دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد
گرایش مهندسی نرم‌افزار

عنوان:

الگوریتم‌های تقریبی برای خوشه‌بندی نقاط در مدل جویبار داده

نگارش:

بهنام حاتمی ورزنه

استاد راهنما:

حمید ضرابی‌زاده

شهریور ۱۳۹۴

بسم الله الرحمن الرحیم

به نام خدا

دانشگاه صنعتی شریف

دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی ارشد

عنوان: الگوریتم‌های تقریبی برای خوشه‌بندی نقاط در مدل جویبار داده

نگارش: بهنام حاتمی ورزنه

کمیته‌ی ممتحنین

استاد راهنما: حمید ضرابی‌زاده   امضاء:

استاد مشاور: حمید بیگی   امضاء:

استاد مدعو: استاد ممتحن   امضاء:

تاریخ:

چکیده

این قسمت باید تکمیل گردد.

کلیدواژه‌ها: خوشه‌بندی، kـ مرکز، جویبار داده، الگوریتم تقریبی

فهرست مطالب

فهرست شکل‌ها

فهرست جدول‌ها

فصل ۱

مقدمه

مسئله‌ی **خوشه‌بندی** یکی از مهم‌ترین مسائل داده‌کاوی[۱] به حساب می‌آید. در این مسئله هدف، دسته‌بندی تعدادی جسم به گونه‌ای است که اجسام در یک دسته (خوشه)، نسبت به یکدیگر در برابر دسته‌های دیگر شبیه‌تر باشند (معیارهای متفاوتی برای تشابه تعریف می‌گردد). این مسئله در حوزه‌های مختلفی از علوم کامپیوتر، از جمله داده‌کاوی، جست‌وجوی الگو[۲]، پردازش تصویر[۳]، بازیابی اطلاعات[۴] و بایوانفورماتیک[۵] مورد استفاده قرار می‌گیرد [۱].

مسئله‌ی خوشه‌بندی، به‌خودی‌خود یک مسئله‌ی الگوریتمی به حساب نمی‌آید. راه‌حل‌های الگوریتمی بسیار زیادی برای خوشه‌بندی تعریف شده است. به طور کلی این الگوریتم‌ها به چهار دسته‌ی زیر تقسیم‌بندی می‌گردند:

- خوشه‌بندی‌های سلسه‌مراتبی[۶]
- خوشه‌بندی‌های مرکزگرا[۷]
- خوشه‌بندی‌های مبتنی بر توزیع[۸] نقاط

[۱] data mining
[۲] pattern recognition
[۳] image analysis
[۴] information retrieval
[۵] bioinformatics
[۶] hierarchical clustering
[۷] centroid-based clustering
[۸] distribution-based

- خوشه‌بندی‌های مبتنی بر چگالی[۹] نقاط

در عمل هیچ کدام از راه‌حل‌های بالا بر دیگری ارجهیت ندارند و باید راه‌حل مد نظر را متناسب با کاربرد مطرح مورد استفاده قرار داد. به طور مثال استفاده از الگوریتم‌های مرکزگرا، برای خوشه‌بندی‌های غیر محدب به خوبی عمل نمی‌کند. یکی از راه‌حل‌های شناخته شده برای مسئله‌ی خوشه‌بندی، الگوریتم k**ــ مرکز** است. در این الگوریتم هدف، پیدا کردن k نقطه به عنوان مرکز دسته‌ها است به طوری‌که شعاع دسته‌ها تا حد ممکن کمینه شود. در نظریه‌ی گراف، مسئله‌ی kــ مرکز متریک[۱۰] یا مسئله‌ی **استقرار تجهیزات متریک**[۱۱] یک مسئله‌ی **بهینه‌سازی ترکیبیاتی**[۱۲] است. فرض کنید که n شهر و فاصله‌ی دوبه‌دوی آن‌ها، داده‌شده است. می‌خواهیم k انبار در شهرهای مختلف بسازیم به‌طوری‌که بیش‌ترین فاصله‌ی هر شهری از نزدیک‌ترین انبار به خود، کمینه گردد. در حالت نظریه‌ی گراف آن، این بدان معناست که مجموعه‌ای شامل k رأس انتخاب کنیم به‌طوری‌که بیش‌ترین فاصله‌ی هر نقطه از نزدیک‌ترین نقطه‌اش داخل مجموعه‌ی k عضوی کمینه گردد. توجه نمایید که فاصله‌ی بین رئوس باید در فضای متریک[۱۳] باشند و یا به زبان دیگر، یک گراف کامل داشته باشیم که فاصله‌ها در آن در رابطه‌ی مثلثی[۱۴] صدق می‌کنند. مثالی از مسئله‌ی ۲ــ مرکز را در شکل ۱ــ۱ نشان داده شده است.

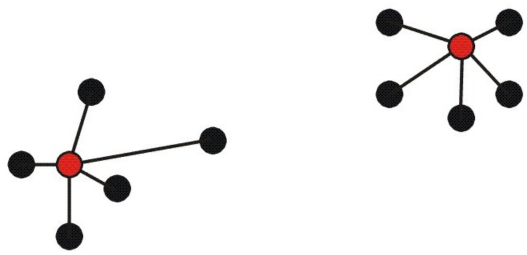
در این پژوهش، مسئله‌ی kــ مرکز با متریک‌های خاص و برای kهای کوچک مورد بررسی قرار گرفته است و از جنبه‌های متفاوتی بهبود یافته است.

۱ــ۱ تعریف مسئله

تعریف دقیق‌تر مسئله‌ی kــ مرکز در زیر آمده است:

مسئله‌ی ۱ــ۱ *یک گراف کامل بدون جهت $G = (V, E)$ با فاصله‌های $d(v_i, v_j)$ که از نامساوی مثلثی پیروی می‌کنند داده‌شده است. زیرمجموعه $S \subseteq V$ با اندازه‌ی k را انتخاب کنید به‌طوری‌که عبارت زیر*

[۹] density-based
[۱۰] metric
[۱۱] metric facility location
[۱۲] combinatorial optimization
[۱۳] metric space
[۱۴] triangle equation

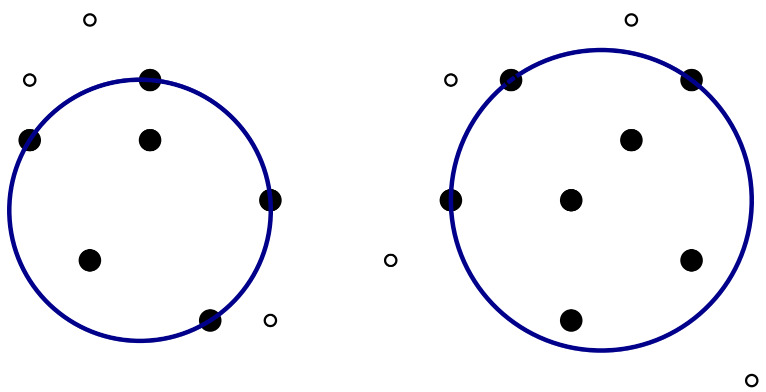


شکل ۱ـ۱: نمونه‌ای ازمسئله‌ی ۲ـ مرکز

را کمینه کند:

$$\max_{v\in V}\{\min_{s\in S} d(v,s)\}$$

گونه‌های مختلفی از مسئله‌ی kـ مرکز با محدودیت‌های متفاوتی به‌وسیله‌ی پژوهشگران مورد مطالعه قرارگرفته است. از جمله می‌توان حالتی که در بین داده‌های ورودی داده‌های پرت وجود دارد و قبل از خوشه‌بندی می‌توانیم تعدادی نقاط ورودی را حذف نماییم و سپس به خوشه‌بندی بپردازیم. سختی این مسئله از آنجاست که نه تنها باید مسئله‌ی خوشه‌بندی را حل نمود، بلکه در ابتدا باید تصمیم گرفت که کدام یک از داده‌ها را به‌عنوان پرت در نظر گرفت که بهترین جواب در زمان خوشه‌بندی به دست آید. در واقع اگر تعداد نقاط پرتی که مجاز به حذف است، برابر صفر باشد، مسئله به مسئله‌ی kـ مرکز تبدیل می‌شود. نمونه‌ای از مسئله‌ی ۲ـ مرکز با ۷ داده‌ی پرت را در شکل ۱ـ۲ می‌توانید ببینید. تعریف دقیق‌تر این مسئله در زیر آمده است:



