دانشگاه فردوسی مشهد

پایان نامه

استفاده از بهینه سازی پیوسته در حل رده ای از مسایل افرازبندی گراف

نگارنده

احمد ابویی مهریزی

تابستان۱۳۹۴

فهرست مطالب

فهرست جدول‌ها

فهرست شکل‌ها

فهرست نمودارها

پیش‌گفتار

از اوایل سال ۱۹۷۰ مساله افرازبندی گراف به طور گسترده مورد تحقیق و بررسی قرار گرفت. در حال حاضر بسیاری از مسایل در علوم و مهندسی را می‌توان به صورت مساله افرازبندی گراف فرمول‌بندی کرد.

در حالت ساده اگر $G = (V, E)$ گرافی بدون جهت و وزن‌دار باشد هدف مساله افرازبندی گراف، افراز کردن راس‌های گراف به دو مجموعه جدا از هم است. به طوری که مجموع وزن یال‌های بین دو مجموعه کمینه شود (جهت‌دار بودن و وزن‌دار بودن راس‌های گراف، همچنین تعداد مجموعه‌های افراز (بیش‌تر از دو مجموعه) و... هر کدام نوع متفاوتی از مساله افرازبندی گراف را ایجاد می‌کند [۲۷، ۳۲]).

با وجود این‌که مساله افرازبندی گراف در رده‌ی مسایل NP-سخت قرار می‌گیرد. الگوریتم‌های دقیق و الگوریتم‌های ابتکاری دو رده کلی برای حل مساله افرازبندی گراف هستند. بر همین اساس این پایان نامه در چهار فصل به صورت زیر تدوین شده است.

در فصل اول، به بیان تاریخچه، معرفی مساله افرازبندی گراف، بیان کاربردها و بررسی مختصر چند روش حل که از بهینه‌سازی پیوسته برای حل مساله افرازبندی گراف، در دو رده الگوریتم‌های دقیق و ابتکاری استفاده می‌کند، خواهیم پرداخت.

در فصل دوم، به بررسی روش تقریبی کرویت[۱] برای حل مساله افرازبندی گراف با استفاده از بهینه‌سازی پیوسته نامقید و به کار بردن الگوریتم گرادیان مزدوج برای حل آن، خواهیم پرداخت. در ادامه این فصل، روش کرویت را با رده‌ی دیگری از الگوریتم گرادیان مزدوج برای حل مساله افرازبندی گراف مورد بررسی قرار خواهیم داد و نتایج

[۱] Kruyt

آن را با روش کرویت مقایسه می‌کنیم.

در فصل سوم، روش هاگر[۱] و همکاران را که با استفاده از الگوریتم شاخه و کران، جواب بهینه مساله افرازبندی گراف را به دست می‌آورند، به تفصیل شرح خواهیم داد. هاگر و همکاران در این روش با تبدیل مساله افرازبندی گراف به مساله بهینه‌سازی پیوسته درجه دوم، با استفاده از شرایط لازم مرتبه اول برای مسایل بهینه‌سازی مقید (شرایط $K.K.T$) مساله افرازبندی گراف را حل می‌کنند.

در فصل چهارم، حالت دیگری از مساله افرازبندی گراف را مورد بررسی قرار خواهیم داد. در این حالت راس‌های گراف G نیز وزن‌دار هستند. در ادامه فصل چهارم با ایده گرفتن از روش هاگر و همکاران شرطی را روی مساله بهینه‌سازی پیوسته درجه دوم اعمال می‌کنیم تا مجموع وزن یال‌های بین دو مجموعه افزار و اختلاف بین وزن هر کدام از مجموعه‌های افراز به حداقل مقدار خود برسد.

[۱] Hager

فصل ۱

تعریف مساله و کاربرد آن

۱.۱ تاریخچه

نظریه گراف[۱] بر خلاف شاخه‌های دیگر ریاضیات نقطه آغاز مشخصی دارد. در سال ۱۷۳۶ زمانی که اویلر[۲] مساله پل‌های کونیگزبرگ[۳] [۱۸] را در مکان و شرایط واقعی مطرح می‌کند، نظریه گراف به وجود آمده است.

در راستای افرازبندی گراف می‌توان به مساله رنگ‌آمیزی گراف به عنوان یکی از زیر مساله‌های افرازبندی گراف اشاره کرد که در سال ۱۸۵۰ توسط فرانسیس گادری[۴] مورد مطالعه و بررسی قرار می‌گیرد [۴۳].

اما از اوایل سال ۱۹۷۰ مساله افرازبندی گراف[۵] به طور گسترده مورد تحقیق و بررسی قرار گرفته است [۸]. در همین سال، یک روش ابتکاری برای حل مساله افرازبندی گراف توسط کرنیگهان[۶] ارایه شد [۳۱].

