دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد

گرایش مهندسی نرم‌افزار

عنوان:

الگوریتم‌های تقریبی برای برخی از مسائل هندسه محاسباتی با داده‌های نادقیق

نگارش:

حمید هماپور

استاد راهنما:

دکتر محمد قدسی

شهریور ۱۳۹۲

بسم الله الرحمن الرحیم

به نام خدا

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد

عنوان: الگوریتم‌های تقریبی برای برخی از مسائل هندسه محاسباتی با داده‌های نادقیق

نگارش: حمید هماپور

کمیته‌ی ممتحنین

استاد راهنما: دکتر محمد قدسی امضاء:

استاد مشاور: دکتر حمید ضرابی‌زاده امضاء:

استاد مدعو: امضاء:

تاریخ:

سپاس

از استاد بزرگوارم، جناب آقای دکتر محمد قدسی که با کمک‌ها و راهنمایی‌های بی‌دریغشان، بنده را در انجام این پروژه یاری داده‌اند، تشکر و قدردانی می‌کنم. از استاد محترم جناب آقای دکتر حمید ضرابی‌زاده که زحمت مطالعه و داوری را تقبل فرمودند و با پیشنهادات خود باعث بهبود این پایان‌نامه شدند، کمال امتنان را دارم. از استاد محترم جناب آقای دکتر محمدعلی آبام که از نقطه نظرات و راهنمایی‌های ایشان در انجام این پایان‌نامه بهره فراوان بردم، صمیمانه تشکر می‌کنم. از آقایان مسعود صدیقین و احسان امام‌جمعه‌زاده به دلیل یاری‌ها و راهنمایی‌های بی چشمداشت ایشان که بسیاری از سختی‌ها را برایم آسان‌تر نمودند سپاس‌گزارم. از خانواده‌ام، به‌ویژه مادر و پدرم که پشتیبان همیشگی من بوده‌اند کمال سپاس و تشکر را دارم و توفیق جبران محبت این عزیزان را از خداوند خواستارم. در پایان از دوستانم و اعضای آزمایشگاه گروه الگوریتم که در دوره‌ی تهیه این پایان‌نامه هیچ‌گاه همکاری خود را از من دریغ ننموده‌اند سپاس‌گزارم.

و تشکر از خداوند.

چکیده

در این پژوهش، پیدا کردن الگوریتم‌های تقریبی برای برخی از مسائل هندسه محاسباتی زمانی‌که نقاط نادقیق هستند را مورد بررسی قرار دادیم. برای مدل کردن نقاط نادقیق، آنها را در مدل مجموعه نقاط رنگی گسسته در نظر گرفتیم. یک مجموعه از n نقطه در فضای d-بعدی داده شده است که با m رنگ رنگ‌آمیزی شده‌اند. مسئله کوچکترین قطر رنگی، پیدا کردن m نقطه با رنگ‌های دوبه‌دو متمایز است بطوری‌که قطر این مجموعه انتخاب شده کمینه باشد. برای این مسئله یک الگوریتم تقریبی با ضریب تقریب $(1+\epsilon)$ با زمان $O(n^2(n+2^{\frac{1}{\epsilon^d}}(\frac{n}{\epsilon^d}+(\frac{1}{\epsilon^d})^2)))$ ارائه کردیم که با این کار نتایج قبلی را بهبود دادیم. در ادامه، مساله جدید پیدا کردن کران برای مسئله پوشش واحد در مدل مجموعه نقاط رنگی گسسته را بررسی کردیم. مسئله پوشش رنگی کمینه پیدا کردن m نقطه با رنگ‌های متفاوت است بطوری‌که تعداد توپ‌های واحد مورد نیاز برای پوشاندن آنها کمینه شود. بطور مشابه مسئله پوشش رنگی بیشینه انتخاب یک نقطه از هر رنگ است به نحوی‌که توپ‌های واحد مورد نیاز برای پوشاندن آنها بیشینه شود. درحالی‌که پوشش رنگی کمینه حتی در یک بعد ان‌پی-سخت است و برای آن الگوریتم تقریبی با ضریب تقریب ثابت وجود ندارد، برای آن یک الگوریتم تقریبی با ضریب تقریب $ln(m)$ ارائه کردیم. بعلاوه در یک بعد یک الگوریتم تقریبی با ضریب تقریب ثابت برای fهای ثابت نشان دادیم بطوری‌که f بیشترین تعداد تکرار یک رنگ است. برای مسئله پوشش واحد بیشینه ثابت کردیم ان‌پی-سخت است و برای آن یک الگوریتم با ضریب تقریب $\frac{1}{2}$ در یک بعد طرح کردیم.

کلیدواژه‌ها: هندسه محاسباتی، الگوریتم‌های تقریبی، داده‌های نادقیق، قطر، پوشش واحد

فهرست مطالب

فهرست شکل‌ها

فصل ۱

مقدمه

موضوع این پژوهش، نادقیقی[1] در زمینه هندسه محاسباتی[2] است. در یک جمله می‌توان گفت، در این زمینه به توسعه راه‌حل‌هایی درست، کارا و قابل اثبات برای مسائل هندسی و یا اثبات‌هایی ریاضی برای این‌که چنین جوابی وجود نداشته، پرداخته شده است. مسائل هندسی همواره در پیرامون ما وجود داشته و چنین سوال و جواب‌هایی همیشه مورد توجه همگان بوده است. با این حال، یک مانع بزرگ بر سر راه کاربردهای عملی از تکنیک‌های هندسه محاسباتی وجود نادقیقی است. در این تکنیک‌ها تصور بر این است که داده‌هایی که با آنها کار می‌شود دقیق و با دقت نامتناهی هستند و با این فرض‌ها است که تضمین‌های ریاضی داده می‌شود. اما در عمل اغلب چنین اتفاقی نمی‌افتد و داده‌ها با میزانی از خطا همراه هستند که باعث می‌شود درستی این اثبات‌ها قابل تامل باشد.

۱-۱ الگوریتم‌های هندسی

از الگوریتم‌های هندسی برای حل مسائل هندسی استفاده می‌شود. در یک مسئله هندسی، ورودی اشیایی هندسی مثل مجموعه‌ای از نقاط در فضای ۲-بعدی هستند. خروجی نیز می‌تواند یک شی هندسی ویا تنها یک عدد ساده باشد. اگرچه در ابتدایی‌ترین مسائل هندسی از الگوریتم‌های هندسی استفاده می‌شده است اما این مسائل دارای اجرایی خسته کننده بود و انسان‌ها تنها قادر به یافتن جواب برای مجموعه‌های

[1] Imprecision

[2] Computational geometry

کوچک بودند. در این زمان با معرفی کامپیوترها محبوبیت مسائل هندسی تحت تاثیر قرار گرفت. درست است که کامپیوترها هنوز هم برای دیدن جواب‌ها از قابلیت بالایی برخوردار نیستند اما با سرعت بالای خود قادر به حل مسائل بزرگتری هستند. در تمامی علوم برای بررسی برخی از جنبه‌های واقعیات، از مدل‌های هندسی استفاده می‌شود سپس تحلیل، پردازش و محاسبات دیگر صورت می‌گیرد. کامپیوترها با استفاده از الگوریتم‌های هندسی قادر به اجرای تمامی این فرایند هستند.

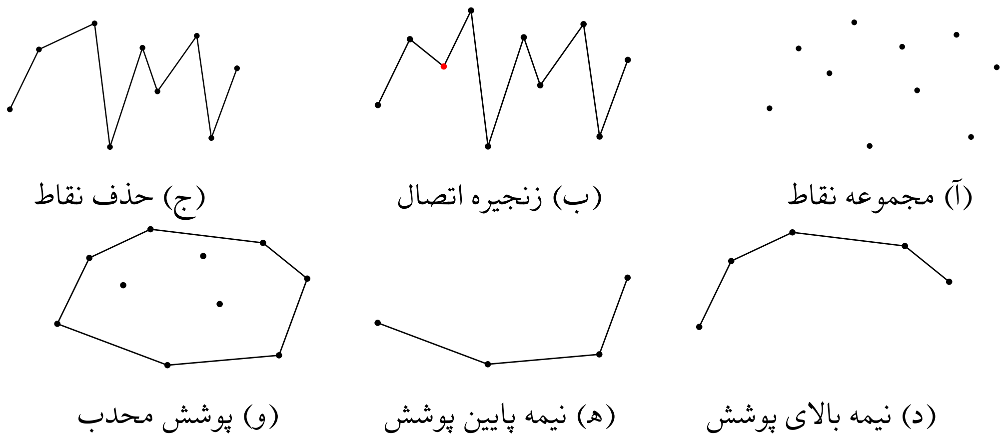
به عنوان یک مثال کلاسیک از الگوریتم‌های هندسی، پوشش محدب[۳] یک مجموعه از نقاط روی صفحه را در نظر بگیرید. یک زیرمجموعه از صفحه، محدب نامیده می‌شود اگر برای هر دو نقطه p و q در این مجموعه، تمام نقاط روی پاره‌خط pq جزء همین مجموعه باشند. برای یک مجموعه از نقاط داده شده، پوشش محدب کوچکترین مجموعه‌ی محدب است که این نقاط را شامل می‌شود. برای مثال مجموعه‌ی شامل ده نقطه در شکل ۱-۱آ را در نظر بگیرید. برای یک انسان دیدن پوشش محدب شامل این نقاط آسان است. اما اگر این مجموعه به جای ده نقطه، از میلیون‌ها نقطه تشکیل شده بود یا اینکه به جای یک مجموعه ده‌تایی، صدهزار مجموعه ده‌تایی داشتیم، محاسبه پوشش محدب بدون استفاده از کامپیوترها کاری دشوار بود. کامپیوترها برای محاسبه از الگوریتم‌ها استفاده می‌کنند. یک الگوریتم ممکن برای حل این مسئله در زیر خلاصه شده است.

- **مرحله اول:** نقاط را از چپ به راست مرتب کنید.
- **مرحله دوم:** نقاط را با استفاده از یک زنجیره به هم متصل کنید، مانند شکل ۱-۱ب.
- **مرحله سوم:** هر نقطه‌ای که در آن زنجیره چرخش به چپ [۴] دارد را حذف کنید، همانند شکل ۱-۱ج.
- **مرحله چهارم:** مرحله سوم را تکرار کنید تا نقطه‌ای با چرخش به چپ باقی نماند.
- **مرحله پنچم:** زنجیره باقی‌مانده نیمه‌ی بالایی پوشش محدب است، شکل ۱-۱د.
- **مرحله ششم:** مرحله ۱ تا ۴ را تکرار کنید با این تفاوت که به جای حذف نقاط با چرخش به چپ، نقاط با چرخش به راست [۵] را حذف کنید.

[۳] Convex hull

[۴] Turn left

[۵] Turn right



شکل ۱-۱: اجرای الگوریتم پوشش محدب روی یک مجموعه شامل ده نقطه داده‌شده در صفحه.

- **مرحله هفتم:** زنجیره باقی‌مانده نیمه‌ی پایینی پوشش محدب است، شکل ۱-۱ه.
- **مرحله هشتم:** دو زنجیره را به هم متصل کنید تا نتیجه نهایی حاصل شود، مثل شکل ۱-۱و.

در اکثر موارد از کاربردهای زیادی که الگوریتم‌های هندسی دارند، یک رابطه آشکار بین مدل هندسی فضای اقلیدسی[۶] و دنیای واقعی که در آن زندگی می‌کنیم وجود دارد. برای مثال، در طراحی کامپیوتری [۷] یک شیء از دنیای واقعی در کامپیوتر طراحی می‌شود. در گرافیک[۸] دنیای ۳-بعدی مصنوعی ایجاد می‌شود که در آن تصاویر ایجاد شده طوری پردازش می‌شوند تا همانند دنیای واقعی به نظر برسند، همانطورکه از چشم یک انسان دیده می‌شود. در سیستم‌های اطلاعات جغرافیایی [۹] معمولا از یک مدل ۲-بعدی از پوسته زمین، ذخیره شده در کامپیوترها برای تجزیه و تحلیل استفاده می‌شود. در طراحی مدارهای مجتمع [۱۰] طرح یک مدار الکتریکی قبل از چاپ به صورت هندسی مورد مطالعه قرار می‌گیرد. در زیست‌شناسی مولکولی [۱۱] ، مولکول‌های پیچیده و تعامل آنها توسط یک مدل ۳-بعدی مورد کاوش قرار می‌گیرد.

همچنین در برخی از حوزه‌های کاربردی، از فضای هندسی استفاده می‌شود اما مستقیما با دنیای

[۶] Euclidean space
[۷] Computer Aided Design
[۸] Computer Graphics
[۹] Geographic Information Systems
[۱۰] Integrated Circuit Design
[۱۱] Molecular Biology

واقعی مرتبط نیستند. در پایگاه داده‌ها، ورودی‌ها با ویژگی‌های متعدد عددی (مانند سن، حقوق و دستمزد کارکنان) را می‌توان به عنوان نقاط در یک فضای هندسی با ابعاد بالاتر به نمایش درآورد. در هر یک از این کاربردها، محاسبات در حوزه هندسی مربوطه با استفاده از الگوریتم‌های هندسی انجام می‌شود. در اصل بسیاری از این الگوریتم‌ها توسط متخصصان زمینه‌های مربوطه طراحی شده است. اما علیرغم پیشرفت کامپیوترها با پیچیده‌تر شدن کاربردها، الگوریتم‌های مورد استفاده به اندازه کافی سریع نیستند یا حتی در وضعیت‌های خاص جواب صحیح را محاسبه نمی‌کنند. برای حل این مشکل در الگوریتم‌های هندسه محاسباتی، از تحلیل الگوریتم‌ها استفاده می‌شود.

اولین تجزیه و تحلیل سیستماتیک روی الگوریتم‌های هندسی توسط مایکل شامس انجام گرفت. در سال ۱۹۸۷ ، پایان‌نامه او تحت عنوان *هندسه محاسباتی* [۱] زمینه‌ای را ایجاد کرد که بعد از آن بسیار محبوب شد. اغلب با مطالعه خواص مسائل هندسی اثبات الگوریتم‌ها امکان‌پذیر است و به این ترتیب در عمل الگوریتم‌های سریعتری طراحی می‌شوند.

۱-۲ نادقیقی

جهت استفاده از الگوریتم‌های هندسی برای مطالعه و تحلیل دنیای اطراف، ابتدا باید به مشاهده و ثبت اطلاعات در رابطه با جهان پیرامون بپردازیم، سپس این اطلاعات باید در یک مدل معنی‌دار ریاضی به نمایش در آید و در نهایت در دسترس کامپیوتر قرار گیرد. به این ترتیب، در حالت ایده‌آل، یک توصیف درست از پیرامون برای کامپیوترها فراهم می‌شود و می‌توان محاسبات را انجام داد. اما در عمل، در جریان این داده‌ها چندین محل برای تحت تاثیر قرار گرفتن اطلاعات وجود دارد که منجر به نتایجی نادرست و مبهم از توصیف پیرامون می‌شود.

۱-۲-۱ منابع خطا

در بالاترین سطح، می‌توانیم سه نوع نادقیقی را متناظر با سه فاز متفاوت در جریان داده‌ها از یک کاربرد هندسی مشخص کنیم.

۱. فاز مشاهده اشیاء دنیای واقعی و برگرداندن آنها به اشیاء هندسی.

۲. مدل کردن مشاهدات در یک فضای هندسی.

۳. محاسبات در مدل هندسی.

