



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پایان نامه دکتری
گرایش نرم افزار

عنوان

برنامه ریزی حرکت با قيود قابلیت دید

نگارش

رامتین خسروی

استاد راهنما

دکتر محمد قدسی

آذر ۱۳۸۴

برنامه‌ریزی حرکت با قیود قابلیت دید

چکیده

در این رساله، مسئله‌ی برنامه‌ریزی حرکت با گونه‌های مختلفی از قیود قابلیت دید مورد مطالعه قرار گرفته است. دامنه‌هایی که این مسئله در آنها بررسی شده‌اند عبارتند از چندضلعی‌های ساده و دامنه‌های چندضلعی گونه در دو بعد و رویه‌های چندوجهی گونه در سه بعد (در دو حالت محدب و مقعر). در هر یک از این دامنه‌ها، مسئله‌ی یافتن کوتاه‌ترین مسیر بین دو نقطه که از یک هدف دید نقطه‌ای قابل دید باشد مورد مطالعه قرار گرفته و الگوریتم‌های کارایی برای آن ارائه شده است. همچنین پیش‌پردازش دامنه طوری که با دریافت مقصد حرکت در پرس‌وجو، مسیر بهینه در زمان کمتری پیدا شود مورد بررسی قرار گرفته است. گونه‌ی دیگری از مسئله که در آن هدف دید به عنوان نقطه‌ی پرس‌وجو داده شود نیز مطالعه شده است و در دو حالت مقصد باز و مقصد مشخص برای آن الگوریتم کارا ارائه شده است.

تعمیم قیود قابلیت دید به قیود دیدار نیز یکی از مسائل مورد مطالعه در این رساله است. در

این قیود، متحرک مقید به گذشتن یا تماس با یک ناحیه‌ی مشخص از دامنه در مسیر حرکت از مبدأ

به مقصد است. به این ترتیب، قیود دیدار حالت کلی تری از قیود قابلیت دید است، چون در قیود

قابلیت دید می‌توان مسئله را به دیدار چندضلعی قابلیت دید هدف دید تبدیل کرد. این مسئله در چندضلعی‌های ساده و دامنه‌های چندضلعی‌گونه حل شده است و گونه‌ی پرس‌وجوی مقصد نیز در آنها بررسی شده است.

برای حل مسائل مذکور روش‌هایی ابداع شده است که جزء نوآوری‌های این رساله محسوب می‌شود. تعریف بازه‌های بهینگی به عنوان بازه‌هایی که ناحیه‌ی هدف را بسته به ساختار ترکیبیاتی مسیر از نقاط آن ناحیه به مبدأ و مقصد افراز می‌کند ابزار اصلی برای تبدیل مسئله به یک مسئله‌ی گسسته است. همچنین الگوی بازتابش جبهه‌ی موج، تعمیمی از روش انتشار جبهه‌ی موج است که با نام دایکسترای پیوسته نیز شناخته می‌شود. از این روش برای ساختن نقشه‌های مورد نیاز در حالات پرس‌وجوی مقصد استفاده می‌شود. روش اصلی که برای حل مسئله با قید قابلیت دید نقطه‌ی پرس‌وجو استفاده می‌شود جاروی شعاعی است که در آن رفتار نقطه‌ی تماس بهینه در جریان حرکت دورانی حول یک رأس بازتابی مورد مطالعه قرار می‌گیرد.

کلیدواژه‌ها: هندسه‌ی محاسباتی، کوتاه‌ترین مسیر، قابلیت دید، انتشار جبهه‌ی موج

قدردانی

در این جا لازم است از کلیه‌ی افرادی که مرا در انجام این پروژه کمک نموده‌اند تشکر کنم. استاد گرامی آقای دکتر محمد قدسی که در تمام دوره‌ی تحصیل در مقاطع کارشناسی و کارشناسی ارشد و به خصوص در مدت تحصیل در دوره‌ی دکتری راهنمایی و هدایت من را بر عهده داشتند. از ایشان کمال سپاس و قدردانی را دارم و موفقیت خود را مدیون حمایت‌های مؤثر او می‌دانم. همچنین از اساتید بزرگواری که برای داوری این رساله قبول زحمت کردند آقایان دکتر تشکری، دکتر جم‌زاد، دکتر دانشگر، دکتر رزازی و دکتر مهدوی امیری تشکر می‌کنم. جای آن است که به طور خاص از آقای دکتر دانشگر که در چندین نوبت از راهنمایی‌های بسیار ارزشمند ایشان استفاده کردم تشکر کنم.

بخش قابل ملاحظه‌ای از پژوهش این رساله در مدت اقامت من در پژوهشکده‌ی علوم کامپیوتر مرکز تحقیقات فیزیک نظری و ریاضیات انجام شده است که بدون وجود فضا و امکانات پژوهشکده‌ی یاد شده ممکن نبود. در اینجا از آقای دکتر سربازی آزاد که این امکان را فراهم کردند تشکر می‌کنم. در طول مدت تحصیل و پژوهش، از هم‌فکری و هم‌دلی دوستان و همکاران بهره بردم که بدین وسیله از ایشان سپاسگزاری می‌کنم، به خصوص آقای فرشاد رستم‌آبادی، آقای دکتر جابری‌پور و خانم دکتر سیرجانی. همچنین دوست بسیار عزیز، آقای دکتر کیارش بازرگان که امکان دسترسی من به برخی مقالات و منابع خارجی را مهیا کرد.

شاید از همه مهم‌تر، اگر ایثار مادر، همفکری برادر و همدلی همسر همراه نبود پیمودن این راه دشوار نیز ممکن نمی‌شد. در اینجا بر خود لازم می‌دارم سپاس ویژه‌ی خود از این عزیزان را ابراز کنم.

فهرست مطالب

۱	مقدمه	۱
۱	زمینه	۱.۱
۱	برنامه‌ریزی حرکت	۱.۱.۱
۴	برنامه‌ریزی حرکت با قید قابلیت دید	۲.۱.۱
۶	برنامه‌ریزی حرکت با قید دیدار	۳.۱.۱
۷	انگیزه‌ی پژوهش	۲.۱
۱۰	نتیجه‌های به‌دست آمده	۳.۱
۱۱	دیدن هدف دید نقطه‌ای مشخص در طول مسیر	۱.۳.۱
۱۲	دیدن هدف دید نقطه‌ای پرس‌وجو	۲.۳.۱
۱۴	قید دیدار چندضلعی	۳.۳.۱
۱۵	خلاصه‌ی روش‌ها	۴.۳.۱
۱۸	طرح رساله	۴.۱

۲ مفاهیم پایه و کارهای مرتبط

۲۰

۲۰ تعریف‌های ابتدایی ۱.۲

۲۰ دامنه‌های دوبعدی ۱.۱.۲

۲۵ ابعاد بالاتر ۲.۱.۲

۲۷ کوتاه‌ترین مسیرها در یک چندضلعی ساده ۲.۲

۳۰ کوتاه‌ترین مسیرها در یک دامنه‌ی چندضلعی‌گونه ۳.۲

۳۰ جستجوی گراف قابلیت دید ۱.۳.۲

۳۱ روش دایکسترای پیوسته ۲.۳.۲

۳۳ کوتاه‌ترین مسیرها روی یک رویه‌ی چندوجهی‌گونه ۴.۲

۳۵ قابلیت دید ۵.۲

۳۵ قابلیت دید در دو بعد ۱.۵.۲

۳۷ قابلیت دید در بیش از دو بعد ۲.۵.۲

۳ کوتاه‌ترین مسیرها با قابلیت دید یک نقطه

۳۹

۳۹ الگوی کلی ۱.۳

۴۳	چندضلعی‌های ساده	۲.۳
۴۷	دامنه‌های چندضلعی گونه	۳.۳
۴۹	دروازه‌ی قابلیت دید	۱.۳.۳
۵۱	محاسبه‌ی بازه‌های بهینگی	۲.۳.۳
۵۵	رویه‌های چندوجهی گونه	۴.۳
۵۵	بازه‌های بهینگی	۱.۴.۳
۵۹	محاسبه‌ی بازه‌های بهینگی	۲.۴.۳

۴ کوتاه‌ترین مسیرها با قید قابلیت دید نقطه‌ی پرس‌وجو

۶۴		
۶۵	کوتاه‌ترین مسیر با مقصد باز	۱.۴
۶۵	مفاهیم پایه‌ای	۱.۱.۴
۶۷	مراحل الگوریتم	۲.۱.۴
۷۴	کوتاه‌ترین مسیر با مقصد مشخص	۲.۴
۷۵	تعریف‌ها و مفاهیم پایه‌ای	۱.۲.۴
۷۹	الگوریتم پیش‌پردازش	۲.۲.۴
۸۷	خلاصه و تحلیل الگوریتم	۳.۲.۴

۵ کوتاه‌ترین مسیرها با قید دیدار چندضلعی

۸۹	چندضلعی‌های ساده	۱.۵
۹۰	محاسبه‌ی کوتاه‌ترین مسیر مقید	۱.۱.۵
۹۵	ساختن زیرتقسیم پرس‌وجوی تک‌مبدأ	۲.۱.۵
۱۰۱	الگوی بازتابش جبهه‌ی موج	۲.۵
۱۰۳	زیرتقسیم بازتابی	۱.۲.۵
۱۰۹	محاسبه‌ی زیرتقسیم بازتابی	۲.۲.۵
۱۱۱	دامنه‌های چندضلعی‌گونه	۳.۵
۱۱۳	جستجوی گراف قابلیت دید	۴.۵
۱۱۸	۶ نتیجه‌گیری	
۱۱۸	نتایج به دست آمده	۱.۶
۱۲۰	پژوهش‌های آینده و مسائل باز	۲.۶
۱۲۰	گراف قابلیت دید و قید دیدار	۱.۲.۶
۱۲۱	کوتاه‌ترین مسیر بازرسی در دامنه‌های چندضلعی‌گونه	۲.۲.۶
۱۲۱	نزدیک‌ترین نقطه روی چندضلعی به k نقطه	۳.۲.۶
۱۲۳	دیدار زیردنباله‌ها	۴.۲.۶
۱۲۳	قابلیت دید هدف متحرک	۵.۲.۶

۱۲۵

A واژه‌نامه‌ی فارسی به انگلیسی

۱۳۱

B واژه‌نامه‌ی انگلیسی به فارسی

فهرست شکل‌ها

۳ فضای پیکربندی حرکت روبات	۱
۴ مسیر نگهبان در یک چندضلعی ساده	۲
۱۶ بازه‌های بهینگی روی چندضلعی قابلیت دید	۳
۲۲ گراف قابلیت دید برای یک دامنه‌ی چندضلعی گونه	۱
۲۳ نقشه‌ی کوتاه‌ترین مسیر در یک دامنه‌ی چندضلعی گونه	۲
۲۶ مسیرهای ژئودزیک روی رویه‌ی چندوجهی گونه	۳
۲۹ محاسبه‌ی کوتاه‌ترین مسیر در چندضلعی ساده	۴
۴۱ نقطه‌ی بهینه محلی روی یک بازه‌ی بهینگی	۱

۴۳ کوتاه‌ترین مسیر با قید قابلیت دید در چندضلعی ساده	۲
۴۵ بازه‌های بهینگی تشکیل شده روی پنجره	۳
۴۶ یافتن پنجره‌ی جدا کننده نقطه‌ی دید از مبدأ	۴
۴۸ حالت‌های خاص در کوتاه‌ترین مسیر با قابلیت دید در دامنه‌ی چندضلعی گونه	۵
۵۶ بازه‌ی بهینگی متناظر با یک نقطه در رویه‌ی چندوجهی گونه	۶
۵۶ ساختار بازه‌های بهینگی	۷
۵۸ بازه‌ی بهینگی متناظر با دو نقطه روی رویه‌ی چندوجهی گونه	۸
۶۰ نظریه کوتاه‌ترین مسیر یک رویه‌ی چندوجهی گونه	۹
۶۱ رویه‌ی چندوجهی با $O(n^2)$ بازه‌ی بهینگی	۱۰
۶۶ نقطه‌ی بهینه روی پنجره‌ی جدا کننده‌ی مبدأ از نقطه‌ی دید	۱

۶۷ ساختار قیف مانند	۲
۶۸ پنجره‌های جدا کننده	۳
۶۹ بازه‌های زاویه‌ای مشخص کننده‌ی پنجره‌های جدا کننده	۴
۷۰ افزاز پنجره‌های جدا کننده	۵
۷۲ تغییر ساختار قیف در جریان جاروی شعاعی	۶
۷۶ قیف‌های متناظر با s و t	۷
۷۸ بازه‌های بهینگی مرتبط با دو قیف	۸
۸۰ پنجره‌های جداکننده متناظر با s و t	۹
۸۱ ساختار خم بهینه	۱۰
۸۴ تقاطع‌های خم بهینه و یک خط دلخواه	۱۱
۸۵ ساختن قیف‌ها برای به دست آوردن $O(n^2)$ رخداد بازه	۱۲

۹۱ بررسی خواص مسیرهای دیدارکننده‌ی چندضلعی	۱
۹۴ یافتن دروازه‌ی یک نقطه به چندضلعی هدف	۲
۹۷ بازه‌ی بهینگی روی دروازه‌ی یک نقطه به چندضلعی هدف	۳
۹۸ نقشه‌ی کوتاه‌ترین مسیر دیدارکننده‌ی T	۴
۱۰۱ بازتابش یک موجک	۵
۱۰۲ موجک‌های دیدارکرده‌ی اولیه	۶
۱۰۴ ناحیه‌های بازتابش و ناحیه‌ی درون‌گذر	۷
۱۰۶ زیرتقسیم بازتابی	۸
۱۱۰ محاسبه‌ی زیرتقسیم بازتابی	۹
۱۱۵ گراف قابلیت دید تعمیم‌یافته	۱۰

مقدمه

۱.۱ زمینه

۱.۱.۱ برنامه‌ریزی حرکت

طی چند دهه‌ی گذشته برنامه‌ریزی حرکت به عنوان یک زمینه‌ی پژوهشی اساسی در حیطه‌ی رباتیک و هندسه‌ی محاسباتی مطرح بوده است. مسئله‌ی پایه‌ی برنامه‌ریزی حرکت، یافتن یک مسیر برای یک ربات بین تعدادی مانع است طوری که برخوردی بین ربات و مانع‌ها رخ ندهد [۱]. این مسئله نخستین بار به عنوان مسئله‌ی جابه‌جاکننده‌ی پیانو^۱ مطرح شد که هدف آن یافتن دنباله‌ای از حرکات بود طوری که بتوان یک جسم صلب (مانند یک پیانو) را از بین تعدادی مانع (مثل اشیاء داخل یک اتاق) از یک وضعیت مبدأ به یک وضعیت مقصد برد. مهم‌ترین پیچیدگی در حل این مسئله از ماهیت هندسی آن منجر می‌شود که به خاطر تعداد زیاد دنباله‌های ممکن برای انجام این کار نمایی است. رایف^۲ در سال ۱۹۷۹ نشان داد که این مسئله PSPACE-سخت است [۲] و بعد از آن کنی^۳ و رایف نشان دادند که مسئله‌ی مذکور NP-سخت است [۳]. این موضوع باعث شد که گونه‌های بسیاری از این مسئله ایجاد شوند که معمولاً با محدود شدن درجه‌ی آزادی

^۱piano mover's problem

^۲Reif

^۳Canny

روبات، مسئله را در زمان چندجمله‌ای حل می‌کردند.

بیشتر رهیافت‌هایی که به برنامه‌ریزی حرکت وجود دارد بر مفهوم فضای پیکربندی^۴ استوارند که توسط لوزانو پرز^۵ و وسلی^۶ در [۴] ارائه شد. فضای پیکربندی مجموعه‌ی تمام وضعیت‌هایی است که روبات می‌تواند در آن قرار بگیرد. برخی از این وضعیت‌ها با موانع برخورد دارند و برخی در فضای آزاد واقع می‌شوند. به این ترتیب، مسئله به یافتن دنباله‌ی همبندی از وضعیت‌ها (یا پیکربندی‌ها) در فضای پیکربندی است که هیچ‌کدام با موانع برخورد نداشته باشند. تبدیل دیگری که حل مسئله را ساده‌تر می‌کند استفاده از مجموع مینکوفسکی^۷ است که توسط آن می‌توان یک روبات را که به شکل یک چندضلعی در صفحه یا چندوجهی در فضا قرار دارد به یک روبات نقطه‌ای تبدیل کرد. به این ترتیب، بسیاری از مسائل برنامه‌ریزی حرکت به یافتن یک مسیر برای یک متحرک نقطه‌ای تبدیل می‌شوند. به عنوان مثال شکل ۱ (چپ)، مسیر حرکت یک روبات را که فقط قادر به حرکت انتقالی است، بین تعدادی مانع نشان می‌دهد. در شکل سمت راست، مجموع مینکوفسکی روبات و موانع محاسبه شده‌اند (چندضلعی‌های خاکستری) و به عنوان مانع فرض می‌شوند. به این ترتیب، مسئله به یافتن مسیری برای یک متحرک نقطه‌ای بین موانع خاکستری رنگ تبدیل می‌شود.

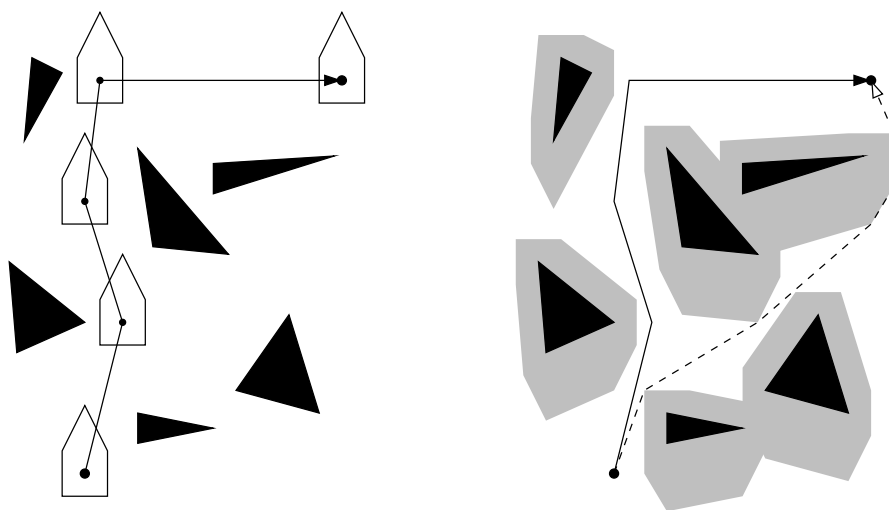
برخی اوقات تنها یافتن یک مسیر بین مبدأ و مقصد برای ما کافی است. حل چنین مسئله‌ای چندان مشکل نیست. اما در بسیاری حالات واقعی هدف ما بهینه کردن یک تابع هزینه است. یکی از متداول‌ترین توابع هزینه طول اقلیدسی مسیر است که عموماً با زمان طی کردن مسیر یا انرژی لازم برای این کار متناسب است. در چنین مواردی یکی از مسائل بنیادین هندسه‌ی محاسباتی که یافتن کوتاه‌ترین مسیرها در بین موانع است خود را نشان می‌دهد. به عنوان مثال در شکل ۱ (راست)، مسیر خط‌چین، کوتاه‌ترین مسیر بین دو نقطه‌ی مبدأ و مقصد را برای حرکت روبات مشخص می‌کند.

configuration space^۴

Lozano-Perez^۵

Wesley^۶

Minkowski sum^۷



شکل ۱: تبدیل مسئله‌ی حرکت روبات به حرکت یک متحرک نقطه‌ای

در شکل کلی مسئله‌ی کوتاه‌ترین مسیر که در دو بعد یا بیشتر قابل تعریف است، تعدادی مانع به عنوان ورودی به مسئله داده می‌شوند. هدف مسئله یافتن کوتاه‌ترین مسیر بین دو نقطه‌ی مبدأ و مقصد است طوری که این مسیر داخل هیچ مانعی نشود. گونه‌های بسیار متنوعی از این مسئله تعریف شده است که مسئله را در دامنه‌های مختلف، شکل‌های مختلف برای موانع، پرس‌وجو^۸ بودن یا مشخص بودن مبدأ یا مقصد و سایر پارامترها مطرح می‌کند [۵].

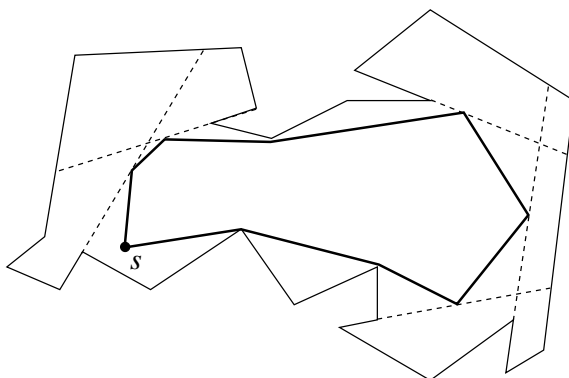
هر چند طول اقلیدسی معیار شناخته‌شده‌ای برای تعریف تابع هزینه است، اما معیارهای مهم دیگری نیز مورد استفاده قرار می‌گیرند. یکی از مهم‌ترین آنها در محیط‌های روباتیک، معیار پیوند^۹ است. در این معیار هزینه‌ی یک مسیر تعداد شکست‌های آن است. دلیل تعریف این معیار پرهزینه بودن تغییر مسیر روبات‌ها است. معیارهای دیگری هم در مسئله‌ی کوتاه‌ترین مسیر تعریف می‌شوند که اینجا از بیان آنها خودداری می‌کنیم.

علاوه بر گوناگونی در تعریف تابع هزینه، قیدهای مختلفی ممکن است مسئله را محدود کنند. نمونه‌ای از این قیدها محدودیت روی انحنای^{۱۰} مسیر است که از محدودیت‌های فیزیکی

^۸query

^۹link metric

^{۱۰}curvature



شکل ۲: یک چندضلعی ساده که با خطوط ممتد نازک نشان داده شده به همراه مسیر نگهبان درون آن که با خطوط ممتد ضخیم نشان داده شده. این مسیر از مبدأ مشخص s شروع و به همان نقطه ختم می‌شود.

ناشی می‌شود. نمونه‌ی دیگر، محدودیت حداکثر ارتفاع یا شیب مسیر روی دامنه‌های سه بعدی است. همچنین برخی اوقات لازم است متحرک به دلیلی مانند سوخت‌گیری در طول مسیر خود از محل‌های معینی دیدار کند. قید دیگری که به آن توجه زیادی شده قید قابلیت دید^{۱۱} است. این قید را به طور مفصل در بخش بعد مطالعه می‌کنیم.

۲.۱.۱ برنامه‌ریزی حرکت با قید قابلیت دید

در هندسه‌ی محاسباتی، دو نقطه از هم قابل دید هستند اگر پاره‌خط واصل آنها هیچ‌یک از موانع موجود در دامنه را قطع نکند. یکی از قدیمی‌ترین مسائلی که نوعی برنامه‌ریزی حرکت با قید قابلیت دید محسوب می‌شود مسئله‌ی مسیر نگهبان^{۱۲} است [۶، ۷، ۸، ۹، ۱۰]. در این مسئله یک چندضلعی ساده داده شده است که مدلی از یک محوطه‌ی بسته مانند یک موزه است. هدف مسئله یافتن یک مسیر بسته داخل چندضلعی است طوری که اگر متحرکی (نگهبان) روی این مسیر یک دور کامل را طی کند، تمام نقاط مرز چندضلعی (دیوارهای محوطه) را ببیند (شکل ۲).

^{۱۱}visibility

^{۱۲}watchman route

دسته‌ی دیگری از مسائل که برنامه‌ریزی حرکت را در حضور قیدهای قابلیت دید مطالعه می‌کنند مسائل تعقیب و گریز^{۱۳} هستند. در این مسائل یک یا چند متحرک به عنوان هدف دید^{۱۴} در نظر گرفته می‌شوند و یک یا چند متحرک به عنوان نگهبان یا تعقیب‌کننده قصد یافتن آنها را دارند. برای مثال در [۱۱]، فرض می‌شود که یک یا چند هدف دید در محل نامعلومی در یک چندضلعی ساده قرار دارند و توانایی حرکت دارند. تعدادی نگهبان می‌خواهند محوطه‌ی این چندضلعی را جارو کنند طوری که همواره یک زنجیره‌ی قابلیت دید^{۱۵} ایجاد کنند، به این معنی که در این زنجیره هر دو نگهبان مجاور بتوانند یکدیگر را ببینند. به این ترتیب یک هدف دید نمی‌تواند از این زنجیره بگریزد. هدف مسئله تعیین یک برنامه برای حرکت کم‌ترین تعداد نگهبان‌ها است طوری که بتوانند کل سطح چندضلعی را جارو کنند (که در نهایت به هدف‌های دید برسند). نمونه‌های دیگر مسائل تعقیب و گریز را می‌توان در [۱۲، ۱۳، ۱۴، ۱۵] یافت.

نوع دیگری از مسائل برنامه‌ریزی حرکت با قیدهای قابلیت دید را می‌توان در برنامه‌ریزی مأمورهای خودمختار^{۱۶} دید. این مسائل که تا حدودی مسائل تعقیب و گریز را نیز شامل می‌شوند به برنامه‌ریزی مأمورهایی می‌پردازد که مأموریت آنها با قابلیت دید مرتبط است. یکی از این مسائل یافتن کوتاه‌ترین مسیرهای بازرسی^{۱۷} است [۱۶، ۱۷]. هدف این مسئله یافتن کوتاه‌ترین مسیر از یک مبدأ مشخص به ناحیه‌ای است که بتوان یک هدف دید نقطه‌ای را دید. گروهی که در آزمایشگاه رباتیک دانشگاه استنفورد^{۱۸} فعالیت می‌کنند مفهومی به نام مشاهده‌کننده‌های خودمختار^{۱۹} را معرفی و مطالعه کرده‌اند [۲۱، ۲۲، ۲۰، ۱۹، ۱۸]. سه مسئله‌ی اصلی پیرامون این مشاهده‌کننده‌ها تعریف می‌شوند:

^{۱۳} pursuit-evasion

^{۱۴} عبارت هدف دید به جای target در نظر گرفته شده که از واژه‌ی هدف که به جای goal به کار گرفته شده متفاوت باشد.

^{۱۵} visibility chain

^{۱۶} autonomous agents

^{۱۷} shortest inspection paths

^{۱۸} Stanford

^{۱۹} autonomous observers

ساخت مدل: در این مسئله، هدف قرار دادن یک مشاهده‌کننده‌ی خودمختار در یک محیط است طوری که او بتواند توسط امکانات بینایی مدلی از محیط برای خود بسازد.

یافتن هدف دید: بعد از این که مدل محیط ساخته شد، مشاهده‌کننده‌ی خودمختار به دنبال یک هدف دید که در جای نامعلومی از محیط قرار دارد می‌گردد تا آن را در حیطه‌ی قابلیت دید خود بیاورد.

تعقیب هدف دید: با پیدا کردن هدف دید و با فرض متحرک بودن آن، مشاهده‌کننده‌ی خودمختار باید هدف دید را زیر نظر خود نگه‌دارد.

۳.۱.۱ برنامه ریزی حرکت با قید دیدار

قید قابلیت دید در برنامه‌ریزی حرکت، متحرک را ملزم می‌کند که در طول مسیر خود، ناحیه‌ای از دامنه را که هدف دید از آنجا قابل دیدن است ملاقات کند. این ناحیه در دامنه‌های دوبعدی به صورت یک چندضلعی است و به آن چندضلعی قابلیت دید^{۲۰} گفته می‌شود. حالت عمومی‌تر این قید این است که چندضلعی که مسیر مقید به دیدار آن است لزوماً چندضلعی قابل دید یک نقطه نباشد. نمونه‌هایی از مسائل برنامه‌ریزی حرکت با قیدهایی که دیدار یک یا چند ناحیه از صفحه را الزامی می‌کند در هندسه‌ی محاسباتی مورد مطالعه قرار گرفته‌اند. به عنوان مثال می‌توان از نمونه‌های زیر نام برد:

مسئله‌ی نگهبان باغ‌وحش^{۲۱}: در این مسئله یک چندضلعی ساده (محوطه‌ی باغ‌وحش) داده شده که تعدادی چندضلعی ساده (قفس) در آن قرار دارند که حداقل یک ضلع از آنها با چندضلعی اصلی مشترک است. هدف مسئله یافتن کوتاه‌ترین مسیر بسته‌ای است طوری که مرز تمام چندضلعی‌های داخلی را ملاقات کند ولی داخل آنها نشود [۲۳، ۲۴، ۱۰].

^{۲۰}visibility polygon
^{۲۱}zookeeper's problem

مسئله‌ی مسیر سیاحتی^{۲۲}: این مسئله شبیه به مسئله‌ی نگهبان باغ وحش است با این تفاوت که مسیر

مورد نظر اجازه دارد از داخل چندضلعی‌های داخلی نیز بگذرد [۲۳، ۲۵].

مسئله‌ی برش قطعات^{۲۳}: در این مسئله یک ورقه‌ی فلزی یا چوبی داده شده‌است و مجموعه‌ای از

چندضلعی‌های ساده‌ی از هم جدا که روی ورقه رسم شده‌اند باید از داخل آن بریده شوند.

هدف مسئله، یافتن مسیر بهینه برای حرکت ابزار برش است [۲۶].

مسئله‌ی دیدار دنباله‌ای از چندضلعی‌ها: در این مسئله، تعدادی چندضلعی ساده در صفحه داده

شده‌اند و هدف مسئله یافتن کوتاه‌ترین مسیری است که به ترتیب این چندضلعی‌ها را ملاقات

کند. در حالت خاص‌تر مسئله، مسیر حرکت بین هر دو چندضلعی متوالی با یک چندضلعی

ساده‌ی دیگر که حصار^{۲۴} نامیده می‌شود محدود می‌شود [۲۷].

مسئله‌ی فروشنده‌ی دوره‌گرد با همسایگی^{۲۵}: در این مسئله که تعمیمی از مسئله‌ی فروشنده‌ی

دوره‌گرد است، تعدادی ناحیه از صفحه داده شده‌اند و هدف مسئله یافتن کوتاه‌ترین مسیری

است که تمام این ناحیه‌ها را ملاقات کند [۲۸، ۲۹، ۳۰].

۲.۱ انگیزه‌ی پژوهش

از آنجا که حیطة‌ی برنامه‌ریزی حرکت از یک سو با رباتیک و از سوی دیگر با هندسه‌ی محاسباتی

ارتباط دارد، پژوهش‌های مرتبط با این حیطة نیز معمولاً در یکی از این دو شاخه انجام می‌شود.

پژوهش‌های برخاسته از رباتیک عموماً مسئله را در حالت کاربردی بررسی می‌کنند. هدف عمده‌ی

این پژوهش‌ها معمولاً طراحی یک برنامه‌ریز^{۲۶} برای روبات است طوری که کنشی را که روبات در

^{۲۲} safari route problem

^{۲۳} parts cutting problem

^{۲۴} fence

^{۲۵} TSP with neighborhoods

^{۲۶} planner

هر مقطع زمانی انجام می‌دهد تعیین کند. همانند سایر زمینه‌های کاربردی، در این مسائل، محدودیت‌های عملی از اهمیت زیادی برخوردارند و در مقابل، به دست آوردن یک جواب نسبتاً خوب کافی است. علاوه بر این، در تحلیل کارایی برنامه‌ریزها نیز ارضای محدودیت‌های کاربردی مورد توجه قرار می‌گیرد تا تحلیل پیچیدگی محاسباتی الگوریتمی برنامه‌ریز. به عنوان نمونه، پژوهش‌هایی که در زمینه‌ی مسائل تعقیب و گریز یا مشاهده‌کننده‌های خودمختار انجام شده‌اند (که همگی در آزمایشگاه‌های رباتیک شکل گرفته‌اند) چنین خصوصیتی دارند.

علاوه بر این، در بسیاری از مسائل بررسی شده، متحرک (تعقیب‌کننده یا مشاهده‌کننده‌ی خودمختار) دانش کاملی از شرایط مسئله ندارد. در برخی از مسائل مانند ساخت مدل، متحرک شناختی از شکل هندسی دامنه‌ای که در آن حرکت می‌کند ندارد. این موضوع در بعضی گونه‌های مسائل تعقیب و گریز نیز صادق است. در دسته‌ی دیگری از مسائل مانند یافتن هدف دید متحرک، محل هدف دید نامعلوم است و در مسئله‌ای مانند تعقیب هدف دید متحرک، مسیر حرکت هدف دید ممکن است نامشخص باشد. به این ترتیب، مسائل بررسی شده در این پژوهش‌ها معمولاً در حیطه‌ی مسائل برخط^{۲۷} می‌گنجند.

در مقابل، پژوهش‌های برخاسته از هندسه‌ی محاسباتی، عمدتاً مسئله را از جنبه‌ی نظری بررسی می‌کنند. هدف عمده‌ی این پژوهش‌ها معمولاً یافتن الگوریتم‌هایی برای یافتن مسیر حرکت متحرک است طوری که تابع هزینه‌ی مشخصی (معمولاً طول اقلیدسی مسیر) بهینه شود. ارزیابی این الگوریتم‌ها نیز همانند روش مرسوم، با تحلیل پیچیدگی زمان اجرا و فضای حافظه‌ی مورد نیاز الگوریتم برحسب اندازه‌ی ورودی انجام می‌شود. از آنجا که در حالات کاربردی، تعداد پارامترهای ورودی مسئله که از شرایط موجود در دنیای واقعی تعیین می‌شود زیاد است، عموماً در بررسی نظری، تعدادی از پارامترها ثابت فرض می‌شوند و رده‌ی محدودی از مسائل بررسی می‌شود. انواع مسائل کوتاه‌ترین مسیر در هندسه‌ی محاسباتی یا مسائلی مانند مسیر نگهبان، چنین خصوصیتی

online^{۲۷}

دارند.

همانند تمام حیطه‌های کاربردی، در زمینه‌ی برنامه‌ریزی حرکت نیز پژوهش‌های کاربردی بر مبنای نتایج نظری بنا می‌شوند. نتایجی مانند چارچوب‌های مدل‌سازی نظری مسائل مختلف [۳۱، ۳۲، ۳۳]، رام‌نشدنی^{۲۸} بودن دسته‌ای از مسائل [۳، ۲۷، ۳۰] یا ایده‌های به دست آوردن روش‌های تقریبی برای این‌گونه مسائل [۳۴، ۳۵، ۳۶، ۲۸، ۲۹، ۳۷، ۳۸] نمونه‌هایی از تأثیر پژوهش‌های نظری در به دست آمدن نتایج عملی در این زمینه هستند.

برنامه‌ریزی حرکت با قيود قابلیت دید، زمینه‌ای برآمده از کاربرد است و همان‌گونه که اشاره شد، بیشتر پژوهش‌های انجام شده در این زمینه نیز در آزمایشگاه‌های روباتیک بوده است. انگیزه‌ی اصلی پژوهش‌های انجام شده که در قالب این رساله ارائه شده‌اند بررسی نظری برخی مسائل موجود در این زمینه است. هر چند مسائل کوتاه‌ترین مسیر در هندسه‌ی محاسباتی، مبنایی بسیار غنی برای برنامه‌ریزی حرکت در حالت کلی محسوب می‌شوند، ترکیب این مسائل با قيود قابلیت دید کم‌تر بررسی شده است.

اگر به برنامه‌ریزی حرکت با قيود قابلیت دید از دید بهینه‌سازی نگاه کنیم، عموماً دو نوع تابع هزینه برای بهینه‌سازی وجود دارد: یکی تعداد تعقیب‌کننده‌ها یا مشاهده‌کننده‌های خودمختار و یکی طول مسیر طی شده توسط آنها. نوع اول موضوع مسائل نمایشگاه نقاشی^{۲۹} است و نوع دوم توسط مسائل کوتاه‌ترین مسیر^{۳۰} مطالعه می‌شود.

آنچه در رده مسائل نمایشگاه نقاشی مورد توجه قرار می‌گیرد، یافتن موقعیت‌های مناسب برای یک یا چند نگهبان یا دوربین است طوری که تمام (یا بخش مورد نیاز) یک محیط را تحت پوشش تصویری قرار دهند [۳۹]. این محیط می‌تواند یک چندضلعی ساده یا یک چندضلعی با وجود موانعی در آن باشد. همچنین نگهبان‌ها ممکن است ثابت یا متحرک باشند. اگر این مسئله را

^{۲۸} intractable

^{۲۹} art gallery

^{۳۰} shortest paths

به دید یک مسئله‌ی بهینه‌سازی نگاه کنیم، هدف آن کمینه کردن تعداد نگهبان‌های لازم برای پوشش دامنه است. همچنین ممکن است محاسبه‌ی موقعیت نگهبان‌ها یا استراتژی حرکت آنها در حالتی که متحرک باشند نیز جزء صورت مسئله باشد، اما نکته‌ی با اهمیت این است که هدف اولیه‌ی مسئله کمینه کردن تعداد نگهبان‌ها است. به عنوان نمونه‌هایی از پژوهش‌هایی که مبانی نظری برنامه‌ریزی حرکت با قیود قابلیت دید را با هدف کمینه کردن تعداد نگهبان‌ها بررسی کرده‌اند، می‌توان به [۱۱] و [۱۲] اشاره کرد.

در مقابل، اگر تابع هزینه، بهینه کردن طول مسیر طی شده توسط نگهبان‌ها باشد، میزان پژوهش‌های انجام شده بسیار کم است. شاید دسته مسائل مسیر نگهبان تنها زمینه‌ی نظری بررسی شده است که هدف آن کمینه کردن طول مسیر در حضور قیود قابلیت دید است. با وجود این که این مسئله پایه‌ی نظری دسته‌ای از مسائل است، اما ساختار ایستای آن مانع از این می‌شود که بتوان از آن به عنوان مدلی برای مسائل تعقیب و گریز یا مسائل مربوط به مشاهده‌کننده‌های خودمختار استفاده کرد. منظور از ساختار ایستا این است که قیود قابلیت دید در این مسئله به اضلاع چندضلعی ساده‌ای که دامنه‌ی مسئله را تشکیل می‌دهد محدود می‌شوند. در حالتی که قید قابلیت دید در مسئله در ارتباط با اهدافی مستقل از دامنه تعریف شود، مبنا‌ی نظری مسئله کم‌تر مورد توجه قرار گرفته است. این کمبودی است که پژوهش حاضر سعی در رفع آن دارد.

۳.۱ نتیجه‌های به دست آمده

همان‌گونه که در بخش پیش ذکر شد، هدف این پژوهش بررسی مسائل برنامه‌ریزی حرکت با قید قابلیت دید است در حالی که تابع هزینه، طول مسیر فرض می‌شود. به این منظور، مسائل تعریف شده همگی گونه‌هایی از مسئله‌ی کوتاه‌ترین مسیر در هندسه‌ی محاسباتی هستند. مطابق آنچه در قیود قابلیت دید مرسوم است، موجودی را که باید دیده شود با یک نقطه در نظر می‌گیریم. در

تعریف مسائل هنگامی که از عبارت دامنه^{۳۱} نام می‌بریم، منظور فضایی است که حرکت در آن صورت می‌گیرد. در این پژوهش مسائل مختلفی در دامنه‌های چندضلعی ساده، دامنه‌های چندضلعی گونه (چندضلعی‌های ساده با مانع) و رویه‌های چندوجهی گونه مطالعه شده‌اند. برای دیدن تعریف دقیق این دامنه‌ها فصل ۲ را ببینید.

۱.۳.۱ دیدن هدف دید نقطه‌ای مشخص در طول مسیر

ابتدایی‌ترین مسئله‌ای که آن را مطالعه کرده‌ایم یافتن کوتاه‌ترین مسیر برای حرکت بین دو نقطه است طوری که در طول حرکت یک هدف دید نقطه‌ای موجود در دامنه دیده شود. به طور دقیق‌تر، می‌خواهیم کوتاه‌ترین مسیری را پیدا کنیم که حداقل یک نقطه روی آن یافت شود که هدف دید از آن قابل دید باشد. این مسئله برای سه دامنه‌ی مختلف حل شده است.

دامنه‌ی اول چندضلعی‌های ساده را شامل می‌شود. در این حالت، فرض می‌شود چندضلعی ساده‌ی P به همراه سه نقطه‌ی s و t و p درون آن داده شده‌اند. هدف مسئله، یافتن کوتاه‌ترین مسیر بین s و t است طوری که مسیر کاملاً داخل P واقع باشد و یک نقطه روی آن یافت شود که بتوان p را از آن دید. الگوریتمی که در این حالت ارائه کرده‌ایم مسئله را در زمان $O(n)$ حل می‌کند که n تعداد رأس‌های P است. حافظه‌ی مصرفی این الگوریتم نیز $O(n)$ است. این الگوریتم هم از نظر زمان اجرا و هم از نظر حافظه‌ی مصرفی بهینه است. لازم به ذکر است که حالت بدون قید مسئله (یافتن کوتاه‌ترین مسیر بین s و t) در همین زمان قابل انجام است.

دامنه‌ی دوم دامنه‌های چندضلعی گونه^{۳۲} را شامل می‌شود. به این معنی که ممکن است درون چندضلعی ساده‌ی حالت قبل، تعدادی مانع چندضلعی شکل نیز باشند طوری که یک مسیر نتواند از درون آنها رد شود. در چنین دامنه‌ای با داشتن نقاط s و t و p الگوریتمی ارائه کرده‌ایم که مسئله را در زمان و حافظه‌ی $O(n \log n)$ حل می‌کند که n تعداد کل رئوس موجود در دامنه است. این

^{۳۱} domain

^{۳۲} polygonal domains

الگوریتم هم از نظر زمان اجرا و حافظه‌ی مصرفی بهینه است. در اینجا هم الگوریتم به دست آمده در حالت مقید از نظر مرتبه‌ی زمانی و حافظه‌ی مصرفی با حالت بدون قید یکسان هستند.

