

الله اعلم
الله اعلم
الله اعلم



دانشگاه صنعتی شاهرود

دانشکده علوم ریاضی

رشته ریاضی کاربردی، گرایش گراف و ترکیبیات

رساله دکتری

رهیافتی جدید برای حل مسئله رنگ‌آمیزی مجموع و تعمیم آن روی گراف‌های ساده

نگارنده: خلیل عرفانی حیدرنیا

استادان راهنما

دکتر صادق رحیمی شعرباف

دکتر جعفر فتحعلی

استاد مشاور

دکتر میثم علیشاهی

اردیبهشت ۱۳۹۷



فرم شماره ۱۲: صورت جلسه نهایی دفاع از رساله دکتری (Ph.D)

(ویژه دانشجویان ورودی های ۹۴ و ما قبل)

بدینوسیله گواهی می شود آقای خلیل عرفانی حیدرنیا دانشجوی دکتری رشته ریاضی کاربردی - گراف و ترکیبیات به شماره دانشجویی ۹۱۲۴۸۱۵ ورودی مهر ماه سال ۱۳۹۱ در تاریخ ۹۷/۲/۵ از رساله نظری / عملی خود با عنوان: رهیافتی جدید برای حل مسئله رنگ آمیزی مجموع و تعمیم آن روی گراف های ساده دفاع و با اخذ نمره عالی به درجه عالی نائل گردید.

<input checked="" type="checkbox"/> الف) درجه عالی: نمره ۱۹-۲۰	<input type="checkbox"/> ب) درجه بسیار خوب: نمره ۱۸/۹۹-۱۷
<input type="checkbox"/> ج) درجه خوب: نمره ۱۶/۹۹-۱۵	<input type="checkbox"/> د) غیر قابل قبول و نیاز به دفاع مجدد دارد
<input type="checkbox"/> ه) رساله نیاز به اصلاحات دارد	

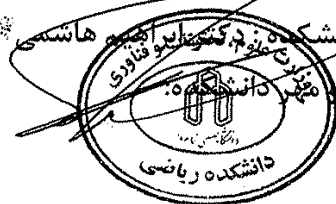
ردیف	هیئت داوران	سمت	مرتبه علمی	امضاء
1	دکتر صادق رحیمی شهرباف	استاد راهنمای اول	استادیار	
2	دکتر جعفر فتحعلی	استاد راهنمای دوم	دانشیار	
3	دکتر میثم علیشاهی	استاد مشاور	استادیار	
4	دکتر سامان بابایی کفاکی	استاد داور (خارجی)	دانشیار	
5	دکتر نادر جعفری راد	استاد داور (داخلی)	دانشیار	
6	دکتر عبدالله آل هوز	استاد داور (داخلی)	استادیار	
7	دکتر ابراهیم هاشمی	سرپرست (نماینده) تحصیلات تکمیلی دانشکده	استاد	

مدیر محترم تحصیلات تکمیلی دانشگاه:

ضمن تأیید مراتب فوق مقرر فرمائید اقدامات لازم در خصوص انجام مراحل دانش آموختگی آقای خلیل عرفانی حیدرنیا بعمل آید.

نام و نام خانوادگی رئیس دانشکده: دکتر ابراهیم هاشمی

تاریخ و امضاء



زیباترین نام را بر زبان جاری می‌کنم ... که هر کس زبان به حمد تو گشود بی‌تردید نگاه تو بر او افتاده. پس بر قلبم آن جاری کن که خود می‌پسندی در ثنایت لب گشایم. در وادی معرفت ننگجد، سرچشمه هدایت نجوشد، سر بر قامت بندگی فرو نیافتد ... گر گنجینه‌ای را که مقدسش خواندی و به آن قسم یاد کردی^۱، کوچک شمرده شود و تنها خاطره جوهر خشک شده‌ای از آن بر برگ برگ صفحات زندگی باقی ماند.

تو علم را روشنی قرار دادی و فانوسی در بیغوله راه که مسیر را، راه نماید و تزکیه را مقدم بر آن دانستی تا نگاهبانش باشد که تزکیه و تعلیم در معیت هم گوهر وجودی انسان را به نور تو منور کند، پرده از واقعیات کنار زند. آن جاست که حقیقت رخ نمایاند، نظر فراتر افتد، خوان گنجینه‌های دانش رنگین شود و ... آری آنجاست که آدمی معنا یابد.

من اگر وعده‌هایم با تو زیر خروارها تل فراموشی و غفلت مدفون گردیده، اگر زشتی طغیان در نظرم زیبا جلوه‌گری می‌کند و چشمانم خشک‌تر از آن است که در مقام توبه اشکی بر آن جاری شود، بدان از سر جهل است و نسیان... اما بار الها چشم طمع بر رحمت دوخته‌ام و در تمنای رهایی از ظلمت ضلالت، ترنم باران معرفت را می‌طلبم، امید آنکه جوانه‌های حقیقت را در وجودم برویاند و انعکاس آن چشمانم را روشن کند.

اکنون چهره بر چهره خاک می‌سایم و تو را به حبیبیت قسم می‌دهم که... "هر آن خصلت ناپسند که در من می‌بینی به لطف واسع خویش اصلاحش فرمای تا پسندیده شود و هر آن عیب که نفسم را به فساد بیالاید از من بازگیر و هر آن نقص که جانم را از کمال باز دارد برطرفش فرمای!"

و در آن روز که نوبت زندگانی به سر رسد و بیک مرگ حلقه بر در خانه تن بکوبد و دعوت واجب الاجابه تو از آسمان‌ها به گوش آید... پروردگارا! بر محمد (ص) و آل پاکش درود فرست و به حق ایشان عمر ما را با رستگاری به پایان آور و عاقبتمان را ختم به خیر فرمای...!

زبان قاصراست و مجال کوتاه...

تو خود قصیده‌ی مهران از لوح نانوشتی قلمم بخوان...!

تقدیم به طالبان علم و تمام کسانی که در این
راه، یاری ام داده‌اند...

سپاس‌گزاری...

ستایش مخصوص خداوند، پروردگار جهانیان است. سپاس مخصوص اوست که ستایشگران از ادای حق سپاسش عاجزند، حسابگران از شمارش نعمت‌های بی‌پایانش وامانده و تلاشگران در ادای حشش درمانده.

هرچند واژه‌ها در ادای حق تشکر آنانی که در تمام دوران زندگی جرعه نوش محبت‌شان بوده‌ام، ناتوانند ولی به رسم ادب و احترام بوسه بر دستانشان زده و بر خود واجب می‌دانم تشکر زبانی خود را از زحمات پدر و مادر عزیز و مهربانم را که همواره راهگشای مشکلاتم در تمام مراحل زندگی بوده‌اند، ارج نهم و مراتب تشکر قلبی را از الطاف و مهربانی آنان ابراز دارم و از همسر که سایه مهربانیش سایه سار زندگیم است، او که اسوه صبر و تحمل بوده و مشکلات مسیر را برایم تسهیل نمود.

سپاس و تقدیر بی‌پایان خود را تقدیم استادان راهنمای بزرگوار جناب آقای دکتر رحیمی و جناب آقای دکتر فتحعلی می‌نمایم که با صبر حوصله فراوان و نظرات و راهنمایی‌های ارزنده نقش مهمی در به ثمر رسیدن این پایان‌نامه ایفا کردند.

همچنین از جناب آقای دکتر علیشاهی که مطالعه و مشاوره این پایان‌نامه را به عهده گرفته‌اند و در طول این مدت، صادقانه و خالصانه پذیرای تمامی سوالات بنده بودند، از اساتید محترم، جناب آقای دکتر بابایی، جناب آقای دکتر جعفری راد و جناب آقای دکتر آل‌هوز که زحمت داوری را تقبل کردند، صمیمانه تشکر و قدردانی می‌کنم.

همچنین از تمام مسئولین و کارکنان محترم دانشکده علوم ریاضی دانشگاه شاهرود و بویژه دوستان عزیزم آقایان دکتر سلیمان حسین‌پور، مرتضی نظری و امیر فیضی که در این مدت متحمل زحماتی از ناحیه بنده شده‌اند، صمیمانه متشکرم و برای آنها بهترین‌ها را از خداوند متعال مسئلت دارم.

ضمناً از خوانندگان محترم تقاضا می‌شود در صورت داشتن هر گونه سوال یا نقدی در خصوص روش ارائه شده و سایر مطالب مربوطه، از طریق ایمیل‌های زیر با اینجانب تماس حاصل نمایند.

khalilerfani@gmail.com

erfani@shahroodut.ac.ir

خلیل عرفانی حیدرنیا
اردیبهشت ۱۳۹۷

تعهد نامه

اینجانب خلیل عرفانی حیدرنیا دانشجوی دکتری رشته ریاضی کاربردی علوم ریاضی دانشگاه شاهرود، نویسنده پایان نامه با عنوان رهیافتی جدید برای حل مسئله رنگ آمیزی مجموع و تعمیم آن روی گراف های ساده ، تحت راهنمایی صادق رحیمی شعرباف و جعفر فتحعلی متعهد می شوم:

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

خلیل عرفانی حیدرنیا

اردیبهشت ۱۳۹۷

مالکیت نتایج و حق نشر

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

چکیده

این رساله به مطالعه نوع خاصی از رنگ آمیزی با عنوان رنگ آمیزی مجموع می‌پردازد. هدف اصلی در این مسئله یافتن کمینه‌ی ممکن برای مجموع رنگ‌ها (اعداد طبیعی) در بین رنگ آمیزی‌های مجاز است. بطور کلی دو دیدگاه در برخورد با این مسئله وجود داشته است. رویکرد تحلیلی و رویکرد الگوریتمی. رویکرد اصلی این رساله در بررسی این مسئله بیشتر متوجه دیدگاه دوم است. در این راستا یکی از روش‌های فرا ابتکاری برای این مسئله NP-سخت، روش جستجوی همسایگی متغیر است. برای افزایش سرعت جستجوی همسایگی‌ها، از مفهوم جدیدی بنام نگهدارندگی استفاده شده است. بعد از پیاده سازی این روش برای مسئله رنگ آمیزی مجموع، کارایی آن روی چند نمونه گراف مورد بررسی و مقایسه قرار گرفته است. همچنین ارایه دو مفهوم مجموع رنگی راسی تفاضلی و جمعی به عنوان دو تعمیم برای مجموع رنگی راسی گراف، موضوعی است که در این رساله مورد توجه قرار گرفته است. شرایط عدم وجود هم‌ریختی بین دو گراف و یافتن کران برای عدد رنگی کسری از جمله نتایج این دو مفهوم جدید است.

کلمات کلیدی: مجموع رنگی راسی، جستجوی همسایگی متغیر، مجموعه‌های نگهدارنده و کاهنده، مجموع رنگی راسی تفاضلی و جمعی، عدم وجود هم‌ریختی بین دو گراف، عدد رنگی کسری، گراف کسری.

لیست مقالات مستخرج از پایان نامه

۱. S. Rahimi Sharbaf, K. Erfani. On the sum and difference edge-coloring problem of simple graphs. Algebraic Structures and Their Applications, Vol. 4, No. 1 (2017), pp 33-42.

۲. S. Rahimi Sharbaf, J. Fathali, K. Erfani, New Variable Neighborhood Search Method for Minimum Sum Coloring on Simple Graphs, Iranian Journal of Numerical Mathematics and Optimization, Ferdowsi University of Mashhad, Accepted.

۳. خلیل عرفانی حیدرنیا، جعفر فتحعلی و صادق رحیمی شعرباف، روش جستجوی همسایگی متغیر برای حل مسئله رنگ آمیزی مجموع کمینه روی گراف‌های ساده، دهمین کنفرانس بین المللی انجمن ایرانی تحقیق در عملیات، دانشگاه مازندران، اردیبهشت ۹۶.

۴. خلیل عرفانی حیدرنیا و صادق رحیمی شعرباف، رهیافتی جدید بر مدل بندی غیر خطی عدد صحیح برای مسئله رنگ آمیزی مجموع روی گراف‌های ساده، اولین کنفرانس ملی بهینه سازی و تصمیم گیری، دانشگاه مازندران، اسفند ۹۴.

