

الحمد لله
الذي هدانا لهذا
الذي كنا لنهتدي لولا
هدى الله لنا



دانشکده علوم ریاضی

رشته ریاضی کاربردی، گرایش تحقیق در عملیات (بهینه سازی)

پایان نامه کارشناسی ارشد

نمودار ورونوی و کاربردهای آن در مسائل بهینه سازی

نگارنده: پردیس رحمتی

استاد راهنما

دکتر جعفر فتحعلی

استادان مشاور

دکتر ابوالفضل پورعیدی

دکتر میثم علیشاهی

شهریور ۱۳۹۸

شماره:
تاریخ:

باسمه تعالی



مدیریت تحصیلات تکمیلی

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خاتم: پردیس رحمتی
با شماره دانشجویی: ۹۵۳۴۳۷۴ رشته: ریاضی کاربردی گرایش: بهینه سازی
تحت عنوان: نمودار ورونوی و کاربردهای آن در مسائل بهینه سازی که در تاریخ ۹۸/۹/۱۲ با حضور هیأت
محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

☐ قبول (با درجه: ضعیف، خوب، ...) ☐ مردود ☐ عملی ☒ نظری: نوع تحقیق:			
اعضای	مرتبه علمی	نام و نام خانوادگی	عضو هیأت داوران
	دانشیار	دکتر جعفر فتحعلی	۱- استادیار اهنمای اول
	استادیار	دکتر ابوالفضل پورمحمدی	۲- استادمشاور اول
	دانشیار	دکتر میثم علیشاهی	۳- استاد مشاور دوم
	استادیار	دکتر سمیه مغری	۴- نماینده تحصیلات تکمیلی
	استادیار	دکتر مهرداد غزنوی	۵- استاد سمین اول
	دانشیار	دکتر علیرضا ناظمی	۶- استاد سمین دوم



نام و نام خانوادگی: دکتر ابوالفضل فتحعلی
تاریخ و امضاء: ۹۸/۹/۱۲

تصدیق بر صورتی که کسی مردود شود حداقل یکبار دیگر (در مدت مجاز تحصیل) می تواند از پایان نامه خود دفاع نماید (دفاع مجدد نباید زودتر از ۴ ماه برگزار شود).

تقدیم به مقدس ترین واژه‌ها در لغت نامه دلم
”مادر مهربانم”
که زندگیم را مدیون مهر و عطوفتش میدانم
پدرم مهربانی بردبار مشفق و حامی
خواهران و برادران همراهان همیشگی و پشتوانه‌های زندگیم

سپاس‌گزاری...

حمد بی پایان خدای علیم را که مجدد توفیق کسب ذره‌ای دیگر از خزانه علم بی پایانش را عطا فرمود.

حال که به یاری پرورگار این کار به پایان رسید بر خود لازم میدانم از تمام عزیزانی که در این راه به من کمک کرده‌اند نهایت سپاس‌گزاری را به عمل آورم: در آغاز از استاد راهنمای گرانقدر آقای دکتر جعفر فتحعلی که راهنمایی این پایان نامه را به عهده داشتند، تشکر و سپاس ویژه دارم که استاد علم و اخلاق بودند .

همچنین از اساتید مشاور آقای دکتر ابوالفضل پورعیدی و آقای دکتر میثم علیشاهی کمال تشکر و قدردانی را دارم . از داوران گرامی آقایان دکتر مهرداد غزنوی و دکتر علیرضا ناظمی که زحمت داوری و تصحیح این پایان‌نامه را به عهده داشتند، کمال سپاس را دارم. خالصانه از اساتید دوره کارشناسی خانم دکتر زهرا براتی سده، آقای دکتر علیرضا موذن، خانم دکتر زیبا ارجمندزاده، آقای دکتر عمران توحیدی و همچنین معلم اول ابتدایی سرکار خانم فروغ کیخاه و سایر معلمان و مدرسانی که در مقاطع مختلف تحصیلی به من علم آموختند، متشکرم. و در پایان از کلیه همراهان عزیز و دوستان خوبم نهایت سپاس را دارم.

پردیس رحمتی

شهریور ۱۳۹۸

تعهد نامه

اینجانب پردیس رحمتی دانشجوی کارشناسی ارشد رشته ریاضی کاربردی علوم ریاضی دانشگاه شاهرود، نویسنده پایان نامه با عنوان نمودار ورونوی و کاربردهای آن در مسائل بهینه سازی، تحت راهنمایی جعفر فتحعلی متعهد می شوم:

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش های دیگر پژوهشگران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان نامه، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه صنعتی شاهرود تعلق دارد، و مقالات مستخرج با نام “دانشگاه صنعتی شاهرود” یا “Shahrood University of Technology” به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آوردن نتایج اصلی پایان نامه تاثیرگذار بوده اند، در مقالات مستخرج از پایان نامه رعایت می گردد.
- در تمام مراحل انجام این پایان نامه، در مواردی که از موجود زنده (یا بافت های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده شده است)، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

پردیس رحمتی
شهریور ۱۳۹۸

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه های رایانه ای، نرم افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان نامه بدون ذکر منبع مجاز نمی باشد.

چکیده

نمودار ورونوی ساختاری جالب توجه در هندسه محاسباتی است که کاربردهای زیادی در سایر علوم نیز دارد. این نمودار روشی برای تقسیم صفحه به تعدادی ناحیه است بطوریکه این نواحی دارای شرایط خاصی هستند و محدودیت‌هایی را برآورده می‌کنند. در این پایان نامه پس از معرفی نمودار ورونوی پیچیدگی این نمودار روی گراف‌ها را توصیف می‌کنیم و سپس روش‌های یافتن گراف نمودار ورونوی روی درخت‌ها و گراف‌ها را بیان می‌کنیم. برخی از کاربردهای نمودار ورونوی نیز در این پایان‌نامه ذکر شده است.

کلمات کلیدی: نمودار ورونوی، گراف، درخت، پیچیدگی

فهرست مطالب

ف	فهرست تصاویر
ق	فهرست جداول
۱	۱ مفاهیم اولیه نمودار ورونوی
۳	۱.۱ تعاریف و اصطلاحات گرافی
۶	۲.۱ تعاریف و مفاهیم مربوط به نمودار ورونوی
۱۷	۱.۲.۱ محاسبه نمودار ورونوی
۲۵	۲ نمودار ورونوی روی گرافها
۲۷	۱.۲ نمودار ورونوی روی درختان (VD_T)
۳۲	۱.۱.۲ نمودارهای ورونوی وزن دار روی درختان (WVD_T)
۳۵	۲.۲ نمودار ورونوی دورترین رنگ ($FCVD$)
۳۶	۱.۲.۲ دورترین رنگ نمودار ورونوی وزندار
۳۶	۳.۲ نمودار ورونوی روی گرافها (VD_G)
۳۷	۱.۳.۲ نمودار ورونوی وزن دار روی گرافها (WVD_G)
۳۹	۳ نمودار ورونوی گراف و کاربردهای آن
۴۲	۱.۳ خواص اولیه
۴۴	۲.۳ محاسبه نمودارهای ورونوی گراف
۴۴	۱.۲.۳ الگوریتم دایکسترای موازی
۴۷	۳.۳ بروزرسانی نمودارهای ورونوی
۴۸	۴.۳ کاربردها
۵۳	مراجع

فهرست تصاویر

۱۰.۱	اجزای نمودار ورونوی	۲
۲.۱	نمودار ورونوی فروشگاه‌ها در بخشی از شهر تهران	۹
۳.۱	سلول ورونوی	۱۰
۴.۱	یک نمودار ورونوی	۱۰
۵.۱	اضافه کردن راسی در بینهایت	۱۲
۶.۱	بزرگترین دایره خالی	۱۳
۷.۱	قرار گرفتن q روی یال یا راسی از سلول ورونوی	۱۵
۸.۱	پوش بالایی که با رنگ قرمز مشخص شده و پوش پایینی که با رنگ سبز مشخص شده است.	۱۶
۹.۱	خط ساحلی	۱۸
۱۰.۱	ظهور کمان بر روی خط ساحلی	۱۹
۱۱.۱	رویداد دایره	۲۰
۱۲.۱	حذف کمان از خط ساحلی	۲۰
۱۳.۱	درخت جست و جوی دودویی نشان دهنده خط ساحلی	۲۱
۱۴.۱	صف رویدادها	۲۲
۱۵.۱	فاصله یک نقطه از شکل	۲۲
۱۶.۱	نمودار ورونوی پاره‌خط‌ها به کمک الگوریتم جاروب	۲۳
۱۷.۱	نمونه‌ای از نمودار ورونوی پاره‌خط‌ها	۲۴
۱.۲	وارد کردن درخت و مرتب کردن سایت‌ها	۲۸
۲.۲	حذف سایت‌های درونی	۲۸
۳.۲	مشخص کردن ماکسیمال زیر درخت‌هایی که حاوی سایت نیستند. (u_i ها)	۲۹
۴.۲	مشخص کردن گره‌های آویزان برای u_i ها یعنی $h(u_i)$ ها	۳۰
۵.۲	حذف ماکسیمال زیر درخت‌هایی که حاوی سایت نیستند یعنی حذف u_i ها	۳۰
۶.۲	جایگزینی رئوس درجه دو با یک تک یال	۳۱

۷.۲	نمودار ورونوی وزنی در یک خط معادل با پوش پایینی توابع فاصله است، می توانیم وزن‌ها و سایت‌ها را پیدا کرده بنابراین n تابع فاصله خطی $n - 1$ منصف خواهد داشت.	۳۳
۸.۲	تابع فاصله‌ای از S توسط تابع فاصله‌ای از p و q با وزن‌های پایین‌تر مسدود شده است.	۳۴
۱.۳	نمودار ورونوی درونی و بیرونی برای گره‌های v و w	۴۱
۲.۳	نشان دادن عینی اثبات لم فوق	۴۴

فهرست جداول

۱۰۲	خلاصه‌ای از نتایج	۳۷
-----	-------------------	----

فصل ۱

مفاهیم اولیه نمودار ورونوی

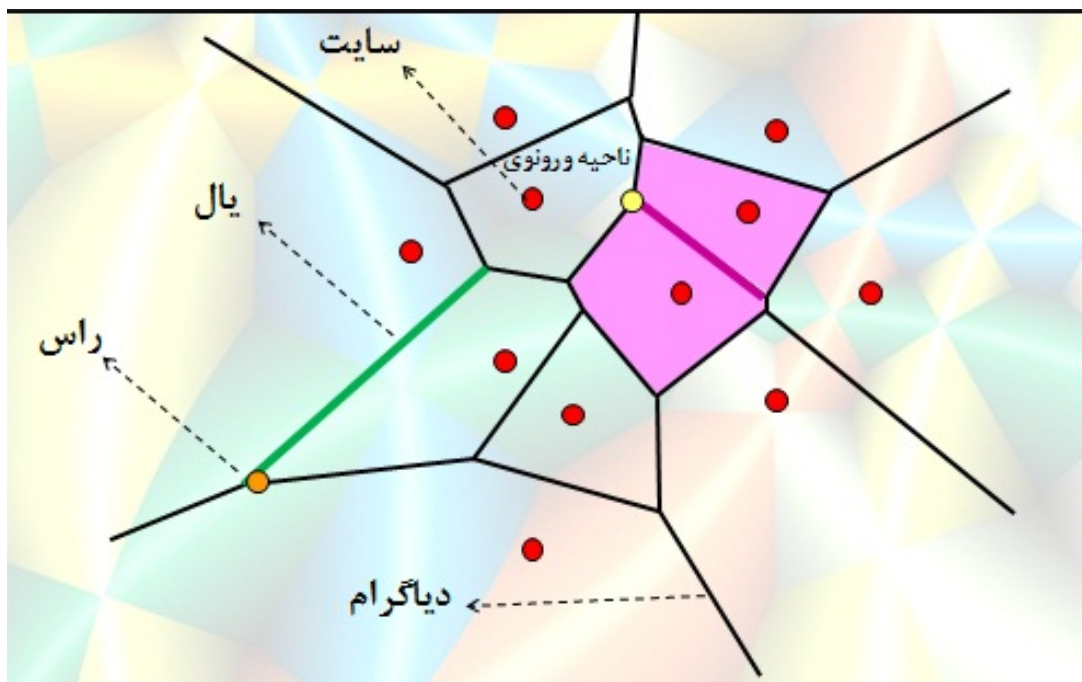
مقدمه

در این فصل به بیان مفاهیم اولیه مربوط به نمودارهای ورونوی می‌پردازیم. (تمامی قضایای این فصل برگرفته از مرجع [۱] می‌باشد.)

تاریخچه نمودار ورونوی : در علم ریاضیات نمودار ورونوی روشی برای تقسیم صفحه به تعدادی ناحیه می‌باشد. در این نمودار به هر مجموعه‌ای از نقاط (که سایت^۱ یا دامنه می‌نامیم) ناحیه‌ای اختصاص داده می‌شود که سلول ورونوی^۲ نامیده می‌شود. این نمودار صفحه‌ای با n نقطه را به چند ضلعی‌های محدب تقسیم می‌کند به این صورت که هر چندضلعی دقیقا شامل یک سایت است و هر نقطه در یک چندضلعی خاص به سایت نزدیک‌تر از هر نقطه دیگر است. (شکل ۱.۱ را ببینید.)

^۱ Site

^۲ Voronoi Cell



شکل ۱۰۱: اجزای نمودار ورونوی

این نمودار به افتخار دیریکله^۳ به نام موزاییک‌کاری دیریکله^۴ و بعد از ورونوی به نام موزاییک‌کاری ورونوی^۵ و یا تجزیه ورونوی شناخته می‌شود. استفاده غیررسمی از این نمودار به سال ۱۶۴۴ و توسط دکارت برمی‌گردد. جان اسنو^۶ در سال ۱۸۵۴ در زمان شیوع وبا در سوهو از این نمودار به منظور پیدا کردن ارتباط بین نواحی روی نقشه لندن که از پمپ‌های آب خاصی استفاده می‌کردند و نواحی با بیشترین آمار مرگ‌ومیر به دلیل وبا، بهره برد. درواقع وی از نمودار ورونوی استفاده کرد تا بتواند پاسخ مناسبی برای این سوال پیدا کند که چگونه اکثریت مردم ساکن سوهو که در اثر وبا جان خود را از دست دادند در نزدیکی پمپ‌های خیابان برود^۷ زندگی می‌کنند در صورتیکه اقلیت از سایر پمپ‌های آب استفاده می‌کنند.

کاربردهای نمودار ورونوی: این نمودار در ژئوفیزیک و هواشناسی به منظور آنالیز داده‌های توزیع فضایی (مانند اندازه‌گیری میزان بارش) بر مبنای اندازه‌گیری مجموعه‌ای از نقاط کاربرد دارد، در فیزیک مواد متراکم و فشرده مانند موزاییک‌کاری‌ها به عنوان سلول واحد شناخته می‌شود و در زمینه فضاهاى متریک عمومی سلول‌ها غالباً چندضلعی اساسی متریک نامیده می‌شوند. سایر نام‌های معادل برای این مفهوم (و یا موارد مهم خاص آن) عبارتند از: چندوجهی ورونوی، چندضلعی ورونوی، دامنه (های) نفوذ، تجزیه ورونوی، موزاییک‌کاری (های) ورونوی، موزاییک‌کاری (های) دیریکله.

^۳Dirichlet

^۴Dirichlet Tessellations

^۵Voronoi Tessellations

^۶John Snow

^۷broad street

نمودار ورونوی می‌تواند در استخراج ظرفیت شبکه بی‌سیم استفاده شود و در علم بوم‌شناسی به منظور مطالعه الگوهای رشد جنگل و همچنین استفاده موثر در توسعه مدل‌های پیش‌گویانه آتش‌سوزی جنگل‌ها کاربرد دارد. در گرافیک کامپیوتر این نمودار به منظور ایجاد متن‌های ساختمانی به کار می‌رود و در علم رباتیک در هدایت ربات‌های مستقل به منظور یافتن مسیر کاربرد مهمی دارد. در علم بیماری‌های واگیردار نمودار ورونوی می‌تواند در شناخت منابع وابسته به عفونت در بیماری‌های مسری مورد استفاده قرار گیرد.

روش ورونوی در سنجش مدور بودن در زمان ارزیابی مجموعه داده‌ها توسط دستگاه سنجش مختصات کاربرد موثری دارد. در فیزیک پلیمر نمودار ورونوی می‌تواند در نمایش حجم خالی پلیمرها مورد استفاده قرار گیرد.

۱.۱ تعاریف و اصطلاحات گرافی

تعریف ۱.۱.۱. گراف: یک ساختار دو تایی $G = (V, E)$ است که دو مجموعه $V(G)$ و $E(G)$ به صورت زیر تعریف می‌شود:

$$E(G) = \{e_1, e_2, \dots, e_m\} \quad m \in \mathbb{N}$$

$$V(G) = \{v_1, v_2, \dots, v_n\} \quad n \in \mathbb{N}, V(G) \neq \emptyset$$

هر e_i یک یال و هر v_i یک راس گراف نامیده می‌شود.

تعریف ۲.۱.۱. گراف مسطح: ^۸ گرافی است که بتوان آن را در صفحه طوری رسم کرد که هیچ دو یالی متقاطع نباشند.

تعریف ۳.۱.۱. گراف همبند: ^۹ گرافی است که برای هر افراز از مجموعه رئوس به دو مجموعه ناتهی X و Y ، یک یال وجود داشته باشد که یک سر آن در X و سر دیگر آن در Y قرار گیرد.

تعریف ۴.۱.۱. درخت: ^{۱۰} گراف همبند و بدون دور، درخت نامیده می‌شود.

تعریف ۵.۱.۱. برگ: ^{۱۱} راس درجه یک در درخت، برگ نامیده می‌شود.

تعریف ۶.۱.۱. گراف ۲- همبند: گراف همبندی که غیر جدا شدنی باشد یعنی با حذف یک راس همبند باقی بماند.

^۸Planar Graph

^۹Connected Graph

^{۱۰}Tree

^{۱۱}Leaf

تعریف ۷.۱.۱. گراف وزن دار: گراف $G = (V, E)$ وزن دار نامیده می شود هرگاه تابعی به صورت

$$f : E \rightarrow R | f(e) = r \in R$$

روی مجموعه یال ها قابل تعریف و موجود باشد.

تعریف ۸.۱.۱. گراف منتظم: گراف منتظم به گرافی گفته می شود که تمام رئوس آن درجه یکسانی دارند یا به عبارت دیگر تعداد یال مساوی از تمامی رئوس می گذرد. گراف منتظمی که درجه هر راس l باشد گراف l - منتظم خوانده می شود.

تعریف ۹.۱.۱. یک درخت پوشا (فراگیر) T از گراف همبند و بدون جهت G درختی شامل تمام رئوس و حداقل برخی از یال ها است. به بیان ساده تر میتوان گفت درخت پوشا درختی است که مجموعه ای از یال ها را شامل می شود در حالیکه تمام رئوس را پوشش می دهد در واقع تمام رئوس G در درخت پوشا وجود دارند با این شرط که هیچ دوری ایجاد نشود و درخت همبند نیز باشد. درخت پوشا را میتوان اینگونه نیز تعریف کرد:

- مجموعه ای حداقلی از یال های G که همه رئوس را به یکدیگر متصل می کند و یا

- مجموعه ای حداکثری از یال های G که هیچ دوری در آن وجود ندارد.