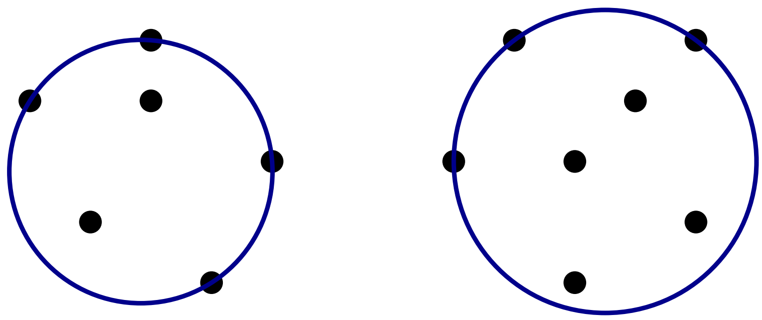
شکل ۱ـ۲: نمونه‌ای ازمسئله‌ی ۲ـ مرکز با داده‌های پرت

مسئله‌ی ۱ـ۲ *یک گراف کامل بدون جهت $G = (V, E)$ با فاصله‌های $d(v_i, v_j)$ که از نامساوی مثلثی پیروی می‌کنند داده‌شده است. زیرمجموعه $S \subseteq V$ با اندازه‌ی k و $Z \subseteq V$ به‌اندازه‌ی z را انتخاب کنید به‌طوری‌که عبارت زیر را کمینه کند:*

$$\max_{v \in (V-Z)} \{\min_{s \in S} d(v, s)\}$$

گونه‌ای دیگری که در مسئله‌ی kـمرکز که در سال‌های اخیر مورد توجه قرار گرفته است، حالت جویبار داده‌ی آن است. در این گونه، در ابتدا تمام نقاط در دسترس نیست، بلکه به مرور زمان نقاط در دسترس قرار می‌گیرند. محدودیت دومی که وجود دارد، محدودیت شدید حافظه است، به طوری که نمی‌توان تمام نقاط را در حافظه نگه داشت و به طور معمول باید مرتبه‌ی حافظه‌ای کم‌تر از مرتبه حافظه‌ی خطی[۱۵] متناسب با تعداد نقاط استفاده نمود. مدلی که ما در این پژوهش بر روی آن تمرکز داریم مدل جویبار داده تک‌گذره[۲] است. یعنی تنها یک بار می‌توان از ابتدا تا انتهای داده‌ها را بررسی کرد.

یکی از دغدغه‌هایی که در مسائل جویبار داده وجود دارد، عدم داشتن تمام نقاط است. بنابراین در بعضی موارد ممکن است برای مسئله‌ی kـمرکز، مرکزی برای یک دسته انتخاب شود که در بین نقاط ورودی نیست. نمونه‌ای از مسئله‌ی ۲ـمرکز در حالت پیوسته را در شکل ۱ـ۳ نشان داده شده است. این مسئله تنها برای L_pـمتریک مطرح می‌شود، زیرا مرکز دسته‌ها ممکن است در هر نقطه از فضا قرار بگیرد و ما نیاز داریم که فاصله‌ی آن را از تمام نقاط بدانیم. تعریف دقیق مسئله در زیر آمده است:



شکل ۱ـ۳: نمونه‌ای ازمسئله‌ی ۲ـمرکز در حالت پیوسته

مسئله‌ی ۱ـ۳ *مجموعه‌ی U از نقاط فضای d بعدی داده شده است. زیرمجموعه $S \subseteq U$ با اندازه‌ی k*

[۱۵] sublinear

را انتخاب کنید به‌طوری‌که عبارت زیر را کمینه کند:

$$\max_{u \in U}\{\min_{s \in S} L_p(u,s)\}$$

از آنجایی که گونه‌ی جویبار داده و داده پرت مسئله‌ی kـ مرکز به تازگی مورد توجه قرار گرفته است و نتایج به‌دست‌آمده قابل بهبود است، در این تحقیق سعی شده است که تمرکز بر روی این‌گونه‌ی خاص از مسئله باشد. همچنین در این پژوهش سعی می‌شود گونه‌های مسئله را برای انواع متریک‌ها و برای kهای کوچک نیز مورد بررسی قرار گیرد.

۱ـ۲ اهمیت موضوع

مسئله‌ی kـ مرکز و گونه‌های آن کاربردهای زیادی در داده‌کاوی دارند. این الگوریتم یکی از رایج‌ترین الگوریتم‌های مورد استفاده برای خوشه‌بندی محسوب می‌شود. به علت افزایش حجم داده‌ها و تولید داده‌ها در طول زمان، مدل جویبار داده‌ی مسئله در سال‌های اخیر مورد توجه قرار گرفته است. مسئله‌ی kـ مرکز در بعضی مسائل مانند ارسال نامه از تعدادی مراکز پستی به گیرنده‌ها، نیاز دارد که تمام نامه‌ها را به دست گیرنده‌ها برساند و در نتیجه باید نقاط را پوشش دهد، ولی برای بعضی از مسائل تجاری، نیازی به پوشش تمام نقاط هدف نیست و از لحاظ اقتصادی به‌صرفه است که نقاط پرت را در نظر نگرفت. به طور مثال Kmarts اعلام کرده است که به ۸۸ درصد جمعیت آمریکا با فاصله‌ای حداکثر ۶ مایل، می‌تواند سرویس دهد. درصورتی‌که اگر این شرکت قصد داشت تمام جمعیت آمریکا را پوشش دهد، نیاز داشت تعداد شعب یا شعاع پوشش خود را به میزان زیادی افزایش دهد که از لحاظ اقتصادی به‌صرفه نیست[۳]. علاوه بر دو گونه‌ی مطرح‌شده در این قسمت، گونه‌های دیگر از مسئله‌ی kـ مرکز در مرجع [۳] آمده است.

۱ـ۳ ادبیات موضوع

همان‌طور که ذکر شد مسئله‌ی kـ مرکز در حالت داده‌های پرت و جویبار داده، گونه‌های مختلف مسئله‌ی kـ مرکز هستند و در حالت‌های خاصی به مسئله‌ی kـ مرکز کاهش پیدا می‌کنند. مسئله‌ی kـ مرکز در

حوزه‌ی مسائل ان‌پی‌ـ‌سخت[۱۶] قرار می‌گیرد و با فرض $P \neq NP$ الگوریتم دقیق با زمان چندجمله‌ای برای آن وجود ندارد. بنابراین برای حل کارای[۱۷] این مسائل از الگوریتم‌های تقریبی[۱۸] استفاده می‌شود.

برای مسئله‌ی kـ‌مرکز، دو الگوریتم تقریبی معروف وجود دارد. در الگوریتم اول، که به روش حریصانه[۱۹] عمل می‌کند، در هر مرحله بهترین مرکز ممکن را انتخاب می‌کند به طوری تا حد ممکن از مراکز قبلی دور باشد. این الگوریتم، الگوریتم تقریبی با ضریب تقریب ۲ ارائه می‌دهد. در الگوریتم دوم، با استفاده از مسئله‌ی مجموعه‌ی غالب کمینه[۲۰]، الگوریتمی با ضریب تقریب ۲ ارائه می‌گردد. همچنین ثابت شده است، که بهتر از این ضریب تقریب، الگوریتمی نمی‌توان ارائه شود مگر آنکه $P = NP$ باشد.

۱ـ۴ اهداف تحقیق

در این پایان‌نامه مسئله‌ی kـ‌مرکز در حالت جویبار داده با داده‌های پرت در حالت‌های مختلف مورد بررسی قرار می‌گیرد و سعی خواهد شد که نتایج قبلی در این مسائل را از جنبه‌های مختلفی مورد بهبود قرار دهد.