فهرست مطالب

ن	فهرست تصاویر
ع	فهرست جداول
۱	پیش‌گفتار
۵	۱ تعاریف مقدماتی
۵	۱.۱ تعاریف
۱۰	۲.۱ تعریف برخی از انواع رنگ آمیزی روی گراف‌های ساده
۱۳	۳.۱ رنگ‌آمیزی مجموع رنگی
۱۳	۴.۱ پیشینه مسئله
۱۵	۲ اهمیت مسئله رنگ‌آمیزی مجموع و پیچیدگی محاسباتی آن
۱۶	۱.۲ برخی از کاربردها
۱۷	۱.۱.۲ زمانبندی و تخصیص منابع
۱۸	۲.۱.۲ طراحی $VLSI$ مدار الکتریکی
۲۰	۲.۲ پیچیدگی محاسباتی
۲۰	۱.۲.۲ مقدماتی بر نظریه NP
۲۳	۲.۲.۲ NP -کامل بودن مسئله تصمیم‌گیری رنگ‌آمیزی مجموع
۲۵	۳.۲.۲ مسئله رنگ‌آمیزی مجموع روی گراف‌های خاص
۲۹	۳ کران مجموع رنگی
۳۰	۱.۳ کران تحلیلی
۳۴	۲.۳ کران الگوریتمی
۳۴	۱.۲.۳ افراز به خوشه‌ها
۳۹	۴ تعمیم مجموع رنگی و کاربرد آن
۴۰	۱.۴ مقدمه
۴۲	۲.۴ مجموع رنگی تفاضلی و مجموع رنگی جمعی

۴۵	شرایط لازم برای وجود همریختی بین دو گراف	۳.۴
۴۷	کران بالا برای گراف کنسر	۴.۴
۵۱	کران پایین برای عدد رنگی کسری	۵.۴
۵۵	۵ روش جستجوی همسایگی متغیر برای مسئله MSC	
۵۶	مدل ریاضی مسئله MSC	۱.۵
۵۷	مدل ریاضی غیرخطی عدد صحیح مسئله MSC	۱.۱.۵
۵۸	افراز فضای شدنی	۲.۱.۵
۶۳	معرفی همسایگی های N_k	۲.۵
۶۵	همسایگی برای مسئله MSC	۱.۲.۵
۶۶	نگهدارندگی و کاهندگی	۳.۵
۶۷	مجموعه های نگهدارنده و کاهنده	۱.۳.۵
۶۸	ارتباط مجموعه های نگهدارنده و کاهنده با مسئله MSC	۲.۳.۵
۶۹	ترتیب ریزش	۳.۳.۵
۷۲	مکانیسم آشفستگی برای خروج از بهینه محلی	۴.۳.۵
۷۳	روش VNS برای MSC	۴.۵
۷۴	جستجوی همسایگی متغیر مورب	۱.۴.۵
۷۸	تعمیم روش $SVNS$ برای GCP	۵.۵
۸۳	۶ پیاده سازی و نتایج محاسباتی	
۸۴	نتایج تجربی	۱.۶
۸۷	نتیجه گیری و پیشنهادات	۲.۶
۸۹	مراجع	
۹۵	آ برنامه های متلب	
۱۰۳	واژه نامه فارسی به انگلیسی	
۱۰۵	واژه نامه انگلیسی به فارسی	
۱۰۷	نمایه	

فهرست تصاویر

۷	مثالی از گراف G و برخی از زیرگراف‌های آن	۱۰.۱
۱۴	$\chi(G) = 2 < 3 = s(G)$	۲۰.۱
۱۷	رنگ آمیزی مجموع چندگانه و رنگ آمیزی بهینه مجموع چندگانه	۱۰.۲
۱۸	نمونه ای از زمانبندی سیستم توزیع شده با ۴ فعالیت و ۵ پردازنده	۲۰.۲
۱۹	مثالی از اتصالات بین پایانه‌ها در طراحی $VLSI$	۳۰.۲
۲۰	گراف بازه‌ای متناظر با مثال شکل (۳.۲)	۴۰.۲
۲۷	درخت $B(3, 5)$	۵۰.۲
۳۶	دو نمونه زیر گراف جزئی با تجزیه به خوشه برای گراف G	۱۰.۳
۴۳	مجموع رنگی تفاضلی با $ds(G) = 4$ ($\chi(G) = 3$)	۱۰.۴
۴۳	$ss(G) \neq s(G)$	۲۰.۴
۴۴	$\Delta(G) = 4, \delta(G) = 1$	۳۰.۴
۴۴	گراف پترسن	۴۰.۴
۵۰	S یک مجموعه مستقل دلخواه از G و I مجموعه یال‌های بین S و $G \setminus S$	۵۰.۴
۶۱	برگ‌های مجاور با راس x در رنگ آمیزی بهینه وضعیت یکسانی دارند	۱۰.۵
۶۱	برگ‌هایی که حتما باید دارای رنگ ۲ در رنگ آمیزی بهینه باشند	۲۰.۵
۶۳	جهت گذاری مفروض و رنگ آمیزی بهینه متناظر با آن	۳۰.۵
۶۴	مسئله بهینه سازی یک بعدی با همسایگی فاصله اقلیدسی	۴۰.۵
۶۶	همسایگی $N_1(X)$	۵۰.۵
۶۹	گراف myciel4 با ۲۳ راس و نمودار دو بعدی رنگ راس متناظر با رنگ آمیزی بهینه	۶۰.۵
۷۱	ترتیب ریزش	۷۰.۵
۷۳	توضیح مکانیسم ایجاد آشفتگی	۸۰.۵
۹۰.۵	خروجی عملکرد $SVNS$ روی مسئله $myciel^3$ در سه گام متوالی منتهی به جواب	
۷۸	بهینه	

۱۰.۵	خروجی عملکرد <i>SVNS</i> روی مسئله <i>myciel</i> ^۴ در چهار گام متوالی منتهی به
۷۹ جواب بهینه
۱۱.۵	خروجی عملکرد <i>SVNS</i> روی مسئله 5×5 <i>queen</i> در سه گام متوالی منتهی به
۸۰ جواب بهینه

فهرست جداول

۲۵	گراف‌هایی که برای آنها مسئله رنگ آمیزی مجموع یک مسئله NP -سخت است	۱.۲
۸۵	نتایج محاسباتی حاصل از اجرای الگوریتم SVNS	۱.۶
۸۶	مقایسه نتایج بین روش SVNS و پنج روش دیگر	۲.۶

پیش‌گفتار

مسائل بهینه‌سازی به طور معمول به دو دسته، مسائل بهینه‌سازی پیوسته^۲ و مسائل بهینه‌سازی گسسته^۳ تقسیم می‌شوند. مسائل بهینه‌سازی پیوسته آن دسته از مسائلی هستند که متغیرهای آن‌ها در فضای پیوسته تغییر می‌کنند. معمولاً هدف اصلی در این مسائل یافتن یک عدد حقیقی یا تابع حقیقی است که بازای آن معیار مشخصی، بهینه شود. از سویی دیگر، مسائل بهینه‌سازی گسسته به آن دسته از مسائلی اطلاق می‌شود که متغیرهای مربوطه در آن‌ها به فضای گسسته تعلق داشته باشند و در آن فضا تغییر کنند. هدف اصلی در این گونه مسائل یافتن یک شیء یا مفهوم ترکیبیاتی مانند یک جایگشت، برداری از اعداد صحیح، گراف و غیره است که بازای آن تابعی با معیار مشخص در فضای متناهی از جواب‌های موجه بهینه شود. بطور کلی هرکدام از این دو دسته از مسائل بهینه‌سازی، روش‌ها و تکنیک‌های حل متفاوتی را دارند.

امروزه به گواهی تجربیات مشترک بسیاری از برنامه‌نویسان، سرعت رشد محاسبات ترکیبیاتی نسبت به محاسبات عددی در بسیاری از برنامه‌های کاربردی، در حال افزایش است. دلیل این امر بخاطر آن است که در پس زمینه بسیاری از مسائل گذشته در ارتباط با کاربرد ریاضیات در مهندسی و فیزیک، ساختارهای گسسته ظهور و بروز بیشتری نسبت به ساختارهای پیوسته دارند. بهینه‌سازی گسسته، شاخه‌ای از ریاضیات کاربردی و علوم کامپیوتر است که شامل موضوعات متنوعی در زمینه نظریه گراف^۴، طراحی شبکه^۵، تحقیق در عملیات^۶، برنامه ریزی ریاضی^۷، توالی و زمانبندی^۸ و غیره است. یکی از زیر شاخه‌ها مربوط به آن، مسئله رنگ‌آمیزی گراف است که این رساله، به نوع خاصی از آن می‌پردازد.

رنگ‌آمیزی گراف، یکی از قدیمی‌ترین و شناخته شده ترین مسائل، در نظریه گراف است. آمارها نشان می‌دهد که موضوع رنگ‌آمیزی گراف‌ها یکی از مباحث اصلی، از میان مجموعه‌ای از صدها مسائل ترکیبیاتی کلاسیک است [۹]. این امر به دلیل ساده بودن ماهیت مسئله به انضمام قابلیت کاربرد گسترده‌اش در مسائل عملی، در دنیای واقعی است. متأسفانه ویژگی پیچیدگی محاسباتی بالای

²Continuous optimization problems

³Discrete optimization problems

⁴Graph theory

⁵Network design

⁶Operations research

⁷Mathematical programming

⁸Sequencing and scheduling

این مسئله منجر می‌شود که گاهی اوقات جواب‌های کارایی در زمان معقول، حتی برای گراف‌های نه چندان بزرگ، بدست نیایند. به این معنی که یافتن الگوریتمی با زمان چند جمله‌ای نسبت به اندازه ورودی (تعداد رئوس یا یال‌های گراف) کار طاقت فرسا و احتمالاً غیر ممکن است. در نتیجه بدیهی است که برای حل این مسئله روی گراف‌هایی که در دنیای واقعی با آن‌ها روبرو می‌شویم، دچار مشکل هستیم. لذا مطالعه و پژوهش روی روش‌های تقریبی حل این مسائل از جمله موضوعاتی است که در چند دهه اخیر، ذهن بسیاری از دانشمندان این رشته را بخود جلب کرده‌است. تاریخچه مسئله رنگ آمیزی گراف به سال ۱۸۵۲ میلادی بر می‌گردد، زمانی که دمورگان^۹ در نامه‌ای به دوست خود همیلتون^{۱۰} نوشت که یکی از دانشجویانش متوجه شده است که برای رنگ آمیزی یک نقشه از شهرهای انگلستان، تنها چهار رنگ کافی است بطوریکه رنگ شهرهای مجاور هم یکسان نباشند. این مسئله بعدها به مسئله چهار رنگ^{۱۱} شهرت یافت. در واقع سوال اصلی به این صورت مطرح شده بود که حداقل تعداد رنگ برای رنگ کردن شهرهای یک نقشه چند تا است بطوریکه شرط یکسان نبودن رنگ شهرهای همسایه رعایت شود؟

این مسئله اولین بار توسط کیلی^{۱۲} در سال ۱۸۷۸ بصورت پازل در دسترس عموم قرار گرفت و اولین اثبات توسط کمپ^{۱۳} [۲۶] ارائه شد. تقریباً بعد از ده سال که مسئله چهار رنگ را حل شده می‌پنداشتند، هیوود^{۱۴} [۲۰] نشان داد که اثبات ارائه شده توسط کمپ دارای خطاست. سپس او با تکنیک کمپ نشان داد که هر نقشه ۵- رنگ پذیر است. در نهایت پس از فراز و نشیب‌های بسیار و بعد از طی تقریباً یک قرن در سال ۱۹۷۶ میلادی اپیل^{۱۵} و همکارانش [۳، ۲] با استفاده از رایانه این قضیه مهم را ثابت کردند. این اولین قضیه مهمی بود که با استفاده از رایانه به اثبات رسید و نشان داده شد که نه تنها از رایانه در فرایند جستجو و حل مسائل می‌توان استفاده کرد، بلکه ممکن است در اثبات قضایا نیز نقش مهمی ایفا کند. البته هنوز هم برخی دانشمندان در صدد ارائه اثباتی موجز تر و راحت تر برای این مسئله مهار نشدنی هستند.

مسئله رنگ آمیزی گراف در فرم کلاسیک آن به دو صورت رنگ آمیزی راسی^{۱۶} و رنگ آمیزی یالی^{۱۷} بیان می‌شود که عبارت است از یافتن کمترین تعداد رنگ برای رنگ آمیزی رئوس (یال‌ها) گراف، بطوریکه رئوس (یال‌های) مجاور دارای رنگ یکسان نباشند.

بسیاری از مسائل به سادگی قابل تبدیل به مدل رنگ آمیزی کلاسیک نیستند، همین امر منجر به معرفی مدل‌های رنگ آمیزی عمومی‌تری شد. این گونه مدل‌های غیرکلاسیک رنگ آمیزی، می‌تواند شامل تخصیص رنگ‌ها به رئوس یا یال یا وجوه یک گراف یا حتی تلفیقی از موارد بیان شده باشد.