۲.۱ مساله افرازبندی گراف

تعریف ۱.۲.۱. *[۸] فرض کنید V یک مجموعه دلخواه باشد. در این صورت، P، مجموعه‌ای از زیرمجموعه‌های V را یک افراز برای V می‌نامیم، هرگاه*

۱. *مجموعه P ناتهی باشد و شامل تهی نباشد.*

۲. *اعضای P دو به دو مجزا باشند. به عبارتی، برای هر $V_1, V_2 \in P$ رابطه $V_1 \cap V_2 = \emptyset$ برقرار باشد.*

۳. *اجتماع اعضای P برابر با V شود.*

تعریف ۲.۲.۱. *[۸] فرض کنید V یک مجموعه ناتهی با n_V عضو و E مجموعه‌ای با n_E زوج مرتب از اعضای V باشد. زوج مرتب (V, E) را گراف G می‌گوییم که در آن اعضای V راس‌های گراف و اعضای E یال‌های گراف هستند.*

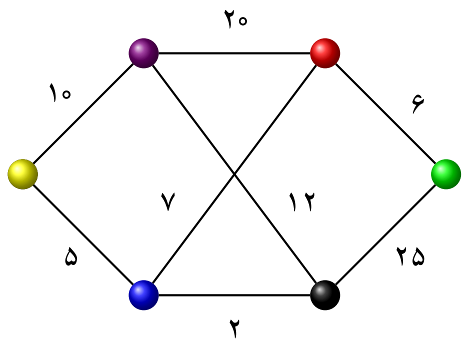
---

[۱] Graph theory [۲] Leonhard Euler [۳] Bridges of Konigsberg [۴] Francis Guthrie [۵] Graph partitioning

[۶] Kernighan

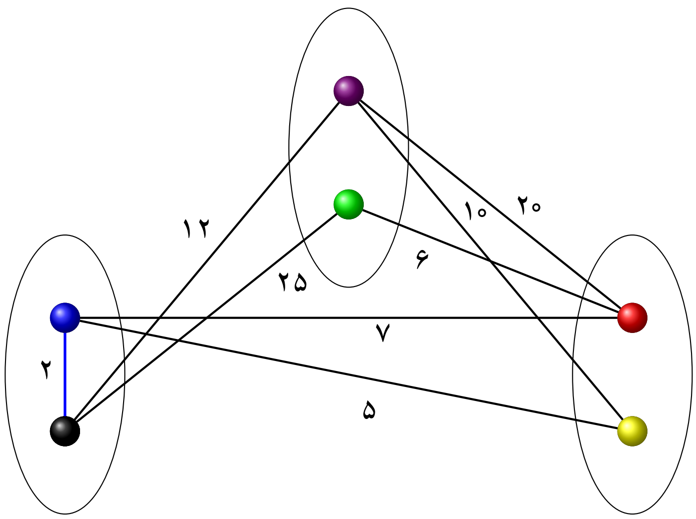
تعریف ۳.۲.۱. *[۸] فرض کنید $G=(V,E)$ یک گراف و $P_k=\{V_۱,\cdots,V_k\}$ مجموعه‌ای متشکل از k زیرمجموعه V باشد. در این صورت، گوییم P_k یک افراز گراف G است، هرگاه P_k یک افراز برای V مطابق با تعریف ۱.۲.۱ باشد.*

مثال ۴.۲.۱. *گراف G را به شکل زیر در نظر بگیرید:*



شکل ۱.۱: گراف بدون جهت G با شش راس

می‌خواهیم راس‌های گراف G را افراز کنیم به طوری که تعداد مجموعه‌های افراز برابر با ۳ باشد، همچنین اندازه هر یک از مجموعه‌های افراز مقداری معین بوده و برابر با ۲ باشد. بر این اساس، یک نمونه از افراز گراف G با ویژگی‌های ذکر شده، در شکل ۲.۱ نشان داده شده است.



شکل ۲.۱: یک افراز از راس‌های گراف G

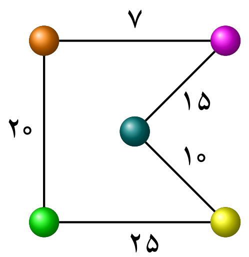
تعریف ۵.۲.۱. *[۲۲] در شکل ۲.۱ به یال‌هایی‌که با رنگ مشکی نشان داده شده‌اند، یال‌های برشی[۱] ، (یال‌هایی که یک سر آن‌ها در یک مجموعه افراز و انتهای دیگر آن در مجموعه افراز دیگری قرار دارد) و به یال‌هایی‌که با رنگ آبی نشان داده شده‌اند، یال‌های غیر برشی[۲] (یال‌هایی‌که هر دو سر آن‌ها در یک مجموعه افراز قرار دارد) گفته می‌شود.*

تعریف ۶.۲.۱. *[۸] فرض کنید $G = (V, E)$ گرافی با یال‌های وزن‌دار باشد. هدف مساله k-افرازبندی گراف، افرازکردن راس‌های گراف به k مجموعه با اندازه معین است به طوری‌که مجموع وزن یال‌های بین مجموعه‌های افراز کمینه شود.*

ملاحظه ۷.۲.۱. *در همه الگوریتم‌های مورد بررسی در این پایان‌نامه، هدف افرازکردن راس‌ها، به دو مجموعه مجزا است ($k = ۲$).*

نکته ۸.۲.۱. *مطابق تعریف ۶.۲.۱، اگر برای حالت ($k = ۲$) الگوریتم حل داشه باشیم، برای حالتی‌که k توانی از ۲ است، می‌توان با استفاده از همین الگوریتم یک روش حل ارایه نمود [۸].*