برای مطالعه‌ی مسئله در دامنه‌های سه بعدی، به دامنه‌های چندوجهی گونه^{۳۳} بر می‌خوریم. در این دامنه‌ها، تعدادی مانع به شکل چندوجهی در فضا وجود دارند طوری که مسیرها نمی‌توانند از درون آنها عبور کنند. در چنین دامنه‌ای مسئله‌ی کوتاه‌ترین مسیر (بدون قید دیگری) از پیچیدگی محاسباتی بالایی برخوردار است و جزء مسائل NP-سخت به شمار می‌رود [۳]. به دلیل این پیچیدگی، از مطالعه‌ی مسئله در این دامنه چشم‌پوشی می‌کنیم و توجه خود را به حالت خاص رویه‌های چندوجهی گونه^{۳۴} محدود می‌کنیم. در این حالت، یک مسیر روی رویه‌ی یک چندوجهی قرار می‌گیرد. فرض کنید نقاط s و t روی چنین رویه‌ای داده شده باشند. در این حالت هدف دید p می‌تواند روی رویه یا در فضای خارج از آن باشد. یافتن کوتاه‌ترین مسیر بین s و t که p از حداقل یک نقطه از آن قابل دید باشد برحسب این که رویه محدب باشد یا مقعر متفاوت است. اگر پیچیدگی رویه $O(n)$ باشد، زمان اجرای الگوریتم برای حالت محدب $O(n^2)$ و برای حالت مقعر $O(n^3)$ است. پیچیدگی بیشتر حالت مقعر به این خاطر است که ناحیه‌ای از رویه‌ی مقعر که از یک نقطه‌ی دید قابل دید است پیچیدگی بیشتری نسبت به یک رویه‌ی محدب دارد.

نتایج به دست آمده در این مسئله در [۴۰] منتشر شده است. نتایج ابتدایی مربوط به رویه‌های چندوجهی گونه در [۴۱] یافت می‌شود.

۲.۳.۱ دیدن هدف دید نقطه‌ای پرس وجو

گونه‌ی دیگری از مسئله که در این پژوهش بررسی شده است حالتی را شامل می‌شود که محل هدف دید از ابتدا مشخص نیست و می‌خواهیم دامنه را طوری پیش‌پردازش^{۳۵} کنیم که اگر نقطه‌ای

^{۳۳} polyhedral domains

^{۳۴} polyhedral surfaces

^{۳۵} preprocess

به عنوان مکان هدف دید داده شود (در قالب یک پرس و جو)، بتوان کوتاه‌ترین مسیری را که از آن می‌شود هدف دید، دیده شود به سرعت پیدا کرد. این مسئله در دو حالت قابل بررسی است:

مقصد باز^{۳۶}: در این حالت، می‌خواهیم از یک نقطه‌ی مبدأ حرکت کنیم و با طی کم‌ترین مسافت ممکن هدف داده شده در پرس و جو را ببینیم. این مسئله به نام کوتاه‌ترین مسیر بازرسی نیز شناخته می‌شود.

مقصد مشخص^{۳۷}: این حالت مانند حالت مقصد باز است با این تفاوت که می‌خواهیم بعد از دیدن هدف دید، به نقطه‌ی مشخصی که مقصد حرکت است برویم.

ما این مسئله را در چندضلعی‌های ساده مطالعه کرده‌ایم. برای حالت مقصد باز، ورودی را در زمان $O(n^2)$ پیش‌پردازش می‌کنیم و داده‌ساختاری با اندازه‌ی $O(n^2)$ می‌سازیم طوری که بتوان پرس‌وجوها را در زمان $O(\log n)$ پاسخ داد. بعد از انتشار این نتیجه در [۴۲]، کنوثر^{۳۸} و روته^{۳۹} در [۱۷] این نتیجه را به زمان پیش‌پردازش $O(n \log n)$ و اندازه‌ی داده‌ساختار $O(n)$ بهبود دادند.

برای حالت مقصد مشخص، مسئله پیچیده‌تر می‌شود و در نتیجه‌ی حاصل شده، زمان پیش‌پردازش و اندازه‌ی داده‌ساختار ساخته شده هر دو از مرتبه‌ی $O(n^2 + nK)$ هستند که K پارامتری وابسته به ورودی مسئله است که در بدترین حالت از مرتبه‌ی $O(n^2)$ است ولی در خیلی از موارد عملی مقداری کم‌تر است. این نتیجه در [۴۳] ارائه شده است. متأسفانه، بهبود کنوثر و روته قابل تعمیم به حالت مقصد مشخص مسئله نیست.

^{۳۶} open destination

^{۳۷} fixed destination

^{۳۸} Knauer

^{۳۹} Rote

۳.۳.۱ قید دیدار چندضلعی

در مسئله‌ی یافتن کوتاه‌ترین مسیر با قید قابلیت دید از یک هدف دید نقطه‌ای مشخص، می‌خواهیم کوتاه‌ترین مسیری را بین مبدأ و مقصد پیدا کنیم که اشتراک آن با ناحیه‌ی قابل دید هدف دید ناتهی باشد. این ناحیه به شکل یک چندضلعی است که به خاطر این که ناحیه‌ی قابل دید یک نقطه را مشخص می‌کند ویژگی‌های خاصی دارد. ما از این ویژگی‌ها در حل مسئله‌ی یادشده استفاده کرده‌ایم.

اگر ناحیه‌ای را که باید در طول مسیر ملاقات شود از یک چندضلعی قابلیت دید به یک چندضلعی دلخواه تعمیم دهیم، گونه‌ی دیگری از مسئله به دست می‌آید که به آن مسئله‌ی کوتاه‌ترین مسیر با قید دیدار چندضلعی^۴ می‌گوییم. در این حالت، می‌خواهیم کوتاه‌ترین مسیری بین دو نقطه را پیدا کنیم طوری که اشتراک مسیر با یک چندضلعی هدف داده شده ناتهی باشد.

می‌توان مسئله را به حالت کلی‌تری نیز تعمیم داد که در آن نقطه‌ی مبدأ و چندضلعی هدف مشخص هستند ولی نقطه‌ی مقصد از پیش داده نشده و در قالب پرس‌وجو داده می‌شود. به این ترتیب، می‌خواهیم ورودی مسئله شامل دامنه‌ی ورودی، نقطه‌ی مبدأ و چندضلعی هدف را پیش‌پردازش کنیم طوری که با دریافت یک نقطه‌ی مقصد در یک پرس‌وجو، بتوانیم در زمان لگاریتمی کوتاه‌ترین مسیر از مبدأ به مقصد را پیدا کنیم طوری که این مسیر با چندضلعی هدف اشتراک داشته باشد.

ما این مسئله را برای دو دامنه‌ی چندضلعی ساده و دامنه‌ی چندضلعی گونه مطالعه کرده‌ایم. برای چندضلعی ساده و در حالتی که چندضلعی هدف محدب باشد، در زمان $O(n)$ می‌توان داده‌ساختاری با اندازه‌ی $O(n)$ ساخت که به پرس‌وجوی مورد نظر در زمان $O(\log n)$ پاسخ دهد. در این مسئله n مجموع تعداد رأس‌های دامنه و چندضلعی هدف فرض می‌شود. این نتیجه در [۴۴] منتشر شده است.

^۴ polygon visit constraint

برای حالتی که چندضلعی هدف نامحدوب باشد یا در دامنه مانع وجود داشته باشد، مسئله در زمان و حافظه‌ی $O(n \log n)$ پیش‌پردازش می‌شود تا زمان پرس‌وجوی $O(\log n)$ به دست آید. نمونه‌ی ابتدایی این راه‌حل در [۴۵] منتشر و نمونه‌ی کامل‌شده‌ی آن در [۴۶] ارائه شده است. نتیجه‌های به دست آمده برای این مسئله برای هر دو دامنه‌ی یاد شده از نظر زمان اجرا در بدترین حالت بهینه هستند.

۴.۳.۱ خلاصه‌ی روش‌ها

بازه‌های بهینگی

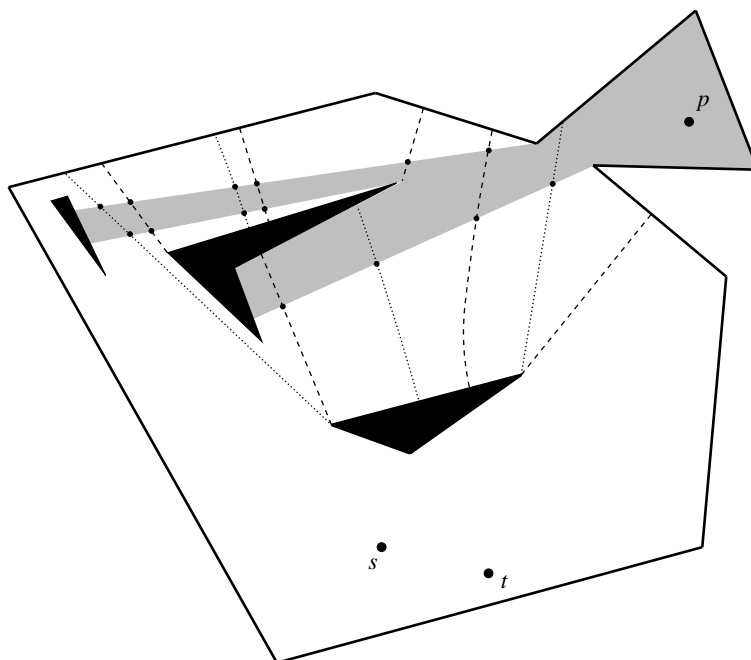
در تمام مسائل بررسی شده، می‌خواهیم یک کوتاه‌ترین مسیر در دامنه پیدا کنیم که با یک ناحیه که فعلاً آن را ناحیه‌ی هدف^{۴۱} می‌نامیم اشتراک داشته باشد. این ناحیه ممکن است ناحیه‌ی قابلیت دید یک هدف دید نقطه‌ای باشد (قید قابلیت دید) یا یک چندضلعی دلخواه (قید دیدار). به راحتی می‌توان مشاهده کرد که اگر مبدأ و مقصد درون ناحیه‌ی هدف نباشند (که حالت بدیهی مسئله محسوب می‌شود)، دست کم یکی از نقاط اشتراک مسیر با ناحیه‌ی هدف روی مرز ناحیه‌ی هدف رخ می‌دهد. در این حالت، هر دو بخش مسیر از مبدأ به آن نقطه و از آن نقطه تا مقصد مسیرهای بهینه هستند (یعنی کوتاه‌تر نمی‌شوند). ایده‌ی اولیه‌ای که به ذهن می‌رسد این است که تمام نقاط روی مرز ناحیه‌ی هدف را در نظر بگیریم و نقطه‌ای را که مجموع طول کوتاه‌ترین مسیرهای آن به نقاط مبدأ و مقصد کمینه باشد پیدا کنیم. به این نقطه، نقطه‌ی تماس بهینه^{۴۲} می‌گوییم.

برای این که بتوانیم جستجو برای نقطه‌ی تماس بهینه را در دامنه‌ی گسسته‌ای انجام دهیم از مفهوم بازه‌های بهینگی^{۴۳} استفاده می‌کنیم. این بازه‌ها که بر اساس ساختار ترکیبیاتی کوتاه‌ترین مسیرهای نقاط آنها به مبدأ و مقصد تعریف می‌شوند دارای این خاصیت هستند که برای هر بازه در

^{۴۱} target region

^{۴۲} optimal contact point

^{۴۳} intervals of optimality



شکل ۳: بخش خاکستری چندضلعی، ناحیه‌ای است که از p قابل دید است. چندضلعی‌های سیاه مانع‌ها هستند. نقاط انتهایی بازه‌های بهینگی با دایره‌های کوچک روی مرز p مشخص شده‌اند.

زمان ثابت می‌توان نقطه‌ای را روی آن به دست آورد که بین نقاط آن بازه، کم‌ترین مجموع فاصله‌ها را از مبدأ و مقصد داشته باشد. به این ترتیب، با بررسی تمام بازه‌های بهینگی می‌توان نقطه‌ی تماس بهینه را به دست آورد (شکل ۳).

چالش اصلی در به‌کارگیری این چارچوب این است که تعداد این بازه‌ها در بیشتر گونه‌های مسئله در بدترین حالت از مرتبه‌ی $\Omega(n^2)$ است و محاسبه‌ی کامل آنها عموماً زمانی از مرتبه‌ی $\Omega(n^2 \log n)$ نیاز دارد. برای کم‌کردن این زمان از مفهومی به نام دروازه^{۴۴} استفاده می‌کنیم که بازه‌هایی را که ممکن است شامل نقطه‌ی تماس بهینه باشند به $O(n)$ بازه محدود می‌کند. به این ترتیب، مشکل اصلی در حل این مسائل یافتن کارای بازه‌های بهینگی محدود شده به دروازه خواهد بود.

gate^{۴۴}

جاروی شعاعی

برای حل مسئله در حالتی که ناحیه‌ی هدف، چندضلعی قابلیت دید یک هدف دید نقطه‌ای است که به صورت پرس وجو مشخص می‌شود، فقط چندضلعی‌های ساده را بررسی کرده‌ایم. در این دامنه، بخشی از ناحیه‌ی هدف که ممکن است شامل نقطه‌ی تماس بهینه باشد به صورت یک پاره‌خط است که به آن پنجره^{۴۵} گفته می‌شود. در این گونه‌ی مسئله، تمام پاره‌خط‌هایی را که ممکن است پنجره‌ی یک نقطه‌ی پرس وجوی فرضی باشند پردازش می‌کنیم و آنها را در داده‌ساختاری دسته‌بندی می‌کنیم. هنگام دریافت نقطه‌ی پرس وجو، پنجره‌ی مربوط را محاسبه می‌کنیم و آن را در داده‌ساختار یادشده پیدا می‌کنیم. در این داده‌ساختار اطلاعاتی ذخیره کرده‌ایم که با استفاده از آن می‌توان محل نقطه‌ی تماس بهینه را در زمان ثابت به دست آورد.

پیش پردازش تمام پاره‌خط‌هایی که ممکن است پنجره‌ی یک نقطه‌ی پرس وجو باشند از طریق یک جاروی شعاعی^{۴۶} حول برخی از رأس‌های چندضلعی صورت می‌گیرد. در جریان جارو، رفتار نقطه‌ی بهینه را تعقیب می‌کنیم و پنجره‌هایی ممکن را بر اساس بازه‌ی بهینگی شامل نقطه‌ی مذکور دسته‌بندی می‌کنیم و در داده‌ساختار گفته شده قرار می‌دهیم.

بازتابش جبهه‌ی موج

یکی از نتایج به دست آمده در این پژوهش، چارچوبی است برای ساختن یک زیرتقسیم از دامنه طوری که با داشتن یک مبدأ و یک ناحیه‌ی هدف، بتوان فضای آزاد را به ناحیه‌هایی تقسیم کرد که برای یک ناحیه، کوتاه‌ترین مسیر از مبدأ به تمام نقاط آن ناحیه که با ناحیه‌ی هدف اشتراک دارند ساختاری یکسان داشته باشند. به این ترتیب با دریافت یک نقطه‌ی مقصد در قالب پرس وجو، می‌توان چنین مسیری را به سرعت به دست آورد. این چارچوب که تعمیمی از الگوی انتشار

^{۴۵} window

^{۴۶} radial sweep

جبهه‌ی موج برای ساخت نقشه‌ی کوتاه‌ترین مسیر^{۴۷} است، یک الگوی کلی ارائه می‌کند که برای هر گونه ناحیه‌ی هدف و هر دامنه‌ای قابل استفاده است.

ایده‌ی اصلی این روش انتشار یک جبهه‌ی موج از مبدأ است تا به ناحیه‌ی هدف برسد. در این حالت، یک نسخه از جبهه‌ی موج اولیه به راه خود ادامه می‌دهد و یک نسخه از آن توسط بخشی از مرز ناحیه‌ی هدف که برخورد با آن صورت گرفته است، بازتابیده می‌شود. در عمل، ناحیه‌ی هدف نقشی شبیه به یک مبدأ مجازی را بازی می‌کند که بخش‌های مختلف آن بسته به این که فاصله‌ی آنها تا مبدأ اصلی چقدر باشد با تأخیرهای متفاوت شروع به انتشار موجک می‌کنند. در طراحی چارچوب بازتابش جبهه‌ی موج از زیر تقسیم‌ی به نام زیر تقسیم بازتابی^{۴۸} استفاده شده است. این زیر تقسیم علاوه بر کاربرد در حل مسئله‌ی یاد شده در محاسبه‌ی کوتاه‌ترین مسیر با قید دیدار بر مبنای گراف قابلیت دید نیز کاربرد دارد.

۴.۱ طرح رساله

فصل دوم این رساله به بیان تعریف‌ها و مفاهیم پایه در حیطه‌های کوتاه‌ترین مسیر و قابلیت دید می‌پردازد. علاوه بر این، کلیات برخی روش‌های مهم در این زمینه مانند انتشار جبهه‌ی موج یا محاسبه‌ی کوتاه‌ترین مسیرها در چندضلعی‌های ساده مرور می‌شوند. در فصل سوم مسئله‌ی یافتن کوتاه‌ترین مسیر با قید قابلیت دید یک هدف دید نقطه‌ای مشخص می‌پردازد. این مسئله در سه دامنه‌ی چندضلعی‌های ساده، دامنه‌های چندضلعی‌گونه و رویه‌های چندوجهی‌گونه بررسی می‌شود. از آنجا که روش استفاده شده در این سه دامنه شباهت زیادی با هم دارد و متکی بر مفهوم بازه‌های بهینگی است، کلیات این روش که جنبه‌ی عمومی دارد پیش از پرداختن به جزئیات این سه دامنه ارائه شده است. در فصل چهارم گونه‌ای از مسئله بررسی می‌شود که در آن هدفی که باید دیده شود

^{۴۷} shortest path map

^{۴۸} reflective subdivision

از پیش مشخص نیست و به عنوان پرس وجود داده می شود. این مسئله در دو گونه‌ی مقصد باز و مقصد ثابت در دو بخش جدا مطالعه می شود. فصل پنجم قید دیدار چندضلعی را بررسی می کند. این قید ابتدا در دامنه‌ی چندضلعی‌های ساده بررسی می شود. برای مطالعه‌ی این قید در دامنه‌های چندضلعی گونه زیرتقسیم بازتابی به عنوان یک مفهوم بررسی می شود و بعد از آن الگوی بازتابش جبهه‌ی موج و استفاده از گراف قابلیت دید به عنوان دو روش برای حل این مسئله در دامنه‌های چندضلعی گونه بررسی می شوند. سرانجام، در فصل ششم خلاصه‌ی نتایج به دست آمده به همراه راه‌هایی برای ادامه و گسترش این پژوهش ارائه می شوند.

مفاهیم پایه و کارهای مرتبط

در این فصل برخی از مفاهیم پایه را در زمینه‌ی کوتاه‌ترین مسیرها و قابلیت دید معرفی خواهیم کرد. برخی از الگوریتم‌های موجود را که در روش‌های ما تأثیر مستقیم دارند نیز مرور خواهیم کرد. همچنین به بررسی تعدادی از الگوریتم‌هایی که در حیطه‌ی برنامه‌ریزی حرکت با قيود قابلیت دید جای می‌گیرند می‌پردازیم.

۱.۲ تعریف‌های ابتدایی

۱.۱.۲ دامنه‌های دوبعدی

یک مسیر تصویری پیوسته از یک بازه است. یک مسیر چندضلعی گونه^۱ از تعدادی متناهی پاره‌خط (یال یا پیوند) تشکیل شده است که تعدادی نقطه (رأس) را به هم متصل می‌کند. طول یک مسیر، عددی نامنفی متناظر با مسیر است که هزینه‌ی کل آن را طبق یک معیار تعریف شده بیان می‌کند. به جز مواردی که صریحاً ذکر شود، طول مسیر، طول اقلیدسی آن فرض می‌شود که مجموع طول پاره‌خط‌های تشکیل‌دهنده‌ی مسیر است.

بین تمام مسیرهای ممکن (آنهایی که محدودیت‌های دامنه را ارضاء می‌کنند)، مسیری که

^۱ polygonal

کمترین طول را داشته باشد یک کوتاه‌ترین مسیر نامیده می‌شود. این مسیر را مسیر بهینه نیز می‌گویند. مسئله‌ی کوتاه‌ترین مسیر یک معیار جدید تعریف می‌کند که به آن معیار کوتاه‌ترین مسیر می‌گویند و طبق آن، فاصله‌ی بین دو نقطه، طول یک کوتاه‌ترین مسیر بین آن دو نقطه است. به این معیار، معیار فاصله‌ی ژئودزیک^۲ نیز می‌گویند.

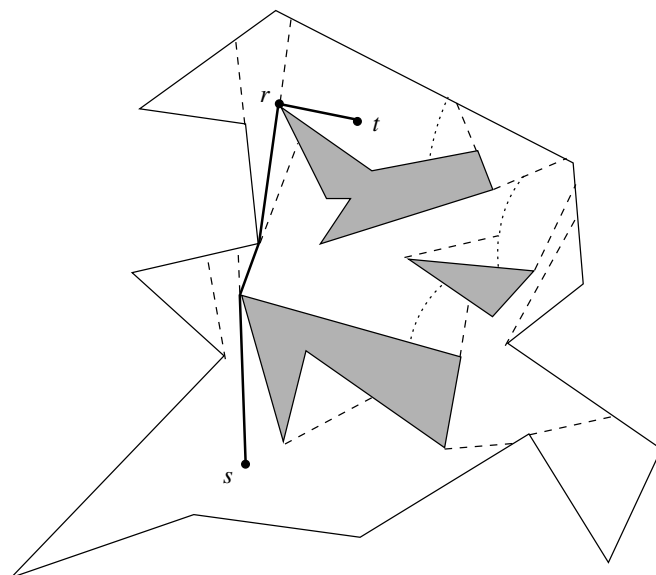
یک چندضلعی ساده P با n رأس، یک ناحیه‌ی بسته‌ی همبند ساده است که مرز آن از n پاره‌خط (یال) تشکیل شده است. نقاط انتهایی این پاره‌خط‌ها را رأس‌های P می‌نامیم. به رأسی که یال‌های حادث به آن در داخل چندضلعی زاویه‌ای بیشتر از 180° بسازند رأس انعکاسی می‌گوییم. یک دامنه‌ی چندضلعی گونه P با n رأس و h حفره یک ناحیه‌ی بسته‌ی همبند چندگانه است که مرز آن از اجتماع n پاره‌خط تشکیل شده است که $h + 1$ دور چندضلعی گونه تشکیل می‌دهند. (به این ترتیب، یک چندضلعی ساده، یک دامنه‌ی چندضلعی گونه با $h = 0$ است.) ما عبارت مانع را به هر ناحیه‌ای از صفحه اطلاق خواهیم کرد که مسیرها حق ورود به داخل آنها را ندارند. مکمل مجموعه مانع را فضای آزاد می‌نامیم. اگر فضای آزاد یک دامنه‌ی چندضلعی گونه P باشد، مانع $h + 1$ مؤلفه‌ی همبند مکمل P هستند (h حفره و وجهی که تا بی‌نهایت ادامه دارد).

یک مثلث‌بندی^۳ از P ، تجزیه‌ی P به تعدادی مثلث است که اشتراک هر دو مثلث یا تهی است یا فقط در یک رأس یا فقط در یک یال است. یک مثلث‌بندی برای یک چندضلعی ساده در $O(n)$ [۴۷] و برای یک دامنه‌ی چندضلعی گونه در $O(n \log n)$ [۴۸] یا در $O(n + h \log^{1+\epsilon} h)$ [۴۹] قابل محاسبه است.

یک کوتاه‌ترین مسیر محلی یا مسیر بهینه‌ی محلی، مسیری است که نتوان طول آن را با تغییری کوچک کمتر کرد. منظور از تغییر کوچک، تغییری است که ساختار ترکیباتی مسیر را ثابت نگه‌دارد. به این معنی که دنباله مثلث‌هایی که مسیر آنها را ملاقات می‌کند (به ازای یک مثلث‌بندی دلخواه از P) ثابت بماند.

^۲ geodesic

^۳ triangulation



شکل ۲: نقشه‌ی کوتاه‌ترین مسیر در یک دامنه‌ی چندضلعی گونه r - ریشه‌ی سلول شامل t است.

است، مثلاً تعداد یال‌های مسیر).

یکی از راه‌های پاسخ دادن به پرس‌وجوی تک-مبدأ ساختن یک نقشه‌ی کوتاه‌ترین مسیر $SPM(s, P)$ است (شکل ۲). این نقشه، تجزیه‌ی فضای آزاد به تعدادی ناحیه (سلول) است با توجه به ساختار ترکیبیاتی کوتاه‌ترین مسیرها از مبدأ s به نقطه‌های ناحیه‌ها. برای یک دامنه‌ی چندضلعی گونه‌ی P ، $SPM(s, P)$ تجزیه‌ی P به مجموعه‌ای از سلول‌ها است طوری که برای هر نقطه داخل یک سلول مثل t ، دنباله‌ی رأس‌های موانع روی یک کوتاه‌ترین مسیر از s به t یکسان است. آخرین رأس مانع روی یک کوتاه‌ترین مسیر از s به t ریشه‌ی سلول شامل t نامیده می‌شود. به وضوح ریشه‌ی یک سلول همیشه روی مرز آن است. هر سلول نسبت به ریشه‌ی خود یک چندضلعی ستاره‌شکل^۷ است، به این معنی که پاره‌خط واصل ریشه و هر نقطه داخل سلول به طور کامل داخل سلول واقع می‌شود. لازم به توضیح است که در جایی که ابهامی ایجاد نشود به جای نمادهای $SPM(s, P)$ و $SPT(s, P)$ به ترتیب از $SPM(s)$ و $SPT(s)$ استفاده می‌کنیم.

ما معمولاً برای هر رأس v از P ، فاصله‌ی ژئودزیک s از آن را که با $d(s, v)$ نمایش داده

^۷start-shaped polygon

می شود نگهداری می کنیم. علاوه بر این، اشاره گری به رأس قبلی v در کوتاه ترین مسیر از s به v نیز ذخیره می شود. (به کمک این اشاره گرها $SPT(s, P)$ مشخص می شود.) توجه داشته باشید که v روی مرز سلول ستاره شکل به ریشه ی رأس قبلی v قرار دارد. مرز سلول ها از بخش هایی از یال های مواع، پاره خط های امتداد (امتداد یال های گراف قابلیت دید که حادث به ریشه هستند) و خم های منصف^۸ تشکیل شده است. خم های منصف مکان هندسی نقاطی هستند که فاصله ی ژئودزیک آنها از دو ریشه ی u و v یکسان است. به عبارت دیگر $d(s, v) + |vp| = d(s, u) + |up|$ که نماد $|xy|$ نمایانگر فاصله ی اقلیدسی دو نقطه ی x و y است. این خم ها در حالت کلی خم های هذلولوی هستند و پاره خط های امتداد را می توان به عنوان حالت حدی آنها در نظر گرفت.

اگر $SPM(s)$ را برای جایابی نقطه^۹ پیش پردازش کنیم $[5^\circ]$ ، پرس و جوهای تک-مبدأ را می توان با جایابی نقطه ی پرس و جوی t در نقشه ی یادشده به طرز کارایی پاسخ داد: اگر t در سلولی با ریشه ی r قرار داشته باشد، فاصله ی ژئودزیک t از s از رابطه ی $d(s, t) = d(s, r) + |rt|$ به دست می آید. یک کوتاه ترین مسیر بین این دو نقطه نیز از دنبال کردن اشاره گرها به رئوس قبلی در زمان $O(k)$ قابل محاسبه خواهد بود که k تعداد رئوس روی مسیر است.

در یک مسئله ی پرس و جوی دو نقطه ای^{۱۰}، هدف ما ساختن داده ساختاری است که توسط آن بتوان به پرس و جوهای پاسخ داد که در آن دو نقطه ی s و t داده شده اند و طول یک کوتاه ترین مسیر بین آنها خواسته شده است. طبق معمول، در صورتی که خود مسیر نیز خواسته شود، انتظار این است که با پرداخت هزینه ای متناسب با پیچیدگی آن بتوان یک کوتاه ترین مسیر را نیز پیدا کرد.

bisector curves^۸

point-location^۹

two-point query^{۱۰}

۲.۱.۲ ابعاد بالاتر

از آنجا که بیشترین حجم پژوهش در مسئله‌ی کوتاه‌ترین مسیرها در ابعاد بیشتر از دو روی سه بعد متمرکز شده، ما نیز به بیان نتایج این حالت بسنده می‌کنیم.

یک دامنه‌ی چندوجهی گونه‌ی زیرمجموعه‌ی همبندی از R^3 مثل P است که مرز آن از اجتماع تعداد متناهی مثلث تشکیل شده است. مکمل P از تعدادی مؤلفه‌ی همبند (چندوجهی گونه) تشکیل شده که هر یک از آنها را مانع می‌نامیم. یک رویه‌ی چندوجهی گونه اجتماع تعداد همبندی از وجوه چندضلعی گونه است طوری که اشتراک هر دو چندضلعی یا تهی است، یا فقط در یک رأس و یا در فقط یک یال است. هر یال نیز دقیقاً به دو چندضلعی متعلق است.

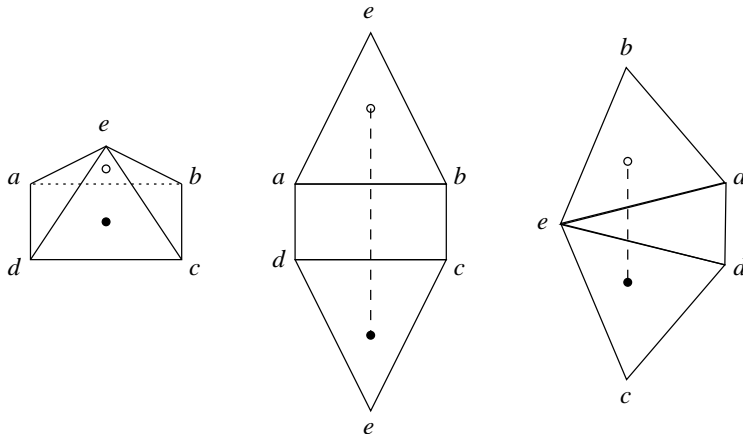
در این بخش از n به عنوان تعداد کل یال‌های دامنه استفاده می‌کنیم. بدون از دست دادن کلیت می‌توانیم فرض کنیم تمام وجه‌های یک رویه‌ی چندوجهی گونه مثلث هستند. در غیر این صورت، می‌توان با استفاده از الگوریتم‌های مثلث‌بندی چندضلعی با اضافه کردن $O(n)$ یال و وجه، این شرط را برقرار کرد.

دو وجه f و f' از یک رویه‌ی چندوجهی گونه را مجاور یالی^{۱۱} می‌گوییم اگر یک یال مثل e بین آنها مشترک باشد. یک دنباله از وجه‌های مجاور یالی دنباله‌ی یک یا بیشتر از وجه‌ها مثل $F = (f_1, f_2, \dots, f_{k+1})$ است طوری که f_i مجاور یالی f_{i+1} باشد (یال e_i بین آنها مشترک باشد). در این صورت به دنباله $E = (e_1, e_2, \dots, e_k)$ یک دنباله یالی^{۱۲} می‌گوییم و رأسی از f_1 را که مقابل e_1 است ریشه‌ی E می‌نامیم.

اگر دو وجه f و f' مجاور یالی باشند و یال e بین آنها مشترک باشد، می‌توانیم نقاط f' را حول خطی که از e می‌گذرد دوران دهیم طوری که در صفحه‌ی شامل f قرار بگیرند. این دوران طوری صورت می‌گیرد که نقاط f' نسبت به نقاط f ، در طرف مقابل e قرار می‌گیرد. مجموعه‌ی

^{۱۱} edge-adjacent

^{۱۲} edge sequence



شکل ۳: چپ: یک چندوجهی محدب با دو نقطه روی رویه‌ی آن - نقطه‌ی توپر روی وجه جلو (مثلث edc) و نقطه‌ی توخالی روی وجه پشتی (مثلث eab) قرار دارد. وسط: یک مسیر ژئودزیک بین دو نقطه که دنباله یال (dc, ab) را متصل می‌کند. راست: یک مسیر ژئودزیک بین دو نقطه که دنباله یال (ed, ea) را متصل می‌کند. این مسیر از مسیر شکل وسط کوتاه‌تر است.

حاصل، بی تا شده f' ی f روی f نام دارد. به همین ترتیب می‌توانیم بی تا شده‌ی دنباله یال $E = (e_1, e_2, \dots, e_k)$ را به این شکل تعریف کنیم: f_1 را حول e_1 دوران می‌دهیم تا صفحه‌ی آن با صفحه‌ی f_2 یکی شود، f_1 و f_2 را حول e_2 دوران می‌دهیم تا صفحه‌ی آنها با صفحه‌ی f_3 یکی شود. این کار را تا آنجا ادامه می‌دهیم که تمام وجه‌های f_1, f_2, \dots, f_k در صفحه‌ی f_{k+1} قرار بگیرند.

یک مسیر π روی یک رویه‌ی چندوجهی گونه دنباله یال $E = (e_1, e_2, \dots, e_k)$ را متصل می‌کند اگر از پاره‌خط‌هایی تشکیل شده باشد که نقاط داخلی از e_1, e_2, \dots, e_k را به همان ترتیب به هم وصل کند.

یک مسیر روی یک رویه‌ی چندوجهی گونه‌ی P ژئودزیک نامیده می‌شود اگر به طور محلی بهینه باشد، به این معنی که نتوان با تغییرات کوچک طول آن را کمتر کرد (شکل ۳). لم‌های زیر برخی از خواص این گونه مسیرها را بیان می‌کنند. اثبات این لم‌ها در [۵۱] یافت می‌شوند.

لم ۱.۲ اگر π یک مسیر ژئودزیک باشد که دنباله یال E را متصل می کند، تصویر π روی بی تا شده ی E یک پاره خط راست است.

لم ۲.۲ اگر π یک مسیر ژئودزیک روی یک چندوجهی محدب باشد، به جز نقاط انتهایی نمی تواند شامل رأس های چندوجهی باشد.

لم ۳.۲ شکل کلی یک مسیر ژئودزیک مسیری است که به تناوب از رأس ها و دنباله یال های احتمالاً تهی می گذرد طوری که تصویر مسیر روی بی تا شده ی هر یک از دنباله یال ها یک پاره خط راست است. همچنین، زاویه ی دو بخش مسیر که در دو طرف رأس ها قرار می گیرند مساوی π یا بیشتر از آن است. شکل کلی یک مسیر بهینه مانند یک مسیر ژئودزیک است با این تفاوت که هیچ یالی در بیش از یک دنباله یال ظاهر نمی شود و هیچ دنباله یالی نیز یال تکراری ندارد.

لم ۴.۲ اگر $\pi(x)$ و $\pi(y)$ دو مسیر بهینه از s به x و y روی یک رویه ی چندوجهی گونه ی P باشند، این دو مسیر فقط در رأس های P امکان تقاطع دارند و در صورت تقاطع در رأس v ، طول زیرمسیر $\pi(x)$ از s به v مساوی طول زیرمسیر $\pi(y)$ از s به v است.

۲.۲ کوتاه ترین مسیرها در یک چندضلعی ساده

در این بخش به بررسی مسئله ی یافتن کوتاه ترین مسیر درون یک چندضلعی ساده با معیار اقلیدسی می پردازیم. بر اساس نامساوی مثلثی به سادگی می توان صحت دو ادعای زیر را تحقیق کرد:

گزاره ۱.۲ کوتاه‌ترین مسیر بین دو نقطه s و t در یک چندضلعی ساده‌ی P یکتا است. در نتیجه $SPT(s, P)$ نیز یکتا است.

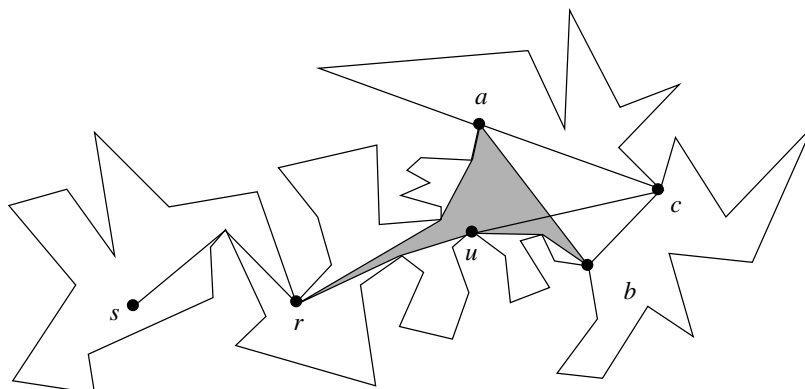
گزاره ۲.۲ شکل کلی کوتاه‌ترین مسیر بین دو نقطه در یک چندضلعی ساده‌ی P ، مسیری چندضلعی گونه است که فقط در رأس‌های انعکاسی P می‌شکند. علاوه بر این، در هر یک از این شکست‌ها، زاویه‌ای که مسیر در داخل P می‌سازد بزرگتر از 180° است.

الگوریتم‌های کارا برای یافتن کوتاه‌ترین مسیر در چندضلعی ساده همگی بر داشتن یک مثلث‌بندی از چندضلعی است که در زمان خطی قابل محاسبه است [۴۷]. گراف دوگان مثلث‌بندی یک درخت خواهد بود. با در نظر گرفتن مثلث‌هایی که شامل نقاط s و t هستند، یک مسیر در این درخت مشخص می‌شود. به دنباله مثلث‌هایی که متناظر با این مسیر هستند آستین^{۱۴} می‌گوییم. در [۵۲، ۵۳] نشان داده شده است که چگونه با بررسی مثلث‌ها به ترتیب در امتداد آستین می‌توان در زمان $O(n)$ کوتاه‌ترین مسیر را محاسبه کرد.

در یک قدم کلی از الگوریتم، مثلث abc را در نظر می‌گیریم. دنباله مثلث‌های قبلی آستین را می‌توان به ساختاری که به آن قیف^{۱۵} اطلاق می‌شود کاهش داد. این ساختار از کوتاه‌ترین مسیر از s به یک رأس مثل r (ریشه‌ی قیف) و دو کوتاه‌ترین مسیر (مقعر) تشکیل شده است که r را به دو انتهای پاره‌خط ab وصل می‌کند که دهانه‌ی قیف نامیده می‌شود (شکل ۴). هنگام بررسی مثلث abc ، قیف را با توجه به کوتاه‌ترین مسیر از r به c ، به دو قیف دیگر تجزیه می‌کنیم. این کار با متصل کردن c به رأسی مانند u که روی یکی از دو دیواره‌ی مقعر قیف قرار دارد انجام می‌شود. پس از این، کار را با یکی از دو مثلی ادامه می‌دهیم که مجاور ac یا bc است و در امتداد آستین قرار دارد. کاری را که برای یافتن رأس u انجام می‌دهیم به راحتی می‌توانیم با کنار گذاشتن رأس‌هایی که بررسی

^{۱۴}sleeve

^{۱۵}funnel



شکل ۴: محاسبه‌ی کوتاه‌ترین مسیر در چندضلعی ساده - در لحظه‌ی بررسی مثلث abc ، قیف با رنگ خاکستری نشان داده شده است که توسط پاره‌خط uc به دو قیف تجزیه می‌شود.

نخواهند شد جبران کنیم. به این ترتیب کوتاه‌ترین مسیر بین s و t در زمان $O(n)$ قابل محاسبه است.

برای پاسخ دادن به پرس‌وجوهای تک-نقطه باید نقشه‌ی کوتاه‌ترین مسیر از مبدأ s را محاسبه

کنیم. این نقشه ساختار ساده‌ای دارد. مرز بین سلول‌ها در این نقشه وترهای P هستند که از امتداد

برخی از یال‌های گراف قابلیت دید به دست می‌آیند. گویاس^{۱۶} و دیگران در [۵۴] الگوریتمی با

زمان اجرای $O(n)$ برای محاسبه‌ی این نقشه ارائه کرده‌اند. این الگوریتم برای این که بتواند قیف را

به طور کارایی تجزیه‌کند از داده‌ساختار درخت جستجوی انگشتی^{۱۷} [۵۵] استفاده می‌کند.

استفاده از چنین داده‌ساختاری به خاطر این است که در این حالت نمی‌توان یکی از دو طرف قیف

تجزیه‌شده را کنار گذاشت. از آنجا که در این مسئله، نقطه‌ی مقصد هنگام پیش‌پردازش مشخص

نیست، مفهوم آستین به تعبیری که در مسئله‌ی قبل ارائه شد قابل تعریف نیست. به جای آن،

الگوریتم این مسئله یک پیمایش عمق-اول روی درخت دوگان مثلث‌بندی انجام می‌دهد و تمام

«آستین‌های ممکن» به این طریق مورد بررسی قرار می‌گیرد.

به این ترتیب، $SPM(s)$ در یک داده‌ساختار مناسب برای جایابی نقطه ذخیره می‌شود و

می‌توان به پرس‌وجوهای تک-مبدأ در زمان $O(\log n)$ پاسخ داد.

^{۱۶} Guibas

^{۱۷} finger search tree

این نتایج به حالت پرس وجوی دو نقطه‌ای نیز تعمیم داده شده است. گویاس و هرشبرگر^{۱۸} [۵۶] نشان داده‌اند که چگونه می‌توان یک چندضلعی ساده را در زمان $O(n)$ پیش‌پردازش کرد و داده‌ساختاری با اندازه‌ی $O(n)$ ساخت که برای هر دو نقطه‌ی مورد پرس وجو بتواند در زمان $O(\log n)$ طول کوتاه‌ترین مسیر بین آن دو را برگرداند. هرچند این نتیجه به نوعی از نتیجه‌ی قبلی برای پرس وجوی تک مبدأ قوی‌تر محسوب می‌شود، اما به دلیل این که زیرتقسیمی از چندضلعی ارائه نمی‌کند (به تعبیر دیگر نقشه‌ای تولید نمی‌کند)، برای استفاده در الگوی ما مناسب نیست.