دلیل اول و سرراست‌ترین منبع خطا زمان مشاهده پیرامون به‌وجود می‌آید. داده‌های ورودی توسط ابزار اندازه‌گیری [۱۲] جمع‌آوری می‌شوند. ابزارها مانند ارتفاع‌سنج‌ها [۱۳] ، اسکنرهای لیزری [۱۴] یا گیرنده‌های جی‌پی‌اس [۱۵] مختصات یک نقطه را برمی‌گردانند در حالی‌که این تنها یک تقریب از محل واقعی آن است. این اتفاق بیشتر به دلیل آن است که ابزارها به اندازه کافی دقیق نیستند و میزان نادقیقی به کیفیت ابزارهای اندازه‌گیری برمی‌گردد. در حقیقت از لحاظ علمی امکان اندازه‌گیری دقیق یک مکان بدون دگرگونی به عنوان یکی از اثرات مشاهده کننده[۱۶] غیرممکن است.

امروزه یکی از روش‌های مشهور برای مشخص کردن مکان‌ها، استفاده از جی‌پی‌اس است. در این سیستم یک مکان با استفاده از اندازه‌گیری فاصله‌ها تا تعدادی ماهواره که در مدار زمین قرار دارند صورت می‌گیرد، شکل ۱-۲آ. این فاصله‌ها کره‌هایی با شعاع مشخص را معین می‌کنند و به این ترتیب ابزار اندازه‌گیری باید در نقطه اشتراکی این کره‌ها باشد. اگرچه فاصله تا یک ماهواره را نمی‌توان به صورت دقیق اندازه‌گیری کرد اما می‌تواند هر مقداری با یک بازه خطای مشخص داشته باشد. به عنوان نتیجه، در واقع کره‌ها به پوسته‌هایی با یک عرض مشخص تبدیل می‌شوند و محل گیرنده می‌تواند هر نقطه‌ای در ناحیه‌ای که از تقاطع این پوسته‌ها بدست می‌آید باشد. خطا در این نتایج برای محل گزارش شده می‌تواند تا بالای ۲۰ متر باشد، هرچند با اضافه کردن تعداد ماهواره‌ها و استفاده از یک تکنیک به نام جی‌پی‌اس افتراقی [۱۷] به طور قابل ملاحظه‌ای می‌توان این خطا را کاهش داد.

به عنوان یک مثال دیگر، اطلاعات ارتفاع مورد استفاده برای ساخت مدل‌های دیجیتال زمین [۱۸] اغلب توسط هواپیماهای که بالای زمین به پرواز در می‌آیند و از فاصله تا زمین نمونه‌گیری می‌کنند بدست می‌آید. برای مثال این اندازه‌گیری‌ها با استفاده از تکنیکی به نام تشخیص نور و محدوده [۱۹]

[۱۲] Measuring equipment

[۱۳] Altimeters

[۱۴] Laser scanners

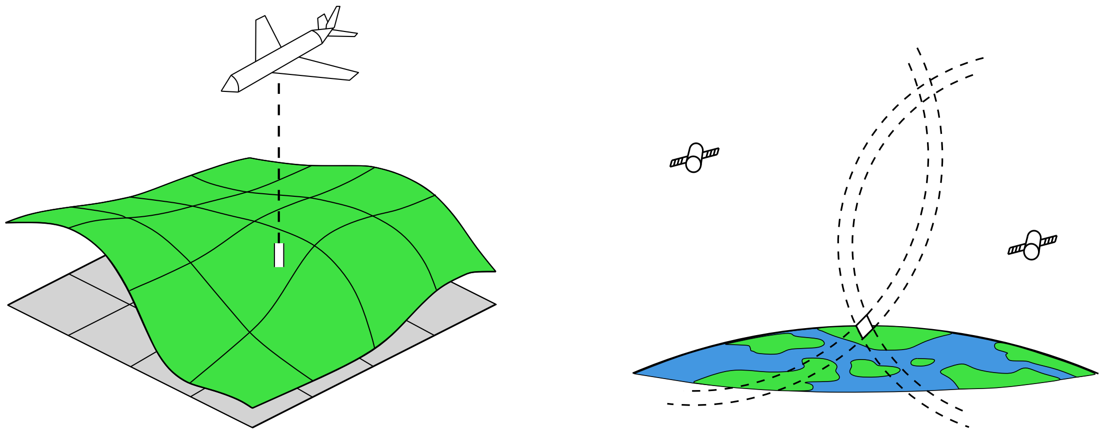
[۱۵] GPS receivers

[۱۶] Observer effect

[۱۷] Differential GPS

[۱۸] Digital Terrain Models

[۱۹] Light Detection and Ranging (LIDAR)



شکل ۱-۲: (آ) محاسبه یک مکان با جی‌پی‌اس از طریق محاسبه فاصله تا تعدادی از ماهواره‌ها در مدار زمین. (ب) در یک سیستم لیدار، یک هواپیما بالای سطح زمین به پرواز در می‌آید و برای محاسبه فاصله‌اش تا زمین اشعه‌هایی لیزری به صورت عمودی به سمت پایین ساطع می‌کند.

محاسبه می‌شوند، شکل ۱-۲ب. داده‌های جمع‌آوری شده با استفاده از این تکنیک نادقیق هستند، برای مثال به دلیل عدم دقت در ارتفاع پرواز هواپیما و یا به دلیل وجود اشیاء دیگر روی زمین. در مدل‌های دیجیتال زمین با کیفیت بالا که توسط سازمان زمین شناسی و اکتشافات معدنی آمریکا [۲۰] توزیع شده است داشتن خطاهای عمودی بالای ۱۵ متر غیر معمول نیست [۲] اگرچه مجموعه داده‌های دیگر به دقتی در حد سانتی‌متر رسیده‌اند.

دوم و شاید اساسی‌ترین منبع عدم دقت، در مدل هندسی فعلی مورد استفاده برای توصیف جهان پیرامون نهفته است. معمولا فضایی که در آن زندگی می‌کنیم را به صورت یک فضای ۳-بعدی اقلیدسی [۲۱] مدل می‌کنیم. با این وجود این یک بحث قدیمی است که آیا مدل هندسی رسمی که توسط اقلیدس ارائه شده واقعا یک توصیف واقعی از دنیای حقیقی است؟ امانوئل کانت [۲۲] (۱۸۰۴ - ۱۷۲۴) برای اولین بار در *نقد عقل محض*[۲۳] [۳] استدلال می‌کند که هندسه اقلیدسی بر پایه مشاهدات نیست و بنابراین هیچ دلیلی وجود ندارد که چرا باید مدل خوبی باشد. مدت کوتاهی پس از آن، هندسه‌های

[۲۰] The United States Geological Survey

[۲۱] 3-dimensional euclidean space

[۲۲] Immanuel Kant

[۲۳] Critique of Pure Reason

مختلف جایگزینی معرفی شدند. از زمانی که آلبرت انیشتین [۲۴] (۱۹۵۵ - ۱۸۷۹) نظریه نسبیت [۲۵] [۴] خود را منتشر کرد به طور عام اعتقاد بر این است که در واقع ساختار جهان از مدل اقلیدسی پیروی نمی‌کند. با این حال، آنچه که باید به عنوان مدل *واقعی* باشد و اینکه آیا می‌توان یک چنین مدلی را ساخت، بحثی دشوار و اغلب فلسفی است که در این بحث نمی‌گنجد.

زمانی که هندسه اقلیدسی را برای مدل کردن دنیای واقعی انتخاب می‌کنیم، همیشه نمی‌توانیم اشیاء دنیای واقعی را به مدل‌های هندسی تبدیل کنیم. یک مشکل درون‌یابی [۲۶] است: حتی اگر بتوانیم برای برخی از کمیت‌ها اندازه‌گیری دقیق انجام دهیم، چون تعداد نقاط بی‌شمار است نمی‌توانیم آن‌ها را در هر نقطه از دامنه اندازه‌گیری کنیم. در نتیجه، داده‌ها همواره درون‌یابی می‌شوند. به عنوان یک مثال دیگر، در یک برنامه جی‌آی‌اس [۲۷] ممکن است بخواهیم خط ساحلی را با منحنی‌های هندسی نمایش بدهیم. در حالیکه خط ساحلی به طور کامل تعریف نشده است و خوش تعریف نیست: با جزر و مد به جلو و عقب حرکت می‌کند و حتی در مقیاس کوچکتر از زمان با هر موجی تغییر می‌کند. این یک عدم دقت ذاتی از مدل را نشان می‌دهد. و یا یک ناحیه از زمین را در نظر بگیرید که به عنوان جنگل طبقه‌بندی شده است. مرز دقیق یک چنین جنگلی چیست؟ چه تعداد از درختان نیاز است تا یک گروه از درخت‌ها جنگل نامیده شود؟ این چنین مشکلات طبقه‌بندی نیز موجب عدم دقت می‌شود. ژانگ و گودچیلد به طور گسترده در مورد این عدم دقت در کتاب خودشان به نام *عدم قطعیت در اطلاعات جغرافیایی*[۲۸] [۵] بحث کرده‌اند.

در نهایت، سومین منبع خطا در ساخت کامپیوترهای امروزی است. کامپیوترها اطلاعاتی را که استفاده می‌کنند بصورت رشته‌های بیتی ذخیره می‌کنند که ذاتا یک مدل گسسته است. در حالیکه هندسه اقلیدسی روی یک فضای پیوسته از نقاط کار می‌کند. در دو بعد این فضا $\mathbb{R}^2$ است، فضایی از همه زوج‌ها شامل دو عنصر از $\mathbb{R}$ که اعداد حقیقی نامیده می‌شود. برای کار کردن با این فضا، اعداد و ارقام واقعی با نزدیکترین نقطه‌ای که می‌تواند توسط کامپیوتر توصیف شود تقریب زده می‌شود. از سوی دیگر، الگوریتم‌های هندسی فرض مدل ماشین با دسترسی تصادفی واقعی [۲۹] را دارند: فرض می‌کنند

[۲۴] Albert Einstein

[۲۵] Theory of Relativity

[۲۶] Interpolation

[۲۷] Geographic information system

[۲۸] Uncertainty in Geographical Information

[۲۹] Real RAM (Random Access Machine)

که یک کامپیوتر می‌تواند عملیات‌های دقیق را بطور مستقیم روی مقادیر واقعی انجام دهد. کامپیوترها داده‌ها را به مقادیری که خود می‌توانند با آنها کار کنند تبدیل می‌کنند. این می‌تواند مشکلات زیادی را برای اجرای الگوریتم‌ها ایجاد کند.

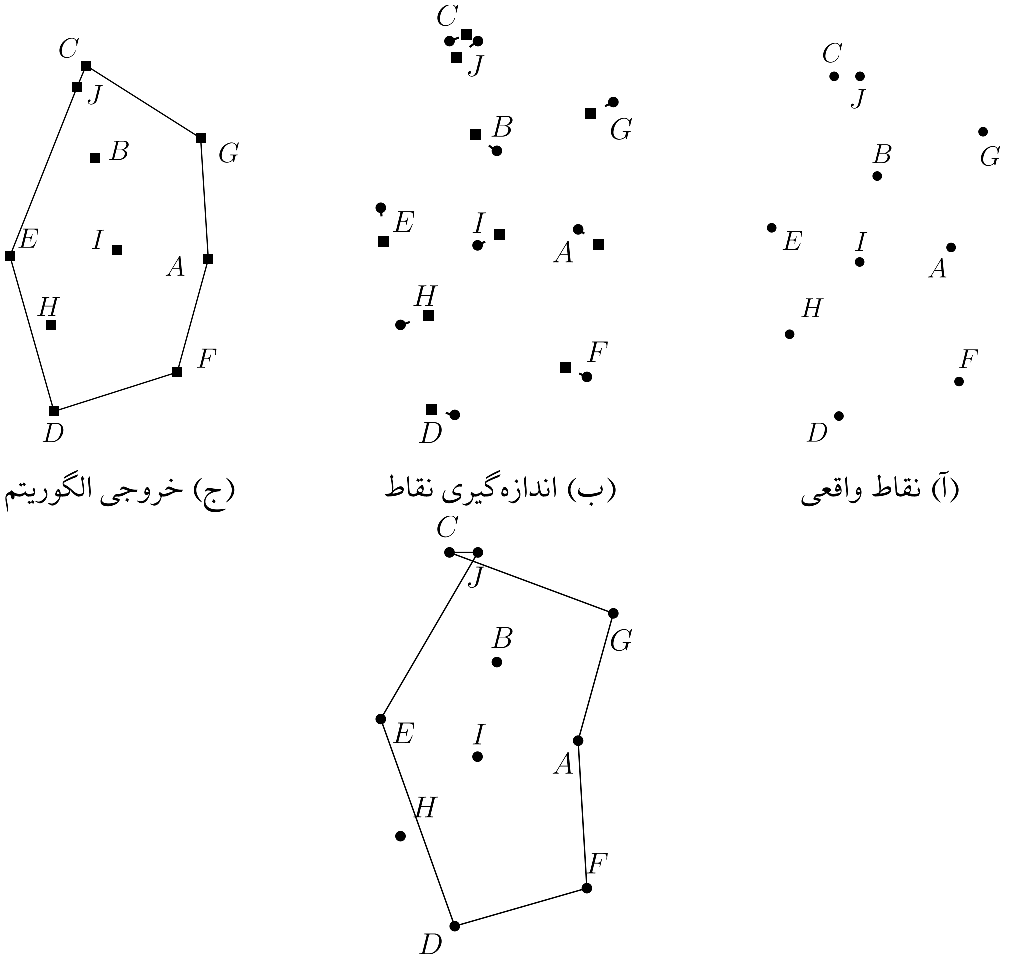
۱-۳ اثرات نادقیقی

منابع نادقیقی ذکر شده در بالا و اجرای الگوریتم‌های منطقا درست در کاربردهای واقعی، باهم می‌توانند باعث بوجود آمدن نتایج غیرمنتظره شوند. این نتایج از مقادیر خروجی با کمی تغییر یا بیشتر شدن زمان محاسبات تا پاسخ‌های کاملا اشتباه متفاوت هستند.

برای مثال، وضعیت زیر را در نظر بگیرید. فرض کنید یک مجموعه از نقاط روی صفحه داریم و می‌خواهیم پوشش محدب این نقاط را بدست بیاوریم. برای محاسبه، از الگوریتم کاملا صحیح، کارا و اثبات شده استفاده می‌کنیم، مثلا الگوریتمی که در بخش ۱-۱ شرح داده شد. با این حال ممکن است در نهایت با یک پوشش محدب نادرست مواجه شویم.

شکل ۱-۳آ را در نظر بگیرید که یک مجموعه از نقاط را در واقعیت نشان می‌دهد و آنها را از A تا J نام‌گذاری شده است. سپس این نقاط را با استفاده از یک وسیله اندازه‌گیری که خطای کمی دارد اندازه‌گیری می‌کنیم. این عملیات یک مجموعه دیگر از نقاط را به عنوان نتیجه برمی‌گرداند، همانطور که در شکل ۱-۳ب میبینید شبیه به مجموعه واقعی اما کمی متفاوت است. این مجموعه از نقاط در حافظه کامپیوتر ذخیره و به عنوان ورودی الگوریتم استفاده می‌شوند. حالا الگوریتم را با این ورودی به اجرا در می‌آوریم و یک پوشش محدب صحیح از نقاط اندازه‌گیری شده به عنوان خروجی برگردانده می‌شود، شکل ۱-۳ج. می‌توانیم این پوشش محدب را مختصرا با بیان ترتیب نقاط روی آن توصیف کنیم. در این مورد ترتیب $D-E-J-C-G-A-F-D$ خواهد بود. با این حال، اگر این ترتیب را در فضای واقعی نگاه کنیم، قبل از اینکه نقاط به دلیل خطای اندازه‌گیری دچار دگرگونی شوند، در شکل ۱-۳د می‌بینیم که این یک پوشش درست نیست. چندضلعی پاسخ، محدب نیست (در نقطه A)، تمام نقاط را شامل نمی‌شود (نقطه H داخل پوسته نیست) و حتی پوسته خودش را قطع کرده است (نزدیک نقاط C و J).