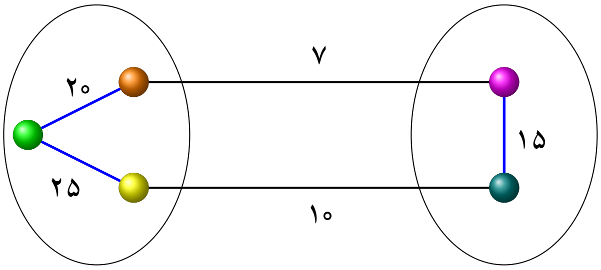
مثال ۹.۲.۱. *گراف G را به شکل زیر در نظر بگیرید:*



شکل ۳.۱: گراف G با پنج راس

[۱] Cut edges [۲] Uncut edges

حال، بر پایه تعریف ۶.۲.۱، جواب بهینه مساله افرازبندی گراف، متناظر با گراف شکل ۳.۱ به صورت زیر است.



شکل ۴.۱: افراز بهینه در گراف G

مطابق شکل ۴.۱ مجموع وزن یال‌های برشی برابر با ۱۷ است، که در بین همه افرازهای گراف G کمترین مقدار را دارد.

۳.۱ کاربردها

پس از بیان تاریخچه و معرفی مساله افرازبندی گراف، به تعدادی از کاربردهای آن اشاره می‌کنیم. بسیاری از مسایل در علوم و مهندسی می‌توانند به عنوان مسایل افرازبندی گراف فرمول‌بندی شوند. از این بین می‌توان به محاسبات موازی[1] [۲۴،۱۰]، طراحی مدارهای الکتریکی مانند VLSI [2] [۲۹]، کنترل ترافیک هوایی[3] [۷،۶] ، مرتب‌سازی محاسبات در ماتریس‌های تُنُک[4] [۳۸]، زمان‌بندی کارها[5] [۲۸] و قطعه‌بندی تصویر[6] [۴۲،۴۱،۱] اشاره کرد. چند نمونه از این کاربردها را به اختصار شرح می‌دهیم.

[1]Parallel computations [2]Very larg scale integration [3]Air traffic control [4]Sparse matrix [5]Task scheduling [6]Image segmentation

۱.۳.۱ کاربرد افرازبندی گراف در محاسبات موازی

تعریف و ویژگی‌های محاسبات موازی. به اجرای هم‌زمان یک برنامه که به بخش‌های کوچک‌تری تبدیل شده است، محاسبات موازی گفته می‌شود. ایده این کار از آن‌جا گرفته شده است که فرایند حل یک مساله را معمولاً می‌توان به زیر وظایف کوچکتری تقسیم کرد که با اجرای هم‌زمان این زیر وظایف و هماهنگ کردن آن‌ها مساله اصلی در زمان کوتاهتری حل می‌شود. یک سیستم محاسبات موازی به رایانه‌ای گفته می‌شود که بیش از یک پردازنده، برای انجام محاسبات موازی دارد (هر یک از زیر وظایف توسط یک پردازنده اجرا می‌شود). از مزایای محاسبات موازی می‌توان به کاهش زمان محاسبه، حل مشکلات مربوط به محدودیت حافظه، امکان حل مسایل بزرگتر و صرفه اقتصادی اشاره کرد.

ارتباط بین گراف و محاسبات موازی. گراف‌ها به طور گسترده برای توصیف وابستگی داده‌ها در محاسبات استفاده می‌شوند. در واقع راس‌های گراف نشان دهنده واحد محاسبه و یال‌ها نشان دهنده وابستگی میان داده‌ها هستند. وزن مربوط به راس‌ها و یال‌ها هم می‌تواند به ترتیب نشان دهنده تعداد کارها در واحد محاسبه و تعداد داده‌ها باشد (برای روشن‌تر شدن ساختار گراف، فرایند محاسبه ریشه یک معادله درجه دوم را در نظر بگیرید، در این فرایند چهار عمل اصلی را به همراه جذرگیری داریم، که هر کدام از این عمل‌ها را می‌توان به عنوان یک واحد محاسبه و در نتیجه یک راس در گراف در نظر گرفت، همچنین ارتباط بین هر یک از این عمل‌ها را در این فرایند، می‌توان به عنوان یال در گراف نشان داد).

به کار بردن مساله افرازبندی گراف در محاسبات موازی در راستای انجام محاسبات موازی می‌توان شبیه‌سازی مساله افرازبندی گراف را در نظر گرفت، به این مفهوم که هر چه ارتباط بین زیر مساله‌هایی که بر روی پردازنده‌های مختلف حل می‌شوند، کمتر باشد (وزن یال‌های بین مجموعه‌های افراز کمینه شود)، نتیجه نهایی حل مساله نیز سریع‌تر به دست می‌آید (اگر فرایند ریشه‌یابی معادله درجه دوم را در نظر بگیرید، می‌توان گراف متناظر آن را افرازبندی کرد، به طوری که هر یک از راس‌های گراف به عنوان یک زیر مساله، توسط یک پردازش‌گر

(مجموعه افراز) حل شود و ارتباط بین زیر مسالهها کمینه شود تا ریشه معادله درجه دوم سریع‌تر محاسبه شود) [۲۴].