برای تعداد درخت های فراگیر در گراف یک فرمول بازگشتی (فرمول کیلی) وجود دارد که این فرمول مستلزم انقباض یک یال است که در ادامه به معرفی آن خواهیم پرداخت.

تعریف ۱۰.۱.۱. انقباض: یال e گراف G را منقبض می کند اگر آن یال حذف شده و دو انتهایش بر هم منطبق شوند، گراف حاصل را با $G.e$ و یا G_e نشان می دهند .

تعریف ۱۱.۱.۱. فرمول کیلی : این فرمول بیان می کند که تعداد درخت های n راسی برچسب دار برابر با n^{n-2} است .

تعریف ۱۲.۱.۱. جست و جوی عمق اول: جست و جوی عمق اول ^{۱۲} که به اختصار با DFS نشان می دهیم یک الگوریتم پیمایش گراف است که برای پیمایش یا جست و جوی یک درخت یا یک گراف بکار می رود .

پیمایش با انتخاب راس r بعنوان ریشه آغاز می شود. r بعنوان یک راس دیده شده برچسب می خورد. راس دلخواه r_1 از همسایگان r انتخاب شده و الگوریتم به صورت بازگشتی از r_1 به عنوان ریشه ادامه می یابد. از این پس در هر مرحله وقتی در راسی مانند v قرار گرفتیم که همه همسایگانش دیده شده اند، اجرای الگوریتم را برای آن راس خاتمه می دهیم. حال اگر بعد از اجرای الگوریتم با ریشه r_1 همه همسایگان r برچسب خورده باشند، الگوریتم پایان می یابد . در غیر این صورت راس دلخواه r_2 از همسایگان r را که هنوز برچسب نخورده انتخاب می کنیم و جست و جو را به صورت بازگشتی از r_2 به عنوان ریشه ادامه می دهیم. این روند تا مادامی که همه همسایگان r برچسب نخورده اند ادامه می یابد.

^{۱۲}Depth-First Search

تعریف ۱۳.۱.۱. گوییم تابع $f(n)$ از مرتبه $O(g(n))$ است و می‌نویسیم $f(n) = O(g(n))$ هرگاه اعداد ثابتی مانند n_0 و C موجود باشد به قسمی که به ازای هر n که $n \geq n_0$ داشته باشیم:

$$0 < f(n) \leq Cg(n)$$

تعریف ۱۴.۱.۱. گوییم تابع $f(n)$ از مرتبه $\Omega(g(n))$ است و می‌نویسیم $f(n) = \Omega(g(n))$ هرگاه اعداد ثابتی مانند n_0 و C موجود باشد به قسمی که به ازای هر n که $n \geq n_0$ داشته باشیم:

$$0 < Cg(n) \leq f(n)$$

تعریف ۱۵.۱.۱. گوییم تابع $f(n)$ از مرتبه $\theta(g(n))$ است و می‌نویسیم $f(n) = \theta(g(n))$ هرگاه اعداد ثابتی مانند n_0 و C موجود باشد به قسمی که به ازای هر n که $n \geq n_0$ داشته باشیم: $f(n) = O(g(n))$ و $f(n) = \Omega(g(n))$.

تعریف ۱۶.۱.۱. یک زنجیره^{۱۳} از گراف G یک توالی از رئوس و یال‌ها به صورت

$$x_0, e_1, x_1, e_2, \dots, e_n, x_n$$

است که شروع و پایان آن با راس است و هر یال بین دو راس قبل و بعد خود قرار دارد.

تعریف ۱۷.۱.۱. جاسازی گراف G بر روی سطح Σ نمایش G بر روی Σ است که نقاط Σ به رئوس گراف و منحنی‌های Σ (همسانریختی تصویری در بازه $[0, 1]$) به یال‌های گراف متناظر شده‌اند.

تعریف ۱۸.۱.۱. در نظریه گراف هر گراف ساده دارای حداقل یک مولفه همبندی^{۱۴} است. یک زیرگراف مانند H از G یک مولفه همبندی برای G است اگر و تنها اگر بین هر دو راس H حداقل یک مسیر وجود داشته باشد و با افزودن هر راس (و یا یال) دیگری از G به H این خاصیت از بین برود.

تعریف ۱۹.۱.۱. در نظریه ریاضی گراف‌های جهت‌دار، گرافی قویا همبند (همبند قوی) نامیده می‌شود که هر راسش قابل دسترسی از هر راس دیگرش باشد.

تعریف ۲۰.۱.۱. یک گراف القایی زیرگرافی است که مجموعه رئوس آن زیر مجموعه‌ای از مجموعه رئوس گرافی دیگر باشد با این ویژگی که این زیرگراف دارای تمامی یال‌هایی است که بین رئوس نظیر خود در مجموعه رئوس گراف اولیه موجود است.

تعریف ۲۱.۱.۱. گراف تنک: ^{۱۵} گرافی که یال‌های اندکی داشته باشد.

^{۱۳}Chain

^{۱۴}Connected Component

^{۱۵}Sparse Graph

تعریف ۲۲.۱.۱. تابع آکرمن: تابع بازگشتی چندضابطه‌ای که یک رایانه یا پردازشگر با حافظه بیکران می‌تواند آن را محاسبه کند.

$$A(m, n) = \begin{cases} n + 1, & m = 0, \\ A(m - 1, 1), & m > 0 \text{ and } n = 0, \\ A(m - 1, A(m, n - 1)), & m > 0 \text{ and } n > 0. \end{cases}$$

معکوس تابع آکرمن به صورت زیر تعریف می‌شود:

$$\alpha(m, n) = \min\{i \geq 1 : A(i, \lceil m/n \rceil) \geq \log n\}$$

۲.۱ تعاریف و مفاهیم مربوط به نمودار ورونوی

یک مجموعه متناهی از نقاط $\{p_1, p_2, p_3, \dots, p_n\}$ را در فضای اقلیدسی در نظر بگیرید. برای هر $k = 1, 2, \dots, n$ یک سلول ورونوی $\mathcal{V}_k$ تعریف می‌کنیم. فرض کنید $\mathcal{V}_k$ علاوه بر p_k شامل نقاط دیگری نیز است که فاصله هر نقطه درون $\mathcal{V}_k$ تا p_k کمتر یا برابر فاصله سایر سایت‌ها تا p_k است. این سلول از به اشتراک گذاشتن نیمی از فضا حاصل می‌شود و بنابراین یک چندضلعی محدب نامیده می‌شود. بخش‌های نمودار ورونوی تمامی نقاطی از سطح است که با دو سایت از نزدیک‌ترین سایت‌ها هم فاصله است. راس‌های ورونوی (گره‌ها) نقاط هم‌فاصله با سه سایت (یا تعداد بیشتر) است.

نمودار ورونوی در حالت‌های ساده

- نمودار ورونوی یک نقطه تنها از همان نقطه تشکیل شده است.
- عمودمنصف پاره‌خط واصل دو نقطه، نمودار ورونوی آن دو نقطه است.
- برای سه نقطه غیر هم‌خط نمودار ورونوی آن‌ها از سه نیم‌خط تشکیل شده است.

در ادامه به تعریف ریاضی نمودار ورونوی می‌پردازیم: فرض کنید X یک فضا (مجموعه ناتهی) با تابع فاصله d باشد، فرض کنید K یک مجموعه از اندیس‌ها و $(p_k)_{k \in K}$ یک چندتایی مرتب شده از زیرمجموعه‌های ناتهی (سایت‌ها) در فضای X باشد، سلول ورونوی یا ناحیه ورونوی $\mathcal{V}_k$ (متناظر با p_k) مجموعه‌ای از نقاط در فضای X است که فاصله آن‌ها تا p_k کوچکتر یا مساوی با فاصله سایر نقاط p_j است ($j \neq k$).

به عبارت دیگر اگر $d(x, A) = \inf\{d(x, a) | a \in A\}$ مشخص کننده فاصله بین x و زیرمجموعه A باشد آنگاه:

$$\mathcal{V}_k = \{x \in X \mid d(x, p_k) \leq d(x, p_j) \forall j \neq k\}.$$

نمودار ورونوی یک چندتایی ساده از سلول‌های (V_k) $k \in K$ است.

به‌عنوان یک توضیح ساده مجموعه‌ای از فروشگاه‌های سطح شهر را در نظر بگیرید، فرض کنید بخواهیم تعداد مشتریان را تخمین بزنیم (با فرض اینکه پارامترهای قیمت، محصولات، کیفیت خدمات و ... ثابت باشد) و تنها فاکتور انتخاب فروشگاه توسط مشتریان فاصله تا فروشگاه باشد واضح است مشتریان فروشگاه‌ی را انتخاب می‌کنند که از موقعیت آنها حداقل فاصله را دارا باشد.

{جست و جوی نزدیک‌ترین همسایه^{۱۶} (که به اختصار NNS می‌نامیم) ، یک مسئله بهینه‌سازی برای پیدا کردن نزدیک‌ترین نقاط در فضاها می‌تربیک است و همچنین با نام‌های جست و جوی مجاورت، جست و جوی همسانی و یا جست و جوی نزدیک‌ترین نقطه شناخته می‌شود .

مساله به این صورت است که مجموعه S شامل تعدادی نقطه در یک فضای متریک M و نیز یک نقطه پرس و جو $q \in M$ داده شده است . هدف پیدا کردن نزدیک‌ترین نقطه در S به q است . در بسیاری از موارد فضای M به صورت یک فضای اقلیدسی d - بعدی و فاصله بین نقاط با معیار فاصله اقلیدسی ، فاصله منتهن یا دیگر فاصله‌های متریک سنجیده می‌شود .

پروفسور دونالد کنوت^{۱۷} در جلد سوم از هنر برنامه نویسی کامپیوتر^{۱۸} این مسئله را مسئله دفتر پست^{۱۹} نامید . این نام‌گذاری اشاره به یک برنامه کامپیوتری دارد که برای اختصاص نزدیک‌ترین اداره پست به یک محل اقامت نوشته شده است .

NNS کاربردهای بسیاری در هندسه محاسباتی ، پایگاه داده (بعنوان مثال بازیابی تصویر مبتنی بر محتوا)، نظریه رمزنگاری، سامانه توصیه‌گر (بعنوان مثال پالایش گروهی)، بازیابی اینترنتی مانند تبلیغات محتوایی و هدف‌گیری رفتاری، تعیین توالی DNA ، غلطیاب (پیشنهاد املائی درست)، کشف دستبرد علمی، Google Maps و ... دارد .

راه حل‌های متعددی برای مسئله NNS پیشنهاد شده‌اند (بعنوان مثال جست و جوی خطی و پارتیشن بندی فضا). کیفیت و سودمندی این الگوریتم‌ها براساس پیچیدگی زمانی و همچنین پیچیدگی حافظه مصرفی ساختمان داده‌های آنها ، ارزیابی می‌شود . مشاهدات غیر رسمی بیان می‌دارد که در واقع هیچ راه حل دقیق و همه منظوره‌ای برای مسائل NNS وجود ندارد .

جست و جوی خطی : ساده‌ترین راه حل برای مسئله NNS محاسبه فاصله نقطه پرسش شده تا تمامی نقاط پایگاه داده ، همراه با نگهداری بهترین جواب پیدا شده تا کنون است . این الگوریتم که به آن الگوریتم سر راست نیز گفته می‌شود دارای مرتبه زمانی $O(ND)$ است که در آن N اندازه مجموعه S و D تعداد ابعاد فضای M است . در این روش هیچ داده ساختاری از جست و جوی ذخیره نخواهد شد و الگوریتم به جز ذخیره نقاط پایگاه داده ، هزینه حافظه‌ای دیگری ندارد .

پارتیشن بندی فضا : از سال ۱۹۷۰ روش شاخه و کران بر روی مسئله NNS اعمال شده است. در

^{۱۶}Nearest Neighbor Search^{۱۷}Donald Knuth^{۱۸}The Art Of Computer Programming^{۱۹}The Post Office Problem

فضای اقلیدسی این کار با نام شاخص‌های مکانی یا روش دسترسی مکانی شناخته می‌شود. روش‌ها متعددی برای پارتیشن بندی فضا به منظور حل مسئله NNS ارائه و توسعه داده شده‌اند. در این بین ساده‌ترین روش استفاده از ساختمان داده درخت کی دی^{۲۰} است که به صورت تکراری فضای جست و جو را در یکی از ابعاد، به دو نیم بخش تقسیم می‌کند به گونه‌ای که هر نیم بخش تقریباً نیمی از نقاط بخش بزرگتر را در بر خواهد گرفت. پرس و جو با پیمایش درخت از ریشه تا برگ صورت می‌پذیرد به گونه‌ای که در هر مرحله مقدار بردار گره در بعد مشخصی با همان بعد از نقطه پرس و جو مقایسه می‌شود، اگر مقدار نقطه پرس و جو بیشتر بود به زیردرخت سمت راست و در غیر این‌صورت به زیر درخت چپ حرکت می‌کنیم. همچنین بسته به اینکه فاصله این دو نقطه چقدر است شاخه‌های کناری نیز ممکن است نیاز به بررسی داشته باشند. برای مسائلی که تعداد بعد فضا ثابت است میانگین پیچیدگی زمانی این روش از مرتبه $O(\log N)$ است. در مورد نقاط تصادفی توزیع شده تحلیل پیچیدگی زمانی مربوط به بدترین حالت صورت می‌گیرد.^{۲۱}

شکل ۲.۱ نمودار ورونوی فروشگاه‌ها در بخشی از شهر تهران را نشان می‌دهد. در این مثال p_k فروشگاه و v_k سلول ورونوی است. در مدل‌سازی ما هر فروشگاه در شهر در حقیقت یک نقطه از نمودار ورونوی است، فرض بر این است که فاصله بین نقاط به صورت استاندارد (فاصله اقلیدسی) اندازه‌گیری شده باشد یعنی:

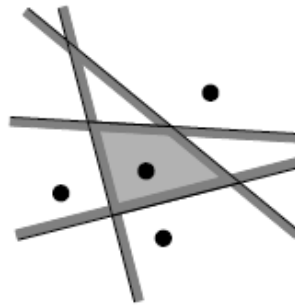
$$d((a_1, a_2), (b_1, b_2)) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}.$$

اگر مسئله را طوری بررسی کنیم که مشتریان تنها با وسیله نقلیه به فروشگاه رفته و مسیرهای ترافیکی به موازات محورهای X و Y باشد، بنابراین تابع فاصله واقعی ℓ_1 خواهد بود و داریم:

$$d((a_1, a_2), (b_1, b_2)) = |a_1 - b_1| + |a_2 - b_2|.$$

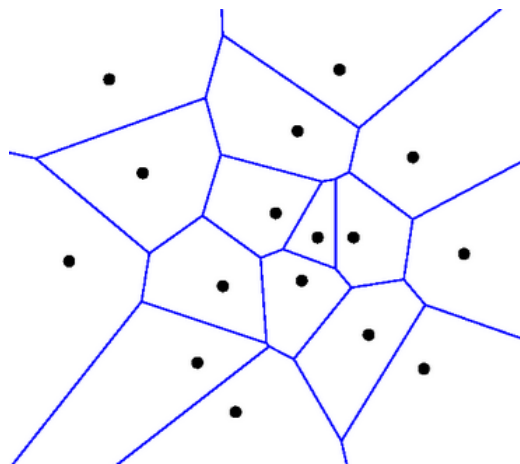
^{۲۰} k-d Tree

^{۲۱} مطالب این قسمت برگرفته از ویکی پدیا فارسی می‌باشد.



شکل ۳.۱: سلول ورونوی

قضیه ۱.۲.۱. اگر P مجموعه‌ای از n سایت نقطه‌ای غیر هم خط در صفحه باشد، یال‌های $Vor(P)$ پاره خط یا نیم خط هستند و $Vor(P)$ همبند است.



شکل ۴.۱: یک نمودار ورونوی

برهان. می‌دانیم یال‌های دیاگرام ورونوی بخشی از خطوط مستقیم (بخشی از عمودمنصف‌های بین جفت سایت‌ها) است؛ به برهان خلف فرض کنید یال e از $Vor(P)$ وجود دارد که خط کامل^{۲۳} است. فرض کنید e روی مرز سلول‌های ورونوی $\mathcal{V}p_i$ و $\mathcal{V}p_j$ بوده و $p_k \in P$ سایتی باشد که با دو سایت ذکر شده همراستا نیست، منصف p_k و p_j موازی با e نیست و آن را قطع می‌کند در این صورت بخشی از e که در ناحیه داخلی $h(p_k, p_j)$ قرار دارد نمی‌تواند روی مرز $\mathcal{V}(p_j)$ باشد زیرا p_k نزدیک‌تر از p_j است و این خلاف فرض است لذا e نمی‌تواند خط کامل باشد. (شکل ۴.۱ را ببینید).
حال به اثبات همبندی می‌پردازیم. فرض کنید $Vor(p)$ همبند نباشد در این صورت یک سلول ورونوی

مانند $\mathcal{V}(p_i)$ خواهیم داشت که صفحه را به دو قسمت تقسیم می‌کند از آن‌جا که سلول‌های ورونوی محدب هستند $\mathcal{V}(p_i)$ شامل یک نوار ^{۲۴} محدود شده به وسیله دو خط کامل موازی خواهد بود اما در قسمت قبل ثابت کردیم یال‌های دیاگرام ورونوی نمی‌توانند خط کامل باشند بنابراین فرض خلف باطل و حکم برقرار است. $\square$

حال که ساختار نمودار ورونوی را درک کردیم به بررسی پیچیدگی آن می‌پردازیم. منظور از پیچیدگی یک نمودار ورونوی تعداد کل رئوس و یال‌ها است.

قضیه ۲.۲.۱. به ازای $n \geq 3$ که n مجموعه‌ای از سایت‌های نقطه‌ای در صفحه است؛ حداکثر تعداد رئوس در نمودار ورونوی آن $2n - 5$ و حداکثر تعداد یال آن $3n - 6$ است.

برهان. اگر سایت‌ها همگی هم‌خط باشند آنگاه $Vor(p)$ شامل $n - 1$ خط موازی است، بنابراین فرض کنید چنین نباشد. با استفاده از فرمول اویلر ^{۲۵} به اثبات می‌پردازیم. طبق این رابطه برای هر گراف جاسازی شده مسطح همبند ^{۲۶} با m_v راس (گره) و m_e کمان (یال) و m_f وجه رابطه زیر برقرار است:

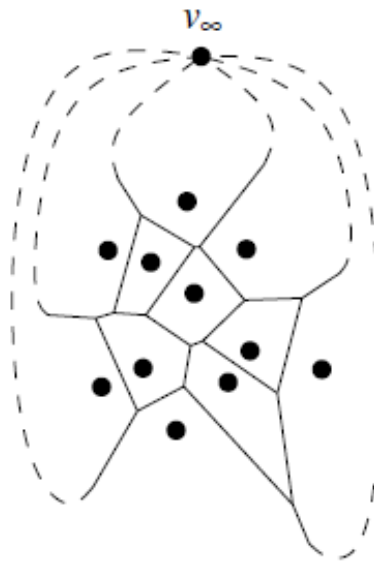
$$m_v - m_e + m_f = 2$$

ما نمی‌توانیم فرمول اویلر را مستقیماً برای $Vor(P)$ استفاده کنیم زیرا دارای یال‌های نیمه نامتناهی است و گراف نیست. برای حل این مشکل یک راس اضافه v_∞ را در بینهایت به مجموعه رئوس اضافه می‌کنیم و تمام یال‌های نیمه بینهایت $Vor(P)$ را به این راس متصل می‌کنیم. (شکل ۵.۱ را ببینید.) حال یک گراف همبند مسطح داریم که می‌توانیم در آن فرمول اویلر را اعمال کنیم.