۳.۲ کوتاه‌ترین مسیرها در یک دامنه‌ی چندضلعی گونه

برخلاف حالت چندضلعی ساده، در یک دامنه‌ی چندضلعی گونه مسیر بهینه‌ی محلی یکتایی وجود ندارد. در حقیقت ممکن است بین هر دو نقطه تعدادی نمایی مسیر بهینه‌ی محلی وجود داشته باشد. همین امر مسئله‌ی یافتن یک کوتاه‌ترین مسیر در این دامنه را پیچیده‌تر از حالت چندضلعی ساده می‌کند. توجه داشته باشید که در این حالت کوتاه‌ترین مسیر لزوماً یکتا نیست و هدف مسئله در درجه‌ی اول یافتن طول یک کوتاه‌ترین مسیر و در صورت نیاز یافتن یک کوتاه‌ترین مسیر بین مبدأ و مقصد است.

برای حل این مسئله از دو روش کلی استفاده می‌شود: جستجوی گراف قابلیت دید و

جستجوی مبتنی بر روش دایکسترای پیوسته^{۱۹}.

^{۱۸}Hershberger

^{۱۹}continuous Dijkstra

۱.۳.۲ جستجوی گراف قابلیت دید

در این روش فرض می‌شود که نقاط مبدأ و مقصد s و t دو رأس از دامنه‌ی چندضلعی P هستند (در غیر این صورت می‌توان دو مانع نقطه‌ای در این دو نقطه در نظر گرفت). با استدلال‌های ساده‌ای مبتنی بر بهینگی محلی می‌توان صحت گزاره‌ی زیر را تحقیق کرد:

گزاره ۳.۲ هر مسیر بهینه‌ی محلی بین دو نقطه‌ی s و t در یک دامنه‌ی چندضلعی گونه‌ی P روی گراف قابلیت دید $VG(P)$ قرار دارد؛ چنین مسیری از اجتماع پاره‌خط‌هایی تشکیل شده است که جفت رأس‌های قابل دید را به هم وصل می‌کنند.

بنابراین اگر گراف قابلیت دید P ساخته شود برای یافتن کوتاه‌ترین مسیر بین s و t کافی است الگوریتم کوتاه‌ترین مسیر دایکسترا را روی گراف یادشده اعمال کنیم.

الگوریتم‌های متعددی برای ساخت گراف قابلیت دید ارائه شده‌اند که از بین آنها می‌توان به الگوریتم ارائه شده در [۵۷] اشاره کرد که این مسئله را در زمان بهینه‌ی $O(E_{VG} + n \log n)$ و فضای بهینه‌ی $O(n)$ حل می‌کند. در اینجا E_{VG} تعداد یال‌های گراف قابلیت دید است. این مقدار در بدترین حالت $O(n^2)$ است ولی در بسیاری از حالات عملی، از این مقدار کمتر است. پس از ساخت گراف قابلیت دید، با استفاده از الگوریتم دایکسترا می‌توان کوتاه‌ترین مسیر بین مبدأ و مقصد را در زمان $O(E_{VG} + n \log n)$ به دست آورد.

در اینجا لازم است به الگوریتم ارائه شده در [۵۸] اشاره کنیم. این الگوریتم نیز بر گراف قابلیت دید متکی است اما وابستگی زمان اجرای آن به n خطی و به h درجه‌ی دو است. پیچیدگی زمانی این الگوریتم $O(n + h^2 \log n)$ و فضای لازم برای اجرای آن $O(n)$ است.

۲.۳.۲ روش دایکسترای پیوسته

شیوه‌ی دیگر حل این مسئله با استفاده از الگوی دایکسترای پیوسته است. در این روش، مستقیماً نقشه‌ی کوتاه‌ترین مسیر ساخته می‌شود. در الگوی دایکسترای پیوسته اثر انتشار یک جبهه‌ی موج 2° از نقطه‌ی مبدأ s شبیه‌سازی می‌شود. جبهه‌ی موج در فاصله‌ی δ از s شامل تمام نقاطی از P است که در فاصله‌ی ژئودزیک δ از s قرار دارند. این جبهه‌ی موج از قطعه خم‌هایی تشکیل شده است که موجک 2^1 نامیده می‌شوند. موجک‌ها کمان‌هایی از دایره‌هایی به مرکز رئوسی هستند که تا به حال ملاقات شده‌اند.

در رخدادهای مشخصی ساختار جبهه‌ی موج تغییرات مهمی می‌کند. این رخدادها را می‌توان در قالب چهار دسته‌ی زیر بیان کرد:

(۱) یک موجک ناپدید می‌شود (توسط موجک‌های مجاور از بین می‌رود).

(۲) یک موجک با یک رأس یک مانع برخورد می‌کند.

(۳) یک موجک با یک موجک دیگر برخورد می‌کند.

(۴) یک موجک در یک نقطه‌ی داخلی از یک یال یک مانع، با آن یال برخورد می‌کند.

نکته‌ی قابل توجه در مورد این الگو این است که هر موجک متناظر با یک سلول از $SPM(s)$ به ریشه‌ی رأسی است که مرکز موجک است. به این ترتیب نقاط تقاطع موجک‌ها یال‌های این نقشه را جاروب می‌کنند.

از این که اندازه‌ی نقشه‌ی کوتاه‌ترین مسیر $SPM(s)$ اندازه‌ی خطی دارد به سادگی می‌توان نتیجه گرفت که تعداد رخدادهای $O(n)$ است. چالش اصلی در طراحی الگوریتم‌های مبتنی بر این الگو به دست آوردن چارچوب‌های مناسبی است که بتوان توسط آنان به طرز کارایی فهمید که چه

wavefront^{۲۰}

wavelet^{۲۱}

رخدادهایی قرار است اتفاق بیفتند و پردازش هر رخداد نیز با هزینه‌ای کم انجام شود.

یک رهیافت که در [۵۹] و [۶۰] استفاده شده است، به این شرح است که اجازه دهیم یک بخش از جبهه‌ی موج بدون در نظر گرفتن رخداد برخورد دو موجک از روی بخش دیگر رد شود و هنگامی که یک موجک با یک رأس که از پیش برچسب گذاری شده برخورد کند آن را بگیریم. به این ترتیب نیازی به در نظر گرفتن رخدادهای برخورد موجک‌ها با یکدیگر نیست و به این ترتیب هزینه‌ی نگهداری داده‌ساختارهای لازم برای کشف چنین رخدادهایی از الگوریتم حذف می‌شود. پیچیدگی زمانی این الگوریتم $O(n^{\frac{7}{3}+\epsilon})$ است.

رهیافت دیگری که در [۶۱] و [۶۲] استفاده شده است، مسئله را به این ترتیب ساده می‌کند که دامنه‌ی P را توسط یک نظریف منطبق^{۲۲} به ناحیه‌هایی تقسیم می‌کند که اجازه می‌دهند یک جبهه‌ی موج تقریبی ناحیه به ناحیه در نظریف منتشر شود. خاصیت کلیدی این نظریف این است که به ازای هر یال (به طول L) از نظریف، تنها تعداد ثابتی ناحیه وجود دارند که فاصله‌ی آنها از یال یادشده L است (اندازه‌ی این ناحیه‌ها نیز ثابت است). زمان اجرای این الگوریتم $O(n \log n)$ و فضای حافظه‌ی لازم برای آن نیز $O(n \log n)$ است.

اگرچه الگوریتم ذکر شده هنگامی که تعداد موانع زیاد باشد ($h = \Omega(n)$) در بدترین حالت بهینه است، اما از دو نظر بهینه نیست: یکی از نظر فضای مصرفی و دیگری از نظر پیچیدگی زمانی به عنوان تابعی از n و h . یکی از مسائل باز در این حیطه که بسیار مورد توجه است به دست آوردن یک الگوریتم بهینه است که پیچیدگی زمانی آن منطبق با کران پایین $\Omega(n + h \log h)$ باشد و تنها از $O(n)$ حافظه استفاده کند. در حال حاضر تنها الگوریتم شناخته شده که زمان اجرای آن برحسب n خطی است، برحسب h درجه‌ی دو است [۵۸].

^{۲۲}conforming subdivision

۴.۲ کوتاه‌ترین مسیرها روی یک رویه‌ی چندوجهی‌گونه

در حالت سه بعدی مسئله اگر توجه خود را به مسیرهایی که روی سطح یک چندوجهی داده شده قرار دارند محدود کنیم، مسئله از حالت NP-سخت خارج می‌شود و می‌توان برای آن راه‌حل چندجمله‌ای پیدا کرد. این تحدید از مسئله از جنبه‌ی کاربردی آن خیلی کم نمی‌کند. برای همین حالت از مسئله نیز کاربردهای زیادی می‌توان یافت که به همین دلیل توجه به این مسئله در حیطه‌ی کوتاه‌ترین مسیر زیاد است.

ابتدا شریر^{۲۳} و شور^{۲۴} در [۶۳] از خواص مربوط به بی‌تاشده‌ی مسیرهای بهینه روی رویه‌های چندوجهی استفاده کردند و یک الگوریتم از مرتبه‌ی زمانی $O(n^3 \log n)$ را برای رویه‌های محدب ارائه کردند. محاسبه‌ی کوتاه‌ترین مسیر روی چندوجهی‌های کلی (نه لزوماً محدب) در [۶۴] و [۶۵] و [۵۱] بررسی شده‌اند. در [۵۱] از روش دایکسترای پیوسته برای حل مسئله استفاده می‌شود و این منجر به الگوریتمی از مرتبه‌ی زمانی $O(n^2 \log n)$ و حافظه‌ی $O(n^2)$ شده است. چن^{۲۵} و هان^{۲۶} در [۶۴] الگوریتمی از مرتبه‌ی زمانی $O(n^2)$ و حافظه‌ی $O(n)$ ارائه کرده‌اند. این الگوریتم ساختاری ساده‌تر دارد و از روش دیگری برای حل مسئله استفاده می‌کند که به [۶۳] شبیه‌تر است. در این الگوریتم طی یک مرحله‌ی پیش‌پردازش با زمان $O(n^2)$ ، زیرتقسیمی از سطح به دست می‌آید که کوتاه‌ترین مسیر از مبدأ به نقاط هر ناحیه از آن، دنباله‌یال‌های یکسانی را قطع می‌کنند. با توجه به لم ۱.۲ و با داشتن نظریف یادشده، توسط پرس‌وجوی جایابی نقطه می‌توان کوتاه‌ترین فاصله از مبدأ تا هر نقطه‌ی دلخواه را در زمان لگاریتمی بیابیم.

مسئله‌ی ارائه‌ی یک الگوریتم با زمان کمتر از درجه‌ی دو تا مدت زیادی یک مسئله‌ی باز

Sharir^{۲۳}

Schorr^{۲۴}

Chen^{۲۵}

Han^{۲۶}

بسیار مورد توجه بود تا این که کاپور^{۲۷} در [۶۵] الگوریتمی از مرتبه $O(n \log^2 n)$ ارائه کرد. مشکل اصلی در این راه این واقعیت است که حتی روی یک چندوجهی محدب ممکن است $\Omega(n^2)$ تقاطع بین یال‌های چندوجهی و کوتاه‌ترین مسیرها به رئوس وجود داشته باشد. بنابراین اگر الگوریتمی بخواهد این مسیرها را دنبال کند و تقاطع‌های یادشده را به عنوان رخداد در الگوریتم خود در نظر بگیرد، کران پایین $\Omega(n^2)$ را خواهد داشت.

الگوریتم کاپور از الگوی دایکسترای پیوسته پیروی می‌کند. برای مقابله با مشکل فوق دو نکته در این الگوریتم گنجانده شده است: اول این که برای کشف این که رخداد بعدی چیست، متناظر با هر موجک w ، مجموعه یال‌هایی مرزی که نزدیک‌ترین موجک به آنها w است نگهداری می‌شوند. همین طور متناظر با هر یال مرزی e ، مجموعه موجک‌هایی که نزدیک‌ترین یال مرزی به آنها w است نگهداری می‌شوند. این امر یافتن نزدیک‌ترین جفت موجک - یال را در زمان لگاریتمی ممکن می‌کند. دومین نکته‌ی قابل توجه در این الگوریتم استفاده از نمایش محدب برای نگهداری بخش موج‌ها (مجموعه‌ای از چند موجک مجاور) و بخش یال‌ها (مجموعه‌ای از چند یال مجاور) است. با استفاده از این نمایش، محاسبه‌ی فاصله‌ی یک موجک تا یک بخش یال و فاصله‌ی یک یال تا یک بخش موج نیز با هزینه‌ی سرشکن شده‌ی^{۲۸} لگاریتمی ممکن است.

روی این مسئله در حیطه‌ی الگوریتم‌های تقریبی نیز زیاد کار شده است. در [۳۸] الگوریتمی تقریبی ارائه شده است که مسیری که پیدا می‌کند، حداکثر $\epsilon(1 + \epsilon)$ برابر مسیر بهینه است. الگوریتم در زمان $O(n^{\frac{1}{\epsilon}} \log^{\frac{1}{\epsilon}} n)$ اجرا می‌شود. با افزایش ضریب ϵ در تقریب مسئله به ۱۵ می‌توان الگوریتمی با زمان اجرای $O(n^{\frac{1}{15}} \log^{\frac{1}{15}} n)$ به دست آورد که در مقاله به آن اشاره شده است. در همه‌ی روش‌های فوق، پس از اجرای الگوریتم، یک نظریف به دست می‌آید که توسط آن می‌توان به پرس‌وجوهای کوتاه‌ترین مسیر بین هر دو نقطه صرفاً با یک الگوریتم جایابی نقطه در زمان لگاریتمی به دست آورد.

Kapoor^{۲۷}
amortized^{۲۸}

۵.۲ قابلیت دید

۱.۵.۲ قابلیت دید در دو بعد

در اینجا به بررسی برخی تعاریف و مفاهیم پایه مرتبط با قابلیت دید در چندضلعی‌های ساده و دامنه‌های چندضلعی گونه می‌پردازیم. از آنجا که وابستگی الگوریتم‌های ما به ساختار الگوریتم‌های قابلیت دید زیاد نیست از بیان جزئیات آنها خودداری می‌کنیم و صرفاً به اشاره به بهترین الگوریتم‌های موجود برای هر یک از دامنه‌ها بسنده می‌کنیم.

فرض کنید P یک چندضلعی ساده یا یک دامنه‌ی چندضلعی گونه است. دو نقطه‌ی p و q از P قابل دید هستند اگر پاره‌خط pq کاملاً داخل P قرار بگیرد. چندضلعی قابلیت دید نقطه‌ی p ، V_p مجموعه نقاطی از P است که از p قابل دید هستند. یک چندضلعی قابلیت دید یک چندضلعی ستاره‌شکل است که دو نوع یال دارد: برخی از آنها قطعاتی از یال‌های P هستند و بقیه از امتداد پاره‌خط‌های اصل p به رئوس P به دست می‌آیند. به یال‌های نوع دوم پنجره^{۲۹} گفته می‌شود. هر پنجره w چندضلعی P را به دو قسمت تقسیم می‌کند. یک قسمت شامل V_p است و یک قسمت که نقاط آن از p قابل دید نیستند و به آن جیب^{۳۰} چندضلعی گفته می‌شود که توسط پنجره‌ی w ایجاد شده است.

نمایش یک چندضلعی قابلیت دید توسط یک لیست حلقوی از رأس‌های آن به همراه مختصات آنها صورت می‌گیرد. اگر P یک چندضلعی ساده باشد، می‌توان یک چندضلعی قابلیت دید در آن را به صورت ترکیبیاتی نمایش داد. این نمایش به شکل فهرستی حلقوی از رأس‌ها و یال‌های P است به ترتیبی که روی مرز V_p ظاهر می‌شوند. با داشتن این اطلاعات مختصات یک رأس از V_p را می‌توان در زمان $O(1)$ به دست آورد. به این لیست نمایش ترکیبیاتی V_p می‌گوییم. توجه داشته باشید که این نحوه‌ی نمایش برای حالتی که P یک دامنه‌ی چندضلعی گونه باشد به

^{۲۹} window

^{۳۰} pocket

درستی کار نمی‌کند.

چندضلعی قابلیت دید یک نقطه در یک چندضلعی ساده توسط الگوریتم لی^{۳۱} [۶۶، ۶۷] در زمان $O(n)$ قابل محاسبه است. برای یک دامنه‌ی چندضلعی گونه می‌توان از الگوریتم هفرنان^{۳۲} و میچل^{۳۳} [۶۸] برای محاسبه‌ی چندضلعی قابلیت دید استفاده کرد. این الگوریتم بهینه است و مسئله را در مرتبه‌ی زمانی $O(n + h \log h)$ حل می‌کند که h تعداد حفره‌های دامنه است.

۲.۵.۲ قابلیت دید در بیش از دو بعد

قابلیت دید در بیش از دو بعد با مسئله‌ی شناخته‌شده‌ی برداشتن رویه‌ی پنهان^{۳۴} رابطه‌ی نزدیکی دارد. از آنجا که ما مطالعه‌ی مسئله‌ی کوتاه‌ترین مسیر با قید قابلیت دید در بیش از دو بعد را به دامنه‌ی رویه‌های چندوجهی گونه محدود کرده‌ایم، در بیان مفاهیم مرتبط با قابلیت دید نیز به همین دامنه بسنده می‌کنیم.

در یک رویه‌ی چندوجهی گونه P ، بخشی از دامنه که یک نقطه‌ی دید می‌بیند، لزوماً به شکل یک چندضلعی نیست. به همین دلیل V_p را ناحیه‌ی قابلیت دید^{۳۵} نقطه‌ی p می‌نامیم که مجموعه‌ی نقاطی از دامنه است که از p قابل دید هستند. موقعیت p نسبت به P و محدب بودن یا نبودن P ، روی ساختار V_p تأثیر دارد:

- اگر P رویه‌ی یک چندوجهی محدب باشد و p داخل یکی از وجه‌های P مثل f واقع شده باشد، V_p شامل f و تمام وجه‌های هم‌صفحه با f است. چون P محدب است، این مجموعه، ناحیه‌ی همبندی از P را تشکیل می‌دهند.

^{۳۱} Lee

^{۳۲} Heffernan

^{۳۳} Mitchell

^{۳۴} hidden surface removal

^{۳۵} visibility region

- اگر P رویه‌ی یک چندوجهی محدب باشد و p یکی از رأس‌های چندوجهی باشد، تمام وجه‌های مجاور p و تمام وجه‌های هم‌صفحه با آنها V_p را تشکیل می‌دهند. در این حالت نیز V_p همبند است.

- اگر P رویه‌ی یک چندوجهی محدب باشد و p خارج از سطح P باشد، به سادگی می‌توان دریافت که اگر یک نقطه از یک وجه از p قابل دید باشد، آنگاه تمام نقاط وجه از p قابل دید هستند. به همین دلیل در این حالت V_p مجدداً از تعدادی همبند از وجه‌های P تشکیل شده است.

- اگر P یک چندوجهی مقعر باشد، مستقل از موقعیت p نسبت به P ، V_p از تعدادی ناحیه‌ی همبند تشکیل می‌شود. هر کدام از ناحیه‌ها شامل بخش‌هایی از وجه‌های P هستند. این به خاطر سایه‌هایی است که برخی از بخش‌های P روی وجه‌های دیگر ایجاد می‌کنند. پیچیدگی هر یک از نواحی همبند $O(n)$ و تعداد نواحی نیز $O(n)$ است.

الگوریتم‌های زیادی برای حل مسئله‌ی حذف سطوح پنهان داده شده است [۶۹، ۷۰، ۷۱].

این الگوریتم‌ها همگی در چارچوب حدود پایین زمان و حافظه‌ی مرتبه‌ی n^2 [۷۲] عمل می‌کنند.

کوتاه‌ترین مسیرها با قابلیت دید یک نقطه

در این فصل مسئله یافتن کوتاه‌ترین مسیر بین دو نقطه را با قید قابلیت دید یک هدف دید نقطه‌ای بررسی می‌کنیم. این مسئله در دامنه‌های هندسی متفاوتی قابل بررسی است. در بخش‌های آینده، مسئله برای دامنه‌های چندضلعی ساده، چندضلعی با مانع و رویه چندوجهی را در حالات محدب یا غیرمحدب بررسی خواهیم کرد. از آنجا که روش حل این مسائل تا حد زیادی به هم شبیه است، ابتدا این روش را مستقل از دامنه بیان می‌کنیم و در بخش‌های مربوط جزئیات مربوط به دامنه‌ها را به طور جداگانه مطالعه می‌کنیم.

۱.۳ الگوی کلی

فرض کنید P دامنه هندسی مسئله مورد مطالعه باشد. هدف مسئله یافتن یک مسیر با طول کمینه بین دو نقطه داده شده s و t در P است طوری که حداقل یک نقطه از این مسیر از نقطه‌ی داده شده p قابل دید باشد. این شرط معادل این است که اشتراک این مسیر با ناحیه‌ی قابل دید p که آن را V_p می‌نامیم ناتهی باشد. این اشتراک ممکن است در یک یا بیش از یک نقطه باشد. در حالت اول، مسیر در نقطه‌ی اشتراک ناحیه‌ی قابل دید را لمس می‌کند و برمی‌گردد و در حالت دوم، بخشی از مسیر

درون ناحیه‌ی قابل دید قرار می‌گیرد.

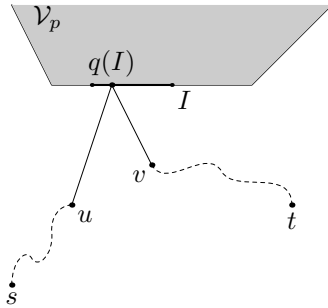
به مسیری که حداقل یک نقطه از آن از p قابل دید باشد یک مسیر قابل دید از p می‌گوییم. توجه داشته باشید که کوتاه‌ترین مسیرها در دامنه‌های چندوجهی گونه و رویه‌های چندوجهی گونه یکتا نیستند. بنابراین در این حالات ممکن است چندین کوتاه‌ترین مسیر قابل دید از p بین دو نقطه وجود داشته باشد. ما در اینجا به دنبال یافتن فقط یکی از آنها هستیم.

فرض کنید q نخستین نقطه قابل دید در حرکت از s به t روی یک کوتاه‌ترین مسیر قابل دید از p باشد. واضح است که q روی مرز V_p قرار دارد و زیرمسیرهای از s به q و از q به t بهینه هستند. بنابراین مسئله ما به یافتن نقطه‌ای مانند q تبدیل می‌شود. با پیدا کردن چنین نقطه‌ای می‌توان کوتاه‌ترین مسیر از s و t به آن را پیدا کرد و از اتصال آن پاسخ مسئله را یافت.

ایده اصلی الگوریتم، افراز مرز V_p به بازه‌هایی است که به ازای هر بازه مانند I ، بتوان به سرعت نقطه‌ای مانند $q(I)$ را پیدا کرد که مجموع فاصله‌های ژئودزیک آن تا s و t کمینه باشد. به این نقطه، نقطه‌ی بهینه‌ی محلی نیز می‌گوییم. پس از محاسبه‌ی افراز، می‌توان به ازای تمام بازه‌های موجود در افراز، نقطه‌ی بهینه‌ی متناظر را محاسبه کرد. بین این نقاط نقطه‌ای که کمترین مجموع فاصله‌ها را دارد q است. برای خلاصه‌سازی، از نماد B_p به عنوان مجموعه‌ی نقاط روی مرز V_p استفاده می‌کنیم. در تمام دامنه‌هایی که بررسی خواهد شد، B_p متشکل از تعدادی پاره‌خط است.

برای محاسبه‌ی افراز مورد نظر از این خاصیت استفاده می‌کنیم که در دامنه‌های مورد نظر، تظریف‌هایی قابل تعریف هستند که تمام کوتاه‌ترین مسیرها به نقاط یک تظریف دارای ساختار ترکیبیاتی ثابتی هستند. این تظریف در چندضلعی‌های ساده و دامنه‌های چندضلعی‌گونه همان نقشه‌ی کوتاه‌ترین مسیر است و همان‌گونه که خواهیم دید، برای رویه‌های چندوجهی‌گونه نیز چنین تظریفی قابل تعریف است.

به این ترتیب به ازای هر نقطه از P مانند x ، می‌توان B_p را توسط تظریف متناظر با x به بازه‌هایی تقسیم کرد که ساختار ترکیبیاتی کوتاه‌ترین مسیرها از x به نقاط روی هر بازه یکسان باشد.



شکل ۱: در یک دامنه‌ی دوبعدی، بازه‌ی بهینگی $I \in L_{s,t}$ با پاره‌خط پرننگ نشان داده شده است. u آخرین رأس روی کوتاه‌ترین مسیر از s به نقاط I و v آخرین رأس روی کوتاه‌ترین مسیر از s به این نقاط است. مجموع فواصل ژئودزیک نقطه‌ی $q(I)$ از s و t بین تمام نقاط I کمینه است.

با توجه به ساختار خاص کوتاه‌ترین مسیرها در معیار اندازه‌گیری اقلیدسی، هر یک از این بازه‌ها با یک نقطه در P متناظرند که آخرین نقطه‌ی شکست کوتاه‌ترین مسیرها از x به نقاط روی بازه است. به چنین بازه‌ای بازه‌ی بهینگی^۱ متناظر با x می‌گوییم و مجموعه بازه‌های بهینگی متناظر با x روی B_p را با L_x نشان می‌دهیم. به این ترتیب با محاسبه‌ی L_s و L_t و ذخیره‌ی آخرین نقاط شکست‌ها می‌توان به ازای هر نقطه‌ی دلخواه مانند a از یک بازه، کوتاه‌ترین فاصله از s به a و کوتاه‌ترین فاصله از t به a را در زمان ثابت به دست آورد.

حال اگر نقاط انتهایی بازه‌های درون L_s و L_t را با هم ادغام کنیم و توسط آنها B_p را به مجموعه جدیدی از بازه‌ها مانند $L_{s,t}$ تقسیم کنیم، به ازای هر نقطه روی یک بازه، مجموع کوتاه‌ترین فاصله‌های آن نقطه از s و t در زمان ثابت قابل محاسبه خواهد بود. به این بازه‌ها، بازه‌های بهینگی متناظر با جفت (s, t) می‌گوییم (شکل ۱). در جایی که ابهامی ایجاد نکند، از عبارت «بازه‌ی بهینگی» به جای «بازه‌ی بهینگی متناظر با جفت (s, t) » استفاده خواهیم کرد.

طبق لم زیر، پیدا کردن نقطه‌ی بهینه‌ی محلی در حیطه‌ی یک بازه‌ی بهینگی به سادگی قابل انجام است.

interval of optimality^۱

لم ۱.۳ اگر I یک بازه در $L_{s,t}$ باشد، می توان نقطه‌ی $q(I)$ از I را که کمترین مجموع فاصله‌ها از s و t دارد در زمان ثابت پیدا کرد.

اثبات. I دارای این خاصیت است که آخرین نقطه‌ی شکست کوتاه‌ترین مسیرها از s و t به نقاط روی آن ثابت است. فرض کنیم u آخرین نقطه‌ی شکست روی کوتاه‌ترین مسیرها از s و v آخرین نقطه‌ی شکست روی کوتاه‌ترین مسیرها از t به نقاط I باشد. برای پیدا کردن $q(I)$ می توانیم قرینه‌ی v حول خطی که I را دربر می‌گیرد به u متصل کنیم. اگر پاره‌خط به دست آمده I را قطع کند، نقطه‌ی تقاطع $q(I)$ است و گرنه $q(I)$ یکی از دو نقطه‌ی انتهای I است (بسته به این که پاره‌خط مذکور، خط دربرگیرنده‌ی I را در کدام طرف I قطع می‌کند). \square

بر مبنای توضیحات بالا، طرح یک الگوریتم کلی محاسبه‌ی یک کوتاه‌ترین مسیر قابل دید از p بین s و t به صورت زیر خواهد بود:

(۱) B_p را محاسبه کن.

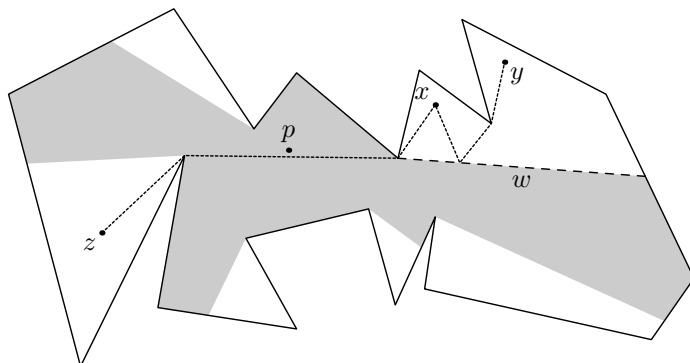
(۲) مجموعه $L_{s,t}$ را روی B_p محاسبه کن.

(۳) برای هر بازه‌ی $I \in L_{s,t}$ نقطه‌ی $q(I)$ را که کمترین مجموع فاصله‌ها را از s و t دارد پیدا کن.

(۴) فرض کنیم q نقطه‌ای از $\{q(I) \mid I \in L_{s,t}\}$ باشد که کمترین مجموع فاصله‌ها را دارد.

(۵) یک کوتاه‌ترین مسیر قابل دید از p بین s و t از به دنبال هم گذاشتن یک کوتاه‌ترین مسیر از s به q و یک کوتاه‌ترین مسیر از q به t به دست می‌آید.

نکته‌ی قابل توجه در این الگوریتم این است که مرتبه‌ی زمان اجرای الگوریتم بستگی به تعداد بازه‌های $L_{s,t}$ دارد. اگر فرض کنیم هر یک از مجموعه‌های L_s و L_t در بدترین حالت k عضو داشته باشند، اندازه‌ی $L_{s,t}$ نیز $O(k)$ خواهد بود. قدم‌های سوم و چهارم الگوریتم نیازمند $O(k)$ زمان هستند. این نکته با توجه به این موضوع اهمیت پیدا می‌کند که در دامنه‌های بررسی شده، k از



شکل ۲: یک چندضلعی ساده با نقطه‌ی دید p داخل آن - ناحیه‌ی خاکستری V_p است که شش جیب در چندضلعی ایجاد می‌کند (ناحیه‌های سفید). کوتاه‌ترین مسیر بین x و z قابل دید از p هم هست ولی کوتاه‌ترین مسیر قابل دید از p بین x و y ، پنجره‌ی w را لمس می‌کند و برمی‌گردد.

مرتب‌بندی $O(n^2)$ است. به همین دلیل در مورد هر یک از دامنه‌ها تلاش ما بر محدود کردن بازه‌هایی خواهد بود که باید مورد بررسی قرار بگیرند.

نکته‌ی قابل توجه دیگر زمان لازم برای محاسبه‌ی $L_{s,t}$ از روی L_s و L_t است. اگر نقاط انتهایی این دو مجموعه را به طور نامرتب داشته باشیم، ناچار به صرف زمان $O(k \log k)$ برای مرتب کردن اعضای دو مجموعه و ساخت $L_{s,t}$ هستیم. در بخش‌های مربوط نشان خواهیم داد که چگونه مجموعه‌های L_s و L_t را محاسبه می‌کنیم که بازه‌های آنها به طور مرتب روی B_p یا بخش مرتبط با آن به دست آیند.

۲.۳ چندضلعی‌های ساده

برای حالت چندضلعی‌های ساده، بسیاری از الگوریتم‌های مرتبط با کوتاه‌ترین مسیرها الگوریتم‌های خطی دارند. به همین علت، ما نیز تمایل داریم مسئله‌ی کوتاه‌ترین مسیر با قید قابلیت دید را نیز در زمان خطی حل کنیم.

فرض کنید P یک چندضلعی ساده با n یال باشد. s و t نقاط مبدأ و مقصد داده شده هستند و V_p چندضلعی قابلیت دید نقطه‌ی داده شده‌ی p است. اگر حداقل یکی از s و t در V_p باشند،

کوتاه‌ترین مسیر بین s و t از p قابل دید هست. همچنین اگر s و t در دو جیب متفاوت باشند باز هم این خاصیت برقرار است. دلیل این امر این است که تنها راه خارج شدن مسیر از یک جیب این است که یکی از یال‌های V_p قطع شود (شکل ۲). بنابراین می‌توانیم فرض کنیم s و t هر دو در یک جیب قرار دارند. فرض کنید نام این جیب P' باشد که توسط پنجره‌ی w ایجاد شده است. لم زیر به ما کمک می‌کند که محدوده‌ی جستجو به دنبال نقطه‌ی تماس q را کوچک کنیم.

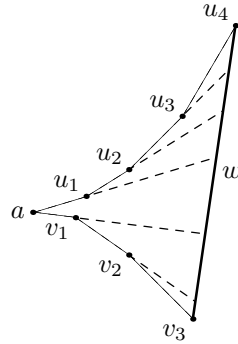
لم ۲.۳ اگر s و t هر دو در یک جیب P' باشند، کوتاه‌ترین مسیر قابل دید از p به طور کامل در P' یا روی مرز آن قرار دارد.

اثبات. فرض کنیم π یک مسیر قابل دید از p باشد که در بعضی قسمت‌ها از P' خارج شده است. اگر در جریان پیمایش π از s به t اولین نقطه‌ای باشد که π از P' خارج می‌شود و v آخرین نقطه‌ی ورود به P' باشد، u و v هر دو روی w قرار دارند. چون w پاره‌خطی است که به طور کامل در P قرار دارد، می‌توان بخشی از π را که بین u و v قرار دارد با پاره‌خط مستقیم واصل دو نقطه میان‌بر زد و این مسیر را با حفظ قید قابلیت دید کوتاه‌تر کرد. \square

بر این اساس می‌توانیم محاسبه‌ی بازه‌های بهینگی را به یال w محدود کنیم.

لم ۳.۳ مجموعه‌ی $L_{s,t}$ روی یال جداکننده‌ی جیب شامل s و t (P') در زمان $O(n)$ قابل محاسبه است.

اثبات. فرض می‌کنیم P' به عنوان ورودی به الگوریتم کوتاه‌ترین مسیر گویاس و دیگران [۵۴] داده شده است (با در نظر گرفتن نقطه‌ی s به عنوان مبدأ). در جریان پیمایش عمق‌اول، بالاخره به یال w می‌رسیم (چون این یال جزئی از مثلث‌بندی خواهد بود). اکنون فرض کنید ساختار قیف‌مانندی که در جریان پیمایش نگهداری شده است، به صورت $[u_l, u_{l-1}, \dots, a, v_1, \dots, v_k]$ است طوری که



شکل ۳: بازه‌های بهینگی تشکیل شده روی پنجره‌ی w که از اجرای الگوریتم کوتاه‌ترین مسیر در چندضلعی P' به دست می‌آیند.

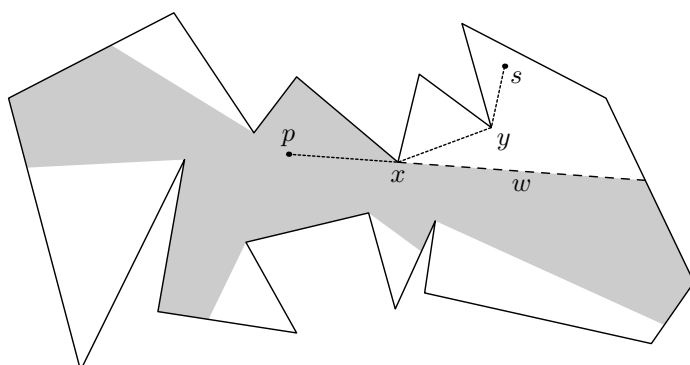
$a = u_0 = v_0$ نوک قیف است و $w = u_l v_k$ (شکل ۳). به این ترتیب امتداد پاره‌خط‌های $u_i u_{i+1}$ و $v_j v_{j+1}$ را به $k+l-1$ بازه تقسیم می‌کند که هر آخرین رأس روی کوتاه‌ترین مسیر از s به نقاط هر بازه یکی از رأس‌های u_i یا v_j (یا a) است. این بازه‌ها مجموعه‌ی L_s را تشکیل می‌دهند. به دست آوردن مجموعه‌ی L_t نیز به همین صورت خواهد بود. برای محاسبه‌ی $L_{s,t}$ کافی است دنباله نقاط انتهایی بازه‌های L_s و L_t را با هم ادغام کنیم تا بازه‌های $L_{s,t}$ به دست آیند.

به دست آوردن L_s و L_t به صورتی که گفته شد از دو بار اجرای الگوریتم کوتاه‌ترین مسیر در P' به دست می‌آید که به زمان $O(n)$ نیاز دارد. در الگوریتم یادشده، قیف در داده‌ساختار درخت جستجوی انگشتی نگهداری می‌شود که این داده‌ساختار مبتنی بر درخت بی^۲ است. به همین دلیل، بازیابی رأس‌های دیواره‌ی قیف به طور مرتب در زمان خطی قابل انجام است و این موجب می‌شود بازه‌های L_s و L_t بدون هزینه‌ی لازم برای مرتب‌سازی در زمان $O(n)$ قابل ادغام باشند. □

برای پیدا کردن پنجره‌ی w ، از نتیجه‌ی لم زیر استفاده می‌کنیم.

لم ۴.۳ با داشتن نقشه‌ی کوتاه‌ترین مسیر $SPM(s)$ ، پنجره‌ی w که s را از V_p جدا می‌کند، در زمان $O(\log n)$ قابل محاسبه است.

^۲B-tree



شکل ۴: x آخرین رأس چندضلعی روی کوتاه‌ترین مسیر از s به p است که پنجره‌ی جدا کننده‌ی V_p از P' را تعیین می‌کند.

اثبات. فرض بر این است که p و s قابل دید نیستند. در این صورت کوتاه‌ترین مسیر از s به p را در نظر بگیرید و آن را π بنامید. این مسیر دقیقاً یک بار به V_p وارد می‌شود. فرض کنید x آخرین رأس در π پیمایش از s به p باشد. واضح است که این نقطه از p قابل دید است. اگر رأس قبل از x در این مسیر باشد، می‌توان مشاهده کرد که y از p قابل دید نخواهد بود (در غیر این صورت طبق نامساوی مثلثی می‌توان مسیر y به x به p را با پاره‌خط yp جایگزین کرد و π را کوتاه‌تر کرد). بنابراین x نقطه‌ی ورود π به V_p است. در این حالت w از امتداد پاره‌خط px به دست می‌آید (شکل ۴).

به این ترتیب در حالتی که $SPM(s)$ محاسبه شده باشد با جایابی نقطه‌ی p در این نقشه، نقطه‌ی x به عنوان ریشه‌ی سلول شامل p در زمان $O(\log n)$ به دست می‌آید. با به دست آمدن این نقطه، کفایت محل تقاطع امتداد px و مرز P را محاسبه کنیم. این کار یا با الگوریتم‌های شلیک پرتو یا با یک جستجوی دودویی روی مرز P در زمان $O(\log n)$ قابل انجام است. \square

طبق نکات یادشده، مراحل انجام شده برای حل مسئله به این ترتیب خواهد بود:

(۱) اگر p از s یا t قابل دید باشد، آنگاه کوتاه‌ترین مسیر بین s و t جواب مسئله است و توقف کن.

۲) نقشه‌های $SPM(s)$ و $SPM(t)$ را محاسبه کن.

۳) طبق لم ۴.۳ پنجره‌هایی را که s و t را از p جدا می‌کنند محاسبه کن.

۴) اگر دو پنجره متمایز باشند، آنگاه کوتاه‌ترین مسیر بین s و t جواب مسئله است و توقف کن.

۵) طبق لم ۳.۳ مجموعه بازه‌های $L_{s,t}$ را محاسبه کن.

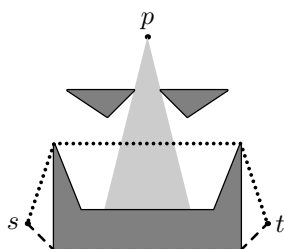
۶) طبق سه مرحله‌ی آخر الگوی کلی ارائه شده در بخش ۱.۳ کوتاه‌ترین مسیر قابل دید از p را بین s و t محاسبه می‌کنیم.

با توجه به این که تمام قدم‌های الگوریتم در زمان خطی قابل انجامند، می‌توان نتیجه‌ی نهایی درباره‌ی این مسئله را به صورت قضیه‌ی زیر بیان کرد.

قضیه ۱.۳ با داشتن یک چندضلعی ساده P با n یال و نقاط s, t و p داخل چندضلعی، می‌توان کوتاه‌ترین مسیر بین s و t را که از p قابل دید است در زمان $O(n)$ و حافظه‌ی $O(n)$ محاسبه کرد.

۳.۳ دامنه‌های چندضلعی گونه

در یک دامنه‌ی چندضلعی گونه‌ی P با n یال، حد پایین مسئله‌ی کوتاه‌ترین مسیر برحسب n ، $\Omega(n \log n)$ است. به همین دلیل، هدف ما یافتن الگوریتمی با همین مرتبه‌ی زمانی است طوری که قید قابلیت دید را هم شامل باشد. می‌توان نشان داد دامنه‌های چندضلعی گونه‌ای وجود دارند که در آنها اندازه‌ی مجموعه‌های L_s و L_t و در نتیجه اندازه‌ی $L_{s,t}$ از مرتبه‌ی $O(n^2)$ است. به این ترتیب به‌کارگیری سراسر الگوی کلی مطرح شده در بخش ۱.۳ به زمان اجرای $O(n^2)$ می‌انجامد. از این رو در الگوریتم ارائه شده از خواصی استفاده می‌کنیم که نشان می‌دهد بررسی تمام این بازه‌ها ضروری نیست.



شکل ۵: بخشی از یک دامنه‌ی چندضلعی گونه - چندضلعی‌های تیره‌رنگ موانع هستند و بخش خاکستری رنگ، قسمتی از V_p است. کوتاه‌ترین مسیر قابل دید از p بین s و t (مسیر نقطه‌چین)، V_p را قطع می‌کند در حالی که s و t هر دو در یک ناحیه‌ی غیرقابل دید هستند و کوتاه‌ترین مسیر بین آنها (مسیر خط‌چین) V_p را قطع نمی‌کند.