⁹De Morgan

¹⁰Hamilton

¹¹Four Coloring Problem (FCP)

¹²Cayley

¹³Kampe

¹⁴Heawood

¹⁵Appel

¹⁶Vertex Coloring

¹⁷Edge Coloring

علاوه بر این، در هر یک از این مدل‌های رنگ آمیزی می‌توان قوانینی در ارتباط با مجاز بودن یا بهینه بودن جواب‌ها در نظر گرفت. بطور کلی انواع مختلفی از مدل‌های رنگ آمیزی در مقالات متعدد مورد بررسی قرار گرفته‌است [۲۲]. جهت آشنایی بیشتر خواننده، توضیح مختصری در مورد هر یک از رنگ آمیزی‌های فوق در فصل اول آمده‌است.

موضوع این رساله به نوعی خاص از رنگ آمیزی که به مدل رنگ آمیزی کلاسیک نزدیک است و از آن با عنوان **رنگ آمیزی مجموع رنگی** یاد می‌شود، اختصاص دارد. علیرغم سادگی مدل این نوع رنگ آمیزی، حل این مسئله به معنی یافتن رنگ آمیزی بهینه به هیچ عنوان ساده نیست. از طرف دیگر این مسئله قابلیت کاربرد زیادی در مسائل عملی دارد. از جمله مهمترین کاربردهای آن می‌توان به موضوع تخصیص منابع توزیع شده^{۱۸} اشاره کرد. همین امر موجبات توجه بیشتر به حل این مسئله در مباحث نظری را فراهم آورده‌است. بعنوان مثال در سال‌های اخیر مقالات متعددی در زمینه یافتن کران‌های تحلیلی و الگوریتمی،^{۱۹} بررسی پیچیدگی محاسباتی روی گراف‌های خاص و ارائه الگوریتم‌های کارا برای یافتن رنگ آمیزی بهینه روی آن گراف‌ها، ارائه الگوریتم‌های ابتکاری و فراابتکاری^{۲۰} به منظور یافتن جواب‌های زیربهینه ارائه شده است [۳۹، ۴۳، ۴۳، ۱۳، ۳۱، ۱۵، ۲۳، ۴۹]. مباحث این رساله در شش فصل به شرح زیر ارائه شده است:

- در فصل اول، تعاریف مقدماتی مورد نیاز از نظریه گراف آورده شده است.
- در فصل دوم کاربرد و پیچیدگی محاسباتی آنرا بیان و اثبات می‌کنیم. در قسمت کاربرد از مفهوم رنگ آمیزی مجموع چندگانه استفاده می‌کنیم و دو کاربرد مهم این مسئله یعنی زمانبندی و تخصیص منابع به همراه طراحی *VLSI* مدار الکتریکی بحث می‌شود. برای اثبات پیچیدگی محاسباتی مسئله، دانستن مقدماتی از نظریه *NP* ضروری است که به اختصار اشاره شده است.
- از آنجایی که مسئله رنگ آمیزی مجموع به رده مسائل *NP* تعلق دارد، لذا دسته وسیعی از مقالات به سراغ یافتن بازه‌ای رفته‌اند که مجموع رنگی یا قدرت راسی به آنها تعلق دارند. در واقع این کران‌ها (بالا یا پایین) بر اساس سایر پارامترهای موجود گراف مثل عدد رنگی، عدد استقلال، بیشترین درجه و غیره است (کران تحلیلی). اما دسته‌ای دیگر از مقالات سراغ کرانهایی برای این مسئله رفته‌اند که بسیار به روش اتخاذ شده متکی است. از این کران‌ها با نام کران الگوریتمی یاد می‌شود. لذا فصل سوم به موضوع تعیین کران برای مسئله رنگ آمیزی مجموع اختصاص دارد.

۱ فصل چهارم به ارائه دو مفهوم جدید که تعمیم‌هایی از مجموع رنگی است می‌پردازد. هدف اصلی از ارائه این مفاهیم دستیابی به نتایجی همچون شرایط لازم برای وجود هم‌ریختی بین دو

¹⁸Distributed Resource Allocation

¹⁹Analytic and algorithmic bounds

²⁰Heuristic and metaheuristic algorithms

گراف با استفاده از این مفاهیم است. از جمله نتایج دیگر این مفاهیم، یافتن کران پایین برای عدد رنگی کسری است که در این فصل بررسی شده‌است.

- در فصل پنجم بنا بر روال معمول و منطقی در مواجهه با مسائل NP -سخت، به ارائه روشی فرا ابتکاری برای مسئله رنگ آمیزی مجموع به منظور یافتن جواب‌های نزدیک بهینه پرداخته می‌شود. روش جستجوی همسایگی متغیر با تعریف همسایگی جدید موضوعی است که بیشتر در این فصل مورد توجه قرار گرفته‌است. مفاهیمی همچون نگهدارندگی و کاهندگی راسی و مجموعه‌ای، دو مفهوم جدیدی است که در این فضا مطرح شده‌است. از این مفاهیم در افزایش سرعت جستجوی فضای شدنی استفاده شده‌است.
- در فصل ششم با استفاده از نرم افزار متلب روش ارائه شده‌ی در فصل پنجم، پیاده سازی و کارایی الگوریتم بیان شده روی دسته‌ای از نمونه مسائل پر تکرار با روش‌های ارائه شده در دهه گذشته مقایسه شده‌است.

فصل ۱

تعاریف مقدماتی

علیرغم وجود تنوع مدل‌های رنگ‌آمیزی گراف، همچنان رنگ‌آمیزی کلاسیک یکی از مهمترین مدل‌های پر کاربرد است. موضوع اصلی رنگ‌آمیزی عبارت است از یافتن کمترین تعداد رنگ برای رنگ‌آمیزی رئوس (یاال‌ها) گراف، بطوریکه رئوس (یاال‌های) مجاور دارای رنگ یکسان نباشند. بسیاری از مسائل مربوط به رنگ‌آمیزی به سادگی قابل تبدیل به مدل رنگ‌آمیزی کلاسیک نیستند، همین امر منجر به معرفی مدل‌های رنگ‌آمیزی عمومی‌تری شد. در این فصل ابتدا برخی از تعاریف اولیه [۷] و سپس چند مدل رنگ‌آمیزی غیر کلاسیک [۲۸] ارائه می‌گردد تا امکان مقایسه بین این مدل‌ها برای خواننده محترم وجود داشته باشد.

۱.۱ تعاریف

تعریف ۱.۱.۱. یک سه‌تایی مرتب $(V(G), E(G), \psi_G)$ را گراف می‌گویند که در آن $V(G)$ مجموعه‌ای ناتمامی از رئوس G ، $E(G)$ مجموعه‌ای مجزا از $V(G)$ ، شامل یاال‌های G و ψ_G تابع وقوع^۱ از $E(G)$ به $V(G)$ ، به طوری که به هر یاال از $E(G)$ دو رأس از $V(G)$ که الزاماً متمایز نیستند، اختصاص می‌دهد.

اگر تابع ψ_G به یاال e ، دو رأس u و v را اختصاص دهد (یعنی $\psi_G(e) = uv$)، آنگاه می‌گویند یاال e ، دو رأس u و v را به یک‌دیگر متصل کرده است و رأس‌های u و v را دو انتهای (دو سر) یاال e

^۱Incidence function

می‌نامند و می‌گویند، رأس‌های u و v بر یال e واقع هستند و برعکس، یال e بر روی رأس‌های u و v واقع است. سه‌تایی مرتب $(V(G), E(G), \psi_G)$ را می‌توانیم به صورت گرافیکی نیز نمایش دهیم. برای سادگی، نمادهای $V(G)$ و $E(G)$ را گاهی به ترتیب، به صورت V و E می‌نویسیم و گراف G را به صورت $G(V, E)$ نشان می‌دهیم.

تعریف ۲.۱.۱. یک یال با دو انتهای یکسان را طوقه^۲ و یک یال با دو انتهای متمایز را یال پیوندی^۳ گوئیم. اگر بین دو رأس مشخص از یک گراف، بیش از یک یال وجود داشته باشد، آن یال‌ها را موازی^۴ هستند.^۵

تعریف ۳.۱.۱. تعداد یال‌های متصل به رأس v را درجه رأس v در گراف G می‌نامیم که آن را با نماد $deg(v)$ نمایش می‌دهند، یعنی: $deg(v) = |\{e \in E : v \in e\}|$. همچنین بیشترین و کمترین درجه گراف G را به ترتیب با نماد $\Delta(G)$ و $\delta(G)$ نمایش می‌دهند. بعلاوه مقدار عبارت $g(G) = \frac{2m}{n(n-1)}$ چگالی^۶ گراف G نام دارد.

تعریف ۴.۱.۱. در گراف G ، همسایگی^۷ هر رأس $v \in V(G)$ را با $N(v)$ نشان می‌دهند و عبارت است از

$$N(v) = \{u \in V(G) : uv \in E(G)\}$$

تعریف ۵.۱.۱. گراف H را زیرگراف G می‌گویند، هرگاه $E(H) \subseteq E(G)$ ، $V(H) \subseteq V(G)$ باشد. فرض آن که H زیرگرافی از G است، H را زیرگراف سره^۸ G گویند هرگاه $V(H) \neq V(G)$ باشد. گراف $G = (V, E)$ را در نظر بگیرید. اگر $V' \subseteq V$ و $V' \neq \emptyset$ ، آنگاه زیرگرافی از G که مجموعه‌ی رأس‌های آن V' و مجموعه‌ی یال‌هایش برابر مجموعه‌ی یال‌هایی از G باشد که هر دو سر آنها در V' واقع است، زیرگراف القاء شده^{۱۰} (یا زیرگراف القایی راسی) توسط V' نامیده شده و با $G[V']$ نمایش داده می‌شود.

زیرگراف القایی $G[V \setminus V']$ را با $G - V'$ نیز نمایش می‌دهیم و آن زیرگرافی از G است که با حذف رأس‌های V' و یال‌های واقع بر آنها، به دست می‌آید.

مشابهاً اگر $E' \subseteq E$ و $E' \neq \emptyset$ ، آنگاه زیرگرافی از G با مجموعه‌ی رأس‌های دو سر یال‌های E' و مجموعه یال‌های E' را زیرگراف القاء شده به وسیله‌ی E' (یا زیرگراف القایی یالی^{۱۱} E') می‌نامیم و

²Loop

³Link

⁴Parallel

^۵عمده بحث ما در این رساله مربوط به گراف ساده است. گراف ساده گرافی است که شامل یال جهت دار، طوقه و یال موازی نباشد.

⁶Density

⁷Neighborhood

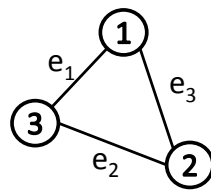
⁸Subgraph

⁹Proper

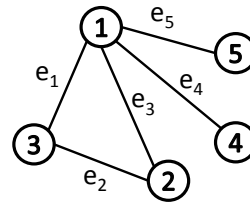
¹⁰Induced Subgraph

¹¹Edge Induced Subgraph

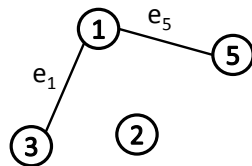
با $G[E']$ نمایش می‌دهیم. برای مثال، شکل (آ.۵) را ملاحظه کنید.



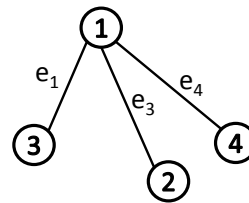
(ب) زیر گراف القایی
یالی و راسی G



(آ) گراف G



(د) زیر گراف G که نه القایی
یالی است نه القایی راسی



(ج) زیر گراف القایی یالی G
که راسی نیست

شکل ۱.۱: مثالی از گراف G و برخی از زیرگراف‌های آن

تعریف ۶.۱.۱. دنباله v_1, v_2, \dots, v_k را یک مسیر^{۱۲} بین v_1 و v_k می‌نامیم هرگاه اولاً هر راس حداکثر یکبار ظاهر شود و ثانیاً برای هر i داشته باشیم: $\{v_i, v_{i+1}\} \in E$. بعلاوه مسیر فوق یک دور^{۱۳} است، هرگاه $v_1 = v_k$.

تعریف ۷.۱.۱. گراف G با حداقل دو راس را همبند^{۱۴} می‌گویند، هرگاه بین هر دو راس آن مسیری موجود باشد. گراف با یک راس نیز همبند فرض می‌شود. همچنین فاصله بین دو راس u و v در گراف G که آنرا با نماد $d(u, v)$ نمایش داده می‌شود، برابر است با طول کوتاهترین مسیر (تعداد یال‌های مسیر) بین u و v . همچنین قرار داد می‌کنیم $d(u, u) = 0$.