^{۲۴}Strip

^{۲۵}Euler's Formula

^{۲۶}Connected Planar Embedded Graph



شکل ۵.۱: اضافه کردن راسی در بینهایت

با جای‌گذاری در رابطه قبل، رابطه زیر بدست می‌آید که در آن n تعداد سایت‌هاست.

$$(n_v + 1) - n_e + n = 2$$

$$n_v = n_e - n + 1 \quad (2.1)$$

$$n_e = n_v + n - 1 \quad (3.1)$$

به علاوه هر یال در گراف افزوده ^{۲۷} دقیقاً دو راس دارد، بنابراین اگر درجه تمام رئوس را جمع بزنیم به دو برابر تعداد یال‌ها خواهیم رسید. از آنجا که درجه هر راس حداقل ۳ است بنابراین:

$$2n_e \geq 3(n_v + 1) \quad (4.1)$$

با جای‌گذاری (۲.۱) در معادله (۴.۱) داریم:

$$n_e \geq \frac{3}{2}(n_v + 1) \Rightarrow n_e \geq \frac{3}{2}(n_e - n + 2) \Rightarrow n_e \leq 3n - 6$$

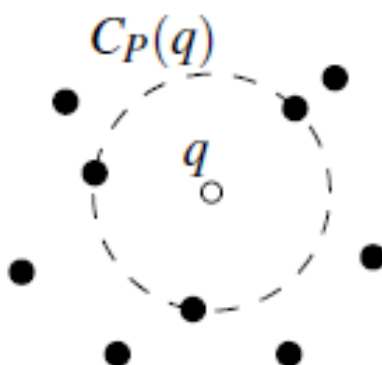
و از جای‌گذاری (۳.۱) در (۴.۱) به حداکثر تعداد رئوس خواهیم رسید.

$$n_v + n - 1 \geq \frac{3}{2}n_v + \frac{3}{2} \Rightarrow n_v \leq 2n - 5$$

□

می‌دانیم پیچیدگی (تعداد رئوس و یال‌ها) $Vor(P)$ خطی است و لذا تمام منصف‌ها یال‌های $Vor(P)$ و نیز تمام تقاطع‌ها رئوسی از آن نخواهند بود، برای مشخص شدن اینکه کدام منصف‌ها و تقاطع‌ها ویژگی نمودار ورونوی دارند تعریف و قضیه زیر را ارائه می‌دهیم.

تعریف ۱.۲.۱. برای یک نقطه q ، بزرگترین دایره خالی^{۲۸} مربوط به q تحت P را با $C_P(q)$ نشان می‌دهیم و عبارت است از بزرگترین دایره به مرکز q که هیچ سایتی از P را داخل خودش ندارد (شکل ۶.۱ را ببینید).



شکل ۶.۱: بزرگترین دایره خالی

تعریف ۲.۲.۱. یک زیرمجموعه S از صفحه $\mathbb{R}^2$ محدب^{۲۹} نامیده می‌شود اگر و تنها اگر برای هر جفت از نقاط $p, q \in S$ پاره خط $\overline{pq}$ بطور کامل در S قرار گیرد.

تعریف ۳.۲.۱. پوسته محدب^{۳۰} مجموعه S که با $CH(S)$ نشان می‌دهیم، کوچک‌ترین مجموعه محدبی است که شامل S باشد به بیانی دقیق‌تر پوسته محدب اشتراک تمام مجموعه‌های محدبی است که شامل S هستند.

الگوریتم برای یافتن بزرگ‌ترین دایره خالی :

- رئوس نمودار ورونوی را در نظر گرفته ، در بین دایره‌های خالی به مرکز این رئوس بزرگترین دایره را پیدا می‌کنیم .
- در بین دایره‌هایی که مرکز آنها تقاطع یک ضلع ورونوی و یک ضلع پوسته محدب است بزرگ‌ترین دایره خالی را پیدا می‌کنیم .

^{۲۸}Largest Empty Circle

^{۲۹}Convex

^{۳۰}Convex Hull

• بزرگترین دایره بین جواب‌های بالا ، جواب مسئله (بزرگترین دایره خالی) است .

اگر $f(p)$ شعاع دایره خالی C به مرکز p باشد ، آنگاه در جست و جوی $\max_{p \in CH(S)} f(p)$ هستیم .

قضیه ۳.۲.۱. برای نمودار ورونوی $Vor(P)$ یک مجموعه از نقاط P در صفحه ، موارد زیر برقرار است:

آ- یک نقطه q یک راس از $Vor(P)$ است اگر و فقط اگر بزرگترین دایره خالی آن ، یعنی $C_P(q)$ دارای سه یا بیشتر از سه سایت بر روی مرز خود باشد.

ب- منصف بین سایت‌های p_i و p_j یک یال از $Vor(P)$ را تعریف می‌کند اگر و فقط اگر یک نقطه q بر روی منصف وجود داشته باشد به طوری که مرز $C_P(q)$ فقط شامل دو سایت p_i و p_j باشد.

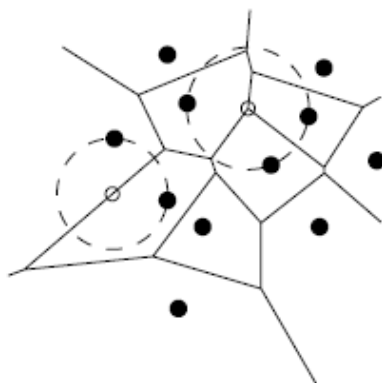
برهان. آ- فرض کنید نقطه q طوری قرار گرفته باشد که مرز $C_P(q)$ شامل سه سایت یا بیشتر باشد و p_i و p_j و p_k سه سایت بر روی مرز این دایره باشند. از آنجا که ناحیه داخلی $C_P(q)$ تهی است q باید بر روی مرز سه سلول ورونوی $\mathcal{V}(p_i)$ و $\mathcal{V}(p_j)$ و $\mathcal{V}(p_k)$ قرار داشته و یکی از رئوس $Vor(P)$ نیز باشد .

از دیگر سو ، راس q از $Vor(P)$ حداقل با سه یال مجاور است ، بنابراین حداقل برای سه سلول ورونوی $\mathcal{V}(p_i)$ و $\mathcal{V}(p_j)$ و $\mathcal{V}(p_k)$ این راس باید از p_i و p_j و p_k به یک فاصله بوده و هیچ سایت دیگری که به q نزدیک‌تر باشد وجود نداشته باشد زیرا در غیر این صورت $\mathcal{V}(p_i)$ و $\mathcal{V}(p_j)$ در نقطه q مجاور نخواهند بود بنابراین هیچ سایت دیگری در داخل دایره‌ای که p_i و p_j و p_k بر روی مرز آن هستند وجود نخواهد داشت .

ب- فرض کنید نقطه q با ویژگی‌های ذکر شده در قضیه ، وجود دارد. از آنجا که $C_P(q)$ شامل هیچ سایتی در درون خود نیست و p_i و p_j بر روی مرز هستند برای هر $1 \leq k \leq n$ داریم :

$$d(q, p_i) = d(q, p_j) \leq (q, p_k).$$

این مطلب ایجاب می‌کند که q روی یال یا راسی از $Vor(P)$ قرار داشته باشد ؛ (شکل ۷.۱ را ببینید .)



شکل ۷.۱: قرار گرفتن q روی یال یا راسی از سلول ورونوی

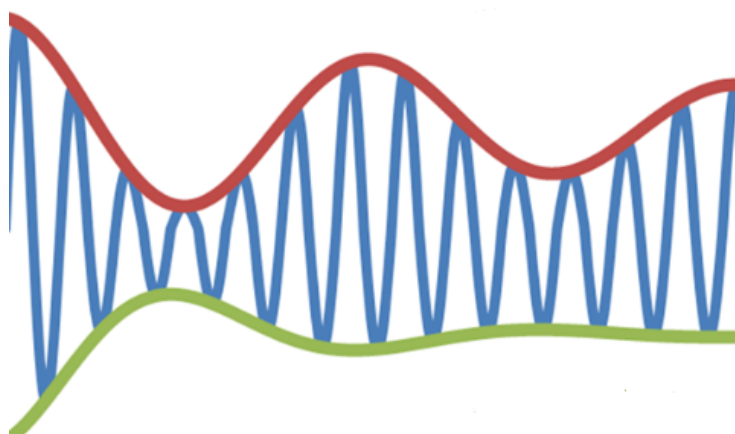
طبق قسمت قبل q نمی‌تواند راسی از $Vor(P)$ باشد بنابراین q روی یالی از آن قرار دارد که توسط منصف p_i و p_j تعریف شده است .
 بالعکس فرض کنید منصف p_i و p_j یک یال ورونوی باشد، بنابراین مرز $C_P(q)$ که روی دو سر یال قرار نمی‌گیرد باید تنها شامل p_i و p_j باشد .

□

تعریف ۴.۲.۱. نمودار ورونوی وزن‌دار: نموداری است که در آن فاصله یک سایت از یک نقطه به صورت فاصله اقلیدسی، بعلاوه وزن اضافی برای هر سایت می‌باشد .

تعریف ۵.۲.۱. یک پوش^{۳۱} از یک خانواده از منحنی‌ها در صفحه یک منحنی است که مماس بر هریک از اعضای خانواده است و این نقاط اشتراک با هم کل پوش را تشکیل می‌دهند. (شکل ۸.۱ را ببینید.)

^{۳۱}Envelope



شکل ۸.۱: پوش بالایی که با رنگ قرمز مشخص شده و پوش پایینی که با رنگ سبز مشخص شده است.

پوش پایینی در نمودار ورونوی معادل "خط ساحلی" است.

تعریف ۶.۲.۱. مسئله یافتن کوتاهترین مسیر در واقع مسئله یافتن مسیری بین دو راس (یا گره) است به گونه‌ای که مجموع وزن یال‌های تشکیل دهنده آن کمینه شود. یکی از مهم‌ترین الگوریتم‌ها در حل این مسئله الگوریتم دایکسترا می‌باشد.

الگوریتم دایکسترا: ۳۲

۱. قرار دهید $g(s) = 0$ و $U = \{s\}$ (برچسب دائمی رئوس است و U شامل مجموعه رئوسی است که بررسی شده‌اند.)

$$h(j) = \begin{cases} c_{sj} & (s, j) \in E, \\ \infty & \text{در غیر اینصورت} \end{cases}$$

۲. $i = \arg \min_{j \notin U} h(j)$ و قرار دهید: ۳۳

$$g(i) = h(i)$$

$$U = U \cup \{i\}$$

اگر $U = V$ توقف کنید.

۳. به ازای تمام رئوس $j \notin U$ و $(i, j) \in E$ قرار دهید: $h(j) = \min\{h(j), g(i) + c_{ij}\}$ و به گام دوم بروید.

^{۳۲} این الگوریتم در یافتن کوتاهترین مسیر از یک راس (مثلاً s) به سایر رئوس در گراف با وزن مثبت کاربرد دارد.

^{۳۳} مقصود از $\arg \min$ اندیسی است که h را مینیمم می‌کند.

۱.۲.۱ محاسبه نمودار ورونوی

در این قسمت به چگونگی محاسبه نمودار ورونوی در صفحه می‌پردازیم. دو الگوریتم برای محاسبه نمودار ورونوی وجود دارد.

۱. با استفاده از اشتراک نیم صفحه‌ها

۲. الگوریتم جاروب صفحه

که روش اول در زمان $O(n^2 \log n)$ انجام می‌پذیرد. سلول ورونوی طبق (۱.۱) $h(p_i, p_j)$ با شرط $i \neq j$ از طریق الگوریتم زیر بدست می‌آید:

Algorithm Intersect Half Planes (H)

Input : A set H of n half-planes in the plane.

Output : The convex polygonal region $C := \cap_{h \in H} h$

if card(H)=1

then $C \leftarrow$ The unique half-plane $h \in H$

else split H into sets H_1 and H_2 of size $\lceil n/2 \rceil$ and $\lfloor n/2 \rfloor$

$C_1 \leftarrow$ Intersect Half Planes (H_1)

$C_2 \leftarrow$ Intersect Half Planes (H_2)

$C \leftarrow$ Intersect Convex Regions (C_1, C_2)

با این روش، محاسبه هر سلول ورونوی از مرتبه $O(n \log n)$ است که به الگوریتمی از مرتبه زمانی $O(n^2 \log n)$ برای محاسبه کل دیاگرام ورونوی منتهی می‌شود.

البته ما به کمک الگوریتم جاروب صفحه می‌توانیم بهتر عمل کنیم.

ایده اصلی این الگوریتم، جاروب صفحه از بالا به پایین با یک خط افقی (خط جاروب) ^{۳۴} است در هنگام جاروب صفحه اطلاعات با توجه به ساختاری که دنبال محاسبه آن هستیم، نگهداری می‌شوند به بیانی دقیق‌تر اطلاعاتی که درباره تقاطع خط جاروب با ساختار است، نگهداری می‌شوند و هنگامیکه خط جاروب رو به پایین حرکت می‌کند اطلاعات جز در نقاط ویژه‌ای که نقاط رویداد ^{۳۵} نامیده می‌شوند، تغییری نمی‌کنند.

این الگوریتم که معمولاً به نام مخترع آن فورچون ^{۳۶} شناخته می‌شود دیاگرام ورونوی را در زمان $O(n \log n)$ محاسبه می‌کند. هر الگوریتمی برای محاسبه دیاگرام ورونوی در بدترین حالت از مرتبه زمانی $\Omega(n \log n)$ خواهد بود و از این رو الگوریتم فورچون الگوریتمی بهینه است. طبق الگوریتم جاروب صفحه به جای نگهداری تقاطع دیاگرام ورونوی با خط جاروب، اطلاعات آن قسمت از

^{۳۴}sweep line

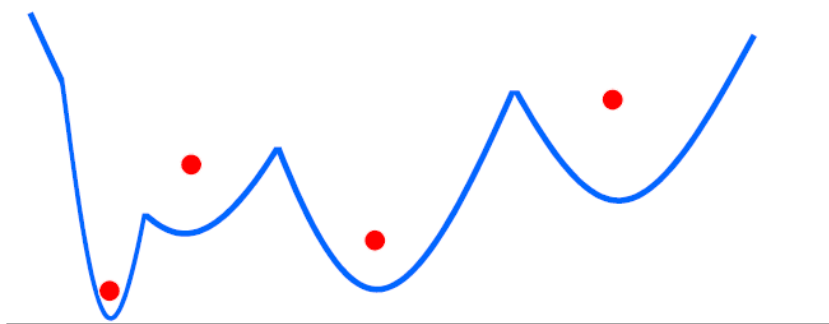
^{۳۵}event points

^{۳۶}Fortune's algorithm

نمودار ورونوی را نگهداری می‌کند که بالای خط l قرار دارد و نمی‌تواند توسط سایت‌هایی که پایین‌تر از خط l قرار می‌گیرند تغییر کند. مکان هندسی نقاطی که فاصله آنها تا هر سایت بالای l کمتر از فاصله آنها از l است به صورت کمان‌های سهمی شکل است که این زنجیره از کمان‌های سهمی شکل را **خط ساحلی**^{۳۷} می‌نامند. به عبارت دیگر خط ساحلی تابعی است که (برای هر مختص x) از پایین‌ترین نقطه بر روی تمام سهمی‌ها عبور می‌کند. (شکل ۹.۱ را ببینید.)

خط ساحلی دارای خواص زیر است:

- خط ساحلی نسبت به محور x یکنواست.
- یک سهمی ممکن است در دو قسمت از خط ساحلی ظاهر شود.
- محل تقاطع سهمی‌ها روی ضلع دیاگرام ورونوی قرار دارد.



شکل ۹.۱: خط ساحلی

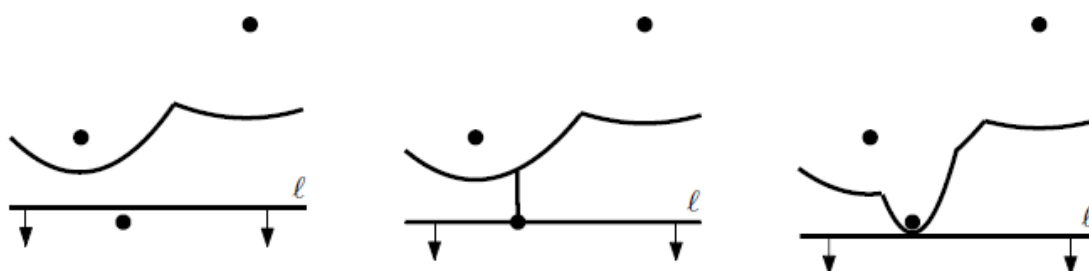
قضیه ۴.۲.۱. نمودار ورونوی یک مجموعه از n سایت نقطه‌ای در صفحه، می‌تواند با یک الگوریتم جاروب در زمان $O(n \log n)$ با استفاده از حافظه $O(n)$ محاسبه گردد.

ساختن نمودار ورونوی به کمک خط ساحلی:

با حرکت دادن خط جاروب و دنبال کردن محل تقاطع سهمی‌ها در خط ساحلی میتوان دیاگرام ورونوی تشکیل داد. خط ساحلی به طور پیوسته تغییر می‌کند که این تغییرات به دو صورت اتفاق می‌افتد: یا هنگامی روی می‌دهد که کمان سهمی شکل جدیدی بر روی خط ساحلی ظاهر شود (در این حالت خط l به پایین آن سایت حرکت می‌کند) و یا هنگامی که کمان سهمی کوچک و کوچکت‌ر شده و پس از تبدیل به نقطه، حذف می‌شود. ابتدا حالتی را در نظر بگیرید که کمان جدیدی بر روی خط ساحلی ظاهر می‌شود (شکل ۱۰.۱ را ببینید.) در این حالت خط l با حرکت به سمت پایین به سایت

^{۳۷}beach line

جدیدی می‌رسد . سهمی تعریف شده با این سایت در ابتدا یک سهمی با عرض صفر است و یک پاره خط عمودی سایت جدید را به خط ساحلی متصل می‌کند. این رویداد که در آن با یک سایت جدید مواجه می‌شویم را رویداد سایت ^{۳۸} می‌نامیم. تنها راه اضافه شدن یال جدید به خط ساحلی، رویداد سایت است.