همانند مفهوم جیب در چندضلعی ساده، مجموعه‌های همبندی از نقاط را که همگی از p غیرقابل دید هستند ناحیه‌های غیرقابل دید می‌نامیم. در اینجا نیز به سادگی می‌توان مشاهده کرد که در حالتی که s و t در دو ناحیه‌ی غیرقابل دید باشند، کوتاه‌ترین مسیر بین آنها از p قابل دید است. این مسئله ما را به این سمت راهنمایی می‌کند که همانند حالت چندضلعی ساده، بازه‌های بهیجی را روی مرز V_p با ناحیه‌ی غیرقابل دیدی که s و t را شامل می‌شود بسازیم.

در حالت چندضلعی ساده می‌توان نشان داد مسیر بهینه یا وارد V_p نمی‌شود یا همان کوتاه‌ترین مسیر بین s و t است (لم ۲.۳). این خصوصیت کمک می‌کند که با اجرای الگوریتم ساخت نقشه‌ی کوتاه‌ترین مسیر، مجموعه‌های L_s و L_t را محاسبه کنیم. با توجه به این که اندازه‌ی این نقشه‌ها در دامنه‌های چندضلعی گونه $O(n)$ است، می‌توان نتیجه گرفت که تعداد بازه‌های تشکیل شده روی مرز اشاره شده نیز $O(n)$ باشد. مشکل در اینجا است که به خاطر وجود موانع و در نتیجه یکتا نبودن مسیرهای بهینه‌ی محلی در دامنه‌های چندضلعی گونه، می‌توان حالاتی را ایجاد کرد که s و t هر دو در یک ناحیه‌ی غیرقابل دید واقع باشند و کوتاه‌ترین مسیر بین آنها نیز به طور کامل در همان ناحیه باشد اما کوتاه‌ترین مسیر قابل دید از p وارد V_p و دوباره از آن خارج شود. در بعضی از حالات، این ورود و خروج از یال‌های متفاوتی از V_p صورت می‌گیرد که امکان میان‌بر زدن را نیز از بین می‌برد (شکل ۵).

۱.۳.۳ دروازه‌ی قابلیت دید

تعریف مفهومی به نام دروازه به ما کمک می‌کند که نیازی به تولید و بررسی تمام $O(n^2)$ بازه‌ی L_s و L_t نداشته باشیم. مجموعه‌ی Π از تمام کوتاه‌ترین مسیرهای قابل دید از p از مبدأ s به مقصد تمام نقاط فضای آزاد را در نظر بگیرید. به ازای هر مسیر $\pi \in \Pi$ ، فرض کنید q_π اولین نقطه‌ی قابل دید از p هنگام پیمایش مسیر از s به مقصد باشد (به عبارت دیگر، q_π نقطه‌ای با کمترین فاصله‌ی ژئودزیک از s از بین تمام نقاط قابل دید از p روی π است). مجموعه‌ی $\{q_\pi \mid \pi \in \Pi\}$ را دروازه‌ی s به V_p (یا به طور خلاصه، دروازه‌ی s) می‌نامیم. توجه داشته باشید که برای یک کوتاه‌ترین مسیر قابل دید از p مثل π ، زیرمسیر از s به q_π یک مسیر بهینه بین این دو نقطه است. همین خاصیت در مورد زیرمسیر از q_π به نقطه‌ی مقصد برقرار است.

مجموعه‌ی نقاط دروازه‌ی s روی B_p تشکیل تعدادی پاره‌خط می‌دهد. این پاره‌خطها بازه‌های بهینگی s هستند که به دروازه‌ی s محدود شده‌اند. تعداد این پاره‌خطها را اندازه‌ی دروازه‌ی s می‌نامیم. این تعداد نمایانگر پیچیدگی این مجموعه است. لم زیر نشان می‌دهد اندازه‌ی دروازه خطی است.

لم ۵.۳ اندازه‌ی دروازه‌ی s به V_p از مرتبه‌ی $O(n)$ است.

اثبات. فرض کنید f یک سلول از $SPM(s)$ به ریشه‌ی r باشد. با اضافه کردن دو دسته پاره خط به شرح زیر می‌توان f را به تعدادی ناحیه‌ی مثلث‌مانند تجزیه کرد.

(۱) r را به نقاط تقاطع B_p و مرز f وصل می‌کنیم.

(۲) r را به رأس‌های دیگر f وصل می‌کنیم.

از آنجا که f ستاره‌شکل با هسته‌ی r است، تمام این پاره‌خطها درون f قرار می‌گیرند و r را به یک نقطه از مرز f وصل می‌کنند. نواحی ایجاد شده یا مثلث هستند، یا شکلی که از دو پاره‌خط

حادث به r و یک خم هذلولوی تشکیل شده‌اند (به سادگی می‌توان مشاهده کرد که دو بخش مرز f که حادث به r هستند پاره‌خط‌های مستقیم هستند). از آنجا که V_p یک چندضلعی قابلیت دید است، رأس‌های آن همگی روی مرز P قرار دارند. هر یک از این ناحیه‌ها با تعدادی از یال‌های B_p تقاطع دارند (ممکن است این تعداد صفر باشد)، ولی هیچ رأسی از B_p یا f در داخل آنها نیست. به این ترتیب، تقاطع یک ناحیه با یال‌های B_p تعدادی پاره‌خط به وجود می‌آورد که نقاط انتهایی آنها روی دو ضلع مجاور به r در آن ناحیه قرار می‌گیرند. چون هیچ رأسی از B_p داخل ناحیه نیست، پاره‌خط‌ها را می‌توان به ترتیب فاصله از r مرتب کرد. برای یک ناحیه‌ی دلخواه فرض کنیم این پاره‌خط‌ها به ترتیب $\sigma_0, \sigma_1, \dots, \sigma_k$ باشند. ما ادعا می‌کنیم که فقط یکی از این پاره‌خط‌ها جزء دروازه‌ی s محسوب می‌شود که نزدیک‌ترین آنها به r یعنی σ_0 است. به این منظور، نقطه‌ی $x \in f \cap V_p$ را در نظر بگیرید و فرض کنید که متعلق به یک دروازه باشد. بنابراین باید کوتاه‌ترین مسیر قابل دیدی از p مانند π وجود داشته باشد که x اولین نقطه‌ی قابل دید از p روی این مسیر است. چون π یک مسیر بهینه است، زیرمسیر s به x نیز بهینه خواهد بود. پس r آخرین رأس این زیرمسیر خواهد بود و آخرین یال آن نیز پاره‌خط مستقیم rx است. rx باید با σ_0 تقاطع داشته باشد. پس x روی σ_0 قرار دارد. بنابراین تنها نقاط روی σ_0 می‌توانند جزء دروازه‌ی s باشند. تعداد پاره‌خط‌هایی که به این شکل می‌توانند جزء دروازه‌ی s باشند حداکثر برابر تعداد ناحیه‌ها است. چون تعداد ناحیه‌ها $O(n)$ است، پس اندازه‌ی دروازه‌ی s نیز $O(n)$ خواهد بود. \square

به این ترتیب، تعریف دروازه به ما کمک می‌کند تا به جای بررسی $O(n^2)$ بازه‌ی موجود در L_s ، فقط با $O(n)$ تا از آنها سروکار داشته باشیم. اینها بازه‌هایی متناظر با نخستین برخورد کوتاه‌ترین مسیرها از s با V_p هستند. مشکلی که وجود دارد این است که اگر بخواهیم بازه‌های بهینگی متناظر با s و t را به دروازه‌های آنها محدود کنیم و بعد آنها را در قالب مجموعه‌ی $L_{s,t}$ ادغام کنیم ممکن است بعضی از حالات را بررسی نکنیم. به عبارت دیگر، لزومی ندارد که نقطه‌ی q که در چارچوب

کلی الگوریتم به دنبال آن می‌گردیم در پیمایش مسیر بهینه از هر دو طرف اولین نقطه‌ای باشد که از p قابل دید است. حتی ممکن است بازه‌های بهینگی متناظر با s و t اصلاً اشتراک نداشته باشند. لم زیر به حل این مشکل می‌پردازد.

لم ۶.۳ برای محاسبه‌ی یک کوتاه‌ترین مسیر قابل دید از p بین s و t کافی است بازه‌های بهینگی t را روی دروازه‌ی s به V_p بررسی کنیم.

اثبات. فرض کنید π یک کوتاه‌ترین مسیر قابل دید از p بین s و t باشد. این مسیر دقیقاً یک بار دروازه‌ی s را قطع می‌کند (در نقطه‌ای مانند x). به وضوح دو زیرمسیر از s به x و از x به t بهینه هستند. به این خاطر اگر بازه‌های بهینگی t را روی دروازه‌ی s محاسبه کنیم و با بازه‌های بهینگی s روی همین دروازه ادغام کنیم و مجموعه‌ی ادغام شده را به عنوان مجموعه‌ی $L_{s,t}$ در الگوی کلی بخش ۱.۳ به کار گیریم، قادر به یافتن یک کوتاه‌ترین مسیر قابل دید از p بین s و t هستیم. \square

۲.۳.۳ محاسبه‌ی بازه‌های بهینگی

با توجه به توضیحات یادشده، برای ساختن مجموعه‌ی $L_{s,t}$ باید قدم‌های زیر را برداریم:

- محاسبه‌ی دروازه‌ی s و تجزیه‌ی آن به مجموعه بازه‌های بهینگی (L_s)
- محاسبه‌ی بازه‌های بهینگی L_t روی دروازه‌ی s
- ادغام دو مجموعه بازه و به دست آوردن $L_{s,t}$

همان‌گونه که در پیش یاد شد، اگر s و t در دو ناحیه‌ی غیرقابل دید متفاوت باشند، هر کوتاه‌ترین مسیر بین آنها از p قابل دید است. بنابراین فرض می‌کنیم s و t هر دو در یک ناحیه‌ی غیرقابل دید مثل P' هستند. P' یک دامنه‌ی چندضلعی‌گونه است که برخی از یال‌های آن با V_p

مشترک است. مجموعه نقاط این یال‌ها را A می‌نامیم. به سادگی می‌توان دید که A شامل دروازه‌ی s است (به این خاطر که اولین نقاط قابل دید از p روی کوتاه‌ترین مسیرها از مبدأ s باید روی مرز ناحیه‌ی غیرقابل دید شامل s باشند).

محاسبه‌ی بازه‌های بهینگی s

اگر نقشه‌ی $SPM(P', s)$ را در نظر بگیریم، A توسط یال‌های این نقشه به $O(n)$ پاره‌خط تقسیم می‌شود. این پاره‌خط‌ها دو دسته‌اند. دسته‌ی اول پاره‌خط‌هایی هستند که نقاط آنها به طور کامل جزء دروازه‌ی s هستند و دسته‌ی دوم آنهایی هستند که هیچ‌یک از نقاط آنها در دروازه‌ی s قرار ندارند.

برای تفکیک این دو دسته پاره‌خط به این نکته توجه کنید که هر یک از این پاره‌خط‌ها بخشی از یک سلول نقشه‌ی M' هستند. یکی از این پاره‌خط‌ها را در نظر بگیرید و فرض کنید ریشه‌ی سلول متناظر آن r باشد (r حتماً یک رأس Q است). برای تشخیص این که پاره‌خط مورد نظر جزء دروازه هست یا نه باید ببینیم که آیا کوتاه‌ترین مسیرها از s به نقاط آن داخل Q همان کوتاه‌ترین مسیرها از s به نقاط آن داخل P هستند یا این که مانع فرض کردن V_p باعث تغییر این مسیرها شده است. به این منظور دو درخت کوتاه‌ترین مسیر $T = SPT(P, s)$ و $T' = SPT(P', s)$ را در نظر بگیرید. اگر مسیر از s به r در T' همان مسیر از s به r در T باشد، مانع فرض کردن V_p تأثیری در کوتاه‌ترین مسیرها نداشته است. در چنین حالتی پاره‌خط مورد نظر به طور کامل در دروازه‌ی s قرار می‌گیرد و در غیر این صورت به طور کامل خارج از دروازه است.

به این ترتیب برای محاسبه‌ی بازه‌های بهینگی L_s روی دروازه‌ی s به این ترتیب عمل می‌شود:

(۱) ناحیه‌ی غیرقابل دید P' شامل s را محاسبه کن.

(۲) درخت‌های کوتاه‌ترین مسیر T و T' از مبدأ s را به ترتیب در P و P' محاسبه کن.

(۳) نقشه‌ی کوتاه‌ترین مسیر $SPM(P', s)$ را محاسبه کن.

(۴) به ازای هر پاره‌خط ایجاد شده روی مرز مشترک P' و V_p به ریشه‌ی r :

• مسیر از s به r در دو درخت T و T' محاسبه کن.

• اگر دو مسیر یکسان بودند، پاره‌خط مورد نظر را به L_s اضافه کن.

زمان مورد نیاز برای انجام قدم اول $O(n)$ است که شامل بر محاسبه‌ی V_p و پیمایش مرز P' است که در زمان خطی قابل انجام هستند. مراحل دوم و سوم با دو بار اجرای الگوریتم هرشبرگر و سوری [۷۳]، یک بار در P و یک بار در P' انجام می‌شوند. هر بار اجرای این الگوریتم زمان و فضای از مرتبه‌ی $O(n \log n)$ نیاز دارد. در مرحله‌ی چهارم، به ازای $O(n)$ ریشه‌ی مورد بررسی، $O(\log n)$ زمان صرف بازیابی مسیر در دو درخت کوتاه‌ترین مسیر می‌شود. به این ترتیب، کل محاسبه در زمان $O(n \log n)$ قابل انجام است.

محاسبه‌ی بازه‌های بهینگی s و t

برای محاسبه‌ی بازه‌های بهینگی t روی دروازه‌ی s دیگر نمی‌توان داخل V_p را به عنوان مانع در نظر گرفت و دامنه را به P' محدود کرد. شکل ۵ نشان دهنده‌ی حالتی است که چنین روشی به مشکل بر می‌خورد. برای محاسبه‌ی این بازه‌ها قدم‌های زیر انجام می‌گیرد.

(۱) نقشه‌ی کوتاه‌ترین مسیر $M = SPM(P, t)$ را محاسبه کن و آن را در یک داده‌ساختار DCEL

ذخیره کن. علاوه بر این، M را برای پرس‌وجوی جایابی نقطه پیش‌پردازش کن.

(۲) به ازای هر پاره‌خط $\sigma \in L_s$:

• یکی از دو نقطه‌ی انتهایی σ را در M جایابی کن.

• با استفاده از اطلاعات ذخیره شده در ساختار ^۳DCEL، سلول‌های M را که σ را قطع می‌کنند به ترتیب پیمایش کن (هر تقاطع σ با یک سلول از M ، یک بازه در $L_{s,t}$ را مشخص می‌کند).

توضیح این که در قدم آخر، چون بازه‌های بهینگی s (مثل σ) یک به یک بررسی می‌شوند و تقاطع هر کدام با M محاسبه می‌شود، بازه‌های $L_{s,t}$ به یک‌باره به دست می‌آیند.

زمان و حافظه‌ی لازم برای ایجاد نقشه‌ی M و نگهداری آن $O(n \log n)$ است. پیش‌پردازش نقشه برای جایابی نقطه نیز $O(n \log n)$ زمان و حافظه نیاز دارد. به ازای هر پاره‌خط σ ، $O(\log n)$ صرف جایابی انتهای آن در M می‌شود و بعد از آن به تعداد بازه‌های ایجاد شده روی آن زمان صرف پیمایش سلول‌های متقاطع با آن می‌شود. از آنجا که تعداد کل این بازه‌ها $O(n)$ است، هزینه‌ی زمانی کل برابر $O(n \log n)$ خواهد بود.

به این ترتیب، با محاسبه‌ی L_s و از روی آن $L_{s,t}$ طبق آنچه گفته شد، مطابق الگوی کلی ارائه شده در بخش ۱.۳ یافتن کوتاه‌ترین مسیر ممکن می‌شود. با توجه به این که تعداد بازه‌های بهینگی $L_{s,t}$ ، $O(n)$ است، یافتن نقطه‌ی بهینه نیز در زمان خطی قابل انجام است. در مجموع، می‌توان نتیجه‌ی به دست آمده در این بخش را در قضیه‌ی زیر خلاصه کرد.

قضیه ۲.۳ با داشتن یک دامنه‌ی چندضلعی گونه‌ی P با n رأس و دو نقطه‌ی s و t و نقطه‌ی دید p درون آن، می‌توان طول کوتاه‌ترین مسیر قابل دید از p بین s و t را در زمان و حافظه‌ی $O(n \log n)$ به دست آورد. یک کوتاه‌ترین مسیر قابل دید از p نیز در زمان اضافی $O(k)$ قابل یافتن است که k تعداد یال‌های مسیر است.

^۳doubly connected edge list

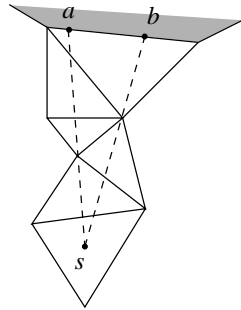
۴.۳ رویه‌های چندوجهی گونه

فرض کنید P یک رویه‌ی چندوجهی گونه باشد. با داشتن یک نقطه‌ی دید p که روی رویه یا خارج از آن قرار دارد، هدف ما یافتن کوتاه‌ترین مسیر بین دو نقطه‌ی داده شده‌ی s و t است طوری که حداقل یک نقطه در مسیر باشد که از p قابل دید است. همان طور که پیش از این اشاره شد، یافتن ناحیه‌ی قابل دید از p که آن را با V_p و مرز آن را با B_p نشان می‌دهیم در بدترین حالت $O(n^2)$ زمان نیاز دارد. در صورتی که چندوجهی مورد بررسی محدب باشد، این زمان به $O(n)$ کاهش می‌یابد. به همین نسبت، پیچیدگی V_p در حالت محدب $O(n)$ و در حالت مقعر $O(n^2)$ است. علاوه بر این در حالت مقعر، یال‌های V_p می‌توانند پاره‌خط‌هایی درون وجه‌های چندوجهی باشند. این پیچیدگی باعث می‌شود که مرتبه‌ی زمانی و فضای حافظه‌ی مورد نیاز الگوریتم یافتن کوتاه‌ترین مسیر قابل دید از p در حالت مقعر نسبت به حالت محدب ضریبی از مرتبه‌ی n بیشتر داشته باشد.

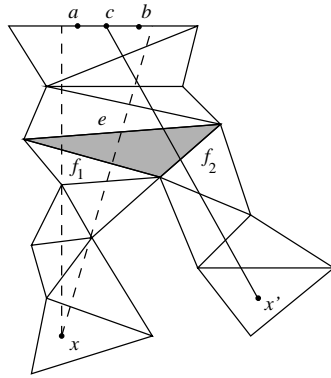
۱.۴.۳ بازه‌های بهینگی

در رویه‌های چندوجهی گونه، به طور مستقیم مفهومی مانند نقشه‌ی کوتاه‌ترین مسیر وجود ندارد، اما خواهیم دید چگونه از ساختارهایی که در الگوریتم‌های یافتن کوتاه‌ترین مسیر پیشنهاد شده می‌توان استفاده کرد تا بازه‌های بهینگی را پیدا کنیم.

تعریف بازه‌های بهینگی را همانند حالت دو بعدی بر مبنای ساختار ترکیباتی کوتاه‌ترین مسیرها از s و t به نقاط مرزی V_p قرار می‌دهیم و برای آن از خواص ذکر شده در بخش ۲.۱.۲ استفاده می‌کنیم. برای سادگی، ابتدا تعریف این بازه‌ها را برای چندضلعی‌های محدب ارائه می‌کنیم و بعد آن را به چندضلعی‌های مقعر تعمیم می‌دهیم. با توجه به لم‌های ۲.۲ و ۳.۲، یک کوتاه‌ترین مسیر روی رویه‌ی یک چندوجهی محدب از هیچ یک از رئوس چندوجهی نمی‌گذرد و در نتیجه توسط دنباله‌ی یالی که آن را متصل می‌کند به طور کامل مشخص می‌شود.



شکل ۶: یک بازه‌ی بهینگی متناظر با s روی یک یال B_p است (بخش خاکستری، قسمتی از V_p است). کوتاه‌ترین مسیرها از s به نقاط روی این بازه از وصل کردن s به آن نقاط روی تصویر بی‌تاشده‌ی دنباله یال رسم شده در شکل به دست می‌آیند.



شکل ۷: اثبات لم ۷.۳ - a و b دو نقطه از یک بازه‌ی بهینگی متناظر با x هستند. x' تصویر مبدأ در یک بی‌تاشده‌ی دیگر است.

به این منظور زیرمجموعه‌ای از B_p که کوتاه‌ترین مسیرها از s به نقاط آن دنباله یال یکسانی را متصل می‌کند، یک بازه‌ی بهینگی متناظر با s می‌نامیم (شکل ۶). بازه‌ی بهینگی متناظر با t نیز به نحو مشابهی تعریف می‌شود. لم زیر نشان می‌دهد که چنین زیرمجموعه‌ای واقعاً یک بازه است.

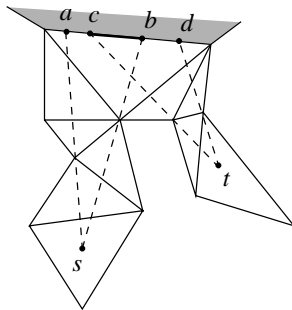
لم ۷.۳ به ازای یک نقطه‌ی دلخواه x روی سطح چندوجهی گونه، مجموعه نقاطی از یک یال B_p که کوتاه‌ترین مسیر از x به آنها دنباله یال یکسانی را متصل می‌کند، یک بازه‌ی همبند را تشکیل می‌دهند.

اثبات. فرض کنید I مجموعه نقاطی روی یک یال B_p باشد که کوتاه‌ترین مسیر از x به نقاط I دنباله یال یکسان $E = (e_1, e_2, \dots, e_k)$ را متصل می‌کنند. فرض کنید a و b دو نقطه از I باشند. یک

نقطه‌ی c را در نظر بگیرید که روی یال مورد بررسی از B_p بین a و b واقع شده باشد. اگر $c \notin I$ ، کوتاه‌ترین مسیر از x به c دنباله یال متفاوتی از E مانند $E' = (e'_1, e'_2, \dots, e'_l)$ را متصل می‌کند (شکل ۷). طولانی‌ترین زیردنباله‌ی مشترک E و E' را از انتهای دنباله‌ها در نظر بگیرید. چون هر سه نقطه‌ی a ، b و c روی یک یال از B_p قرار دارند، این زیردنباله ناتهی است. فرض کنید e نخستین یال از این زیردنباله باشد و f_1 و f_2 دو یال دیگر از وجه مجاور e باشند که در طرف مقابل نقاط I (و c) نسبت به e واقع شده است. از آنجا که E و E' در یال‌های قبل از e با هم متفاوتند، $f_1 \in E$ و $f_2 \in E'$ یا برعکس. بنابراین، کوتاه‌ترین مسیرها از x به نقاط I یکی از این دو یال و کوتاه‌ترین مسیر از x به c یال دیگر را قطع می‌کنند. چون a و b در دو طرف c قرار دارند و تصویر بی‌تای کوتاه‌ترین مسیرها به شکل پاره‌خط است، کوتاه‌ترین مسیر به c ، کوتاه‌ترین مسیر به a یا b را قطع می‌کند که این خلاف لم ۴.۲ است. بنابراین به ازای هر دو نقطه از I ، تمام نقاط بین آنها نیز در I است و این همبندی این مجموعه را ثابت می‌کند. \square

در حالتی که رویه‌ی چندوجهی‌گونه محدب نباشد، این فرض که کوتاه‌ترین مسیر از s یا t به هر نقطه قابل بی‌تای کردن به یک پاره‌خط است درست نخواهد بود. در حقیقت چنین مسیری طبق لم ۳.۲ به طور متناوب از رأس‌های رویه و دنباله یال‌ها می‌گذرد. برای یک نقطه‌ی دلخواه a از P ، کوتاه‌ترین مسیر از s به P توسط دنباله‌ی $(s = v_1, E_1, v_2, E_2, \dots, v_k, E_k, v_{k+1} = a)$ مشخص می‌شود که E_i یک دنباله یال (احتمالاً تهی) است که توسط بخشی از کوتاه‌ترین مسیر بین v_i و v_{i+1} متصل می‌شود. در این حالت مجموعه نقاط a روی یک یال از B_p را که v_k و E_k برای آنها یکسان است در نظر بگیرید. با استدلالی مشابه لم ۷.۳ این مجموعه یک بازه روی یال یادشده را تشکیل می‌دهد. چنین بازه‌ای را یک بازه‌ی بهینگی متناظر با s می‌نامیم.

بازه‌های بهینگی متناظر با s و t که مجموعه‌ی $L_{s,t}$ را تشکیل می‌دهند مشابه قبل قابل تعریف هستند (شکل ۸). لم زیر نشان می‌دهد بازه‌های بهینگی در رویه‌های چندوجهی‌گونه نیز مانند



شکل ۸: پاره خط ab یک بازه‌ی بهینگی در L_s و پاره خط cd یک بازه در L_t است. اشتراک این دو پاره خط یعنی cb در $L_{s,t}$ خواهد بود.

دامنه‌های دو بعدی به یافتن نقطه‌ی تماس بهینه کمک می‌کنند.

لم ۸.۳ اگر I یک بازه در $L_{s,t}$ باشد، آنگاه می‌توان نقطه‌ی $q(I)$ از I را که کمترین مجموع فاصله‌ها را از s و t دارد در زمان ثابت پیدا کرد.

اثبات. فرض کنید آخرین رأسی از P که کوتاه‌ترین مسیرها از s به نقاط I از آن می‌گذرند u و آخرین رأسی از P که کوتاه‌ترین مسیرها از t به نقاط I از آن می‌گذرند v باشد (برای چندوجهی محدب، $u = s$ و $v = t$). با بی‌تا کردن دو دنباله یالی که مسیرهای دسته‌ی اول از u تا I متصل می‌کنند و دنباله یالی که مسیرهای دسته‌ی دوم از v تا I متصل می‌کنند، دو صفحه به دست می‌آید. در حالتی که این دو دسته مسیر از یک طرف به I برسند، این دو صفحه با هم یکی است و توسط آخرین وجهی که دو دسته مسیر ملاقات می‌کنند تا به I برسند تعیین می‌شود. در این حالت یافتن نقطه‌ی بهینه‌ی $q(I)$ مانند حالت دو بعدی طبق لم ۱.۳ در زمان ثابت قابل اجرا است. اگر دو دسته مسیر از دو طرف به I برسند، کافی است یکی از صفحات را دوران دهیم تا با صفحه‌ی اول منطبق شود. در این حالت $q(I)$ از تقاطع پاره خط uv و خط شامل I در صفحه‌ی به دست آمده تعیین می‌شود. اگر این تقاطع داخل I رخ دهد، $q(I)$ همان نقطه‌ی تقاطع است و گرنه مانند حالت دو بعدی یکی از دو انتهای I خواهد بود. □

۲.۴.۳ محاسبه‌ی بازه‌های بهینگی

برای محاسبه‌ی بازه‌های بهینگی متناظر با s ، از روش چن و هان [۶۴] برای ساخت نظریف رویه‌ی چندوجهی‌گونه استفاده می‌کنیم. در این روش نشان داده شده است که چگونه برای یک رویه‌ی چندوجهی‌گونه‌ی P و نقطه‌ی مبدأ s ، می‌توان درخت کوتاه‌ترین مسیر را در زمان $O(n^2)$ و حافظه‌ی $O(n)$ ساخت. در این درخت اطلاعاتی ذخیره می‌شود که به ازای هر رأس P دنباله یالی که توسط کوتاه‌ترین مسیر از s به آن رأس متصل می‌شود، در زمانی متناسب با اندازه‌ی این دنباله یال قابل بازیابی است.

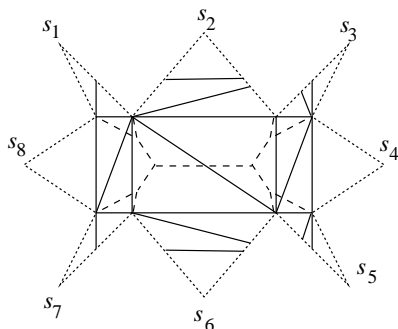
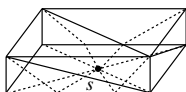
در این جا دوباره حالت‌های محدب و مقعر P را تفکیک می‌کنیم و ابتدا به حالت محدب می‌پردازیم و سپس حالت مقعر را نیز بررسی می‌کنیم.

رویه‌های محدب

اگر P را در امتداد کوتاه‌ترین مسیرها از s به تمام رأس‌های P برش دهیم، رویه‌ای حاصل می‌شود که می‌توان آن را طوری بی‌تا کرد که به طور کامل در یک صفحه قرار گیرد. شکل مسطح حاصل چیدمان روبه‌داخل P^4 نام دارد (در [۷۴] به آن بی‌تاشده‌ی ستاره‌ای نیز گفته می‌شود). مرز این شکل یک چندضلعی ستاره‌شکل است. رأس‌های این چندضلعی، رأس‌های P به همراه تصاویر بی‌تاشده‌ی s هستند. یال‌های این چندضلعی، بی‌تاشده‌ی کوتاه‌ترین مسیرها از s به رأس‌های P هستند.

با تعریف نظریفی از چیدمان روبه‌داخل P می‌توان رویه‌ی P را به ناحیه‌هایی تقسیم کرد که کوتاه‌ترین مسیرها از s به نقاط آنها ساختار ترکیبیاتی ثابتی داشته باشند. این نظریف نقشی شبیه به نقشه کوتاه‌ترین مسیر در دامنه‌های دو بعدی بازی می‌کند. برای به دست آوردن این نظریف، نمودار وورونوی چیدمان روبه‌داخل را رسم می‌کنیم طوری که تصویرهای s محل‌های وورونوی محسوب

^۴ inward layout

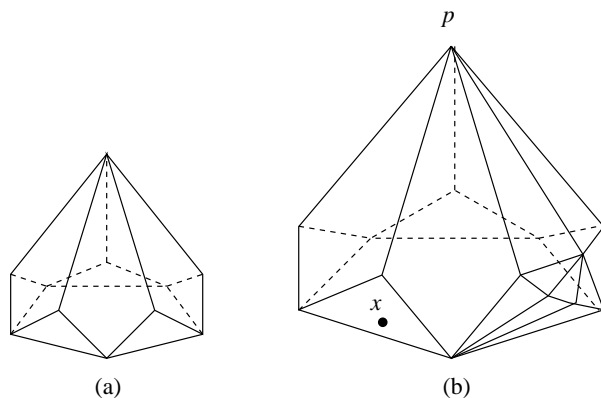


شکل ۹: بالا - کوتاه‌ترین مسیرها از مبدأ s به رئوس یک مکعب مستطیل با نقطه چین مشخص شده اند (s روی وجه زیرین واقع شده است). پایین - چیدمان رو به داخل شکل بالا - s_1, \dots, s_8 تصاویر مبدأ روی بی‌تاشده‌های مختلف هستند. یال‌های نقطه‌چین برش‌یافته‌ی کوتاه‌ترین مسیرها و پاره‌خط‌های خط‌چین، یال‌های نمودار وورونوی هستند.

شوند (شکل ۹). این نمودار سطح P را به سلول‌هایی تقسیم می‌کند که تمام نقاط یک سلول، به تصویر s متناظر با آن سلول نزدیک‌ترند تا تصاویر دیگر s . به این ترتیب می‌توان نشان داد که دنباله یال متصل شده توسط کوتاه‌ترین مسیر از s به تمام این نقاط نیز دنباله یال ثابتی را متصل می‌کند. به این ترتیب، مجموعه بازه‌های بهینگی L_s از تقاطع B_p با یال‌های این نظریف به دست می‌آیند. بازه‌های بهینگی L_t هم با روشی مشابه قابل به دست آوردن هستند.

برای به دست آوردن مرتبه‌ی اندازه‌ی مجموعه‌های L_s و L_t ، توجه کنید که نظریف ساخته شده $O(n)$ یال دارد ($O(n)$ یال بی‌تاشده‌ی کوتاه‌ترین مسیرها و $O(n)$ یال مربوط به نمودار وورونوی هستند). اندازه‌ی B_p نیز در این حالت $O(n)$ است و در حالت کلی وجود $O(n^2)$ بازه‌ی بهینگی متناظر با هر یک از s و t ممکن است. شکل ۱۰ حالتی را نشان می‌دهد که رسیدن به چنین حدی ممکن است. بنابراین در کل $O(n^2)$ بازه‌ی بهینگی وجود دارد.

محاسبه‌ی این بازه‌ها با استفاده از الگوریتم چن و هان در زمان $O(n^2)$ ممکن است، چون این الگوریتم در جریان اجرا نقاط تقاطع کوتاه‌ترین مسیرها به رئوس P و یال‌های P را نگهداری می‌کند. به دست آوردن تقاطع یال‌های نمودار وورونوی و مرز B_p نیز در زمان ساختن نمودار



شکل ۱۰: چپ - یک رویه‌ی چندوجهی گونه‌ی محدب - تمام وجه‌های مجاور p ، V_p را تشکیل می‌دهند. راست - اضافه کردن $O(n)$ وجه به یکی از وجه‌های پایینی رویه‌ی سمت چپ، تعداد بازه‌های بهینگی x را $O(n^2)$ خواهد کرد.

وورونوی ممکن خواهد بود. علاوه بر این، الگوریتم یادشده، کوتاه‌ترین مسیرها به رئوس را به ترتیب زاویه‌ای حول s مرتب نگهداری می‌کند. به همین خاطر ادغام کردن بازه‌های L_t و L_s در زمان $O(n^2)$ ممکن است.

با داشتن مجموعه‌ی $L_{s,t}$ و استناد به لم ۸.۳، سایر محاسبات طبق الگوی کلی انجام می‌شود.

قضیه ۳.۳ اگر P رویه‌ی یک چندوجهی محدب باشد و p یک نقطه دلخواه روی P یا خارج آن باشد، آنگاه یافتن کوتاه‌ترین مسیر بین دو نقطه‌ی s و t طوری که p از آن قابل دیدن باشد در زمان حافظه‌ی $O(n^2)$ قابل انجام است.

رویه‌های مقعر

همان‌گونه که پیش از این مطرح شد، کوتاه‌ترین مسیر بین دو نقطه روی یک رویه‌ی مقعر ممکن است از رئوس چندوجهی نیز بگذرد. فرض کنید v آخرین رأس یک کوتاه‌ترین مسیر بین s و x باشد. در اینجا می‌توان رأس v را به عنوان یک شبه‌مبدأ انگاشت که مقدار اولیه‌ای برابر طول کوتاه‌ترین مسیر از s به v داشته باشد. در این حالت درخت کوتاه‌ترین مسیر اندکی با حالت محدب متفاوت است و دارای گره‌هایی متناظر با شبه‌رأس‌های P خواهد بود. با این حال زمان و حافظه‌ی

لازم برای ساخت درخت تفاوتی با حالت محدب ندارد. برای ساختن نظریف رویه، لازم است در ساخت نمودار وورونوی، تصاویر مبدأ و شبه‌مبدأها به عنوان محل‌های وورونوی در نظر گرفته شوند. توضیح این که در حالت مقعر ممکن است چیدمان روبه‌داخل P هنگامی که روی یک صفحه نشانده می‌شود با خودش هم‌پوشانی^۵ داشته باشد. این وضعیت را باید هنگام ساختن نمودار وورونوی در نظر گرفت (ملاحظات مربوط در [۶۴] آمده است).

برخلاف حالت محدب که B_p مجموعه‌ای از یال‌های P با اندازه‌ی $O(n)$ است، در اینجا ممکن است تعداد یال‌های B_p از مرتبه‌ی n^2 باشد و لزوماً بر یال‌های P نیز منطبق نباشد. به همین دلیل نمی‌توان از اطلاعات به دست آمده در الگوریتم ساختن نظریف برای یافتن نقاط تقاطع B_p و نظریف استفاده کرد. در اینجا از الگوریتم بهینه‌ی فینکه^۶ و هینریکز^۷ [۷۵] برای روی هم‌نشانی دو نظریف استفاده می‌کنیم. زمان لازم برای اجرای این الگوریتم $O(n+k)$ است که در آن k تعداد نقاط تقاطع دو نظریف است. این تعداد در مسئله‌ی ما $O(n^2)$ است. الگوریتم یادشده نظریف خروجی را در داده‌ساختار نمای چهارگانه^۸ [۷۶] ذخیره می‌کند که پیمایش مرتب بازه‌های L_t و L_s را توسط عملیات پیمایش حلقه‌های رأسی و حلقه‌های یالی در زمانی متناسب با اندازه‌ی این مجموعه‌ها مهیا می‌کند. به این ترتیب محاسبه‌ی هر یک از این دو مجموعه بازه‌ی بهینگی در زمان $O(n^3)$ قابل انجام است و ادغام آنها در مجموعه‌ی $L_{s,t}$ نیز در همین مرتبه‌ی زمانی انجام می‌شود. به این ترتیب با پیمایش بازه‌های $L_{s,t}$ در زمان $O(n^3)$ و پی‌گیری الگوی کلی ارائه شده، نتیجه‌ی زیر به دست می‌آید.

قضیه ۴.۳ اگر P رویه‌ی یک چندوجهی مقعر باشد و p یک نقطه دلخواه روی P یا خارج آن باشد،

overlap^۵

Finke^۶

Hinrichs^۷

quad view data structure^۸

آنگاه یافتن کوتاه‌ترین مسیر بین دو نقطه‌ی s و t طوری که p از آن قابل دیدن باشد در زمان و حافظه‌ی $O(n^3)$ قابل انجام است.

کوتاه‌ترین مسیرها با قید قابلیت دید نقطه‌ی پرس و جو

در این فصل به بررسی مسئله‌ی یافتن کوتاه‌ترین مسیرهایی می‌پردازیم که مقید به دیدن یک نقطه‌ی پرس و جو هستند. به تعبیر دقیق‌تر، هدف مسئله پیش‌پردازش دامنه است طوری که به ازای هر نقطه‌ی دیدی که به عنوان پرس و جو مطرح می‌شود، بتوان در زمان کوتاهی کوتاه‌ترین مسیر با قابلیت دید آن نقطه را محاسبه کرد. در اینجا این مسئله فقط در دامنه‌ی چندضلعی ساده مورد مطالعه قرار می‌گیرد ولی در انتهای فصل به دامنه‌های دیگر نیز اشاره خواهد شد.

در چندضلعی‌های ساده دو شکل از مسئله را در نظر خواهیم گرفت. شکل اول، یافتن کوتاه‌ترین مسیری است که باید از نقطه‌ی مبدأ s طی کرد تا این که نقطه‌ی پرس و جوی q دیده شود. در این صورت از مسئله، این که مسیر به کجا ختم می‌شود اهمیتی ندارد و تنها مهم است که در کوتاه‌ترین زمان به مرز چندضلعی قابلیت دید q برسیم. در صورت دوم، پس از دیدن q باید به نقطه‌ی مقصد داده شده‌ی t برگردیم و با داشتن این قید می‌خواهیم کوتاه‌ترین مسیر را پیدا کنیم. اصطلاحاً می‌گوییم صورت اول مسئله با مقصد باز و صورت دوم با مقصد مشخص روبه‌رو هستیم. در ادامه این دو مسئله را به طور جداگانه بررسی می‌کنیم.

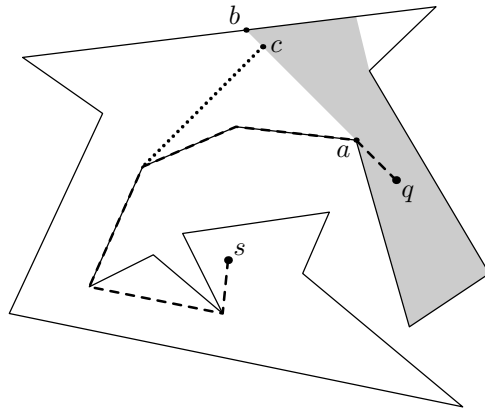
۱.۴ کوتاه‌ترین مسیر با مقصد باز

فرض کنید P یک چندضلعی ساده با n رأس باشد و نقطه‌ی مبدأ s در آن داده شده باشد. هدف مسئله، پیش‌پردازش جفت (P, s) است که توسط آن بتوان به پرس‌وجویی از این دست پاسخ داد: با داشتن یک نقطه‌ی پرس‌وجوی $q \in P$ ، طول کوتاه‌ترین مسیری را پیدا کنید که با طی کردن آن از s بتوان q را دید. به عبارت دیگر می‌خواهیم نقطه‌ای مانند c را که از q قابل دید است پیدا کنیم طوری که کوتاه‌ترین فاصله ژئودزیک از s را داشته باشد. توجه داشته باشید که اگر q از s قابل دیدن باشد، همان s است و نیازی به محاسبه‌ی مسیر نخواهد بود. ما هنگام بیان قدم‌های الگوریتم در زمان پاسخ به پرس‌وجو اشاره خواهیم کرد که چگونه باید این حالت را تشخیص دهیم، ولی تا آن زمان فرض می‌کنیم q و s از هم قابل دید نیستند.

از آنجا که طبق روش گفته شده در فصل قبل، پرس‌وجویی از این دست بدون پیش‌پردازش در زمان خطی قابل پاسخ‌گویی است، هدف ما رسیدن به زمان پاسخ $O(\log n)$ برای پرس‌وجوها است. به همین منظور دو نوع پرس‌وجو تعریف می‌کنیم: یکی برای یافتن کوتاه‌ترین فاصله‌ی ژئودزیک و یکی برای یافتن مسیری که این طول را داشته باشد. هدف ما این است که اولی را در زمان لگاریتمی و دومی را با پرداخت هزینه‌ی اضافی متناسب با طول مسیر به دست آمده پردازش کنیم. فعلاً توضیحات ما پیرامون پرس‌وجوی نوع اول است و در زمان لازم در مورد نوع دوم توضیحاتی ارائه می‌کنیم.