فرض کنید آفرازی از مجموعه رئوس V به زیرمجموعه‌های ناتهی V_1 و V_2 و ... و V_w وجود دارد بطوریکه در آن دو رأس u و v همبندند اگر و تنها اگر u و v هر دو متعلق به مجموعه‌ی V_i یکسانی باشند. در این صورت زیرگراف‌های $G[V_1]$ ، $G[V_2]$ ، ... و $G[V_w]$ ، مؤلفه^{۱۵}های G نامیده می‌شوند.

تعریف ۸.۱.۱. یک خوشه^{۱۶} در گراف G عبارتست از زیر مجموعه‌ای از راس‌ها که در آن هر دو راس با هم مجاورند. خوشه V' در گراف G ماکسیمال است، هرگاه خوشه‌ی V'' موجود نباشد

¹²Path

¹³Cycle

¹⁴Connected

¹⁵Component

¹⁶Clique

که $V' \subset V''$ با به عبارت دیگر هرگاه برای هر مجموعه‌ای مثل V'' که شامل V' است، V'' خاصیت خوشه بودن را نداشته باشد. به علاوه عدد خوشه‌ای^{۱۷} گراف G را با نماد $\omega(G)$ نمایش داده می‌شود و برابر است با مرتبه بزرگترین خوشه در گراف G .

تعریف ۹.۱۰.۱. یک مجموعه مستقل^{۱۸} در گراف G عبارتست از زیر مجموعه‌ای از راس‌ها که در آن هیچ دو راسی با هم مجاور نیستند. مجموعه مستقل V' در گراف G ماکسیمال است، هرگاه زیر مجموعه هیچ مجموعه مستقل دیگری نباشد. همچنین عدد استقلال^{۱۹} گراف G را با نماد $\alpha(G)$ نمایش داده می‌شود و برابر است با مرتبه بزرگترین مجموعه مستقل در گراف G .

تعریف ۱۰.۱۰.۱. برای گراف $G = (V, E)$ گراف $G' = (V, E')$ گراف مکمل،^{۲۰} نامیده می‌شود، هر گاه داشته باشیم

$$\forall (u, v) \in E \iff (u, v) \notin E'$$

تعریف ۱۱.۱.۱. گراف G را k -بخشی^{۲۱} می‌نامیم هرگاه بتوان مجموعه راس‌های آن را به k مجموعه‌ی مستقل V_1, \dots, V_k افراز کرد. به علاوه گراف G ، k -بخشی کامل است هرگاه k -بخشی باشد و شامل حداکثر تعداد یالهای ممکن باشد. به عبارت دیگر هر راس با تمام راس‌های مجموعه‌های دیگر مجاور باشد. نماد K_{n_1, n_2, \dots, n_k} برای معرفی گراف k -بخشی کامل استفاده می‌شود که در آن $|V_i| = n_i$ برای $i = 1, \dots, k$. همچنین گراف ستاره^{۲۲} S_n گراف دو بخشی کامل است که یکی از بخش‌های آن تنها یک راس داشته باشد.

تعریف ۱۲.۱.۱. گراف $G_1 = (V_1, E_1)$ و $G_2 = (V_2, E_2)$ را در نظر بگیرید. جمع (اتصال^{۲۳}) این دو گراف با نماد $G_1 + G_2$ نمایش داده و بصورت زیر تعریف می‌شود:

$$V(G_1 + G_2) = V(G_1) \cup V(G_2)$$

$$E(G_1 + G_2) = E_1 \cup E_2 \cup \{\{a, b\} \mid a \in V_1, b \in V_2\}$$

تعریف ۱۳.۱.۱. گراف $G_1 = (V_1, E_1)$ و $G_2 = (V_2, E_2)$ را در نظر بگیرید. حاصلضرب دکارتی^{۲۴} این دو گراف با نماد $G_1 \times G_2$ نمایش داده و بصورت زیر تعریف می‌شود:

$$V(G_1 \times G_2) = \{(u, v) \mid u \in V_1, v \in V_2\}$$

¹⁷Clique number

¹⁸Independent set

¹⁹Independence number

²⁰Complementary

²¹k-partite

²²Star

²³Joint

²⁴Cartesian Product

همچنین دو راس $(u_1, v_1), (u_2, v_2)$ در حداقل یکی از دو حالت زیر با هم مجاورند:

$$(۱) \quad v_1 v_2 \in E_2 \text{ و } u_1 = u_2$$

یا

$$(۲) \quad u_1 u_2 \in E_1 \text{ و } v_1 = v_2$$

تعریف ۱۴.۱.۱. تابع $c: V \rightarrow \mathbb{N}$ یک رنگ آمیزی راسی مجاز^{۲۵} برای گراف $G = (V, E)$ است. هرگاه برای هر $\{a, b\} \in E$ ، $c(a) \neq c(b)$. به تابع c تابع رنگ آمیزی گفته می شود و به گراف G که دارای یک تابع رنگ آمیزی مجاز با k رنگ باشد، k -رنگ پذیر^{۲۶} گفته می شود. در این حالت رنگ آمیزی c ، راس های گراف G را به زیر مجموعه های مستقل V_1, V_2, \dots, V_k افراز می کند که در آن $i = 1, \dots, k$ برای هر $V_i = \{v \in V \mid c(v) = i\}$

مشابهاً یک رنگ آمیزی یالی مجاز^{۲۷} برای G عبارتست از تابعی مثل $c: E(G) \rightarrow \mathbb{N}$ ، به طوری که یال های مجاور رنگ یکسان نداشته باشند.

تعریف ۱۵.۱.۱. عدد رنگی^{۲۸} گراف G که آنرا با $\chi(G)$ نمایش می دهیم عبارتست از کوچکترین میزان k برای اینکه گراف G ، k -رنگ پذیر باشد. در این حالت گراف G را k -رنگی^{۲۹} است.

تعریف ۱۶.۱.۱. گراف r -منتظم^{۳۰} گرافی است که درجه تمام راس های آن برابر r باشد. گراف مکعبی^{۳۱} نام دیگر گراف های ۳-منتظم است. همچنین گراف Q_r نوعی از گراف r -منتظم است که دارای 2^r راس و $r \cdot 2^{r-1}$ یال می باشد. در واقع مجموعه رئوس این گراف متناظر مجموعه تمام دنباله های دودویی بطول k است و دو راس در صورتی مجاورند که دنباله های دودویی متناظر فقط در یک حرف با هم تفاوت داشته باشند.

تعریف ۱۷.۱.۱. گراف دوری^{۳۲} C_n یک گراف ۲-منتظم همبند از مرتبه n است. گراف چرخ^{۳۳} W_n برابر است با $C_{n-1} + K_1$. گراف مسیر^{۳۴} P_n همان دور C_n است که یک یال از آن حذف شده باشد.

تعریف ۱۸.۱.۱. گرافی که شامل هیچ دوری نباشد جنگل^{۳۵} و جنگلی که همبند باشد درخت^{۳۶} نام دارد.

²⁵Legal vertex coloring

²⁶k-colorable

²⁷Legal edge coloring

²⁸Chromatic number

²⁹k-chromatic

³⁰r-regular

³¹Cubic

³²Cycle

³³Wheel

³⁴path

³⁵Forest

³⁶Tree

تعریف ۱۹.۱.۱. گراف بازه ای^{۳۷} یا فاصله گرافی است که بتوان به هر راس آن یک بازه نسبت داد. در این گراف دو راس باهم مجاورند اگر و فقط اگر بازه‌های متناظر باهم اشتراک داشته باشند.

لازم به ذکر است که تعاریف مورد نیاز از جمله تعریف هم‌ریختی، یک‌ریختی، خودریختی، گراف کنسر، عدد رنگی کسری در ابتدای همان فصل آورده مربوطه شده‌است.

۲.۱ تعریف برخی از انواع رنگ آمیزی روی گراف‌های ساده

در این قسمت به معرفی مختصری از سایر مدل‌های رنگ آمیزی پرداخته می‌شود. بعضی از سایر مدل‌های رنگ آمیزی عبارتند از [۲۸]:

- رنگ آمیزی منصفانه^{۳۸}
- T -رنگ آمیزی^{۳۹}
- رنگ آمیزی رتبه‌ای^{۴۰}
- رنگ آمیزی موزون^{۴۱}
- رنگ آمیزی بازه‌ای^{۴۲}
- رنگ آمیزی دوری^{۴۳}
- رنگ آمیزی مسیری^{۴۴}
- رنگ آمیزی لیستی^{۴۵}
- رنگ آمیزی مجموع رنگی^{۴۶}

³⁷Interval graph

³⁸Equitable coloring

³⁹T-coloring

⁴⁰Rank coloring

⁴¹Harmonious coloring

⁴²Interval coloring

⁴³Circular coloring

⁴⁴Path coloring

⁴⁵List coloring

⁴⁶Chromatic sum coloring

رنگ آمیزی منصفانه

تعریف ۱.۲.۱. اگر مجموعه رئوس گراف G را بتوان به k رده V_1, V_2, \dots, V_k طوری افراز کرد که اولاً هر V_i یک مجموعه مستقل باشد و ثانیاً برای هر $i \neq j$ داشته باشیم $||V_i| - |V_j|| \leq 1$ ، آنگاه رنگ آمیزی مذکور را منصفانه k -رنگ پذیر می‌نامیم و کوچک‌ترین عدد k که به ازای آن گراف G منصفانه k -رنگ پذیر است را عدد رنگی منصفانه نامیده شده و آنرا با نماد $\chi=(G)$ نمایش داده می‌شود.

T -رنگ آمیزی

تعریف ۲.۲.۱. گراف $G = (V, E)$ و T زیر مجموعه ای از اعداد صحیح نامنفی شامل صفر را در نظر بگیرید. در این صورت تابع $c : V \rightarrow \mathbb{Z}$ را یک T -رنگ آمیزی می‌نامیم هرگاه برای هر $\{u, v\} \in E$ داشته باشیم $|c(u) - c(v)| \notin T$. عدد T -رنگی گراف G برابر است با کوچکترین مقدار $|c(V(G))|$ است که در آن c یک T -رنگ آمیزی برای G است. بدیهی است که با تعریف فوق هر رنگ آمیزی راسی گراف، یک $\{0\}$ -رنگ آمیزی است و بالعکس هر T -رنگ آمیزی یک رنگ آمیزی راسی گراف است.

رنگ آمیزی رتبه‌ای

تعریف ۳.۲.۱. هر k -رتبه‌بندی برای گراف G عبارتست تابعی مثل $c : V \rightarrow \{1, \dots, k\}$ بطوری که هر مسیر بین هر دو راسی که دارای رنگ یکسان هستند، دارای حداقل یک راس با رنگ بیشتر باشد. $\chi_r(G)$ عبارتست از کمترین مقدار k برای اینکه گراف G دارای k -رتبه‌بندی باشد. بدیهی است که هر رتبه‌بندی با تعریف فوق یک رنگ آمیزی مجاز برای G خواهد بود. لذا $\chi_r(G) \leq \chi(G)$.

رنگ آمیزی موزون

تعریف ۴.۲.۱. k -رنگ آمیزی موزون برای گراف G یک k -رنگ آمیزی مجاز برای G است که در آن هیچ دو یالی رنگ یکسان نداشته باشند. رنگ یال $\{u, v\}$ بصورت زوج نامرتب $\{c(u), c(v)\}$ تعریف می‌شود. $\chi_h(G)$ عبارتست از کمترین مقدار k برای اینکه گراف G دارای k -رنگ آمیزی موزون باشد.

رنگ آمیزی بازه‌ای

تعریف ۵.۲.۱. زیر مجموعه متناهی $A \subseteq \mathbb{N}$ را یک بازه می‌گوییم، هرگاه شامل تمام اعداد بین $Min(A)$ و $Max(A)$ باشد.

یک رنگ آمیزی مجاز یالی از گراف G بازه ایست هرگاه برای هر راس v مجموعه رنگ های یال های مجاور راس v تشکیل بازه بدهند.

رنگ آمیزی دوری

تعریف ۶.۲.۱. فرض کنید k, d دو عدد صحیح مثبت باشند که $k \geq d$ در این صورت (k, d) -رنگ آمیزی تابعی مثل $c: V(G) \rightarrow \{1, \dots, k-1\}$ است بطوریکه برای هر $\{u, v\} \in E$ داشته باشیم

$$d \leq |c(v) - c(u)| \leq k - d$$

عدد رنگی دوری $\chi_c(G)$ برابر با کمترین مقدار کسر $\frac{k}{d}$ که برای (k, d) -رنگ آمیزی از گراف G است.