اولین مسئله‌ای که مورد بررسی قرار گرفته است، ارائه الگوریتمی تقریبی، برای مسئله‌ی ۱ـ‌مرکز در حالت جویبار داده است به طوری که، نه تنها تمام نقاط ورودی را می‌پوشاند بلکه تضمین می‌کند که دایره‌ی بهینه‌ی جواب ۱ـ‌مرکز برای نقاط ورودی را نیز می‌پوشاند. در تلاش اول، الگوریتم جدیدی با حافظه و زمان به‌روزرسانی $\mathcal{O}(\frac{d}{\epsilon})$ و با ضریب تقریب $\epsilon+۱/۸$، برای این مسئله ارائه گردید. با بررسی‌های بیش‌تر، الگوریتم دیگری با ضریب تقریب ۱/۶۹، با الگوریتمی کاملا متفاوت، با حافظه‌ی $\mathcal{O}(d)$ برای این مسئله ارائه گردید.

مسئله‌ی دومی که مورد بررسی قرار گرفت، مسئله‌ی ۱ـ‌مرکز در حالت جویبار داده با داده‌های پرت است. در ابتدا الگوریتمی ساده با حافظه‌ی $\mathcal{O}(zd)$ و زمان به‌روزرسانی $\mathcal{O}(zd\log(z))$ با ضریب تقریب ۲ برای این مسئله ارائه گردید. در بررسی‌های بعدی، با استفاده از نتایج بدست آمده در قسمت قبل، برای z های کوچک، الگوریتمی با حافظه‌ی $\mathcal{O}(dz^d)$ با ضریب تقریب ۱/۶۹ برای این مسئله ارائه شد

[۱۶] NP-hard
[۱۷] efficient
[۱۸] Approximation Algorithm
[۱۹] greedy
[۲۰] dominating set

که ضریب تقریب بهترین الگوریتم موجود را، از ۱/۷۹ به ۱/۶۹ کاهش می‌دهد.

مسئله‌ی سومی که مورد بررسی قرار گرفت، مسئله‌ی ۲ ـ مرکز در حالت جویبار داده با داده‌های پرت است. بهترین الگوریتمی که در حال حاضر برای این مسئله وجود دارد، یک الگوریتم با ضریب تقریب $4 + \epsilon$ است. ما با ارائه الگوریتمی جدید، الگوریتمی با ضریب تقریب $1.8 + \epsilon$ برای این مسئله ارائه شد که بهبودی قابل توجه برای این مسئله است.

۱ـ۵ ساختار پایان‌نامه

این پایان‌نامه در پنج فصل به شرح زیر ارائه خواهد شد. در فصل دوم به بیان مفاهیم و تعاریف مرتبط با موضوعات مورد بررسی در بخش‌های دیگر خواهیم پرداخت. فصل سوم این پایان‌نامه شامل مطالعه و بررسی کارهای پیشین انجام شده مرتبط با موضوع این پایان‌نامه خواهد بود. این فصل در سه بخش تنظیم گردیده است. در بخش اول، مسئله‌ی kـ مرکز مورد بررسی قرار می‌گیرد. در بخش دوم، حالت جویبار داده‌ی مسئله و مجموعه هسته‌های[۲۱] مطرح برای این مسئله مورد بررسی قرار می‌گیرد. در نهایت، در بخش سوم، مسئله‌ی kـ مرکز با داده‌های پرت مورد بررسی قرار می‌گیرد.

در فصل چهارم، نتایج جدیدی که در این پایان‌نامه به دست آمده است، ارائه می‌شود. این نتایج شامل الگوریتم‌های جدید برای مسئله‌ی ۱ ـ مرکز در حالت جویبار داده، مسئله‌ی ۱ ـ مرکز با داده‌های پرت در حالت جویبار داده و مسئله‌ی دو مرکز در حالت جویبار داده با داده‌های پرت می‌شود.

در فصل پنجم، به جمع‌بندی کارهای انجام شده در این پژوهش و ارائه‌ی پیشنهادهایی برای انجام کارهای آتی و تعمیم‌هایی که از راه‌حل ارائه شده وجود دارد، خواهیم پرداخت.

[۲۱] coreset

فصل ۲

مفاهیم اولیه

در این فصل به تعریف و بیان مفاهیم پایه‌ای مورد استفاده در فصل‌های بعد می‌پردازیم. با توجه به مطالب مورد نیاز در فصل‌های آتی، مطالب این فصل به سه بخش، مسائل ان‌پی_سخت، الگوریتم‌های تقریبی و الگوریتم‌های جویبارداده تقسیم می‌شود.

۲_۱ مسائل ان‌پی_سخت

یکی از اولین سوال‌های بنیادی مطرح در علم کامپیوتر، اثبات **عدم حل پذیری** بعضی از مسائل است. به عنوان نمونه، می‌توان از دهمین مسئله‌ی هیلبرت[۱] در کنگره‌ی ریاضی یاد کرد. هیلبرت این مسئله را اینگونه بیان کرذ: "فرآیندی طراحی کنید که در تعداد متناهی گام بررسی کند که آیا یک چندجمله‌ای، ریشه‌ی صحیح[۲] دارد یا خیر ." با مدل محاسباتی که به‌وسیله‌ی تورینگ[۳] ارائه شد، این مسئله معادل پیدا کردن الگوریتمی برای این مسئله است که اثبات می‌شود امکان‌پذیر نیست [۴]. برخلاف مثال بالا، عمده‌ی مسائل علوم کامپیوتر از نعد بالا نیستند و برای طیف وسیعی از آن‌ها، الگوریتم‌های پایان‌پذیر وجود دارد. بیش‌تر تمرکز علوم کامپیوتر هم بر روی چنین مسائلی است.

اگر چه برای اکثر مسائل الگوریتمی پایان‌پذیر وجود دارد، اما وجود چنین الگوریتمی لزومی بر حل

[۱] Hilbert
[۲] integral root
[۳] Touring

شدن چنین مسائلی نیست. در عمل، علاوه بر وجود الگوریتم، میزان کارامدی[۴] الگوریتم نیز مطرح می‌گردد. به طور مثال، اگر الگوریتم حل یک مسئله مرتبه‌ی بالا یا نمایی داشته باشد، الگوریتم ارائه شده برای آن مسئله برای ورودی‌های نسبتا بزرگ قابل اجرا نیست و نمی‌توان از آن‌ها برای حل مسئله استفاده کرد. برای تشخیص و تمیز کارآمدی الگوریتم‌های مختلف و همچنین میزان سختی مسائل در امکان ارائه‌ی الگوریتم‌های کارآمد یا غیرکارآمد، نظریه‌ی پیچیدگی[۵]، دسته‌بندی‌های مختلفی برای سختی مسائل و حل‌پذیری آن‌ها ارائه داده است تا بتوان به‌طور رسمی[۶] در مورد این معیارها صحبت کرد. برای دسته‌بندی مسائل در نظریه‌ی پیچیدگی، ابتدا آن‌ها را به صورت تصمیم‌پذیر بیان می‌کنند.

مسئله‌ی ۲ـ۱ (*مسائل تصمیم‌گیری*)[۷] *به دسته‌ای از مسائل گفته می‌شود که پاسخ آن‌ها تنها بله یا خیر است.*

به عنوان مثال، اگر بخواهیم مسئله‌ی ۱ـ مرکز در فضای $\mathbb{R}^d$ را به صورت تصمیم پذیر بیان کنیم، به مسئله‌ی زیر می‌رسیم:

مسئله‌ی ۲ـ۲ (*نسخه‌ی تصمیم پذیر ۱ـ مرکز*) *مجموعه‌ی نقاط در فضا $\mathbb{R}^d$ و شعاع r داده شده است، آیا دایره‌ای به شعاع r وجود دارد که تمام نقاط را بپوشاند؟*

در نظریه‌ی پیچیدگی، می‌توان گفت عمده‌ترین دسته‌بندی موجود، دسته‌بندی مسائل تصمیم‌گیری به مسائل پی(P) و ان‌پی(NP) است. رده‌ی مسائل P، شامل تمامی مسائل تصمیم‌گیری است که راه‌حل چندجمله‌ای برای آن‌ها وجود دارد. از طرفی رده‌ی مسائل NP، شامل تمامی مسائل تصمیم‌گیری است که در زمان چندجمله‌ای قابل صحت‌سنجی[۸] اند. تعریف صحت‌سنجی در نظریه پیچیدگی، یعنی اگر جواب مسئله‌ی تصمیم‌گیری بله باشد، می‌توان اطلاعات اضافی با طول چندجمله‌ای ارائه داد، که در زمان چندجمله‌ای از روی آن، می‌توان جواب بله الگوریتم را تصدیق نمود. به طور مثال، برای مسئله‌ی ۱ـ مرکز، کافی است به عنوان تصدیق جواب بله، مرکز دایره‌ی پوشاننده را الگوریتم ارائه دهد. در این صورت، می‌توان با مرتبه‌ی خطی بررسی نمود که تمام نقاط داخل این دایره قرار می‌گیرند یا نه. برای مطالعه‌ی بیش‌تر و تعاریف دقیق‌تر می‌توان به [۴] مراجعه نمود.