۱.۱.۴ مفاهیم پایه‌ای

ما از نماد V_q به عنوان چندضلعی قابلیت دید q در P استفاده می‌کنیم و فرض می‌کنیم $\pi(x, y)$ نشان دهنده‌ی کوتاه‌ترین مسیر بین x و y در P است. چون فرض کرده‌ایم s از q قابل دید نیست، در یکی از جیب‌های چندضلعی که از برداشتن V_q از P حاصل می‌شوند قرار دارد. فرض کنیم w پنجره‌ای



شکل ۱: بخش خاکستری V_q است. پنجره ab مبدأ s را از V_q جدا می‌کند. کوتاه‌ترین مسیر بین s و q با خط چین ضخیم نشان داده شده است. c نقطه‌ای با کوتاه‌ترین فاصله از s است که q را می‌بیند.

است که این جیب را از V_q جدا می‌کند. به سادگی می‌توان مشاهده کرد که نقطه‌ی c روی w قرار دارد. به طور دقیق‌تر، c نقطه‌ای از w است که کوتاه‌ترین فاصله را از s دارد (شکل ۱). از این پس،

چنین نقطه‌ای را نقطه‌ی بهینه‌ی $c(w)$ می‌نامیم تا صریحاً نشان دهیم به چه پنجره‌ای تعلق دارد.

ما از مفهوم قیف همان‌گونه که در [۵۴] تعریف شده است استفاده خواهیم کرد. فرض کنید

a و b دو انتهای پنجره‌ی w باشند. قیف $F(w)$ را به عنوان $\pi(r, a) \cup \pi(r, b)$ تعریف می‌کنیم

که r آخرین رأس مشترک بین دو مسیر $\pi(s, a)$ و $\pi(s, b)$ است هنگامی که از s به سمت a و

b دیده شوند (شکل ۲). فرض کنید رأس‌های قیف به ترتیب پیمایش از a به b دنباله‌ی

$a = v_0, v_1, \dots, v_k, v_{k+1} = b$ را تشکیل می‌دهند. ناحیه‌ی محصور بین $F(w)$ و w را با امتداد

دادن یال‌های $F(w)$ تا این که w را قطع کنند، می‌توان به ناحیه‌های مثلث شکل تقسیم کرد.

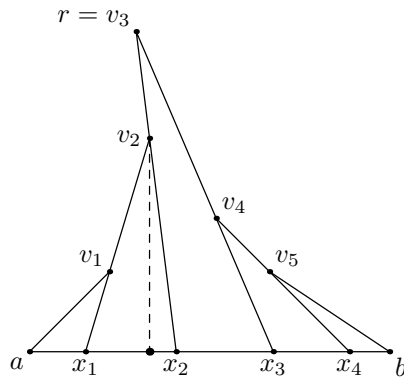
فرض کنید امتداد یال $w, v_i v_{i+1}$ ($0 \leq i \leq k$) را در x_i قطع کند (در نتیجه $x_0 = a$ و $x_k = b$).

به این ترتیب، آخرین رأس روی کوتاه‌ترین مسیرهای بین s تا نقاط روی پاره‌خط $x_i x_{i+1}$

خواهد بود. دنباله زاویه‌های بین امتداد یال‌ها و پنجره‌ی w را با $(\theta_0, \theta_1, \dots, \theta_k)$ نشان می‌دهیم که

$\theta_i = \angle b x_i v_i$ ($0 \leq i \leq k-1$) و $\theta_k = 180^\circ - \angle a v_k b$ از تحدب رو به بیرون دیواره‌های قیف،

مشاهده‌ی ۱.۴ به سادگی پذیرفته می‌شود.



شکل ۲: قیف $F(w)$ روی پنجره‌ی $w = ab$ - نقطه‌ی بهینه‌ی $c(w)$ پای عمود از v_2 به ab است.

مشاهده ۱.۴ دنباله زاویه‌های بین امتداد یال‌های قیف و پنجره‌ی $w = (\theta_0, \theta_1, \dots, \theta_k)$ ، یک دنباله‌ی صعودی است.

مبتنی بر این مشاهده و بسته به این که برای دنباله زاویه‌های یادشده کدام یک از حالت‌های زیر برقرار باشد، نقطه‌ی بهینه‌ی $c(w)$ مشخص می‌شود:

(۱) یکی از زاویه‌ها مثل θ_i مساوی 90° است. در این حالت $c(w) = x_i$.

(۲) دو زاویه‌ی مجاور $\theta_i < 90^\circ$ و $\theta_{i+1} > 90^\circ$ وجود دارند. در این حالت $c(w)$ پای عمود از v_{i+1} به w است.

(۳) تمام زاویه‌ها در دنباله از 90° بزرگ‌ترند. در این حالت $c(w) = a$.

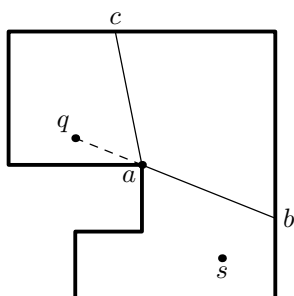
(۴) تمام زاویه‌ها در دنباله از 90° کوچک‌ترند. در این حالت $c(w) = b$.

۲.۱.۴ مراحل الگوریتم

هنگام دریافت یک نقطه‌ی پرس‌وجوی q ، قدم‌های زیر طی می‌شود:

(۱) کنترل این که آیا q از s قابل دیدن است یا نه.

(۲) محاسبه‌ی پنجره‌ی w که s را از V_q جدا می‌کند.



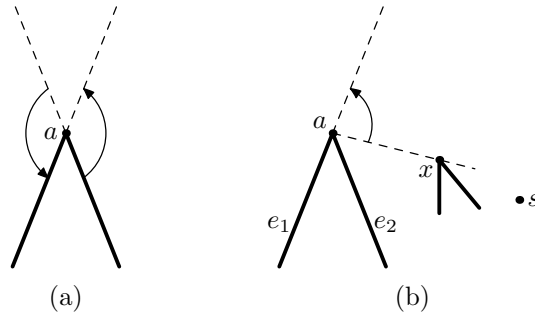
شکل ۳: پاره خط ab یک پنجره‌ی جدا کننده است در صورتی که ac چنین خاصیتی ندارد.

(۳) محاسبه‌ی نقطه‌ی بهینه‌ی $c(w)$.

برای این که بتوان پرس و جو را به صورت کارا پاسخ داد، لازم است هر سه قدم در زمان لگاریتمی قابل اجرا باشند. با فرض این که $SPM(s)$ در پیش پردازش محاسبه شده باشد، برای کنترل قدم اول کافی است q را در نقشه‌ی یادشده جایابی کنیم. اگر s ریشه‌ی سلول شامل q باشد، آنگاه دو نقطه از هم قابل دیدند و در این حالت الگوریتم خاتمه پیدا می‌کند. برای انجام قدم دوم، از این نکته استفاده می‌کنیم که پنجره‌ای که s را از V_q جدا می‌کند با آخرین رأس P روی کوتاه‌ترین مسیر از s به q مشخص می‌شود (شکل ۱). بنابراین، هر دو قدم اول و دوم با جایابی q در نقشه‌ی کوتاه‌ترین مسیر در زمان $O(\log n)$ قابل انجام هستند.

برای یافتن نقطه‌ی بهینه‌ی $c(w)$ ، در مرحله‌ی پیش پردازش تمام پنجره‌های جدا کننده را در نظر می‌گیریم. منظور از یک پنجره‌ی جدا کننده، پنجره‌ای از چندضلعی قابلیت دید نقطه‌ای مانند x است که s را از V_x جدا کند. برای مثال در شکل ۳، پاره خط ab یک پنجره‌ی جدا کننده است، چون نقطه‌ای (مانند q در شکل) وجود دارد که از s قابل دید نیست و V_q از s که s را از آن جدا می‌کند. در مقابل، چنین نقطه‌ای برای پاره خط ac وجود ندارد و باعث می‌شود نتوان آن را یک پنجره‌ی جدا کننده به حساب آورد.

برای مشخص کردن تمام پنجره‌های جدا کننده، فرض کنید a یک رأس انعکاسی از P باشد (رأسی که زاویه‌ای که داخل P می‌سازد از 180° بیشتر باشد). اگر تمام نقاط پرس و جوی ممکن در

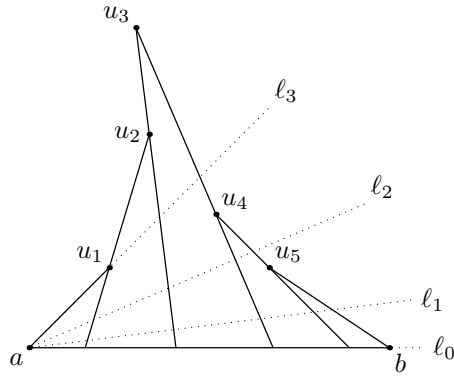


شکل ۴: چپ: دو بازه‌ی زاویه‌ای مشخص کننده پنجره‌های حول رأس انعکاسی a . راست: تک بازه‌ای که مشخص کننده‌ی پنجره‌های جدا کننده حول a است. x آخرین رأس روی $\pi(s, a)$ است.

P را در نظر بگیریم، مجموعه‌ای از پنجره‌ها را متناظر با a ایجاد می‌کنند که از امتداد پاره‌خط واصل آنها به a به دست می‌آیند. این مجموعه با دو بازه‌ی زاویه‌ای که از امتداد دادن دو یال حادث به a به دست می‌آیند مشخص می‌شود (شکل ۴ - چپ). اما تمام پنجره‌هایی که در این مجموعه هستند، پنجره‌های جدا کننده نیستند. برای محدود کردن این مجموعه به پنجره‌های جدا کننده، فرض کنید x آخرین رأسی است که $\pi(s, a)$ از آن می‌گذرد. اگر x خارج از هر دو بازه‌ی زاویه‌ای واقع شود، هیچ پنجره‌ی جدا کننده‌ای حول a وجود ندارد. در غیر این صورت، فرض کنید x داخل بازه‌ای که بین e_2 و امتداد e_1 تشکیل شده است قرار بگیرد (e_1 و e_2 دو یال حادث به x هستند - شکل ۴ - راست). در این حالت، بازه‌ی زاویه‌ای که محدود به امتداد e_1 و پاره‌خط ax است، مشخص کننده‌ی مجموعه پنجره‌های جدا کننده‌ی ممکن است که آن را با $Sep(a)$ نمایش می‌دهیم.

هدف ما این است که به ازای هر رأس انعکاسی P مثل a ، مجموعه پنجره‌های جدا کننده‌ی متناظر با a را به مجموعه‌هایی افراز کنیم که با دانستن مجموعه‌ای که پنجره‌ی w به آن تعلق دارد، بتوان به طرز کارایی نقطه‌ی بهینه‌ی $c(w)$ را به دست آورد. برای این کار، طی یک جاروی شعاعی^۱ حول a ، سعی می‌کنیم این افراز را به دست آوریم. دو نوع رخداد در این جارو تعریف می‌شوند: زاویه‌هایی که در آن آخرین رأس $\pi(s, c)$ تغییر می‌کند (رخدادهای بازه) و زاویه‌هایی که در آن ساختار قیف عوض می‌شود (رخدادهای قیف). در ادامه هر یک از این دو نوع رخداد را بررسی

^۱ radial sweep



شکل ۵: افراز پنجره‌های جدا کننده حول a - بازه‌ی زاویه‌ای محدود به l_0 و l_3 مجموعه‌ی تمام پنجره‌های جدا کننده را تعریف می‌کند. l_1 و l_2 به ترتیب عمودهای صادر شده از a به امتدادهای v_3v_4 و v_2v_3 هستند.

می‌کنیم.

رخدادهای بازه

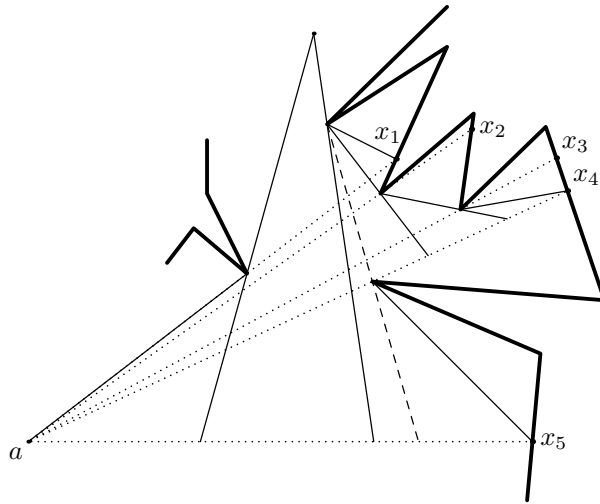
فرض کنید $Sep(a)$ توسط بازه‌ی زاویه‌ای $[\alpha, \alpha']$ تعریف شده باشد. برای $\alpha \leq \varphi \leq \alpha'$ فرض کنید w_φ نماینده‌ی پنجره‌ای با نقطه‌ی انتهایی a و زاویه‌ی φ باشد. عمودهای صادر شده از a به امتدادهای دو یال از قیف مانند $v_i x_i$ و $v_{i+1} x_{i+1}$ را در نظر بگیرید. اگر پنجره‌ی w_φ بین این دو عمود واقع شود، می‌توان گفت $\theta_i < 90^\circ$ و $\theta_{i+1} > 90^\circ$. بنابراین، نقطه‌ی بهینه‌ی $c(w_\varphi)$ پای عمود از v_{i+1} به w_φ است. برای مثال، در شکل ۵، l_1 و l_2 عمودهای صادر شده به امتدادهای v_3v_4 و v_2v_3 هستند. بنابراین، برای یک پنجره‌ی فرضی بین l_1 و l_2 نقطه‌ی بهینه، پای عمود از v_3 به پنجره است. به این ترتیب، مجموعه عمودهای صادر شده به امتداد یال‌های قیف، رخدادهای بازه را مشخص می‌کنند که در آن آخرین رأس $\pi(s, c)$ تغییر می‌کند. از آنجا که دوران یک پنجره حول رأس انعکاسی که یک سر آن است به سمت نوک قیف زاویه‌های بین امتداد یال‌های قیف و پنجره‌ها کاهش می‌دهد، تعداد رخدادهای بازه $O(n)$ است.

رخدادهای قیف

به این نکته توجه کنید که مجموعه نقاط روی پنجره‌های جدا کننده $Sep(a)$ زیرمجموعه‌ای از چندضلعی قابلیت دید a یعنی V_a است که توسط w_α و w'_α محدود شده باشد. به سادگی می‌توان مشاهده کرد که لحظاتی که در آن پنجره‌ی جاروکننده از رأس‌های V_a عبور می‌کند، ساختار قیف تغییر می‌کند. بنابراین، برای هر رأس انعکاسی، حداکثر $O(n)$ رخداد قیف وجود دارد. مسئله این است که بتوانیم در این لحظات ساختار قیف را به طرز کارایی بهنگام سازیم. فرض می‌کنیم $SPT(s)$ و $SPM(s)$ و V_a از قبل محاسبه شده‌اند. کلید بهنگام سازی کارای ساختار قیف این است که فرایند جارو را از نزدیک‌ترین پنجره‌ی جدا کننده به s شروع کنیم. طبق نمادگذاری ما، یکی از w_α و w'_α این خاصیت را دارند که در اینجا فرض می‌کنیم w_α است.

در ابتدا قیف را روی w_α محاسبه می‌کنیم. در حالی که پنجره‌ی جارو حول a دوران می‌کند، ممکن است به رأس جدیدی از V_a مانند p برسیم. اگر p یک رأس انعکاسی P نباشد، تنها تأثیر این رخداد تغییر کردن یالی است که انتهای متحرک پنجره جارو روی آن حرکت می‌کند. در غیر این صورت، ما به یک گره‌ی میانی از $SPT(s)$ رسیده‌ایم و این ممکن است رأس‌هایی به قیف اضافه کند. اضافه شدن رأس جدید ممکن است موجب حذف بعضی رأس‌ها از قیف نیز بشود. این در حالتی اتفاق می‌افتد که والد رأس تازه اضافه شده قبل از اضافه کردن رأس جدید برگ نباشد. برای مثال در شکل ۶، قیف اولیه روی ax_1 بنا می‌شود. هنگامی که پنجره‌ی جارو نقاط x_2 و x_3 را ملاقات می‌کند، رأس‌های جدیدی به قیف اضافه می‌شوند. در x_4 ، یک رأس جدید به قیف اضافه می‌شود و پاره‌خط چین نیز به عنوان امتداد یکی از یال‌های قیف به مجموعه امتداد یال‌ها اضافه می‌شود. در همین هنگام تعدادی از رأس‌ها از قیف حذف می‌شوند.

برای اعمال تغییر در ساختار قیف، لازم است همیشه آخرین رأس انعکاسی ملاقات شده از V_a را به خاطر داشته باشیم. با داشتن این اطلاع، تغییری را که باید در ساختار قیف اعمال شود



شکل ۶: رخدادهای قیف برای یک رأس انعکاسی a - بازه‌ی زاویه‌ای بین ax_1 و ax_5 محدوده‌ی پنجره‌های جدا کننده را تعریف می‌کند. جارو از ax_1 آغاز می‌شود. خطوط ممتد ضخیم، یال‌های P هستند. خطوط ممتد نازک امتداد یال‌های قیف هستند. خطوط نقطه‌چین وضعیت پنجره‌ی جارو را در رخدادهای قیف نشان می‌دهند. پاره‌خط خط‌چین امتداد اضافه شده پس از ملاقات x_4 است.

می‌توان در زمان ثابت محاسبه کرد. همچنین، لازم است رأس‌های قیف اولیه را به طور مرتب داشته باشیم. برای این کار باید فراخوانی‌های بازگشتی در جریان پیمایش عمق‌اول در الگوریتم ساخت نقشه‌ی کوتاه‌ترین مسیر در جهت مشخصی (مثلاً ساعت‌گرد) انجام شوند. از آنجا که ساختار قیف‌ها در داده‌ساختار درخت جستجوی انگشتی ذخیره می‌شود بازیابی رأس‌های قیف در زمان خطی ممکن است.

مطابق آنچه گفته شد، برای یک رأس a مجموعه‌ای از $O(n)$ بازه‌ی زاویه‌ای داریم که به طور مرتب ذخیره شده‌اند. بنابراین هنگام دریافت پنجره‌ی پرس و جو، می‌توان بازه‌ی حاوی آن را با جستجوی دودویی به دست آورد. پس از آن به دست آوردن نقطه‌ی بهینه در زمان ثابت انجام می‌شود.

به طور خلاصه، قدم‌های طی شده در مرحله‌ی پیش‌پردازش به این شرحند:

(۱) $SPT(s)$ و $SPM(s)$ را محاسبه کن.

(۲) برای هر رأس انعکاسی a قدم‌های زیر را انجام بده:

(a) محاسبه‌ی $Sep(a)$

(b) محاسبه‌ی بخشی از V_a که به $Sep(a)$ محدود است

(c) محاسبه‌ی قیف اولیه و امتداد یال‌های قیف

(d) انجام جاروی شعاعی با شروع از نزدیک‌ترین پنجره‌ی جدا کننده به s و رسیدگی به رخدادهای طبق توضیحات ارائه شده.

قدم ۱ در زمان و حافظه‌ی $O(n)$ با الگوریتم گویباس و دیگران [۵۴] قابل انجام است. قدم ۲ شامل یافتن آخرین رأس روی $\pi(s, a)$ است که در زمان $O(\log n)$ با داشتن درخت کوتاه‌ترین مسیر قابل انجام است. محاسبه‌ی قابلیت دید در قدم ۲ با الگوریتم لی [۶۶، ۶۷] در زمان و حافظه‌ی خطی انجام می‌شود. محاسبه‌ی قیف و امتداد یال‌های آن در قدم ۲ به طور مستقیم از $SPM(s)$ در زمان $O(n)$ به دست می‌آید. قدم ۲ شامل محاسبه و رسیدگی به هر دو نوع رخداد است. تعداد کل رخدادهای $O(n)$ است و هر کدام در زمان ثابت قابل رسیدگی هستند.

بنابراین در کل زمان $O(n)$ صرف هر رأس انعکاسی می‌شود. اندازه‌ی افراز متناظر با هر یک از این رأس‌ها نیز $O(n)$ است. از آنجا که $O(n)$ رأس از این دست در P وجود دارد زمان کل پیش‌پردازش $O(n^2)$ است و حافظه‌ی مورد نیاز برای نگهداری افرازاها نیز $O(n^2)$ است.

هنگام دریافت یک پرس‌وجو نیز ابتدا پنجره‌ی $w = ab$ که s را از V_q جدا می‌کند را با روشی که گفته شد پیدا می‌کنیم. از آنجا که $SPM(s)$ در جریان پیش‌پردازش ساخته شده است، این زمان $O(\log n)$ خواهد بود. پس از آن، جستجوی دودویی پنجره در افراز متناظر با a به زمان $O(\log n)$ نیاز دارد و با به دست آوردن بازه‌ی شامل aw نقطه‌ی بهینه‌ی $c(w)$ در زمان ثابت محاسبه می‌شود. به این ترتیب نتیجه‌ی این بخش در قضیه‌ی زیر خلاصه می‌شود.

قضیه ۱.۴ یک چندضلعی ساده‌ی P و نقطه‌ی مبدأ داده‌شده‌ی s درون P را می‌توان در زمان $O(n^2)$

پیش‌پردازش کرد و داده‌ساختاری با اندازه‌ی $O(n^2)$ به دست آورد که بتوان با آن پرس‌وجویی به این صورت را در زمان $O(\log n)$ پاسخ داد: برای یک نقطه‌ی پرس‌وجوی q ، نقطه‌ی c را که کم‌ترین فاصله‌ی ژئودزیک از s داشته باشد و از q قابل دید باشد محاسبه کنید. کوتاه‌ترین مسیر از s به c نیز با پرداخت هزینه‌ای متناظر با تعداد یال‌های مسیر قابل محاسبه است.

۲.۴ کوتاه‌ترین مسیر با مقصد مشخص

در این بخش به بررسی مسئله در حالتی مقتیدتر می‌پردازیم به این صورت که مسیر پیدا شده در حالت قبل باید به مقصد مشخصی برگردد. فرض کنید P یک چندضلعی ساده با n رأس باشد و نقاط مبدأ و مقصد s و t در آن داده شده باشند. هدف مسئله پیش‌پردازش ورودی (P, s, t) است طوری که بتوان به پرس‌وجویی از این دست پاسخ داد: به ازای یک نقطه‌ی پرس‌وجوی q ، کوتاه‌ترین مسیری بین s و t را بیابید که q از حداقل یک نقطه از مسیر قابل دید باشد. طبق لم ۲.۳، این مسیر وارد V_q نمی‌شود و فقط در یک نقطه مانند c با آن اشتراک دارد. به چنین نقطه‌ای نقطه‌ی تماس بهینه می‌گوییم. همانند حالتی از مسئله که مقصد آن باز است، در این حالت نیز دو نوع پرس‌وجو تعریف می‌شود. هدف نوع اول یافتن طول کوتاه‌ترین مسیر یاد شده است و نوع دوم یافتن خود مسیر را نیز شامل می‌شود. پیش‌پردازش باید طوری انجام شود که پرس‌وجوی نوع اول در زمان $O(\log n)$ و نوع دوم در زمان $O(k + \log n)$ قابل پاسخ‌گویی باشند (k طول مسیر خروجی است).

بسته به موقعیت نقطه‌ی پرس‌وجوی q امکان وقوع حالت‌های زیر وجود دارد:

(۱) s یا t (یا هر دو) از q قابل دید هستند. در این حالت کوتاه‌ترین مسیر بین s و t جواب مسئله است.

(۲) s و t در دو جیب چندضلعی هستند (یعنی توسط دو پنجره‌ی متفاوت از V_q جدا شده‌اند). باز

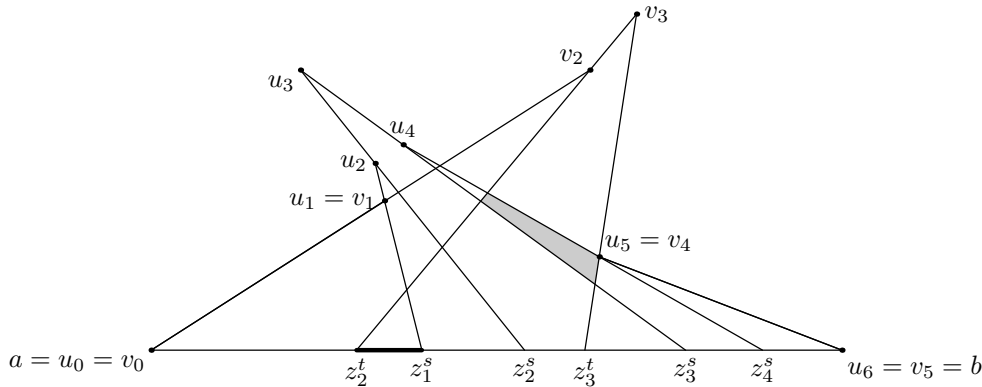
در این حالت هم کوتاه‌ترین مسیر بین این دو نقطه جواب مسئله است.

۳) s و t هر دو در یک جیب چندضلعی هستند که با w از V_q جدا می‌شود. در این حالت باید به دنبال نقطه‌ی تماس بهینه‌ی $c(w)$ بگردیم و جواب مسئله از الصاق کوتاه‌ترین مسیرهای بین s و $c(w)$ و $c(w)$ به t به دست می‌آید.

تشخیص دادن دو حالت اول کار ساده‌ای است. با فرض این که در مرحله‌ی پیش‌پردازش، $SPM(s)$ و $SPM(t)$ محاسبه شده‌اند و برای جایابی نقطه پیش‌پردازش شده‌اند، می‌توان با جایابی q در این دو نقشه اولاً فهمید که آیا q از s یا t قابل دید است یا نه و در صورت قابل دید نبودن، پنجره‌ای را که هر یک از این دو نقطه را از V_q جدا می‌کند به دست آورد. یادآوری می‌کنیم که این دو پنجره‌ها توسط آخرین رأس‌ها روی $\pi(s, q)$ و $\pi(t, q)$ مشخص می‌شوند. در صورتی که این دو پنجره متفاوت باشند، با برگرداندن کوتاه‌ترین مسیر بین s و t که از نقشه‌های آنها قابل استخراج است جواب مسئله را یافته‌ایم. توجه داشته باشید که تمام اعمال بالا (با فرض داشتن ساختارهای مناسب که در مرحله‌ی پیش‌پردازش ساخته شده‌اند)، در زمان $O(\log n)$ قابل انجام هستند. به این ترتیب، هدف ما یافتن جواب مسئله در حالت سوم است که پنجره‌ی w هر دو نقطه را از V_q جدا می‌کند.

۱.۲.۴ تعریف‌ها و مفاهیم پایه‌ای

نظیر تعریف‌ها و مفاهیم استفاده شده در حالت قبلی مسئله، اینجا نیز بر ساختار قیف متکی هستیم. فرض کنید a و b دو نقطه‌ی انتهای w باشند. نمادهای $F_s(w)$ و $F_t(w)$ را برای قیف‌های ساخته شده توسط الگوریتم کوتاه‌ترین مسیر گویاس و دیگران به ترتیب با در نظر گرفتن s و t به عنوان مبدأ استفاده می‌کنیم. دنباله رأس‌های روی این دو قیف را به ترتیب $a = u_0, u_1, \dots, u_{m_s}, u_{m_s+1} = b$ و $a = v_0, v_1, \dots, v_{m_t}, v_{m_t+1} = b$ همچنین نقاط تقاطع امتداد یال‌های قیف $F_s(w)$



شکل ۷: دو قیف متناظر با s و t روی پنجره‌ی ab . پاره‌خط $z_1^s z_4^s$ ، بازه‌ی بهینگی I_{13} است. بخش خاکستری $Cell_{43}$ است.

مانند $(z_{m_s+1}^s = b$ و $z_0^s = a$) در نتیجه می‌دهیم z_i^s را با w با $u_i u_{i+1}$ ($0 \leq i \leq m_s$) صورت، $(z_j^t$ قطع می‌کند (شکل ۷).

در استفاده از تمام نمادهای تعریف شده در این بخش که پارامتری برای مشخص کردن پنجره دارند، هر جا که ابهامی ایجاد نکند این پارامتر را حذف می‌کنیم و فرض می‌کنیم پنجره‌ی $w = ab$ مورد نظر است.

برای یک رأس x در یکی از قیف‌ها، نماد $Cell(x)$ را به عنوان ناحیه‌ی مثلثی شکل متناظر با x داخل قیف مربوط استفاده می‌کنیم. به عبارت دقیق‌تر، برای رأس u_i از قیف F_s داریم $Cell(u_i) = \Delta z_{i-1}^s u_i z_i^s$ و برای رأس v_j از F_t ، $Cell(v_j) = \Delta z_{j-1}^t v_j z_j^t$. تعریف می‌کنیم $Cell_{xy} = Cell_x \cap Cell_y$. به راحتی می‌توان دید $Cell_{xy}$ توسط سه یا چهار پاره‌خط محدود می‌شود. همچنین می‌توان حالاتی ایجاد کرد که تعداد این ناحیه‌ها $\Theta(n^2)$ باشد. در جایی که ابهامی پیش نیاید، از نماد $Cell_{ij}$ به عنوان خلاصه به جای $Cell_{u_i v_j}$ استفاده می‌کنیم (شکل ۷).

با داشتن دو قیف ساخته شده روی w ، همانند قبل با استفاده از مفهوم بازه‌های بهینگی می‌توان نقطه‌ی تماس بهینه را به دست آورد. برای تعریف دقیق‌تر بازه‌های بهینگی در این حالت فرض کنید u_i و v_j به ترتیب دو رأس از F_s و F_t باشند. مجموعه نقاطی روی w که آخرین رأس روی کوتاه‌ترین مسیر از s به آنها u_i است پاره‌خط $z_{i-1}^s z_j^s$ و آنهایی که آخرین رأس روی کوتاه‌ترین

مسیر از t به آنها v_j است پاره خط $z_{j-1}^t z_j^t$ را تشکیل می دهند. این دو پاره خط یا از هم جدا هستند یا این که اشتراک آنها به شکل یک پاره خط است که آن را $I_{ij}(w)$ می نامیم. مجموعه‌ی تمام پاره خط‌های I_{ij} ، بازه‌های بهینگی $L_{s,t}$ را روی w می سازند. اگر این بازه‌ها را به ترتیب صعودی فاصله از a مرتب کنیم، یک دنباله به دست می آید که آن را با \mathcal{I} نشان می دهیم.

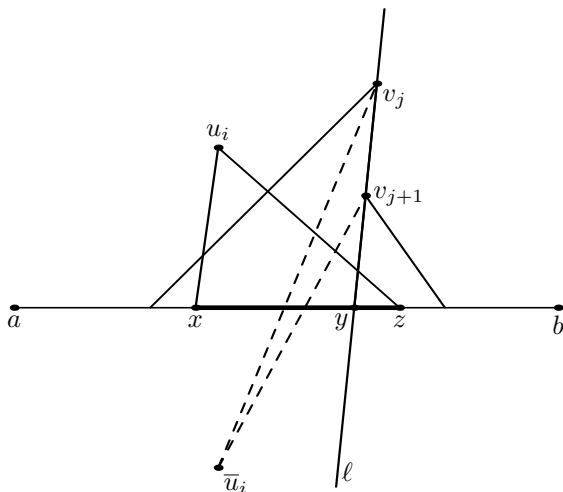
با به دست آوردن بازه‌های بهینگی، می توان مطابق لم ۱.۳ عمل کرد. به این ترتیب که با داشتن یک پنجره‌ی w و دو رأس u_i و v_j ، $Ref_{ij}(w)$ را به عنوان نقطه‌ی تقاطع پاره خط $u_i \bar{v}_j$ و خط شامل w تعریف می کنیم که \bar{v}_j قرینه‌ی v_j نسبت به خط یاد شده است. $c_{ij}(w)$ نقطه‌ای از I_{ij} است که مجموع کوتاه‌ترین فاصله‌های آن تا s و t کمینه باشد. به این نقطه، نقطه‌ی بهینه‌ی بازه‌ی I_{ij} می گوئیم. بر اساس موقعیت Ref_{ij} ، یکی از سه حالت زیر رخ می دهد:

- Ref_{ij} درون I_{ij} است. در این حالت c_{ij} همان Ref_{ij} است. به چنین بازه‌ی c -بهینه می گوئیم.
- Ref_{ij} از تمام نقاط I_{ij} به a نزدیک تر است. در این حالت c_{ij} نقطه‌ی انته‌ای I_{ij} است که به a نزدیک تر است. به چنین بازه‌ی a -بهینه می گوئیم.
- Ref_{ij} از تمام نقاط I_{ij} از a دورتر است. در این حالت c_{ij} نقطه‌ی انته‌ای I_{ij} است که از a دورتر است. به چنین بازه‌ی b -بهینه می گوئیم.

بین مجموعه نقاط $\{c_{ij}(w) \mid I_{ij}(w) \in L_{s,t}\}$ یکی کم ترین مجموع فاصله‌ها تا s و t را دارند که همان نقطه‌ی تماس بهینه است. برای شناسایی بازه‌ای که شامل این نقطه است از لم ۱.۴ استفاده می کنیم.

لم ۱.۴ اگر I_{ij} یک بازه‌ی c -بهینه باشد، هر بازه‌ی بعد از آن در \mathcal{I} a -بهینه و هر بازه‌ی قبل از آن در \mathcal{I} b -بهینه است.

اثبات. فرض کنید I_{ij} آخرین بازه در \mathcal{I} نباشد و بازه‌ی بلافاصله بعد از آن در \mathcal{I} را در نظر بگیرد. این بازه یا $I_{i,j+1}$ است یا $I_{i+1,j}$. بدون از دست دادن کلیت فرض کنید $I_{i,j+1}$ باشد. (حالت دیگر



شکل ۸: اثبات لم ۱.۴ - پاره خط xy بازه I_{ij} را مشخص می‌کند که یک بازه c -بهینه است. پاره خط yz که $I_{i,j+1}$ (بازه بلافاصله بعدی I_{ij}) است، یک بازه a -بهینه است.

نیز استدلال مشابهی خواهد داشت. فرض کنید \bar{u}_i قرینه u_i نسبت به w و ℓ خط دربرگیرنده v_j و v_{j+1} باشند (شکل ۸). دو نیم‌صفحه‌ی ایجاد شده توسط ℓ را که شامل a است H_a و دیگری را H_b بنامید. به سادگی می‌توان دید $\bar{u}_i \in H_a$ (چون طبق فرض، $c_{ij} \in I_{ij}$ در H_a است). از این موضوع و این که v_{j+1} روی ℓ است نتیجه می‌گیریم $Ref_{i,j+1}$ در H_a قرار دارد، پس $c_{i,j+1}$ نقطه‌ی انتهایی $I_{i,j+1}$ است که به a نزدیک‌تر است. بنابراین بازه بلافاصله بعدی I_{ij} ، a -بهینه است. حال اگر یک بازه a -بهینه را در نظر بگیریم، با استدلالی شبیه به آنچه گفته شد نتیجه می‌گیریم بازه بلافاصله بعدی آن نیز a -بهینه است. به این ترتیب به کمک استقراء می‌توان نشان داد تمام بازه‌های بعدی I_{ij} ، a -بهینه هستند. استدلالی کاملاً مشابه منجر به اثبات b -بهینه بودن بازه‌های قبلی I_{ij} می‌شود. \square

با در نظر گرفتن نتیجه‌ی لم بالا، می‌توان نتیجه گرفت حالت‌های ممکن برای دنباله‌ی \mathcal{I} به

این صورت است:

- تمام بازه‌ها a -بهینه هستند. در این صورت نقطه‌ی تماس بهینه، a است.
- تمام بازه‌ها b -بهینه هستند. در این صورت نقطه‌ی تماس بهینه، b است.

• بخشی از ابتدای دنباله b -بهینه، یک بازه‌ی c -بهینه و بقیه‌ی دنباله a -بهینه هستند. هر یک از دو بخش ابتدا یا انتها می‌توانند تهی باشند، اما مستقل از این، نقطه‌ی بهینه‌ی بازه‌ی c -بهینه نقطه‌ی تماس بهینه است.

• بخشی از ابتدای دنباله b -بهینه و بقیه‌ی آن a -بهینه هستند. این حالت خاصی است که نقطه‌ی بهینه‌ی هر دو بازه‌ی b -بهینه و a -بهینه‌ی مجاور بر هم منطبق و روی نقطه‌ی مرزی مشترک این دو بازه قرار می‌گیرد.

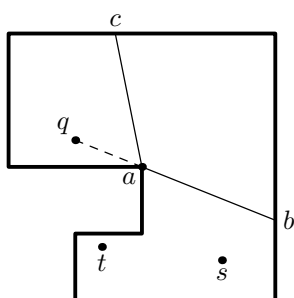
با مشاهدات بالا می‌توان لم ۲.۴ را نتیجه گرفت:

لم ۲.۴ با داشتن دنباله مرتب بازه‌های بهینگی متناظر با جفت s و t ، نقطه‌ی تماس بهینه در زمان $O(\log n)$ قابل محاسبه است.

اثبات. درستی این لم از نتایج قبلی و با انجام یک الگوریتم جستجوی دودویی روی دنباله نتیجه می‌شود. \square

۲.۲.۴ الگوریتم پیش پردازش

در این حالت از مسئله، مانند حالتی که مسئله با مقصد باز مطرح می‌شود، توسط یک جاروی شعاعی پیرامون رأس‌های انعکاسی P ، مجموعه پنجره‌های جدا کننده را افزای می‌کنیم. در اینجا به خاطر وجود مقصد ثابت t ، لازم است مفهوم پنجره‌ی جدا کننده را دوباره تعریف کنیم. به یک پاره‌خط پنجره‌ی جدا کننده می‌گوییم اگر پنجره‌ی چندضلعی قابلیت دید نقطه‌ی دلخواهی مانند q باشد که هر دو نقطه‌ی s و t را از V_q جدا کند (شکل ۹). به سادگی می‌توان مشاهده کرد که مجموعه پنجره‌های جدا کننده‌ی s و t از نقاط پرس و جوی احتمالی q ، $Sep(a) = Sep_s(a) \cap Sep_t(a)$ است. از آنجا که $Sep_s(a)$ و $Sep_t(a)$ هر کدام توسط یک بازه‌ی زاویه‌ای مشخص می‌شوند، $Sep(a)$ نیز چنین خاصیتی خواهد داشت.



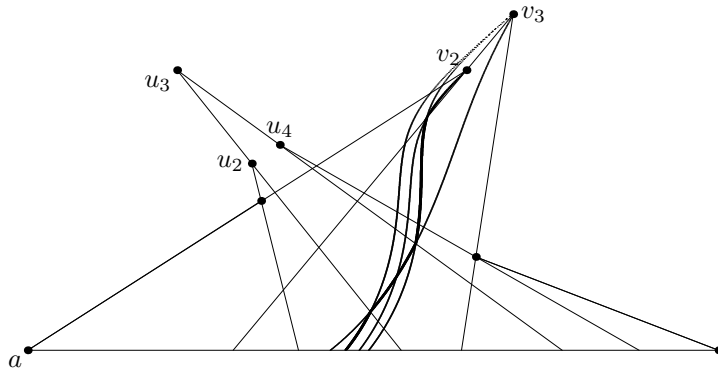
شکل ۹: پاره خط ab یک پنجره‌ی جداکننده متناظر با s و t است ولی ac چنین خاصیتی ندارد.

هدف مرحله‌ی پیش‌پردازش، افزایش کردن $Sep(a)$ به تعدادی مجموعه است طوری که برای تمام پنجره‌های یک مجموعه (مثل w)، بازه‌ای که شامل نقطه‌ی تماس بهینه یا $c(w)$ شود یکسان باشد. در این صورت، آخرین رأس $\pi(s, c(w))$ و $\pi(t, c(w))$ یکسان خواهد بود. به این ترتیب با دریافت نقطه‌ی پرس‌وجوی q ، کافی است پنجره‌ی جداکننده‌ی s و t از V_q را محاسبه کنیم و مجموعه‌ی شامل آن را در افزایش به دست آمده در پیش‌پردازش بیابیم. با مشخص شدن مجموعه‌ی یاد شده، نقطه‌ی تماس بهینه در زمان ثابت به دست خواهد آمد.

همانند حالت قبل مسئله، پیش‌پردازش توسط جاروی شعاعی پیرامون رأس‌های انعکاسی P انجام می‌شود و دو نوع رخداد برای آن تعریف می‌شود: رخداد بازه که در آن بازه‌ی شامل نقطه‌ی تماس بهینه عوض می‌شود و رخداد قیف که در آن ساختار قیف تغییر می‌کند. در ادامه این دو رخداد را بررسی خواهیم کرد.

رخدادهای بازه

برای پردازش و رسیدگی به رخدادهای بازه، نیاز داریم رفتار نقطه‌ی تماس بهینه را هنگامی که $Sep(a)$ را جارو می‌کنیم بررسی کنیم. فرض کنید $Sep(a)$ توسط بازه‌ی زاویه‌ای $[\alpha, \alpha']$ تعریف شود و به ازای $\alpha \leq \varphi \leq \alpha'$ ، w_φ پنجره‌ای با نقطه‌ی انتهای a و زاویه‌ی φ باشد. رأس‌های u_i و v_j را به ترتیب از قیف‌های $F_s(w_\alpha)$ و $F_t(w_\alpha)$ در نظر بگیرید و از بین دو زاویه‌ای که av_j و aw_φ



شکل ۱۰: خم C^* که پررنگ تر نشان داده شده از بخش‌هایی از خم‌های C_{ij} تشکیل شده. در این شکل، خم‌هایی که در ساختن C^* شرکت می‌کنند به ترتیب عبارتند از: $C_{۲۳}$, $C_{۳۳}$, $C_{۳۲}$ و $C_{۴۲}$.

می‌سازند، کوچکترین β بنامیم. C_{ij} را به عنوان مجموعه نقاط $\{Ref_{ij}(w_\varphi) \mid \alpha \leq \varphi \leq \beta\}$ تعریف کنیم. این مجموعه، یک خم پیوسته‌ی ساده را می‌سازد که یک انتهای آن نقطه‌ای از بین u_i و v_j است که زاویه‌ی β را ساخته است.

