مسألهٔ کوتاهترین مسیر پیوندی[1] اعمال می‌کند. الگوریتم از طریق افراز چندضلعی رؤیت نقطهٔ مستلزم رؤیت، به مجموعه‌هایی از نواحی که در آن هر ناحیه مجموع فاصله‌های پیوندی مشخصی از نقاط شروع و پایان دارد، نزدیکترین سکو(یا سکوها)ی مشاهده را می‌یابد و از این نقاط مسیرهایی کمینه به نقاط شروع و پایان را محاسبه می‌کند تا مسیر نهایی جواب بدست آید.مبنای کار این الگوریتم گزارش **[8]** بعنوان مهمترین کار انجام شده در این زمینه می‌باشد و نکته قابل توجه در الگوریتم ما آنست که در پیچیدگی الگوریتم **[8]** نمی‌افزاید.

۱.مقدمه

مسائل کوتاهترین مسیر تنوع زیادی دارند و این تنوع، ناشی از گوناگونی در پارامترهای تعریف کننده مسأله می‌باشد یک مسأله کوتاهترین مسیر با پارامترهای زیر تعریف می‌شوند.

- تابع هدف: اینکه در اندازه‌گیری فاصله از چه معیاری استفاده می‌شود.گزینه‌ها عبارتند از "فاصلهٔ اقلیدسی"، "فاصلهٔ پیوندی"، "طول Lp" و ...
- محدودیتهای اعمال شده: بعنوان مثال آیا متحرک بسادگی باید از نقطهٔ مبدأ به نقطهٔ مقصد برود یا اینکه در طول مسیو باید نقطهٔ مشخصی را نیز مشاهده کند؟
- جغرافیای ورودی: چه نوعی از موانع یا اشیاء دیگر در محیط حرکت متحرک وجود دارد؟
- بعد مسأله: مسأله ممکن است در فضای دوبعدی، سه‌بعدی، یا n بعدی تعریف شود.
- نوع شیء متحرک: آیا متحرک یک شیء نقطه‌ای است یا دارای شکل پیچیده‌تری می‌باشد؟
- پرس و جوی واحد در مقابل پرس و جوی مکرر: آیا هدف ساختن یک ساختار داده‌ای برای جوابگویی به پرسشهای مکرر است؟
- محیط پویا در مقابل ایستا: نیز آیا موانع موجود در محیط ثابتند یا متحرک؟
- الگوریتم دقیق در مقابل تقریبی: آیا هدف یافتن جواب دقیق است یا تقریبی؟
- محیط شناخته شده در مقابل محیط ناشناخته: آیا محیط حرکت متحرک از قبل شناخته شده‌است یا ناشناخته است؟

در این نوشته برای اختصار هرجا عبارت " مسیر" و " فاصله" استعمال می‌شود منظور مسیر و فاصله پیوندی است مگر آنکه خلاف آن تصریح شود.

مسأله کوتاهترین فاصله پیوندی بدون قید قابلیت رؤیت در حالات خاص گوناگون و همچنین در حالت کلی بررسی شده و الگوریتمهای متعددی ارائه شده است.

ایده اصلی حل مسأله کوتاهترین مسیر پیوندی بسیار طبیعی وساده است. ابتدا تمام نقاطی از محیط را که به فاصله یک از s می‌باشند محاسبه می‌کنیم اگر نقطه t در این مجموعه باشد، فاصله پیوندی s از t برابر یک می‌باشد در غیر اینصورت بررسی می‌کنیم که آیا t در این مجموعه قراردارد یا نه و به همین ترتیب پیش می‌رویم تا به t برسیم.

نحوه تعیین نقاط به فاصله i از s نیز مبتنی بریک ایده طبیعی است.ابتدا در s یک منبع نورانی تصور می‌شود و تمام نقاطی از محیط که توسط این منبع روشن

۱- فاصله پیوندی(Link Distance) بین دو نقطه s و t واقع در صفحه حداقل تعداد پاره خطهای لازم جهت تشکیل یک خط شکسته است که بدون تلاقی با موانع موجود در صفحه s و t را به هم وصل کند.