در این مثال، فرض کردیم نادقیقی ناشی از تجهیزات اندازه‌گیری است، اما تمام منابع خطا که در



(آ) نقاط واقعی (ب) اندازه‌گیری نقاط (ج) خروجی الگوریتم

(د) خروجی واقعی

شکل ۱-۳: خروجی الگوریتم صحیح روی نقاط اندازه‌گیری شده درست است اما خروجی واقعی غلط است.

قسمت قبلی توضیح داده شد می‌توانند اثرات مشابه داشته باشند. بسته به اینکه جلوتر خروجی چطور پردازش می‌شود، عواقب ناشی از نادقیقی می‌تواند از یک خطای کم در مقادیر عددی تا هنگ کردن برنامه‌ها، به دلیل فرض سازگاری از سوی الگوریتم‌های بعدی برای داده‌های ورودی خود که در این مورد صادق نیست، تاثیرات منفی داشته باشد.

در اینجا باید توجه داشت، هرچند در مثال بالا که بطور خاص ساخته شده برای نشان دادن اینکه چه اشتباهاتی می‌تواند رخ دهد، نادقیقی در عمل همیشه باعث بوجود آمدن مشکلات جدی نمی‌شود. در واقع به همین دلیل است که اغلب استفاده از الگوریتم‌ها حتی با نادیده گرفتن نادقیقی موفقیت‌آمیز است. با این وجود، هرچند به ندرت، ممکن است اشتباه به‌وجود بیاید. بعلاوه این خیلی واضح است که در برخی از کاربردها این وضعیت قابل قبول نیست و به تضمین‌های علمی در یک مدل ریاضی احتیاج

داریم.

۱-۴ مدل‌های نادقیقی در ورودی‌های هندسی

ورودی در بسیاری از مسائل هندسه محاسباتی، یک مجموعه P از n نقطه در $\mathbb{R}^d$ است. مدل نادقیق برای هر نقطه یک توزیع یا یک محدوده برای محل قرارگیری آن مشخص می‌کند. به این ترتیب انواع مدل‌های مطرح برای نقاط نادقیق به شرح زیر خواهد بود:

- به طور کلی، ما این اطلاعات را به عنوان نقاط **نامشخص**[۳۰] $\mathcal{P} = \{P_۱, P_۲, ..., P_n\}$ توصیف می‌کنیم. در این مدل مکان هر نقطه با استفاده از یک تابع توزیع احتمال μ_i تعریف شده است (به عنوان مثال توسط یک توزیع گاوسی[۳۱]). این مدل کلی، مدل‌هایی که در ادامه مطرح می‌شوند را شامل می‌شود اما به دلیل مشکلات محاسباتی برخاسته از کلیت، کمتر مورد استفاده قرار می‌گیرد.

- یک مدل محدودتر که در این پژوهش استفاده شده است، مدل **مجموعه نقاط رنگی گسسته** [۳۲] است که در آن مکان هر نقطه از میان یک مجموعه متناهی از نقاط انتخاب می‌شود. برای راحتی فرض می‌کنیم هر نقطه دقیقا k مکان ممکن دارد که دامنه تابع توزیع احتمال را برای نقطه فراهم می‌کند. به بیان دیگر، هر نقطه نامشخص P_i در یکی از مکان‌های $\{P_{i۱}, P_{i۲}, ..., P_{ik}\}$ قرار دارد. به جز در مواردی که با صراحت مشخص شده، احتمال وجود نقطه در هر مکان یکسان و برابر با $\frac{۱}{k}$ خواهد بود. همچنین می‌توانیم برای هر مکان $p_{i,j}$ یک وزن $w_{i,j}$، به عنوان احتمال حضور P_i در $p_{i,j}$، اختصاص بدهیم طوری‌که برای تمام iها $\sum_{j=۱}^{k} w_{i,j} = ۱$. برای نام‌گذاری این مدل از لفظ رنگی به این دلیل استفاده کردیم که مجموعه مکان‌های ممکن برای قرارگیری یک نقطه نادقیق در این مدل را می‌توان با مکان‌هایی با یک رنگ یکسان نمایش داد.

نقاط رنگی کاربردهای زیادی دارند. برای مثال نقش زیادی را در دیتابیس‌ها [۳۳] [۶، ۷، ۸، ۹،

[۳۰] Uncertain

[۳۱] Gaussian distribution

[۳۲] Indecisive

[۳۳] Databases

۱۰، ۱۱، ۱۲]، یادگیری ماشین [34] [۱۳] و شبکه‌های حسگر [35] [۱۴] دارند که در آن تعداد محدودی کاوش از روی یک مجموعه داده‌های خاص جمع‌آوری شده است که هر یک به‌طور بالقوه یک نماینده برای محل واقعی یک نقطه هستند.

- یک نقطه **غیردقیق** [36] نقطه‌ای است که مکان آن به صورت دقیق مشخص نیست اما مکانش به یک ناحیه محدود شده است. در فضای یک بعدی این نواحی به صورت بازه مدل می‌شوند اما در فضای دو-بعدی به صورت نواحی هندسی مدل می‌شوند. یک مدل اولیه برای تعیین میزان عدم دقت در داده‌های هندسی، مدل ϵ-هندسه [37] است که توسط گویباس و همکارانش [۱۵] ارائه شد بطوریکه برای هر نقطه مکانش در جایی درون یک توپ به شعاع ϵ حدس زده می‌شود. سادگی این مدل کاربرده‌های زیادی را در هندسه فراهم کرده است. گوبیاس و همکارن [۱۶] چند ضلعی‌های قویا محدب [38] را تعریف کردند: چند ضلعی‌هایی که تضمین می‌شود محدب باقی بمانند حتی اگر راس‌های چند ضلعی به اندازه ϵ دچار آشفتگی شوند. حلد و میچل [۱۷] و لوفلر و سون‌ینک [۱۸] پیش‌پردازش روی داده‌های غیردقیق را مورد مطالعه قرار دادند تا زمانی‌که محل دقیق نقاط مشخص شد، محاسبات در زمان کمتری انجام شوند.

یک مدل پیچیده‌تر برای نقاط غیردقیق می‌تواند به این صورت باشد که شعاع نادقیقی ϵ برای همه نقاط یکسان نباشد یا حتی اَشکال دیگری برای ناحیه نادقیقی استفاده شود. با توجه به این مدل منبع نادقیقی می‌تواند متفاوت باشد، نادقیقی مستقل در ابعاد مختلف ورودی و غیره. این آزادی در مدل همراه با راه‌حل‌های الگوریتمی پیچیده خواهد بود، با این حال نتایج زیادی در دسترس است. ناگیا و توکورا [۱۹] اجتماع و اشتراک تمام پوشش‌های محدب ممکن را برای مشخص کردن محدوده جواب بدست آوردند. ون‌کرولد و لوفلر [۲۰] مسائل مربوط به محاسبه کوچکترین و بزرگترین مقدار ممکن برای برخی اندازه‌گیری‌های هندسی مانند قطر، شعاع کوچکترین دایره پوشا، در حالتی که محل قرارگیری نقاط به نواحی داده شده محدود است را در فضای دو-بعدی مطالعه کردند. کروگر [۲۱] بعضی از این نتایج را به ابعاد بالاتر گسترش داد.

[34] Machine Learning

[35] Sensor networks

[36] Imprecise

[37] ϵ-geometry

[38] Strongly convex polygons

توجه کنید، اگرچه نقاط غیردقیق همراه با توزیع احتمالی نیستند اما می‌توانیم اینطور استدلال کنیم که آنها حالت خاصی از نقاط نامشخص هستند، با این فرض که مثلا یک توزیع یکنواخت روی نواحی وجود دارد و سپس در مورد بزرگترین و کوچکترین مقدار خروجی از یک تابع که احتمال وقوع آن غیر صفر است سوال می‌کنیم که معادل مشخص کردن حد بالا و پایین در مدل کلاسیک است.

- یک نقطه **تصادفی**[۳۹] p دارای مکان ثابت است اما تنها با یک احتمال ρ وجود دارد. این نقاط به طور طبیعی در بسیاری از سناریوهای پایگاه داده ظاهر می‌شوند [۷، ۹]. اخیرا کاموسی، چان و سوری [۲۲، ۲۳] مسائل هندسی در مدل نقاط تصادفی و گراف‌های هندسی با یالهای تصادفی را مطرح کرده‌اند. این نقاط تصادفی را می‌توان نقاط نامشخص در نظر گرفت که در آن توزیع احتمال حاکم بر نقاط نامشخص، شامل احتمال معینی برای نبود نقاط است.

۱-۵ برخورد با نقاط نادقیق

یک راه‌حل برای این مشکل این است که الگوریتم‌هایی طراحی کنیم که برای داده‌های نادقیق باشند و به طور صریح با داده‌های نادقیق محاسبات را انجام دهند. برای یک مجموعه P از نقاط دقیق معمولا به دنبال یک اندازه $\mu(P)$ مانند قطر، عرض و ... هستیم. حال سوال این است که برای یک مجموعه P از نقاط نادقیق داده شده، چطور این اندازه‌ها را محاسبه کنیم (به طور مثال، قطر، عرض، ...) و یا حتی برای ساختارهای هندسی مثل پوشش محدب، نمودار ورونوی یا مثلث‌بندی دلانی آنها چگونه خواهند بود. هرچند ساختار پوشش محدب، نمودار ورونوی یا مثلث‌بندی دلانی و ... به محل دقیق نقاط وابسته است اما می‌توانیم جواب‌های ممکن برای آنها را به‌دست بیاوریم. برای مثال به رویکردهای زیر توجه کنید:

- تنها یک بخش خاص از جواب را محاسبه کنید.
- محاسبه خصوصیات تمام خروجی‌های ممکن.
- محاسبه عدم دقت در نتایج.

[۳۹] Stochastic

- خروجی را به عدم دقت وابسته کنید.
- خروجی دقیق را برای یک نمونه از وروی محاسبه کنید.
- نقاط نادقیق ورودی را پیش‌پردازش کنید تا با دریافت نقاط دقیق در آینده، خروجی را سریعتر محاسبه کنید.

در این پژوهش رویکرد ما محاسبه حد بالا و پایین برای خروجی خواهد بود. مثلا در رابطه با مساله پوشش محدب می‌توان این سوال را مطرح کرد که کوچک‌ترین یا بزرگ‌ترین پوشش محدب برای نقاط نادقیق داده شده چیست.

۱-۶ ساختار پایان‌نامه

در بخش مقدمه ابتدا با نادقیقی و منابع آن در هندسه محاسباتی آشنا شدیم. در ادامه تعدادی مثال از اثرات نادقیقی و اجرای الگوریتم‌های هندسی با ورودی‌های نادقیق را دیدیم که منجر به خروجی‌هایی نادرست می‌شد. سپس با مدل کردن و برخورد با نقاط نادقیق آشنا شدیم. در ادامه و در فصل دوم تاریخچه‌ی برخورد با نقاط نادقیق در مدل مجموعه نقاط رنگی گسسته و نتایج به‌دست آمده از آن را بررسی می‌کنیم. سعی شده است در فصل دوم روند نتایج به دست آمده و مساله‌های مختلف مطرح در این زمینه معرفی شوند. همچنین بعضی نتایج اساسی مرتبط با موضوع نشان داده شده‌است. در فصل سوم مسئله کوچکترین قطر رنگی در مدل مجموعه نقاط رنگی گسسته را بررسی می‌کنیم و با استفاده از مفهوم هسته مجموعه[۴۰] برای آن یک الگوریتم تقریبی با ضریب تقریب $(1+\epsilon)$ ارائه می‌کنیم. در فصل چهارم مسئله پوشش واحد را در مدل مجموعه نقاط رنگی گسسته مورد مطالعه قرار دادیم و نشان دادیم هر دو مسئله در هر دو حالت کمینه و بیشنه حتی در یک بعد ان‌پی-سخت است. برای اثبات در حالت کمینه، برای کاهش از مسئه پوشش مجموعه‌ای و برای حالت بیشینه از مسئله ۳-صدق‌پذیری استفاده کردیم. برای حالت کمینه نشان دادیم الگوریتم تقریبی با ضریب تقریب ثابت وجود ندارد و برای آن یک الگوریتم تقریبی با ضریب تقریب $ln(m)$ ارائه کردیم. بعلاوه در یک بعد یک الگوریتم تقریبی با ضریب تقریب ثابت برای fهای ثابت نشان دادیم بطوری‌که f بیشترین تعداد تکرار یک رنگ است.

[۴۰] Core-set

برای حالت بیشینه یک الگوریتم با ضریب تقریب $\frac{۱}{۲}$ در یک بعد طرح کردیم. در نهایت در فصل پنجم، به نتیجه‌گیری و ارائه پیشنهادهایی برای ادامه این کار پرداخته می‌شود.

فصل ۲

مروری بر کارهای انجام شده

همانطور که قبلا اشاره شد یکی از راه‌کارهای اتخاذ شده برای برخورد با نقاط نادقیق، پیدا کردن انتخابی از بین تمام انتخاب‌های ممکن برای نقاط است که یک ویژگی خاص (مثل محیط، فاصله، قطر و ...) از یک ساختار هندسی (مثل پوشش محدب، کوچکترین درخت پوشا و ...) برای نقاط انتخاب شده، کوچکترین و بزرگترین در بین همه انتخاب‌های ممکن باشد. در ادامه کارهای صورت گرفته در این زمینه را معرفی می‌کنیم.

۲-۱ کارهای پیشین در مدل مجموعه نقاط رنگی گسسته

یک مجموعه شامل n نقطه و $k \leqslant n$ رنگ داده شده و هر نقطه با یک رنگ همراه است. به یک ناحیه انتخاب رنگی [۱] گفته می‌شود اگر شامل حداقل یک نقطه از هر رنگ باشد. برای ناحیه‌های مختلف همراه با یک تابع هدف به دنبال کوچکترین و بزرگترین انتخاب‌های رنگی ممکن از آن نوع (ناحیه به همراه تابع هدف) هستیم.

در واقع اصلی‌ترین خصیصه در این‌گونه مسائل این است که عناصر ورودی آنها چندین نسخه (یا کپی) دارند اما خروجی این مسائل محدود به این است که برای هر یک از اعضا دقیقا یکی از کپی‌ها استفاده شود. که در اینجا هر یک از اعضاء به همراه کپی‌هایش (مکان‌های ممکن برای یک نقطه نادقیق)

[۱] Color selection

را با یک رنگ نشان می‌دهیم. برای این‌گونه از مسائل کاربردهای زیادی می‌توان متصور شد. برای مثال مسائل مکان‌یابی تسهیلات [۲] را در نظر بگیرید. فرض کنید k نوع مختلف از امکانات موجود است مثلا مدرسه، پست‌خانه و سوپرمارکت و ... که با n نقطه رنگی بطوریکه هر نوع از امکانات با یک رنگ مجزا در صفحه مدل شده‌اند. یک هدف اولیه انتخاب مکانی است که در نزدیکی آن از هر کدام از امکانات حداقل یکی موجود باشد.