رنگ آمیزی مسیری

تعریف ۷.۲.۱. گراف G و مجموعه P شامل تمام مسیرهای گراف را در نظر بگیرید. یک k -رنگ آمیزی مجموعه P عبارتست از تابع $c: P \rightarrow \{1, 2, \dots, k\}$ بطوریکه هیچ دو مسیر متقاطع^{۴۸} رنگ یکسان نداشته باشند. در صورت وجود چنین تابعی می گوییم P, k -رنگ پذیر است. دو مسیر p_1 و p_2 متقاطع هستند هرگاه یال مشترکی داشته باشند. کوچک ترین میزان k که به ازای آن یک k -رنگ آمیزی برای P موجود باشد را عدد رنگی P می نامیم و آنرا با نماد $\chi_G(P)$ نمایش می دهیم.

رنگ آمیزی لیستی

تعریف ۸.۲.۱. برای گراف $G = (V, E)$ تابع تخصیص لیست^{۴۹} $L: V(G) \rightarrow 2^{\mathbb{N}}$ تعریف می شود، بطوریکه به هر راس v لیستی از رئوس مجاز، $L(v)$ ، اختصاص می دهد. اگر برای هر راس v داشته باشیم $|L(v)| = k$ ، آنگاه تابع L را k -تخصیص^{۵۰} می نامیم. اکنون برای هر تابع k -تخصیص L ، تابع رنگ آمیزی $c_L: V(G) \rightarrow \mathbb{N}$ را بصورتی تعریف می شود که به هر راس v عددی از لیست $L(v)$ اختصاص دهد، بطوری که نهایتاً رئوس مجاور رنگ یکسان نداشته باشند. تابع c_L با این ویژگی را یک L -رنگ آمیزی می نامیم. بعلاوه به کوچکترین عدد k که به ازای هر k -تخصیص L روی گراف G ، یک L -رنگ آمیزی برای گراف G وجود داشته باشد، عدد رنگی لیستی^{۵۱} گراف G می گوییم و آنرا با نماد $\chi_l(G)$ نمایش می دهیم.

⁴⁸Collining path

⁴⁹List assignment

⁵⁰k-assignment

⁵¹List chromatic number

۳.۱ رنگ آمیزی مجموع رنگی

این قسمت به تعریف مسئله رنگ آمیزی مجموع رنگی اختصاص دارد که موضوع اصلی این رساله است.

۴.۱ پیشینه مسئله

مفهوم مجموع رنگی، برای اولین بار در رساله‌ی دکتری کوبیکا^{۵۲} در سال ۱۹۸۷ میلادی معرفی شد. نتایج مهم ارائه شده در آن رساله، در سال ۱۹۸۹ در مقاله‌ی مشترک کوبیکا و شوئنک^{۵۳}، تحت عنوان مقدمه‌ای بر مجموع رنگی [۳۰] آورده شده است. در این مقاله ثابت شده است که مسأله‌ی محاسبه‌ی مجموع رنگی برای یک گراف دلخواه، NP -سخت است. همچنین آنها در این مقاله نشان دادند که برای هر عدد طبیعی $s \geq 2$ ، یک درخت با قدرت رأسی s و تعداد رئوس t_s وجود دارد جایی که:

$$t_s = \frac{1}{\sqrt{3}}((2 + \sqrt{2})^{s-1} - (2 - \sqrt{2})^{s-1})$$

در سال ۱۹۹۰، ارداش^{۵۴} و کوبیکا در مقاله‌ای مشترک [۱۳] نشان دادند که برای هر عدد صحیح $k \geq 2$ و هر عدد صحیح و مثبت $t \geq 1$ یک گراف با عدد رنگی k و قدرت رأسی حداقل $k + t$ وجود دارد و به این صورت نشان دادند که قدرت رأسی می‌تواند بسیار بزرگتر از عدد رنگی باشد. در همین مقاله آنها الگوریتمی خطی برای ساختن درختی با قدرت رأسی k ارائه کردند که ماکسیمم درجه‌ی آن درخت، تقریباً $\frac{k^2}{3}$ است.

از آن زمان به بعد مقالات متعددی در ارتباط با مجموع رنگی و قدرت رأسی گراف نگاشته شد که در فصل (۳) به تفصیل به این مقالات اشاره شده است. (در صورت علاقمندی به دانستن بیشتر در این خصوص به فصل مذکور مراجعه شود.)

تعریف ۱.۴.۱. گراف ساده $G = (V, E)$ به همراه رنگ آمیزی مجاز c از آن را در نظر بگیرید. رنگ آمیزی مجموع رأسی G که آنرا با علامت $\sum_c G$ نمایش می‌دهیم، به صورت زیر تعریف می‌شود:

$$\sum(G, c) = \sum_{v \in V} c(v)$$

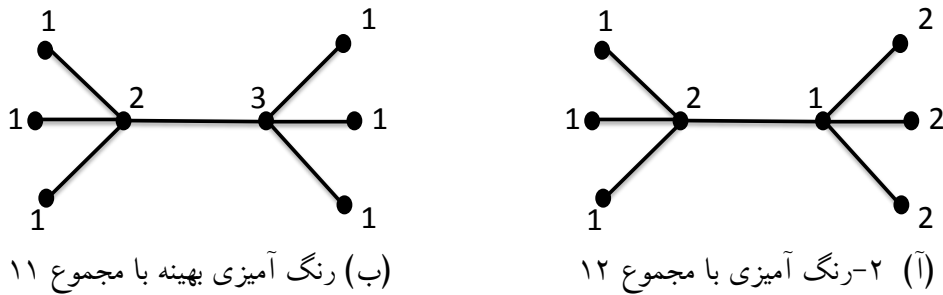
که در آن $c(v)$ یک عدد طبیعی است که از آن بعنوان رنگ راس v یاد می‌شود. مجموع رنگی رأسی گراف G که آنرا با نماد $\sum G$ نمایش می‌دهند برابر کمترین مقدار عبارت $\sum(G, c)$

⁵²Kubicka

⁵³Schwenk

⁵⁴Erdos

⁵⁵Chromatic Number



شکل ۲.۱: $\chi(G) = 2 < 3 = s(G)$

به ازای تمام رنگ آمیزی‌های مجاز c است. یعنی:

$$\sum G = \min_c \sum(G, c)$$

تعریفی مشابه برای مجموع رنگی یالی وجود دارد که بر پایه تعریف رنگ آمیزی یالی استوار است که موضوع مورد بحث ما نیست. در واقع مجموع رنگی یالی گراف G ، که با نماد $\sum' G$ نمایش داده می‌شود و براساس رنگ آمیزی یالی گراف G تعریف می‌شود، برابر است با مجموع رنگی راسی گراف خط G . یعنی

$$\sum' G = \sum L(G)$$

لذا تمرکز این رساله بر روی مفهوم رنگ آمیزی مجموع راسی است که از این به بعد از آن با عنوان رنگ آمیزی مجموع یاد می‌شود.

مشابه عدد رنگی گراف، $\chi(G)$ ، در اینجا هم پارامتری بنام قدرت راسی گراف^{۵۶} تعریف می‌شود که برابر است با کمترین تعداد رنگ برای رنگ آمیزی مثل c که به ازای آن داشته باشیم:

$$\sum(G) = \sum(G, c) \quad (۱.۱)$$

قدرت راسی G را با $s(G)$ نمایش داده می‌شود. همچنین به رنگ آمیزی‌هایی مثل c که در رابطه (۱.۱) صدق کند، رنگ آمیزی بهینه^{۵۷} گفته می‌شود. بدیهی است که $\chi(G) \leq s(G)$ زیرا $\chi(G)$ کمترین عددی است که با آن می‌توان گراف را رنگ کرد. در واقع نامساوی اخیر می‌تواند بصورت اکید باشد. اولین و معروف‌ترین مثالی که در این باره وجود دارد مثال موجود در شکل (۵.۵ب) است که این عدم تساوی را نمایش می‌دهد [۲۹].

⁵⁶Vertex strength

⁵⁷Best coloring

فصل ۲

اهمیت مسئله رنگ‌آمیزی مجموع و
پیچیدگی محاسباتی آن

گفته شد که در مسئله رنگ آمیزی مجموع، علاقمند به یافتن رنگ آمیزی مجازی هستیم که کمترین مقدار مجموع را داشته باشد. علیرغم اینکه این مسئله شباهت زیادی با مسئله رنگ آمیزی کلاسیک دارد، ولی نسبت به آن سخت تر به نظر می رسد. زیرا ممکن است افزایش تعداد رنگ های بکار رفته، موجب بهتر شدن مجموع رنگ ها شود و همین امر موجب می شود که فضای مربوط به رنگ آمیزی های مجاز برای این مسئله نسبت به رنگ آمیزی کلاسیک بسیار گسترده تر باشد. ممکن است انگیزه تعریف مفهوم رنگ آمیزی مجموع گراف، از ایده کاربرد آن در زمینه مدل های زمان بندی و تخصیص منابع^۱ آمده باشد. این مسئله در دنیای واقعی شاید دارای کاربردهای زیادی باشد ولی از میان آنها، موضوعاتی که بیشتر مورد توجه مقالات مختلف قرار گرفته است عبارتند از: زمان بندی و تخصیص منابع توزیع شده [۴، ۱۷، ۱۶، ۲۸، ۳۵] و همچنین طراحی VLSI مدار الکتریکی [۴۷، ۴۸، ۴۳]. این دو کاربرد مهم ما را با دو مفهوم در نظریه گراف درگیر می کند. در واقع نخستین کاربرد با مفهوم رنگ آمیزی چندگانه گراف ها و کاربرد دوم با مفهوم رنگ آمیزی گراف بازه ای مرتبط می شود.

به دلیل کاربردهای عملی جدی مسئله رنگ آمیزی مجموع در مدیریت صنایع، در این فصل ابتدا تشریح مهمترین کاربردهای آن اشاره شده و سپس طریقه یافتن جوابی معقول برای این مسئله ارائه می گردد. در این رابطه توجه به تکنیک های موجود در ریاضیات کاربردی و علوم کامپیوتر بیشتر نمایان می شود و این روش ها بیشتر در کانون توجهات قرار می گیرند. در این مسئله نیز، همانند سایر مسائل بهینه سازی در نظریه گراف، پرداختن به بحث بررسی پیچیدگی محاسباتی آن در حالت کلی و جزئی لازم به نظر می رسد.

بنابر توضیحات فوق در این فصل ابتدا به بررسی کاربردهای مهم این مسئله که عبارتند از زمان بندی و تخصیص منبع توزیع شده و طراحی VLSI مدار الکتریکی پرداخته شده است. سپس برای ورود به بحث پیچیدگی محاسباتی مسئله، به بیان مقدماتی از نظریه NP بسنده می شود. در ادامه ی بحث پیچیدگی محاسباتی برای اثبات NP-کامل بودن مسئله رنگ آمیزی مجموع روی گراف های ساده، دو اثبات مختلف ارائه می شود. در انتهای این فصل نیز به بحث پیچیدگی محاسباتی این مسئله روی گراف های خاص مثل درخت ها، گراف های بازه ای و... اشاره ای شده است.

۱.۲ برخی از کاربردها

قبل از پرداختن به کاربرد مهم مسئله رنگ آمیزی مجموع، لازم است نوعی از این مسئله با عنوان رنگ آمیزی چندگانه مجموع^۲ معرفی شود. در این نوع رنگ آمیزی رئوس می توانند بیش از یک رنگ دریافت کنند. رنگ آمیزی مجاز در این حالت به این صورت است که مجموعه های نسبت داده شده به رئوس مجاور باهم برخوردی^۳ نداشته باشند و از هم مجزا باشند.

تعریف ۱.۱.۲. [۳۵] منظور از رنگ آمیزی چندگانه مجموع برای گراف $G = (V, E)$ عبارت است

^۱Scheduling and resource allocation

^۲Sum multicoloring

^۳Conflict

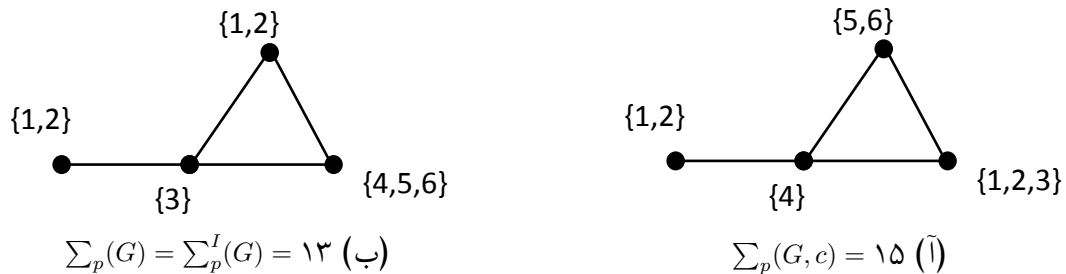
از تابعی مثل $c: V \rightarrow \mathbb{N}$ بطوری که برای هر یال $uv \in E$ داشته باشیم $c(u) \cap c(v) = \emptyset$. به علاوه رنگ آمیزی چندگانه فوق را پیوسته^۴ نامیده می‌شود هرگاه برای هر راس v ، مجموعه $c(v)$ یک بازه باشد به این معنی که داشته باشیم: $|c(v)| = \max c(v) - \min c(v) + 1$

اکنون تابع $p: V \rightarrow \mathbb{N}$ در نظر بگیرید. این تابع تعداد رنگ‌هایی است که باید برای هر راس v مورد استفاده قرار بگیرد. مشابه حالت معمولی مفهومی بنام مجموع رنگی چندگانه^۵ وجود دارد که بصورت زیر محاسبه می‌شود.

$$\sum_p G = \min_c \sum_p (G, c)$$

که در آن $\sum_p(G, c) = \sum_{v \in V} \max c(v)$ و c یک رنگ آمیزی چندگانه برای G است که در رابطه $|c(v)| = p(v)$ برای هر $v \in V$ صدق می‌کند. همچنین از نماد $\sum_p^I(G)$ برای نمایش مفهوم مجموع رنگ آمیزی چندگانه پیوسته استفاده می‌شود.