می‌شود مشخص می‌شود. طبق تعریف این نقاط از s به فاصله یک می‌باشند. . این نقاط یک چندضلعی را تشکیل می‌دهند که آن را چندضلعی یا منطقه رؤیت s می‌نامیم و با $VIS_1(s)$ نمایش می‌دهیم برای محاسبه $VIS_2(s)$ نقاط $VIS_1(s)$ را منابع نورانی جدید در نظر گرفته تمام نقاطی از منطقه تاریک را که توسط این منابع، روشن می‌شوند محاسبه می‌کنیم و همین عملیات را برای محاسبه مناطق رؤیت بعدی تکرار می‌کنیم.

Mitchell،Rote وWoeginger در [8] الگوریتمی برای حل حالت کلی مسأله کوتاهترین مسیر پیوندی طبق تعریف ما و بدون قید قابلیت رؤیت ارائه کرده‌اند که در آن با بهره‌گیری از واقعیات حاکم بر مسأله، تکنیکهای موثری اعمال شده و با محدود کردن فضای حل مسأله، پیچیدگی الگوریتم را پایین آورده‌اند.آنها ثابت کرده اند که الگوریتمشان فاصله پیوندی را بین دو نطقه در زمان $O(E\alpha(n)\log^2 n)$ و فضای $O(E)$ محاسبه می‌کند که در آن n تعداد رؤوس موانع و E اندازه گراف مشاهده و $\alpha(n)$ عکس تابع Ackermanاست که رشدی بینهایت کند دارد.تا کنون الگوریتمی با کارایی بهتر مشاهده نشده است.ما مبنای کار خود را این الگوریتم قرار دادیم و تلاش کردیم تا قید "قابلیت رؤیت نقطه r درحداقل یکی از نقاط مسیر" را بر این مسأله اعمال کنیم بگونه‌ای که حتی‌المقدور تأثیری در پیچیدگی زمانی الگوریتم بوجود نیاورد.

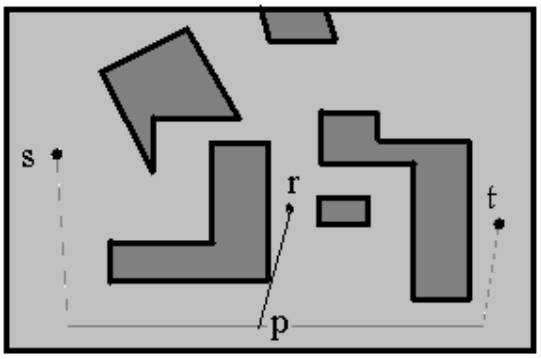
۲.تعاریف و قراردادها

یک چندضلعی (احتمالاً با حفره) یک زیر مجموعه از نقاط صفحه است که مرز آن اجتماع تعدادی محدود از پاره‌خطها یا نیم‌خطهاست. یک چندضلعی که خودش یا مکملش همبند باشد را چندضلعی ساده می‌نامیم. ذکر این نکته لازم است که این تعریف به یک چندضلعی (با یا بدون حفره) اجازه می‌دهد که نامحدود باشد.

تعریف مسأله

یک چندضلعی(با حفره) P (که آن را فضای آزاد می‌نامیم) وسه نقطه s (نقطه منبع) و t (نقطه مقصد) و r (نقطه مستلزم رؤیت) در درون آن داده شده‌است. مطلوبست یافتن یک چندضلعی از s به t که در فضای آزاد قرارگیرد و از حداقل یالهای (پیوندهای) ممکن تشکیل شده‌باشد و از حدقل یکی از نقاط آن بتوان پاره‌خطی به نقطه r رسم‌کرد که کاملاً در فضای آزاد قرارگیرد . این تعداد از یالها را "فاصله پیوندی مقید به رؤیت r بین s وt" می‌نامیم (شکل۱).

برای سادگی فرض می‌کنیم فضای آزاد محدود به مرز بسته است (این فرض به کلیت مسأله لطمه نمی‌زند.).مکمل فضای آزاد مجموعه موانع (یا حفرهها) نامیده می‌شود. این مجموعه از تعداد محدودی چندضلعی ساده تشکیل می‌شود که یکی از آنها نامحدود است و کل صحنه را احاطه کرده‌است.