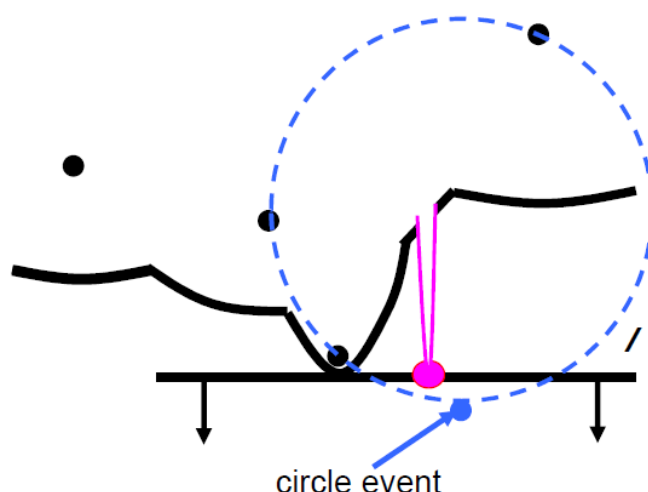


شکل ۱۰.۱: ظهور کمان بر روی خط ساحلی

رویداد دیگری که رخ می‌دهد، کوچک شدن یک کمان، تبدیل آن به یک نقطه و سپس حذف آن از خط ساحلی است. هنگامی که یک کمان از خط ساحلی حذف می‌شود (شکل ۱۲.۱ را ببینید)، و دو نقطه شکستگی به هم می‌رسند دو یال نمودار ورونوی نیز به هم می‌رسند و این نقطه هم‌رسی، یک رأس از نمودار ورونوی خواهد بود، این رویداد را که در آن خط جاروب به پایین‌ترین نقطه دایره‌ای برسد که از سه سایت تعریف کننده کمان‌های متوالی روی خط ساحلی عبور می‌کند را رویداد دایره ^{۳۹} نامیم. (شکل ۱۱.۱ را ببینید).

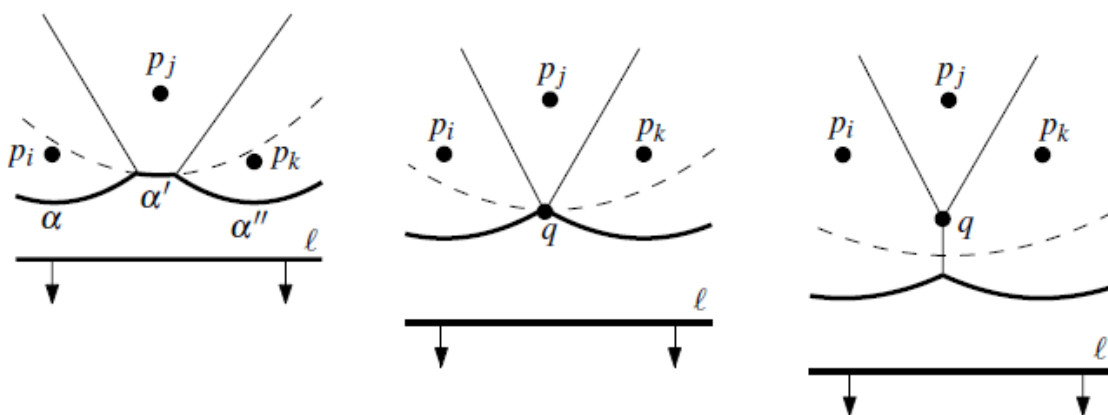
^{۳۸}site event

^{۳۹}Circle Event



شکل ۱۱.۱: رویداد دایره

تنها راه حذف یک کمان از خط ساحلی رویداد دایره است و هر راس نمودار ورونوی به وسیله این رویداد به وجود می‌آید.



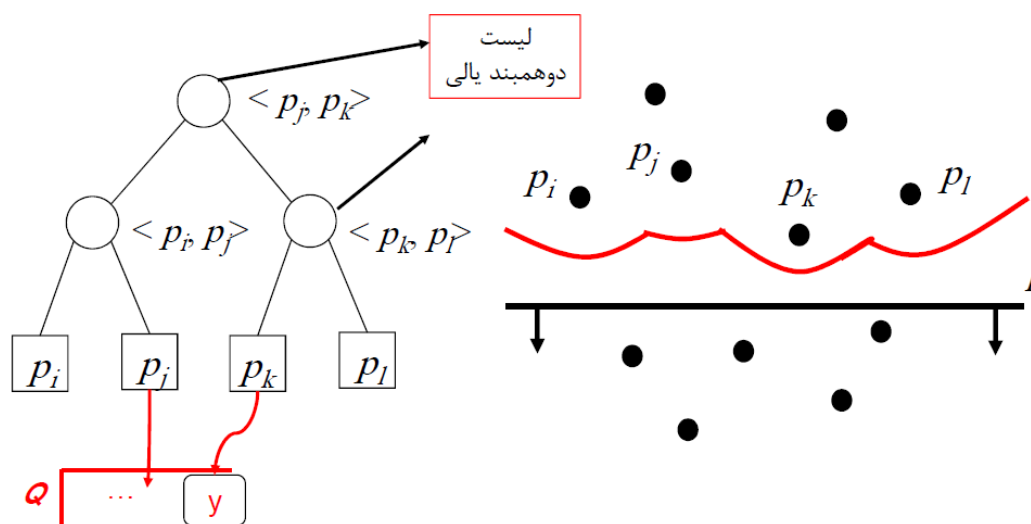
شکل ۱۲.۱: حذف کمان از خط ساحلی

الگوریتم جاروب رویدادها را از بالا به پایین بررسی می‌کند بنابراین هنگامی که دو یا چند رویداد بر روی یک خط افقی مشترک قرار گیرند، با یک حالت تباهیده رو به رو هستیم. علاوه بر تباهیده‌گی‌هایی که در ترتیب انتخاب رویدادها وجود دارد، ممکن است در هنگام بررسی یک رویداد نیز با حالت‌های تباهیده‌گی مواجه شویم. این حالت هنگامی رخ می‌دهد که سایت در حال پردازش p_i دقیقاً پایین‌تر از نقطه شکستگی بین دو کمان از خط ساحلی قرار گیرد که در این حالت الگوریتم هر دو کمان را به دو قسمت تقسیم می‌کند که طول یک قسمت صفر است و کمانی برای سایت p_i بین کمان‌های تقسیم

شده قرار می‌دهد.

اکنون می‌دانیم که چگونه این مطالب به نمودار ورونوی در حال ساخت مرتبط می‌شوند: در یک پیشامد سایت یک یال جدید شروع به رشد می‌کند و در یک پیشامد دایره دو یال در حال رشد به منظور تشکیل یک راس، با هم تلاقی می‌کنند، آنچه برای پیدا کردن باقی می‌ماند یک ساختمان داده به منظور ذخیره اطلاعات در حین جاروب است. ما نیازمند دو ساختمان داده استاندارد برای هر الگوریتم جاروب خط هستیم: یک درخت جست و جوی دودویی (شکل ۱۳.۱ را ببینید) برای حفظ وضعیت خط جاروب و یک صف از پیشامدها (شکل ۱۴.۱ را ببینید). در درخت جست و جوی دودویی قسمت‌های سهمی‌ها به ترتیب نشان داده می‌شوند.

- در هر برگ سائیتی که سهمی متناظر را می‌سازد، ذخیره می‌شود.
- رئوس داخلی نقاط تقاطع سهمی‌ها را نشان می‌دهد، در این رئوس زوج مرتب سایت‌های متناظر با سهمی‌ها ذخیره می‌شود.
- هر برگ از این درخت یک اشاره‌گر به رویداد حذف سهمی متناظر از خط ساحلی است و در صورتی که این پیشامد موجود نباشد، اشاره‌گر پوچ می‌شود.
- هر راس داخلی اشاره‌گری به نیم ضلع متناظر در دیاگرام ورونوی دارد.

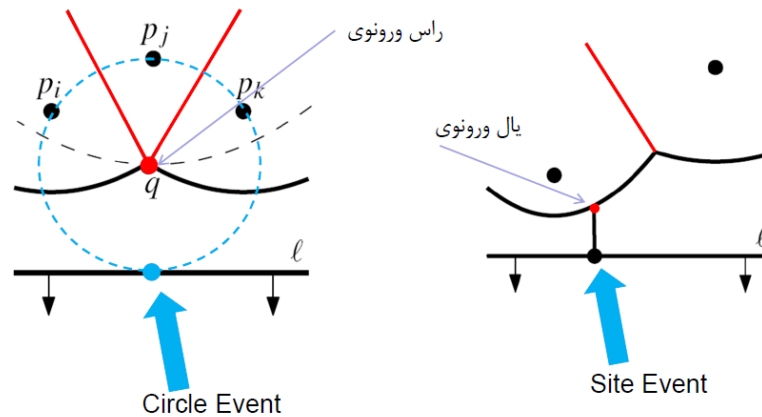


شکل ۱۳.۱: درخت جست و جوی دودویی نشان دهنده خط ساحلی

در صف پیشامدها:

- پیشامدهایی که تا کنون مشخص شده‌اند، در صف (بر اساس مختصات y) ذخیره می‌شوند.
- در هر رویداد سایت خود سایت نیز ذخیره می‌شود.

- در هر رویداد دایره پایین‌ترین نقطه دایره بعلاوه یک اشاره‌گر به برگه از درخت جست و جوی دودویی که کمانی که در این پیشامد حذف می‌شود را نشان می‌دهد، ذخیره می‌شود.

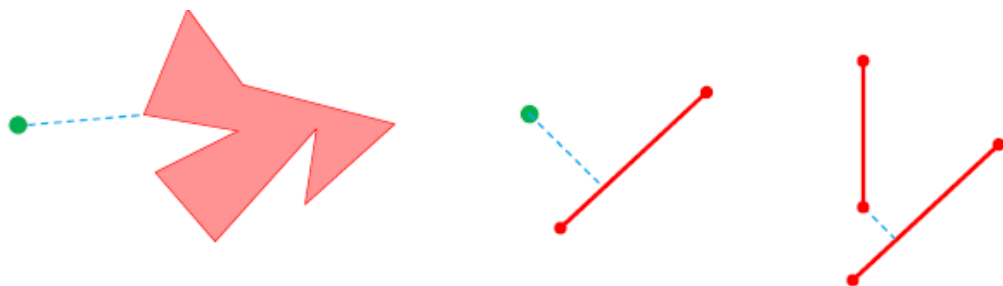


شکل ۱۴.۱: صف رویدادها

نمودار ورونوی پاره‌خطها :

در بسیاری از کاربردها مانند تعیین مسیر حرکت ربات در ناحیه دارای مانع، نمودار ورونوی اشیا دیگری به جز نقطه را باید تعیین کرد.

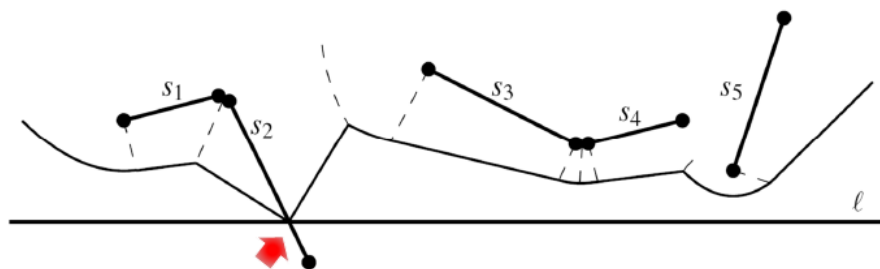
- فاصله یک نقطه از شکلی هندسی فاصله آن نقطه تا نزدیک‌ترین نقطه شکل است. (شکل ۱۵.۱ را ببینید.)



شکل ۱۵.۱: فاصله یک نقطه از شکل

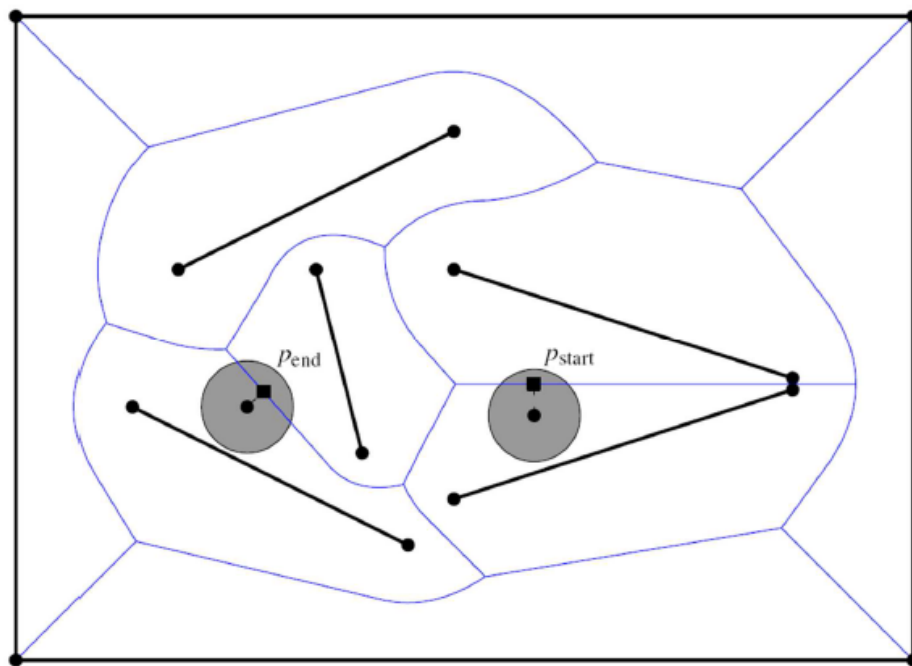
برای ساختن نمودار ورونوی پاره‌خطها نیز مجدداً از الگوریتم جاروب بهره می‌بریم. (شکل ۱۶.۱ را ببینید.)

فرض کنید p یک نقطه شکستگی روی خط ساحلی باشد:



شکل ۱۶.۱: نمودار ورونوی پاره‌خط‌ها به کمک الگوریتم جاروب

- اگر p نزدیک‌ترین نقطه به رئوس انتهایی دو سایت باشد و فاصله آن تا l برابر فاصله آن تا هر یک از این دو راس باشد آنگاه p یک پاره‌خط را می‌سازد.
- اگر p نزدیک‌ترین نقطه به نقاط داخلی دو سایت باشد و فاصله آن تا l برابر فاصله آن تا هر یک از این دو نقطه باشد آن p یک پاره‌خط را می‌سازد.
- اگر p نزدیک‌ترین نقطه به یک راس انتهایی یک سایت و یک نقطه درونی سایت دیگر باشد که یک پاره‌خط را می‌سازد، اگر p به نقطه انتهایی یک سایت نزدیک باشد نزدیک‌ترین فاصله پاره‌خطی است که عمود بر پاره‌خط متناظر بر همان سایت است.
- اگر نقطه درونی یک سایت با خط جاروب تقاطع داشته باشد آنگاه نقطه تقاطع یک نقطه شکست است و ضلعی از نمودار ورونوی را طی نمی‌کند زیرا این نقطه تنها به یک سایت نزدیک‌ترین است. (شکل ۱۷.۱ را ببینید.)



شکل ۱۷.۱: نمونه‌ای از نمودار ورونوی پاره‌خط‌ها

فصل ۲

نمودار ورونوی روی گرافها

مقدمه

در این فصل به بررسی نمودار ورونوی روی گرافها می‌پردازیم.

(تمامی قضایای این فصل برگرفته از منبع [۲] است.)

نمودار ورونوی یکی از جالبترین ساختارها در هندسه محاسباتی است، چندین کاربرد در منابع [۳، ۴، ۵] ذکر شده است. این نمودار فضای مشخص شده G را به مناطق ورونوی تجزیه می‌کند بطوریکه برای هر عنصر p از مجموعه سایت داده شده S ، ناحیه سایت p ، شامل تمام نقاطی از G است که به p نزدیک‌تر از هر سایت دیگری از S است. در اغلب موارد G یک فضای حداقل دوبعدی است. برخی از اقسام نمودار ورونوی کاربردهای خاصی در برنامه‌ریزی مکان دارند.

فرض کنید n نقطه سایت روی رئوس یا یال‌های گراف هندسی G قرار داشته‌باشد. برای هر سایت میتوان یک عامل وزن مثبت C برای اهمیت مدلسازی و یک رنگ K (که کاربرد آن در مسائل مکانیابی تسهیلات است) اختصاص داد. بحث ما در این فصل پیرامون پیچیدگی و ارائه الگوریتم‌های کارا برای ساختار نمودار ورونوی در حالت خاص وزن دار و رنگ دار است. الگوریتم‌های مخصوصی برای مواردی که رنگ‌ها یکسان هستند، وزن‌ها یکسان هستند یا هنگامی که G یک درخت است؛ ارائه شده‌است.

در فضای اقلیدسی اورن‌همر و ادلزبرونر^۱ [۶] نشان دادند که نمودار ورونوی وزن دار n نقطه سایت

^۱Aurenhammer and Edelsbrunner

می‌تواند پیچیدگی $\Theta(n^2)$ داشته باشد و نیز الگوریتمی بهینه برای ساخت آن ارائه دادند. در حالت رنگ‌دار هر یک از سایت‌ها یکی از انواع مختلف امکانات مانند: مدارس، مراکز خرید، بیمارستان‌ها و ... را ارائه می‌دهد و با رنگ K (باتوجه به نوع آن رنگ شده باشد)، نمایش داده می‌شود.

شخص ممکن است بخواهد نزدیک‌ترین مکان یک نوع خاص از تسهیلات به خود را انتخاب کند، این مسئله را می‌توان با استفاده از نمودار ورونوی دورترین رنگ حل کرد. در فضای اقلیدسی هوتن لوچر و همکاران^۲ [۷] نشان دادند که چگونه می‌توان این دیاگرام را در مرتبه زمانی $O(kn \log n)$ ساخت.

آبلانز و همکاران^۳ [۸] الگوریتم دیگری ارائه دادند که بر پایه دیاگرام ورونوی نبود و برای یافتن کوچکترین نوار موازی محور (مستطیل) کارا بود.

در این فصل ما هر دو نوع وزن‌دار و رنگ‌دار و ترکیب آن‌ها را در محیط یک بعدی هنگامی که G یک درخت یا نمودار هندسی است؛ مطالعه می‌کنیم. لازم به ذکر است که این چارچوب یک استاندارد در زمینه مکانیابی تسهیلات هنگامی که مشتریان و تامین‌کنندگان بر روی یک شبکه اتصال که معمولاً درخت است؛ می‌باشد.

تاکنون تنها به نمودارهای ورونوی استاندارد (بدون وزن و بدون رنگ) توجه شده است. اکیب و همکاران^۴ [۵] به معرفی نمودار ورونوی استاندارد و بحث در مورد ساختار آن پرداختند. در میان مقالات می‌توان حکیمی و همکاران [۹] را نام برد که یک الگوریتم $O(en + m \log m)$ را براساس کوتاه‌ترین مسیر ارائه دادند. در این جا e و m به ترتیب نشان دهنده تعداد یال‌ها و تعداد رئوس است. اگر سایت‌ها رئوس گراف باشند؛ مهلهورن^۵ [۱۰] و ارویگ^۶ [۱۱] با استفاده از الگوریتم کوتاه‌ترین مسیر دایکسترا^۷ الگوریتمی کارا ارائه دادند که به بهبود الگوریتم‌های قبل منجر شد، این الگوریتم پیچیدگی $O(e + m + (m - n) \log(m - n))$ داشت.