۲-۱-۱ قطر، نزدیک‌ترین زوج، کوچکترین دایره پوششی، عرض

فن و همکاران [۲۴] یک الگوریتم تصادفی کارا با زمان $O(n^{1+\epsilon})$ برای بزرگترین قطر رنگی ارائه کردند که در آن ϵ به اندازه داخواه می‌تواند کوچک باشد. ژانگ و همکاران [۲۵] یک الگوریتم با زمان اجرای $O(n^m)$ برای کوچکترین قطر رنگی نشان دادند که در آن m نشان‌دهنده تعداد رنگ‌ها است. فلایشر و ژو [۲۶، ۲۷] نشان دادند این مسئله برای متریک‌های L_1 و L_∞ در زمان چندجمله‌ای قابل حل است درحالیکه برای متریک‌های L_p $(p = 2, 3, ...)$ ان‌پی-سخت است. بعلاوه یک الگوریتم تقریبی کارا با ضریب تقریب ثابت برای این مسئله ارائه کردند.

یک سوال طبیعی در این مدل کوچکترین دایره رنگی است (منظور از آن، پیدا کردن یک انتخاب رنگی است بطوریکه کوچکترین دایره پوششی این انتخاب رنگی در بین تمام انتخاب‌های رنگی ممکن کوچکترین باشد)، می‌توان آنرا توسط پوشش فوقانی [۳] از پوسته‌های نمودار ورونوی [۴] پیدا کرد که توسط هوتنلوچر و همکاران [۲۸] و شریر و آگاروال [۲۹] ارائه شده است. الگوریتم آنها برای محاسبه جواب، $O(kn \log n)$ زمان صرف می‌کند. آبلانس و همکاران [۳۰] نشان دادند که نمودار دورترین ورونوی رنگی با شرط $k \leqslant \frac{n}{2}$ از مرتبه $O(nk)$ است. سپس الگوریتمی‌هایی برای ساخت نمودار دورترین ورونوی رنگی، کوچکترین دایره پوشای رنگی، کوچکترین مستطیل پوشای رنگی و کوچکترین عرض رنگی دربرگیرنده نقاط در تمام جهت‌ها ارائه کردند. داس و همکاران [۳۱] یک الگوریتم برای کوچکترین کوریدر [۵] پوشای رنگی در زمان $O(n^2 \log n)$ و یک الگوریتم برای برای کوچکترین مستطیل پوشای رنگی با جهت دلخواه در زمان $O(n^3 \log n)$ ارائه کردند.

---

[۲] Facility location

[۳] Upper envelop

[۴] Voronoi diagram

[۵] Corridor

خانتیموری و همکاران [۳۲] نوع دیگری از این مسائل، برای زمانی که هدف پوشاندن نقاط رنگی با دو شیء مشابه است، را مورد بررسی قرار دادند. آنها مسئله پوشاندن نقاط رنگی در یک بعد توسط دو بازه را مورد بررسی قرار دادند (به بیان دیگر دو بازه‌ای را پیدا می‌کنند به‌طوری‌که از هر رنگ حداقل یک نقطه در حداقل یکی از این بازه‌ها موجود باشد). برای حالتی که بازه بزرگتر، کوچکترین اندازه را داشته باشد الگوریتمی در زمان $O(n^2 \log n)$ با فضای مصرفی $O(n)$ معرفی کردند.

فن و همکاران [۲۴] مسئله پیدا کردن حد بالا برای نزدیک‌ترین زوج در یک بعد را مورد مطالعه قرار دادند و نشان دادند این مسئله حتی در یک بعد هم ان‌پی-سخت است و هیچ الگوریتم تقریبی با ضریب تقریب بهتر از $\frac{1}{2}$ در زمان چندجمله‌ای وجود ندارد. بعلاوه کانسگرا و همکاران [۳۳] نشان دادند این مسئله نه تنها در یک بعد بلکه حتی زمانی که از هر رنگ حداکثر ۳ نقطه داریم بازهم ان‌پی-سخت است و زمانی که برای از هر رنگ کمتر مساوی دو نقطه داریم و نقاط در فضای d-بعدی هستند الگوریتمی با زمان $O(n^2 \log n)$ ارائه کردند. همچنین مسئله پیدا کردن حد پایین برای بیشترین فاصله را مورد بررسی قرار دادند و یک الگوریتم تقریبی با ضریب تقریب ۲ با زمان $O(n^3.k^3 \log nk)$ زمانی‌که نقاط در یک بعد قرار دارند ارائه کردند.

۲-۱-۲ ساختارهای هندسی

لی و همکاران [۳۴] مسئله درخت پوشای کمینه تعمیم یافته [۶] را مورد بررسی قرار دادند. در این مسئله یک گراف که راس‌های آن به مجموعه‌هایی دوبه‌دو مجزا تقسیم شده‌اند داده شده و مسئله پیدا کردن یک درخت با کمترین هزینه است بطوریکه شامل دقیقا یک راس از هر مجموعه باشد. لی و همکاران نشان دادند این مسئله ان‌پی-سخت است. یک مسئله دیگر در این حوزه، فروشنده دوره‌گرد با همسایگی[۷] است که در آن n ناحیه (همسایگی) بعنوان ورودی داده شده و به دنبال کوتاه‌ترین دور هستیم که تمام ناحیه‌ها را بازدید کند. از آنجایی که این مسئله تعمیم مسئله فروشنده دوره‌گرد است بدیهی است که ان‌پی-سخت است. دومیترسکو و میچل [۳۵] یک الگوریتم با ضریب تقریب ثابت برای مسئله در حالت کلی و یک الگوریتم با ضریب تقریب $(1+\epsilon)$ برای مسئله زمانیکه ناحیه‌ها دیسک‌های واحد مجزا هستند ارائه کردند. میچل [۳۶] برای حالتی‌که ناحیه‌ها هر شکل فت [۸] مجزایی در صفحه هستند یک

[۶] Generalized Minimum Spanning Tree

[۷] TSP with neighborhoods

[۸] Fat

الگوریتم با ضریب تقریب $(1+\epsilon)$ در زمان چندجمله‌ای ارائه کرد. الباسینو و همکاران [۳۷] مسئله فروشنده دوره‌گرد گروهی اقلیدسی [۹] را بررسی کردند، یک مجموعه P شامل n نقطه در در صفحه و یک مجموعه از m ناحیه‌ی بسته داده شده است بطوریکه هرکدام حداقل شامل یک نقطه از P است و می‌خواهیم کوتاه‌ترین دوری را پیدا کنیم که حداقل یک نقطه از هر ناحیه را ملاقات کند. آنها در حالتی که ناحیه‌ها مجزا و α-فت هستند یک الگوریتم با ضریب تقریب $(1+\alpha/9/1)$ ارائه کردند. بعلاوه یک الگوریتم با ضریب تقریب ثابت برای مسئله در حالتی‌که ناحیه‌ها مجزا نیستند ارائه کردند. فن و همکاران [۲۴] مسئله درخت پوشای کمینه را در مدل مجموعه نقاط رنگی گسسته در نظر گرفتند و ثابت کردند محاسبه هر دو کران پایین و بالا برای این مسئله ان‌پی-سخت است. بعلاوه در [۳۳، ۳۸] نشان داده شده مسئله درخت پوشای کمینه حتی زمانی که از هر رنگ حداکثر دو نقطه وجود دارد ان‌پی-سخت است.

فن و همکاران [۲۴] کوچکترین پوشش محدب رنگی را در نظر گرفتند و نشان دادند این مسئله ان‌پی-کامل است. بعلاوه دو الگوریتم تقریبی با ضریب تقریب π در زمان $O(n^2+nk\log k)$ و دیگری با ضریب تقریب $\sqrt{2}$ در زمان $O(min\{n(n-k)^2), nk(n-k)\})$ ارائه کردند. بعلاوه کانسگرا و همکاران [۳۳] با تعمیم مفهوم ϵ-کرنل به آواتار ϵ-کرنل (مفهوم آواتار معادل نقاط رنگی است) در فضای d-بعدی توانستند الگوریتم‌هایی با ضریب تقریب $(1+\epsilon)$ در زمان چندجمله‌ای برای کوچکترین پوشش محدب رنگی، کوچکترین ابرباکس رنگی موازی محورهای مختصات با کم‌ترین حجم و محیط و کوچکترین قطر رنگی ارائه کنند. الگوریتمی که ما در بخش بعدی برای مسئله کوچکترین قطر ارائه می‌کنیم دارای زمانی کمتر با ضریب تقریب مشابه برای این مسئله است.

[۹] Euclidean Group TSP

فصل ۳

هسته مجموعه برای مجموعه نقاط رنگی گسسته

۳-۱ هسته مجموعه

یکی از تکنیک‌های کلاسیک در توسعه الگوریتم‌های تقریبی استخراج میزان کوچکی از داده‌های مناسب از اطلاعات و انجام محاسبات روی این داده‌ها است. برای مسائل در زمینه هندسه محاسباتی که در آن داده‌های ورودی یک مجموعه از نقاط است، این سوال به پیدا کردن یک مجموعه کوچک از نقاط (هسته مجموعه) کاهش پیدا می‌کند به طوریکه بتوان محاسبات مورد نظر را روی هسته مجموعه انجام داد.

کارهای قابل توجهی در اندازه‌گیری‌های مختلف روی توصیف‌های متفاوت از مجموعه P شامل n نقطه در فضای d-بعدی صورت گرفته است. ما به این مدل اندازه‌گیری‌ها به عنوان معیارهای اندازه‌گیری از P یاد می‌کنیم. در واقع این اندازه‌گیری‌ها روی مجموعه P، یا یک مشخصه خاص از P را محاسبه می‌کند یا یک شکل (در صورت امکان غیرمحدب) که مجموعه P را می‌پوشاند. به طور مثال برای اولی می‌توان محاسبه k-امین بزرگترین فاصله بین دو نقطه از P و برای دومی کوچکترین دایره پوشاننده P را اشاره کرد.

مسائل برازش اشکال که از مسائل پایه در هندسه محاسباتی، بینایی رایانه‌ای، یادگیری ماشین، داده

کاوی و تعدادی حوزه دیگر است، در ارتباط خیلی نزدیک با معیارهای اندازه‌گیری است. یک معیار نوعی برای اندازه‌گیری این‌که یک شکل γ چه میزان برای P خوب است، که با $\mu(P,\ \gamma)$ نمایش داده می‌شود، برابر است با بیشترین فاصله بین یک نقطه از P و یک نقطه همسایه آن روی γ. به عبارت دیگر:

$$\mu(P,\gamma) = max_{p\in P} min_{q\in\gamma}||p-q||.$$

سپس می‌توان یک معیار برای اندازه‌گیری P به صورت $\mu(P) = min_{\gamma}\mu(P,\ \gamma)$ تعریف کرد بطوری‌که "کوچکترین برازش" از میان یک خانواده از اشکال مثل نقاط، خطوط، ابر-صفحه‌ها، کره‌ها و ... گرفته شده است. به طور مثال مساله پیدا کردن کوچکترین کره [سیلندر] دربرگیرنده P، همانند پیدا کردن نقطه‌ای [خط] است که برای P بهترین برازش است و مسائل پیدا کردن کوچکترین عرض جهت‌دار، پوسته استوانه‌ای و پوسته کروی، به‌ترتیب همانند پیدا کردن ابرصفحه، کره و سیلندری است که برای P بهترین برازش است.

الگوریتم‌های دقیق برای محاسبه معیارهای اندازه‌گیری اغلب پرهزینه هستند. بنابراین نظرها به سمت توسعه الگوریتم‌های تقریبی متمرکز شده است [۳۹، ۴۰]. هدف محاسبه یک ϵ-تقریب، برای $۰ < \epsilon < ۱$، از معیارهای اندازه‌گیری با زمان $O(nf(\epsilon))$ یا حتی $O(n + f(\epsilon))$ است که این زمان نزدیک به خطی یا خطی نسبت به n است. به تازگی، چارچوب هسته به عنوان یک روش کلی برای رسیدن به این هدف مطرح شده است. برای هر معیار اندازه‌گیری μ و یک مجموعه ورودی P که می‌خواهیم معیار اندازه‌گیری را برای آن محاسبه کنیم، ایده کلی این است که آیا یک $Q \subseteq P$، به نام هسته، به اندازه $\frac{۱}{\epsilon^{O(۱)}}$ وجود دارد بطوری‌که حل مسئله روی Q یک جواب تقریبی از مسئله اصلی باشد. بطور مثال، اگر $\mu(Q) \geqslant (۱-\epsilon)\mu(P)$ آنگاه این روش یک تقریب از معیار اندازه‌گیری برای P خواهد بود. در زمینه مسائل برازش اشکال، یک ویژگی خوب برای Q این است که برای هر شکل γ از یک خانواده داریم $\mu(Q,\ \gamma) \geqslant (۱-\epsilon)\mu(P,\ \gamma)$. با این ویژگی، این روش یک شکل γ^* را به عنوان جواب برمی‌گرداند که یک تقریب از بهترین برازش برای P است.

قبلا، آگاروال و همکاران [۴۱]، با ایجاد مفهوم ϵ-کرنل، یک چارچوب رسمی ایجاد کردند و نشان دادند که این چارچوب باعث بدست آمدن یک هسته مجموعه برای بسیاری از مسائل بهینه‌سازی است.

۳-۱-۱ کرنل برای مجموعه نقاط

به تابع f یک‌نواخت گفته می‌شود اگر $P_1 \subseteq P_2 \Rightarrow f(P_1) \leqslant f(P_2)$ برقرار باشد (برای مثال، تابع عرض یک مجموعه از نقاط یک‌نواخت است). فرض کنید μ یک تابع اندازه‌گیری یک‌نواخت از یک زیرمجموعه از R^d به یک عدد حقیقی مثبت $R^d \cup \{0\}$ است. برای یک پارامتر داده شده $\epsilon > 0$، می‌گوییم $Q \subseteq P$ نسبت به μ، یک ϵ-هسته مجموعه از P است اگر:

$$(1-\epsilon)\mu(P) \leqslant \mu(Q)$$

آگاروال و همکاران [۴۱]، مفهوم ϵ-کرنل را معرفی کردند و نشان دادند که ϵ-کرنل یک $f(\epsilon)$-هسته مجموعه برای تعداد زیادی مسائل کمینه‌سازی است.

ϵ-کرنل: فرض کنید S^{d-1} نشان‌دهنده یک کره‌ی واحد به مرکز مبدا مختصات در فضای R^d است. برای هر مجموعه P از نقاط در فضای R^d و برای هر جهت $u \in S^{d-1}$، ما عرض جهتدار از P را در جهت u، با $w(u,\ P)$ نمایش داده و به صورت زیر تعریف می‌کنیم:

$$w(u,\ P) = max_{p\in P} < u,\ P > - min_{p\in P} < u,\ P >$$

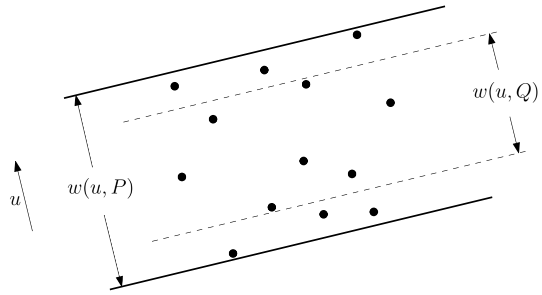
بطوریکه $< .,. >$ همان ضرب داخلی استاندارد است. فرض کنید $0 < \epsilon$ یک پارامتر است. زیرمجموعه $Q \subseteq P$ یک ϵ-کرنل است برای P اگر برای هر $u \in S^{d-1}$ شرط زیر برقرار باشد:

$$(1-\epsilon)w(u,\ P) \leqslant w(u,\ Q)$$

آنگاه به وضوح طبق یک‌نواخت بودن تابع w داریم:

$$(1-\epsilon)w(u,\ P) \leqslant w(u,\ Q) \leqslant w(u,\ P)$$

آگاروال و همکارانش [۴۱]، یک تابع اندازه‌گیری μ را پایدار (faithful) می‌گویند اگر یک ثابت c وجود داشته باشد، وابسته به μ، به طوریکه برای هر $P \subseteq R^d$ و برای هر ϵ، یک ϵ-کرنل از P، یک ϵc-هسته مجموعه برای P با توجه به μ باشد. به عنوان مثال توابع قطر، عرض، شعاع کوچکترین



شکل ۳-۱: عرض جهت‌دار.