۲.۳.۱ کاربرد افرازبندی گراف در طراحی مدارهای الکتریکی VLSI

تعریف مدارهای مجتمع مدارهای دیجیتال با مدارهای مجتمع ساخته می‌شوند. یک مدار مجتمع (IC) یک کریستال کوچک نیمه رسانا به نام تراشه است که قطعات الکترونیکی را برای دروازه‌های دیجیتال در خود دارد. اتصالات داخل تراشه، مدار مورد نیاز را به وجود می‌آورند. تراشه، در داخل یک محفظه پلاستیک و یا سرامیک جاسازی می‌شود و اتصالات آن با سیم‌های طلایی نازک به پایه‌های خارجی جوش داده می‌شود تا مدارهای مجتمع به وجود آیند.

با پیشرفت تکنولوژی مدارهای مجتمع، تعداد دروازه‌هایی که می‌تواند در یک تراشه جای گیرد به میزان قابل توجهی افزایش یافته است. بر این اساس، مدارهای مجتمع به چهار دسته تقسیم می‌شوند:

۱. مدارهای مجتمع با مقیاس کوچک (SSI[۱]) که دارای چند دروازه در هر بسته واحد هستند. تعداد دروازه‌ها معمولا کمتر از ۱۰ هستند.

۲. مدارهای مجتمع با مقیاس متوسط (MSI[۲]) تقریبا دارای ۱۰ تا ۲۰۰ دروازه در هر بسته هستند.

۳. مدارهای مجتمع با مقیاس بزرگ (LSI[۳]) این مدارها بین ۲۰۰ تا چند هزار دروازه در هر بسته دارند.

۴. مدارهای مجتمع با مقیاس بسیار بزرگ (VLSI) که حاوی هزاران دروازه در یک بسته هستند.

بررسی کارایی مدار الکتریکی VLSI. پس از ساخت یک مدار VLSI باید بررسی شود که آیا این مدار به درستی عمل می‌کند یا خیر. به همین منظور مساله‌هایی را به نام مسایل آزمون برای بررسی کارایی یک مدار تولید می‌کنند. اما، بررسی یک مدار VLSI به کمک مساله آزمون با توجه به ساختار مدار دارای محاسبات پیچیده است. از این رو مدار را بخش‌های کوچکتر تقسیم می‌کنند و پس بررسی هر بخش، ارتباط بخش‌ها با یکدیگر مورد

[۱] Small scale integration [۲] Medium scale integration [۳] Larg scale integration

بررسی قرار می‌گیرد و در نهایت نحوه عملکرد مدار مشخص می‌شود. این عمل باعث می‌شود فرایند تعیین کارایی یک مدار با سرعت بیش‌تری صورت بگیرد.

به کار بردن مساله افرازبندی گراف . در راستای طراحی و بررسی مدار ابتدا مدارهای الکتریکی را به عنوان گراف $G = (V, E)$ در نظر می‌گیرند که در آن V مجموعه راس‌ها، نشان دهنده اجزای اصلی مدار مانند دروازه‌ها، فیلپ-فلاپ‌ها[۱] (امکان ذخیره‌سازی ۱ بیت را دارند و برای طراحی سلول‌های حافظه به کار می‌روند)، ورودی و خروجی‌ها است و E مجموعه یال‌ها نشان دهنده ارتباطات موجود در شبکه است.

مساله افرازبندی گراف در طراحی مدارهای الکتریکی، یک مدار را به بخش‌های کوچک‌تری تقسیم (افراز) می‌کند به طوری که ارتباط بین این بخش‌ها کمینه شود، این عمل موجب می‌شود پس از اینکه نحوه عملکرد هر بخش با استفاده از مسایل آزمون مشخص شد، بررسی ارتباط بخش‌ها با یک‌دیگر با سرعت بیشتر و پیچیدگی کمتر انجام پذیرد [۲۹].

شکل ۱.۵: نمونه‌ای از مدار VLSI

[۱] Flip-flops

۳.۳.۱ کاربرد افرازبندی گراف در قطعه‌بندی تصویر

تعریف و کاربرد قطعه‌بندی تصویر. پردازش تصویر شامل چندین مرحله بوده که مهمترین آن‌ها قطعه‌بندی است. قطعه‌بندی، فرایندی است که تصویر را به قسمت‌های اصلی سازنده‌اش تقسیم می‌کند. به این معنی که اشیا مختلف موجود در تصویر، با توجه به کاربرد مورد نظر، از هم جدا می‌شوند تا تحلیل تصویر در مراحل بعدی راحت‌تر انجام بگیرد. برای روشن‌تر شدن فرایند قطعه‌بندی تصویر فرض کنید که از طریق هوایی به دنبال رهگیری یک وسیله نقلیه هستیم (شکل ۶.۱ را ببینید). برای این منظور، ابتدا باید جاده شناسایی شده، و سپس تشخیص وسیله نقلیه مورد نظر صورت گیرد. بر این اساس، ابتدا جاده از تصویر جدا می‌شود، سپس جاده به اجزایی تقسیم می‌شود که تقریبا به بزرگی هدف مورد نظر هستند، این عمل موجب می‌شود تا بتوان وسیله نقلیه مورد نظر را در تصویر پیدا کرد.