در تمام طول فصل G یک درخت هندسی یا گراف با m راس و n نقطه سایت است که روی یال‌ها یا رئوس G قرار می‌گیرد.

برای ساده بودن فرض کنید G مسطح و یال‌های آن با خطوطی مستقیم مشخص شده‌اند. می‌توان فرض کرد که درخت یا گراف در یک ساختمان داده مناسب؛ مثلاً یک لیست یالی پیوندی نشان داده شود. دی‌برگ و همکاران [۱] را ببینید.

^۲Huttenlocher et al

^۳Abellanas et al

^۴Okabe et al

^۵Mehlhorn

^۶Erwig

^۷Dijkstra's Algorithm

۱.۲ نمودار ورونوی روی درختان (VD_T)^۸

فرض کنید درخت $T = (V, E, d)$ از سایز m که V مجموعه رئوس، E مجموعه یال‌ها و $m = |V| = |E| + 1$ داده شده باشد، برای هر دو نقطه x و y روی درخت، $d(x, y)$ اندازه کوتاه‌ترین مسیر از x به y است. مجموعه $S = \{p_1, p_2, \dots, p_n\}$ که n نقطه از درخت است را در نظر بگیرید، می‌خواهیم نمودار ورونوی‌ای از S را روی درخت (یعنی $VD_T(S)$) تعریف کنیم، که شامل اعضای $\{VR_T(p_1), VR_T(p_2), \dots, VR_T(p_n)\}$ است که $VR_T(p_i)$ شامل تمام x هایی از T است که به p_i نزدیک‌تر از هر سایت دیگر p_j است. (در نظر داشته باشید فاصله تعریف شده d است.) ناحیه ورونوی $VR_T(p_i)$ از یک سایت واحد p_i که زیردرختی از T است، ساخته شده است. هر برگه از زیر درخت قسمتی از یالی که متعلق به منصف p_i ها است، می‌باشد. بنابراین زیردرخت $VR_T(p_i)$ به طور منحصر به فرد توسط منصف‌های p_i مشخص می‌شود.

در صورتی که اطلاعات زیر را داشته باشیم می‌توانیم توصیف دقیقی از نمودار ورونوی ارائه دهیم:

۱. مجموعه نقاط منصف، به عنوان مثال تمام نقاط مرزی‌ای که بین نواحی ورونوی مجاور قرار دارند.

۲. برای هر نقطه منصف، موقعیت آن در درخت و یال‌های متعلق به آن یا رئوسی که با آن در ارتباط است، سایت‌های مجاور مناطق ورونوی و فاصله از آنها.

۳. برای هر راس درخت نزدیک‌ترین سایت (ها) و فاصله از آن (ها).

گر چه توصیف ما از این نمودار دقیق نیست می‌توانیم نواحی ورونوی را برای همه سایت‌ها در زمان خطی $O(m + n)$ محاسبه کنیم.

با شروع از برگ‌ها می‌توانیم ناحیه $VR_T(p_i)$ با پیمودن T محاسبه کنیم یعنی آن را در یک ساختمان داده مناسب در زمانی متناسب با توجه به تعداد یال‌های $VR_T(p_i)$ محاسبه کنیم.

قضیه ۱.۱.۲. نمودار ورونوی مجموعه n نقطه روی درخت با اندازه m دارای پیچیدگی $n + m - 1$ است و می‌توان آن را در زمان بهینه $\Theta(m + n \log n)$ محاسبه کرد.

برهان. پیچیدگی VD_T : واضح است که مناطق باید همبند باشند، مسیر منحصر به فرد از یک سایت به سایتی دیگر هرگز نمی‌تواند به منطقه دیگری تعلق داشته باشد بعلاوه ما دقیقاً $n - 1$ منصف داریم. یک منصف درخت را به n سایت شامل زیردرخت‌های مستقل تقسیم می‌کند. مثلاً l و o سایت که $n = l + o$. با بکاربردن استقرا برای زیردرختی از درخت اصلی، پیچیدگی برای درخت اصلی نتیجه می‌شود. ساختمان داده این نمودار می‌تواند توسط الگوریتم زیر حاصل شود: درگام اول درخت را وارد می‌کنیم که در نتیجه پیچیدگی $O(n)$ است. روند انقباض^۹ که در ادامه توضیح داده

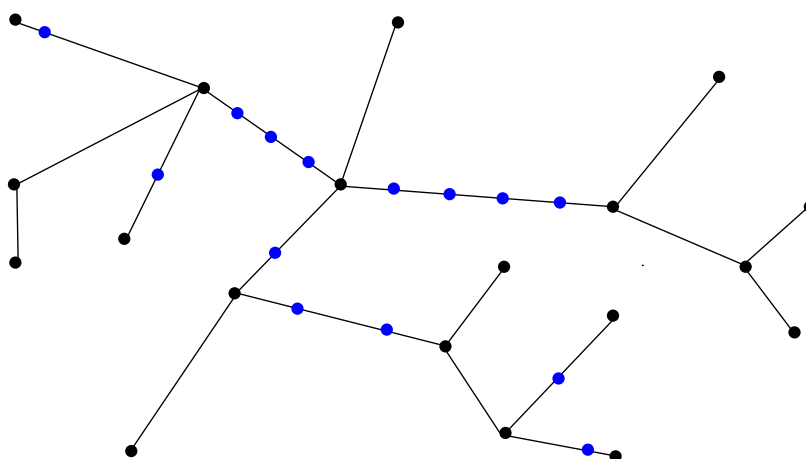
^۸نمودار ورونوی روی درخت (Voronoi Diagram on Tree) را به اختصار با VD_T نمایش می‌دهیم.

^۹contraction

می‌شود را می‌توان در زمان $O(m)$ انجام داد. سپس نمودار روی درخت فشرده شده را می‌توان در زمان $O(n \log n)$ با توجه به کوتاه‌ترین مسیر از یک سایت به گره (راس) محاسبه کرد و در زمان $O(m)$ جایگزین کرد.

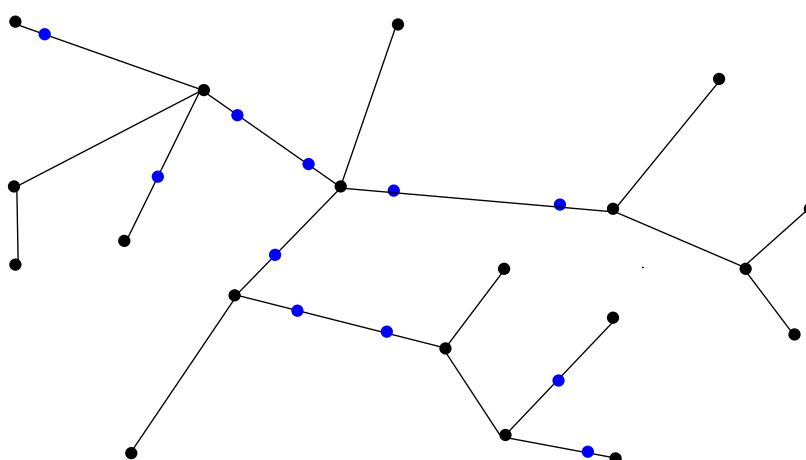
الگوریتم انقباض درخت T:

۱. سایت‌ها را برای هر یال وارد کنید. (شکل ۱۰۲ را ببینید).



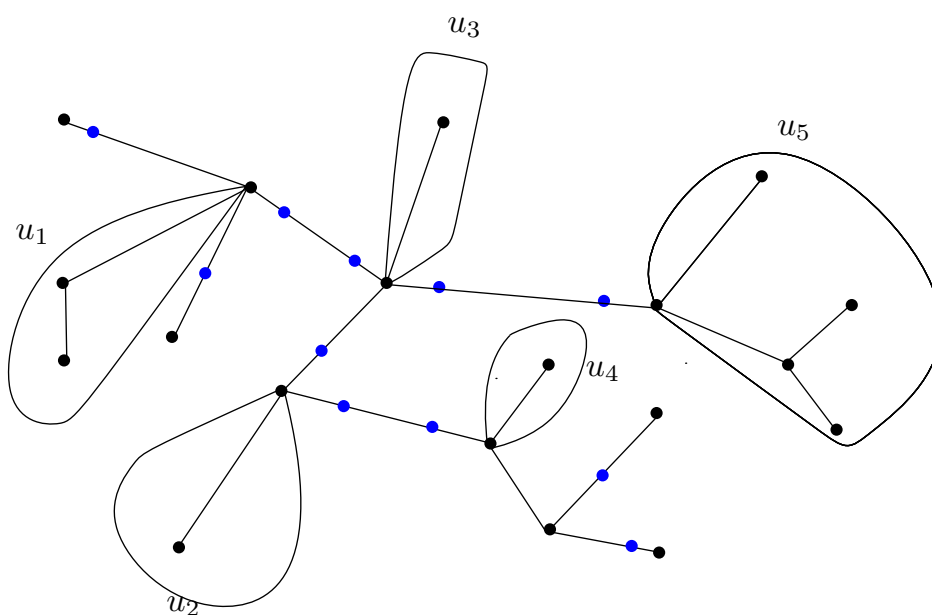
شکل ۱۰۲: وارد کردن درخت و مرتب کردن سایت‌ها

اگر بیش از دو سایت روی یک یال وجود داشته باشد، تنها دو سایت ابتدایی و انتهایی برای ادامه انتخاب می‌شوند. توجه داشته باشید تنها $O(m)$ سایت می‌توانند به مرحله بعد منتقل شوند. (شکل ۲۰۲ را ببینید.)



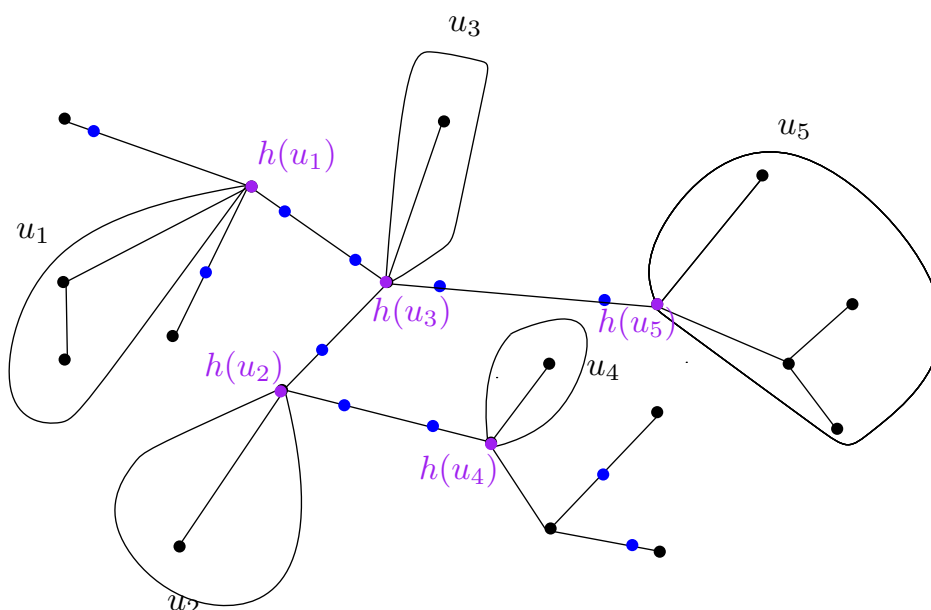
شکل ۲۰۲: حذف سایت‌های درونی

۲. محاسبه درخت پوشای S_T از $O(m)$ سایت‌های باقی مانده در T : بدین معنی که S_T کوچکترین زیردرخت همبند T است که حاوی یال‌هایی از سایت‌هاست. این را میتوان با استفاده از روش DFS روی T در زمان $O(m)$ انجام داد. فرض کنید u_i ها شامل ماکسیمال زیر درخت‌های درون $T - S_T$ باشد که شامل سایتی نیستند (شکل ۳.۲ را ببینید).



شکل ۳.۲: مشخص کردن ماکسیمال زیر درخت‌هایی که حاوی سایت نیستند. (u_i ها)

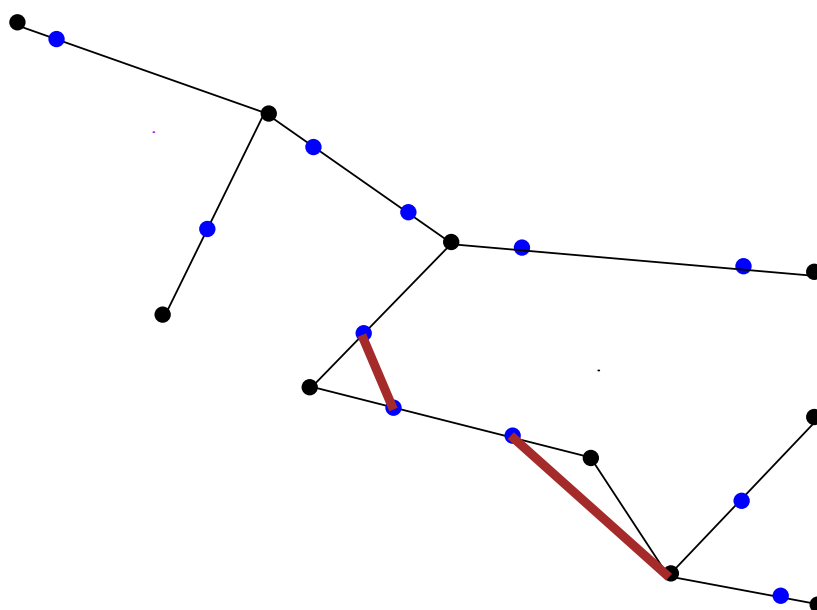
گره آویزان $h(u)$ از زیر درخت u یک گره منحصر به فرد u است که متعلق به S_T می‌باشد. (شکل ۴.۲ را ببینید)



شکل ۴.۲: مشخص کردن گره‌های آویزان برای u_i ها یعنی $h(u_i)$ ها

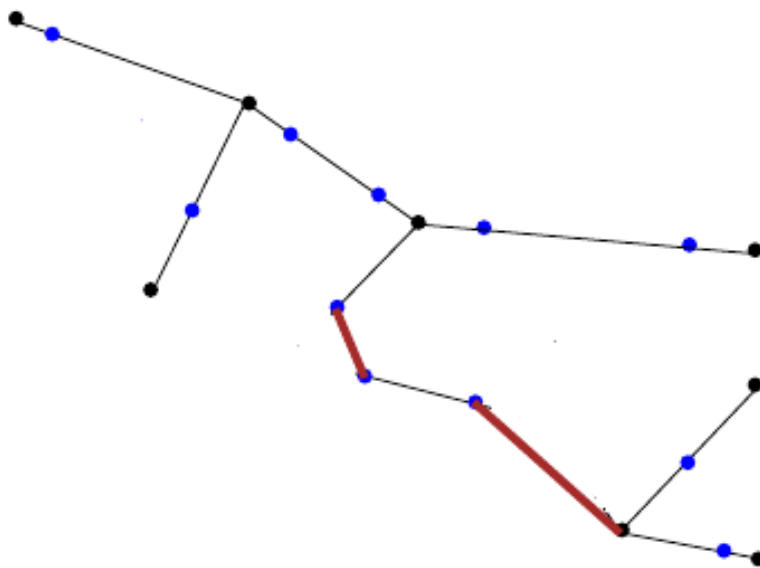
سایت گره آویزان را به هر گره از u اختصاص می دهیم.

۳. سرانجام هر حداکثر زنجیره S_T که تنها شامل رئوس درجه دو است را با یک یال جایگزین می‌کنیم. اگر چندین سایت روی زنجیره رئوس درجه دو باشد از میانبر سایت به سایت (یا سایت به گره در انتهای زنجیره) استفاده می‌کنیم. (شکل ۵.۲ را ببینید).



شکل ۵.۲: حذف ماکسیمال زیر درخت‌هایی که حاوی سایت نیستند یعنی حذف u_i ها

در نهایت شکل ۶.۲ را خواهیم داشت.



شکل ۶.۲: جایگزینی رئوس درجه دو با یک تک یال

بنابراین برخی از یال‌های منقبض شده را به دست می‌آوریم. در کل درخت S'_T از S_T با استفاده از DFS روی T در زمان $O(m)$ بدست می‌آید. خوش بختانه S'_T از اندازه $O(n)$ است. وابستگی اندازه $O(n)$ به S_T از این حقیقت ناشی می‌شود که در بدترین حالت S'_T را می‌توان به عنوان یک درخت ریشه‌دار با n برگ در نظر گرفت که هر گره داخلی حداقل دارای درجه سه است.

محاسبه VD_T :

۱. با استفاده از الگوریتم دایکسترا (همانند کاری که در [۱۱] انجام شده است) و نیز هیپ فیبوناتچی (که در [۱۲] به کار برده‌ایم) میتوان در زمان $O(n \log n)$ نزدیک ترین سایت و همچنین فاصله آن را به هر گره در S'_T مشخص کرد.

۲. سپس می‌توان نمودار کامل را روی درخت اصلی در زمان $O(m)$ جایگزین کرد. ما نزدیک‌ترین سایت‌ها و فاصله آن به هر گره T را تعیین می‌کنیم بعلاوه نزدیک‌ترین منصف‌ها را به گره‌های آویزان و نزدیک به رئوس را با S'_T جایگزین می‌کنیم.

برای کران پایین حالت خاص زیر را در نظر می‌گیریم:
 - در یک خط - (هنگامیکه گراف تک یال باشد) هر الگوریتمی که نمودار ورونوی مجموعه‌ای از n سایت را تولید می‌کند، مسئله ε -همسایگی را به کمک $n - 1$ منصف حل می‌کند.
 دورترین سایت نمودار ورونوی قسمتی از یک درخت داخل محدوده است که دارای دورترین فاصله است. توجه داشته باشید برگ درخت‌هایی که منقبض هستند چنین محدوده‌ای را به دست می‌دهند.