جهت توضیح بیشتر در این زمینه مثال شکل (۱.۲) ارائه می‌شود. در این شکل تابع p تعداد رنگ‌های مورد نیاز برای رئوس به ترتیب از چپ به راست برابر مقادیر ۲ و ۱ و ۲ و ۳ هستند. در شکل‌های (آ) و (ب) به ترتیب رنگ آمیزی مجموع چندگانه و بهترین رنگ آمیزی مجموع چندگانه نمایش داده شده‌اند. (خواننده علاقمند می‌تواند برای آگاهی بیشتر در مورد ویژگی‌های مفهوم مجموع رنگی چندگانه به [۳۶] مراجعه کند).



شکل ۱.۲: رنگ آمیزی مجموع چندگانه و رنگ آمیزی بهینه مجموع چندگانه

۱.۱.۲ زمانبندی و تخصیص منابع

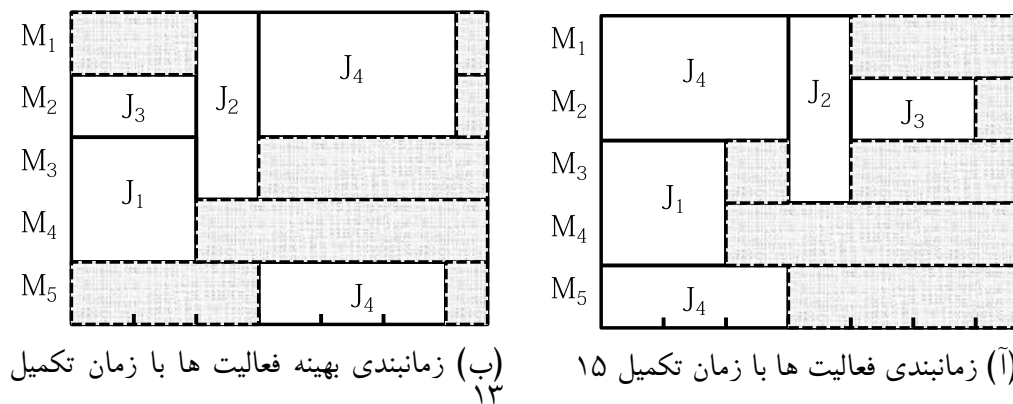
همانطور که قبلاً اشاره شد، یکی از کاربردهای مهم مسئله رنگ آمیزی مجموع در زمینه تخصیص منابع است. فرض کنید در یک سیستم، n فعالیت J_1, J_2, \dots, J_n وجود دارد. هر فعالیت J_i باید با یک پردازشگر خاص با دسترسی انحصاری به زیر مجموعه‌ای از پردازنده‌های M_1, \dots, M_m با زمان انجام p_i انجام شود. بدیهی است که دو فعالیتی که یک پردازنده را درگیر می‌کند نباید همزمان انجام شوند. هدف این مسئله زمانبندی تمام فعالیت‌هاست بطوریکه اولاً هیچ تداخل و همزمانی برای فعالیت‌ها

⁴Contiguous

⁵Multichromatic sum

اتفاق نیافتد و ثانيا میانگین زمان تکمیل فعالیت‌ها کمینه باشد. اکنون می‌خواهیم به بررسی ارتباط این مسئله با مسئله رنگ آمیزی چندگانه که در قسمت قبل بیان شد، بپردازیم. هنگام مدل کردن این مسئله، از ایده گراف برخوردار^۶ استفاده می‌شود که بصورت زیر قابل تعریف است. هر راس این گراف را متناظر با هر فعالیت در نظر بگیرید. به علاوه دو راس در این گراف با هم مجاورند، اگر و فقط اگر فعالیت‌های متناظر با هم تداخل داشته باشند (فعالیت‌هایی که یک پردازنده یکسان را درگیر می‌کنند). انجام هر فعالیت J_i که در یک بازه زمانی واحد $[i-1, i]$ اتفاق می‌افتد، معادل است با تخصیص رنگ i به راس J_i . بنابراین با این تعریف مسئله زمانبندی و تخصیص منابع را می‌توان با مسئله رنگ آمیزی مجموع چندگانه برای گراف برخوردار متناظر یکسان در نظر گرفت.

برای درک بیشتر در شکل (۲.۲) نمونه ای از سیستم توزیع یافته با ۴ فعالیت J_1, J_2, J_3, J_4 و ۵ پردازنده M_1, M_2, M_3, M_4, M_5 نمایش داده شده است. شکل (۲.۲) دو نوع زمانبندی برای انجام این فعالیت‌ها را نشان می‌دهد بطوریکه تداخلی اتفاق نیافتد. زمانبندی‌های نشان داده شده در شکل‌های (۲.۲) و (۱.۲) به ترتیب متناظر رنگ آمیزی‌های چندگانه پیوسته گراف برخوردار نشان داده شده در شکل‌های (۱.۲) و (۱.۲) است. به عنوان مثال اگر راس ۱ (فعالیت J_1) با مجموعه $\{1, 2\}$ رنگ شوند، این بدین معنی است که فعالیت J_1 در فاصله زمانی $[0, 2]$ انجام می‌شود.



شکل ۲.۲: نمونه ای از زمانبندی سیستم توزیع شده با ۴ فعالیت و ۵ پردازنده

۲.۱.۲ طراحی VLSI مدار الکترونیکی

از مهمترین کاربردهای دیگر مسئله رنگ آمیزی مجموع در طراحی VLSI به منظور مینیمم کردن طول مدار الکترونیکی است. در واقع مجموع رنگ‌ها در این مسئله معادل هزینه کل سیم بکار رفته در مدار است.

با توجه به شکل (۳.۲)، پایانه‌هایی الکترونیکی را در نظر بگیرید که در یک خط راست (سطح پایه)

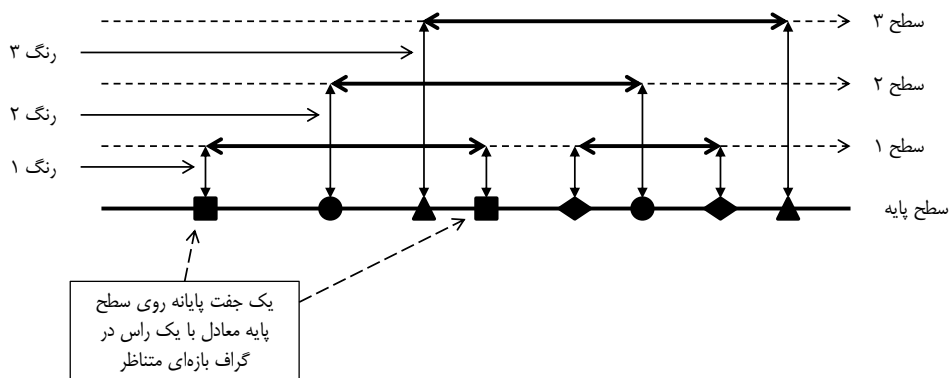
^۶Conflict graph

در مکان‌های مشخص قرار گرفته‌اند. پایانه‌های متصل به هم، یک شبکه را تشکیل می‌دهند. مکان پایانه‌ها در سطح پایه مشخص است. همچنین ۳ سطح، موازی با سطح پایه رسم شده است. فاصله سطوح از یکدیگر و فاصله اولین سطح از سطح پایه برابر واحد است. برای اتصال دو پایانه از یک سیم افقی و دو سیم عمودی استفاده شود بطوریکه اولاً هیچ دو زوج پایانه‌ای که سیم افقی بکار رفته در آنها در یک سطح قرار دارند نباید با هم همپوشانی داشته باشند و ثانیاً طول کل سیم بکار رفته مینیمم باشد. از آنجا که موقعیت پایانه‌ها در مدار (سطح پایه) ثابت و مشخص است، لذا برای مینیمم کردن سیم بکار رفته باید سراغ سیم‌های عمودی برویم.

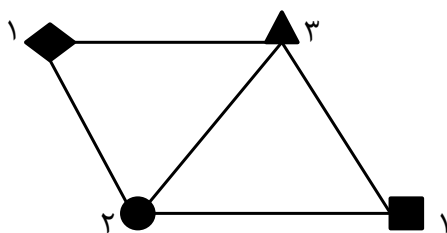
اکنون طراحی *VLSI* برای مدار الکتریکی، بصورت زیر با یک مسئله رنگ آمیزی مجموع روی گراف مرتبط می‌شود. مجموعه رئوس گراف G را برابر مجموعه همه شبکه‌های موجود در مدار الکتریکی در نظر بگیرید. بعلاوه دو راس از گراف G با هم مجاور اند هرگاه شبکه‌های متناظر با هم همپوشانی داشته باشند. در واقع هر شبکه روی سطح پایه یک فاصله را تداعی می‌کند، لذا گراف G که با تعریف فوق حاصل می‌شود یک گراف بازه‌ای است. بدیهی است که هر رنگ آمیزی مجاز گراف G معادل با یک طراحی موجه برای سیم کشی در مدار مذکور است. به عبارت دیگر رنگ هر راس مشخص می‌کند که سیم افقی مربوط به شبکه متناظر در چه سطحی قرار بگیرد. در واقع با هر رنگ آمیزی مجاز برای گراف بازه‌ای، طول سیم‌های عمودی مورد استفاده برای هر شبکه از مدار مشخص می‌شود. به عنوان مثال مدار نمایش داده شده در شکل (۳.۲) دارای چهار شبکه است که گراف بازه‌ای متناظر همراه با رنگ آمیزی بهینه آن در شکل (۴.۲) نمایش داده شده است. بنابراین مجموع رنگی این گراف برابر $\sum G = 7$ و طول کل سیم‌های عمودی بکار رفته برابر

$$2 \times \sum G = 2 \times 7 = 14$$

خواهد بود. جزئیات بیشتر در [۴۳، ۴۸] است.



شکل ۳.۲: مثالی از اتصالات بین پایانه‌ها در طراحی *VLSI*



شکل ۴.۲: گراف بازه‌ای متناظر با مثال شکل (۳.۲)

۲.۲ پیچیدگی محاسباتی

موضوع اصلی در این قسمت بررسی NP-کامل بودن مسئله MSC روی گراف‌های ساده بطور کلی و روی گراف‌های خاص بطور جزئی است. ابتدا برخی از تعاریف مقدماتی در مورد رده مسائل NP از جمله مسئله تصمیم‌گیری، الگوریتم غیر قطعی چند جمله‌ای، مفهوم کاهش پذیری مسائل تصمیم‌گیری بیان می‌شود. عمده توضیحات بیان شده در مورد این مفاهیم از مرجع [۴۰] اتخاذ شده است.

۱.۲.۲ مقدماتی بر نظریه NP

در بررسی فرایند شکل‌گیری مسائل بهینه‌سازی، دسته‌ای از مسائل وجود دارند که هنوز الگوریتمی کارآمد (چند جمله‌ای) برای حل آنها ابداع نشده است ولی غیر ممکن بودن آن هم نیز به اثبات نرسیده است. این گونه مسائل بهینه‌سازی را NP-کامل^۷ می‌گویند. برای توضیح بیشتر نظریه NP تمرکز خود را به حالت تصمیم‌گیری مسائل بهینه‌سازی معطوف می‌کنیم. یک مسئله تصمیم‌گیری^۸ مسئله‌ای است که خروجی آن پاسخ آری یا خیر است. به عنوان مثال مسئله تصمیم‌گیری رنگ آمیزی معمولی گراف (II) و مسئله تصمیم‌گیری رنگ آمیزی مجموع گراف (II') بصورت زیر هستند.