---

[۴] Efficiency

[۵] Complexity theory

[۶] formal

[۷] Decision problems

[۸] verifiable

فصل ۳

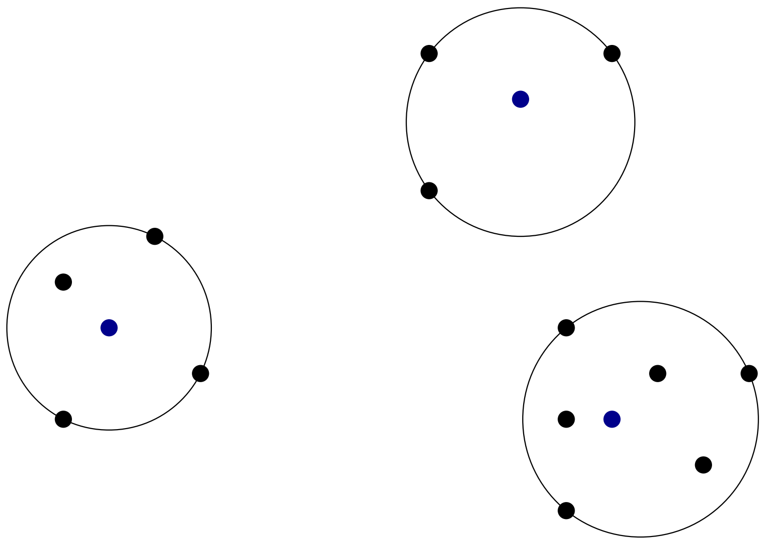
کارهای پیشین

در این فصل کارهای پیشین انجام‌شده روی مسئله‌ی kـ مرکز، در سه بخش مورد بررسی قرار می‌گیرد. در بخش اول، مسئله‌ی kـ مرکز مورد بررسی قرار می‌گیرد. در بخش دوم، حالت جویبار داده‌ی مسئله و مجموعه هسته‌های مطرح برای این مسئله مورد بررسی قرار می‌گیرد. در نهایت، در بخش سوم، مسئله‌ی kـ مرکز با داده‌های پرت مورد بررسی قرار می‌گیرد.

۳ـ۱ kـ مرکز در حالت ایستا

مسئله‌ی kـ مرکز به عنوان مسئله‌ی شناخته شده در علوم کامپیوتر مطرح است. این مسئله، در واقع یک مسئله‌ی بهینه سازی است که سعی در کاهش بیش‌ترین فاصله نقاط از مرکز دسته‌ها را دارد. سختی اصلی این مسئله در انتخاب مرکز دسته‌هاست. زیرا اگر بتوانیم مرکز دسته‌ها را به درستی تشخیص دهیم، کافی است هر نقطه را به دسته‌ای که نزدیک‌ترین مرکز را دارد، تخصیص دهیم. به وضوح چنین تخصیصی بهینه‌ترین تخصیص ممکن است. نمونه‌ای از این تخصیص را در شکل ۳ـ۱ نشان داده شده است.

در سال ۱۹۷۹، اثبات گردید که این مسئله یک مسئله‌ی ان‌پی‌ـ سخت است [۵]. حتی ثابت شده است که این مسئله در صفحه‌ی دو بعدی و با متریک اقلیدسی نیز ان‌پی‌ـ سخت است [۶]. فراتر از این، ثابت شده است که برای مسئله‌ی kـ مرکز با متریک دلخواه هیچ الگوریتم تقریبی با ضریب تقریب بهتر از ۲ وجود ندارد. ایده‌ی اصلی این کران پایین، کاهش مسئله‌ی پوشش رأسی، به مسئله‌ی kـ مرکز



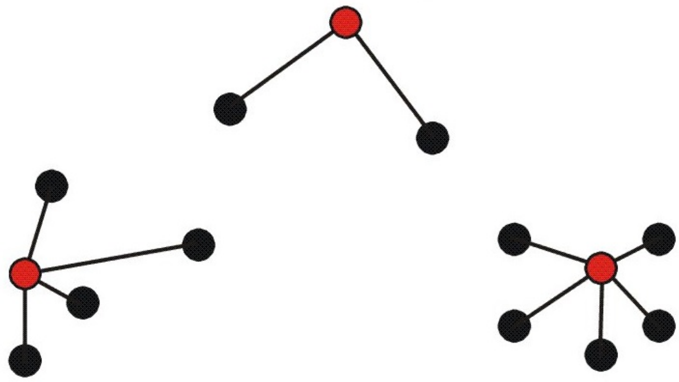
شکل ۳ــ۱: نمونه‌ای از تخصیص نقاط به ازای مراکز آبی رنگ.

است. برای چنین کاهشی کافی است، از روی گراف اصلی، یک گراف کامل بسازیم به طوری که معادل یال‌های گراف اصلی یال با وزن یک و به ازای بقیه یال‌های ممکن که در گراف اصلی نیستند، یال با وزن ۲ قرار می‌دهیم. حال اگر الگوریتمی بتواند مسئله‌ی kـ مرکز را با ضریب تقریب بهتر از ۲ حل نماید، آن‌گاه گراف جدید دارای یک kـ مرکز با شعاع کم‌تر از ۲ است، اگر و تنها اگر گراف اصلی دارای یک پوشش رأسی با اندازه‌ی k باشد. برای متریک L_2 یا فضای اقلیدسی[1] نیز ثابت شده است الگوریتم تقریبی با ضریب تقریب بهتر از ۱٫۸۲۲ وجود ندارد [۷].

گنزالز[2] اولین الگوریتم تقریبی برای مسئله‌ی kـ مرکز را ارائه داده است [۸]. این الگوریتم یک الگوریتم تقریبی با ضریب ۲ است و در زمان $O(kn)$ قابل اجراست. الگوریتم گنزالز، یک الگوریتم حریصانه[3] است. روش اجرای این الگوریتم به این گونه است که در ابتدا یک نقطه‌ی دلخواه را به عنوان مرکز دسته‌ی اول در نظر می‌گیرد. سپس دورترین نقطه از آن را به عنوان مرکز دسته‌ی دوم در نظر می‌گیرد. در هر مرحله، دورترین نقطه از مرکز مجموعه دسته‌های موجود را به عنوان مرکز دسته‌ی جدید به مجموعه مراکز دسته‌ها اضافه می‌شود. با اجرای الگوریتم تا k مرحله، مراکز دسته‌ها انتخاب می‌شود. حال اگر هر نقطه را به نزدیک‌ترین مرکز انتخابی تخصیص دهیم، می‌توان نشان داد که شعاع بزرگ‌ترین

[1] Euclidean space
[2] Gonzalez
[3] greedy

دسته، حداکثر دو برابر شعاع بهینه برای مسئله‌ی kـ مرکز است. فدر[۴] و سایرین، زمان اجرای الگوریتم گنزالز را به $\mathcal{O}(n \log k)$ برای هر L_pـ متریک بهبود بخشیدند. نمونه‌ای از اجرای الگوریتم گنزالز، در شکل ۳ـ۲ نشان داده شده است.