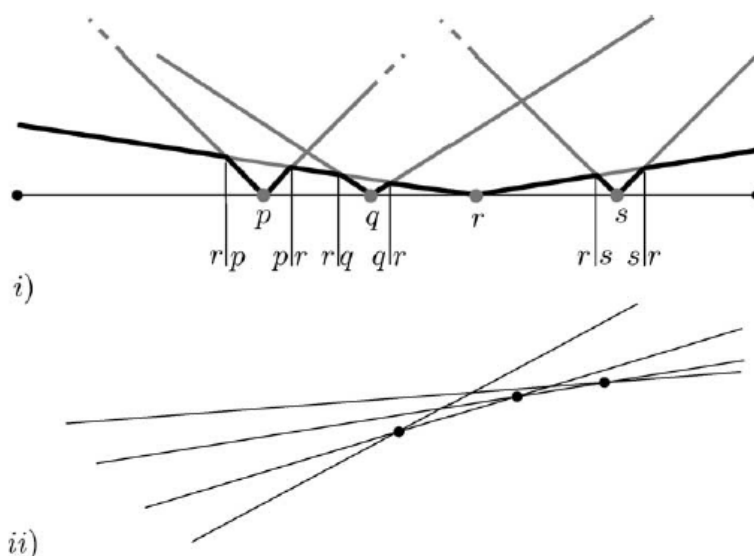
۱.۱.۲ نمودارهای ورونوی وزن‌دار روی درختان (WVD_T)^{۱۰}

نمودار ورونوی وزن‌دار S یعنی $WVD_T(S)$ را در نظر بگیرید که هر سایت $p_i \in S$ با یک فاکتور وزن W_i که به صورت صعودی مرتب شده‌اند؛ اندازه‌گیری می‌شود. به عبارت دیگر نواحی ورونوی وزن‌دار که همان $WVR(p_i)$ است به صورت زیر تعریف می‌شود:

$$WVR(P_i) = \{x \in T \mid W_i \cdot d(x, p_i) < W_j \cdot d(x, p_j) \forall j \neq i\}.$$

توجه داشته باشید که مناطق ورونوی وزن‌دار^{۱۱} ممکن است متصل نباشند و نیز زیردرخت‌ها همگی متعلق به همان ناحیه ورونوی وزن‌دار نیستند، به علاوه عامل وزن بزرگ‌تر منطقه ورونوی کوچک‌تری ایجاد می‌کند. بنابراین تعریف نمودار باید کامل شود.
 نقاط منصف را بر اساس ترتیب آنها بر روی یال مرتب می‌کنیم. نمودار ورونوی وزن‌دار بر روی یک خط معادل با پوش پایین توابع فاصله وزن‌دار f_i با شروع از p_i و شیب W_j و $-W_j$ است. شکل ۷.۲ را ببینید.

^{۱۰} نمودار ورونوی وزن‌دار روی درخت (Weighted Voronoi Diagram on the Tree) را به اختصار با WVD_T نشان می‌دهیم.
^{۱۱} نواحی ورونوی وزن‌دار (Weighted Voronoi Regions) را به اختصار با WVR نمایش می‌دهیم.



شکل ۷.۲: نمودار ورونوی وزنی در یک خط معادل با پوش پایینی توابع فاصله است، می توانیم وزن ها و سایت ها را پیدا کرده بنابراین n تابع فاصله خطی $1 - n$ منصف خواهد داشت.

هر تابع فاصله وزن دار تنها دوبار قطع می شود طبق نتایج بدست آمده از شریر و آگاروال^{۱۲} [۱۳] پیچیدگی پوش پایین توابعی که تنها دوبار قطع می شوند $O(n)$ است. از سوی دیگر وزن ها را می توان طوری یافت که $\Omega(n)$ منصف وجود داشته باشد و فقط دو نقطه تقاطع داشته باشیم.

قضیه ۲.۱.۲. نمودار ورونوی وزنی مجموعه ای از n نقطه روی یک درخت از اندازه m دارای پیچیدگی $\Theta(nm)$ است و می تواند در زمان $\Theta(mn + n \log n)$ محاسبه شود.

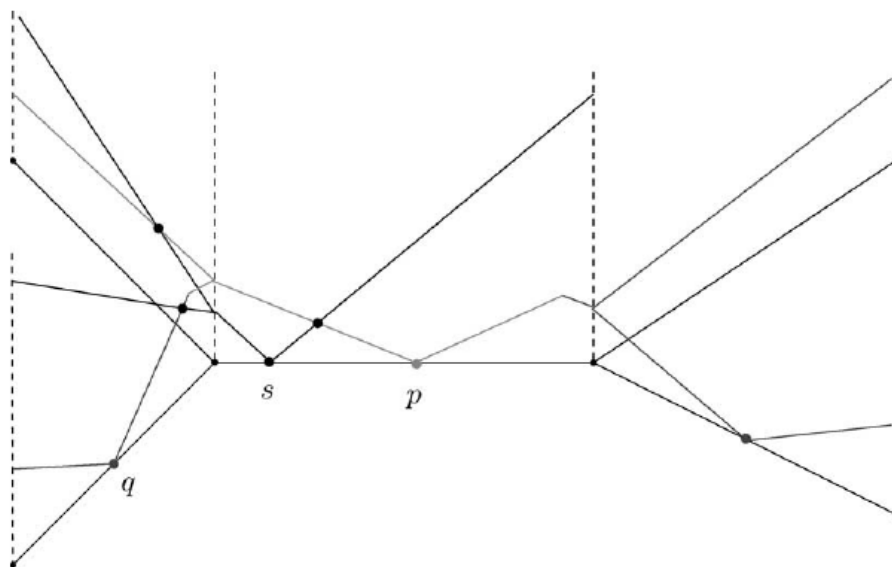
برهان. پیچیدگی WVD_T : کران بالایی این نمودار از این حقیقت ناشی می شود که پیچیدگی نمودار روی یک خط $O(n)$ است. برای کران پایین، یک تعداد خطی از منصف ها، ممکن است خارج از فاصله داده شده توسط دو راس انتهایی ظاهر شوند (شکل ۷.۲ (ii) را ببینید). بنابراین ما می توانیم تعداد n را در m که درجه یک راس افزوده شده است، ضرب کنیم. با توجه به پوش پایین ما (nm) بخش خطی داریم. زمان محاسبه از قضیه ۱.۱.۲ نتیجه می شود. توجه داشته باشید بخش هایی که در یک گره هستند با توجه به درجه گره شمارش شده اند.

^{۱۲}Sharir and Agarwal

محاسبه WVD_T :

ابتدا سایت‌ها را با وزن‌هایشان در زمان $O(n \log n)$ دسته بندی می‌کنیم. از یک الگوریتم افزایشی وارد کردن مکان‌ها به صورت صعودی استفاده می‌کنیم. از این رو سایت S وارد شده همیشه دارای یک ناحیه همبند بوده و در حداکثر $O(m)$ منصف جدید، تولید خواهد شد. با شروع از S توابع فاصله‌ای از S را تا زمانی که برای اولین بار به پوش پایینی نمودار قبلی در نقطه x برسد، دنبال می‌کنیم.

فرض کنید x متعلق به سایت p باشد، با توجه به اینکه سایت‌ها به ترتیب افزایش وزن یعنی صعودی مرتب شده‌اند، تابع فاصله p تابع فاصله S را برای تمام نقاط روی درخت که دورتر از S هستند؛ مسدود می‌کند. همه بخش‌های تابع فاصله‌ای از S فراتر از x هرگز نمی‌توانند متعلق به پوش پایین باشند. (شکل ۸.۲ را ببینید.)



شکل ۸.۲: تابع فاصله‌ای از S توسط تابع فاصله‌ای از p و q با وزن‌های پایین‌تر مسدود شده است.

فرض کنید $S_1, S_2, \dots, S_n$ نشان دهنده سایت‌هایی صعودی (از لحاظ وزنی) باشد با قرار دادن در نمودار $S_i - 1$ باید عملیات زیر را انجام دهیم:

ابتدا S_i را روی یالش در نمودار $S_i - 1$ مکانیابی می‌کنیم که برای هر i یک $O(\log n)$ نیاز است، بنابراین باید پایین‌ترین پوش $S_i - 1$ را دنبال کنیم تا زمانی که اولین تقاطعات در همه جهات را پیدا کنیم. در طی این ردیابی از منصف‌های نمودار $S_i - 1$ عبور خواهیم کرد که می‌توانند حذف شوند. توجه داشته باشید که دیگر هیچ سایتی بازدید نمی‌شود و فقط منصف‌ها و گره‌های T را بررسی خواهیم کرد. بنابراین هزینه ردیابی به تعداد منصف‌های حذف شده از گره‌ها یا یال‌های بازدید شده بستگی

خواهد داشت.

برای وارد کردن S_i فرض کنید k_i نشان دهنده تعداد منصف‌های حذف شده و m_i تعداد گره‌های بازید شده را نشان دهد و l_i تعداد منصف‌های وارد شده جدید را مشخص کند، ثابت پیچیدگی الگوریتم را به صورت $O(n \log n) + \sum_{i=1}^n l_i + k_i + m_i$ خواهیم داشت. قسمت $O(n \log n)$ به هزینه وارد کردن سایت‌ها اشاره دارد.

چون: $m_i + l_i \leq 2m$ کفایت کرانی برای $\sum_{i=1}^n k_i$ بیابیم. تعداد کل منصف‌های نمودار توسط $b := \sum_{i=1}^n (l_i - k_i)$ داده شده است، بدیهی است $b \geq 0$ و نامساوی $\sum_{i=1}^n l_i \leq mn$ را خواهیم داشت. توجه داشته باشید که $l_i \leq m$ است تا زمانی که تنها m منصف جدید داشته باشیم بنابراین می‌توانیم $\sum_{i=1}^n k_i \leq mn$ را محاسبه کنیم که کران $O(nm + n \log n)$ را برای الگوریتم می‌دهد. این کران نزدیک به پیچیدگی زمانی نمودار در $\Omega(nm)$ است و برای $m = O(1)$ متناسب با نتایج بدست آمده از نمودار ورونوی وزن دار روی خط راست است.

□

۲.۲ نمودار ورونوی دورترین رنگ (FCVD)^{۱۳}

در نظریه گراف رنگ‌آمیزی گراف یکی از حالت‌های خاص مسئله برچسب‌گذاری است. رویکرد کلی آن نظیر کردن رنگ‌هایی به یال‌ها یا رئوس است که این رنگ‌آمیزی محدودیت‌های خاصی را برآورده کند.

نمودارهای ورونوی ذکر شده در قبل را آبلانز و همکاران [۸] تعمیم داده‌اند. فرض کنید n سایت در K نوع طبقه بندی شده باشد و توسط k رنگ مدل شود که $k \leq n$. اگر p نشان دهنده سایت با رنگ C باشد، ناحیه ورونوی دورترین رنگ روی درخت ^{۱۴} $FCVR_T(p)$ توسط تمام نقاطی از T ساخته می‌شود که C دورترین رنگ و p نزدیک‌ترین سایتی است که با C رنگ آمیزی شده است. برای هر نقطه z از صفحه فرض کنید $c(i, z)$ نزدیک‌ترین محل رنگی i را نشان دهد. بنابراین طبق تعریف، z متعلق به ناحیه ورونوی $c(1, z)$ ، $c(2, z)$ ، ...، $c(k, z)$ است که $c(k, z)$ دورترین نسبت به z است.

قضیه ۱.۲.۲. نمودار ورونوی دورترین رنگ n نقطه از k رنگ روی درخت با پیچیدگی m ، پیچیدگی $2(n - k) + m + 1$ دارد و می‌تواند در زمان $O(km + n \log n)$ محاسبه شود.

□

برهان. [۲] را ببینید.

^{۱۳} نمودار ورونوی دورترین رنگ (Farthest Color Voronoi Diagram) را به اختصار با $FCVD$ نمایش می‌دهیم.
^{۱۴} ناحیه ورونوی دورترین رنگ روی درخت (Farthest Color Voronoi Region on The Tree) را به اختصار با $FCVR_T$ نشان می‌دهیم.

۱.۲.۲ دورترین رنگ نمودار ورونوی وزندار

حال مساله $WFCVD_T(S)$ را در نظر می‌گیریم که دورترین رنگ نمودار ورونوی وزندار روی یک درخت است. این مساله تعمیمی از $FCVD_T(S)$ که در آن هر سایت دارای وزن معین (معمولا به روش افزاینده) است.

قضیه ۲.۲.۲. پیچیدگی دورترین رنگ نمودار ورونوی وزندار n نقطه از k رنگ روی یک درخت، کراندار است که از بالا به $O(mn\alpha(k))$ که $\alpha(k)$ معکوس تابع آکرمن است و از پایین به $\Omega(mn)$ محدود است و می‌تواند در زمان $O(nm\alpha(k) \log k + n \log n)$ محاسبه شود.

برهان. به [۲] مراجعه شود. $\square$

۳.۲ نمودار ورونوی روی گراف‌ها $(VD_G)^{15}$

در مورد گراف‌ها مسئله کمی فرق دارد، بعنوان مثال کوتاهترین فاصله دو نقطه منحصر به فرد نیست علیرغم چنین مسائلی، استدلال‌هایی مشابه می‌توان اعمال کرد.

قضیه ۱.۳.۲. نمودار ورونوی مجموعه‌ای از n نقطه روی گراف $G = (V, E)$ که $|V| = m$ و $|E| = e$ دارای پیچیدگی $\Theta(e + n)$ است و می‌تواند در زمان $O(e + n \log n + m \log m)$ محاسبه شود.

برهان. پیچیدگی VD_G :

نمودار ورونوی روی گراف‌ها دارای پیچیدگی $O(n + e)$ است. فرض کنید n_e نشان دهنده سایت‌های یال e باشد.

هر یال $n_e + 1$ منصف دارد. یعنی به تعداد $n_e - 1$ ، بین سایت‌ها و دو منصف که ممکن است توسط یک سایت خارجی منفرد ایجاد شده باشند. در مجموع داریم: $\sum_e (n_e + 1) = n + e$. از سوی دیگر سایت‌ها را بین دو راس مجزا v_1 و v_2 تقسیم می‌کنیم که میتوان یال‌های مختلفی روی این دو ساخت. هر یال شامل یک منصف است که با $\Omega(e)$ نشان می‌دهیم. در نتیجه نمودار پیچیدگی $\Theta(n + e)$ دارد.

محاسبه VD_G : برای این منظور مجدداً از دایکسترا استفاده می‌کنیم. در ابتدا سایت‌های داخلی را در زمان $O(n \log n)$ تقسیم بندی و حذف می‌کنیم. نمیتوان از درخت پوشا استفاده کرد بنابراین مکرراً دایکسترا را برای مکان‌های باقی مانده در زمان $O(e + m \log m)$ اجرا می‌کنیم. در پایان منصف‌ها را در زمان $O(e)$ قرار می‌دهیم. روی هم رفته پیچیدگی زمانی $O(e + n \log n + m \log m)$ خواهد بود. $\square$

^{۱۵} نمودار ورونوی روی گراف‌ها (Voronoi Diagram on Graphs) را به اختصار با VD_G نشان می‌دهیم.

۱.۳.۲ نمودار ورونوی وزن دار روی گرافها (WVD_G)^{۱۶}

ایده‌های بخش نمودار ورونوی وزن دار روی درختان را به گرافها تعمیم می‌دهیم.

قضیه ۲.۳.۲. نمودار ورونوی مجموعه‌ای از n نقطه روی گراف $G = (V, E)$ که $|V| = m$ و $|E| = e$ دارای پیچیدگی $\Theta(en)$ است و می‌تواند در زمان $O(en + nm \log m + n \log n)$ محاسبه شود.

برهان. به [۲] مراجعه شود. $\square$

برای $FCVD_G$ می‌توان رویکردی مشابه بخش نمودار ورونوی دورترین رنگ روی درختان استفاده کرد. اثبات قضیه زیر در [۲] آمده است.

قضیه ۳.۳.۲. نمودار ورونوی دورترین رنگ n نقطه از k رنگ روی گراف $G = (V, E)$ که $|V| = m$ و $|E| = e$ پیچیدگی $O(ek + n)$ دارد و می‌تواند در زمان $O(ek \log k + n \log n + km \log m)$ محاسبه شود.

جدول ۱.۲ خلاصه‌ای از نتایج این فصل را نمایش می‌دهد.

جدول ۱.۲: خلاصه‌ای از نتایج

نمودار	اندازه	درخت	گراف
VD	compl	$n - 1 + m$	$\Theta(n + e)$
	comput	$\Theta(m + n \log n)$	$O(e + n \log n + m \log m)$
WVD	compl	$\Theta(mn)$	$\Theta(en)$
	comput	$\Theta(mn + n \log n)$	$O(en + n \log n + nm \log m)$
$FCVD$	compl	$2(n - k) + 1 + m$	$O(ke + n)$
	comput	$O(km + n \log n)$	$O(ek \log k + n \log n + km \log m)$
$WFCVD$	compl	$O(mn\alpha(k)), \Omega(mn)$	$O(en\alpha(k)), \Omega(en)$
	comput	$O(mn\alpha(k) \log k + n \log n)$	$O(en\alpha(k) \log k + n \log n + nm \log m)$

مقصود از compl و comput به ترتیب زمان محاسبه (Computing Time) و پیچیدگی (complexity) می‌باشد.

^{۱۶} نمودار ورونوی وزن دار روی گرافها (Weighted Voronoi Diagram On The Graphs) را به اختصار با (WVD_G) نمایش می‌دهیم.

فصل ۳

نمودار ورونوی گراف و کاربردهای آن

نمودار ورونوی یکی از معروفترین ساختارها در هندسه محاسباتی است [۱۴]. در این فصل که مطالب آن از مرجع [۱۱] گرفته شده است، نشان می‌دهیم در نظریه گراف معادلی مستقیم برای آن وجود دارد که می‌تواند در محاسبات کارا باشد.

بطور کلی دو روش برای محاسبه گراف نمودار ورونوی ارائه می‌دهیم: بهبود کران پایین مسئله و شناسایی مواردی که الگوریتم‌های ارائه شده بهینه باشند.

حافظه مورد نیاز برای یک گراف نمودار ورونوی نسبتاً متوسط است، زیرا حافظه بیشتری از خود گراف لازم ندارد. بررسی کاربردهای نمودار ورونوی می‌تواند جالب توجه باشد. بسیاری از مسائل در شبکه به کمک گراف نمودار ورونوی به راحتی حل می‌شوند که این مسائل شامل نزدیک‌ترین تسهیلات، تمام نزدیک‌ترین همسایه‌ها، نزدیک‌ترین جفت‌ها، ضد مرکز و نزدیک‌ترین نقاط است. نمودار ورونوی در اصل نواحی مجاورت را برای k سایت در صفحه (فاصله تعریف شده، فاصله اقلیدسی می‌باشد) مشخص می‌کند. الگوریتم‌های بهینه‌ای برای محاسبه نمودار ورونوی در زمان $\theta(k \log k)$ وجود دارد و این نمودار به اندازه $\theta(k)$ از حافظه را اشغال می‌کند. بسیاری از مباحث و مسائل پیرامون این موضوع در [۳] بطور مفصل بیان و تشریح شده است. ساختار نمودار ورونوی را میتوان در قالب گراف مشخص کرد زیرا برای گراف‌ها کران بالا ممکن است قابل دسترس نباشد. (عمومی‌ترین تنظیمات اغلب در مرزهای بالا مشخص می‌شوند و به اصلاح "متریک خوب" نامیده می‌شوند. [۱۵، ۱۶]) در ادامه محدودیت‌های زمان و حافظه را در نظر نمی‌گیریم و تنها به بررسی ساختار گراف نمودار ورونوی می‌پردازیم که مجاورت در گراف‌ها را بر اساس کوتاهترین مسیر بین

گره‌ها مشخص می‌کند. البته در گراف جهت‌داری که یال‌ها وزن دارند، فاصله کوتاهترین مسیر، یک متریک نیست. و نیز خواهیم دید که هم زمان اجرا و هم حافظه نه تنها به تعداد سایت‌ها بلکه به تعداد گره‌ها و یال‌ها نیز بستگی دارد.

پیرامون موضوع الگوریتم‌ها برای ساخت نمودار ورونوی گراف، مهلهورن تنها از گسترش الگوریتم کوتاهترین مسیر دایکسترا کمک گرفت. [۱۰]

فرض کنید $G = (N, E)$ یک گراف جهت‌دار با مجموعه گره N و مجموعه یال $E \subseteq N \times N$ باشد برای گره v مجموعه پیشروهای آن بستگی به پیشرو^۱های آن گره (یعنی $suc(v)$) و $deg^+(v) = |suc(v)|$ دارد.