شکل۱-هدف یافتن کوتاهترین فاصله پیوندي بین s و t است بطوریکه در حداقل يك نقطه از این مسیر، r قابل رؤیت باشد.

n را تعداد تمام اضلاعی در نظر می‌گیریم که P را احاطه می‌کند. همچنین فرض می‌کنیم که فضای آزاد بسته است و بنابراین مسیرها می‌توانند موانع را لمس‌کنند و یا در امتداد اضلاع موانع باشند. همچنین فرض‌می‌کنیم که فضای آزاد همبند است در غیر اینصورت می‌توانیم توجه خود را به جزئی معطوف‌کنیم که t،s وr را در خود دارد (در صورتی که آنها در یک جزء قرار گیرند.) این نوع از پیش پردازش درزمان O(n) انجام می‌شود و در نهایت بدون آنکه به کلیت مسأله لطمه‌ای بخورد فرض می‌کنیم t،s وr رؤوس P می‌باشند(ما همواره می‌توانیم یک مانع نقطه‌ای اختیاری در t،s وr تصور کنیم.)

می‌گوئیم نقطه y از نقطه x قابل رؤیت است اگر پاره خط xy کاملاً در فضای آزاد قرارگیرد. منطقه k-رؤیتی $VIS_k(s)$ برای یک نقطه مثل s، عبارتست از تمام نقاطی از فضای آزاد که فاصله پیوندی آنها از s حداکثر k باشد. طبق این تعریف $VIS_0(s)=\{s\}$ و $VIS_1(s)$ چندضلعی مشاهده s می‌باشد.

تصور کنید $VIS_k(s)$ مجموعه‌ای از نقاط نورانی است آنگاه $VIS_{k+1}(s)$ عبارتست از $VIS_k(s)$ باضافه آن قسمت ازصفحه که بوسیله $VIS_k(s)$ روشن می‌شود. بعلاوه هر آنچه که از منطقه‌ای بتواند روشن شود و متعلق به آن منطقه نباشد می‌تواند از مرز آن منطقه روشن شود و برعکس. هنگامی که مناطق رؤیت را مورد توجه قرار می‌دهیم عموماً بین دو نوع از اضلاع مرزی تمایز قائل می‌شویم : اول ضلع نورانی که درفضای آزاد قرار دارد و بالقوه برای استفاده در روشن کردن مناطق تاریک در مراحل بعدی می‌تواند مورد استفاده قرارگیرد و دوم اضلاع مرزی متعلق به موانع که بخشهایی از مرزهای موانع اولیه اند.

۳.کلیات الگوریتم

مسأله کوتاهترین فاصله پیوندی بدون قید قابلیت رؤیت در حالات خاص گوناگون و همچنین در حالت کلی بررسی شده و الگوریتمهای متعددی ارائه شده است([1],[5],[6],[7],[8],[9],[10],[12]).

ما مبنای کار خود را بر بهترین این کارها قرار داده ایم و تلاش کردیم تا قید قابلیت رؤیت نقطه r را چنانکه تعریف شد بر مسأله اعمال کنیم به گونه‌ای که در پیچیدگی زمانی الگوریتم اثری نگذارد.

طبیعی‌ترین روش در حل مسأله فاصله پیوندی محاسبه مکرر مناطق k-رؤیتی $VIS_k(s)$ از نقطه منبع s برای $k = 1,2,...$ تا زمان رسیدن به نقطه مقصد t است.

در مرحله k ام در صورتی که t در $VIS_k(s)$ قرار داشته باشد الگوریتم متوقف می‌شود و فاصله پیوندی s از t را k اعلام می‌کند در غیر اینصورت به محاسبه $VIS_{k+1}(s)$ می‌پردازد. $VIS_{k+1}(s)$ شامل $VIS_k(s)$است باضافه تمام نقاطی از فضای آزاد که از مرزهای $VIS_k(s)$ قابل رؤیتند لذا منطقه k+۱-رؤیتی می‌تواند از منطقه k-رؤیتی بدست‌آید.با این حال پیروی کورکورانه از این روش ممکن است منجر به دشواریهای عمده شود Suri و O'Rourkeدر[1] نشان داده‌اند که حتی زمانی که مرز $VIS_k(s)$ تنها از یک ضلع نورانی تشکیل می‌شود، پیچیدگی $VIS_{k+1}(s)$ ممکن است از مرتبه $\Omega(n^4)$ باشد.