شکل ۳ـ۲: نمونه‌ای از حل مسئله‌ی ۳ـ مرکز با الگوریتم گنزالز

تا به اینجا تنها برای k و d دلخواه صحبت شد. برای حالاتی که در مسئله‌ی k ـ مرکز، k تعداد دسته‌ها و d ابعاد فضا ثابت باشند، آگاروال[۵] و سایرین الگوریتمی دقیق با زمان اجرای $n^{\mathcal{O}(k^{1-\frac{1}{d}})}$ برای مسئله kـ مرکز در فضای L_pـ متریک ارائه داده‌اند [۹]. قابل توجه است که اگر d ثابت نباشد، مسئله‌ی kـ مرکز حتی برای متریک اقلیدسی(L_2ـ متریک) با تعداد دسته‌ی ثابت $k \geqslant 2$، ان‌پی‌ـ سخت است [۱۰].

تا به اینجا، نتایج کلی برای k و d را مورد بررسی قرار دادیم. برای بعضی از مقادیر خاص از k و d الگوریتم‌های بهینه‌تری وجود دارد. به طور مثال، برای مسئله‌ی ۱ ـ مرکز در فضای اقلیدسی با ابعاد ثابت، الگوریتم خطی با زمان اجرای $\mathcal{O}((d+1)!n)$ وجود دارد [۱۱]. الگوریتم ارائه شده بر پایه‌ی دو نکته‌ی اساسی بنا شده است. اول اینکه دایره‌ی بهینه را می‌توان با حداکثر $d+1$ نقطه‌ی واقع در پوسته‌ی کره‌ی بهینه مشخص نمود و دوم اینکه اگر نقاط ورودی را با ترتیبی تصادفی پیمایش کنیم احتمال اینکه نقطه‌ی پیمایش شده جزء نقاط مرزی باشد $\mathcal{O}(\frac{d}{n})$ است که با توجه به ثابت بودن d این احتمال کوچک می‌باشد.

در صفحه اقلیدسی(L_2ـ متریک) برای مسئله‌ی ۲ ـ مرکز، بهترین الگوریتم را چن[۶] با زمان اجرای

[۴] Feder
[۵] Agarwal
[۶] Chan

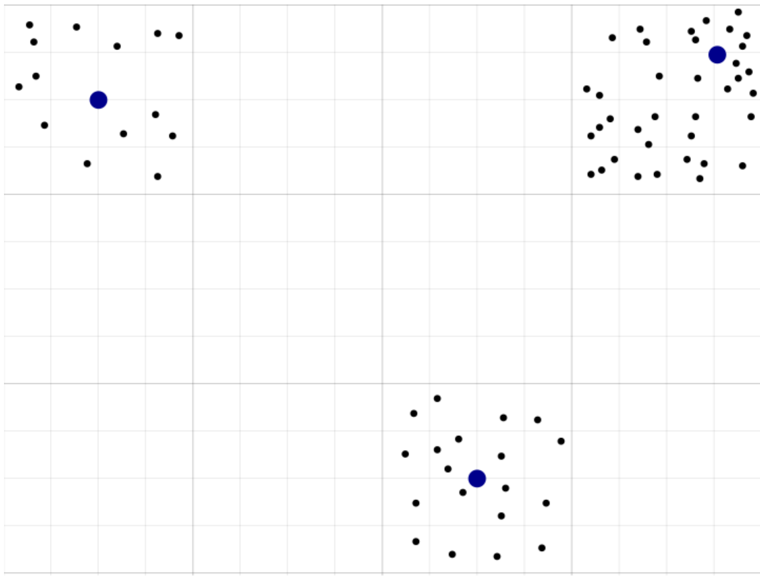
$\mathcal{O}(c \log^2 n \log^2 \log n)$ و حافظه‌ی $\mathcal{O}(n)$ ارائه داده است [۱۲]. برای فضای سه‌بعدی اقلیدسی نیز آگاروال و سایرین، الگوریتمی با متوسط زمان اجرای $\mathcal{O}(n^3 \log^8 n)$ ارائه داده است [۱۳].

۳ـ۲ kـ مرکز در حالت جویبار داده

در مدل جویبار داده، مشکل اصلی عدم امکان نگه‌داری تمام داده‌ها در حافظه است و باید سعی شود تنها داده‌هایی که ممکن است در ادامه مورد نیاز باشد را، نگه‌داریم. یکی از راه‌های رایج برای این کار نگه‌داری مجموعه‌ای از نقاط (نه لزوما زیرمجموعه‌ای از نقاط ورودی) به عنوان نماینده‌ی نقاط به طوری که جواب مسئله‌ی kـ مرکز برای آن‌ها منطبق با جواب مسئله‌ی kـ مرکز برای نقاط اصلی باشد (با تقریب قابل قبولی). به چنین مجموعه‌ای مجموعه‌ی هسته‌ی نقاط گفته می‌شود.

بهترین مجموعه هسته‌ای که برای مسئله‌ی kـ مرکز ارائه شده است، روش ارائه‌شده به وسیله‌ی ضرابی‌زاده برای نگه‌داری یک ϵـ هسته با حافظه‌ی $\mathcal{O}(\frac{k}{\epsilon^d})$ برای L_pـ متریک‌ها ارائه داده است [۱۴]. در روش ارائه شده، از چند ایده‌ی ترکیبی استفاده شده است. در ابتدا، الگوریتم با استفاده یک الگوریتم تقریبی یک تقریب از جواب بهینه به دست می‌آورد. به طور مثال با استفاده از الگوریتم گنزالز، یک دو تقریب از شعاع بهینه به علاوه‌ی مرکز دسته‌های پیدا شده را به ما بدهد. حال کافی است که با طول شعاع الگوریتم دو تقریب حول هر مرکز، یک توری با شبکه‌بندی $\mathcal{O}(\frac{1}{\epsilon})$ در هر بعد تشکیل دهیم و چون هر نقطه در حداقل یکی از توری‌ها قرار می‌گیرد، می‌توانیم با حداکثر ϵ تقریب در جواب نهایی نقاط را به نقاط شبکه‌بندی توری گرد نمود. با این کار، دیگر ما نیازی به نگهداری تمام نقاط ورودی نداشته و تنها نیاز به نگهداری نقاط شبکه‌بندی توری داریم. با این روش می‌توان به یک ϵـ هسته برای مسئله‌ی kـ مرکز رسید. نکته‌ی اساسی برای سازگار سازی روش ارائه‌شده با مدل جویبار داده‌ی تک‌گذره استفاده از روش دوبرارسازی رایج در الگوریتم‌های جویبار داده است. نمونه‌ای از اجرای الگوریتم ضرابی‌زاده را در شکل ۳ـ۳ نشان داده شده است. برای دیدن اثبات‌ها و توضیح بیش‌تر در مورد روش ارائه شده می‌توانید به مرجع [۱۴] مراجعه کنید.

از جمله مشکلات وارده به الگوریتم ضرابی زاده، وابستگی سایز مجموعه‌ی هسته به ابعاد فضا است. بنابراین هسته‌ی ارائه شده به وسیله‌ی ضرابی‌زاده را نمی‌توان برای ابعاد بالا مورد استفاده قرار داد. از طرفی، حساب کردن جواب از روی هسته در زمان چندجمله‌ای قابل انجام نیست. با توجه با موارد گفته شده، برای قابل استفاده شدن الگوریتم‌ها برای ابعاد بالا ، الگوریتم‌هایی ارائه می‌شود که ضریب تقریب



شکل ۳ـ۳: نمونه‌ای از توری‌بندی الگوریتم ضرابی‌زاده (نقاط آبی، مراکز به دست آمده از الگوریتم تقریبی است). پس از توری‌بندی کافی است برای هر کدام از خانه‌های شبکه‌بندی، تنها یک نقطه را در نظر بگیریم.

بدتری دارند، اما میزان حافظه‌ی مصرفی و سایز مجموعه هسته‌ی آن‌ها چندجمله‌ای بر اساس d و k و $\log n$ باشد. به چنین الگوریتم‌هایی الگوریتم‌های جویبار داده برای ابعاد بالا گفته می‌شود.