مجموعه نقاط تماس بهینه روی پنجره‌های $Sep(a)$ را C^* می‌نامیم. به عبارت دقیق‌تر، $C^* = \{c(w_\varphi) \mid \alpha \leq \varphi \leq \alpha'\}$ از روی لم ۱.۴ نتیجه می‌گیریم که بخش‌هایی از C_{ij} که داخل $Cell_{ij}$ باشند، بخش‌هایی از C^* هستند (شکل ۱۰). لم ۳.۴ ساختار C^* را مشخص می‌کند.

لم ۳.۴ C^* یک خم پیوسته‌ی ساده است و از الصاق بخش‌هایی از خم‌های C_{ij} و پاره‌خط‌هایی که بخش‌هایی از یال‌های P هستند تشکیل می‌شود.

اثبات. از آنجا که C^* در نتیجه‌ی یک جاروی شعاعی حاصل می‌شود، نمی‌تواند خود را قطع کند. فرض کنید در لحظه‌ای از جارو، نقطه‌ی تماس بهینه در $Cell_{ij}$ باشد. خم C_{ij} را در جهت جارو دنبال می‌کنیم تا برای اولین بار به مرز $Cell_{ij}$ برسیم. بدون از دست دادن کلیت، فرض کنید این اتفاق در نقطه‌ی x که روی مرز مشترک با سلول $Cell_{i,j+1}$ قرار دارد رخ دهد. به راحتی می‌توان بررسی کرد که دو خم C_{ij} و $C_{i,j+1}$ در نقطه‌ی x یکدیگر را قطع می‌کنند. این خاصیت برای تمام همسایه‌های این سلول برقرار است. به این ترتیب، اگر از $c(w_\alpha)$ شروع کنیم و خم‌های C_{ij} را دنبال

کنیم و در مرز بین سلول‌ها، خمی را که دنبال می‌کنیم نیز تغییر دهیم، در نهایت C^* را به دست می‌آوریم. تنها مشکل مواقعی است که یکی از خم‌ها از چندضلعی P خارج می‌شود. در این بخش‌ها، خمی که دنبال می‌کنیم با بخش مربوط از مرز P جایگزین می‌شود. به این ترتیب پیوستگی C^* نیز از پیوستگی خم‌های C_{ij} و مرز P نتیجه می‌شود. \square

برای محاسبه‌ی بازه‌های رخداده، از بازه‌ی اولیه‌ای که شامل $c(w_\alpha)$ می‌شود شروع می‌کنیم. هنگامی که C^* وارد یک سلول $Cell_{ij}$ می‌شود، یالی از این سلول را که خم C_{ij} از آن خارج می‌شود محاسبه می‌کنیم تا رخداده بعدی را که در آن C^* از آن یال وارد یک سلول همسایه می‌شود تعیین کند. از آنجا که هر سلول حداکثر چهار یال دارد، این محاسبه در زمان ثابت انجام می‌شود.

برای به دست آوردن کران تعداد رخدادهای بازه باید تعداد دفعاتی که C^* امتداد یال‌های قیف‌ها را قطع می‌کند بشماریم. از آنجا که C^* از قطعات خم‌های C_{ij} تشکیل شده است، این تعداد را با بررسی این خم‌ها به دست می‌آوریم. همان‌گونه که گفته شد، یک رخداده بازه زمانی اتفاق می‌افتد که یک خم C_{ij} مرز سلول $Cell_{ij}$ را قطع می‌کند. در این حالت تقاطعی بین خم یاد شده و امتداد یکی از یال‌های $u_{i-1}u_i$ ، $u_i u_{i+1}$ ، $v_{j-1}v_j$ و $v_j v_{j+1}$ رخ می‌دهد. بنابراین هدف ما یافتن تعداد دفعاتی است که یک خم C_{ij} یک خط را که از u_i یا v_j می‌گذرد می‌تواند قطع کند. برای این کار می‌توان از روش‌های جبری استفاده کرد، اما در اینجا از تعبیری هندسی استفاده می‌کنیم که در اثبات بعضی خواص دیگر نیز مفید خواهد بود.

فرض کنید پنجره‌ای که در ابتدا جارو روی آن شروع می‌شود $w = ab$ است که a یک رأس انعکاسی P است. موقعیت نقاط را در مختصات قطبی به مرکز a می‌سنجیم و بدون از دست دادن کلیت فرض می‌کنیم زاویه‌ی قطبی b صفر است (در غیر این صورت از یک تبدیل دوران استفاده می‌کنیم). قیف F_s را در نظر بگیرید و فرض کنید زاویه‌ی قطبی u_i برابر θ_i است. هنگامی که پنجره‌ی جارو حول a دوران می‌کند، روی کمان A_i از دایره‌ی $Circ_i$ که به مرکز a است و از u_i

می‌گذرد حرکت می‌کند. این کمان از زاویه‌ی θ_i شروع می‌کند و اندازه‌ی آن دو برابر کل بازه‌ی زاویه‌ای جارو شده است. با استفاده از این مدل و نکات یاد شده در بالا، لم ۴.۴ برای محاسبه‌ی تعداد رخدادهای به کار می‌آید.

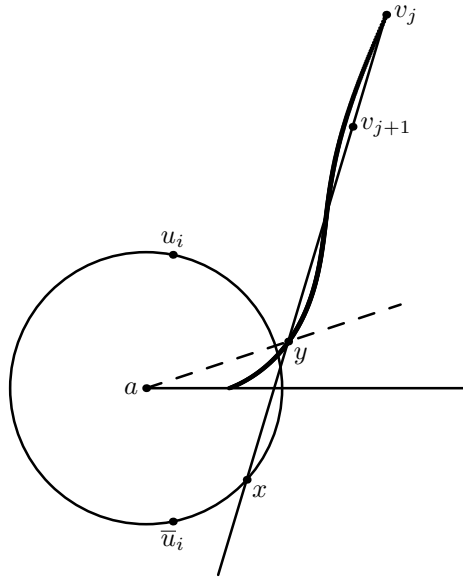
لم ۴.۴ یک خم C_{ij} یک خط دلخواه را که از u_i یا v_j می‌گذرد حداکثر در دو نقطه غیر از u_i و v_j قطع می‌کند.

اثبات. فرض کنید در زاویه‌ی جاروی φ یک تقاطع بین C_{ij} و خط l که از v_j می‌گذرد رخ دهد. نقطه‌ی C_{ij} در این زاویه، از وصل کردن قرینه‌ی u_i نسبت به $w_\varphi(\bar{u}_i)$ و v_j به دست می‌آید. چون در این زاویه تقاطع رخ داده است، این نقطه روی l است. چون l از v_j می‌گذرد، \bar{u}_i نیز باید روی l باشد. در این حالت \bar{u}_i در تقاطع l و کمان A_i قرار دارد. چون خط و کمان یاد شده در حداکثر دو نقطه یکدیگر را قطع می‌کنند، C_{ij} و l نیز حداکثر دو نقطه‌ی تقاطع خواهند داشت (شکل ۱۱). برای حالتی که l از u_i بگذرد نیز استدلال مشابهی همین نتیجه را می‌دهد. \square

از روی لم ۴.۴، مستقیماً نتیجه می‌شود که تعداد رخدادهای بازه $O(n^2)$ است. لم ۵.۴ نشان می‌دهد که حد پایین این تعداد نیز از مرتبه‌ی $\Omega(n^2)$ است.

لم ۵.۴ می‌توان قیف‌های F_s و F_t را طوری ایجاد کرد که تعداد $\Theta(n^2)$ رخداد بازه وجود داشته باشد.

اثبات. ابتدا نشان می‌دهیم که مکان رأس‌های $U = (u_1, u_2, \dots, u_k)$ از قیف F_s را چگونه انتخاب کنیم تا خم بهینه امتداد یک یال از F_t (مانند $v_j z_j^t$) را در $\Theta(k)$ نقطه قطع کند. برای شروع از یک حالت ساده، فرض کنید پاره‌خط $v_j z_j^t$ بر موقعیت اولیه‌ی w عمود باشد و l خطی باشد که این پاره‌خط را دربر دارد. توجه کنید که موقعیت اولیه‌ی w منطبق بر محور x فرض شده است (a مبدأ

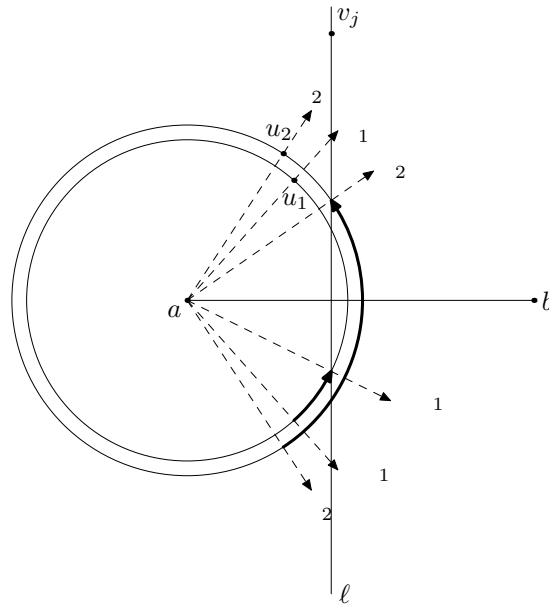


شکل ۱۱: تقاطع خم C_{ij} و امتداد یال $v_j v_{j+1}$ حداکثر در دو نقطه اتفاق می‌افتد که متناظر با تقاطع‌های امتداد مذکور و دایره‌ای است که از u_i می‌گذرد. در اینجا یکی از این تقاطع‌ها بین خم و امتداد یال یاد شده در نقطه‌ی y رخ داده است که متناظر با زمانی است که قرینه‌ی u_i نسبت به پنجره‌ی جارو (خط‌چین) بر یک تقاطع امتداد $v_j v_{j+1}$ و دایره‌ی $Circ_i$ (x) منطبق شده است.

مختصات است و b زاویه‌ی قطبی صفر دارد). همچنین هر دو مؤلفه‌ی x و y نقطه‌ی v_j مثبت باشند (شکل ۱۲).

فرض کنید چنین مکانی برای رأس‌های U یافته‌ایم و u_i در موقعیت (r_i, θ_i) قرار دارد. بنابراین خم‌های $C_{1j}, C_{2j}, \dots, C_{kj}$ خط l را در دنباله‌ای از نقاط که به ترتیب کاهشی فاصله تا v_j مرتب هستند قطع می‌کند (چون این نقاط در نتیجه‌ی جاروی شعاعی به دست آمده‌اند). برای مشخص کردن موقعیت نقاط تقاطع مجدداً از تعبیر هندسی که در بالا ارائه شد استفاده می‌کنیم. بر اساس لم ۴.۴، تقاطع‌های بین خط l و خم C_{ij} متناظر با نقاط تقاطع بین l و A_i هستند. یک تقاطع بین l و $Circ_i$ را یک تقاطع پایین می‌نامیم اگر مختصه‌ی y آن منفی باشد، در غیر این صورت آن را یک تقاطع بالا می‌نامیم. فرض کنید γ_i زاویه‌ی قطبی تقاطع بالای l و $Circ_i$ باشد. تقاطع پایین این دو در زاویه‌ی قطبی $-\gamma_i$ رخ می‌دهد.

برای این که خم بهینه پیوسته باشد، نقاط تقاطع این خم با l باید به طور متناوب تقاطع‌های



شکل ۱۲: ساختار ایجاد شده برای دو نقطه‌ی اول در U - زاویه‌هایی که توسط کمان‌های جهت‌دار ضخیم نشان داده شده‌اند، مقدار جاروی لازم برای رخدادهای بازه‌ی اول و دوم هستند.

بالا و پایین باشند، به این علت که در یک تقاطع پایین، خم بهینه از سمت چپ l به سمت راست آن می‌رود و در تقاطع‌های بالا، عکس این اتفاق می‌افتد. با فرض این که اولین تقاطع پایین باشد، این تقاطع موقعی رخ می‌دهد که زاویه‌ی قطبی \bar{u}_1 برابر $-\gamma_1$ است. این اتفاق در زاویه‌ی جاروی $\theta_1 - \gamma_1$ صورت می‌گیرد. دنباله‌ی تقاطع‌ها متناظر خواهد بود با قرار گرفتن \bar{u}_2 در زاویه‌ی γ_2 و قرار گرفتن \bar{u}_3 در زاویه‌ی $-\gamma_3$ و در ادامه به همین ترتیب. به این ترتیب، دنباله زاویه‌های جارو در نقاط تقاطع به صورت $(\theta_1 - \gamma_1, \theta_2 + \gamma_2, \theta_3 - \gamma_3, \dots)$ است. چون این دنباله صعودی است، داریم:

$$\theta_2 - \theta_1 > -(\gamma_1 + \gamma_2)$$

$$\theta_3 - \theta_2 > \gamma_2 + \gamma_3$$

$$\theta_4 - \theta_3 > -(\gamma_3 + \gamma_4)$$

⋮

با این فرض که d فاصله‌ی a از خط l باشد، r_1 را برابر $d + \varepsilon_r$ قرار می‌دهیم که ε_r یک عدد

مثبت کوچک است (ما بعداً در مورد این که این عدد چقدر کوچک باشد توضیح خواهیم داد) و برای $1 < i \leq k$ قرار می‌دهیم $x_i = r_{i-1} + \varepsilon_r$. به همین ترتیب تعریف می‌کنیم $\theta_1 = \varepsilon_\theta$ که ε_θ نیز یک عدد مثبت کوچک است. برای محاسبه‌ی θ_i از روی θ_{i-1} با دو نوع محدودیت مواجه می‌شویم: یکی که توسط ساختار قیف روی U اعمال می‌شود (محدب بودن از بیرون دیواره‌های قیف) و دیگری که از نامساوی‌های بالا ناشی می‌شوند. کمترین مقدار مورد نیاز برای ارضای هر دو محدودیت را به θ_{i-1} اضافه می‌کنیم تا θ_i به دست آید. یک مشکل هنگام محاسبه‌ی θ_i برای مقادیر فرد i رخ می‌دهد. برای این حالات، برای ارضای نامساوی‌های یاد شده، ناچار به اضافه کردن مقداری به θ_{i-1} هستیم که از $\gamma_{i-1} + \gamma_i$ بزرگ‌تر باشد. چون دنباله‌ی $(\gamma_1, \gamma_2, \dots)$ یک دنباله‌ی افزایشی است، مقدار $\gamma_{i-1} + \gamma_i$ با زیاد شدن i افزایش می‌یابد. این موضوع ممکن است حالتی را پیش بیاورد که در آن مختصه‌ی y از u_i از u_{i-1} کمتر شود که موجب این می‌شود که دنباله‌ی U بر ساختار قیف منطبق نباشد. برای این مشکل باید مقادیر ε_θ و ε_r را به اندازه‌ی کافی کوچک تعریف کنیم (برحسب تعداد نقاط دنباله‌ی U).

دنباله‌ی U که به روش بالا محاسبه شد، یک دیواره از قیف F_s را می‌سازد. دیواره‌ی دیگر را می‌توان به سادگی با متصل کردن u_k به b ایجاد کرد. برای ساختن قیف F_t ، دو رأس v_1 و v_2 را با مختصه‌ی x یکسان طوری قرار می‌دهیم که ساختن دنباله رأس‌های U مطابق روشی که گفته شد و با فرض این که l خطی است که از v_1 و v_2 می‌گذرد، ممکن باشد. رأس‌های v_1, v_2, \dots, v_l را به سمت x ‌های مثبت صفحه طوری قرار می‌دهیم که امتداد پاره‌خط‌های $v_j v_{j+1}$ به اندازه‌ی کافی به l نزدیک باشند طوری که خم‌های $c_{1j}, c_{2j}, \dots, c_{kj}$ این امتدادها را در دنباله‌ای متناوب از تقاطع‌های بالا و پایین قطع کند.

به این ترتیب، خم‌هایی که در ساختن خم بهینه‌ی C^* شرکت می‌کنند، به ترتیب جاروی شعاعی دنباله‌ی $(c_{11}, c_{12}, \dots, c_{1l}, c_{21}, \dots, c_{kl})$ را می‌سازند. به این ترتیب با انتخاب k و l از مرتبه‌ی n ، می‌توان حالت‌هایی ایجاد کرد که تعداد رخداد‌های بازه برای یک رأس انعکاسی $\Theta(n^2)$

باشد.

□

با روشی مشابه اثبات لم ۵.۴ می توان نشان داد که چندضلعی هایی وجود دارند با $\Theta(n)$ رأس انعکاسی که هر یک از این رأس ها $\Theta(n^2)$ رخداد بازه دارند. با این وجود، در بسیاری حالت های عملی تعداد رخداد های بازه به ازای هر رأس انعکاسی کم تر از این است. به همین خاطر پارامتر جدیدی به نام K معرفی می کنیم که حداکثر تعداد رخداد های بازه به ازای یک رأس انعکاسی در چندضلعی P است و زمان اجرای الگوریتم را بر اساس این پارامتر بیان می کنیم. انتظار داریم که برای دسته ی زیادی از چندضلعی ها این پارامتر $O(n)$ باشد.

رخداد های قیف

رخداد های قیف در این حالت، بسیار شبیه به این رخداد ها در مدل مقصد باز مسئله هستند، چون تغییر ساختار هر یک از دو قیف F_t و F_s مستقل از یکدیگر اتفاق می افتد. به این علت می توان رخداد های قیف را برای هر یک از این دو قیف به طور مستقل و طبق همان روش حالت مقصد باز محاسبه کرد و دو مجموعه رخداد را با یکدیگر ادغام نمود. به این ترتیب تعداد کل رخداد های قیف $O(n)$ خواهد بود.

۳.۲.۴ خلاصه و تحلیل الگوریتم

قدم هایی که در مرحله ی پیش پردازش برداشته می شود به صورت زیر هستند:

(۱) $SPM(s)$ ، $SPM(t)$ ، $SPT(s)$ و $SPT(t)$ را محاسبه کن.

(۲) برای هر رأس انعکاسی a قدم های زیر را طی کن:

(a) محاسبه ی $Sep(a)$

(b) محاسبه ی بخشی از V_a که به $Sep(a)$ محدود است

(c) محاسبه‌ی وضعیت اولیه‌ی قیف‌های F_t و F_s و امتداد یال‌های آنها

(d) انجام جاروی شعاعی حول a با شروع از نزدیک‌ترین پنجره‌ی جداکننده به s و t

قدم ۱ با استفاده از الگوریتم [۵۴] در زمان و حافظه‌ی $O(n)$ قابل انجام است. قدم‌های a^2 و b^2 و c^2 نیز در زمان و حافظه‌ی $O(n)$ طبق همان روش‌های استفاده شده در حالت مقصد باز انجام می‌شوند. تعداد رخدادهای جاروی قدم d^2 برابر $O(n+K)$ است که برای هر یک زمان ثابتی صرف می‌شود. بنابراین به ازای هر رأس انعکاسی زمان $O(n+K)$ صرف می‌شود تا داده‌ساختاری با همین اندازه ساخته شود. از آنجا که تعداد رأس‌های انعکاسی از مرتبه‌ی $O(n)$ است، زمان کل پیش‌پردازش $O(n^2 + nK)$ و اندازه‌ی داده‌ساختار نهایی نیز $O(n^2 + nK)$ است.

پردازش یک پرس‌وجو نیز بسیار شبیه به حالت مقصد باز مسئله است، به این ترتیب که با دریافت نقطه‌ی پرس‌وجوی q ، ابتدا قابل دید بودن s و t از q کنترل می‌شود و پنجره‌ی w که این دو نقطه را از V_q جدا می‌کند محاسبه می‌شود. تمام این قدم‌ها با جایابی q در نقشه‌های کوتاه‌ترین مسیر s و t قابل انجام هستند. پس از آن، با یافتن بازه‌ی زاویه‌ای شامل w در بازه‌های حول رأس انعکاسی انتهایی w ، آخرین رأس‌های روی $\pi(s, c(w))$ و $\pi(t, c(w))$ در نتیجه نقطه‌ی تماس بهینه در زمان ثابت به دست می‌آیند.

به این ترتیب نتیجه‌ی حالت مقصد مشخص مسئله در قضیه‌ی زیر خلاصه می‌شود.

قضیه ۲.۴ یک چندضلعی ساده‌ی P و دو نقطه‌ی s و t درون آن را می‌توان در زمان و حافظه‌ی $O(n^2 + nK)$ پیش‌پردازش کرد طوری که بتوان طول کوتاه‌ترین مسیر بین s و t که حداقل یک نقطه از آن از نقطه‌ی پرس‌وجوی q قابل دید باشد را در زمان $O(\log n)$ پیدا کرد. خود مسیر نیز با پرداخت هزینه‌ای متناسب با تعداد یال‌های آن قابل محاسبه است.

کوتاه‌ترین مسیرها با قید دیدار چندضلعی

در این فصل به بررسی قیدی روی کوتاه‌ترین مسیرها می‌پردازیم که تعمیمی از قید قابلیت دید است که در فصل ۳ بررسی شد. هدف مسئله‌ی کوتاه‌ترین مسیر با قید قابلیت دید، یافتن کوتاه‌ترین مسیر بین دو نقطه‌ی داده شده‌است طوری که اشتراک آن با چندضلعی قابلیت دید یک نقطه ناتهی باشد. اگر این مسئله را طوری تعمیم دهیم که به جای چندضلعی قابلیت دید، یک چندضلعی کلی داده شده قرار دهیم به مسئله‌ی کوتاه‌ترین مسیر با قید دیدار چندضلعی خواهیم رسید. هدف این مسئله یافتن کوتاه‌ترین مسیر بین دو نقطه‌ی داده شده است طوری که اشتراک مسیر با یک چندضلعی کلی داده شده ناتهی باشد. این مسئله در دامنه‌های دو بعدی تعریف می‌شود: چندضلعی‌های ساده و دامنه‌های چندضلعی‌گونه. در ادامه خواهیم دید برای هر یک از این دو دامنه چگونه می‌توان مسئله‌ی مورد بحث را در زمان بهینه حل کرد.

۱.۵ چندضلعی‌های ساده

فرض کنید چندضلعی ساده‌ی P به همراه چندضلعی ساده‌ی T درون آن داده شده باشند و مجموع تعداد رأس‌های P و T برابر n باشد. به T ، چندضلعی هدف می‌گوییم. به یک مسیر در P که با T

اشتراک ناتهی داشته باشد، یک مسیر دیدارکننده T می‌گوییم. هدف مسئله یافتن کوتاه‌ترین مسیر دیدارکننده T بین دو نقطه‌ی داده شده‌ی s و t درون P است. هدف دیگر ما، ساختن زیرتقسیمی از P است که نقشی شبیه به نقشه‌ی کوتاه‌ترین مسیر را ایفا کند، به این صورت‌که با داشتن نقطه‌ی مبدأ s طوری ورودی (P, T, s) را پیش‌پردازش کنیم که با دریافت نقطه‌ی مقصد t در یک پرس‌وجو، کوتاه‌ترین مسیر دیدارکننده‌ی T را در زمان $O(\log n)$ پیدا کنیم.

۱.۱.۵ محاسبه‌ی کوتاه‌ترین مسیر مقید

لم ۱.۵ یک خاصیت اساسی کوتاه‌ترین مسیر دیدارکننده‌ی T را بیان می‌کند که محاسبه‌ی سریع این مسیرها را ممکن می‌کند.

لم ۱.۵ در چندضلعی ساده‌ی P ، اگر کوتاه‌ترین مسیر بین s و t داخل چندضلعی T را قطع نکند، کوتاه‌ترین مسیر دیدارکننده‌ی T نیز داخل T را قطع نمی‌کند.

اثبات. فرض کنید p^* کوتاه‌ترین مسیر بین s و t و $(\pi(s, t))$ و p یک مسیر دلخواه بین s و t باشد که داخل T را قطع می‌کند. به سادگی می‌توان دید p و p^* از ابتدا تا یک رأس از P ساختار یکسانی دارند. در این رأس، این دو مسیر از یکدیگر جدا می‌شوند و p وارد T می‌شود و بعد از آن خارج می‌شود و دوباره دو مسیر به هم می‌پیوندند. چون P یک چندضلعی ساده است، فضای بین این دو مسیر به طور کامل داخل P قرار می‌گیرد. از آنجا که دو مسیر p و p^* یکی نیستند و با استناد به گزاره‌های ۱.۲ و ۲.۲، p حتماً دارای یک یا چند شکست محدب^۱ است. منظور از شکست محدب، شکستی است که زاویه‌ای که یال‌های حادث به آن داخل چندضلعی می‌سازند کوچکتر از 180° باشد. چنین شکستی حتماً در داخل P قابل میان‌بر زدن است. ما می‌توانیم به طور مکرر شکست‌های محدب p را میان‌بر بزنیم تا اولین جایی که مسیر حاصل اشتراکی با T نداشته باشد. این اتفاق حتماً رخ می‌دهد، چون با میان‌بر زدن طول مسیر کوتاه‌تر می‌شود تا در نهایت بر p^* منطبق

^۱convex bend

لم ۲.۵ در چندضلعی ساده P ، اگر کوتاه‌ترین مسیر بین s و t با T اشتراک نداشته باشد، کوتاه‌ترین مسیر دیدارکننده T فقط در یک نقطه با T اشتراک خواهد داشت.

اثبات. فرض کنید p کوتاه‌ترین مسیر دیدارکننده T بین s و t باشد. طبق فرض لم و با استناد به لم ۱.۵ این اشتراک فقط روی مرز T رخ می‌دهد. در صورتی که p و T در یک رأس انعکاسی P اشتراک داشته باشند، می‌توان شکست‌های محدب این مسیر را میان‌بر زد و به $\pi(s, t)$ رسید که خلاف فرض لم است. بنابراین تمام اشتراک‌ها در شکست‌های محدب رخ می‌دهد که این نیز نمی‌تواند در بیش از یک مورد رخ دهد. \square

با این نتایج، می‌توان مسئله را به دو حالت تقسیم کرد: اگر $\pi(s, t)$ با T اشتراک داشته باشد، خود این مسیر جواب مسئله است. در غیر این صورت باید به دنبال تنها نقطه‌ی تماس T و دیدارکننده T مثل c باشیم. با پیدا کردن این نقطه، مسیر جواب از دنبال هم چسباندن $\pi(s, c)$ و $\pi(c, t)$ به دست می‌آید.

برای یافتن نقطه‌ی تماس بهینه، مانند مسائل دیگری که بررسی کرده‌ایم، مرز T را به بازه‌های بهینگی متناظر با s و t تقسیم می‌کنیم و برای هر بازه در زمان ثابت نقطه‌ای از آن را که کم‌ترین مجموع فاصله‌ها تا s و t دارد پیدا می‌کنیم و نقطه‌ی تماس بهینه را بین این نقاط پیدا می‌کنیم.

به دلیل این که چندضلعی T هر شکلی می‌تواند داشته باشد، تعداد بازه‌های بهینگی متناظر با یک نقطه دلخواه در بدترین حالت $O(n^2)$ است که این موجب بالا رفتن مرتبه‌ی زمان اجرای الگوریتم می‌شود. در اینجا می‌توان از مفهوم دروازه که در بخش ۱.۳.۳ به منظور محدود کردن بازه‌های بهینگی مورد بررسی در دامنه‌های چندضلعی‌گونه تعریف شده استفاده کرد. در اینجا نیز دقیقاً همان مفهوم را برای تعریف دروازه‌ی s روی T به کار می‌گیریم. به این ترتیب دروازه‌ی s مجموعه نقاطی روی مرز T خواهد بود که هنگام حرکت روی کوتاه‌ترین مسیریها از s به نقاط مختلف P برای اولین بار ملاقات می‌شوند. با همان استدلال لم ۵.۳ اندازه‌ی این دروازه (تعداد

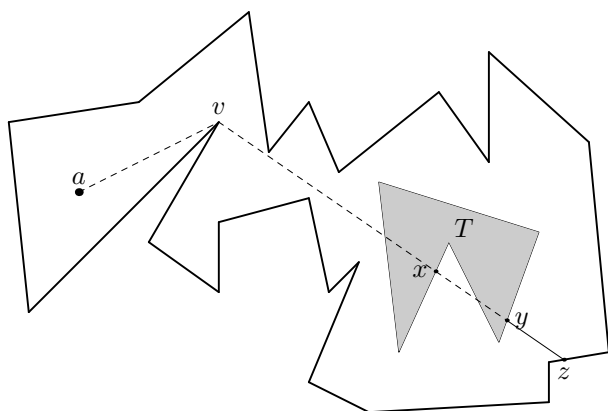
پاره‌خط‌های تشکیل دهنده‌ی آن $O(n)$ است.

به سادگی می‌توان دید که نقطه‌ی c که در بالا تعریف شده است متعلق به دروازه‌ی s و دروازه‌ی t روی T است. به همین دلیل اگر بازه‌های بهیئگی را روی دروازه‌ی s را محاسبه کنیم و آن را L_s بنامیم و آن را با L_t که مجموعه بازه‌های بهیئگی t روی دروازه‌ی t است ادغام کنیم تا مجموعه بازه‌های بهیئگی $L_{s,t}$ را به دست آوریم، $O(n)$ بازه خواهیم داشت که به کمک آنها می‌توان در زمان خطی نقطه‌ی تماس بهینه را پیدا کرد.

لم ۳.۵ نشان می‌دهد که چگونه مجموعه‌ی L_s را محاسبه کرد. محاسبه‌ی L_t نیز با روشی مشابه انجام می‌شود.

لم ۳.۵ مجموعه بازه‌های بهیئگی متناظر با یک نقطه‌ی $a \in P$ روی دروازه‌ی a به T در زمان $O(n)$ قابل محاسبه است.

اثبات. مجموعه بازه‌های مورد نظر، زیرمجموعه‌ای از بازه‌هایی هستند که از تقاطع T با یال‌های $SPM(a, P)$ روی مرز T ایجاد می‌شوند. برای این که مجبور نباشیم تمام $O(n^2)$ بازه‌ی حاصل از این تقاطع را محاسبه کنیم، داخل T را به عنوان یک مانع در نظر می‌گیریم و فضای حاصل از برداشتن T از P را P' می‌نامیم ($P' = P - T$). P' یک چندضلعی ساده با دقیقاً یک حفره است. اگر P' را روی یک پاره‌خط که یک نقطه از مرز درونی آن را به یک نقطه از مرز بیرونی آن وصل می‌کند برش دهیم، یک چندضلعی ساده به دست می‌آید که مرز T را روی مرز خود دارد. در این کار باید دقت کرد که برش ایجاد شده مانع رسیدن یک کوتاه‌ترین مسیر از a به نقطه‌ای روی دروازه‌ی a نشود. برای پیدا کردن برش مناسب، یک نقطه‌ی دلخواه x را روی مرز T انتخاب می‌کنیم (شکل ۲). با الگوریتم گویباس و دیگران برای یافتن کوتاه‌ترین مسیرها، آخرین رأس P از $\pi(a, x)$ را پیدا می‌کنیم و آن را v می‌نامیم. نیم‌خطی که v انتهای آن است و از x می‌گذرد ممکن است مرز T را در چندین نقطه قطع کند. ما دورترین آنها را در نظر می‌گیریم و آن را y می‌نامیم. حالا برش P' را در



شکل ۲: اثبات لم $P - T - ۳.۵$ را در امتداد پاره خط yz برش می‌زنیم تا چندضلعی ساده‌ی P'' حاصل شود.

امتداد نیم خط یاد شده از y شروع می‌کنیم و تا اولین تقاطع با مرز P که در سمت مقابل v واقع شده (z) ادامه می‌دهیم. شکل حاصل یک چندضلعی ساده خواهد بود که آن را P'' می‌نامیم. اگر x را نقطه‌ای به جز رأس‌های T انتخاب کنیم طوری که vx با هیچ یک از یال‌های T هم خط نباشد (که کار ساده‌ای است)، برش yz به طور کامل خارج T واقع می‌شود. دقت کنید که yz بخشی از کوتاه‌ترین مسیر از a به z است. از آنجا که هیچ دو کوتاه‌ترین مسیری در یک چندضلعی ساده یک‌دیگر را قطع نمی‌کنند، yz مانعی برای هیچ کوتاه‌ترین مسیری از a به یک نقطه روی دروازه‌ی a نخواهد بود.

برای پیدا کردن بازه‌های بهینگی a روی دروازه‌ی a ، $SPM(a, P'')$ را محاسبه می‌کنیم. این نقشه بازه‌هایی را روی مرز P'' ایجاد می‌کند. در نتیجه مرز T نیز به تعدادی بازه تقسیم می‌شود. این بازه‌ها متناظر با سلول‌های نقشه‌ی یاد شده هستند. همانند دامنه‌های چندضلعی‌گونه برخی از این بازه‌ها، بازه‌های بهینگی مورد نظر هستند و برخی از آنها ناشی از مانع در نظر گرفتن داخل T به وجود آمده‌اند. برای جداسازی این دو نوع بازه، مانند روش بخش ۲.۳.۳ عمل می‌کنیم و ریشه‌ی سلول سازنده‌ی هر یک از بازه‌ها (مانند r) را در درخت‌های $SPT(a, P)$ و $SPT(a, P'')$ پیدا می‌کنیم. اگر مسیر a به r در این دو درخت یکسان بود، بازه‌ی مورد بررسی یک بازه‌ی بهینگی a روی دروازه‌ی a است. مشکل اینجاست که اگر این جداسازی را بخواهیم پس از محاسبه‌ی کامل

نقشه و به دست آمدن تمام بازه‌ها انجام دهیم باید زمان $O(n \log n)$ صرف این کار شود. به همین خاطر این جداسازی باید در حین ساختن نقشه‌ی $SPM(a, P'')$ هنگام رسیدن به رأس‌های T ، ادامه‌ی ساخت نقشه در آن شاخه متوقف شود. □

به این ترتیب، اگر بخواهیم قدم‌های طی شده در الگوریتم یافتن کوتاه‌ترین مسیر دیدارکننده‌ی T را خلاصه کنیم، ابتدا در زمان خطی می‌توان کنترل کرد که آیا s یا t (یا هر دو) داخل T هستند یا این که $\pi(s, t)$ با T اشتراک دارد. در این حالت‌ها $\pi(s, t)$ دیدارکننده‌ی T است. در غیر این حالات، از الگوریتم گویاس و دیگران [۵۴] استفاده می‌کنیم تا کوتاه‌ترین فاصله‌ها از s و t تا رأس‌های P را محاسبه کنیم. علاوه بر این، بازه‌های بهینگی روی دروازه‌های s و t را طبق روش ارائه شده در لم ۳.۵ محاسبه می‌کنیم. تمام این قدم‌ها در زمان و حافظه‌ی $O(n)$ قابل انجام است. برای این که بتوانیم این دو مجموعه بازه را در زمان خطی با هم ادغام کنیم، همانند مسئله‌ی یافتن کوتاه‌ترین مسیر با قید قابلیت دید، لازم است فراخوانی‌های بازگشتی الگوریتم ساخت کوتاه‌ترین مسیر را به صورت مرتب انجام دهیم. سایر بخش‌های الگوریتم نیز مشابه قید قابلیت دید است. به این ترتیب نتیجه‌ی این بخش در قضیه‌ی زیر خلاصه می‌شود.

قضیه ۱.۵ برای یک چندضلعی ساده‌ی P و یک چندضلعی ساده‌ی T درون آن، برای هر دو نقطه‌ی s و t در P ، طول کوتاه‌ترین مسیر دیدارکننده‌ی T را می‌توان در زمان و حافظه‌ی خطی برحسب تعداد رأس‌های دو چندضلعی P و T محاسبه کرد. یافتن کوتاه‌ترین مسیر نیز با پرداخت هزینه‌ای اضافی متناسب با تعداد یال‌های مسیر قابل انجام است.

۲.۱.۵ ساختن زیرتقسیم پرس وجوی تک‌مبدأ

در این بخش مسئله‌ی ساختن زیرتقسیمی از چندضلعی P را بررسی می‌کنیم که بتوان با آن به پرس وجوی تک‌مبدأ پاسخ داد. در اینجا هدف، پیش‌پردازش ورودی (P, T, s) است طوری که به

ازای نقطه‌ی پرس و جوی t بتوان کوتاه‌ترین مسیر دیدارکننده‌ی T را در زمان $O(\log n)$ پیدا کرد. به این زیرتقسیم $SPM_T(s)$ می‌گوییم. همان‌طور که خواهیم دید این مسئله برای حالت‌های محدب و مقعر T متفاوت است که این دو حالت را در ادامه بررسی می‌کنیم.

چندضلعی هدف محدب

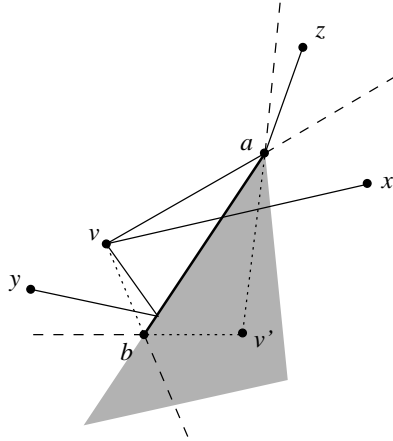
در این حالت، می‌توان مسئله را به عنوان حالت خاصی از صورت کلی مسئله‌ی سفر چندضلعی‌ها^۲ در نظر گرفت که توسط درور و دیگران در [۲۷] بررسی شده است. در آن مسئله هدف یافتن کوتاه‌ترین مسیر بین دو نقطه در صفحه است طوری که با k چندضلعی محدب با ترتیب مشخص دیدار کند. در صورت کلی این مسئله، ورودی شامل تعدادی ناحیه به شکل چندضلعی است که حصار^۳ نامیده می‌شوند و مسیر نهایی مقید است که در فاصله‌ی دیدار دو چندضلعی متوالی درون حصار مشخص شده‌ای بماند. زمان اجرای این الگوریتم $O(nk^2 \log n)$ است. در نتیجه‌ی اجرای این الگوریتم، زیرتقسیمی از صفحه ساخته می‌شود که با آن می‌توان به پرس و جوی تک‌مبدأ پاسخ داد. اگر این مسئله را به ازای $k = 1$ در نظر بگیریم که T تنها چندضلعی باشد که لازم است دیدار شود و P را نیز به عنوان حصار فرض کنیم، نتیجه‌ی اجرای الگوریتم ساخت $SPM_T(s)$ خواهد بود که در زمان $O(n \log n)$ قابل به دست آوردن است. در اینجا نشان می‌دهیم که چگونه می‌توان این زمان را به $O(n)$ تقلیل داد.

در حالتی که T یک چندضلعی محدب باشد به سادگی می‌توان مشاهده کرد که دروازه‌ی s روی T یک زنجیره‌ی چندضلعی گونه^۴ خواهد بود. به این معنی که بخشی همبند از مرز T را تشکیل می‌دهد. این مجموعه توسط بازه‌های بهینگی به چند پاره‌خط تقسیم می‌شود. هر یک از پاره‌خط‌ها متناظر با یک رأس از P است که آخرین رأس روی کوتاه‌ترین مسیرها به نقاط آن بازه

^۲ general touring polygons problem (general TPP)

^۳ fence

^۴ polygonal chain



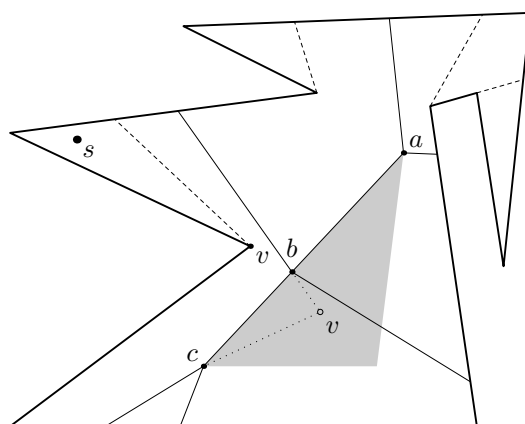
شکل ۳: اگر پاره خط ab یک بازه‌ی بهینگی متناظر با رأس v باشد، آخرین بخش از کوتاه‌ترین مسیر دیدارکننده‌ی T به نقطه‌های x ، y و z با خطوط ممتد نشان داده شده‌اند. بخش خاکستری رنگ چندضلعی هدف (T) است. v' قرینه‌ی v نسبت به بازه‌ی ab است.

است. به عنوان مثال در شکل ۳، v آخرین رأس بازه‌ی بهینگی ab روی دروازه‌ی s به T فرض شده است. برای نقطه‌ای مانند x ، کوتاه‌ترین مسیر دیدارکننده‌ی T از v مستقیماً به x می‌رود، برای y ، از v به ab می‌رود و از آن به y منعکس می‌شود و برای z ، از طریق a به z می‌رود.

مجموعه‌ی R را به عنوان مجموعه پرتوهایی تعریف می‌کنیم که دروازه‌ی s را در راستای کوتاه‌ترین مسیرهای دیدارکننده‌ی T ترک می‌کنند. لم ۴.۵ با اندک تغییراتی از [۲۷] اقتباس شده و از یکسان بودن مسیر بهینه محلی و مسیر بهینه‌ی سراسری ناشی می‌شود.

لم ۴.۵ اگر مرز P مانع نباشد، هر نقطه در صفحه دقیقاً توسط یک پرتو از R قابل دسترسی است.

زیرتقسیم S^R از صفحه را طوری تعریف می‌کنیم که مجموعه نقاطی از صفحه که پرتوهای رسیده به آنها در R از یک رأس T یا یک طرف یک بازه‌ی بهینگی آغاز شده باشد در یک بخش قرار بگیرند. این زیرتقسیم از امتداد دادن پاره‌خط‌های واصل رأس‌های متناظر با بازه‌های بهینگی و قرینه‌های آنها به دو انتهای بازه به دست می‌آیند (شکل ۴).



شکل ۴: نقشه‌ی کوتاه‌ترین مسیر دیدارکننده‌ی T - یال‌های ممتد نازک که از نقاط a ، b و c به بیرون رسم شده‌اند در امتداد پرتوهای S^R هستند که به همراه یال‌های P و بازه‌های بهیئنگی ab و bc زیرتقسیم S^P را می‌سازند. یال‌های خط‌چین متعلق به نقشه‌ی کوتاه‌ترین مسیر سلول‌های S^P هستند.