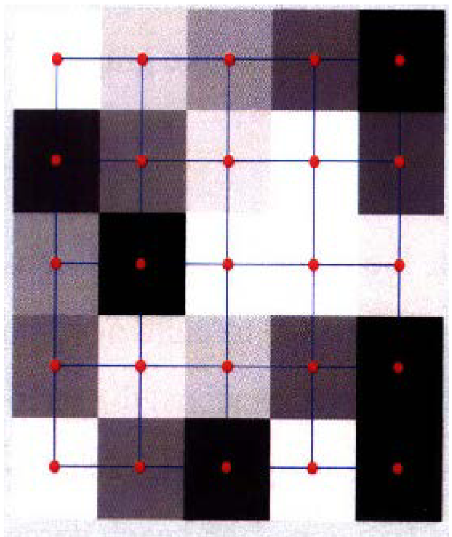
شکل ۶.۱: کاربرد قطعه‌بندی تصویر برای تشخیص اتومبیل، از طریق تصاویر هوایی

از میان کاربردهای قطعه‌بندی تصویر می‌توان به تجزیه تصاویر پزشکی برای تعیین محل تومورها، اندازه‌گیری حجم بافت و تشخیص بیماری اشاره کرد. قطعه‌بندی تصویر همچنین در تشخیص چهره و اثر انگشت و تعیین محل اشیا در تصاویر ماهواره‌ای، مانند جاده‌ها و جنگل‌ها کاربرد دارد [۱].

استفاده از گراف در قطعه‌بندی تصویر یکی از اولین روش‌های قطعه‌بندی تصویر به وسیله نظریه گراف ارایه شده است. بر این اساس، یک تصویر را می‌توان به گراف تبدیل کرد به طوری که هر پیکسل[۱] (کوچکترین عنصر غیر قابل تقسیم در تصویر) نماینده یک راس در گراف است. به عبارت دیگر، اگر یک تصویر $m \times n$ پیکسل باشد،

[۱] Pixel

در این صورت، تعداد راس‌ها در گراف متناظر با آن تصویر، برابر با $m \times n$ خواهد بود (شکل ۷.۱ را ببینید). با توجه به نحوه تعیین راس‌ها در گراف، می‌توان نتیجه گرفت تعداد راس‌ها ثابت و برابر با تعداد پیکسل‌هاست. دو راس (پیکسل) در گراف که در همسایگی یک‌دیگر هستند، به وسیله یال به یک‌دیگر متصل می‌شوند. بر خلاف راس‌ها تعداد یال‌ها در گراف می‌تواند متفاوت باشد، در یک نمونه هر راس تنها می‌تواند در جهت‌های شمال، جنوب، غرب و شرق به راس‌های مجاور خود متصل شود، به این مدل ۴- اتصال[1] گفته می‌شود (شکل ۷.۱ را ببینید).

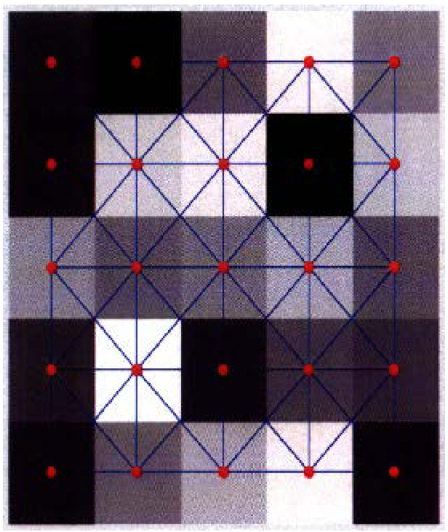


شکل ۷.۱: تصویر با مشخصات۵ × ۵ پیکسل و گراف متناظر با آن به شکل ۴- اتصال

در نمونه‌ای دیگر، هر راس می‌تواند به راس‌های مجاور خود در جهت‌های شمال، جنوب، غرب، شرق، شمال غرب، شمال شرق، جنوب غرب و جنوب شرق متصل شود، به این مدل ۸- اتصال گفته می‌شود (شکل ۸.۱ را ببینید).

مدل‌های دیگری هم برای رسم گراف وجود دارد ([۴۲] را ببینید). در مرحله بعد، یال‌ها بر اساس توابعی به نام توابع وزن[2] ، با توجه به درجه اختلاف پیکسل‌ها (راس‌های گراف)، وزن‌دار می‌شوند [۴۲].

[1] Connected [2] Weighting functions



شکل ۸.۱: گراف به شکل ۸- اتصال، متناظر با تصویر

به کار بردن مساله افرازبندی گراف بعد از پیش پردازش‌های اولیه تصویر ورودی به گراف تبدیل می‌شود، گراف به دست آمده افرازبندی شده، بر اساس افراز بهینه این گراف بر روی تصویر مدل و در نتیجه تصویر قطعه‌بندی می‌شود.

۴.۱ روش‌های حل مساله افرازبندی گراف

مساله افرازبندی گراف در رده مسایل NP-سخت[۱] [۱۷] قرار می‌گیرد. با این وجود، دو رده کلی حل برای مسایل افرازبندی گراف وجود دارد. الگوریتم‌های دقیق[۲] که با وجود NP-سخت بودن مساله می‌توانند روی مسایل با اندازه متوسط کارا باشند و الگوریتم‌های ابتکاری[۳] که سعی دارند در زمان بسیار کم یک جواب تقریبی قابل قبول پیدا کنند.