تعریف ۱.۲.۲.۱ مسئله II. به ازای یک عدد صحیح مثبت d ، آیا رنگ آمیزی مجازی برای گراف G وجود دارد بطوری که تعداد رنگ‌های بکار رفته در آن حداکثر d باشد؟

تعریف ۱.۲.۲.۲ مسئله II'. به ازای یک عدد صحیح مثبت d آیا رنگ آمیزی مجازی مثل c برای گراف G وجود دارد بطوری که $\sum(G, c) \leq d$ ؟

این مسائل دارای همان پارامترهای مسئله بهینه‌سازی مربوطه هستند به علاوه پارامتر اضافی d . هر مسئله بهینه‌سازی متناظر با یک مسئله تصمیم‌گیری است. واضح است که اگر یک الگوریتم زمان چند جمله‌ای برای مسئله بهینه‌سازی وجود داشته باشد، همان الگوریتم می‌تواند مسئله تصمیم‌گیری متناظر را حل کند. حتی ارتباط یک مسئله بهینه‌سازی با مسئله تصمیم‌گیری متناظرش از این هم بیشتر است. در واقع برای بسیاری از مسایل تصمیم‌گیری نشان داده شده است که از یک الگوریتم

⁷NP-complete

⁸Decision problem

زمان چندجمله‌ای برای مسئله تصمیم‌گیری می‌توان به یک الگوریتم زمان چند جمله‌ای برای مسئله بهینه‌سازی مربوطه دست پیدا کرد. برای همین اغلب توضیحات مربوط به نظریه NP روی مسائل تصمیم‌گیری بنا می‌شوند.

مجموعه مسائل تصمیم‌گیری که برای آنها حل چندجمله‌ای وجود دارد به مجموعه P تعلق دارند یا عبارتی مجموعه P شامل تمامی مسائل تصمیم‌گیری است که توسط الگوریتم‌های زمان چندجمله‌ای قابل حل هستند. برای اینکه بگوییم یک مسئله تصمیم‌گیری به رده P تعلق ندارد باید ثابت کنیم که طراحی یک الگوریتم زمان چند جمله‌ای برای آن ممکن نیست. برای مسائل Π و Π' این کار انجام نشده است.

برای اینکه بتوانیم تعریفی برای مجموعه NP ارائه دهیم باید از مفهومی بنام الگوریتم غیر قطعی^۹ سخن بگوییم که متشکل از دو مرحله جداگانه ذیل است:

- **مرحله حدس زدن (غیر قطعی):** با داشتن نمونه‌ای از مسئله، این مرحله صرفاً یک رشته S را تولید می‌کند. رشته را می‌توان به عنوان یک حل حدسی برای نمونه در نظر گرفت. حتی می‌تواند رشته کاملاً بی ربط باشد.

- **مرحله تصدیق کردن (قطعی):** نمونه و رشته S ورودی این مرحله هستند. این مرحله به شیوه‌های قطعی ادامه می‌یابد تا سرانجام یکی از سه حالت زیر رخ دهد: الف) به ازای ورودی S خروجی آری باشد. ب) به ازای ورودی S خروجی خیر باشد. ج) اصلاً پایان نیابد.

الگوریتم غیر قطعی زمان چندجمله‌ای^{۱۰} یک الگوریتم غیرقطعی است که مرحله تصدیق آن زمان چند جمله‌ای باشد. اکنون با این مقدمات تعریف مجموعه NP به صورت زیر خواهد بود.

تعریف ۳.۲.۲. مجموعه NP ^{۱۱} مجموعه تمامی مسائل تصمیم‌گیری است که توسط الگوریتم‌های غیرقطعی زمان چند جمله‌ای قابل حل هستند.

بنابراین برای آنکه یک مسئله تصمیم‌گیری در NP باشد باید الگوریتمی وجود داشته باشد که عمل تصدیق را در زمان چند جمله‌ای انجام دهد. البته تأکید بر این نکته ضروری است که تعریف فوق بدان معنا نیست که الزاماً الگوریتمی با زمان چندجمله‌ای برای یک مسئله NP وجود دارد بلکه مرحله تصدیق در زمان چند جمله‌ای انجام می‌گیرد. به عنوان مثال برای مسئله Π' مرحله تصدیق به این معنی که آیا اولاً رنگ آمیزی داده شده c برای گراف G مجاز است؟ (بدیهی است که الگوریتمی با زمان چندجمله‌ای برای این تصدیق این قسمت وجود دارد) و ثانیاً در صورت مجاز بودن c آیا برای عدد داده شده d ، رابطه $\sum(G, c) \leq d$ برقرار است؟ (برای این قسمت نیز الگوریتم با زمان چند جمله‌ای وجود دارد) در زمان چندجمله‌ای انجام می‌گیرد لذا مسئله Π' به مجموعه NP تعلق دارد. برخی از مسائل تصمیم‌گیری هستند که به لحاظ سختی ظاهراً با هم متفاوت‌اند ولی ثابت می‌شوند که

⁹Non-deterministic algorithm

¹⁰Non-deterministic polynomial time algorithm

¹¹Non-deterministic Polynomial

حل یکی از آنها منجر به حل هزاران مسئله تصمیم‌گیری دیگر می‌شود. به عبارت دیگر اگر یکی از آنها در مجموعه P قرار داشته باشد، آنگاه همگی باید در P قرار داشته باشند. اینجاست که مفهومی بنام کاهش پذیری چند جمله‌ای^{۱۲} مطرح می‌شود. این گونه مسائل را مسائل NP -کامل می‌گویند که در ادامه به صورت خلاصه به ارائه تعریف آنها می‌پردازیم.

تعریف ۴.۲.۲. کاهش پذیری: دو مسئله تصمیم‌گیری A و B را در نظر بگیرید. در این صورت می‌گوییم مسئله A به مسئله B کاهش پذیر چند جمله‌ای^{۱۳} است و آنرا با نماد $A \times B$ نمایش می‌دهیم هرگاه یک الگوریتم تبدیل با زمان چند جمله‌ای وجود داشته باشد بطوری که هر نمونه مسئله از A را به یک نمونه از B تبدیل کند.

گزاره زیر از توضیحات فوق براحتی قابل استنتاج است.

گزاره ۱.۲.۲. اگر مسئله تصمیم‌گیری B در P باشد و داشته باشیم $A \times B$ آنگاه مسئله تصمیم‌گیری A نیز در P خواهد بود.

اکنون به تعریف مسئله NP -کامل می‌پردازیم.

تعریف ۵.۲.۲. مجموعه NP -کامل: مسئله B را مجموعه NP -کامل می‌گوییم هرگاه هر دو شرط زیر برقرار باشند:

(۱) B در NP باشد

(۲) به ازای هر مسئله دیگر A در NP داشته باشیم $A \times B$.

بنابر تعریف فوق برای اینکه بگوییم $P \neq NP$ ، از آنجایی که $P \subseteq NP$ لذا کافی است نشان دهیم مسئله‌ای در NP هست که در P نیست. یعنی الگوریتمی با زمان چند جمله‌ای برای آن وجود ندارد. همچنین برای اینکه نشان دهیم $P = NP$ فقط کافی است نشان دهیم حداقل یک مسئله NP -کامل هست که در P قرار دارد. یعنی برای آن الگوریتم با زمان چند جمله‌ای وجود دارد. این موضوع از شرط دوم که در تعریف مسائل NP -کامل آمده، قابل نتیجه‌گیری است. برای اثبات NP -کامل بودن یک مسئله محقق شدن شرط دوم تعریف (۵.۲.۲) برای تمام مسائل NP امری نشدنی است. در سال ۱۹۷۱ استفان کوک^{۱۴} توانست مسئله‌ای بیابد که NP -کامل باشد. این مسئله همان مسئله معروف CNF -صدق پذیری^{۱۵} است که در اثبات آن از خواص متداول مسائل NP استفاده شده است. وی نشان داد که هر مسئله‌ای که در NP قرار دارد باید به مسئله CNF -صدق پذیری کاهش یابد. هنگامی که این قضیه بنیادین ثابت شد، NP کامل بودن بسیاری از مسائل دیگر نیز به اثبات رسید. این اثبات‌ها بر گزاره زیر متکی هستند.

¹²Reduction

¹³Polynomial-time reducible

¹⁴Stephen Cook

¹⁵CNF-satisfiability (CNF-SAT)

گزاره ۲.۲.۲. مسئله C یک مسئله NP -کامل است، هرگاه

(۱) در NP باشد

(۲) حداقل یک مسئله NP -کامل مثل B وجود داشته باشد بطوریکه $B \propto C$.

□

برهان. اثبات با توجه به تعریف (۵.۲.۲) بدیهی است.

۲.۲.۲ NP -کامل بودن مسئله تصمیم‌گیری رنگ‌آمیزی مجموع

اکنون بنا به توضیحات بخش قبل این آمادگی وجود دارد که در مورد NP -کامل بودن مسئله تصمیم‌گیری رنگ‌آمیزی مجموع (مسئله Π') بحث شود. در این قسمت دو اثبات برای NP -کامل بودن مسئله Π' ارائه می‌شود تا با مفهوم کاهش‌پذیری بیشتر آشنا شویم. اثبات اول در سال ۱۹۸۷ ارائه شد، مبتنی بر کاهش‌پذیری مسئله تصمیم‌گیری رنگ‌آمیزی معمولی (Π) به مسئله Π' است. اثبات دوم در سال ۲۰۰۲ مبتنی بر کاهش‌پذیری حالت خاصی از مسئله تصمیم‌گیری تطابق ۳-بعدي ($3DM^*$) به مسئله Π' است.

با فرض NP -کامل بودن مسائل Π و $3DM^*$ و بنابر گزاره (۲.۲.۲) کافی است ثابت کنیم که اولاً مسائل فوق به رده NP تعلق دارند و ثانياً این مسائل به مسئله Π' قابل کاهش با زمان چند جمله‌ای هستند.

قضیه ۱.۲.۲. [۳۰] مسئله تصمیم‌گیری Π' ، یک مسئله NP -کامل است.

برهان. (اثبات اول) [۳۰]

بدیهی است که $\Pi' \in NP$. در واقع یک الگوریتم با زمان چند جمله‌ای وجود دارد که مرحله تصدیق کردن (یعنی مجاز بودن رنگ‌آمیزی به همراه نامساوی مربوطه) را مورد بررسی قرار دهد. اکنون بنابر توضیحات فوق کافی است ثابت کنیم که $\Pi \propto \Pi'$. در واقع باید نشان دهیم هر نمونه مسئله Π را می‌توان با یک الگوریتم تبدیل با زمان چند جمله‌ای به یک نمونه مسئله تصمیم‌گیری Π' تبدیل کرد. برای این منظور گراف $G = (V, E)$ به همراه عدد d دلخواه را در نظر بگیرید، بطوریکه d -رنگ پذیر باشد. اکنون باید گرافی مثل $G' = (V', E')$ به همراه عدد صحیح d' و رنگ‌آمیزی مجاز c' معرفی کنیم، بطوریکه $\sum(G', c') \leq d'$ اگر و فقط اگر G ، d -رنگ پذیر باشد. قرار دهید $G' = G \times K_d$ حاصلضرب دکارتی G با گراف کامل d راسی. همچنین قرار دهید $d' = n \frac{d(d+1)}{4}$ که در آن n اندازه G است. اکنون رنگ‌آمیزی c' برای گراف G' را از روی c بصورت زیر در نظر می‌گیریم. رنگ اولین کپی G را همان رنگ G در نظر می‌گیریم. برای دومین کپی G رنگ رئوس یک واحد بصورت دوری شیف‌ت می‌دهیم. یعنی اگر $1, 2, \dots, r$ رنگ رئوس اولین کپی باشند آنگاه $1, 2, \dots, r$ رنگ دومین کپی G خواهد بود. همین فرایند را تا d -امین کپی G ادامه می‌دهیم تا همه رئوس G' رنگ شوند. بدیهی است که c' یک رنگ‌آمیزی مجاز برای G' است و بعلاوه داریم

$$\sum(G', c') \leq n^{\frac{d(d+1)}{4}} = d'$$

برعکس فرض کنید رنگ آمیزی مجازی مثل c' برای G' وجود دارد بطوریکه $\sum(G', c') \leq d'$. چون G' شامل n کپی از K_d است و هر کدام از آنها دارای حداقل مجموع رنگ $\frac{d(d+1)}{4}$ است که با استفاده از رنگ‌های $1, 2, \dots, d$ بدست می‌آید، لذا داریم $d' \leq \sum(G', c')$ بنابراین برای هر رنگ آمیزی مجاز c' داریم $\sum(G', c') = d'$. لذا برای دستیابی به چنین مجموع رنگی مجبوریم هر کپی K_d از G' را تنها با رنگ‌های $1, 2, \dots, d$ رنگ آمیزی کنیم. لذا نتیجه می‌شود که d -رنگ پذیر است و بنابراین G (زیر گراف G') نیز d -رنگ پذیر است. \square

لم ۱.۲.۲. [۳۵] مسئله تصمیم‌گیری زیر که حالت خاصی از مسئله تصمیم‌گیری تطابق ۳-بعدی است، یک مسئله NP -کامل است:

$3DM^*$: چهار مجموعه مجزای W, Y, X, M و عدد d را طوری در نظر بگیرید که داشته باشیم $|X| = |Y| = |W| = d$ و $M \subseteq W \times Y \times X$. علاوه اگر (w, y, x) و (w, y', x) و (w', y, x) اعضای M باشند، آنگاه $(w, y, x) \in M$. آیا $M' \subseteq M$ وجود دارد، بطوریکه $|M'| = d$ و همه‌ی مولفه‌های هر دو عضو $m_1, m_2 \in M'$ با هم متفاوت باشند؟

برهان. (اثبات دوم) [۳۵]

فرض کنید W, Y, X, M و عدد d در فرضیات لم (۱.۲.۲) صدق کند. قرار دهید: $d' = \frac{3d(d+1)}{4}$. اکنون G را بصورت زیر می‌سازیم. $V(G) = W \cup Y \cup X$ و $E(G) = E_1 \cup E_2$ که در آن $E_1 = \{(a, b) : a \neq b \wedge (a, b \in W \vee a, b \in Y \vee a, b \in X)\}$ و E_2 برابر است با مجموعه تمام $\{a, b\}$ هایی که در آن a و b مولفه هیچ سه‌تایی m از M نیست (یعنی a و b تواما در هیچ مولفه‌ای از اعضای M ظاهر نشده باشد).