فرض کنید $n = |N|$ و $m = |E|$ ، نگاشت $c : E \rightarrow \mathbb{R}^+$ بعنوان وزن برای هر یال، یک هزینه (مثبت) در نظر می‌گیرد. وزن یا همان هزینه برای یال $(v, w) \in E$ را با $c(v, w)$ نشان می‌دهیم و از اصطلاح "شبکه" برای چنین گراف برجسب داری استفاده می‌کنیم.

مسیر P با طول l از گره s به t یک توالی از گره‌های $v_0, v_1, \dots, v_l$ با $s = v_0$ و $t = v_l$ خواهد بود که برای $1 \leq i \leq l$ داریم: $(v_{i-1}, v_i) \in E$.

هزینه کل مسیر از رابطه $c(P) = \sum_{i=1}^l c(v_{i-1}, v_i)$ به دست می‌آید و هزینه کوتاهترین مسیر بین دو گره s و t را با $d(s, t) = \min\{c(P) \mid P \text{ مسیری بین } s \text{ و } t \text{ است}\}$ نشان می‌دهیم و اگر مسیری بین s و t وجود نداشته باشد، $d(s, t) = \infty$ تعریف می‌شود.

در گراف جهت‌دار $G = (N, E)$ نمودار ورونوی درونی برای مجموعه گره‌های $K = \{v_1, \dots, v_k\}$ که $K \subseteq N$ یک افزاز $\{N_1, \dots, N_k, U\}$ از N خواهد بود بطوریکه:

- برای هر گره $v \in N_i$ ، $d(v, v_j) \leq d(v, v_i)$ برای هر $j \in \{1, \dots, k\}$ و
- U شامل تمام گره‌های v با $d(v, v_i) = \infty$ برای هر $i \in \{1, \dots, k\}$ باشد.

این به این معنا است که نمودار ورونوی درونی بر اساس مسیرهایی است که بسوی گره‌های ورونوی هدایت می‌شوند.

فرض کنید $u = |U|$ ، گره‌های K گره‌های ورونوی نامیده می‌شوند، N_i مجموعه‌های ورونوی و U مجموعه ایزوله‌ها یا گره‌های غیر قابل دسترس نامیده می‌شود.

برای مجموعه ثابت K ، $d(v)$ را برای $\min\{d(v, v_i) \mid v_i \in K\}$ و $\bar{d}(v)$ را برای $\min\{d(v_i, v) \mid v_i \in K\}$ بکار می‌بریم.

نزدیک‌ترین گره ورونوی به گره v توسط رابطه زیر بدست می‌آید:

$$V(v) = \begin{cases} v_i, & \text{اگر وجود داشته باشد } i \in \{1, \dots, k\} \text{ بطوریکه } v \in N_i \\ \perp, & \text{در غیر اینصورت} \end{cases}$$

^۱ در ریاضیات اصطلاح پیشرو (Successor) اشاره به عدد بعدی یک عدد مشخص دارد، برای پیدا کردن پیشرو باید به عدد یک واحد اضافه کنیم.

علامت $\perp$ بیان می‌کند که V برای گره‌های ایزوله، تعریف نشده است. توجه داشته باشید نمودار ورونوی منحصر به فرد تعریف نشده است بعنوان مثال اگر برای هر $i \neq j$ داشته باشیم: $d(v, v_i) = d(v, v_j)$ آنگاه v ممکن است به هر کدام از مجموعه‌های ورونوی N_i یا N_j تعلق داشته باشد. اگر به هر دلیلی نیازمند این باشیم که نمودار ورونوی، یکتا تعریف شود می‌توانیم شرط زیر را اضافه کنیم:

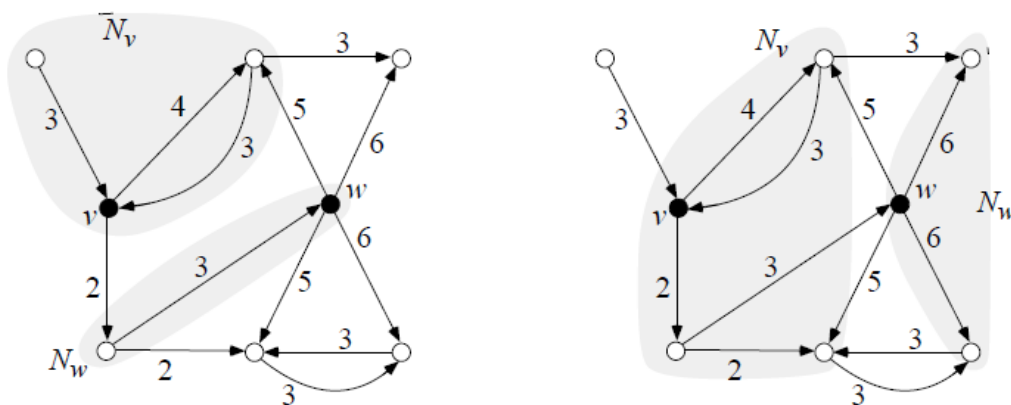
$$v \in N_i \implies i = \min\{j | d(v, v_i) = d(v, v_j)\}$$

به این معنی که اگر چندین مقدار بعنوان $\min$ وجود داشته باشد، مقدار با اندیس کمتر را بعنوان $\min$ انتخاب می‌کنیم.

در ادامه مجموعه پل‌های ورونوی E_B را تعریف می‌کنیم که به صورت اجتماع مجموعه‌های E_{ij} با $i, j \in \{0, \dots, k\}$ و $i \neq j$ است بطوریکه

$$E_{ij} = \{(v, w) \in E | v \in N_i, w \in N_j\}.$$

نمودار ورونوی بیرونی دوگان نمودار ورونوی درونی است. به این معنی که برای هر گره $v \in N_i$ رابطه $d(v_i, v) = \infty$ را به ازای هر $j \in \{0, \dots, k\}$ داریم و U شامل تمام گره‌های v با $d(v_i, v) = \infty$ برای هر $i \in \{0, \dots, k\}$ است. (شکل ۱.۳ را ببینید.)



شکل ۱.۳: نمودار ورونوی درونی و بیرونی برای گره‌های v و w

گره ورونوی خارجی از گره v با $\bar{V}(v)$ نشان داده می‌شود. هرگاه از عبارت نمودار ورونوی استفاده کنیم مقصود نمودار ورونوی درونی است. در زیر دو حالت خاص $k = 1$ که در آن $Vor(K) = \{N - U, U\}$ و $k = n$ که $N = K = \{v_1, \dots, v_n\}$ در نتیجه $Vor(K) = \{\{v_1\}, \dots, \{v_n\}, \emptyset\}$ را نادیده می‌گیریم، یعنی فرض می‌کنیم $1 < k < n$ باشد.

۱.۳ خواص اولیه

از قبل می‌دانیم برخلاف حالت هندسی گراف نمودار ورونوی ممکن است حاوی گره‌های ایزوله باشد و از این رو میتوان مجموعه U را بخش منفصل G در نظر گرفت. فرض کنید $S_1, \dots, S_\delta$ گره مجموعه مولفه‌های همبند قوی از G و $G' = (N', E')$ گراف بدون دور جهت‌دار القایی بوسیله $N' = \{S_1, \dots, S_\delta\}$ و $E' = \{(S_i, S_j) | \exists \delta \in S_i, w \in S_j : (v, w) \in E\}$ باشد. یک مولفه قوی S_i مسدود درونی (با توجه به K) است اگر و تنها اگر شرایط زیر توامان برقرار باشند:

• $K \cap S_i = \emptyset$ هیچ اشتراکی با S_i ها نداشته باشد به عبارت دیگر

• S_i پیشرو در G' نداشته باشد و یا تمام پیشروهای S_i از G' مسدود باشند.

لم ۱.۱.۳. مجموعه U از گره‌های ایزوله نمودار ورونوی $Vor(G, K)$ برابر با تمام مسدود شده‌ها با توجه به K است.

برهان. یک گره دلخواه w از مولفه‌های همبند قوی از S_i را در نظر بگیرید. ابتدا نشان می‌دهیم تمام گره‌های مسدود شده در U قرار دارند. اگر در S_i گره ورونوی وجود نداشته باشد مسیر از w به یک گره ورونوی باید از طریق یال صورت گیرد که منجر به ایجاد E' می‌شود. حال اگر S_i یال خروجی در G نداشته باشد برای هر $v_j \in K$ داریم: $d(w, v_j) = \infty$ بنابراین $w \in U$. با استقرا روی طول مسیر در G' نتیجه می‌شود که $w \in U$ ، اگر S_i یال‌های خروجی مسدود در G' باشند. از دیگر سو فرض کنید S_i مسدود نباشند، اگر $K \cap S_i \neq \emptyset$ مسیری از w به یک گره ورونوی در S_i وجود دارد بنابراین $w \notin U$ و اگر $K \cap S_i = \emptyset$ باید حداقل یک مسیر در G' از S_i به مولفه S_j شامل گره‌های ورونوی وجود داشته باشد زیرا در غیر اینصورت تمام مسیرهایی که از S_i شروع می‌شوند باید به یک گره مسدود شده منتهی شوند که با استقرا روی طول مسیر ممکن است S_i در انتها خود را مسدود کند که تناقض است پس U تنها شامل گره‌های مسدود شده است و این تکمیل کننده اثبات است. $\square$

نتیجه ۱.۱.۳. G همبند قوی (قویا همبند) است $\iff U = \emptyset$.

یک نمایش ساده از گراف نمودارهای ورونوی یک آرایه است که برای هر گره نزدیک‌ترین گره ورونوی را ذخیره می‌کند. (برای گره‌های غیر قابل دسترس $\perp$ را ذخیره می‌کند.)

لم ۲.۱.۳. نمودار ورونوی روی گراف می‌تواند در حافظه $O(n)$ ذخیره شود.

گاهی اوقات مفید است اطلاعات اضافی زیر را داشته باشیم:

۱. برای هر گره فاصله d تا نزدیک‌ترین گره ورونوی.

۲. برای هر گره ورونوی v_i مجموعه ورونوی متناظر N_i

۳. برای هر گره ورونوی v_i پل‌های ورونوی E_{ij}

واضح است که دو مورد اول نیازمند حافظه $O(n)$ و پل‌ها ممکن است تا مقدار $O(m)$ از حافظه را اشغال کنند. ما نمودار ورونوی به دست آمده را توسعه یافته^۲ می‌نامیم.

لم ۳.۱.۳. نمودار ورونوی گراف توسعه یافته را میتوان در حافظه $O(m+n)$ ذخیره کرد.

برای ذخیره صحیح گراف حافظه $O(m+n)$ مورد نیاز است بنابراین طبق این دیدگاه کران‌های حافظه فوق محدود است.

نمودارهای ورونوی درونی و بیرونی در گراف‌های بدون جهت از هم متمایز نیستند (تفاوتی میان آنها نیست) و نیز پل‌های مربوطه یکسان هستند به این معنی که $E_{ij} = E_{ji}$ برای هر $i \geq 1$ و $j \leq k$ که این یک پیش شرط مهم برای استفاده از نمودارهای ورونوی به منظور جست‌وجوی نزدیک‌ترین همسایه‌ها است.

گوییم دو گره ورونوی v_i و v_j همسایه‌های مستقیم هستند اگر $E_{ij} \neq \emptyset$ و نزدیک‌ترین همسایه به گره ورونوی v_i ، گره v_j ($v_j \neq v_i$) است که کمترین $d(v_j, v_i)$ را دارا است.

لم ۴.۱.۳. در گراف‌های بدون جهت یک نزدیک‌ترین همسایه به گره ورونوی همیشه یکی از همسایه‌های مستقیم آن است.

برهان. v_j را نزدیک‌ترین همسایه غیر مستقیم v_i در نظر بگیرید، نشان می‌دهیم این فرض منجر به تناقض می‌شود. کوتاهترین مسیر از v_j به v_i از طریق حداقل دو پل امکانپذیر است زیرا کوتاهترین مسیر باید به یک همسایه مستقیم v_j داخل و از آن خارج شود که آن را N_h می‌نامیم. فرض کنید این دو پل روی کوتاهترین مسیر از v_j به v_i ، (v, w) و (x, y) باشند بطوریکه $v \in N_j$ و $y \in N_i$ و $x \in N_h$ و نیز v_h همسایگی مستقیمی از v_i باشد. (این موقعیت در شکل ۲.۳ نشان داده شده است.)
می‌دانیم:

$$d(v_j, v_i) = d(v_j, v) + c(v, w) + d(w, x) + c(x, y) + d(y, v_i)$$

از طرفی

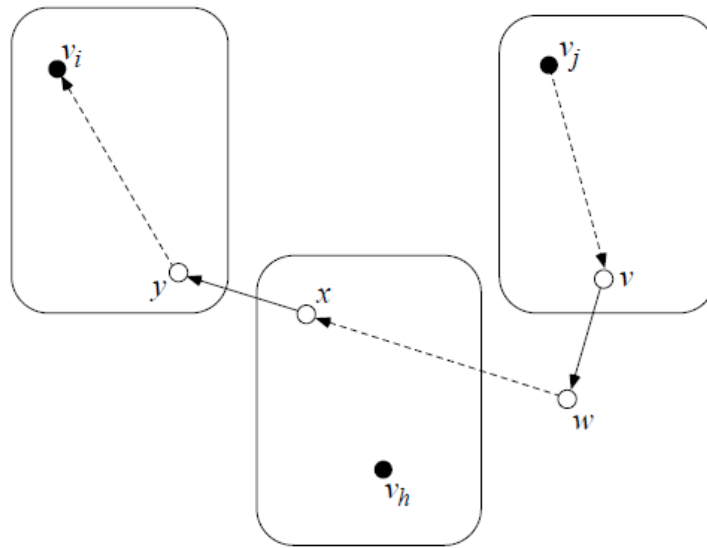
$$d(v_j, v_i) < d(v_h, x) + c(x, y) + d(y, v_i)$$

زیرا در غیر اینصورت ممکن است v_h به جای v_j نزدیک‌ترین همسایه v_i باشد. بنابراین میتوان نوشت:

$$d(v_j, v) + c(v, w) + d(w, x) < d(v_h, x)$$

یعنی x به v_j نزدیکتر از v_h است. (زیرا در گراف‌های بدون جهت داریم: $d(w, x) = d(x, w)$) که این با فرض $x \in N_h$ در تناقض است. □

^۲Extended



شکل ۲.۳: نشان دادن عینی اثبات لم فوق

۲.۳ محاسبه نمودارهای ورونوی گراف

فرض کنید در ابتدا یک کران پایین برای محاسبه نمودار ورونوی گراف ارائه دهیم. مشاهده می‌شود که $Vor(k)$ می‌تواند با نگاشت V از N به $K \cup \{\perp\}$ نمایش داده شود.

قضیه ۱.۲.۳. یک کران پایین برای محاسبه گراف نمودار ورونوی توسط $\Omega(\max(n, (n-k) \log k))$ به دست می‌آید.

□

برهان. به مرجع [۱۱] مراجعه شود.

۱.۲.۳ الگوریتم دایکسترای موازی^۳

برای ارائه الگوریتم نیازمند تعریف گراف معکوس G_r هستیم که از معکوس کردن جهت یال‌های G به دست می‌آید. رویکردی که در ابتدا بیان می‌شود اجرای خاصی از الگوریتم دایکسترا [۱۷] روی G_r با گره‌های ورونوی است. (توجه داشته باشید که نمودار ورونوی بیرونی می‌تواند به همین روش محاسبه شود با این تفاوت که جست و جو به جای G_r روی G انجام می‌گیرد.) در ادامه یک پیاده سازی مبتکرانه از الگوریتم دایکسترا را خواهیم داشت. ابتدا به معرفی چند عملگر می‌پردازیم:

^۳The Parallel Dijkstra Algorithm

عملگر	توضیحات
$\text{insert}(v, h)$	گره v را با ارزش $d(v)$ به هیپ h وارد می‌کند.
$\text{decrease}(\Delta, v, h)$	کاهش اندازه v در هیپ h به اندازه Δ
$\text{min}(h)$	کوچکترین عنصر در هیپ h
$\text{deletemin}(h)$	کوچکترین عنصر را از هیپ h حذف می‌کند.
با استفاده از هیپ فیبوناتجی [۱۲] ۴ تمام عملگرها به جز deletemin زمان $O(1)$ را می‌گیرند. (	
deletemin زمان $O(\log n)$ را به خود اختصاص می‌دهد. (	
فرض کنید هیپ h با d مرتب شده باشد، نخست گام اولیه را به صورت زیر انجام می‌دهیم:	

init :

for each $v \in N$ do

if $v \in K$ then

$d(v) := 0$ ؛ $V(v) := v$ ؛ $\text{insert}(v, h)$

else

$d(v) := \infty$ ؛ $V(v) := \perp$

بدون از دست دادن کلیت میتوان فرض کرد که عضویت مجموعه می‌تواند در زمان $O(1)$ انجام پذیرد که آن را میتوان با نگهداری اولیه K در برچسب گذاری مناسب تمام گره‌های G مشخص کرد. گره‌ها می‌توانند به روش ذکر شده گسترش یابند تا هیپ خالی شود.

زمانی ارزش گره $d(w)$ بروزرسانی می‌شود که در آن یال (v, w) بروزرسانی شده باشد علاوه بر این

$$V(w) := V(v)$$

بنابراین گسترش گره از طریق زیر حاصل می‌شود:

expand next:

$v := \text{deletemin}(h)$

mark v

$\text{scansuc}(v)$

که

$\text{scansuc}(v)$:

^۴Fibonacci Heaps

for each outgoing edge (v, w) with w not marked do

$\Delta := d(v) + c(v, w)$

if $d(w) = \infty$ then

$d(w) := \Delta$; $V(w) := V(v)$; insert(w, h)

if $d(w) < \infty$ and $\Delta < d(w)$ then

$V(w) := V(v)$; decrease(Δ, w, h)

در پایان ما یک الگوریتم دایکسترا داریم که آن را دایکسترا موازی می‌نامند. از آنجا که کوتاهترین مسیر از گره‌های ورونوی شروع می‌شود و به صورت همزمان بزرگ می‌شوند، مشاهده می‌کنیم محاسبه نمودار ورونوی گراف به مراتب بدتر از مسئله کوتاهترین مسیر تک منبع نیست.

قضیه ۲.۲.۳. الگوریتم دایکسترا موازی نمودار ورونوی گراف را در بدترین حالت در زمان $O(m + n \log n)$ محاسبه می‌کند.