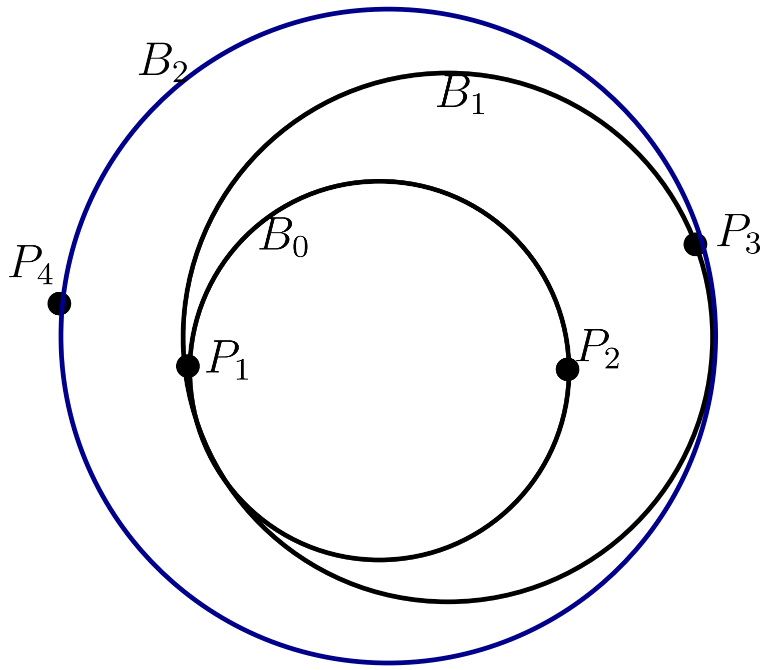
اولین الگوریتم ارائه شده برای ابعاد بالا، الگوریتمی با ضریب تقریب ۸ و حافظه‌ی مصرفی $\mathcal{O}(dk)$ است [۱۵]. پس از آن، گوها[۷]، به طور موازی با مک‌کاتن[۸] و سایرین، الگوریتمی با ضریب تقریب $(۲+\epsilon)$ با حافظه‌ی $\mathcal{O}(\frac{dk}{\epsilon}\log\frac{1}{\epsilon})$ برای مسئله‌ی kـ مرکز در هر فضای متریکی ارائه دادند[۱۶، ۱۷]. در سال ۲۰۱۴، اهن[۹] و سایرین، الگوریتمی با همین ضریب تقریب و حافظه‌ی $\mathcal{O}((k+۳)!۲^k\frac{d}{\epsilon})$ ارائه داده‌اند که برای kهای ثابت، حافظه را از مرتبه‌ی $\mathcal{O}(\log\frac{1}{\epsilon})$ کاهش می‌دهد[۱۸].

تا به اینجا ما به بررسی مسئله‌ی kـ مرکز در حالت جویبار داده بدون محدودیت خاصی پرداختیم. برای حالت‌های خاص k و متریک اقلیدسی، به خصوص ۱ و ۲، مسئله‌ی kـ مرکز مورد توجه زیادی قرار گرفته است و راه‌حل‌های بهینه‌تری نسبت به حالت کلی برای آن‌ها پیشنهاد شده است. به طور مثال،

[۷] Guha
[۸] McCutchen
[۹] Ahn

می‌توان یک هسته با اندازه‌ی $\mathcal{O}(\frac{1}{\epsilon^{\frac{d-1}{2}}})$، با استفاده از نقاط حدی[۱۰] در تعداد مناسبی جهت، به صورت جویبار داده ساخت.

مشکل عمده‌ی مجموعه هسته‌ی ارائه‌شده، وابستگی حافظه‌ی مصرفی آن به d است. ضرابی‌زاده و سایرین[۱۹] برای ابعاد دلخواه و متریک اقلیدسی، الگوریتمی با ضریب تقریب ۱/۵ و حافظه‌ی مصرفی $\mathcal{O}(d)$ ارائه دادند. در واقع در الگوریتم آن‌ها، در هر لحظه تنها یک مرکز و یک شعاع را نگه می‌دارد، که کم‌ترین حافظه‌ی ممکن برای مسئله‌ی ۱ ـ مرکز است. الگوریتم ارائه شده نقطه‌ی اول را به عنوان مرکز با شعاع صفر در نظر می‌گیرد. با فرا رسیدن هر نقطه‌ی جدید، اگر نقطه‌ی مد نظر، داخل دایره‌ی فعلی بیفتد که بدون هیچ تغییری ادامه می‌دهیم و در صورتی که بیرون دایره فعلی بیفتد، آن را با کوچک‌ترین دایره‌ای که نقطه‌ی جدید به علاوه‌ی دایره‌ی قبلی را به طور کامل می‌پوشاند، جایگزین می‌کنیم. به وضوح در هر لحظه دایره‌ی ساخته شده تمام نقاط را می‌پوشاند. از طرفی ثابت می‌شود شعاع دایره در هر لحظه، حداکثر ۱/۵ ـ برابر شعاع دایره‌ی ۱ ـ مرکز بهینه است. نمونه‌ای از اجرای الگوریتم را برروی چهار نقطه می‌توان در شکل ۳ـ۴ دید. برای اثبات کامل‌تر می‌توانید به مرجع [۱۹] مراجعه کنید.



شکل ۳ـ۴: نمونه‌ای از اجرای الگوریتم ضرابی‌زاده بر روی چهار نقطه $P_1 \cdots P_4$ که به ترتیب اندیس در جویبار داده فرا می‌رسند و دایره‌های $B_0 \cdots B_2$ دایره‌هایی که الگوریتم به ترتیب نگه می‌دارد.

در ادامه، آگاروال و سایرین[۲۰] الگوریتمی تقریبی با حافظه‌ی مصرفی $\mathcal{O}(d)$ ارائه دادند. در

[۱۰] extreme points

الگوریتم ارائه شده، ضریب تقریب، برابر $\frac{1+\sqrt{3}}{2}$ تخمین زده شد، اما با تحلیل دقیق‌تری که چن و سایرین [۲۱] انجام دادند، مشخص شد، همان الگوریتم دارای ضریب تقریب ۱/۲۲ است.

الگوریتم آن‌ها از الگوریتم کلارکسون و سایرین[۲۲] استفاده می‌کند. الگوریتم کلارکسون، الگوریتمی کاملا مشابه الگوریتم گنزالز است و به این گونه عمل می‌کند که در ابتدا یک نقطه دلخواه را به عنوان نقطه‌ی اول انتخاب می‌کند، سپس دورترین نقطه از نقطه‌ی اول را به عنوان دومین نقطه انتخاب می‌کند. ازین به بعد در هر مرحله، نقطه‌ای که از نقاط انتخاب شده‌ی قبلی بیش‌ترین فاصله را دارد به عنوان نقطه‌ی جدید انتخاب می‌کند. اگر این الگوریتم را تا $O(\frac{1}{\epsilon})$ مرحله ادامه بدهیم، به مجموعه‌ای با اندازه‌ی $O(\frac{1}{\epsilon})$ خواهیم رسید که کلارکسون و سایرین اثبات کرده‌اند که یک ϵـ هسته برای مسئله‌ی ۱ ـ مرکز است.