به همین دلیل کارهای انجام شده بگونه‌ایست که در آنها یا یک حالت خاص بررسی شده‌است مثلاً P بدون حفره است یا با تکنیکهای متعدد فضای مسأله محدود شده است.

ضرورت وجود حداقل یک نقطه p از مسیر کمینه بین s وt بطوریکه r از p قابل رؤیت باشد مسیر کمینه بین s و t را مقید می‌سازد تا از منطقه رؤیت r عبور کند ما استدلال خواهیم کرد که مسأله فاصله پیوندی مقید به رؤیت r ، بین sوt هم ارز خواهد بود با مسأله پیدا کردن نقطه یا نقاطی از $VIS_1(r)$ که مجموع فواصل آنها ازs و t کمینه باشد. بنابراین توجه خود را به محیط اطراف r و بخصوص $VIS_1(r)$ معطوف می‌کنیم و نشان می‌دهیم که $VIS_1(r)$ از نظر مجموع فواصل از s و t به زیر مجموعه‌هایی (احیاناً بعضی تهی) افراز می‌شود و پس از بدست آوردن این افراز ، درمیان مجموعه با حداقل مجموع فواصل به جست و جوی جواب می‌پردازیم و از طریق تحمیل نقشهای مختلف به $VIS_1(r)$جواب را پیدا می‌کنیم.

قضیه۱: فرض کنید مسیر $SHP_r(s,t) = (s, v_1, v_2, ..., v_k, t)$ جواب مسأله باشد و r در نقطه $p \in SHP_r(s,t)$ قابل رؤیت باشد (p را سکوی مشاهده r می‌نامیم) . v_iها رؤوس مسیر، $SHP(x,y)$: کوتاهترین فاصله پیوندی از x به y ، $SHP_z(x,y)$: کوتاهترین فاصله پیوندی از x به y مقید به رؤیت z و $SubPath(x,y,P)$: زیر مسیر از x به y از مسیرP باشد(شکل۲).

الف - اگر $p = v_i$ آنگاه داریم:

$$\begin{cases} \forall j : 1 \le j \le i; SubPath(s, v_j, SHP_r(s,t)) = SHP(s, v_j) \\ \forall j : i \le j \le k; SubPath(v_j, t, SHP_r(s,t)) = SHP(v_j, t) \end{cases}$$

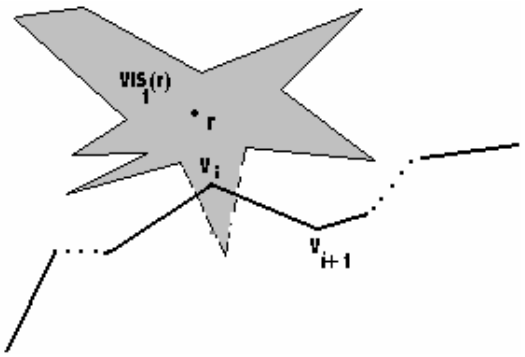
ب- اگر $p \in v_i v_{i+1}$ و $p \ne v_i, v_{i+1}$ آنگاه داریم:

$$\begin{cases} \forall j : 1 \le j \le i; SubPath(s, v_j, SHP_r(s,t)) = SHP(s, v_j) \\ \forall j : i < j \le k; SubPath(v_j, t, SHP_r(s,t)) = SHP(v_j, t) \end{cases}$$

اثبات : اثبات هر دو مورد الف و ب از طریق برهان خلف صورت می‌پذیرد جهت رعایت اختصار به اثبات بند الف اکتفا می‌شود:

اگر وجود داشته باشد $j \ge i$ که مسیری کوتاهتر از $SubPath(v_j, t, SHP_r(s,t))$ از آن به t وجود داشته باشد. این مسیر را با $SubPath(v_j, t, SHP_r(s,t))$ تعویض می‌کنیم مسیر بدست آمده هنوز جواب مسأله است و در عین حال کوتاهتر نیز می‌باشد و این با فرض قضیه تناقض دارد.به همین ترتیب اگر وجود داشته باشد $j \le i$ که کوتاهتر از $SubPath(s, v_j, SHP_r(s,t))$ باشد، قرار دادن آن بجای $SubPath(v_j, t, SHP_r(s,t))$ منجر به کوتاهتر شدن جوا ب مسأله می‌شود و این با فرض تناقض دارد.