در این میان تعدادی از الگوریتم‌های دقیق و ابتکاری را به اختصار شرح خواهیم داد.

[۱] NP-hard [۲] Exact methods [۳] Heuristic methods

۱.۴.۱ الگوریتم‌های ابتکاری

الگوریتم ولکویچ و رندل

قبل از بیان خلاصه‌ای از روش ولکویچ[۱] و رندل[۲] باید توجه کرد که

ملاحظه ۱.۴.۱. *در مساله افرازبندی گراف با توجه به این‌که مجموع وزن همه یال‌ها مقداری ثابت است، می‌توان به جای کمینه کردن وزن یال‌های برشی، وزن یال‌های غیر برشی را بیشینه کرد، به همین جهت مساله افرازبندی گراف را می‌توان به عنوان یک مساله بیشینه‌سازی در نظر گرفت و در نتیجه در روش‌های تقریبی برای حل مساله افرازبندی گراف، می‌توان یک کران بالای مناسب برای مساله پیدا کرد.*

در سال‌های ۱۹۹۴ و ۱۹۹۵ ولکویچ و رندل با استفاده از بهینه‌سازی پیوسته[۳] یک روش ابتکاری برای حل مساله k-افرازبندی گراف ارایه کردند [۱۵، ۳۹]. این روش که مبتنی بر روش‌های بهینه‌سازی[۴] است، ابتدا با آزادسازی قیدها روی برنامه‌ریزی پیوسته یک کران بالای مناسب به دست می‌آورد. سپس، با استفاده از این کران بالا جواب شدنی برای مساله افرازبندی گراف به دست می‌آید. روش ابتکاری ولکویچ و رندل بر اساس مقاله‌ای از دناس[۵] و هافمن[۶] در سال ۱۹۷۳ ارایه شد [۱۴].

الگوریتم کرویت

کرویت در سال ۱۹۹۷، مقاله‌ای بر پایه حل مساله بهینه‌سازی پیوسته نامقید با استفاده از روش گرادیان مزدوج[۷] ارایه داد [۳۳]. روش او مبتنی بر مقاله‌ای از پوشن[۸] و همکاران [۳۸] در سال (۱۹۹۰) بود. پوشن و همکاران در این مقاله نشان دادند می‌توان با استفاده از مساله بهینه‌سازی نامقید، جواب تقریبی قابل قبولی برای مساله افرازبندی گراف به دست آورد. این الگوریتم، در فصل ۲ شرح داده خواهد شد.

---

[۱] Henry Wolkowicz [۲] Franz Rendl [۳] Continuous optimization [۴] Optimization-based methods [۵] Donath
[۶] Hoffman [۷] Conjugate gradient [۸] Pothen

۲.۴.۱ الگوریتم‌های دقیق

الگوریتم کاریش و همکاران

کاریش[۱] و همکاران در سال ۲۰۰۰، الگوریتمی دقیق برای مساله دو بخشی کردن گراف[۲] ارایه دادند. روش شاخه و کران[۳] بر پایه ترکیب رهیافت صفحه برش[۴] با برنامه‌ریزی نیمه معین[۵] (بخش ۲.۳ را ببینید)، الگوریتمی بود که آن‌ها برای حل مساله افرازبندی گراف از آن استفاده کردند [۳۰].

الگوریتم هاگر و همکاران

در سال ۱۹۹۹ هاگر و کریلیوک[۶] نشان دادند، مساله افرازبندی گراف در حالتی که راس‌های گراف به دو مجموعه تقسیم می‌شوند، با یک مساله بهینه‌سازی پیوسته درجه دوم[۷]، معادل است [۱۹] (برای حالتی که تعداد مجموعه‌های افراز بیش از ۲ است، [۲۰] را ببینید). پس از آن در سال ۲۰۱۳، هاگر و همکاران الگوریتمی مبتنی بر روش شاخه و کران، برای حل مساله بهینه‌سازی پیوسته که معادل با حل مساله افرازبندی گراف بود، ارایه دادند [۲۳]. کران پایین در روش شاخه و کران، بر اساس بازنویسی تابع هدف مساله بهینه‌سازی پیوسته، به صورت مجموع دو تابع محدب و مقعر، به دست می‌آید. پس از آن به جای بخش مقعر تابع هدف، از یک تابع آفین (بخش ۲.۳ را ببینید) استفاده می‌شود ([۲۱] را ببینید). این الگوریتم در فصل ۳ به تفصیل شرح داده خواهد شد.

---

[۱] Karisch [۲] Graph bisection problem [۳] Branch-and-bound algorithm [۴] Cutting plane [۵] Semidefinite programming [۶] Krylyuk [۷] Continuous quadratic optimization

مراجع

[۱] سلطان پور، حدیث، وفایی جهان، مجید، و جلالی، مهرداد. بهینه سازی تقطیع تصویر مبتنی بر گراف با استفاده از الگوریتم رقابت استعماری. *کنگره ملی مهندسی برق، کامپیوتر و فناوری اطلاعات* (۱۳۹۲)، ۵۲–۴۵.