الگوریتم آگاروال به این گونه عمل می‌کند که اولین نقطه‌ی جویبار داده را به عنوان تنها نقطه‌ی مجموعه‌ی K_1 در نظر می‌گیرد. حال تا وقتی که نقاطی که فرا می‌رسند داخل $(1+\epsilon)Meb(K_1)$ قرار بگیرند، ادامه می‌دهد. اولین نقطه‌ای که در شرایط ذکر شده صدق نمی‌کند را p_2 بنامید. حال الگوریتم کلارکسون را بر روی $K_1 \cup \{p_2\}$ اجرا کرده و مجموع هسته‌ی به دست آمده را K_2 بنامید. با ادامه‌ی این روند، الگوریتم، دنباله‌ای از مجموعه هسته $\kappa = \{K_1, \cdots, K_u\}$ نگه می‌دارد و زمانی که نقطه‌ی p_{u+1} پیدا می‌شود که در هیچ کدام از $(1+\epsilon)Meb(K_j)$ به ازای $1 \leqslant j \leqslant u$ نباشد، الگوریتم کلارکسون را برای $\cup_{j=1}^{u} K_j \cup \{p_{u+1}\}$ اجرا نموده و مجموعه هسته‌ی به دست آمده را K_{u+1} بنامید. با توجه به نحوه‌ی ساخته شدن K_i ها، به راحتی می‌توان نشان داد رابطه‌ی زیر برقرار است:

$$P \subset \cup_{i=1}^{u} (1+\epsilon)Meb(K_i)$$

حال در نهایت برای به دست آوردن جواب نهایی کافی است کوچک‌ترین دایره‌ای که $\cup_{i=1}^{u}(1+\epsilon)Meb(K_i)$ را می‌پوشاند را به عنوان جواب دهیم. آگاروال و سایرین ثابت کرده‌اند که دایره‌ی نهایی دارای شعاع حداکثر ۱/۲۲ برابر شعاع بهینه است. برای مشاهده جزئیات بیش‌تر، به [۲۰] مراجع کنید.

آگاروال نه تنها الگوریتمی ارائه داد که در نهایت، ثابت شد حداکثر جوابی با ضریب تقریب ۱/۲۲ برابر جواب بهینه می‌دهد، بلکه نشان داد، که با حافظه‌ی چندجمله‌ای بر اساس $\log n$ و d نمی‌توان الگوریتمی ارائه داد که ضریب تقریب بهتر از $\frac{1+\sqrt{2}}{2}$ داشته باشد.

قضیه‌ی ۳ـ۱ *هر الگوریتم تحت مدل جویبار داده که یک αـ تقریب برای مسئله‌ی ۱ ـ مرکز برای*

مجموعه‌ی S شامل n نقطه در فضای $\mathbb{R}^d$ نگه می‌دارد، برای $\alpha \leqslant \frac{1+\sqrt{2}}{2}(1-\frac{2}{d^{\frac{1}{3}}})$ با احتمال حداقل $\frac{2}{3}$ نیاز به $\Omega(\min\{n, e^{d^{\frac{1}{3}}}\})$ حافظه مصرف می‌کند.

اثبات. *ایده‌ی اصلی اثبات بر اساس قضیه‌ی معروف آلیس و باب[11] در نظریه انتقال اطلاعات بنا شده است. برای خواندن اثبات این قضیه می‌توانید به مرجع [۲۰] مراجعه کنید.*

□

علاوه بر مسئله‌ی ۱ـ مرکز، مسئله‌ی دو مرکز نیز در سال‌های اخیر مورد توجه قرار گرفته است و بهبودهایی نیز برای آن مسئله ارائه شده است. آهن و سایرین [۲۳] در سال ۲۰۱۴، اولین الگوریم با ضریب تقریب کم‌تر از دو را برای مسئله‌ی ۲ـ مرکز در فضای اقلیدسی ارائه دادند. این الگوریتم تقریبا پایه‌ی کار این پایان‌نامه برای حالت‌های مختلف است. به همین منظور، تعدادی از لم‌های داخل این الگوریتم که در آینده استفاده می‌شود این‌جا توضیح داده می‌شود.

لم۳ـ۲ *فرض کنید B یک کره‌ی واحد با مرکز c در فضای اقلیدسی $\mathbb{R}^d$ باشد. هر پاره‌خط pq به طول حداقل ۱/۲ که به طور کامل داخل B قرار دارد، دایره‌ی $B'(c, ۰/۸)$ را قطع می‌کند.*

اثبات. *صفحه‌ی گذرنده از پاره‌خط و مرکز دایره را نظر بگیرید. ادامه اثبات تنها به به همین صفحه محدود می‌شود، بنابراین نیاز به در نظر گرفتن ابعاد بزرگ‌تر از ۲ نیست. همان‌طور که در شکل ۳ـ۵ مشخص شده است، پای عمود از مرکز دایره بر پاره‌خط pq را h بنامید. بدون کم شدن از کلیت مسئله فرض می‌کنیم $\|hq\| \leqslant \|hp\|$. بنابراین داریم:*

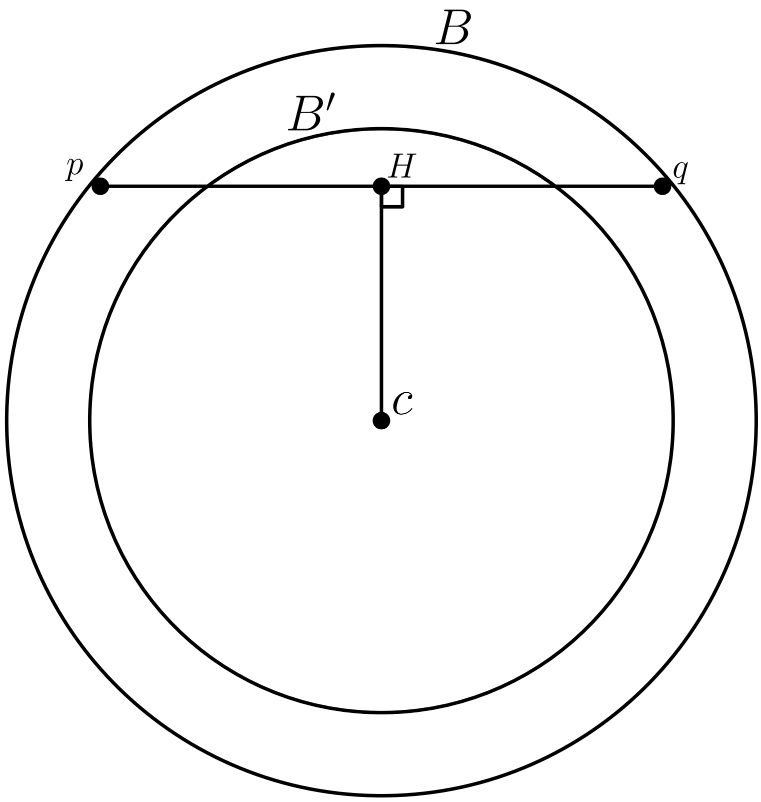
$$۰/۶ = \frac{۱/۲}{۲} \leqslant \|hq\|$$

از طرفی چون پاره‌خط pq به طور کامل داخل کره‌ی واحد قرار گرفته است، بنابراین تمام نقاط آن، شامل دو سر آن، از مرکز دایره، فاصله‌ی حداکثر ۱ دارند. بنابراین طبق رابطه‌ی فیثاغورث، داریم:

$$\|hc\| = \sqrt{\|qc\|^{۲} - \|qh\|} \leqslant \sqrt{۱ - ۰/۶^{۲}} = ۰/۸$$

بنابراین نقطه‌ی h داخل دایره‌ی B' قرار می‌گیرد. از طرفی چون، $\|pq\| \leqslant ۱$ بنابراین، h داخل پاره‌خط قرار دارد و در نتیجه پاره‌خط B' با پاره‌خط pq تقاطع دارد.