قضیه۱ این ایده را القا می‌کند که برای یافتن جواب، چندضلعیهای رؤیت را برای s و t، به سوی r گسترش دهیم قضیه بعد وضعیت $VIS_1(r)$ را نسبت به هر یک از نقاط s و t مشخص می‌کند و ما را به سکوی رؤیت r نزدیکتر می‌کند.



شکل۲- کوتاهترین مسیر پیوندي از چندضلعي رؤیت r مي‌گذرد.

قضیه۲ : اگر x به فاصله پیوندی m از y باشد، آنگاه مجموعه نقاط $VIS_1(x)$ به سه مجموعه S_1، S_2 و S_3 از نقاط که فاصله نقاط هر مجموعه از y به ترتیب m-1، m و m+1 است افرازمی‌شود.

اثبات : اگر $SHP(y,x) = (y, v_1, ..., v_{m-1}, x)$ آنگاه داریم:

$\forall i \in \{1,..,m-1\}, SubPath(y, v_i, SHP(y,x)) = SHP(y, v_i)$

چون در غیر اینصورت مسیری کوتاهتر از $SHP(y,x)$ وجود خواهد داشت و این خلاف فرض است. بنابراین $\left|SHP(y, v_{m-1})\right| = m-1$. از طرفی v_{m-1} متعلق به $VIS_1(x)$ است چون به فاصله پیوندی یک از x قرار دارد.پس وجود دارد نقطه ای از $VIS_1(x)$ که به فاصله m-1 ازy باشد. همچنین تمام نقاط به همسایگی بسیار کوچک x، از y به فاصله m می‌باشند بنابراین وجود دارد نقطه‌ای از $VIS_1(x)$ که از y به فاصله m می‌باشد.

حال توجه کنید که طبق تعریف چندضلعی رؤیت x، تمام نقاط $VIS_1(x)$ از x به فاصله یک هستند بنابراین با توجه به اینکه x از y به فاصله m می‌باشد، تمام نقاط $VIS_1(x)$ از y به فاصله حداکثر m+1 می‌باشند لذا هر نقطه ای از $VIS_1(x)$ که از y به فاصله m-1 و m نباشد لاجرم به فاصله m+1 است.

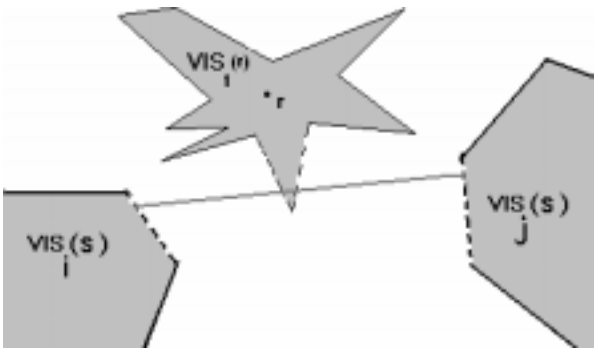
محاسبه این افراز در جریان عمومی الگوریتم یافتن کوتاهترین مسیر میسر است. بدین ترتیب که در مرحله m-1 اشتراک $VIS_{m-1}(y)$ را با $VIS_1(x)$ محاسبه می‌کنیم و بعنوان مجموعه نقاط به فاصله m-1 از y در نظر می‌گیریم حال این اشتراک را از $VIS_1(x)$ تفریق می‌کنیم و مجدداً همین عمل اشتراک گیری را برای $VIS_m(y)$ و قسمت باقیمانده $VIS_1(x)$ انجام می‌دهیم و بعنوان مجموعه نقاط به فاصله m از y می‌گیریم . آنچه از $VIS_1(x)$ می‌ماند را نیز بعنوان مجموعه سوم می‌گیریم.

الگوریتم دو افراز از $VIS_1(r)$ با توجه به فاصله نقاط آن از s و t بدست می‌آورد. سپس با برهم‌گذاری این دو افراز $VIS_1(r)$ را بر اساس مجموع فواصل از هر دو نقطه s و t به ٤ مجموعه افراز می‌کند. در این مرحله مجموعه با کمترین مجموع فواصل از s و t را بعنوان منطقه حاوی سکو(یا سکوها)ی رؤیت جواب مشخص می‌کند.