اگر زیرتقسیم S^R را به فضای درون P محدود کنیم، زیرتقسیم دیگری به وجود می‌آید که آن را S^P می‌نامیم. این زیرتقسیم P را به تعدادی سلول تقسیم می‌کند طوری که کوتاه‌ترین مسیر دیدارکننده‌ی T به تمام نقاط یک سلول، T را از رأس یکسان یا از یک طرف بازه‌ی بهیئنگی یکسانی ترک می‌کند. ریشه‌ی یک سلول از S^P را به این صورت تعریف می‌کنیم که اگر این سلول متناظر با یک رأس از T است، همان رأس ریشه‌ی سلول خواهد بود. اگر سلول متناظر با یک بازه‌ی بهیئنگی باشد دو حالت خواهیم داشت بسته به این که مسیرها به نقاط سلول، از بازه بگذرند یا از آن منعکس شوند. در حالت اول، رأس P که متناظر با بازه‌ی یاد شده است ریشه‌ی سلول است و در حالت دوم، قرینه‌ی این رأس نسبت به آن بازه. برای مثال در شکل ۴، ریشه‌ی سلول شامل s ، نقطه‌ی \bar{v} است که قرینه‌ی v نسبت به بازه‌ی bc است.

برای محاسبه‌ی $SPM_T(s)$ ، هر یک از سلول‌های S^P را به طور جداگانه در نظر می‌گیریم و برای هر یک، نقشه‌ی کوتاه‌ترین مسیر آن را با در نظر گرفتن ریشه‌ی آن به عنوان مبدأ می‌سازیم. برای سلول‌هایی که ریشه‌ی آنها یک رأس T نیست، لازم است مثلثی که ریشه را به دو سر بازه‌ی مربوطه وصل می‌کند به سلول اضافه کنیم تا ریشه نیز جزء سلول قرار بگیرد. پس از

محاسبه‌ی نقشه‌ی کوتاه‌ترین مسیر سلول، مثلث یاد شده را دوباره از سلول حذف می‌کنیم. به این ترتیب، یال‌های $SPM_T(s)$ از سه دسته یال به دست می‌آیند: بازه‌های بهینگی روی دروازه، یال‌های سلول‌های S^P و یال‌های نقشه‌های کوتاه‌ترین مسیر داخل سلول‌ها.

لم ۵.۵ نشان می‌دهد چگونه زیرتقسیم S^P در زمان خطی قابل محاسبه است.

لم ۵.۵ زیرتقسیم S^P در زمان و حافظه‌ی $O(n)$ قابل محاسبه است.

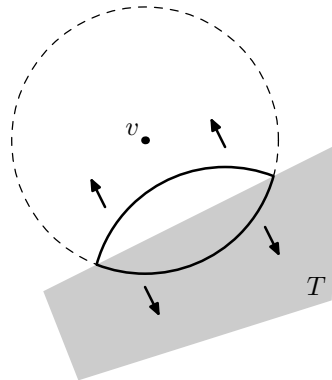
اثبات. طبق لم ۳.۵ می‌توانیم بازه‌های بهینگی s روی T را محاسبه کنیم. روش استفاده شده تضمین می‌کند که بازه‌ها روی مرز T به صورت مرتب به دست می‌آیند. بنابراین می‌توانیم پرتوهای S^R را به صورت مرتب در زمان خطی به دست آوریم. برای محاسبه‌ی S^P باید برای هر یک از پرتوهای S^R اولین یالی از P را پیدا کنیم که با آن تقاطع دارد. اگر بخواهیم برای این کار از الگوریتم‌های شلیک پرتو استفاده کنیم، بهترین زمانی که برای تمام $O(n)$ پرتوی S^R به دست می‌آوریم $O(n \log n)$ است که زمان مطلوبی نیست. به همین خاطر با پیمایش دنباله پرتوها و یال‌های P در یک جهت به ترتیب زیر تقاطع‌های مورد نظر را محاسبه می‌کنیم. فرض می‌کنیم (e_1, e_2, \dots, e_n) دنباله یال‌های P در جهت ساعت‌گرد و (r_1, r_2, \dots, r_m) دنباله پرتوهای S^R در جهت ساعت‌گرد روی مرز T باشند. دو پرتو متوالی مانند r_i و r_{i+1} را در نظر بگیرید و فرض کنیم تقاطع r_i با یال e_j از P را داریم که e_j اولین یال در دنباله یال‌های P است که r_i را قطع می‌کند (توجه کنید که این تقاطع لزوماً اولین تقاطع r_i با یک یال P نیست). ما یال‌های e_{j+1}, e_{j+2}, \dots را یک‌به‌یک و به ترتیب بررسی می‌کنیم و همیشه آخرین یالی را که r_i را قطع می‌کند نگه می‌داریم، تا موقعی که یالی مانند e_k را پیدا کنیم که r_{i+1} را قطع می‌کند. در این لحظه، آخرین تقاطع پیدا شده برای r_i را نهایی می‌کنیم و همین روند را برای شعاع‌های بعدی ادامه می‌دهیم (این بار با داشتن تقاطع r_{i+1} و e_k). در این روش، هر یک از یال‌ها و پرتوها را فقط یک‌بار پردازش می‌کنیم و تمام اعمال انجام شده هزینه‌ی ثابت دارند. برای یافتن اولین تقاطع r_1 نیز زمان $O(n)$ صرف می‌شود. به این ترتیب زمان کل محاسبه‌ی S^P برابر $O(n)$ است. □

زیرتقسیم S^P دارای $O(n)$ سلول است که برای هر یک باید به طور جداگانه الگوریتم ساخت نقشه‌ی کوتاه‌ترین مسیر را اجرا کنیم. از آنجا که پیچیدگی کل زیرتقسیم $O(n)$ است، مجموع زمان صرف شده برای ساخت نقشه‌ها نیز $O(n)$ خواهد بود. به این ترتیب نتیجه‌ی به دست آمده برای این مسئله را می‌توان در قضیه‌ی زیر خلاصه کرد.

قضیه ۲.۵ اگر P یک چندضلعی ساده و T یک چندضلعی ساده درون آن باشد، با داشتن یک نقطه‌ی مبدأ s می‌توان ورودی (P, T, s) را در زمان و حافظه‌ی $O(n)$ پیش‌پردازش کرد طوری که برای یک نقطه‌ی پرس‌وجوی q طول کوتاه‌ترین مسیر دیدارکننده‌ی T از s به t در زمان $O(\log n)$ قابل بازیابی است. خود مسیر نیز با پرداخت هزینه‌ی اضافی متناسب با تعداد یال‌های مسیر قابل محاسبه است.

چندضلعی هدف مقعر

اگر چندضلعی T مقعر باشد، لم ۴.۵ دیگر صادق نخواهد بود. به این معنی که نقاطی در P هستند که می‌توان با بیش از یک مسیر دیدارکننده‌ی T که به طور محلی بهینه است به آنها رسید و این مسیرها T را به نحو متفاوتی ترک می‌کنند. این موضوع باعث به وجود آمدن یال‌های جدید در زیرتقسیم S^P می‌شود که منصف‌های رأس‌هایی از T هستند. علاوه بر این، داشتن بازه‌های بهینگی روی دروازه‌ی s به طور مرتب به این معنی نخواهد بود که تقاطع پرتوهای S^R با مرز P به همان ترتیب رخ خواهد داد. این نیاز به مرتب کردن اضافی خواهد داشت که هزینه‌ی $O(n \log n)$ را به الگوریتم تحمیل می‌کند. در این حالت طبق روش ارائه شده در بخش بعد عمل می‌کنیم که با استفاده از الگوریتم‌های انتشار موج، نقشه‌ی مورد نظر را در زمان $O(n \log n)$ می‌سازد.



شکل ۵: بازتابش یک موجک - دایره‌ی خط‌چین یک موجک به مرکز v است. انتشار بخشی از موجک ابتدایی که داخل T شده به همراه نسخه‌ی بازتابش یافته‌ی آن.

۲.۵ الگوی بازتابش جبهه‌ی موج

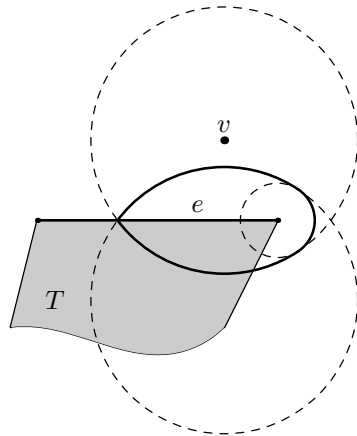
با مشاهداتی که در مورد زیرتقسیم بازتابی صورت گرفت، می‌توان از الگوی دایکسترای پیوسته برای محاسبه‌ی نقشه‌ی کوتاه‌ترین مسیر دیدارکننده‌ی T در یک دامنه‌ی دوبعدی استفاده کرد. ایده‌ی شهودی این روش این است که مانند روش دایکسترای پیوسته برای محاسبه‌ی کوتاه‌ترین مسیرها، در ابتدا یک جبهه‌ی موج از مبدأ (در اینجا p) منتشر می‌شود تا این که به مرز T برخورد کند. هنگام این برخورد، علاوه بر این که جبهه‌ی موج به انتشار خود ادامه می‌دهد، یک نسخه‌ی بازتابش یافته از آن نیز برمی‌گردد که مکان هندسی نقاطی را تعریف می‌کند که با حرکت از مبدأ دیدار مرز T و برگشتن به آن نقاط، مسافت ثابتی طی می‌شود (شکل ۵).

جبهه‌ی موجی را که ابتدا از s منتشر می‌شود جبهه‌ی موج ابتدایی^۵ می‌خوانیم و به اجزای آن، موجک‌های ابتدایی می‌گوییم. هنگام برخورد یک موجک ابتدایی با مرز T ، یک موجک دیدار کرده^۶ از یکی از سه نوع زیر ایجاد می‌شود (شکل ۶):

(۱) موجکی که معادل بخشی از موجک ابتدایی است که داخل T شده است

^۵ original wavefront

^۶ visited wavelet



شکل ۶: سه نوع موجک دیدار کرده‌ی اولیه با کمان‌های ممتد نشان داده شده‌اند.

(۲) موجکی که از بازتابش موجک قبل نسبت به یالی از T که برخورد با آن صورت گرفته است به دست می‌آید

(۳) موجکی با شعاع صفر که هنگام برخورد موجک ابتدایی با یک رأس T ایجاد می‌شود.

ما این سه نوع موجک را که بلافاصله بعد از برخورد یک موجک ابتدایی با مرز T ایجاد می‌شوند موجک‌های دیدار کرده‌ی اولیه^۷ می‌نامیم. در هر سه شکل بالا، موجک ایجاد شده کمانی از یک دایره است. مرکز این دایره برای حالت اول، همان مرکز دایره‌ی موجک ابتدایی است، برای حالت دوم، قرینه‌ی مرکز دایره‌ی موجک ابتدایی نسبت به یال مورد برخورد است و در حالت سوم، رأس T که برخورد با آن صورت گرفته. بلافاصله بعد از برخورد، موجک‌های دیدار کرده‌ی اولیه شروع به انتشار می‌کنند. این انتشار طبق قوانین معمول انتشار جبهه‌ی موج در دامنه‌های چندضلعی گونه صورت می‌گیرد. طبق این قوانین، هنگام برخورد جبهه‌ی موج با رأس یک مانع، یک موجک جدید به جبهه‌ی موج اضافه می‌شود. به این موجک‌ها، موجک‌های دیدار کرده‌ی ثانویه^۸ می‌گوییم. توجه کنید که در جریان انتشار، برخورد موجک‌های دیدار کرده و موجک‌های ابتدایی نادیده گرفته می‌شود. توجه داشته باشید که برخلاف جبهه‌ی موج ابتدایی، جبهه‌ی موج دیدار کرده

^۷ primary visited wavelets

^۸ secondary visited wavelets

ممکن است ناهمبند باشد که در این حالت از تعدادی زنجیره‌ی بسته از موجک‌ها تشکیل می‌شود. از آنجا که انتشار دو جبهه‌ی موج ابتدایی و دیدار کرده به طور همزمان در دامنه کاری مشکل است، برای ساختن نقشه‌ی کوتاه‌ترین مسیر دیدارکننده‌ی T طبق الگوی بازتابش جبهه‌ی موج دو قدم عمده‌ی زیر را طی می‌کنیم:

(۱) انتشار جبهه‌ی موج ابتدایی و محاسبه‌ی موجک‌های دیدار کرده‌ی اولیه

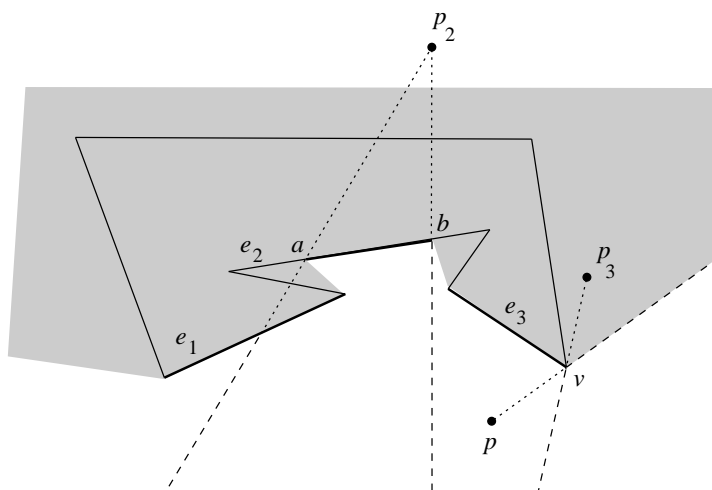
(۲) انتشار جبهه‌ی موج دیدار کرده

برای این که بتوانیم این دو مرحله را به طور دقیق‌تر مشخص کنیم، مفهومی به نام زیرتقسیم بازتابی تعریف می‌کنیم که در ادامه خواص آن را مطالعه می‌کنیم.

۱.۲.۵ زیرتقسیم بازتابی

در این بخش خواص پایه‌ی زیرتقسیمی را مطالعه می‌کنیم که نقشی شبیه به S^R که در گذشته معرفی شد برای چندضلعی هدف مقعر بازی می‌کند. فرض کنید T چندضلعی هدف با m رأس باشد که ممکن است محدب یا نامحدب باشد و p یک نقطه‌ی مشخص خارج T باشد (توجه داشته باشید که فعلاً هیچ مانعی در دامنه‌ی مسئله در نظر گرفته نمی‌شود). دروازه‌ی p به T یا $G(p)$ را به عنوان مجموعه نقاطی از T تعریف می‌کنیم که کوتاه‌ترین مسیرهای دیدارکننده‌ی T برای اولین بار در آنها با T اشتراک پیدا می‌کند.

برای حالتی که T محدب باشد، $G(p)$ بخشی از مرز T است که بین نقاط تماس دو مماس از p به T محدود است و به p نزدیک‌تر است. به این ترتیب $G(p)$ همبند است و از دنباله‌ای از یال‌های مجاور هم تشکیل شده است. برای حالتی که T نامحدب باشد، $G(p)$ لزوماً همبند نیست. اگرچه این مجموعه هنوز روی بخشی از مرز T است که بین دو نقطه‌ی تماس یاد شده قرار دارد اما بعضی از قسمت‌های این بخش جزء دروازه نیستند. از آنجا که T یک چندضلعی ساده است و



شکل ۷: چندضلعی هدف با خطوط ممتد نشان داده شده. $G(p)$ متشکل از پاره‌خط‌های ضخیم است. p_i قرینه‌ی p نسبت به یال e_i است. قسمت خاکستری رنگ، بخشی از ناحیه‌ی درون‌گذر است. ناحیه‌های بازتابش بازتابنده‌های یالی ab و رأسی v به خطوط خط‌چین محدود می‌شوند.

مانعی نیز در صفحه وجود ندارد، برای یک یال e از T حداکثر یک پاره‌خط روی یال e در $G(p)$ قرار دارد (شکل ۷).

به دست آوردن $G(p)$ از طریق الگوریتم‌های محاسبه‌ی چندضلعی قابلیت دید ممکن است چون اگر داخل T را به عنوان یک مانع در نظر بگیریم، $G(p)$ بخشی از مرز ناحیه‌ی قابل دید p خواهد بود.

برای نقطه‌ی دلخواه x ، کوتاه‌ترین مسیر دیدارکننده‌ی T بین p و x را در نظر بگیرید. اگر پاره‌خط واصل p و x با T اشتراک داشته باشد، این پاره‌خط کوتاه‌ترین مسیر مورد نظر است. به مجموعه‌ی نقاط این چنین، که یک ناحیه‌ی همبند از صفحه را تشکیل می‌دهند، ناحیه‌ی درون‌گذر^۹ می‌گوییم، به این تعبیر که مسیرهای مورد نظر، از درون T می‌گذرند. اگر پاره‌خط px با T اشتراک نداشته باشد، مسیر مورد نظر از دو پاره‌خط تشکیل شده است که اولی p را به یک نقطه‌ی c از $G(p)$ وصل می‌کند و دیگری c را به x متصل می‌کند. به نقطه‌ی دیدار x می‌گوییم.

اگر c یک نقطه‌ی داخلی از یک یال e از T باشد، زاویه‌ی دو پاره‌خط حادث به c با e شکل

^۹ pass-through region

بازتابش خواهند داشت. همان‌گونه که پیش‌تر ذکر شد، بخشی از $G(p)$ که روی یال e قرار می‌گیرد یک پاره‌خط است که در این حالت آن را یک بازتابنده‌ی یالی^{۱۰} می‌نامیم. حالت دیگر این است که c منطبق بر یک رأس v از T باشد که در این صورت به v یک بازتابنده‌ی رأسی^{۱۱} می‌گوییم. به سادگی می‌توان مشاهده کرد که رأسی در $G(p)$ بازتابنده‌ی رأسی است که زاویه‌ی داخلی آن در T کمتر از 180° است. اگر r یک بازتابنده‌ی رأسی باشد، رأس متناظر و اگر یک بازتابنده‌ی یالی باشد، قرینه‌ی p نسبت به یال مورد نظر را کانون^{۱۲} آن بازتابنده می‌نامیم.

از آنجا که ناحیه‌ی درون‌گذر به سادگی قابل محاسبه است و ساختار کوتاه‌ترین مسیر دیدارکننده‌ی T به نقاط آن با سایر نقاط صفحه تفاوت عمده‌ای می‌کند، از این پس مطالعه را به مکمل این ناحیه که آن را D می‌نامیم محدود می‌کنیم.

به ازای یک بازتابنده‌ی یالی $r = ab$ ، ناحیه‌ی بازتابش^{۱۳} r یا $A(p, r)$ را به عنوان مجموعه نقاطی از D تعریف می‌کنیم که می‌توان از p با بازتابش از r به آنها رسید. به طور دقیق‌تر اگر p_r قرینه‌ی p نسبت به r باشد، $A(p, r)$ ناحیه‌ای نامتناهی از D است که به دو پرتو تابیده شده از a و b به ترتیب در امتدادهای $\vec{p_r a}$ و $\vec{p_r b}$ محدود شده است. به همین صورت، به ازای یک بازتابنده‌ی رأسی v که e_1 و e_2 دو یال حادث به آن هستند، $A(p, v)$ را به عنوان ناحیه‌ی نامتناهی محدود به پرتوهای تابیده شده از v در امتدادهای $\vec{p_{e_1} v}$ و $\vec{p_{e_2} v}$ تعریف می‌کنیم.

زیرتقسیم بازتابی $RS(p, T)$ ، یا به اختصار $RS(p)$ ، تقسیم D به ناحیه‌هایی همبند است که نقطه‌ی دیدار نقاط یک ناحیه، بازتابنده‌ی رأسی یکسان r یا متعلق به بازتابنده‌ی یالی یکسان r باشند. به چنین ناحیه‌ای یک ناحیه‌ی بازتابی^{۱۴} r می‌گوییم (شکل ۸).

یک ناحیه‌ی بازتابی r به وضوح زیرمجموعه‌ای از $A(p, r)$ است. یک راه ابتدایی برای

edge reflector^{۱۰}
vertex reflector^{۱۱}
focus^{۱۲}
reflex region^{۱۳}
reflective region^{۱۴}

محاسبه‌ی $RS(p)$ این است که تقاطع ناحیه‌های بازتابش تمام بازتابنده‌های ممکن را به دست آوریم. با این کار D به تعدادی ناحیه تقسیم می‌شود که هر یک از آنها از اشتراک تعدادی از ناحیه‌های بازتابش به دست آمده‌اند. فرض کنید ناحیه f که از اشتراک $A(p, r_1), A(p, r_2), \dots, A(p, r_k)$ به دست آمده باشد. در این حالت برای یک نقطه‌ی دلخواه از f ، می‌توان از طریق هر یک از بازتابنده‌های r_i به آن نقطه رسید. برای مشخص کردن بازتابنده‌ای که کوتاه‌ترین مسیر را موجب می‌شود باید f را توسط نمودار وورونوی وزن‌دار جمعی^{۱۵} متناظر با مجموعه‌ی $\{r_1, r_2, \dots, r_k\}$ به ناحیه‌های کوچک‌تری تقسیم کرد. بازتابنده‌های یالی وزن صفر و بازتابنده‌های رأسی وزنی معادل فاصله‌ی رأس متناظر تا p دارند. محاسبه‌ی این نمودار با استفاده از روش خط جارو^{۱۶} در زمان $O(k \log k)$ ممکن است [۷۷، ۷۸]. این الگوریتم‌ها در حالت کلی نمودار وورونوی تعدادی قرص را محاسبه می‌کنند و در نتیجه وزن یک محل ضریب نفوذ آن را مشخص می‌کند. برخلاف این، در مسئله‌ی ما وزن به معنی هزینه است. به همین خاطر برای استفاده از این الگوریتم‌ها باید تغییر لازم در معیار سنجش وزن را اعمال کنیم (این کار با جمع یک مقدار ثابت به همه‌ی وزن‌ها ممکن می‌شود).

برای به دست آوردن $RS(p)$ با این روش، لازم است تقاطع نواحی بازتابش را محاسبه کنیم و برای هر یک از آنها نمودار وورونوی یادشده را برای بازتابنده‌های ایجادکننده‌ی تقاطع به دست آوریم و در آخر نمودارهای وورونوی کل نواحی را با یکدیگر ادغام کنیم تا ناحیه‌های $RS(p)$ به دست آیند. این موضوع مرتبه‌ی زمان اجرای این روش را بالا می‌برد. در ادامه خواهیم دید که چگونه می‌توان این زیرتقسیم را در زمان $O(m \log m)$ به دست آورد.

ابتدا نشان می‌دهیم که پیچیدگی $RS(p)$ خطی است.

additively weighted Voronoi diagram^{۱۵}
sweep line^{۱۶}

لم ۶.۵ برای یک چندضلعی هدف T با رأس، پیچیدگی $RS(s, T)$ از مرتبه $O(m)$ است.

اثبات. ابتدا ثابت می‌کنیم که برای هر بازتابنده r حداکثر یک ناحیه‌ی بازتابی وجود دارد. از آنجا که کوتاه‌ترین مسیر دیدارکننده‌ی T از p به یک نقطه روی بازتابنده‌ی r همان پاره‌خط واصل p به آن نقطه است، تمام نقاط روی r به ناحیه‌های بازتابی r تعلق دارند. اکنون یک نقطه‌ی x را در نظر بگیرید که در یک ناحیه‌ی بازتابی r قرار دارد و فرض کنید c نقطه‌ی دیدار x باشد. نقطه‌ی y را داخل پاره‌خط cx در نظر بگیرید که متعلق به یک ناحیه‌ی بازتابی r' است. چون طول مسیر دیدارکننده‌ی T بین p و y از طریق r' کوتاه‌تر است تا از طریق r ، اضافه کردن پاره‌خط yx به این دو مسیر نتیجه می‌دهد x نیز در ناحیه‌ی r' است که خلاف فرض است. به این ترتیب، برای هر نقطه‌ی x در یک ناحیه‌ی بازتابی، تمام نقاط روی پاره‌خط واصل x به نقطه‌ی دیدار c هم در همان ناحیه‌ی بازتابی واقع می‌شوند.

حالا حالتی را در نظر بگیرید که برای بازتابنده‌ی r با کانون a دو ناحیه‌ی بازتابی جداگانه f_1 و f_2 وجود داشته باشد. طبق آنچه در بالا گفته شد، هر دو این ناحیه‌ها با r اشتراک دارند. چون این دو ناحیه از هم جدا هستند، پرتوی تابیده شده از a در فضای بین f_1 و f_2 قرار دارد که تمام نقاط روی آن از نقطه‌ی تقاطع پرتو و r به بعد در ناحیه‌های بازتابنده‌های دیگری هستند. به این ترتیب اگر نقطه‌ی تقاطع این پرتو و r (مثل x) را در نظر بگیریم، طول کوتاه‌ترین مسیر دیدارکننده‌ی بین p و x از طریق r و بازتابنده‌ی دیگری مانند r' مساوی است. بنابراین x روی خم منصف کانون‌های r و r' قرار دارد. اگر این خم به شکل خط راست باشد، بخشی از یکی از دو ناحیه‌ی f_1 یا f_2 در بخشی از صفحه قرار می‌گیرد که رفتن به آن از طریق r' نزدیک‌تر است و این خلاف فرض است. حالتی که خم یاد شده به شکل خط راست نباشد و دو تقاطع با r داشته باشد نیز مردود است چون در این حالت بخشی از r در ناحیه‌ی متناظر با r' واقع می‌شود که این نیز ممکن نیست. بنابراین، تعداد ناحیه‌های $RS(p)$ از مرتبه $O(m)$ است. بعضی رأس‌های این زیرتقسیم رأس‌های T هستند و عده‌ای از تقاطع منصف‌ها تشکیل شده‌اند. از آنجا که تعداد رأس‌های نوع اول $O(m)$ است و

رأس‌های نوع دوم حداقل بین سه وجه زیر تقسیم مشترکند، تعداد رأس‌ها و در نتیجه پیچیدگی کل زیر تقسیم $O(m)$ است. □

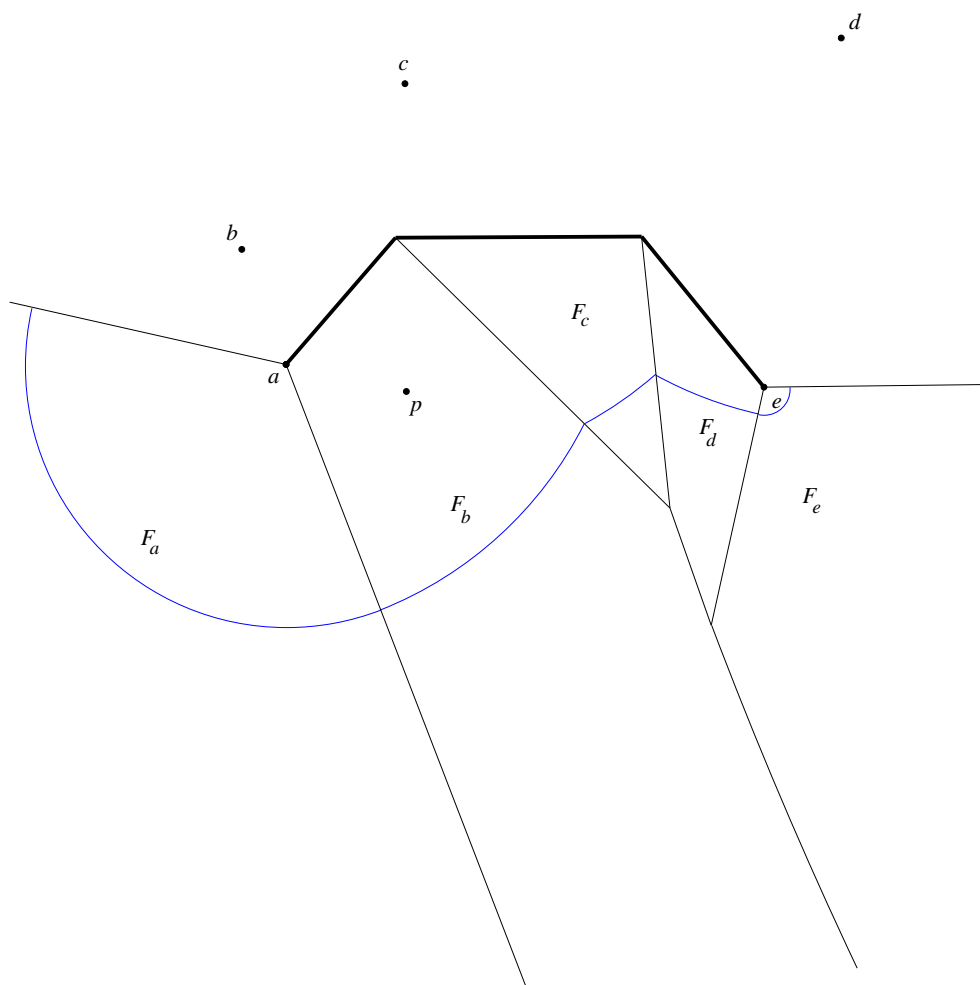
یال‌های $RS(p)$ دو دسته‌اند. دسته‌ی اول بخش‌هایی از مرز D را تشکیل می‌دهند. دسته‌ی دوم یال‌هایی هستند که ناحیه‌های بازتابی مجاور را از هم جدا می‌کنند. در حالت کلی، یالی که دو ناحیه‌ی بازتابی بازتابنده‌های r_1 و r_2 به کانون‌های a و b را از هم جدا می‌کند توسط خم منصف a و b تعریف می‌شود. این خم یک خم هذلولوی است که مکان هندسی نقاطی مثل x است طوری که $w(a) + |ax| = w(b) + |bx|$ که برابر $|pa|$ است اگر a یک بازتابنده‌ی رأسی باشد و صفر است اگر a یک بازتابنده‌ی یالی باشد.

۲.۲.۵ محاسبه‌ی زیر تقسیم بازتابی

فرض کنید $\delta(p, x)$ طول کوتاه‌ترین مسیر دیدارکننده‌ی T از p به یک نقطه‌ی $x \in D$ باشد. برای ساختن زیر تقسیم بازتابی، نقاط D را بر اساس افزایش مقدار δ جارو می‌کنیم. ساختار جارو یک جبهه‌ی موج است که از موجک‌هایی به شکل کمان‌هایی از دایره تشکیل شده است. این کمان‌ها بخش‌هایی از دایره‌هایی به مرکز کانون بازتابنده‌ها هستند. شکل ۹ تصویری از این جبهه‌ی موج را نشان می‌دهد. ابتدا به ازای هر بازتابنده یک موجک وجود دارد. زمان‌هایی^{۱۷} یک بازتابنده‌ی رأسی a برابر طول پاره‌خط pa است. در هر لحظه از فرایند جارو، دو منصف را مجاور می‌گوییم اگر موجک یکسانی را محدود کنند.

تنها رخداد در فرایند جارو هنگامی اتفاق می‌افتد که دو منصف مجاور با هم برخورد کنند. در این لحظه، موجکی که بین این دو منصف قرار دارد از بین می‌رود و دو همسایه‌ی آن با هم مجاور می‌شوند. از آنجا که تقاطع‌ها فقط بین منصف‌های مجاور رخ می‌دهد، هنگام پردازش یک رخداد، هنگامی که یک منصف جدید به وجود می‌آید می‌توان زمان برخورد آن را با منصف‌های

^{۱۷} release time



شکل ۹: کمان‌های دایره، جبهه‌ی موج جارو کننده‌ی D را نشان می‌دهند. مرکز آنها کانون‌های بازتابنده‌ها هستند.

مجاورش محاسبه کرد. به سادگی می توان مشاهده کرد که پردازش تمام $O(m)$ رخداد در زمان $O(m \log m)$ و حافظه $O(m)$ قابل انجام است.

۳.۵ دامنه های چندضلعی گونه

در این بخش، به دنبال ساختن نقشه ی کوتاه ترین مسیر دیدارکننده ی T درون فضای آزاد یک دامنه ی چندضلعی گونه ی P با n رأس هستیم. یک مسیر دیدارکننده ی T ، مسیری درون فضای آزاد است که اشتراک آن با T تهی نباشد. فرض کنید نقطه ی s درون فضای آزاد و بیرون T داده شده است. همانند قبل، دروازه ی s به T را به عنوان مجموعه نقاطی تعریف می کنیم که کوتاه ترین مسیرهای دیدارکننده ی T در آنها برای اولین بار با T اشتراک پیدا می کنند. مانند حالت قبل، به سادگی می توان مشاهده کرد که $G(s)$ از مجموعه ای از پاره خطها روی مرز T تشکیل شده است که بین دو مماس رسم شده از p به T قرار دارند.

مجموعه ای همبند ماکسیمال از نقاط روی یک پاره خط از $G(s)$ را در نظر بگیرید که آخرین رأس کوتاه ترین مسیر از s به آن نقاط یکسان باشد (مثل v). چنین مجموعه ای را که به صورت یک پاره خط است را یک بازتابنده ی یالی می نامیم و کانون آن را به عنوان قرینه ی v نسبت به آن پاره خط تعریف می کنیم. همانند قبل، یک رأس از T را که در $G(s)$ واقع شده یک بازتابنده ی رأسی می نامیم و کانون آن را خود آن رأس فرض می کنیم. بازتابنده های یالی، زیرمجموعه ای از مجموعه پاره خطهایی است که از تقاطع مرز T و $SPM(s, P)$ به دست می آیند. در حالت کلی ممکن است $O(mn)$ پاره خط مانند این وجود داشته باشد، اما با استدلالی مشابه آنچه در اثبات لم ۵.۳ (خطی بودن اندازه ی دروازه ی قابلیت دید) گفته شد، می توان ثابت کرد که تعداد بازتابنده های یالی $O(m+n)$ است. همچنین، محاسبه ی بازتابنده های یالی، مشابه محاسبه ی دروازه ی قابلیت دید انجام می گیرد (بخش ۲.۳.۳ را ببینید). با این روش، یافتن بازتابنده های یالی و رأسی در زمان

$O((m+n)\log(m+n))$ قابل انجام است.

ما ناحیه‌ی درون‌گذر را مانند حالت قبل تعریف می‌کنیم. فرض کنید D_T ناحیه‌ای باشد که از برداشتن ناحیه‌ی درون‌گذر از فضای آزاد حاصل شده باشد. در این صورت، $RS(s, T)$ افزاز D_T به تعدادی ناحیه است بر اساس بازتابنده‌ای که برای اولین بار توسط کوتاه‌ترین مسیرهای دیدارکننده‌ی T از مبدأ s ملاقات شده است. استدلالی مشابه لم ۶.۵ در این حالت نیز نشان می‌دهد که اندازه‌ی این زیرتقسیم $O(m+n)$ است.

محاسبه‌ی نقشه‌ی کوتاه‌ترین مسیر دیدارکننده‌ی T ، $SPM_T(s, P)$ ، می‌تواند توسط الگوی بازتابش جبهه‌ی موج صورت بگیرد. این نقشه دو بخش دارد: یکی متناظر با ناحیه‌ی درون‌گذر و دیگری متناظر با D_T . بخش اول همان $SPM(s, P)$ است که به ناحیه‌ی درون‌گذر محدود شده است. برای بخش دوم، تعدادی مبدأ تعریف می‌شوند که همان کانون‌های بازتابنده‌ها هستند. هر مبدأ زمان‌هایی مشخصی خواهد داشت. برای بازتابنده‌های رأسی، این زمان فاصله‌ی ژئودزیک s به آن رأس است و برای بازتابنده‌های یالی، این زمان، فاصله‌ی ژئودزیک s به آخرین رأس روی کوتاه‌ترین مسیرهای به نقاط آن بازتابنده‌ی یالی است به اضافه‌ی d که فاصله‌ی v از پاره‌خط بازتابنده است. برای پوشاندن نقاط D ، از یک الگوریتم انتشار جبهه‌ی موج استفاده می‌کنیم که جبهه‌ی موج ابتدایی را که از s ایجاد شده است بازتابش کند. توجه داشته باشید که موجک‌های اولیه باید با دقت محاسبه شوند، چون برخی از مبدأها ممکن است بیرون D_T قرار بگیرند. برای یک بازتابنده‌ی یالی، اگر v آخرین رأس کوتاه‌ترین مسیرها از s به نقاط روی بازتابنده باشد، موجک اولیه به مرکز قرینه‌ی v نسبت به بازتابنده و شعاع d است.

هر دو الگوریتم [۷۳] و [۶۰] که الگوریتم‌های شناخته شده‌ی یافتن کوتاه‌ترین مسیرها در دامنه‌های چندوجهی‌گونه هستند که مبتنی بر الگوی انتشار جبهه‌ی موج طراحی شده‌اند، قادر به پردازش چند مبدأ با زمان‌های رهایی متفاوت هستند. اگر از الگوریتم اول (متعلق به هر شبرگر و سوری) استفاده کنیم که در بدترین حالت بهینه است، در زمان و حافظه‌ی $O((m+n)\log(m+n))$

می‌توانیم $SPM_T(s, P)$ را بسازیم. طبق آنچه پیش از این گفته شد، محاسبه‌ی بازتابنده‌ها و در نتیجه موجک‌های دیدارکرده‌ی اولیه هم در همین مرتبه‌ی زمانی قابل انجام است. بنابراین نتیجه‌ی به دست آمده در این حالت را می‌توان در قضیه‌ی زیر خلاصه کرد.

قضیه ۳.۵ برای یک دامنه‌ی چندضلعی گونه‌ی P و یک چندضلعی هدف T داخل P با مجموع رأس‌هایی به تعداد N و داشتن یک نقطه‌ی مبدأ s ، می‌توان ورودی (s, P, T) را در زمان و حافظه‌ی $O(N \log N)$ پیش‌پردازش کرد طوری که زیرتقسیمی به اندازه‌ی $O(N)$ از فضای آزاد تولید شود که با استفاده از آن می‌توان طول یک کوتاه‌ترین مسیر دیدارکننده‌ی T بین s و یک نقطه‌ی پرس‌وجوی t را در زمان $O(\log N)$ محاسبه کرد. چنین مسیری با پرداخت هزینه‌ای متناسب با تعداد یال‌های مسیر قابل بازیابی است.

۴.۵ جستجوی گراف قابلیت دید

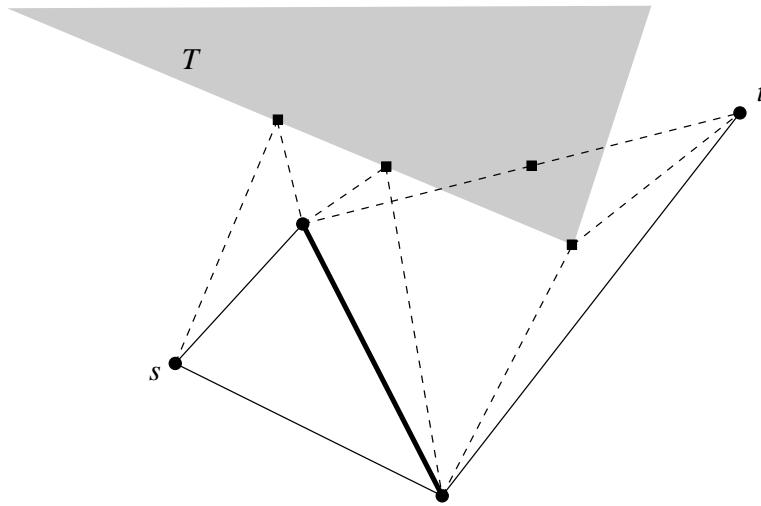
در این بخش مسئله‌ی یافتن یک کوتاه‌ترین مسیر دیدارکننده‌ی T بین دو نقطه‌ی داده شده‌ی s و t را بر مبنای جستجوی گراف قابلیت دید بررسی می‌کنیم. یادآوری می‌کنیم که یافتن کوتاه‌ترین مسیر بین دو نقطه طبق این روش نیازمند ساختن گراف (اقلیدسی) قابلیت دید است که رأس‌های آن، رأس‌های P به همراه s و t هستند و یال‌های آن زوج رأس‌های قابل دید را به هم وصل می‌کند. با ساختن این گراف، می‌توان با اجرای الگوریتم یافتن کوتاه‌ترین مسیر در گراف، کوتاه‌ترین مسیر بین s و t را به دست آورد.

فرض کنید $\Pi_T(s, t)$ مجموعه‌ی کوتاه‌ترین مسیرهای دیدارکننده‌ی T بین s و t باشد و $G = (V, E)$ گراف قابلیت P با در نظر گرفتن s و t جزء رأس‌های گراف باشد. فرض کنید u و v دو رأس از P باشند. $C^\circ(u, v)$ را به عنوان مجموعه نقاط $x \in T$ تعریف می‌کنیم که بین تمام نقاط

T کمترین مقدار $|ux| + |vx|$ را بین تمام نقاط T داشته باشند. فرض کنید $C(u, v)$ زیرمجموعه‌ای از $C^\circ(u, v)$ باشد که از هر دو رأس u و v قابل دید است. در صورتی که T یک چندضلعی محدب باشد، این مجموعه یا تهی است یا شامل فقط یک نقطه روی مرز T . در صورتی که T نامحدب باشد ممکن است در این مجموعه نقاط بیشتری وجود داشته باشند.

فرض کنید π یک مسیر در $\Pi_T(s, t)$ باشد. به سادگی می‌توان مشاهده کرد π هیچ‌گاه در داخل T نمی‌شکند (شکست‌ها فقط در مرز T رخ می‌دهند). این مشاهده بر مبنای خواص بهینگی موضعی مسیرها قابل اثبات است. مجموعه نقاط $A = \pi \cup T$ را در نظر بگیرید. نقاط A یا روی یال‌های گراف قابلیت دید قرار دارند یا این که طبق قوانین بهینگی محلی در $C(u, v)$ یافت می‌شوند که u و v دو رأس از P هستند. در این حالت، مسیر از u مستقیماً به نقطه‌ی تماس می‌رود و مجدداً به v بر می‌گردد. باز بر مبنای این واقعیت که π یک مسیر بهینه است، می‌توان بررسی کرد که فقط یک نقطه در A یافت می‌شود که روی یال‌های گراف قابلیت دید نباشد (که در یک مجموعه‌ی $C(u, v)$ خواهد بود).