[11] alice and bob



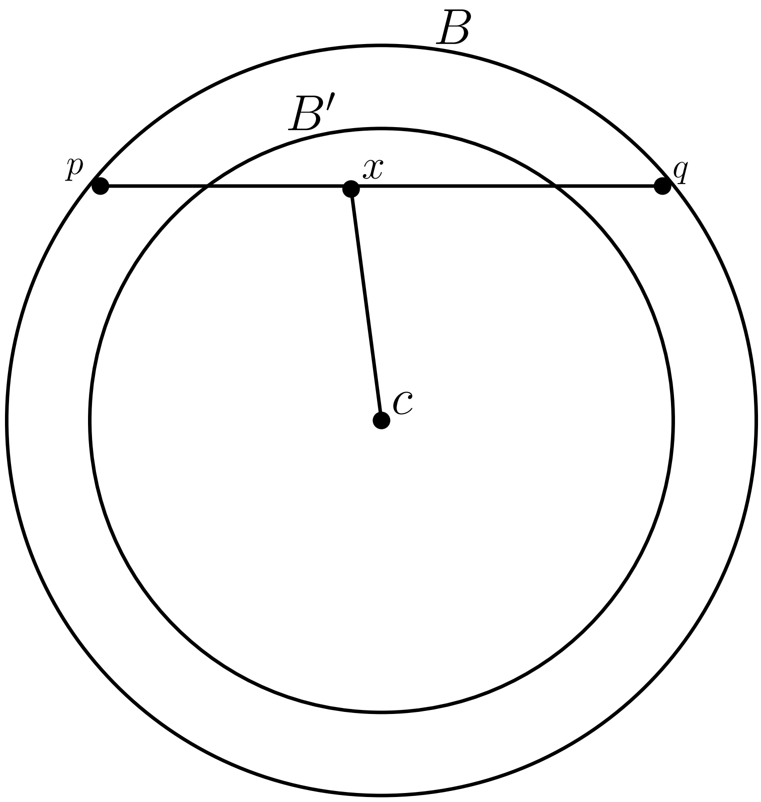
شکل ۳ـ۵: اثبات لم ۳ـ۲

□

لم بالا در واقع نشان می‌دهد اگر در طول الگوریتم بتوانیم دو نقطه‌ی دور نسبت به هم (حداقل ۱/۲ برابر شعاع بهینه) از یکی از دو دایره‌ی بهینه را بیابیم، این پاره‌خط از مرکز دایره‌ی بهینه فاصله‌ی کمی (حداکثر ۰/۸ شعاع بهینه) را دارد.

لم۳ـ۳ *فرض کنید B دایره‌ای به مرکز c و شعاع واحد در $\mathbb{R}^d$ باشد. پاره‌خط دلخواه pq با طول حداقل ۱/۲ که به طور کامل داخل B قرار دارد را در نظر بگیرید.هر نقطه‌ی x از پاره‌خط pq که از دو سر آن حداقل ۰/۶ فاصله داشته باشد، داخل دایره‌ی $B'(c, ۰/۸)$ قرار می‌گیرد.*

اثبات. *اثبات این لم نیز کاملا مشابه لم ۳ـ۲ است. بدون کم شدن از کلیت مسئله همان‌طور که در شکل ۳ـ۶ مشخص شده است، فرض کنید زاویه‌ی $\angle pxc$ بزرگ‌تر مساوی ۹۰ درجه است. در نتیجه*



شکل ۳ـ۶: اثبات لم ۳ـ۳

داریم:

$$\sqrt{\|px\|^{۲} + \|xc\|^{۲}} \leqslant \|pc\| \leqslant ۱$$

از طرفی طبق فرض مسئله داریم:

$$۰/۶ \leqslant \|px\| \implies \|xc\| \leqslant \sqrt{۱ - ۰/۶^{۲}} = ۰/۸$$

□

الگوریتم آهن، با استفاده از دو لم بالا و تقسیم مسئله به دو حالتی که دو دایره‌ی بهینه بیش از $۲r^*$ *یا کم‌تر از* $۲r^*$ *فاصله داشته باشد، دو الگوریتم کاملا جدا ارائه می‌دهد که به طور موازی اجرا می‌گردند. این دو الگوریتم، به طور موازی اجرا می‌شوند و در هر لحظه جوابی درست را به عنوان جواب نهایی ارائه می‌دهند و کافی است برای جواب نهایی بین دو شعاعی که به عنوان جواب خود می‌دهند، شعاع کم‌تر را به عنوان جواب الگوریتم بدهیم. به علت تشابه این الگوریتم، با یکی از الگوریتم‌های فصل کارهای جدید، از تکرار آن خودداری کرده و به تفصیل در فصل نتایج جدید شرح داده می‌شود.*

۳ـ۳ kـ مرکز با داده‌های پرت

فصل ۴

نتایج جدید

در این فصل نتایج جدید به‌دست‌آمده در پایان‌نامه توضیح داده می‌شود. در صورت نیاز می‌توان نتایج جدید را در قالب چند فصل ارائه نمود. همچنین در صورت وجود پیاده‌سازی، بهتر است نتایج پیاده‌سازی را در فصل مستقلی پس از این فصل قرار داد.

فصل ۵

نتیجه‌گیری

در این فصل، ضمن جمع‌بندی نتایج جدید ارائه‌شده در پایان‌نامه، مسائل باز باقی‌مانده و همچنین پیش‌نهادهایی برای ادامه‌ی کار ارائه می‌شوند.

پیوست آ

مطالب تکمیلی

پیوست‌های خود را در صورت وجود می‌توانید در این قسمت قرار دهید.

کتاب‌نامه

[1] *J. Han and M. Kamber.* Data Mining, Southeast Asia Edition: Concepts and Techniques*. Morgan kaufmann, 2006.*

[2] *C. C. Aggarwal.* Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007.*

[3] *M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In* Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 642–651, 2001.*

[4] *M. Sipser.* Introduction to the Theory of Computation*. Cengage Learning, 2012.*

[5] *R. G. Michael and S. J. David. Computers and intractability: a guide to the theory of np-completeness.* WH Freeman & Co., San Francisco*, 1979.*

[6] *N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems.* SIAM Journal on Computing*, 13(1):182–196, 1984.*

[7] *M. Bern and D. Eppstein. Approximation algorithms for geometric problems.* Approximation algorithms for NP-hard problems*, pages 296–345, 1996.*

[8] *J. Han, M. Kamber, and J. Pei.* Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.*

[9] *P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering.* Algorithmica*, 33(2):201–226, 2002.*

[10] *N. Megiddo. On the complexity of some geometric problems in unbounded dimension.* Journal of Symbolic Computation*, 10(3):327–334, 1990.*

[11] *B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension.* Journal of Algorithms*, 21(3):579–597, 1996.*

[12] *T. M. Chan. More planar two-center algorithms.* Computational Geometry: Theory and Applications*, 13(3):189–198, 1999.*

[13] *P. K. Agarwal, R. B. Avraham, and M. Sharir. The 2-center problem in three dimensions.* Computational Geometry*, 46(6):734–746, 2013.*

[14] *H. Zarrabi-Zadeh. Core-preserving algorithms. In* Proceedings of the 20th Canadian Conference on Computational Geometry*, pages 159–162, 2008.*

[15] *M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. In* Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 626–635. ACM, 1997.*

[16] *R. M. McCutchen and S. Khuller. Streaming algorithms for k-center clustering with outliers and with anonymity. In* Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 165–178. 2008.*

[17] *S. Guha. Tight results for clustering and summarizing data streams. In* Proceedings of the 12th International Conference on Database Theory*, pages 268–275, 2009.*

[18] *H.-K. Ahn, H.-S. Kim, S.-S. Kim, and W. Son. Computing k centers over streaming data for small k.* International Journal of Computational Geometry and Applications*, 24(02):107–123, 2014.*

[19] *H. Zarrabi-Zadeh and T. M. Chan. A simple streaming algorithm for minimum enclosing balls. In* Proceedings of the 18th Canadian Conference on Computational Geometry*, pages 139–142, 2006.*

[20] *P. K. Agarwal and R. Sharathkumar. Streaming algorithms for extent problems in high dimensions. In* Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms*, pages 1481–1489, 2010.*

[21] *T. M. Chan and V. Pathak. Streaming and dynamic algorithms for minimum enclosing balls in high dimensions.* Computational Geometry: Theory and Applications*, 47(2):240–247, 2014.*

[22] *M. Bâdoiu and K. L. Clarkson. Smaller core-sets for balls. In* Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 801–802. Society for Industrial and Applied Mathematics, 2003.*

[23] *S.-S. Kim and H.-K. Ahn. An improved data stream algorithm for clustering. In* Proceedings of the 11th*, pages 273–284. 2014.*

واژه‌نامه

الف

ابتکاری .... *heuristic*

ارزش .... *worth*

Abstract

This part should be completed.

Keywords: *Clustering, K-Center, Streaming Data, Approximation Algorithm*

Sharif University of Technology

Department of Computer Engineering

M.Sc. Thesis

Approximation Algorithms for Clustering Points in the Data Stream Model

By:

Behnam Hatami-Varzaneh

Supervisor:

Dr. Hamid Zarrabi-zade

September 2015