[۲] قصاب‌مهرجرد، آمنه. بررسی الگوریتم چندسطحی برای حل مساله‌ی k- افرازبندی متعادل گراف. پایان‌نامه کارشناسی‌ارشد، دانشکده علوم ریاضی، دانشگاه فردوسی مشهد، ۱۳۹۲.

[3] Aliprantis, C., and Border, K. *Infinite Dimensional Analysis: A Hitchhiker's Guide.* Springer, 2007.

[4] Babaie-Kafaki, S., and Ghanbari, R. The dai–liao nonlinear conjugate gradient method with optimal parameter choices. *European Journal of Operational Research 234*, 3 (2014), 625–630.

[5] Bapat, R. *Graphs and Matrices.* Springer, 2010.

[6] Bichot, C. E. A metaheuristic based on fusion and fission for partitioning problems. *20th IEEE International Parallel and Distributed Processing Symposium* (2006), 3499 – 3506.

[7] Bichot, C. E. A new method, the fusion fission, for the relaxed k-way graph partitioning problem, and comparisons with some multilevel algorithms. *Journal of Mathematical Modeling and Algorithms 6*, 3 (2007), 319–344.

[8] Bichot, C., and Siarry, P. *Graph Partitioning.* John Wiley & Sons Ltd, 2013.

[9] Blake, Y. H. R. Numerical experiences with partitioning of unstructured meshes. *Parallel Computing 20*, 5 (1994), 815–829.

[10] Chamberlain, B. L. Graph partitioning algorithms for distributing workloads of parallel computations. *University of Washington Technical Report UW-CSE-98-10 3* (1998).

[11] Chong, E. K. P., and Zak, S. H. *An Introduction to Optimization.* Wiley, 2014.

[12] de Klerk, E. *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications.* Springer, 2002.

[13] Dolan, E. D., and More, J. J. Benchmarking optimization software with performance profiles. *Mathematical Programming 91*, 2 (2002), 201–213.

[14] Donath, W. E., and Hoffman, A. J. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development 17*, 5 (1973), 420–425.

[15] Falkner, J., Rendl, F., and Wolkowicz, H. A computational study of graph partitioning. *Mathematical Programming 66*, 1-3 (1994), 211–239.

[16] Fletcher, R., and Reeves, C. M. Function minimization by conjugate gradients. *The Computer Journal 7*, 2 (1964), 149–154.

[17] Garey, M. R., Johnson, D. S., and Stockmeyer, L. Some simplified np-complete problems. *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing 74* (1974), 47–63.

[18] Gribkovskaia, I., Halskau, O., and Laporte, G. The bridges of konigsberg - a historical perspective. *Networks 49*, 3 (2007), 199–203.

[19] Hager, W. W., and Krylyuk, Y. Graph partitioning and continuous quadratic programming. *SIAM Journal on Discrete Mathematics 12*, 4 (1999), 500–523.

[20] Hager, W. W., and Krylyuk, Y. Multiset graph partitioning. *Mathematical Methods of Operations Research 55*, 1 (2002), 1–10.

[21] Hager, W. W., and Phan, D. T. An ellipsoidal branch and bound algorithm for global optimization. *SIAM Journal on Optimization 20*, 2 (2009), 740–758.

[22] Hager, W. W., Phan, D. T., and Zhang, H. An exact algorithm for graph partitioning. *Mathematical Programming 137*, 1-2 (2013), 531–556.

[23] Hager, W., Phan, D., and Zhang, H. An exact algorithm for graph partitioning. *Mathematical Programming 137*, 1-2 (2013), 531–556.

[24] Hendrickson, B., and Kolda, T. G. Graph partitioning models for parallel computing. *Parallel Computing 26*, 12 (2000), 1519–1534.

[25] Horst, R., and Thoai, N. DC programming: Overview. *Journal of Optimization Theory and Applications 103*, 1 (1999), 1–43.

[26] Hsieh, S. H., Paulino, G. H., and Abel, J. F. Recursive spectral algorithms for automatic domain partitioning in parallel finite element analysis. *Computer Methods in Applied Mechanics and Engineering 121*, 1-4 (1995), 137–162.

[27] Jäger, G., and Srivastav, A. Improved approximation algorithms for maximum graph partitioning problems. in *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science*, volume 3328. Springer Berlin Heidelberg, 2005, pp. 348–359.

[28] Johnson, T., Davis, T. A., and Hadfield, S. M. A concurrent dynamic task graph. *Parallel Computing 22*, 2 (1996), 327–333.

[29] Kahng, A., Lienig, J., and Markov, I. *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer, 2010.

[30] Karisch, S. E., Rendl, F., and Clausen, J. Solving graph bisection problems with semidefinite programming. *INFORMS Journal on Computing 12*, 3 (2000), 177–191.

[31] Kernighan, B. W., and Lin, S. An efficient heuristic procedure for partitioning graphs. *The Bell Systems Technical Journal* (1970).

[32] Konstantin, A., and Räcke, H. Balanced graph partitioning. in *Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*. ACM, 2004, pp. 120–124.

[33] Kruyt, N. A conjugate gradient method for the spectral partitioning of graphs. *Parallel Computing 22*, 11 (1997), 1493–1502.

[34] Luenberger, D., and Yinyu, Y. *Linear and Nonlinear Programming.* Springer, 2008.