اکنون به قضیه۱ بر می‌گردیم و آن را دقیقتر بررسی می‌کنیم.در قضیه۱ تفاوت حالت در بند الف و ب از آنجا ناشی می‌شود که مسیر در نقطه p شکسته شود یا نه. در حالت الف قضیه با اطمینان می‌گوید که نقطه p تمام نقطه هایی از $VIS_1(r)$ است که کمترین فاصله را نسبت به s و t داشته باشند بنابراین در افراز نهایی بدست آمده از $VIS_1(r)$ بر اساس مجموع فواصل از s و t مجموعه با کمترین مجموع فواصل تماماً سکوهای جواب می‌باشند. اما در حالت ب قضیه می‌گوید ممکن است مسیر کمینه $SHP_r(s,t)$ در p شکست پیدا نکند و این مسأله ما را بر آن می‌دارد که در مجموعه با کمترین مجموع فواصل در افراز مورد نظر به دنبال نقطه یا نقاطی بگردیم که مسیر در آنها شکست نداشته باشد.

چنانکه قبلاً ذکر شد یک نقطه از بیرون چندضلعی $VIS_{k+1}(x)$ توسط یکی از نقاط درون $VIS_k(x)$ روشن می‌شود اگر از یکی از نقاط مرزی $VIS_k(x)$ روشن شود و برعکس.این واقعیت گویای این مطلب است که در مسیر کمینه پیوندی $SHP(y,x) = (y, v_1, ..., v_k, x)$ می‌توان v_i را برروی مرز $VIS_i(y)$ گرفت. بنابراین در شرایط قضیه۱ می‌توان تمام رؤوس $SHP_r(s,t)$ بین s و p (مگر خود p در حالت الف) را متعلق به مرزهای مناطق رؤیت s و تمام رؤوس بین p و t را متعلق به مناطق رؤیت t دانست.

با توجه به این مطلب و حالت بند ب از قضیه۱ الگوریتم ابتدا مناطق رؤیت را از s و t به سوی r گسترش می‌دهد، سپس افراز $VIS_1(r)$ را با توجه به مجموع فواصل نسبت به s و t بدست آورده مجموعه با کمترین مجموع فواصل را مشخص می‌کند. فرض کنید این مجموعه S باشد. هرکدام از مجموعه مناطق VIS_k که برای s و t به سمت r محاسبه شده است در یکی از مناطق رؤیت خود برای اولین بار با s اشتراک پیدا می‌کنند. فرض کنید این مناطق رؤیت برای s در $VIS_{i+1}(s)$ و برای t در $VIS_{j+1}(t)$ باشند. بنابراین $VIS_i(s)$ و $VIS_j(t)$ هنوز با s تلاقی ندارند(شکل۳). اگر حالت مسأله مطابق با بند ب قضیه۱ باشداین بدان معنی است که حداقل یکی از نقاط $VIS_i(s)$ از حداقل یکی از نقاط $VIS_j(t)$ و در امتداد s قابل مشاهده است به عبارت دیگر وجود دارد پاره‌خطی که دوسرش بر مرزهای نورانی $VIS_i(s)$ و $VIS_j(t)$ باشد، از S بگذرد و کاملاً در فضای آزاد قرار گیرد.



شکل۳- دو منطقه رؤیت براي اولین بار $VIS_1(r)$ را رؤیت مي‌کنند.

الگوریتم بر این اساس ابتدا وجود چنین پاره خطی را بررسی می‌کند.و این کار را از طریق محاسبه نقاطی از مرز $VIS_j(t)$ که توسط پرتوهای نور منتشر شده از $VIS_i(s)$ و گذرنده از S روشن شده‌است انجام می‌دهد. در صورت عدم وجود چنین نقاطی الگوریتم S را بعنوان مجموعه سکوهای رؤیت r برمی‌گرداند. در صورت وجود نیز یکی از پاره‌خطهایی را که از فضای آزاد S عبور کند و دو سرش بر مرزهای نورانی $VIS_i(s)$ و $VIS_j(t)$ قرار دارد را بعنوان پاره خط حاوی سکوی جواب برمی‌گرداند.