برهان. فرض کنید به G_r گره جدید s و یال (s, v_i) با $c(s, v_i) = 0$ را اضافه کنیم. با اجرای الگوریتم دایکسترا با شروع از s بعد از اولین مرحله (یعنی گسترش s) تمام v_i های موجود در هیپ هزینه اولیه صفر دارند بنابراین گام‌های باقی مانده مشابه الگوریتم دایکسترا موازی است. به این معنی که خواص الگوریتم دایکسترا در حالت موازی نیز حفظ می‌شود. در پایان یک درخت کوتاهترین مسیر را به دست می‌آوریم که در آن فاصله هر گره از s برابر فاصله آن تا v_i است. بنابراین هر گره در کوتاهترین مسیر یک زیردرخت ریشه‌دار در $v_i \in k$ است که از v_i دورتر از هر گره دیگر $v_j \in K - \{v_i\}$ نیست. در صورت استفاده از هیپ فیبوناتچی اجرای الگوریتم دایکسترای فوق مشابه روش موازی به اندازه $O(m + n \log n)$ مرحله خواهد داشت. $\square$

اگر مورد زیر را در نظر بگیریم می‌توانیم الگوریتم را بهبود بخشیم:
 گره ورونوی همیشه اولین گره‌های متعلق به K است که باید گسترش یابند یعنی نیازی نیست که حتما در هیپ قرار داده شوند.
 اصلاح زیر در گام اولیه می‌تواند مفید واقع شود:

init :

for each $v \in N$ do

if $v \in K$ then

$d(v) := 0$; $V(v) := v$; mark v

else

$d(v) := \infty$; $V(v) := \perp$

for each $v \in K$ do

scansuc v

از آنجا که گره‌های ورونوی باید مشخص شوند تا از ورود آنها به هیپ جلوگیری شود، v scansuc باید در حلقه دوم دنبال شود. برای این نسخه از الگوریتم ما یک کران بهتر یافتیم زیرا تعداد عملگرهای deletemin به $n - k$ کاهش یافته و به $O(n)$ مرحله اولیه نیازمند خواهیم بود.

نتیجه ۱.۲.۳. الگوریتم دایکسترای موازی نمودار ورونوی گراف را در بدترین حالت در $O(m + n + (n - k) \log(n - k))$ محاسبه می‌کند.

۳.۳ بروزرسانی نمودارهای ورونوی

میتوان دو نوع بروزرسانی برای نمودارهای ورونوی در نظر گرفت:

۱. تغییر گراف که به معنی اضافه کردن و یا حذف گره و یال است، تغییر در c و یا

۲. تغییر مجموعه گره‌های ورونوی

مورد اول به مسئله بروزرسانی کوتاهترین مسیر روی درخت براساس تغییرات گراف اشاره دارد که در [۱۸] بطور مفصل بیان شده است. بنابراین مورد دوم را در نظر می‌گیریم. اضافه کردن گره ورونوی جدید v_{k+1} به K می‌تواند به صورت زیر انجام شود:

گره جدید را با ارزش $c = \infty$ به هیپ وارد می‌کنیم سپس با استفاده از الگوریتم دایکسترا گره‌ها را مکرراً بروزرسانی می‌کنیم که تنها آن دسته از گره‌های پیشرو که فاصله کمتری را نسبت به نمودار ورونوی قبل کسب کرده‌اند، را در هیپ قرار می‌دهیم بنابراین این گره‌ها به v_{k+1} نزدیکتر از گره‌های ورونوی قبل هستند. زمان اجرای این بروزرسانی $O(m_{k+1} + n_{k+1} + \log n)$ با $n_{k+1} = |N_{k+1}|$ و m_{k+1} نشان دهنده تعداد یال‌های (V, W) با $v \in N_{k+1}$ است. متأسفانه این مورد ممکن است باعث شود کل نمودار از ابتدا ساخته شود یعنی داشته باشیم:

$$m_{k+1} = O(m)$$

و

$$n_{k+1} = O(n)$$

. برای حذف گره ورونوی v_i از K باید به این حقیقت که ممکن است گره‌های ورونوی مجموعه N_i در دسترس نباشند، توجه داشته باشیم بنابراین تمام مقادیر d و V برای گره‌های N_i را به ترتیب به ∞ و $\perp$ بازنشانی می‌کنیم. حال می‌توانیم تمام پل‌های ورونوی گره‌های مربوطه را در هیپ قرار دهیم.

Drope Voronoi Node v_i

for each $v \in N_i$ do

$d(v) := \infty$; $V(v) := \perp$

```

for each  $(w, v) \in E_{ij}$  do
 $\Delta := d(v) + c(w, v)$ 
if  $d(w) := \infty$  then
 $d(w) := \Delta$  ;  $V(w) := V(v)$  ; insert( $w, h$ )
if  $d(w) < \infty$  and  $\Delta < d(w)$  then
 $V(w) := V(v)$  ; decrease ( $\Delta, w, h$ )
set all  $E_{ij}$  to  $\emptyset$ 
while  $h$  is not empty do
expandnext

```

به این نکته توجه داشته باشید که حلقه‌های for به G اشاره دارند در حالیکه عملگرها عملیات را روی G_r انجام می‌دهند. در این مرحله از نمودار ورونوی گسترش یافته برای یافتن پل‌های ورونوی استفاده می‌کنیم و اگر در دسترس نبود می‌توانیم پل‌های مربوطه را با یک الگوریتم دایکسترا با شروع از v_i (مشابه آنچه هنگام اضافه کردن گره‌های ورونوی انجام دادیم) برای دستیابی به رئوس ناحیه ورونوی N_i انجام دهیم که همانند حالت اضافه کردن گره ورونوی زمان اجرا می‌تواند در بدترین حالت $O(m + n \log n)$ شود.

توجه داشته باشید یافتن مجموعه پل‌های E_{ij} طولانی‌تر از حالت حذف گره نیست زیرا v_i دیگر یک گره ورونوی نخواهد بود. در نمودار ورونوی بیرونی در حلقه دوم باید تغییر جزئی زیر را اعمال کنیم:

```

for each  $(v, w) \in E_{ij}$  do
 $\Delta := \bar{d}(v) + C(v, w)$ 
...

```

$\bar{d}$ را به جای d در تمامی مراحل الگوریتم استفاده می‌کنیم و expandnext باید روی G عمل کند.

۴.۳ کاربردها

ایده ساختار داده ورونوی گراف هنگام بررسی اطلاعات شبکه ذخیره اطلاعات مطرح شد [۱۹]. در این بخش برخی از مسائل هندسه محاسباتی که می‌توان به کمک نمودارهای ورونوی گراف آنها را حل کرد مانند: نزدیکترین تسهیلات، نزدیکترین همسایگان و نزدیک‌ترین جفت‌ها، جابه‌جایی حرکت آزاد و ضد مرکزها از نظریه گراف را در نظر می‌گیریم. برای نشان دادن کاربردهای زیر فرض کنید G مدل حمل و نقل در شبکه و K گره‌های تسهیلات شامل مراکز آتش‌نشانی، بیمارستان‌ها، دفاتر پست و یا مراکز خرید باشد.

نزدیکترین تسهیلات

در این دسته از مسائل یک نقطه پرس و جو v که یک گره ورونوی است در اختیار داریم، به دنبال کوتاهترین مسیر از v به یک تسهیلات، با حداقل هزینه هستیم. موارد زیر در مکانیابی تسهیلات مورد توجه قرار می‌گیرد:

۱. کدام گره ورونوی v_i نزدیکترین به v است؟

۲. فاصله v_i از v چقدر است؟

۳. کوتاهترین مسیر از v به v_i چقدر است؟

بعنوان مثال فرض کنید تصادفی در محل v اتفاق افتاده است و ما به دنبال کوتاهترین مسیر برای دستیابی به نزدیکترین بیمارستان هستیم.

در نمودار ورونوی داده شده $Vor(G, K)$ دو مورد اول می‌توانند در $O(1)$ به ترتیب با ارزش‌های $V(v)$ و $d(v)$ اجرا شوند. اگر کوتاهترین مسیر از v به $V(v)$ به طول l باشد، بازسازی مسیر می‌تواند گره به گره انجام شود. با فرض اینکه کوتاهترین مسیر v به x را از قبل یافته‌ایم، گره بعدی روی کوتاهترین مسیر به $V(v)$ ، هر پیشرو w از x با $c(x, w) + d(w) = d(x)$ است.

به اندازه $\sum_{i=0}^{l-1} deg^+(v_i)$ مرحله داریم و در بدترین حالت زمان $O(nl) = O(n^2)$ را خواهیم داشت. اما توجه داشته باشید که در هر حال این روش سریعتر از الگوریتم دایکسترا است. بدون استفاده از نمودار ورونوی گراف موارد ذکر شده در بالا با استفاده از الگوریتم دایکسترا برای کوتاهترین مسیر نیازمند $O(m + n \log n)$ است به این معنی که از یک سو محاسبه صحیح نمودار ورونوی بدتر از محاسبه نزدیکترین امکانات به صورت مستقیم نیست و از دیگر سو باعث می‌شود در مورد نزدیکترین تسهیلات به نتایج مطلوبی دست یابیم.

نزدیکترین همسایه‌ها و نزدیکترین جفت نقاط

در یافتن جفت گره‌های نزدیک به هم به کمک الگوریتم دایکسترا، زمانی یک گره متوقف می‌شود که گره ورونوی دیگری گسترش یافته باشد. واضح است در بدترین حالت $O(km + kn \log n)$ مرحله داریم. در کل به نظر میرسد نمودار ورونوی گراف، برای گراف‌های بدون جهت کارایی بهتری دارد زیرا در این حالت نمودارهای ورونوی درونی و بیرونی بر هم منطبق هستند (پل‌های ورونوی نیز همینطور). بنابراین در گراف‌های بدون جهت می‌توانیم تمام پل‌های ورونوی را که دو آرایه C و C_d (به طول k) را نگهداری می‌کنند؛ ذخیره کنیم. هزینه کوتاهترین مسیر به شرح زیر است:

All Near Neighbors

for each $i = 1, \dots, k$ do

$C(i) := \perp$; $C_d(i) := \infty$

for each $(v, w) \in E_B$ do

$D := d(v) + c(v, w) + d(w)$

if $D < C_d(V(v))$ then $C(V(v)) := V(w)$; $C_d(V(v)) := D$

if $D < C_d(V(w))$ then $C(V(w)) := V(v)$; $C_d(V(w)) := D$

این روش با فرض وجود یک پل بین هر دو گره ورونوی است. با مقداردی اولیه $O(k)$ مرحله و با نگهداری پل‌های ورونوی $O(|E_B|)$ مرحله خواهیم داشت. بنابراین تمامی نزدیکترین همسایه‌ها می‌توانند در $O(k + |E_B|)$ یافت شوند؛ که در گراف‌های تنک تبدیل به $O(k + n)$ و در گراف‌های منظم تبدیل به $O(k + \sqrt{n})$ می‌شود. بعنوان مثال از گره‌های ورونوی میتوان دو مرکز خرید را در نظر گرفت که بیشترین رقابت قابل انتظار را دارا هستند. فهرستی از همه نزدیکترین همسایه‌ها، سایر رقبا را برای این دو مرکز خرید مشخص می‌کند.

جابه جایی ، حرکت آزاد

در صفحه اقلیدسی یال‌های نمودار ورونوی برای مجموعه‌ای از سایت‌های مانع به صورت بیشترین فاصله محلی از موانع تعریف می‌شود. حرکت در امتداد یال‌ها یک راهکار برای جلوگیری از برخورد است.

این یال‌ها به پل‌ها در مدل گراف مربوط می‌شود و به طور کلی هیچ یالی از گراف بین پل‌ها (نقاط انتهایی پل‌ها) وجود ندارد از این رو حفظ بیشترین فاصله از سایت‌ها بی معنی است زیرا همیشه نمیتوان بطور مستقیم از یک پل به پلی دیگر حرکت کرد بنابراین بهتر است در بعضی مواقع حداقل فاصله را در نظر بگیریم.

فرض کنید K مجموعه‌ای از واحدهای تصفیه آب باشد. هدف ما در این مرحله برنامه‌ریزی برای حمل و نقل کالای آلوده به منظور جلوگیری از آلودگی است. در صورت وقوع حادثه باید فاصله D از هر سایت را حداقل نگه داریم که با حذف گره‌های با $d(v) < D$ و جست و جوی کوتاهترین مسیر در گراف اصلاح شده، امکانپذیر است. اگر تعداد سایت‌های یافت شده در مقایسه با n کوچک باشد میتوان به جای محاسبه $d(v) < D$ از $d(v) \geq D$ استفاده کرد.

بعنوان مثالی از حداقل فاصله حمل و نقل، انتقال یک بیمار از یک بیمارستان به بیمارستانی دیگر را در نظر بگیرید. فاصله تا نزدیک‌ترین بیمارستان هرگز نباید از D بیشتر باشد تا در مواقع لزوم جابه‌جایی صورت گیرد.

ضد مرکزها (مرکز گریزها) و دورترین نقاط

مراکز یک مفهوم شناخته شده در نظریه گراف هستند. منظور از مراکز درونی در یک گراف مجموعه‌ای از گره‌ها با کمترین گریز از مرکز است. برای یک گره v گریز از مرکز بصورت $\max\{d(w, v) | w \in N\}$ تعریف می‌شود. معکوس مرکز و یا محیط (پیرامون) بعنوان مجموعه‌ای از گره‌ها با بیشترین گریز از

مرکز است.

در گراف جهت‌دار $G = (N, E)$ ، ضد مرکز درونی مجموعه‌ای از گره‌های $v \in N$ با بیشترین $d(v)$ و ضد مرکز درونی مجموعه‌ای از گره‌های $v \in N$ با بیشترین $\bar{d}(v)$ است.

بعنوان مثالی از ضد مرکز فرض کنید مجموعه گره‌ها ایستگاه‌های پلیس باشند و سارق حریف قصد دستبرد زدن به بانکی در آن محدوده را داشته باشد. واضح است در این مورد بیشترین فاصله تا مرکز پلیس، برای سارق حائز اهمیت است.

ضد مرکز نباید با جست و جوی دورترین نقطه یا دورترین نقطه در جست و جوی گره برای v اشتباه گرفته شود که این با اجرای الگوریتم دایکسترا از v شروع و هنگامی متوقف می‌شود که آخرین گره ورونوی به هیپ اضافه شده باشد. به بیانی دقیق‌تر ضد مرکز مطابق با جست و جوی بزرگترین دایره خالی است.

نتیجه گیری و پیشنهادات:

در این پایان نامه الگوریتم‌هایی کارآمد برای محاسبه نمودارهای ورونوی روی درخت و گراف با در نظر گرفتن احتمال فاکتور وزن را مورد بررسی قرار دادیم. این مسائل به حوزه مکانیابی تسهیلات مربوط هستند و الگوریتم‌های ذکر شده مسئله یافتن بهترین مکان برای هر یک از تاسیسات را به حداقل زمان می‌رساند.

در ادامه به معرفی نمودار ورونوی گراف پرداختیم و نشان دادیم که علاوه بر اینکه فضایی بیشتر از خود گراف نیاز ندارد (صرفه جویی در حافظه)، می‌تواند کاربردهای جالب بسیاری داشته باشد. از دیگر سو کار حاضر هندسه اقلیدسی را به چیزی تعمیم می‌دهد که می‌توان آن را هندسه گراف نامید. بعنوان پیشنهاد به جز نمودارهای ورونوی بررسی مفاهیم و الگوریتم‌هایی دیگر از هندسه محاسباتی در زمینه بهینه سازی وگراف (بعنوان مثال گراف‌های مسطح) می‌تواند جالب توجه باشد.

مراجع

- [1] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf.(2000),”**Computational Geometry: Algorithms and Applications** ” , 2nd Edition, Springer-Verlag, Berlin
- [2] E. Langetepe, F.Hurtado, R.Klein, V.Sacristán. (2004),” The weighted farthest color Voronoi diagram on trees and graphs” **Computational Geometry** , 27(1) , pp 13-26
- [3] F. Aurenhammer.(1991), ”A survey of a fundamental geometric data structure”**ACM Comput** , 23 (3), pp 345–405
- [4] F. Aurenhammer, R. Klein.(2000) , ”**Handbook of Computational Geometry** ”, Elsevier Science,North-Holland, Amsterdam, pp. 201–290
- [5] A. Okabe, B. Boots, K. Sugihara, S.N. Chiu.(2000),”**Concepts and Applications of Voronoi Diagrams**”, 2nd Edition, Wiley, Chichester, UK
- [6] F. Aurenhammer, H. Edelsbrunner.(1984),”An optimal algorithm for constructing the weighted Voronoi diagram in the plane”**Pattern Recogn** , 17 , pp 251–257
- [7] D. P. Huttenlocher, K. Kedem, M. Sharir.(1993) ,” The upper envelope of Voronoi surfaces and its applications” **Discrete Comput.Geom** , 9 , pp 267–291
- [8] B. Palop , C. Icking , E. Langetepe, F. Hurtado, L. Ma, M. Abellanas, R. Klein, V. Sacristán.(2001), ”Smallest color-spanning objects”, **Proc. 9th Annu. European Sympos. Algorithms**
- [9] E. Schmeichel, M. Labbé, S.L. Hakimi.(1992),”The Voronoi partition of a network and its implications in location theory”, **ORSAJ. Comput.** , 4 , (4) , pp 412–417
- [10] K. Mehlhorn.(1988),” A faster approximation algorithm for the steiner problem in graphs”,**Inform. Process. Lett** , 27 , pp 125–128

-
- [11] M. Erwig.(2000),” The graph Voronoi diagram with applications”,**Networks** , 36 , pp 156–163
- [12] M.L. Fredman, R.E. Tarjan.(1987),” Fibonacci heaps and their uses in improved network optimization algorithms ” **J. ACM** , 34 , pp 596–615
- [13] M. Sharir, P.K. Agarwal.(1995),”**Davenport–Schinzel Sequences and Their Geometric Applications**” , Cambridge University Press , New York
- [14] F. P.Preparata, M. I. Shamos. (2012),”**Computational geometry**”, an introduction. Springer Science , Business Media
- [15] R.Klein. (1989),”**Concrete and abstract Voronoi diagrams**” , Vol. 400, Springer Science Business Media
- [16] D.Wood , R.Klein. (1988),”**Annual Symposium on Theoretical Aspects of Computer Science**” , Springer Science Business Media , pp 281-291
- [17] J. D. Ullman , J. E.Hopcroft.(1983),”**Data structures and algorithms**”
- [18] C. Y.Pang, N.Deo. (1984),” Shortest□path algorithms: Taxonomy and annotation” **Networks** , 14(2), pp 275-323.
- [19] M. Erwig, R. H. Gutting. (1994). ”Explicit graphs in a functional model for spatial databases” **IEEE Transactions on Knowledge and Data Engineering** , 6(5),pp 787-804

Aabstract

The Voronoi diagram is an interesting construction in computational geometry, which has many applications in other sciences. This diagram is a way of dividing the plane into a number of areas, so that these areas have certain conditions and satisfy certain constraints.

In this thesis, after introducing the Voronoi diagram, we describe the complexity of this diagram on the graphs and then express the methods of finding the Voronoi diagram on trees and graphs. Some applications of voronoi diagrams are also mentioned in this thesis.

keywords : Voronoi diagram, Graph, Tree, Complexity



Shahrood University of Technology

Faculty Of Mathematical Sciences

MSc Thesis in: Operations Research

Voronoi Diagram And Its Applications On Optimization Problems

By: Pardis Rahmati

Supervisor

Jafar Fathali

Advisors

Dr. Abolfazl Poureidi

Dr. Meysam Alishahi

September 2019