بر مبنای مشاهدات بالا، از روی گراف قابلیت دید G ، گراف دیگری می‌سازیم که جستجو در آن کوتاه‌ترین مسیر دیدارکننده‌ی T را تولید کند. این گراف را $G_T = (V_T, E_T)$ می‌نامیم و برای ساخت آن از G شروع می‌کنیم (یعنی ابتدا $V_T = V$ و $E_T = E$). سپس هر یال $\{u, v\}$ در E را که با T تقاطع داشته باشد از E_T حذف و یک رأس w را به V_T اضافه می‌کنیم که متناظر با یک نقطه‌ی دلخواه از $uv \cup T$ باشد. این رأس را توسط دو یال جدید در E_T به u و v متصل می‌کنیم و وزن یال‌ها را به ترتیب برابر طول پاره‌خط‌های uw و vw قرار می‌دهیم. بعد از این مرحله، برای تمام جفت رأس‌های u و v در V که $C(u, v)$ تهی نباشد، یک رأس w به V_T اضافه می‌کنیم که متناظر با یک نقطه در $C(u, v)$ است و دوباره یال‌های $\{u, w\}$ و $\{v, w\}$ را به ترتیب با وزن‌های $|uw|$ و $|vw|$ به E_T اضافه می‌کنیم (شکل ۱۰).



شکل ۱۰: گراف G_T - پاره خط ضخیم، یک مانع است. رأس های V با دایره ی توپر و رأس های اضافه شده در V_T با مربع توپر نشان داده شده اند. یال های خط چین در E_T اضافه شده اند. یال های ممتد از قبل در E موجود بوده اند.

مجموعه رأس های $M = V_T - V$ را رأس های علامت دار^{۱۸} در G_T می نامیم. به سادگی می توان مشاهده کرد که اگر مسیری بین s و t در G_T باشد که شامل حداقل یک رأس از M باشد، یک مسیر دیدارکننده ی T بین s و t در P است. بر مبنای مشاهدات بالا، متناظر با هر مسیر π در $\Pi_T(s, t)$ یک کوتاه ترین مسیر بین s و t در G_T یافت می شود که شامل حداقل یک رأس علامت دار است.

برای یافتن چنین مسیری در G_T ، با الگوریتم کوتاه ترین مسیر دایکسترا کوتاه ترین فاصله ها از s به تمام رأس های M را پیدا می کنیم. آنگاه گراف G'_T را از روی G_T با اضافه کردن یک رأس s' به V_T و متصل کردن s' به تمام رأس های M می سازیم. برای یک رأس $m \in M$ ، وزن یال $\{s', m\}$ را برابر کوتاه ترین فاصله ی s از m در G_T قرار می دهیم. حال اگر یک کوتاه ترین مسیر p بین s' و t را در G'_T محاسبه کنیم، می توان یک کوتاه ترین مسیر دیدارکننده ی T در P را از روی آن به دست آورد. برای این کار، فرض کنید m دومین رأس روی مسیر p باشد (m حتماً در M است). اگر یک کوتاه ترین مسیر بین s و m و یک کوتاه ترین مسیر بین m و t را در P به دنبال هم بچسبانیم، یک

^{۱۸}marked

کوتاه‌ترین مسیر دیدارکننده‌ی T در P خواهیم داشت.

اندازه‌ی گراف G_T و G'_T از همان مرتبه‌ی گراف G است. به همین خاطر به دست آوردن G'_T از روی G_T و دو بار اجرای الگوریتم دایکسترا روی این گراف‌ها برای محاسبه‌ی کوتاه‌ترین فاصله‌ها تا رأس‌های M و یافتن مسیر نهایی هزینه‌ی اضافی بر یک الگوریتم کوتاه‌ترین مسیر مبتنی بر گراف قابلیت دید تحمیل نمی‌کنند. چالش اصلی در حل این مسئله ساختن کارای گراف G_T از روی گراف G است. برای این کار باید دو دسته محاسبه‌ی زیر انجام شود:

(۱) تشخیص این که آیا پاره‌خط uv با T تقاطع دارد یا نه و در صورت مثبت بودن جواب، یافتن یک نقطه‌ی تقاطع

(۲) محاسبه‌ی $C(u, v)$ به ازای هر زوج رأس u و v در V .

بهترین زمان برای محاسبه‌ی اول، هم‌زمان با ساختن گراف قابلیت دید است. در آنجا می‌توان با انجام محاسبات به دو صورت، یکی با در نظر گرفتن T به عنوان یک مانع و یکی بدون آن تقاطع مورد نظر را در صورت وجود محاسبه کرد. برای قدم دوم، باید تمام جفت رأس‌های u و v مورد بررسی قرار بگیرند و برای هر یک نقاطی از T را که کوتاه‌ترین مجموع فاصله‌ها تا u و v دارند محاسبه کنیم و بین این نقاط، آنهایی را که هر دو از u و v قابل دید هستند جدا کنیم. برای این کار می‌توانیم تمام یال‌ها و رأس‌های T را بررسی کنیم و نقطه‌ی تماس بهینه با آن یال را محاسبه کنیم. این زمان در بدترین حالت $O(m)$ است و قدم دوم الگوریتم زمانی معادل $O(n^2 m)$ نیاز خواهد داشت. در اینجا نشان می‌دهیم که چگونه می‌توان این زمان را بهتر کرد.

اگر چندضلعی T محدب باشد، می‌توان نقطه‌ی بهینه متناظر با u و v را در زمان $O(\log m)$ به دست آورد. این کار بر مبنای این مشاهده صورت می‌گیرد که فقط یک یال از T مانند e هست که نقطه‌ی بهینه‌ی آن (که از تقاطع پاره‌خط واصل u به قرینه‌ی v نسبت به e به دست می‌آید) درون e قرار می‌گیرد. این نقطه برای تمام یال‌های سمت چپ e مثل e' سمت راست e' قرار می‌گیرد و

برای یال‌های سمت راست e برعکس است. این شبیه به خاصیتی است که در بخش ۱.۱.۴ برای قیف‌های ساخته شده روی پنجره‌های قابلیت دید مطالعه شد. بر مبنای این خاصیت می‌توان با یک جستجوی دودویی یال مورد نظر را به دست آورد. البته این ممکن است که هیچ یالی چنین خاصیتی نداشته باشد که در این صورت یکی از رأس‌های T که با همان جستجوی دودویی قابل محاسبه است، تک‌نقطه‌ی $C(u, v)$ را می‌سازد. البته این ممکن است که این نقطه به خاطر نداشتن قابلیت دید از u یا v در مجموعه‌ی یاد شده وارد نشود که تشخیص این نیز به سادگی ممکن است.

اگر چندضلعی T نامحدب باشد، روش بالا کار نمی‌کند. برای حل این مشکل می‌توان از زیرتقسیم بازتابی کمک گرفت و زمان $O(n^2 m)$ را پایین‌تر آورد. به این منظور، هر رأس u از V را که در نظر می‌گیریم، زیرتقسیم بازتابی $RS(u, T)$ را محاسبه می‌کنیم. این کار در زمان $O(m \log m)$ قابل انجام است. پس از آن به ازای تمام رأس‌های $w \in V$ می‌توانیم در زمان $O(\log m)$ با جایابی v در زیرتقسیم ساخته شده، یک نقطه از $C(u, v)$ را پیدا کنیم. به این ترتیب، زمان لازم برای محاسبه‌ی قدم دوم، به $O(nm \log m + n^2 \log m)$ تبدیل می‌شود که برای مقادیر بزرگ m نتیجه‌ی بهتری است.

نتیجه گیری

۱.۶ نتایج به دست آمده

در این پژوهش به بررسی مبانی نظری برخی مسائل برنامه‌ریزی حرکت با قید قابلیت دید پرداخته شده است. سه دسته‌ی عمده از مسائلی که به آن پرداخته شده است عبارتند از محاسبه‌ی کوتاه‌ترین مسیر با قید قابلیت دید نقطه، قابلیت دید نقطه‌ی پرس‌وجو و دیدار چندضلعی. پیچیدگی زمانی و حافظه‌ی اجرای مربوط به این مسئله‌ها به ترتیب در جدول‌های ۱، ۲ و ۳ آمده‌اند. در این جدول‌ها، n به عنوان پیچیدگی ورودی در نظر گرفته شده است که در قیدهای مربوط به قابلیت دید تعداد رأس‌های دامنه‌ی ورودی و در قید دیدار چندضلعی مجموع تعداد رأس‌های دامنه و چندضلعی هدف است. در جدول ۲، K پارامتری وابسته به چندضلعی ورودی است که در بدترین حالت از مرتبه‌ی $O(n^2)$ است ولی در خیلی از حالات واقعی از این مقدار کم‌تر است.

جدول ۱: خلاصه‌ی پیچیدگی مسئله‌ی کوتاه‌ترین مسیر با قید قابلیت دید نقطه

حافظه	زمان	دامنه
$O(n)$	$O(n)$	چندضلعی ساده
$O(n \log n)$	$O(n \log n)$	دامنه‌ی چندضلعی گونه
$O(n^2)$	$O(n^2)$	رویه‌ی چندوجهی گونه محدب
$O(n^3)$	$O(n^3)$	رویه‌ی چندوجهی گونه نامحدب

جدول ۲: خلاصه‌ی پیچیدگی مسئله‌ی کوتاه‌ترین مسیر با قید قابلیت دید نقطه‌ی پرس‌وجو

دامنه	زمان پیش‌پردازش	حافظه	زمان پرس‌وجو
چندضلعی ساده (مقصد باز)	$O(n^2)$	$O(n^2)$	$O(\log n)$
چندضلعی ساده (مقصد مشخص)	$O(n^2 + nK)$	$O(n^2 + nK)$	$O(\log n)$

جدول ۳: خلاصه‌ی پیچیدگی مسئله‌ی کوتاه‌ترین مسیر با قید دیدار چندضلعی

دامنه	زمان	حافظه
چندضلعی ساده با چندضلعی هدف محدب	$O(n)$	$O(n)$
چندضلعی ساده با چندضلعی هدف مقعر	$O(n \log n)$	$O(n \log n)$
دامنه‌ی چندضلعی گونه	$O(n \log n)$	$O(n \log n)$

ایده‌هایی که برای حل این مسائل استفاده شده‌اند، عبارتند از استفاده از بازه‌های بهینگی و دروازه برای گسسته کردن محدوده‌ی جستجو به دنبال نقطه‌ی تماس بهینه و محدود کردن تعداد این بازه‌ها با حذف بازه‌هایی که ملاقات آنها حتماً بعد از ملاقات بخش‌های دیگری از ناحیه‌ی هدف اتفاق می‌افتد. همچنین در مسئله‌ی کوتاه‌ترین مسیر با قید قابلیت دید نقطه‌ی پرس‌وجو در یک چندضلعی ساده، رفتار نقطه‌ی تماس بهینه روی یک پنجره که در حال دوران است بررسی شد که خم‌های بهینه را می‌سازند. بررسی خواص این خم‌ها نیز از نتایج به دست آمده در مطالعه‌ی این مسئله است. در نهایت، چارچوب بازتابش جبهه‌ی موج برای ساختن زیرتقسیم بازتابی و نقشه‌ی کوتاه‌ترین مسیر دیدارکننده‌ی یک چندضلعی ارائه شد. این الگو با به‌کارگیری روش شناخته‌شده‌ی انتشار جبهه‌ی موج کار می‌کند.

همه‌ی این ایده‌های مطرح شده می‌توانند به عنوان عناصر پایه‌ی مطالعه‌ی نظری مسائل پیچیده‌تر و عملی‌تر مربوط به برنامه‌ریزی حرکت به کار بیایند. مثال‌هایی از این مسائل در بخش آینده ذکر خواهد شد.

۲.۶ پژوهش‌های آینده و مسائل باز

از آنجا که مطالعه‌ی نظری مسائل برنامه‌ریزی حرکت از دید بهینه‌سازی طول مسیر موضوعی نسبتاً تازه به شمار می‌آید، نتایج به دست آمده در پژوهش حاضر تنها نقطه‌ی شروع این مطالعات محسوب می‌شوند. این نتایج به همراه ایده‌های مطرح شده قابل تعمیم برای حل تعدادی دیگر از مسائل هستند. برخی از این گونه مسائل را که می‌توان به عنوان ادامه‌ی این پژوهش انجام داد در ادامه ذکر می‌کنیم.

۱.۲.۶ گراف قابلیت دید و قید دیدار

یافتن کوتاه‌ترین مسیر بین دو نقطه با قید دیدار چندضلعی در دامنه‌های چندضلعی گونه با استفاده از گراف قابلیت دید: هدف این مسئله محاسبه‌ی مسیر یاد شده با استفاده از گراف قابلیت دید است. این گراف باید طوری تعمیم داده شود که بتوان با جستجو در آن کوتاه‌ترین مسیر دیدارکننده چندضلعی هدف را پیدا کرد. در ساخت این گراف می‌توان از ایده‌ی مطرح شده در بخش ۴.۵ استفاده کرد. نکته‌ی مهم این است که استفاده از گراف قابلیت دید کامل دامنه ما را به مرتبه‌ی زمانی بهتری از آنچه داریم نمی‌رساند. به همین خاطر باید از روش به‌کارگرفته شده در الگوریتم کاپور و دیگران [۵۸] استفاده کرد. این الگوریتم با بهره‌گیری از ساختاری به نام راهرو^۱ گراف قابلیت دید بخش کوچک‌تری از دامنه را می‌سازد که یافتن کوتاه‌ترین مسیر را در زمان $O(n + h^2 \log n)$ ممکن می‌کند که h تعداد حفره‌های موجود در دامنه است. اهمیت این الگوریتم این است که با تفکیک پارامترهای n و h وابستگی زمان الگوریتم را به تعداد رأس‌ها خطی کرده است.

^۱corridor

۲.۲.۶ کوتاه‌ترین مسیر بازرسی در دامنه‌های چندضلعی گونه

محاسبه‌ی کوتاه‌ترین مسیر بازرسی در دامنه‌های چندضلعی گونه: در این مسئله سعی بر این است که مسئله‌ای را که در فصل ۴ برای چندضلعی‌های ساده بررسی شد برای دامنه‌های چندضلعی گونه حل کند. تفاوت بنیادین این دو دامنه در مورد این مسئله از آنجا ناشی می‌شود که در چندضلعی ساده، مرز ناحیه‌ی قابلیت دید که ممکن است شامل نقطه‌ی تماس بهینه باشد، یک پنجره از چندضلعی است. به این ترتیب با یک جاروی شعاعی ساده می‌توان تمام پنجره‌های ممکن را پیش‌پردازش کرد. در مقابل، در دامنه‌ی چندضلعی گونه این مرز ساختار پیچیده‌تری دارد و از تعداد $O(n)$ پاره خط تشکیل شده است و در نتیجه به شکل قبل قابل پیش‌پردازش نیست. در صورتی که لازم شود چندضلعی قابلیت دید نقطه‌ی پرس‌وجو ساخته شود می‌توان از نتیجه‌ی ارائه شده توسط زارعی و قدسی [۷۹] استفاده کرد که بهترین الگوریتمی است که تا کنون برای این مسئله ارائه شده است.

تنها نتیجه‌ی مرتبط با این مسئله که تا کنون ارائه شده است، الگوریتم دانر^۲ و کاوراکی^۳ [۱۶] است که سعی در حل مسئله‌ی مسیریابی در دامنه‌های چندضلعی گونه و دامنه‌های چندوجهی گونه دارد. این مسئله در این دامنه‌ها NP-سخت است [۶]. الگوریتم مذکور به خاطر نزدیک شدن به حالت‌های واقعی دو محدودیت روی بینایی متحرک گذاشته است: یکی محدودیت فاصله تا متحرک و دیگری زاویه‌ی خط دید. مقاله‌ی یاد شده بدون ارائه‌ی کران مشخصی روی طول مسیر و مرتبه‌ی زمانی اجرای الگوریتم به ارائه‌ی نتایج تجربی اجرای الگوریتم بسنده کرده است.

۳.۲.۶ نزدیک‌ترین نقطه روی چندضلعی به k نقطه

محاسبه‌ی نقطه‌ای روی مرز یک چندضلعی با کم‌ترین مجموع فواصل ژئودزیک تا k نقطه: در دسته‌ای از مسائل بررسی شده در این پژوهش که به دنبال یافتن کوتاه‌ترین مسیر مقید بین دو نقطه

^۲Danner

^۳Kavraki

بودند، هدف اصلی یافتن کوتاه‌ترین مسیر بین مبدأ و مقصد بود. اما این مسئله را می‌توان به صورتی دیگر نیز نگاه کرد. به این ترتیب که خروجی مسئله نقطه‌ای از مرز ناحیه‌ی هدف (چندضلعی قابلیت دید یا چندضلعی هدف) است که مجموع فاصله‌های آن از دو نقطه که به عنوان مبدأ و مقصد داده شده‌اند کمینه است. برای تعمیم این مسئله فرض می‌کنیم به عنوان ورودی k نقطه و یک چندضلعی در دامنه داده شده‌اند و می‌خواهیم نقطه‌ای از مرز چندضلعی را پیدا کنیم که مجموع فاصله‌های ژئودزیک آن از نقاط داده شده کمینه باشد. گونه‌ی دیگری از مسئله می‌تواند به این صورت تعریف شود که طولانی‌ترین فاصله از نقطه‌ی مذکور به k نقطه کمینه باشد. مسئله‌ای که تا حدی به این مسئله شباهت دارد یافتن مرکز ژئودزیک^۴ یک چندضلعی ساده است. این مسئله در زمان $O(n \log^2 n)$ قابل حل است [۸۰، ۸۱]. این مسئله برای دامنه‌های چندضلعی گونه یک مسئله‌ی باز محسوب می‌شود. این مسئله در هر سه دامنه‌ی بررسی شده یعنی چندضلعی‌های ساده، دامنه‌های چندضلعی گونه و رویه‌های چندوجهی گونه قابل بررسی است. دو نکته در تعمیم الگوی استفاده شده در این پژوهش ($k = 2$) به مقادیر بالاتر k به چشم می‌خورند:

- نکته‌ی اول محاسبه‌ی بازه‌های بهینگی مربوط به k مبدأ است. به این ترتیب اگر $T(n)$ زمان لازم برای محاسبه‌ی بازه‌های مربوط به یک نقطه در دامنه‌ی مورد مطالعه باشد، تعمیم سراسر زمانی از مرتبه‌ی $kT(n)$ نیاز خواهد داشت در صورتی که ممکن است بتوان از این نکته که در تمام این k دور محاسبه شکل هندسی دامنه یکسان است استفاده کرد و با پرداخت هزینه‌ی پیش‌پردازش، هزینه‌ی کل را برای مقادیر نسبتاً بزرگ k پایین آورد.
- نکته‌ی دیگر این است که بازه‌های بهینگی متناظر با هر یک از k نقطه در بدترین حالت از مرتبه‌ی $O(mn)$ خواهد بود که در آن پیچیدگی چندضلعی داخلی و n پیچیدگی دامنه است. حتی اگر بتوان با مفهومی شبیه به دروازه این اندازه را به $O(m+n)$ کاهش داد، از آنجا که این بازه‌ها با هم اشتراک ندارند، ملزم به بررسی $O(kmn)$ بازه خواهیم بود. یکی از

^۴ geodesic center

چالش‌های حل این مسئله ابداع روشی برای کم کردن این تعداد است.

۴.۲.۶ دیدار زیردنباله‌ها

تعمیم دادن قید دیدار به دیدن بیش از یک ناحیه با ترتیب دلخواه، مسئله را به رده‌ی NP-سخت می‌برد (مسئله‌ی فروشنده‌ی دوره‌گرد با همسایگی). اگر ترتیب دیدار ناحیه‌ها را از پیش داده شده فرض کنیم، برای ناحیه‌های محدب توسط [۲۷] می‌توان مسئله را در زمان چندجمله‌ای حل کرد. در همان مقاله نشان داده شده که برای ناحیه‌های مقعر این مسئله NP-سخت است. حالت‌های محدودتر دیدار چند ناحیه با ترتیب مشخص را می‌توان در مسائل نگهبان باغ‌وحش، سفر سیاحتی و برش قطعات دید. برای تمام این مسائل می‌توان گونه‌ی پرس‌وجویی تعریف کرد که در آن زیردنباله‌ای از دنباله ناحیه‌ها داده می‌شوند و مسیر مقید به دیدار ناحیه‌های آن زیردنباله با ترتیب است. می‌خواهیم دامنه را طوری پیش‌پردازش کنیم که با دریافت یک زیردنباله بتوان این مسیر را به طرز کارایی محاسبه کرد.

۵.۲.۶ قابلیت دید هدف متحرک

تعمیم دیگری که روی قیدهای قابلیت دید مطرح شده در این رساله می‌توان در نظر گرفت، متحرک بودن هدف دید است. در گونه‌ی اول مسئله، می‌خواهیم از یک نقطه‌ی مبدأ به یک نقطه‌ی مقصد برویم در حالی که در طول مسیر یک هدف متحرک را نیز ببینیم. هدف بهینه‌سازی نیز کمینه کردن طول مسیر است. در گونه‌ی دوم که هدف دید را در قالب پرس‌وجو دریافت می‌کنیم، حالت‌های مقصد باز و مقصد مشخص مسئله نیز قابل تعریف هستند.

زمان بعد بسیار مهمی از این تعمیم را تشکیل می‌دهد، به این ترتیب که سرعت نسبی حرکت هدف دید و تعقیب کننده در راه حل تأثیر می‌گذارند. همین‌طور، ثابت یا متغیر بودن اندازه‌ی سرعت این دو متحرک نیز می‌تواند گونه‌های متفاوتی از مسئله را ایجاد کند. حل این مسئله در

حالتی که هدف دید حرکت یکنواخت در خط راست دارد و سرعت تعقیب‌کننده نیز ثابت است می‌تواند نقطه‌ی شروعی برای بررسی حالت‌های پیچیده‌تر باشد.

واژه‌نامه‌ی فارسی به انگلیسی

reflex	بازتابش
reflector	بازتابنده
reflective	بازتابی
interval	بازه
interval of optimality	بازه‌ی بهینگی
hidden surface removal	برداشتن رویه‌ی پنهان
online	برخط
planner	برنامه‌ریز
planning	برنامه‌ریزی
unfold	بی‌تا کردن
unfolding	بی‌تاشده
extension segment	پاره‌خط امتداد
ray	پرتو
query	پرس‌وجو
window	پنجره

convex hull	پوش محدب
preprocess	پیش پردازش
traversal	پیمایش
link	پیوند
subdivision	تظریف
reflective subdivision	تظریف بازتابی
conforming subdivision	تظریف منطبق
pursuit	تعقیب
contact	تماس
sweep	جارو
radial sweep	جاروی شعاعی
point location	جایابی نقطه
wavefront	جبهه‌ی موج
pocket	جیب
polygon	چندضلعی
visibility polygon	چندضلعی قابلیت دید
polygonal	چندضلعی گونه
polyhedra	چندوجهی
polyhedral	چندوجهی گونه
layout	چیدمان
incident	حادث
hole	حفره

curve	خم
autonomous	خودمختار
data structure	داده ساختار
domain	دامنه
polygonal domain	دامنه‌ی چندضلعی گونه
polyhedral domain	دامنه‌ی چندوجهی گونه
continuous Dijkstra	دایکسترای پیوسته
spanning tree	درخت فراگیر
shortest path tree	درخت کوتاه‌ترین مسیر
gate	دروازه
sequence	دنباله
vertex	رأس
reflex vertex	رأس انعکاسی
intractable	رام‌نشدنی
inward	روبه‌داخل
surface	رویه
overlay	روی هم‌نشانی
chain	زنجیره
subpath	زیرمسیر
geodesic	ژئودزیک
combinatorial structure	ساختار ترکیبیاتی
start-shaped	ستاره‌شکل

amortized	سرشکن شده
pseudo-source	شبه‌مبدأ
bend	شکست
ray shooting	شلیک پرتو
marked	علامت‌دار
depth-first	عمق-اول
free space	فضای آزاد
configuration space	فضای پیکربندی
visible	قابل دید
visibility	قابلیت دید
disc	قرص
funnel	قیف
focus	کانون
node	گره
evasion	گریز
obstacle	مانع
triangulation	مثلث‌بندی
agent	مأمور
edge-adjacent	مجاور یالی
site	محل
local	محلی
boundary	مرز

geodesic center	مرکز ژئودزیک
path	مسیر
observer	مشاهده‌کننده
metric	معیار
bisector	منصف
wavelet	موجک
invisible region	ناحیه‌ی غیرقابل دید
reflex region	ناحیه‌ی بازتابش
reflective region	ناحیه‌ی بازتابی
map	نقشه
shortest path map	نقشه‌ی کوتاه‌ترین مسیر
quad view	نمای چهارگانه
art gallery	نمایشگاه نقاشی
Voronoi diagram	نمودار وورونوی
chord	وتر
target	هدف دید
hyperbolic	هذلولوی
kernel	هسته
connected	همبند
multiply-connected	همبند چندگانه
simply-connected	همبند ساده
overlap	هم‌پوشانی

colinear هم خط

واژه‌نامه‌ی انگلیسی به فارسی

agent	مأمور
art gallery	نمایشگاه نقاشی
amortized	سرشکن شده
autonomous	خودمختار
bend	شکست
bisector	منصف
boundary	مرز
chain	زنجیره
chord	وتر
colinear	هم خط
combinatorial structure	ساختار ترکیبیاتی
configuration space	فضای پیکربندی
contact	تماس
continuous Dijkstra	دایکسترای پیوسته
conforming subdivision	تظریف منطبق

connected	همبند
convex hull	پوش محدب
curve	خم
data structure	داده ساختار
domain	دامنه
depth-first	عمق-اول
disc	قرص
edge-adjacent	مجاور یالی
evasion	گریز
extension segment	پاره خط امتداد
focus	کانون
free space	فضای آزاد
funnel	قیف
gate	دروازه
geodesic	ژئودزیک
geodesic center	مرکز ژئودزیک
hidden surface removal	برداشتن رویه‌ی پنهان
hole	حفره
hyperbolic	هذلولوی
incident	حادث
interval	بازه
interval of optimality	بازه‌ی بهینگی

intractable	رام‌نشدنی
inward	روبه‌داخل
kernel	هسته
layout	چیدمان
link	پیوند
local	محلی
marked	علامت‌دار
map	نقشه
metric	معیار
multiply-connected	همبند چندگانه
node	گره
observer	مشاهده‌کننده
obstacle	مانع
online	برخط
overlap	هم‌پوشانی
overlay	روی‌هم‌نشانی
path	مسیر
planner	برنامه‌ریز
planning	برنامه‌ریزی
pocket	جیب
point location	جایابی نقطه
polygon	چندضلعی

polygonal.....	چندضلعی گونه
polygonal domain.....	دامنه‌ی چندضلعی گونه
polyhedra	چندوجهی
polyhedral.....	چندوجهی گونه
polyhedral domain.....	دامنه‌ی چندوجهی گونه
preprocess	پیش پردازش
pseudo-source	شبه مبدأ
pursuit	تعقیب
quad view	نمای چهارگانه
query.....	پرس و جو
radial sweep	جاروی شعاعی
ray.....	پرتو
ray shooting	شلیک پرتو
reflective	بازتابی
reflective region.....	ناحیه‌ی بازتابی
reflective subdivision.....	تظریف بازتابی
reflector	بازتابنده
reflex	بازتابش
reflex region	ناحیه‌ی بازتابش
reflex vertex.....	رأس انعکاسی
sequence	دنباله
shortest path map.....	نقشه‌ی کوتاه‌ترین مسیر

shortest path tree	درخت کوتاه‌ترین مسیر
site	محل
spanning tree	درخت فراگیر
simply-connected	همبند ساده
star-shaped	ستاره‌شکل
subdivision	تظریف
subpath	زیرمسیر
surface	رویه
sweep	جارو
target	هدف دید
traversal	پیمایش
triangulation	مثلث‌بندی
unfold	بی‌تا کردن
unfolding	بی‌تاشده
vertex	رأس
visibility	قابلیت دید
visibility polygon	چندضلعی قابلیت دید
visible	قابل دید
Voronoi diagram	نمودار وورونوی
wavefront	جبهه‌ی موج
wavelet	موجک
window	پنجره

کتاب نامه

- [1] J.-C. Latombe. Motion planning: a journey of robots, molecules, digital actors, and other artifacts. *International Journal of Robotics Research*, 18:1119–1128, 1999.
- [2] J. H. Reif. Complexity of the mover’s problem and generalizations. In *Proc. 20th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 421–427, 1979.
- [3] J. Canny and J. H. Reif. New lower bound techniques for robot motion planning problems. In *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 49–60, 1987.
- [4] T. Lozano-Pérez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, 1979.
- [5] J. S. B. Mitchell. Geometric shortest paths and network optimization. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.
- [6] W. Chin and S. Ntafos. Optimum watchman routes. *Inform. Process. Lett.*, 28:39–44, 1988.

- [7] W.-P. Chin and S. Ntafos. Watchman routes in simple polygons. *Discrete Comput. Geom.*, 6(1):9–31, 1991.
- [8] S. Ntafos. Watchman routes under limited visibility. *Comput. Geom. Theory Appl.*, 1(3):149–170, 1992.
- [9] X. H. Tan, T. Hirata, and Y. Inagaki. An incremental algorithm for constructing shortest watchman routes. *Internat. J. Comput. Geom. Appl.*, 3(4):351–365, 1993.
- [10] X. Tan. Approximation algorithms for the watchman route and zookeeper’s problems. *Discrete Appl. Math.*, 136(2-3):363–376, 2004.
- [11] A. Efrat, L. J. Guibas, S. Har-Peled, D. C. Lin, J. S. Mitchell, and T. M. Murali. Sweeping simple polygons with a chain of guards. In *Proc. 11th ACM-SIAM Sympos. Discrete Algorithms (SODA ’2000)*, pages 927–936, 2000.
- [12] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9(4/5):471–, 1999.
- [13] G. Rote. Pursuit-evasion with imprecise target location. In *SODA ’03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 747–753, 2003.
- [14] J.-H. Lee, S. Y. Shin, and K.-Y. Chwa. Visibility-based pursuit-evasion in a polygonal room with a door. In *SCG ’99: Proceedings of the fifteenth annual symposium on Computational geometry*, pages 281–290, New York, NY, 1999. ACM Press.

- [15] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion with limited visibility. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1060–1069, 2004.
- [16] T. Danner and L. Kavraki. Randomized planning for short inspection paths. In *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000.
- [17] C. Knauer and G. Rote. Shortest inspection-path queries in simple polygons. Technical Report B-05-05, Institut für Informatik, Freie Universität Berlin, 2005.
- [18] S. M. LaValle, D. Lin, L. J. Guibas, J.-C. Latombe, and R. Motwani. Finding an unpredictable target in a workspace with obstacles. In *Proc. IEEE Internat. Conf. Robot. Autom.*, pages 737–742, 1997.
- [19] S. M. LaValle, H. H. González-Ba nos, C. Becker, and J.-C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Proc. IEEE International Conference on Robotics and Automation*, pages 731–736, 1997.
- [20] S. M. LaValle, B. Simov, and G. Slutzki. An algorithm for searching a polygonal region with a flashlight. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 260–269, 2000.
- [21] H. H. González-Ba nos, L. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, R. Motwani, and C. Tomasi. Motion planning with visibility constraints: Building autonomous observers. In *Proc. Eighth Int'l Symp. on Robotics Research*, pages 95–101. Springer-Verlag, Berlin, 1998.

- [22] Benjamín Tovar, Steven M. LaValle, and Rafael Murrieta-Cid. Optimal navigation and object finding without geometric maps or localization. In *ICRA*, pages 464–470. IEEE, 2003.
- [23] W.-P. Chin and S. Ntafos. The zookeeper route problem. *Inf. Sci.*, 63(3):245–259, 1992.
- [24] X. Tan. Shortest zookeeper’s routes in simple polygons. *Inf. Process. Lett.*, 77(1):23–26, 2001.
- [25] X. Tan and T. Hirata. Finding shortest safari routes in simple polygons. *Inf. Process. Lett.*, 87(4):179–186, 2003.
- [26] M. Dror. Polygon plate-cutting with a given order. *IIE Transactions*, 31(3):271–274, 1999.
- [27] M. Dror, A. Efrat, A. Lubiw, and J. S. B. Mitchell. Touring a sequence of polygons. In *Proc. 35th ACM Sympos. Theory Comput.*, 2003.
- [28] A. Dumitrescu and J. S. B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. In *Symposium on Discrete Algorithms*, pages 38–46, 2001.
- [29] J. Gudmundsson and C. Levcopoulos. A fast approximation algorithm for TSP with neighborhoods and red-blue separation. *Lecture Notes in Computer Science*, 1627:473–482, 1999.

- [30] J. Gudmundsson and C. Levcopoulos. Hardness result for TSP with neighborhoods. Technical Report LU-CS-TR:2000-216, Department of Computer Science, Lund University, Sweden, 2000.
- [31] J. Barraquand, L. E. Kavraki, , J.-C. Latombe, T.Y. Li, R. Motwani, and P. Raghavan. A random sampling scheme for path planning. *International Journal of Robotics Research*, 16(6):759–774, 1997.
- [32] L. J. Guibas, L. Ramshaw, and J. Stolfi. A kinetic framework for computational geometry. In *Proc. 24th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 100–111, 1983.
- [33] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12:566–580, 1996.
- [34] L. Aleksandrov, M. Lanthier, A. Maheshwari, , and J.-R. Sack. An ϵ -approximation algorithm for weighted shortest path queries on polyhedral surfaces. In *14th European Workshop on Computational Geometry*, 1998.
- [35] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Appl. Math.*, 55:197–218, 1994.
- [36] J. Choi, J. Sellen, and C. K. Yap. Approximate Euclidean shortest path in 3-space. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 41–48, 1994.

- [37] C. Mata and J. S. Mitchell. Approximation algorithms for geometric tour and network design problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 360–369, 1995.
- [38] K. Varadarajan and P. K. Agarwal. Approximating shortest paths on a polyhedron. In *Proc. 38th Annu. IEEE Sympos. Found. Comput. Sci.*, 1997.
- [39] J. O’Rourke. *Art Gallery Theorems and Algorithms*. The International Series of Monographs on Computer Science. Oxford University Press, New York, NY, 1987.
- [40] R. Khosravi and M. Ghodsi. Shortest paths with single-point visibility constraints. *Scientia Iranica*, 2004. accepted for publication.
- [41] R. Khosravi, M. Ghodsi, and M. Taghdiri. Shortest point-visible paths on polyhedral surfaces. In *Proc. of the 10th International Conference on Computing and Information (ICCI’2000)*, 2000.
- [42] R. Khosravi and M. Ghodsi. The fastest way to view a query point in simple polygons. In *Proceedings of the 21st European Workshop on Computational Geometry*, pages 187–190, 2005.
- [43] R. Khosravi and M. Ghodsi. Query-point visibility constrained shortest paths in simple polygons. *Theoretical Computer Science*, 2005. submitted.
- [44] R. Khosravi and M. Ghodsi. Shortest paths in simple polygons with polygon-meet constraints. *Information Processing Letters*, 91:171–176, 2004.

- [45] R. Khosravi and M. Ghodsi. Shortest paths in polygonal domains with polygon-meet constraints. In *Proc. 19th European Workshop Comput. Geom.*, pages 137–142, 2003.
- [46] R. Khosravi and M. Ghodsi. Visiting a polygon on the optimal way to a query point. *Information Processing Letters*, 2005. submitted.
- [47] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.
- [48] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, NY, 1985.
- [49] R. Bar-Yehuda and Bernard Chazelle. Triangulating disjoint Jordan chains. *Internat. J. Comput. Geom. Appl.*, 4(4):475–481, 1994.
- [50] D. G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. Comput.*, 12(1):28–35, 1983.
- [51] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16:647–668, 1987.
- [52] Bernard Chazelle. A theorem on polygon cutting with applications. In *Proc. 23rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 339–349, 1982.
- [53] D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.

- [54] L. J. Guibas, J. Hershberger, D. Leven, M. Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.
- [55] L. J. Guibas, E. McCreight, M. Plass, and J. Roberts. A new representation for linear lists. In *Proc. 9th Annu. ACM Sympos. Theory Comput.*, pages 49–60, 1977.
- [56] L. J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39(2):126–152, October 1989.
- [57] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete Comput. Geom.*, 16:419–453, December 1996.
- [58] S. Kapoor, S. N. Maheshwari, and J. S. Mitchell. An efficient algorithm for euclidean shortest paths among polygonal obstacles in the plane. *Discrete Comput. Geom.*, 18:377–383, 1997.
- [59] J. S. B. Mitchell. Shortest paths among obstacles in the plane. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 308–317, 1993.
- [60] J. S. B. Mitchell. Shortest paths among obstacles in the plane. *Internat. J. Comput. Geom. Appl.*, 6:309–332, 1996.
- [61] J. Hershberger and S. Suri. Efficient computation of Euclidean shortest paths in the plane. In *Proc. 34th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 508–517, 1993.
- [62] J. Hershberger and S. Suri. An optimal algorithm for Euclidean shortest paths in the plane. Manuscript, Washington University, 1995.

- [63] M. Sharir and A. Schorr. On shortest paths in polyhedral spaces. *SIAM J. Comput.*, 15:193–215, 1986.
- [64] J. Chen and Y. Han. Shortest paths on a polyhedron. *Internat. J. Comput. Geom. Appl.*, 6:127–144, 1996.
- [65] S. Kapoor. Efficient computation of geodesic shortest paths. In *Proc. 32th Annu. ACM Sympos. Theory Comput.*, pages 770–779, 1999.
- [66] B. Joe and R. B. Simpson. Correction to Lee’s visibility polygon algorithm. *BIT*, 27:458–473, 1987.
- [67] D. T. Lee. Visibility of a simple polygon. *Comput. Vision Graph. Image Process.*, 22:207–221, 1983.
- [68] P. J. Heffernan and J. S. B. Mitchell. An optimal algorithm for computing visibility in the plane. *SIAM J. Comput.*, 24(1):184–201, 1995.
- [69] M. de Berg, D. Halperin, M. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray shooting and hidden surface removal. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 21–30, 1991.
- [70] M. McKenna. Worst-case optimal hidden-surface removal. *ACM Trans. Graph.*, 6:19–28, 1987.
- [71] M. Overmars and Micha Sharir. Output-sensitive hidden surface removal. In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 598–603, 1989.

- [72] A. Schmitt. Time and space bounds for hidden line and hidden surface algorithms. In *Proc. Eurographics 81*, pages 43–56, Amsterdam, Netherlands, 1981. North-Holland.
- [73] John Hershberger and Subhash Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28(6):2215–2256, 1999.
- [74] P. K. Agarwal, B. Aronov, J. O’Rourke, and C. Schevon. Star unfolding of a polytope with applications. Technical Report 031, Dept. Comput. Sci., Smith College, Northampton, MA, July 1993.
- [75] U. Finke and K. Hinrichs. Overlaying simply connected planar subdivisions in linear time. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 119–126, 1995.
- [76] U. Finke and K. H. Hinrichs. The quad view data structure: a representation for planar subdivisions. In *Proc. 6th Sympos. Advances in Spatial Databases*, number 951 in Lecture Notes Comput. Sci., pages 29–46, 1995.
- [77] S. J. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987.
- [78] D. T. Lee and R. L. Drysdale, III. Generalization of Voronoi diagrams in the plane. *SIAM J. Comput.*, 10:73–87, 1981.
- [79] A. Zarei and M. Ghodsi. Efficient computation of query point visibility in polygons with holes. In *SCG ’05: Proceedings of the twenty-first annual symposium on Computational geometry*, pages 314–320, New York, NY, USA, 2005. ACM Press.

- [80] Te. Asano and G. Toussaint. Computing the geodesic center of a simple polygon. In D. S. Johnson, editor, *Discrete Algorithms and Complexity*, Perspectives in Computing, pages 65–79. Academic Press, 1987.
- [81] R. Pollack, Micha Sharir, and Günter Rote. Computing of the geodesic center of a simple polygon. *Discrete Comput. Geom.*, 4:611–626, 1989.

Motion Planning with Visibility Constraints

Abstract

In this thesis, the problem of finding shortest path is studied under various forms of visibility constraints. The geometric domains considered are simple polygons and polygonal domains in two dimensions, and polyhedral surfaces (convex and non-convex) in three dimensions. In all of these domains, the problem of computing a shortest path between two points having a point visible from a point target is studied and efficient algorithms are provided. Also, preprocessing of the domain such that upon receiving the destination point as a query, the path with the desired property can be computed efficiently is considered. Another variation of the problem in which the target is given as a query is also studied.

The extension of the single-point visibility constraint to polygon-visit constraint is another problem studied in this thesis. Here, the path is constrained to have non-empty intersection with a certain region of the domain. This constraint is more general than visibility constraint, since the latter can be viewed as visiting the visibility polygon of the target. This problem is studied in simple polygons and polygonal domains with query version also considered.

To solve the above problems, a number of new methods and techniques are invented. The notion of interval of optimality is used to partition the target region according to the combinatorial structure of the shortest paths to the points in the region and plays a basic

role in all variations of the problem. Also, the wavefront reflection technique which is an extension of the standard wavefront propagation or continuous Dijkstra paradigm is used to construct the subdivisions needed for the query versions of the problem. The method used for solving query-point visibility problems is radial sweep in which the behavior of the optimal contact point is considered during a rotation transformation around a reflex vertex.

Keywords: *computational geometry, shortest paths, visibility, wavefront propagation*



Sharif University of Technology

Computer Engineering Department

Ph.D. Thesis

Motion Planning with Visibility Constraints

:

Ramtin Khosravi

سر :

Prof. Mohammad Ghodsi

November 2005