دایره پوشاننده و حجم کوچکترین جعبه پوشاننده، توابعی پایدار هستند [۴۱]، یک خصیصه مشترک این اندازه‌گیری‌ها این است که $\mu(P) = \mu(conv(P))$. بنابراین می‌توانیم به سادگی با محاسبه یک $\frac{\epsilon}{c}$-هسته مجموعه برای P، یک ϵ-هسته مجموعه برای P را برای برخی از اندازه‌گیری‌ها محاسبه کنیم.

۳-۲ هسته مجموعه در مدل مجموعه نقاط رنگی گسسته

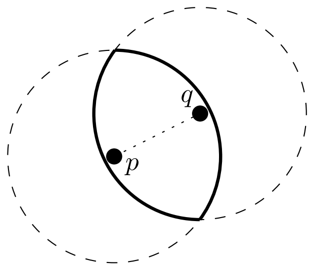
۳-۲-۱ قطر

قطر یک مجموعه P از نقاط، به عنوان بیشترین فاصله بین هر جفت نقطه از P تعریف می‌شود. حال برای یک مجموعه نقطه رنگی P به دنبال محاسبه کوچکترین و بزرگترین قطر در بین تمام **انتخاب** های ممکن از P هستیم.

کوچکترین قطر در مدل مجموعه نقاط رنگی گسسته

فرض کنید $P = \{p_1, p_2, \ldots, p_n\}$ یک مجموعه شامل n نقطه در فضای d-بعدی است که با m رنگ از مجموعه $C = \{c_1, c_2, \ldots, c_m\}$ ، رنگ‌آمیزی شده‌اند (به این معنی که P یک مجموعه از نقاط نادقیق داده شده در مدل مجموعه نقاط رنگی گسسته است). یک **انتخاب** از P، یک زیرمجموعه شامل m نقطه با رنگ‌های متفاوت است و آن را با $Sel(P)$ نمایش می‌دهیم. برای یک مجموعه P، کوچکترین قطر به صورت زیر تعریف می‌شود:

$$D_{min} = \min_{\forall S = Sel(P)} (\max_{\forall p_i, p_j \in S} ||p_i - p_j||)$$



شکل ۳-۲: ناحیه امکان برای دو نقطه p و q نشان داده شده است.

تقریب کوچکترین قطر رنگی

همانطور که در بخش ۳-۱ گفته شد، برای یک مجموعه P از نقاط در R^d، و یک مسئله بهینه‌سازی $\mathcal{X}$، یک زیرمجموعه $Q \subseteq P$، ϵ-مجموعه هسته برای P است اگر حل مسئله بهینه‌سازی X روی Q یک جواب با تقریب ϵ از جواب مسئله روی P باشد. الگوریتم پیشنهادی ما با برگرداندن یک مجموعه از نقاط رنگی $Q \subseteq Sel(P)$ برای تعدادی از انتخاب‌ها از P، یک جواب با ϵ-تقریب برای کوچکترین قطر رنگی روی P پیدا می‌کند. مجموعه Q را می‌توان به عنوان یک ϵ-مجموعه هسته در نظر گرفت. این مفهوم به صورت یک تعریف به عنوان مجموعه هسته رنگی برای قطر در زیر ارائه می‌شود.

تعریف ۱ *یک مجموعه P شامل n نقطه که با m رنگ، رنگ‌آمیزی شده‌اند داده شده است. مجموعه Q برای P یک ϵ-مجموعه هسته برای کوچکترین قطر رنگی است اگر و فقط اگر داشته باشیم:*

$$(1 - \epsilon)D_{min} \leqslant D(Q).$$

دو نقطه p و q را در نظر بگیرید. برای این دو نقطه ناحیه امکان را به این صورت محاسبه می‌کنیم: دو توپ، یکی به مرکز p و دیگری به مرکز q و هر دو به شعاع $|pq| + \epsilon$ رسم می‌کنیم، اشتراک این دو توپ را بدست می‌آوریم و ناحیه بدست آمده را ناحیه امکان می‌نامیم و با C_{pq} نمایش می‌دهیم. فرض می‌کنیم کوچکترین قطر رنگی در C_{pq} باشد، شکل ۳-۲ را نگاه کنید. در واقع ما ناحیه امکان را برای تمام زوج نقاط از P محاسبه می‌کنیم. در این حالت بدیهی است یکی از این زوج نقاط همان کوچکترین قطر رنگی است که می‌دانیم انتخاب رنگی مربوط به این نقاط بیرون از ناحیه امکان نخواهد بود (زیرا pq قطر این مجموعه است).

تمام زیرمجموعه‌های دو عضوی از اعضای P را انتخاب می‌کنیم، برای مثال شکل ۳-۳آ را به عنوان مجموعه نقاط ورودی در نظر بگیرید. برای هر زیرمجموعه دو عضوی $S_i = \{p, q\}$، ابتدا چک می‌کنیم

دو نقطه p و q از یک رنگ نباشند، شکل ۳-۳ب. C_{pq} را در نظر می‌گیریم و چک می‌کنیم از هر رنگ حداقل یک نقطه در C_{pq} وجود داشته باشد و در غیر این‌صورت زیرمجموعه S_i را دور می‌ریزیم، شکل ۳-۳ج. در واقع مجموعه S_i که فاقد این دو ویژگی باشد را غیرمجاز می‌نامیم. در ادامه (یعنی اگر دو نقطه انتخابی از یک رنگ نبودند و در ناحیه امکان آنها از هر رنگ حداقل یک نقطه وجود داشت، شکل ۳-۳د) با توجه به $|pq|$ یک تقریب از کوچکترین قطر تمام نقاط داخل C_{pq} بدست می‌آوریم و در بین تمام این محاسبات برای S_iها، کوچکترین قطر تقریبی بدست آمده را به عنوان خروجی برمی‌گردانیم و با D_{min} نشان می‌دهیم، این رویه در الگوریتم ۱ آمده است. در نهایت نشان می‌دهیم از آنجایی‌که برای بعضی از S_iها $D_{opt} = pq$ است، D_{min} یک ϵ-تقریب از D_{opt} خواهد بود.

الگوریتم ۱ الگوریتم محاسبه یک ϵ-تقریب برای کوچکترین قطر رنگی

---

۱ $D_{min} = null$

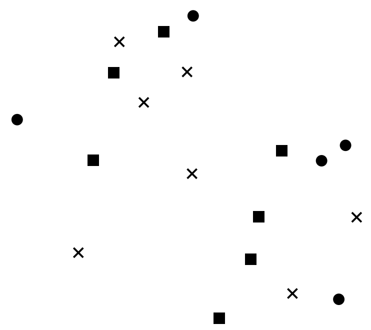
۲ فرض کنید S مجموعه تمام انتخاب‌های دو عضوی از P است

۳ برای هر $S_i \in S$ موارد زیر را انجام بده:

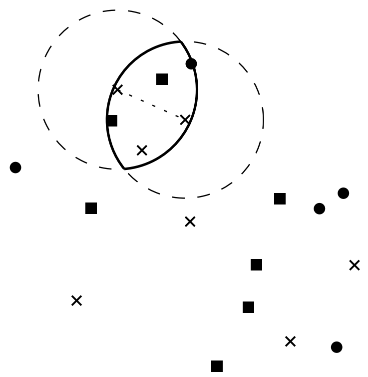
- اگر اعضای S_i یک مجموعه مجاز است آنگاه:
  - فرض کنید C_{pq} ناحیه ممکن برای S_i باشد و B_i تمام نقاط رنگی در ناحیه C_{pq} باشد
  - فرض کنید $D(S_i)$ با توجه به $|pq|$ یک تقریب از کوچکترین قطر رنگی در S_i باشد که توسط الگوریتم ۲ محاسبه شده است
  - $D_{min} = \min\{D_{min}, D(S_i)\}$

۴ D_{min} را برگردان

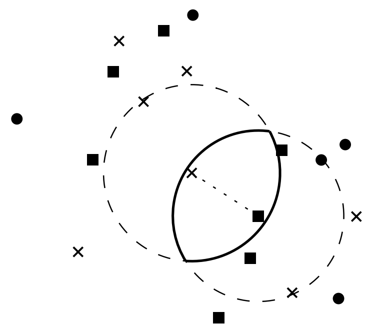
برای تقریب قطر، برای هر یک از S_iها به این‌صورت عمل می‌کنیم: $|pq|$ را با Δ نمایش می‌دهیم. کوچکترین باکس پوشا برای S_i موازی با محورهای مختصات را محاسبه و با $B(S_i)$ نشان می‌دهیم. $B(S_i)$ را به سلول‌هایی با اندازه $\epsilon\Delta$ تقسیم می‌کنیم و توری یک‌نواخت بدست آمده را $\mathbb{G}$ می‌نامیم. حال تمام تخصیص‌های دودویی (۰/۱) را برای سلول‌های $\mathbb{G}$ در نظر بگیرید، یک نمونه از تخصیص در شکل ۳-۴آ آمده است. برای iامین تخصیص فرض کنید Q_i مجموعه تمام سلول‌هایی باشد که با ۱ مقداردهی شده‌اند. مجموعه Q_i را مجاز می‌دانیم اگر برای هر رنگ از مجموعه C حداقل یک نقطه در حداقل یکی از سلول‌های Q_i وجود داشته باشد و بعلاوه بتوان یک نماینده از هر سلول انتخاب کرد



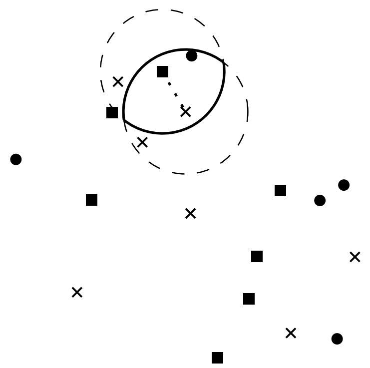
(آ) مجموعه نقاط ورودی شامل سه رنگ (نوع)



(ب) انتخاب دو نقطه هم رنگ (غیرمجاز)



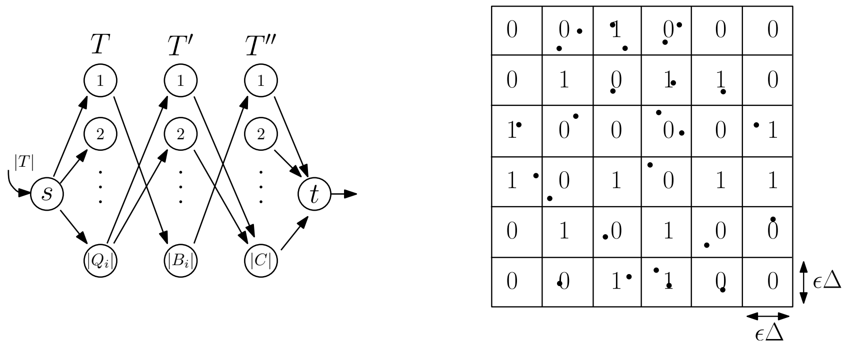
(ج) ناحیه امکان فاقد انتخاب رنگی است (غیرمجاز)



(د) انتخاب دو نقطه که در اطراف آن انتخاب رنگی وجود ندارد

شکل ۳-۳: مجموعه نقاط و وضعیت‌های متفاوت در انتخاب‌ها.

بطوری‌که هیچ دو سلولی دارای نماینده‌های با رنگ یکسان نباشند. از آنجایی‌که تعداد سلول‌ها از مرتبه $O(\frac{1}{\epsilon^d})$ است، حداکثر تعداد مجموعه‌های مجاز از مرتبه $O(2^{\frac{1}{\epsilon^d}})$ خواهد بود، که چون ϵ یک ثابت کوچک است این مقدار نیز یک ثابت خواهد بود. برای هر یک از Q_iها در صورت مجاز بودن، با استفاده از نماینده‌های Q_i یک تقریب از $D_{opt}(S_i)$ (کوچکترین قطر رنگی در B_i) بدست می‌آوریم. برای هریک از تخصیص‌های مجاز در زمان $O((\frac{1}{\epsilon^d})^2)$ قطر مربوط به نماینده‌ها را محاسبه می‌کنیم. بین تقریب‌های بدست آمده برای تمام تخصیص‌های مجاز، کوچکترین قطر تقریبی بدست آمده را به‌عنوان یک تقریب از $D_{opt}(S_i)$ برمی‌گردانیم (توجه کنید که، به وضوح یکی از تخصیص‌ها دقیقا دارای مجموعه سلول‌هایی است که شامل مجموعه نقاط $D_{opt}(S_i)$ هستند)، این رویه در الگوریتم ۲ آمده است.



(ب) تشکیل شبکه جریان (آ) ϵ-توری $\mathbb{G}$

شکل ۳-۴: (آ) ϵ-توری $\mathbb{G}$ و یک نمونه از تخصیص برای آن. (ب) تشکیل شبکه جریان با شاری به اندازه $|T|$ به عنوان ورودی برای بدست آوردن نمایندگان مربوط به سلول‌هایی که با ۱ مقداردهی شده‌اند.

در نهایت برای تمام S_iهایی که مجاز بودند و برایشان یک تقریب از کوچکترین قطر آنها محاسبه کردیم، کوچکترین قطر تقریبی را انتخاب و به عنوان تقریبی از کوچکترین قطر رنگی مجموعه P معرفی می‌کنیم.

از الگوریتم جریان شبکه برای تشخیص اینکه مجموعه سلول‌های Q_i مجاز هستند یا نه، استفاده می‌کنیم. به این صورت که، یک مجموعه از رئوس T ایجاد می‌کنیم بطوریکه هر راس در T نشان‌دهنده یک سلول متفاوت از Q_i است. یک راس مبداء s با یال‌های جهت‌دار به رئوس T ایجاد می‌کنیم. یک مجموعه از رئوس T' ایجاد می‌کنیم بطوری‌که هر راس در T' نشان‌دهنده یک نقطه مشخص است که در سلولی از Q_i قرار دارد. یک یال بین $u \in T$ و $u' \in T'$ قرار می‌دهیم اگر سلول متناظر در Q_i شامل نقطه متناظر باشد. یک مجموعه دیگر از نقاط T'' ایجاد می‌کنیم که هر راس آن نشان‌دهنده یک رنگ از مجموعه C است. یک یال بین $u' \in T'$ و $u'' \in T''$ قرار می‌دهیم اگر نقطه مورد نظر با رنگ متناظر رنگ شده باشد. در نهایت یک راس t ایجاد می‌کنیم و از تمام رئوس T'' یک یال به t رسم می‌کنیم. وزن تمام یال‌ها را با ۱ معین می‌کنیم. یک شار بیشینه با اندازه $|T|$ از s به t، نماینده برای هر سلول را مشخص می‌کند بطوریکه هیچ دو نماینده‌ای دارای رنگ یکسان نیستند، شکل ۳-۴ب را نگاه کنید. به سادگی می‌توان مشاهده کرد یک چنین شاری موجود است اگر و فقط اگر مجموعه سلول‌های متناظر مجاز باشد. از آنجایی‌که ظرفیت تمام یال‌ها ۱ (یک عدد صحیح) است با استفاده از الگوریتم فورد-فالکرسون[۱]، زمان اجرای الگوریتم برابر است با اندازه جریان بیشینه در تعداد یال‌های گراف. اندازه جریان بیشینه

[۱] Ford-Fulkerson

برابر است با تعداد سلول‌هایی که با ۱ مقداردهی شده‌اند که از مرتبه $O(\frac{1}{\epsilon^d})$ است و تعداد یال‌ها از مرتبه $O(n)$ است. بنابراین زمان اجرای این بخش از الگوریتم از مرتبه $O(\frac{n}{\epsilon^d})$ خواهد بود.