۴.جزئیات الگوریتم و تحلیل پیچیدگی

چنانکه گفته شد الگوریتم پس از محاسبه افراز $VIS_1(r)$ بر اساس مجموع فواصل از s و t به بررسی وضعیت $VIS_1(r)$ نسبت به مناطق رؤیت، چنانکه تشریح شد می‌پردازد. درادامه فرض می‌کنیم S مجموعه انتخابی از افراز نهایی $VIS_1(r)$ باشد و $VIS_{i+1}(s)$ و $VIS_{j+1}(t)$ اولین مناطق رؤیتی باشند که در گسترش مناطق رؤیت از s و t به سوی r با $VIS_1(r)$اشتراک پیدا می‌کنند.

الف- محاسبه $VIS_1(r)$: با استفاده ازروشی که Ghosh و Mount در[4] ارائه داده‌اند این عمل در زمان $O(n\log n+E)$ انجام می‌شود.

ب- محاسبه افراز نهایی $VIS_1(r)$ و انتخاب S :

با استفاده از الگوریتم [8] چندضلعیهای رؤیت برای s و t به سمت r محاسبه می‌شود چنانکه گفته شد این عمل در زمان $O(E\alpha(n)\log^2 n)$ انجام می‌شود. ابتد افراز $VIS_1(r)$ بر اساس فاصله نسبت به s محاسبه می‌شود.

- ب۱- محاسبه $S_1 = VIS_{i+1}(s)\cap VIS_1(r)$
- ب۲-محاسبه

$$\{S_1,S_2,S_3\} \Longleftarrow\!\!=\!\!= S_2 = VIS_{i+2}(s)\cap VIS_1(r) - S_1$$

- ب۳- محاسبه $S_3 = VIS_1(r) - (S_1 - S_2)$

همین عمل برای محاسبه افراز $VIS_1(r)$ بر اساس فاصله نسبت به t انجام می‌شود تا افراز $\{T_1,T_2,T_3\}$ بدست آید. هر کدام از عملیات بند ب۱ تا ب۳ با استفاده از الگوریتم Map verley در زمان $O(n_i\log n_i + k_i\log n_i)$ قابل محاسبه است[13]. که در آن n_i مجموع پیچیدگیهای چندضلعیهای مورد بررسی و k_i پیچیدگی S_i هاست.

این زمان اجرا در بدترین حالت برابر $O(n\log n)$ می‌باشد. بنابراین محاسبه هر افراز در زمان $O(n\log n)$ انجام پذیر است. بدست آوردن افراز نهایی نیزمعادل است با اعمال یک Map Overley دیگر که باز در $O(n\log n)$انجام می‌پذیرد. پس در کل محاسبه افراز نهایی و انتخاب S در زمان $O(n\log n)$ قابل انجام است.

ج- یافتن سکوی رؤیت :

با فرض اینکه اولین مناطق رؤیت s و t که با S اشتراک دارند به ترتیب $VIS_{m+1}(s)$ و $VIS_{n+1}(t)$ باشند در این قسمت باید وضعیت $VIS_1(r)$ با توجه به $VIS_m(s)$ و $VIS_n(t)$ ارزیابی شود، اگر رؤیت متقابل هیچ دو نقطه‌ای از مرزهای نورانی این دو چندضلعی از طریق $VIS_1(r)$ و با درنظر گرفتن تمام موانع موجود امکان پذیر نباشد، تمام نقاط S سکوی رؤیت r برای مسأله خواهدبود در غیراینصورت نقاطی از S که ازطریق آنها مرزهای نورانی $VIS_m(s)$ و $VIS_n(t)$ یکدیگر را رؤیت می‌کنند،سکوی رؤیت r برای مسأله‌اند.

نحوه بررسی این مطلب بدین نحو صورت می‌گیرد.