[35] Munkres, J. R. *Topology; a First Course.* Prentice-Hall, 1974.

[36] Murty, K., and Kabadi, S. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming 39*, 2 (1987), 117–129.

[37] Pardalos, P., and Vavasis, S. Quadratic programming with one negative eigenvalue is NP-hard. *Journal of Global Optimization 1*, 1 (1991), 15–22.

[38] Pothen, A., Simon, H. D., and Liou, K. P. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl. 11*, 3 (1997), 430–452.

[39] Rendl, F., and Wolkowicz, H. A projection technique for partitioning the nodes of a graph. *Annals of Operations Research 58*, 3 (1995), 155–179.

[40] Rudin, W. *Principles of Mathematical Analysis.* McGraw-Hill, 1976.

[41] Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 888–905.

[42] Strang, G., and Goh, C. F. *0-1 graph partitioning and image segmentation.* Ph.D. thesis, MIT Massachusetts Institute of Technology, 2008.

[43] Voloshin, V. I. Graph coloring: History, results and open problems. *Alabama Journal of Mathematics 34* (2009), 92–94.

واژه‌نامه فارسی به انگلیسی

ا

آزادسازی .... Relaxation

اتصال .... Connected

افرازبندی گراف .... Graph partitioning

افرازبندی متوازن .... Partitioning balance

الگوریتم‌های ابتکاری .... Heuristic methods

الگوریتم‌های دقیق .... Exact methods

ب

برنامه‌ریزی نیمه معین .... Semidefinite programming

بهترین تخمین آفین .... Affine underestimate

بهینه‌سازی پیوسته .... Continuous optimization

بهینه‌سازی پیوسته درجه دوم .... Continuous quadratic optimization

بهینه‌سازی تفاضل محدب .... Difference convex optimization

پ

پیکسل .... Pixel

ت

تابع آفین .... Function affine

تابع تفاضل محدب .... Difference convex function

تابع محدب .... Convex function

تابع مقعر .... Concave function

توابع وزن .... Weighting functions

د

درجه .... Degree

دو بخشی کردن طیفی .... Spectral bisection

ر

روش شاخه و کران .... Branch-and-bound algorithm

روش‌های بهینه‌سازی .... Optimization-based methods

ز

زمان‌بندی کارها .... Task scheduling

ص

صفحه برش .... Cutting plane

ف

فیلپ–فلاپ‌ها .... Flip-flops

ق

قطعه‌بندی تصویر .... Image segmentation

ک

کنترل ترافیک هوایی .... Air traffic control

گ

گرادیان مزدوج .... Conjugate gradient

م

ماتریس لاپلاسین .... Laplacian matrix

ماتریس‌های تُنُک .... Sparse matrix

مجموعه فشرده .... Compact set

محاسبات موازی .... Parallel computations

مساله برش کمینه .... Min-cut problem

مساله دو بخشی کردن گراف .... Graph bisection problem

ن

نرم اقلیدسی .... Euclidian norm

نظریه گراف .... Graph theory

نیمه‌معین مثبت .... Positive semidefinite

و

وزن متوسط .... Average weight

ی

یال‌های برشی .... Cut edges

یال‌های غیر برشی .... Uncut edges

واژه‌نامه انگلیسی به فارسی

A

Affine underestimate .... بهترین تخمین آفین

Air traffic control .... کنترل ترافیک هوایی

Average weight .... وزن متوسط

B

Branch-and-bound algorithm .... روش شاخه و کران

C

Compact set .... مجموعه فشرده

Concave function .... تابع مقعر

Conjugate gradient .... گرادیان مزدوج

Connected .... اتصال

Continuous optimization .... بهینه‌سازی پیوسته

Continuous quadratic optimization .... بهینه‌سازی پیوسته درجه دوم

Convex function .... تابع محدب

Cut edges .... یال‌های برشی

Cutting plane .... صفحه برش

D

Degree .... درجه

Difference convex function .... تابع تفاضل محدب

Difference convex optimization .... بهینه‌سازی تفاضل محدب

E

Euclidian norm .... نرم اقلیدسی

Exact methods .... الگوریتم‌های دقیق

F

Flip-flops .... فیلپ–فلاپ‌ها

Function affine .... تابع آفین

G

Graph bisection problem .... مساله دو بخشی کردن گراف

Graph partitioning .... افرازبندی گراف

Graph theory .... نظریه گراف

H

Heuristic methods .... الگوریتم‌های ابتکاری

I

Image segmentation .... قطعه‌بندی تصویر

L

Laplacian matrix .... ماتریس لاپلاسین

M

Min-cut problem .... مساله برش کمینه

O

Optimization-based methods .... روش‌های بهینه‌سازی

P

Parallel computations .... محاسبات موازی

Partitioning balance .... افرازبندی متوازن

Pixel .... پیکسل

Positive semidefinite .... نیمه‌معین مثبت

R

Relaxation .... آزادسازی

S

Semidefinite programming .... برنامه‌ریزی نیمه معین

Sparse matrix .... ماتریس‌های تُنُک

Spectral bisection .... دو بخشی کردن طیفی

T

Task scheduling .... زمان‌بندی کارها

U

Uncut edges .... یال‌های غیر برشی

W

Weighting functions .... توابع وزن

نمایه