الگوریتم ۲ الگوریتم محاسبه یک تقریب از کوچکترین قطر رنگی با توجه به فاصله یک زوج از نقاط

---

۱ فرض کنید B_i مجموعه ورودی باشد و $p, q \in B_i$ دو نقطه‌ای هستند که C_{pq} شامل B_iاست.

۲ $\Delta = |pq|$، $D_{pq} = null$

۳ فرض کنید $\mathbb{G}$ یک توری یک‌نواخت با سلول‌های به اندازه $\epsilon\Delta$ در d-بعد باشد

۴ برای هر تخصیص دودویی (۰/۱) به سلول‌ها در توری $\mathbb{G}$ موارد زیر را انجام بده:

- فرض کنید Q_i مجموعه سلول‌هایی باشد که با ۱ مقداردهی شده‌اند
- اگر Q_i مجاز است آنگاه:
  - فرض کنید Q'_i مجموعه نقاط نماینده سلول‌های Q_i باشد
  - $D_{pq} = \min\{D_{pq}, D(Q'_i)\}$

۵ D_{pq} را برگردان

قضیه‌ی ۳-۱ *الگوریتمی وجود دارد که با استفاده از پیدا کردن یک مجموعه هسته رنگی Q از P یک ϵ- تقریب از کوچکترین قطر رنگی را در زمان $O(n^۲(n + ۲^{\frac{1}{\epsilon^d}}(\frac{n}{\epsilon^d} + (\frac{1}{\epsilon^d})^۲)))$ محاسبه می‌کند.*

اثبات. برای اثبات موارد زیر را در نظر بگیرید:

- کوچکترین قطر یکی از pqهای مجاز خواهد بود.
- زمانی‌که pq انتخاب شده همان جواب بهینه باشد، یکی از تخصیص‌های ϵ-توری شامل تمام نقاط انتخاب رنگی مربوط به D_{min} خواهد بود.

با توجه به موارد بالا و این‌که از عمل مینیمم‌گیری بین تقریب‌های به‌دست آمده استفاده می‌کنیم، در نهایت یک ϵ-تقریب از کوچکترین قطر داریم. □

فصل ۴

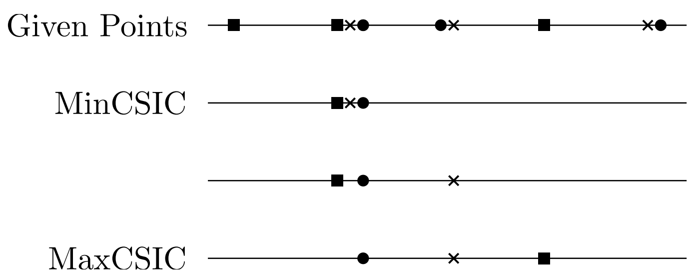
پوشش واحد برای نقاط رنگی

یک مجموعه شامل n نقطه داده شده، پوشاندن آنها با کمترین تعداد توپ‌های واحد را پوشش واحد[۱] می‌گوییم. این مسئله در فضای اقلیدسی ان‌پی-سخت است [۴۲]. در ابعاد ثابت الگوریتم‌های چندجمله‌ای با ضریب تقریب $(1+\epsilon)$ برای مسئله پوشش واحد وجود دارد [۴۳]. این مسئله با توجه به کاربردهای فراوان در زمینه‌های متفاوت، به طور گسترده‌ای مورد مطالعه قرار گرفته است. به عنوان مثال از کاربردهای این مسئله می‌توان به مدیریت داده‌ها در شبکه‌های بیسیم اشاره کرد [۴۴، ۴۵، ۴۶، ۴۷].

در این بخش از این رساله، مسئله پوشش واحد را زمانی که نقاط ورودی نادقیق هستند در نظر می‌گیریم. فرض کنید مجموعه «واقعی» نقاط ورودی $\mathcal{P} = \{p_1, p_2, ..., p_l\}; p_i \in \mathbb{R}^d$ ناشناخته است و در عوض یک مجموعه $P = \{p_1, p_2, ..., p_n\}$ شامل n نقطه که با m رنگ از مجموعه $C = \{c_1, c_2, ..., c_m\}$ رنگ‌آمیزی شده‌اند، در فضای d-بعدی داده شده است. یک انتخاب رنگی از P، یک زیرمجموعه با m نقطه، از هر رنگ یک نقطه است. نقاط با رنگ یکسان نشان‌دهنده مکان‌های ممکن برای یک نقطه نادقیق است. این نادقیقی در ورودی منجر به نادقیقی در خروجی می‌شود. بنابراین به‌جای محاسبه خروجی دقیق، به محاسبه حد پایین و بالا برای مقادیر ممکن خروجی در مسئله پوشش واحد می‌پردازیم. برای مثال در شکل ۴-۱ سه انتخاب رنگی متفاوت به نمایش در آمده است. با توجه به موارد بالا در این بخش به دو مسئله زیر می‌پردازیم:

- **مسئله ۱ (پوشش رنگی کمینه)** یک انتخاب رنگی S از P پیدا کنید بطوریکه تعداد توپ‌هایی که

[۱] Unit covering



شکل ۴-۱: سه انتخاب رنگی متفاوت از یک نمونه ورودی داده شده.

در پوشش واحد روی مجموعه S بکار می‌رود کمینه شود.

- **مسئله ۲ (پوشش رنگی بیشینه)** یک انتخاب رنگی S از P پیدا کنید بطوریکه تعداد توپ‌هایی که در پوشش واحد روی مجموعه S بکار می‌رود بیشینه شود.

۴-۱ علائم و پیشنیازها

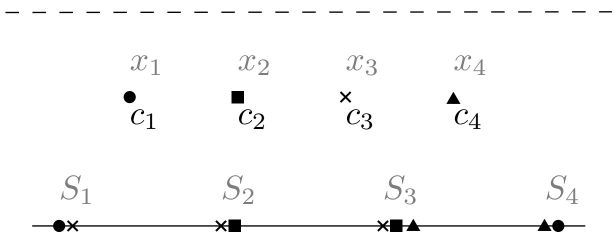
فرض کنید P مجموعه نقاط ورودی مسئله و C مجموعه رنگ‌هایی که در P وجود دارد. برای همه $c_i \in C$، فرکانس را برابر با تعداد نقاط با رنگ c_i در P تعریف می‌کنیم. بیشترین فرکانس در بین رنگ‌ها را با f نشان می‌دهیم. در این صورت تعداد نقاط از یک رنگ بیش از f نخواهد بود و بعلاوه برای رنگ‌هایی که فرکانس آنها ۱ است، یک انتخاب بیشتر نداریم پس فرض می‌کنیم $f \geqslant ۲$ است.

بجز در مواردی که به صراحت بیان می‌شود، تمرکز روی مسئله در حالت یک بعدی است. در این حالت یک توپ واحد، به صورت یک بازه واحد ظاهر می‌شود.

برای یک انتخاب رنگی S از P، فرض می‌کنیم $U(S)$ مجموعه بازه‌هایی هستند که در پوشش واحد برای S بکار رفته‌اند (توجه کنید که پوشش واحد با کمترین تعداد بازه‌ها مورد نظر است). همچنین می‌توان به سادگی این فرض را در نظر داشت که در پوشش واحد بهینه $U(S)$ انتهای سمت چپ تمام بازه‌ها دقیقا روی یکی از نقاط S است و تمام بازه‌ها در $U(S)$ مجزا هستند (این فرض به راحتی با انتقال به راست بازه‌های $U(S)$ امکان‌پذیر است). در ادامه همیشه فرض می‌کنیم $U(S)$ پوشش واحد بهینه به همراه خاصیت گفته شده است.

OPT_{min} و OPT_{max} را انتخاب‌های رنگی در نظر می‌گیریم که منجر به $U(S)$ با کمترین و بیشترین تعداد بازه می‌شود.

$$G = \{x_1, x_2, x_3, x_4\} \quad F = \{S_1, S_2, S_3, S_4\}$$
$$S_1 = \{x_1, x_3\} \qquad S_2 = \{x_2, x_3\}$$
$$S_3 = \{x_2, x_3, x_4\} \quad S_4 = \{x_1, x_4\}$$



شکل ۴-۲: یک مثال از کاهش مسئله پوشش مجموعه‌ای به مسئله پوشش رنگی کمینه.

۴-۲ پوشش رنگی کمینه

۴-۲-۱ سختی مسئله پوشش رنگی کمینه

قضیه‌ی ۴-۱ *پوشش رنگی کمینه ان‌پی- سخت است.*

اثبات. برای اثبات با کاهش از پوشش مجموعه‌ای [۲] نشان می‌دهیم مسئله حتی در یک بعد هم ان‌پی- سخت است. یک نمونه از مسئله پوشش مجموعه‌ای شامل مجموعه پایه [۳] $G = \{x_۱, x_۲, \ldots, x_m\}$ و مجموعه پوششی [۴] $F = \{S_۱, S_۲, \ldots, S_k\}$ را در نظر بگیرید، که در آن $OPT_{sc} \subseteq F$ پوشش بهینه برای این نمونه است. متناظر با هر $x_j \in G$، رنگ c_j را در نمونه برای مسئله پوشش رنگی کمینه در نظر بگیرید. برای هر زیرمجموعه‌ی $S_i \in F$، یک قطعه‌ی واحد روی محور xها تعیین می‌کنیم و با $Cell_i$ نشان می‌دهیم با این ویژگی که نقطه انتهایی بازه‌های متفاوت بیش از یک واحد از هم فاصله داشته باشند. در ادامه برای هر $x_j \in S_i$، یک نقطه با رنگ c_j در $Cell_i$ قرار می‌دهیم. برای مثال در شکل ۴-۲ یک نمونه از این کاهش آمده است.

فرض کنید P مجموعه نقاط ایجاد شده در نمونه مقصد باشد. به دلیل اینکه فاصله بین هر دو سلول بیش از واحد است، هر بازه در $U(OPT_{min})$ تنها نقاط داخل یک سلول را می‌پوشانند. در واقع اگر دو بازه مجزا با یک سلول اشتراک داشته باشند، می‌توان آنها را با یک بازه جایگزین کرد که خلاف فرض

[۲] Set Cover

[۳] Ground set

[۴] Covering set

کمینه بودن است. مجموعه‌هایی را به عنوان نتیجه برمی‌گردانیم که سلول‌های متناظر با آنها در نمونه مقصد با بازه‌های $U(OPT_{min})$ اشتراک دارند و این مجموعه را با R نشان می‌دهیم. از آنجایی‌که حداقل یک نقطه از هر رنگ توسط بازه‌های $U(OPT_{min})$ پوشانده شده‌اند، R یک مجموعه جواب ممکن برای نمونه مبداء است. به عنوان نتیجه داریم:

$$|U(OPT_{min})| \geqslant |OPT_{sc}|$$

از طرف دیگر سلول‌هایی را که متناظر با زیرمجموعه‌های OPT_{sc} هستند را در نظر بگیرید و یک انتخاب رنگی دلخواه توسط نقاط درون این سلول‌ها بردارید. چون OPT_{sc} تمام عناصر G را می‌پوشاند این انتخاب رنگی امکان‌پذیر است. به وضوح این انتخاب رنگی با بازه‌های واحدی به تعداد $|OPT_{sc}|$ قابل پوشش است، بنابراین:

$$|U(OPT_{min})| \leqslant |OPT_{sc}|$$

و به عنوان نتیجه $|U(OPT_{min})| = |OPT_{sc}|$. با توجه به ان‌پی-سخت بودن مسئله پوشش مجموعه‌ای، مسئله پوشش رنگی کمینه حتی در یک بعد ان‌پی-سخت است .

□

با توجه به اینکه مسئله پوشش مجموعه‌ای حتی زمانی که فرکانس هر $x_j \in G$ حداکثر ۲ است (به این معنی که x_j حداکثر در ۲ عنصر از F ظاهر شده است) ان‌پی-سخت است، با یک کاهش مشابه از نسخه محدود شده پوشش مجموعه‌ای، می‌توانیم ادعا کنیم مسئله پوشش رنگی کمینه در یک بعد حتی با $f = ۲$ ان‌پی-سخت است.

بعلاوه، از کاهش بالا می‌توان نتیجه گرفت هر الگوریتم تقریبی با ضریب تقریب ثابت برای مسئله پوشش رنگی کمینه، یک الگوریتم تقریبی با ضریب تقریب مشابه برای مسئله پوشش مجموعه‌ای خواهد بود در حالیکه هیچ الگوریتم تقریبی با ضریب تقریب ثابت برای مسله پوشش مجموعه‌ای وجود ندارد مگر اینکه $P = NP$.

نتیجه: هیچ الگوریتم تقریبی با ضریب تقریب ثابت در زمان چند جمله‌ای برای مسئله پوشش رنگی کمینه وجود ندارد مگر اینکه $P = NP$.

۴-۲-۲ الگوریتم‌های تقریبی برای مسئله پوشش رنگی کمینه

قضیه‌ی ۴-۲ *یک الگوریتم تقریبی با ضریب تقریب $\ln(m)$ برای مسئله پوشش رنگی کمینه در یک بعد وجود دارد.*

اثبات. فرض کنید $I = \{I_۱, I_۲, \ldots, I_n\}$ یک مجموعه از بازه‌ها باشد بطوری‌که I_i یک بازه واحد با انتهای سمت چپ روی iامین نقطه از ورودی باشد (نقاط مجموعه P را از چپ به راست روی محور xها فرض کنید). با توجه به I و طبق فرضی که در مورد $U(OPT_{min})$ داشتیم، می‌توانیم بگوییم:

$$U(OPT_{min}) \subseteq I$$

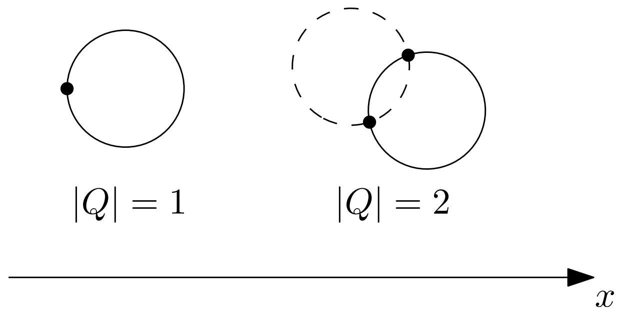
بنابراین، مسئله پوشش رنگی کمینه پیدا کردن $\mathcal{I} \subseteq I$ با کمترین اندازه است بطوری‌که از هر رنگ حداقل یک نقطه پوشانده شود و سپس یک انتخاب رنگی دلخواه از نقاط پوشانده شده داشته باشیم. به منظور نشان دادن این مسئله از طریق مسئله پوشش مجموعه‌ای:

- فرض کنید G مجموعه تمام رنگ‌ها است.
- برای هر $I_i \in I$ یک زیرمجموعه از G شامل رنگ‌های پوشیده شده با I_i تعریف می‌کنیم.