- ج۱- $VIS_{m+1}(s)$ را محاسبه کن.
- ج۲-۱- اشتراک $VIS_{m+1}(s)$ با $VIS_n(t)$ را بدست بیاور.
- ج۲-۲- اگر این اشتراک تهی بود، S را بعنوان سکوی رؤیت برگردان.
- ج۳-۱- در صورتی که این اشتراک تهی نباشد، آن را در S_1 ذخیره‌کن.
- ج۳-۲- S را بعنوان یک چندضلعی مانع وارد مجموعه موانع کن ، مجدداً $VIS_{m+1}(s)$ را این‌بار با ترکیب‌بندی جدید موانع بدست بیاور و اشتراک آن را با $VIS_n(t)$ ، S_2 بنام.
- ج۴- $S_3 = S_1 - S_2$ را محاسبه کن.
- ج۵-۱- اگر $S \equiv \phi$ آنگاه S را بعنوان مجموعه سکوهای رؤیت برگردان.
- ج۵-۲- اگر $S \neq \phi$ آنگاه مشابه عمل فوق در محاسبه S_3 را این بار با تعویض نقش s و t برای محاسبه S_4 انجام بده.
- ج۵-۳- بر روی مرز نورانی S_3 و قسمت مشترک با $VIS_n(t)$ نقطه‌ای مثل A انتخاب کن. با دوران یک خط جاروب حول این نقطه نقطه‌ای را مثل B از مرز نورانی S_4 که از این نقطه قابل رؤیت است پیدا کن و پاره خط AB را بعنوان پاره خط حاوی سکو یا سکوهای رؤیت r برگردان.

تمام بندهای ج۱ تا ج۵-۲ مبتنی بر الگوریتم Map Overley می‌باشند و بنابراین در زمان $O(n \log n)$ قابل انجامند.

قسمت ج۵-۳ یک دوران خط جاروب است که با توجه به مشخص بودن چندضلعی S_4 در محدوده S_4 دوران صورت می‌گیرد و با توجه به پیچیدگی S_4 که از مرتبه $O(n)$ می‌باشد، زمان اجرا از مرتبه $O(n)$ می‌باشد.

۵.نتیجه گیری

در این مقاله الگوریتمی ارائه شد که قید قابلیت رؤیت نقطه سومی را بر مسأله کوتاهترین فاصله پیوندی اعمال می‌کند. ما مبنای کار خود را الگوریتم[8] قراردادیم که به نظر می‌رسد بهترین کار انجام شده در این زمینه تا کنون است این الگوریتم مسیرپیوندی بین دو نقطه را در $O(E\alpha(n)\log^2 n)$و با حافظه O(E) محاسبه‌می‌کند و الگوریتمی ارائه کردیم که با همان زمان اجرا و حافظه عمل می‌کند، صورتهای پیچیده‌تری ازقید رؤیت مطرح است از جمله تعدد نقاط مستلزم رؤیت که همچنان بصورت مسائل باز، باقی مانده‌اند.

۶.مراجع

[1] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. Computer Vision, Graphics, and Image Processing, 35:99--110, 1986.

[2] S. K. Ghosh. Computing the visibility polygon from a convex set and related problems. Journal of Algorithms, 12:75--95, 1991.

[3] B. Chazelle. Triangulating a simple polygon in linear time. Discrete & Computational Geometry, 6:485--524, 1991.

[4] S. K. Ghosh and D. M. Mount. An output sensitive algorithm for computing visibility graphs. SIAM Journal on Computing, 20(5):888--910, 1991.

[5] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. Tarjan. Linear time algorithms for visibility and shortest path problems inside triangulated simple polygons. Algorithmica, 2:209--233, 1987.

[6] L. J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. Journal of Computer and System Sciences, 39(2):126--152, Oct. 1989.

[7] J. S. B. Mitchell, C. Piatko, and E. M. Arkin. Computing a shortest k-link path in a polygon. In Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science, pages 573--582, 1992.

[8] J. S. B. Mitchell, G. Rote, and G. Woeginger. Minimum-link paths among obstacles in the plane. In Proceedings of the Sixth Annual ACM Symposium on Computational Geometry, pages 63--72, 1990.

[9] Y. Ke, An efficient algorithm for link distance problems, Proc. 5th Annual ACM Sym-posium on Computational Geometry, 1989, pp. 69--78.

[10] M. H. Alsuwaiyel and D. T. Lee, ``Minimal Link Visibility Paths inside a Simple Poly-gon,'' Computational Geometry: Theory and Applications, 3, 1, (1993) 1-26.

[11] S. K. Ghosh, ``Computing the visibility polygon from a convex set and related problems,'' J. Algorithms, 12 (1991), 75-95.

[12] J. Hershberger and J. Snoeyink, ``Computing minimum length paths of a given homotopy class,'' Computational Geometry: Theory and Applications, 4, 2, (June 1994), 63-97.

[13]J.L.Bently and T.A. Ottmann. Algorithms for reporting and counting geometric untersections.IEEE Trans. Comput.,C-28 :643-647,1979.