یک الگوریتم حریصانه شناخته شده برای مسئله پوشش مجموعه‌ای با ضریب تقریب $\ln(\Delta)$ وجود دارد بطوریکه Δ اندازه بزرگترین زیرمجموعه در مجموعه پوشش است [۴۸]. از آنجایی‌که در کاهش بالا، اندازه هر زیرمجموعه پوششی حداکثر برابر m (تعداد رنگ‌ها) است، اعمال الگوریتم تقریبی پوشش مجموعه‌ای با ضریب تقریب $\ln(\Delta)$ یک الگوریتم با ضریب تقریب $\ln(m)$ برای پوشش رنگی کمینه خواهد بود. $\square$

این الگوریتم تقریبی به سادگی برای ابعاد بالاتر تعمیم داده می‌شود. در $d = ۲$، برای هر $Q \subseteq P$ با $|Q| \leqslant ۲$، یک دایره واحد پیدا کنید (در صورت وجود) بطوریکه نقاط Q روی مرز دایره قرار بگیرند و مشخصه طول مرکز آن بیشینه باشد. برای مثال شکل ۴-۳ را نگاه کنید. فرض کنید I مجموعه تمام این دایره‌ها باشد.

طبق تعریف $I \in O(n^۲)$ است. علاوه بر این، می‌توان مشاهده کرد برای هر انتخاب رنگی، پوشش واحد بهینه با استفاده از اعضاء I وجود دارد (مشابه آنچه در مورد $U(.)$ گفته شد). بنابراین کاربرد



شکل ۴-۳: مثالی از دایره‌های موجود در I برای $d = ۲$.

الگوریتم با ضریب تقریب $\ln(m)$ برای پوشش مجموعه‌ای در این مورد هم صادق است. با همین روش می‌توان نتیجه را برای ابعاد $d \geqslant ۳$ تعمیم داد.

قضیه‌ی ۴-۳ *یک الگوریتم تقریبی با ضریب تقریب $۲f$ برای مسئله پوشش رنگی کمینه در یک بعد وجود دارد.*

اثبات. در ابتدا یک مجموعه از بازه‌های واحد I را پیدا کنید بطوری‌که $|I| \leqslant n$ و دو شرط زیر برقرار باشد:

- تمام nتا نقطه باید پوشیده شوند.
- هیچ دو بازه‌ای در I باهم اشتراک نداشته باشند (به این معنی که بازه‌ها مجزا از هم باشند).

سپس مسئله زیر را در نظر بگیرید که شبیه به پوشش رنگی کمینه است اما به صورت محدود شده:

مسئله ۳ (پوشش رنگی کمینه با محدودیت) یک زیرمجموعه با کوچکترین اندازه از I پیدا کنید بطوری‌که از هر رنگ حداقل یک نقطه را بپوشاند.

لم ۴-۴ *$|S_I| \leqslant ۲|U(OPT_{min})|$ بطوری‌که S_I با توجه به I، جواب بهینه برای مسئله پوشش رنگی کمینه با محدودیت است.*

اثبات. چون بازه‌ها روی محور xها قرار دارند، برای هر $u \in U(OPT_{min})$، یک $I' \subseteq I$ با $|I'| \leqslant ۲$ وجود دارد بطوری‌که I' تمام نقاط پوشانده شده توسط u را می‌پوشاند. در نتیجه، با جایگزینی هر بازه در $U(OPT_{min})$ با حداکثر دو بازه از I می‌توان $S' \subseteq I$ را ساخت بطوری‌که هر نقطه پوشانده شده

توسط $U(OPT_{min})$ توسط S' پوشانده شود. نظر به اینکه $|S_I| \leqslant |S'|$ و $|S'| \leqslant ۲|U(OPT_{min})|$، داریم $|S_I| \leqslant ۲|U(OPT_{min})|$.

□

با در نظر گرفتن لم ۴-۴، هر الگوریتم با ضریب تقریب f برای پوشش رنگی کمینه با محدودیت یک الگوریتم تقریبی با ضریب تقریب $۲f$ برای پوشش رنگی کمینه خواهد بود.

حال با استفاده از برنامه‌ریزی خطی گرد شده یک الگوریتم با ضریب تقریب f برای مسئله پوشش رنگی کمینه با محدودیت ارائه می‌کنیم. برای هر بازه $i \in I$، یک متغیر x_i در نظر می‌گیریم بطوری‌که $x_i = ۱$ است اگر و فقط اگر i انتخاب شده باشد. بعلاوه، فرض کنید I_c یک زیرمجموعه از I، شامل بازه‌هایی است که نقاط با رنگ c را می‌پوشانند. برنامه‌ریزی صحیح [۵] زیر را در نظر بگیرید:

$$Min\ \Sigma_{i \in I}\ x_i$$

$$s.t.\ \ \forall c \in C\ \ \Sigma_{i \in I_c}\ x_i\ \geqslant\ ۱$$

$$\forall i \in I\ \ x_i \in \{۰, ۱\}$$

برنامه‌ریزی خطی سست شده مربوط به این برنامه‌ریزی صحیح با تغییر دامنه x_i از $\{۰, ۱\}$ به $[۰, ۱]$ بدست می‌آید. OPT_{LP} را به عنوان جواب بهینه برنامه‌ریزی خطی در نظر بگیرید. از آنجایی‌که جواب بهینه برنامه‌ریزی صحیح دقیقا برابر با $|S_I|$ است، پس $OPT_{LP} \leqslant |S_I|$. برای هر $i \in I$، x'_i را به صورت زیر تعریف می‌کنیم:

$$x'_i = \begin{cases} ۱ & x_i \geqslant \frac{۱}{f} \\ ۰ & otherwise \end{cases}$$

طبق تعریف $x'_i \leqslant f.x_i$ است بنابراین:

$$\Sigma_{i \in I} x'_i\ \leqslant\ f \times \Sigma_{i \in I} x_i\ =\ f \times OPT_{LP}\ \leqslant\ f|S_I|. \qquad (۱-۴)$$

[۵] Integer programming

بدلیل این‌که ۱ $\geqslant \Sigma_{i\in I_c} x_i$,$\forall c \in C$ و $|I_c| \leqslant f$، یک $i \in I_c$ وجود دارد که $x_i \geqslant \frac{1}{f}$ است به این معنی که ۱ = x'_i. به عنوان نتیجه، با انتخاب بازه‌های i بطوریکه ۱ = x'_i، تمام رنگ‌ها پوشش داده می‌شوند. پس این یک جواب ممکن برای پوشش رنگی کمینه خواهد بود. با توجه به نامساوی ۴-۱، این الگوریتم یک تقریب f برای پوشش رنگی کمینه با محدودیت است.

علاوه بر این، با توجه به لم ۴-۴ هر الگوریتم با ضریب تقریب f برای پوشش رنگی کمینه با محدودیت یک الگوریتم تقریبی با ضریب تقریب ۲f برای پوشش رنگی کمینه خواهد بود.

□

۴-۳ پوشش رنگی بیشینه

۴-۳-۱ سختی مسئله پوشش رنگی بیشینه

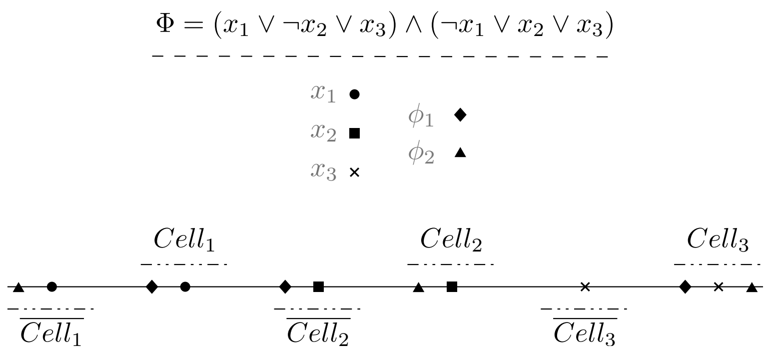
قضیه‌ی ۴-۵ *پوشش رنگی بیشینه ان‌پی- سخت است.*

اثبات. نشان می‌دهیم پوشش رنگی بیشینه برای ۲ < f حتی در یک بعد ان‌پی-سخت است. برای اثبات از کاهش نسخه محدود شده ۳-صدق‌پذیری، که در آن هر متغیر حداکثر ۲ بار به شکل مثبت و یک بار به شکل منفی ظاهر شده است، به مسئله پوشش رنگی بیشینه استفاده می‌کنیم. براحتی می‌توان هر ۳-صدق‌پذیری را به یک نسخه محدود شده تبدیل کرد که در آن شرایط گفته شده برای متغیرها برقرار باشد. بنابراین در ادامه منظور از ۳-صدق‌پذیری همان نسخه محدود شده است. از آنجایی‌که هر مسئله ۳-صدق‌پذیری را می‌توان به یک مسئله ۳-صدق‌پذیری محدود شده تبدیل کرد، بدیهی است که مسئله ۳-صدق‌پذیری محدود شده ان‌پی-سخت است.

یک نمونه از ۳-صدق‌پذیری داده شده است، برای هر متغیر x_i، دو قطعه $Cell_i$ و $\overline{Cell_i}$ به طول ۳ که به ترتیب متناظر با x_i و $\overline{x_i}$ هستند را تعیین کنید. سلول‌ها باید طوری قرار بگیرند که هیچ دو سلولی باهم اشتراک نداشته باشند. بعلاوه، یک رنگ مجزا برای هر متغیر و هر عبارت [۶] در نظر می‌گیریم. رنگ تعیین شده برای متغیر x_i را با c_i و رنگ تعیین شده برای عبارت Φ_j را با c'_j نشان می‌دهیم.

برای هر متغیر x_i، دو نقطه با رنگ c_i در میانه سلول‌های $Cell_i$ و $\overline{Cell_i}$ قرار می‌دهیم. از آنجایی‌که

[۶] Clause



شکل ۴-۴: مثالی از ایجاد یک نمونه پوشش رنگی بیشینه از روی نمونه ۳-صدق‌پذیری.

تنها این نقاط با رنگ c_i رنگ‌آمیزی شده‌اند، هر انتخاب رنگی باید شامل حداقل یکی از آنها باشد. انتخاب نقطه میانی یک سلول به معنی آن است که آن لفظ [۷] ۰ است. به بیان دیگر، اگر نقطه میانی $Cell_i$ انتخاب شده، $x_i = ۰$ و از طرف دیگر، اگر نقطه میانی سلول $\overline{Cell_i}$ انتخاب شود، $\overline{x_i} = ۰$ که معادل آن $x_i = ۱$ است.

در ادامه برای هر عبارت Φ_j، سه نقطه با رنگ c'_j در سلول‌های متناظر با لفظ‌های آن به فاصله $\frac{۳}{۴}$ از نقطه میانی قرار می‌دهیم. توجه کنید که حداکثر دو نقطه از عبارت‌ها در یک سلول یکسان قرار می‌گیرند. اگر دو نقطه که مربوط به دو عبارت متفاوت هستند، در یک سلول یکسان قرار گرفتند، آنها باید در طرفین نقطه میانی قرار بگیرند. برای مثال شکل ۴-۴ را ببینید.

لم ۴-۶ *نمونه ۳-صدق‌پذیری ارضاء شدنی است اگر و فقط اگر یک انتخاب رنگی برای پوشش رنگی بیشینه ساخته شده از این نمونه وجود داشته باشد بطوریکه فاصله بین هر دو نقطه بیش از ۱ باشد.*

اثبات. فرض کنید انتخاب رنگی S وجود دارد بطوری‌که فاصله بین هر دو نقطه بیش از ۱ است. برای هر رنگ c_i، اگر نقطه میانی $Cell_i$ در S باشد، $x_i = ۰$، در غیراینصورت (به این معنی که نقطه میانی $\overline{Cell_i}$ در S قرار دارد) $\overline{x_i} = ۰$. توجه کنید که برای هر عبارت Φ_j، یک نقطه P_j با رنگ c'_j در S وجود دارد. از آنجایی‌که فاصله بین P_j و نقطه میانی سلول که P_j در آن قرار دارد کمتر از ۱ است، این نقطه میانی نمی‌تواند در S باشد، بنابراین یک لفظ در Φ_j وجود دارد بطوریکه مقدارش ۱ است.

از طرف دیگر، اثبات می‌کنیم مقداردهی که نمونه ۳-صدق‌پذیری را ارضاء می‌کند می‌تواند یک انتخاب رنگی را نتیجه دهد که در آن فاصله بین هر دو نقطه بیشتر از ۱ است. متناظر با هر متغیر x_i،

[۷] Literal

برای $x_i = 0$ یا $\overline{x_i} = 0$ به ترتیب نقطه میانی $Cell_i$ یا $\overline{Cell_i}$ را انتخاب کنید. از آنجایی‌که برای هر عبارت Φ_j، حداقل یک لفظ در آن وجود دارد که آنرا ارضاء می‌کند، یک سلول شامل یک نقطه با رنگ c'_j وجود دارد که نقطه میانی آن انتخاب نشده است، بنابراین انتخاب یک نقطه با رنگ c'_j امکان‌پذیر است.

□

لم ۴-۷ *یک نمونه از پوشش رنگی بیشینه داده شده است، $|U(OPT_{max})| = m$ اگر و فقط اگر یک انتخاب رنگی وجود داشته باشد بطوریکه فاصله بین هر دو نقطه بیش از ۱ باشد.*

اثبات. فرض کنید یک انتخاب رنگی S وجود دارد بطوریکه فاصله بین هر دو نقطه در S بیش از ۱ باشد. از آنجایی‌که هیچ بازه واحدی نمی‌تواند دو نقطه از S را بپوشاند، برای پوشاندن S حداقل m بازه واحد نیاز است. از طرف دیگر، اگر دو نقطه در S باشند که حداکثر فاصله‌شان ۱ باشد، یک بازه واحد می‌تواند هردوی آنها را بپوشاند، بنابراین S می‌تواند با کمتر از m بازه واحد پوشیده شود. □

با توجه به لم ۴-۷، می‌توانیم ادعا کنیم نمونه ۳-صدق‌پذیری ارضاء‌پذیر است اگر و فقط اگر $|U(OPT_{max}| = m$.

□

۴-۳-۲ الگوریتم تقریبی برای پوشش رنگی بیشینه

حالا یک الگوریتم تقریبی با ضریب تقریب $\frac{1}{2}$ در زمان $O(n \log n)$ برای پوشش رنگی بیشینه در یک بعد ارائه می‌کنیم.

الگوریتم ۳ الگوریتم تقریبی برای مسئله پوشش رنگی بیشینه

---

۱ مقداردهی‌های اولیه زیر را انجام بده:

- $mark = \emptyset$
- $T = \emptyset$
- $T' = \emptyset$

۲ تا زمانیکه $|mark| < n$ مراحل زیر را انجام بده:

- p را سمت چپ‌ترین نقطه‌ای در نظر بگیرید که در $mark$ نیست
- p را به T اضافه کن ($T = T \cup \{p\}$)
- برای هر نقطه q که همرنگ با p است عمل زیر را انجام بده:
  - q را به مجموعه $mark$ اضافه کن ($mark = mark \cup \{q\}$)
- برای هر نقطه q که فاصله‌اش با p کمتر از ۱ است ($dist(p,q) \leqslant 1$) عمل زیر را انجام بده:
  - q را به مجموعه $mark$ اضافه کن ($mark = mark \cup \{q\}$)

۳ برای هر رنگ c که هیچ کاندیدی در T ندارد عمل زیر را انجام بده:

- یک نقطه دلخواه با رنگ c به مجموعه T' اضافه کن

۴ $T \cup T'$ را به عنوان جواب برگردان

قضیه‌ی ۴-۸ *الگوریتم ۳ یک الگوریتم با ضریب تقریب* $\frac{1}{2}$ *است.*

اثبات. الگوریتم ۳ را در نظر بگیرید. به وضوح $T \cup T'$ به حداقل T بازه واحد نیاز دارند تا نقاط مجموعه T را که هر دو نقطه در آن فاصله بیش از ۱ دارند را بپوشانند. طبق فرضی که در مورد $U(OPT_{max})$ داشتیم، هر بازه در $U(OPT_{max})$ یک نقطه در انتهای سمت چپ خود دارد. فرض کنید $\mathcal{T}$ مجموعه این نقاط باشد. ادعا می‌کنیم $|T| \geqslant \frac{\mathcal{T}}{2}$.

برای این منظور، نشان می‌دهیم با اضافه کردن p به T، حداکثر دو نقطه از $\mathcal{T}$ که نشانه‌گذاری نشده‌اند (به این معنی که در مجموعه $mark$ نیستند) می‌توانند به $mark$ اضافه شوند. توجه کنید که وقتی ما p را به T اضافه می‌کنیم:

- تنها یک نقطه از $\mathcal{T}$ می‌تواند رنگی یکسان با p داشته باشد چون تمام نقاط در $\mathcal{T}$ دارای رنگی متفاوت از هم هستند.
- حداکثر یک نقطه نشانه‌گذاری نشده در $\mathcal{T}$ وجود دارد که با p حداکثر فاصله ۱ دارد زیرا هر دو نقطه در $\mathcal{T}$ دارای فاصله بیش از ۱ هستند و p سمت چپ‌ترین نقطه نشانه‌گذاری نشده است.

بنابراین، با اضافه کردن p در T، حداکثر دو نقطه نشانه‌گذاری نشده از $\mathcal{T}$ می‌توانند نشانه‌گذاری شوند (به این معنی که به مجموعه $mark$ اضافه شوند). در پایان الگوریتم، تمام نقاط $\mathcal{T}$ نشانه‌گذاری شده‌اند، پس $|T| \geqslant \frac{\mathcal{T}}{۲}$. در نتیجه ضریب تقریب این الگوریتم $\frac{۱}{۲}$ است.

$\square$

فصل ۵

نتیجه‌گیری و کارهای آتی

در این پژوهش، در مورد پیدا کردن کران و تقریب آن برای مسائل هندسی با داده‌های نادقیق در فضای d-بعدی بحث کردیم. برای داده‌های نادقیق از مدل مجموعه نقاط رنگی گسسته، که حالتی خاص از مدل نامشخص هستند، استفاده شد.

۵-۱ قطر در مدل مجموعه نقاط رنگی گسسته

در بخش ۳ مسئله قطر در حالت کمینه را مورد بررسی قرار دادیم. برای آن یک الگوریتم با ضریب تقریب $(۱ + \epsilon)$ ارائه کردیم. این نتیجه در مقایسه با نتایج قبلی که توسط ژانگ و همکاران [۲۵]، فلایشر و ژو [۲۶، ۲۷] و کانسگرا و همکاران [۳۳] برای این مسئله ارائه شده بود را بهبود داد.

از جمله مسائلی که می‌تواند در ادامه این پژوهش مورد مطالعه قرار بگیرید، بررسی مسئله کوچکترین قطر در مدل‌های دیگر است. برای مثال مسئله قطر در مدل غیردقیق در فضای دو بعدی توسط لوفلر و ون‌کرولد [۲۰] مورد بررسی قرار گرفت و برای آن یک الگوریتم با زمان $O(n \log n)$ برای حالتی‌که نقاط با مربع مدل شده‌اند و بعلاوه یک الگوریتم تقریبی با ضریب تقریب $(۱ + \epsilon)$ با زمان $O(n^{c\epsilon^{\frac{۱}{۲}}})$ برای حالتی‌که نقاط با دایره مدل شده‌اند ارائه کردند. بعد از آن، کروگر [۲۱] این مسئله را برای تعمیم به ابعاد بالاتر بررسی کرد ولی موفق نبود. به نظر می‌رسد بتوان با استفاده از نوعی نمونه‌گیری و تبدیل مسئله به یک مسئله با مدل مجموعه نقاط رنگی، یک الگوریتم تقریبی کارا، دست‌کم برای حالتی که ورودی‌ها ابعادی متناسب باهم دارند، بدست آورد.

۵-۲ پوشش واحد در مدل مجموعه نقاط رنگی گسسته

در بخش ۴ مسئله پوشش واحد را زمانی‌که نقاط نادقیق هستند و با مدل مجموعه نقاط رنگی گسسته مدل شده‌اند بررسی کردیم. برای پوشش رنگی کمینه، نشان دادیم مسئله ان‌پی-سخت است و برای آن هیچ الگوریتم تقریبی با ضریب تقریب ثابت در زمان چندجمله‌ای وجود ندارد.بعلاوه یک الگوریتم با ضریب تقریب $\ln(m)$ ارائه کردیم و همچنین یک الگوریتم با ضریب تقریب $۲f$ در یک بعد. درحالیکه پوشش رنگی کمینه ان‌پی-سخت است حتی برای حالتی‌که $f = ۲$ است، الگوریتم قبلی یک الگوریتم تقریبی با ضریب تقریب ثابت برای fهای ثابت را نتیجه می‌دهد.

برای پوشش رنگی بیشینه، ثابت کردیم مسئله حتی در یک بعد برای $f \geqslant ۳$ ان‌پی-سخت است و یک الگوریتم با ضریب تقریب ۲ برای آن در یک بعد ارائه کردیم. بعلاوه برای کارهای آتی می‌توان موارد زیر را در نظر گرفت:

- آیا الگوریتمی با ضریب تقریب کمتر از $۲f$ برای مسئله پوشش رنگی کمینه وجود دارد. برای حالت خاص $f = ۲$، الگوریتم ارائه شده یک الگوریتم با ضریب تقریب ۴ را نتیجه می‌دهد. در این مورد، با استفاده مسئله پوشش راسی [۱] و کاهش آن به این مسئله، می‌دانیم نه الگوریتم تقریبی با ضریب تقریب بهتر از ۱/۳۶۰۶ وجود دارد مگر $P = NP$ [۴۹] و نه الگوریتم تقریبی با ضریب تقریب بهتر از ۲ مگر حدس بازی منحصر به فرد صحیح باشد [۵۰]. هنوز بین این حدپایین‌ها و ضریب تقریب الگوریتم ما فاصله وجود دارد.
- با کاهش مسئله ۳-صدق‌پذیری به مسئله پوشش رنگی بیشینه ، نشان دادیم مسئله در یک بعد برای $f > ۲$ ان‌پی-سخت است، اما وضعیت مسئله برای $f = ۲$ باز است.

[1]Vertex Cover

کتاب‌نامه

[1] M. I. Shamos. Computational geometry. *Phd's thesis, Yale University*, 1978.

[2] T. Tasdizen and R. Whitaker. Feature preserving variational smoothing of terrain data. In *IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision, O. Faugeras and N. Paragios, Eds*, 2003.

[3] I. Kant. Kritik der reinen vernunft [critique of pure reason]. *Insel, Darmstadt*, 1781.

[4] A. Einstein. Zur elektrodynamik bewegter körper. *Annalen der physik*, 322(10):891–921, 1905.

[5] M. Goodchild and J. Zhang. *Uncertainty in geographical information.* CRC press, 2002.

[6] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal*, 16(4):523–544, 2007.

[7] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. Nabar, T. Sugihara, and J. Widom. Trio: a system for data, uncertainty, and lineage. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 1151–1154. VLDB Endowment, 2006.

[8] G. Cormode and A. McGregor. Approximation algorithms for clustering uncertain data. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 191–200. ACM, 2008.

[9] G. Cormode and M. Garofalakis. Histograms and wavelets on probabilistic data. *Knowledge and Data Engineering, IEEE Transactions on*, 22(8):1142–1157, 2010.

[10] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao, and S. Prabhakar. Indexing multi-dimensional uncertain data with arbitrary probability density functions. In *Proceedings*

of the 31st international conference on Very large data bases, pages 922–933. VLDB Endowment, 2005.

[11] P. K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi. Indexing uncertain data. In *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 137–146. ACM, 2009.

[12] G. Cormode, F. Li, and K. Yi. Semantics of ranking queries for probabilistic data and expected ranks. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 305–316. IEEE, 2009.

[13] J. B. T. Zhang. Support vector classification with input data uncertainty. *Advances in neural information processing systems*, 17:161–169, 2005.

[14] Y. Zou and K. Chakrabarty. Uncertainty-aware and coverage-oriented deployment for sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):788–798, 2004.

[15] D. Salesin, J. Stolfi, and L. Guibas. Epsilon geometry: building robust algorithms from imprecise computations. In *Proceedings of the fifth annual symposium on Computational geometry*, pages 208–217. ACM, 1989.

[16] L. Guibas, D. Salesin, and J. Stolfi. Constructing strongly convex approximate hulls with inaccurate primitives. *Algorithmica*, 9(6):534–560, 1993.

[17] M. Held and J. S. Mitchell. Triangulating input-constrained planar point sets. *Information Processing Letters*, 109(1):54–56, 2008.

[18] M. Löffler and J. Snoeyink. Delaunay triangulation of imprecise points in linear time after preprocessing. *Computational Geometry*, 43(3):234–242, 2010.

[19] T. Nagai and N. Tokura. Tight error bound of goemetric problems on convex objects with imprecise coordinates. In *Discrete and Computational Geometry*, pages 252–263. Springer, 2001.

[20] M. Löffler and M. van Kreveld. Largest bounding box, smallest diameter, and related problems on imprecise points. *Computational Geometry*, 43(4):419–433, 2010.

[21] H. Kruger. Basic measures for imprecise point sets in r$^d. Master's thesis, Utrecht University$, 2008.

[22] P. Kamousi, T. M. Chan, and S. Suri. Stochastic minimum spanning trees in euclidean spaces. In *Proceedings of the 27th annual ACM symposium on Computational geometry*, pages 65–74. ACM, 2011.

[23] P. Kamousi, T. M. Chan, and S. Suri. Closest pair and the post office problem for stochastic points. In *Proceedings of the 12th international conference on Algorithms and data structures*, pages 548–559, Berlin, Heidelberg, 2011.

[24] C. Fan, W. Ju, J. Luo, and B. Zhu. On some geometric problems of color-spanning sets. In *Frontiers in Algorithmics and Algorithmic Aspects in Information and Management*, pages 113–124. Springer, 2011.

[25] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, ICDE '09, pages 688–699, Washington, DC, USA, 2009. IEEE Computer Society.

[26] R. Fleischer and X. Xu. Computing minimum diameter color-spanning sets. In *Frontiers in Algorithmics*, pages 285–292. Springer, 2010.

[27] R. Fleischer and X. Xu. Computing minimum diameter color-spanning sets is hard. *Information Processing Letters*, 111(21):1054–1056, 2011.

[28] D. P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of voronoi surfaces and its applications. *Discrete & Computational Geometry*, 9(1):267–291, 1993.

[29] M. Sharir and P. K. Agarwal. *Davenport-Schinzel sequences and their geometric applications*. Cambridge university press, 1995.

[30] M. Abellanas, F. Hurtado, C. Icking, R. Klein, E. Langetepe, L. Ma, B. Palop, and V. Sacristán. The farthest color voronoi diagram and related problems. In *Abstracts 17th European Workshop Comput. Geom*, pages 113–116, 2001.

[31] S. Das, P. P. Goswami, and S. C. Nandy. Recognition of minimum width color-spanning corridor and minimum area color-spanning rectangle. In *Computational Science and Its Applications–ICCSA 2005*, pages 827–837. Springer, 2005.

[32] P. Khanteimouri, A. Mohades, M. A. Abam, and M. R. Kazemi. Spanning colored points with intervals. In *CCCG*, pages 265–270, 2013.

[33] M. E. Consuegra, G. Narasimhan, and S.-i. Tanigawa. Geometric avatar problems.

[34] Y.-S. Myung, C.-H. Lee, and D.-W. Tcha. On the generalized minimum spanning tree problem. *Networks*, 26(4):231–241, 1995.

[35] A. Dumitrescu and J. S. Mitchell. Approximation algorithms for tsp with neighborhoods in the plane. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 38–46. Society for Industrial and Applied Mathematics, 2001.

[36] J. S. Mitchell. A ptas for tsp with neighborhoods among fat regions in the plane. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 11–18. Society for Industrial and Applied Mathematics, 2007.

[37] K. Elbassioni, A. V. Fishkin, N. H. Mustafa, and R. Sitters. Approximation algorithms for euclidean group tsp. In *Automata, Languages and Programming*, pages 1115–1126. Springer, 2005.

[38] R. Fraser. *Algorithms for Geometric Covering and Piercing Problems.* PhD thesis, University of Waterloo, 2012.

[39] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *Journal of Algorithms*, 38(1):91–109, 2001.

[40] Y. Zhou and S. Suri. Algorithms for a minimum volume enclosing simplex in three dimensions. *SIAM Journal on Computing*, 31(5):1339–1357, 2002.

[41] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *Journal of the ACM (JACM)*, 51(4):606–635, 2004.

[42] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are np-complete. *Information processing letters*, 12(3):133–137, 1981.

[43] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of the ACM (JACM)*, 32(1):130–136, 1985.

[44] G. K. Das, R. Fraser, A. Lopez-Ortiz, and B. G. Nickerson. *On the discrete unit disk cover problem*, pages 146–157. Springer, 2011.

[45] F. Claude, G. K. Das, R. Dorrigiv, S. Durocher, R. Fraser, A. López-Ortiz, B. G. Nickerson, and A. Salinger. An improved line-separable algorithm for discrete unit disk cover. *Discrete Mathematics, Algorithms and Applications*, 2(01):77–87, 2010.

[46] S. Funke, A. Kesselman, F. Kuhn, Z. Lotker, and M. Segal. Improved approximation algorithms for connected sensor cover. *Wireless networks*, 13(2):153–164, 2007.

[47] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang. Two-tiered constrained relay node placement in wireless sensor networks: efficient approximations. In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, pages 1–9. IEEE.

[48] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of computer and system sciences*, 9(3):256–278, 1974.

[49] I. Dinur and S. Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, pages 439–485, 2005.

[50] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2-$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

Abstract

In this research, we consider the question of finding approximate bounds on some of computational geometry problems with uncertain data. We use color spanning set to model the uncertainty. Given a set of n points colored with m colors in d-dimension. The problem of minimum diameter in color spanning set model is finding a set of m points with distinct colors so as to minimize diameter of the points. We give an algorithm with an approximation factor $(1+\epsilon)$ with running time $O(n^2(n+2^{\frac{1}{\epsilon^d}}(\frac{n}{\epsilon^d}+(\frac{1}{\epsilon^d})^2))$ which improves the previous results. Next, we consider a new problem of finding bounds on unit covering in color-spanning set model: *Minimum color spanning-set ball covering* problem is to select m points of different colors minimizing the minimum number of unit balls needed to cover them. Similarly, *Maximum color spanning-set ball covering* problem is to choose one point of each color to maximize the minimum number of needed unit balls. While Minimum color spanning-set ball covering problem is NP-hard and also hard to approximate within any constant factor, even in one dimension, we propose an $\ln(m)-$approximation algorithm for it. Moreover, in one dimensional case, we present a constant-factor approximation algorithm for a fixed f where f is the maximum frequency of the colors. For Maximum color spanning-set ball covering problem, we prove that it is NP-hard and propose an approximation algorithm within a factor $\frac{1}{2}$ in one dimensional case.

Keywords: Computational Geometry, Approximation algorithms, Uncertain Data, Diameter, Unit Covering

Sharif University of Technology

Department of Computer Engineering

M.Sc. Thesis

Approximation Algorithms for Some Problems in Computational Geometry with Uncertain Data

By:

Hamid Homapour

Supervisor:

Dr. Mohammad Ghodsi

August 2013