از تعریف مسئله $3DM^*$ نتیجه می‌شود هر سه عنصری $w \in W, y \in Y, x \in X$ که یک مجموعه مستقل در G باشند، مولفه‌های عنصری مثل $m \in M$ هستند (یعنی $(w, y, x) \in M$).

اکنون فرض کنید $M' = \{m_1, \dots, m_d\}$ یک جواب موجه برای مسئله $3MD^*$ باشد. برای هر $w, x, y \in M_i$ قرار دهید: $c(w) = c(y) = c(x) = i$. با یک بررسی ساده می‌توان نشان داد که c یک رنگ آمیزی مجاز برای G است و بعلاوه داریم: $\sum(G, c) \leq d'$.

برعکس فرض کنید رنگ آمیزی c برای G وجود دارد بطوریکه $\sum(G, c) \leq d'$ چون

$$G[w] \simeq G[Y] \simeq G[X] \simeq K_d$$

لذا تعداد رنگ‌ها حداقل d است. بنابراین $\sum(G, c) = d'$. اکنون متناظر با هر سه عنصر $w, y, x \in M$ که یک مجموعه مستقل در G تشکیل دهند، عنصری مثل $m \in M$ برابر (w, y, x) وجود دارد. حال مجموعه $M' \subseteq M$ را اجتماع این m_i ها قرار دهید. بدیهی است که $|M'| = d$ و همه‌ی مولفه‌های هر دو عضو $m_1, m_2 \in M'$ با هم متفاوت هستند. بدیهی است که این تبدیل در زمان چند جمله‌ای قابل انجام است. \square

جدول ۱.۲: گراف‌هایی که برای آنها مسئله رنگ آمیزی مجموع یک مسئله NP -سخت است

مرجع	رده گراف
[۳۰]	دلخواه
[۳۴، ۳۵]	r -منتظم ($r \geq 3$)
[۱۷]	سطح
[۳۴، ۳۵]	مکعبی سطح
[۴۸]	بازه‌ای
[۱۴، ۳۵]	دوبخشی با $\Delta \leq 5$

۳.۲.۲ مسئله رنگ‌آمیزی مجموع روی گراف‌های خاص

در مقایسه با مسئله رنگ آمیزی معمولی، مسئله رنگ آمیزی مجموع روی گراف‌های مکعبی سطح یک مسئله NP -سخت است [۳۴]. جدول (۱.۲) شامل رده‌های مختلف گراف‌هایی است که برای آنها مسئله رنگ آمیزی مجموع یک مسئله NP -سخت است.

برخلاف گراف‌های موجود در جدول (۱.۲)، گراف‌هایی وجود دارند که محاسبه مجموع رنگی برای آنها بصورت دقیق در زمان چند جمله‌ای محاسبه می‌کند. قضیه زیر به چند دسته از این گراف‌ها می‌پردازد.

قضیه ۲.۲.۲. فرمول‌های زیر برای محاسبه رنگ آمیزی مجموع گراف‌های بیان شده، برقرار است:

$$\sum P_n = \lfloor \frac{r_n}{3} \rfloor \quad (۱)$$

$$\sum C_n = \lceil \frac{r_n}{3} \rceil \quad \text{برای } n \geq 3 \quad (۲)$$

$$\sum W_n = \begin{cases} \frac{2(n+1)}{3} : & \text{فرد } n \\ \frac{r_n}{3} + 4 : & \text{زوج } n \end{cases} \quad (۳)$$

$$\sum K_n = \frac{n(n+1)}{3} \quad (۴)$$

$$\sum K_{r,s} = r + s + \min\{r, s\} \quad (۵)$$

$$\sum B_k = \frac{r_n}{3} \quad (B_k \text{ گراف دوبخشی } k\text{-منتظم}) \quad (۶)$$

دسته‌ای از گراف‌های دوبخشی وجود دارد بطوریکه برای آنها مجموع رنگی با حداکثر دو رنگ اتفاق می‌افتد.

قضیه ۳.۲.۲. [۳۵] فرض کنید $G = (V_1, V_2; E)$ یک گراف دو بخشی باشد و همچنین تطابقی مثل A وجود داشته باشد بطوری که مجموعه V_1 را پوشاند. در این صورت هر رنگ آمیزی بهینه برای G حداکثر شامل دو رنگ است و بعلاوه $\sum G = n(G) + |V_1|$.

برهان. فرض کنید $V_1^A \subset V_2$ مجموعه رئوسی باشد که توسط A پوشانده می‌شوند. می‌دانیم که $|V_2| \geq |V_1|$. اکنون فرض کنید c رنگ آمیزی بهینه‌ای باشد که از حداقل سه رنگ استفاده می‌کند (فرض خلف). بنابراین

$$\begin{aligned} \sum(G, c) &= \sum_{\{v_1, v_2\} \in A} (c(v_1) + c(v_2)) + \sum_{v \in V_2 \setminus V_1^A} c(v) \\ &> 2|V_1| + |V_2| - |V_1| = n(G) + |V_1| \end{aligned} \quad (۱.۲)$$

ولی از طرفی بدیهی است که $2 -$ رنگ آمیزی با مجموع $n(G) + |V_1|$ وجود دارد که در تناقض با رابطه (۱.۲) است. \square

یکی دیگر از دسته گراف‌هایی که به ازای آنها برای مسئله رنگ آمیزی مجموع الگوریتم با زمان چند جمله‌ای وجود دارد، گراف‌های دوبخشی زیر مکعبی^{۱۷} هستند ($\Delta \leq 3$).

قضیه ۴.۲.۲. [۳۵، ۱۴] مسئله رنگ آمیزی مجموع روی گراف‌های دوبخشی مکعبی در زمان $O(n^2)$ قابل حل است.

برای گراف‌هایی با $\Delta \leq 2$ یافتن مجموع رنگی با الگوریتم زمان چند جمله‌ای امکان پذیر است [۳۴]، ولی همانطور که قبلاً بیان شد رنگ آمیزی مجموع روی گراف‌های دوبخشی با $\Delta \leq 5$ ، NP -سخت است و پیچیدگی همین مسئله روی گراف‌های دوبخشی با $\Delta \leq 4$ هنوز مشخص نشده است [؟]. دسته گراف‌های بسیاری وجود دارد که برای آنها فرمول مشخصی جهت محاسبه مجموع رنگی وجود ندارد. بعنوان مثال درخت‌ها دسته‌ای از این گراف‌ها هستند.

قضیه ۵.۲.۲. درخت T با $n \geq 1$ راس را در نظر بگیرید. در این صورت داریم:

$$n + 1 \leq \sum T \leq \lfloor \frac{3n}{2} \rfloor$$

بعلاوه برای هر عدد مثبت $\lfloor \frac{3n}{2} \rfloor \leq k \leq n + 1$ ، درختی با n راس و مجموع رنگی برابر با k وجود دارد.

برهان. [۳۰]

در رنگ آمیزی درختان برای محاسبه‌ی $\Sigma(T)$ حداقل دو رنگ برای رنگ آمیزی استفاده می‌شود لذا $s(T) \geq \chi(T) = 2$. بنابراین اگر T دارای $n \geq 2$ راس باشد، حداقل یک رأس از رنگ ۲ می‌بایست

¹⁷Subcubic Bipartite

که موجود باشد. لذا در این رنگ آمیزی، حداکثر $n - 1$ رأس از رنگ ۱ و حداقل یک رأس از رنگ ۲ موجود است و این ایجاب می کند

$$\Sigma(T) \geq (n - 1)(1) + 1(2) = n + 1$$

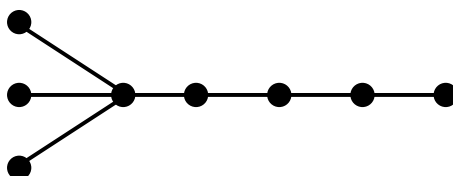
به عنوان مثال برای درخت ستاره‌ی $k, n-1$ ، کران پایین قضیه به دست می آید. از آنجایی که یک درخت، گرافی است دوبخشی لذا می توان آنرا را با رنگ های $\{1, 2\}$ به طور مجاز، رنگ آمیزی نمود. حال رئوس حاضر در بخش بزرگتر از نظر تعداد رأس را با رنگ ۱ و بقیه رئوس در بخش دیگر را با رنگ ۲ رنگ آمیزی می کنیم. بنابراین اگر $\chi(T)$ عدد رنگی، و $s(T)$ قدرت رأسی درخت باشند داریم

$$\Sigma_{s(T)}(T) \leq \Sigma_{\chi(T)}(T) \leq 1 \lceil \frac{n}{2} \rceil + 2 \lfloor \frac{n}{2} \rfloor = \lfloor \frac{3n}{2} \rfloor$$

لذا برای هر درخت $n \geq 2$ خواهیم داشت

$$n + 1 \leq \Sigma(T) \leq \lfloor \frac{3n}{2} \rfloor$$

برای قسمت دوم درخت $B(m, b)$ را بصورت زیر در نظر بگیرید. یک راس که از یک طرف به m یال انتهایی وصل شده است و از طرف دیگر به یک مسیر با طول $b - 1$ متصل است. برای مثال شکل (۵.۲) را ملاحظه کنید. از طرفی می توان فرض کرد برای یک صحیح نامنفی داریم: $k = n + 1 + r$



شکل ۵.۲: درخت $B(3, 5)$

. به سادگی می توان تحقیق کرد که :

$$\sum B(n - 2r - 1, 2r + 1) = k$$

□

قضیه ۶.۲.۲. [۳۰] الگوریتمی با مرتبه زمانی خطی ($O(n)$) برای حل مسئله رنگ آمیزی مجموع روی درختها وجود دارد.

فصل ۳

کران مجموع رنگی

همانطور که گفته شد، مسئله رنگ آمیزی مجموع، یک مسئله NP -کامل است. لذا حل دقیق آن در زمان چند جمله‌ای به دشواری سوال بسیار مشکل برابری یا عدم برابری مجموعه‌های P و NP در علوم کامپیوتر است. از طرفی این موضوع به دلیل کاربرد و همچنین ارتباط پارامترهای آن با سایر پارامترها و مفاهیم موجود در گراف، دارای اهمیت فراوان است. لذا هنگام بررسی مقالاتی که به مطالعه این موضوع پرداخته‌اند و اغلب متعلق به دو دهه اخیر هستند، دو رویکرد کلی ملاحظه می‌شود.

رویکرد اول مربوط به مقالاتی است که به مفهوم مجموع رنگی و قدرت گراف مانند دو پارامتر همانند دیگر پارامترهای گراف می‌نگرند و سعی در تخمین زدن و یافتن کران برای آنها هستند، فارغ از اینکه چه گرافی بررسی می‌شود (کران تحلیلی^۱). رویکرد دوم که بیشتر دید حل مسئله در دنیای واقعی دارد، مربوط به مقالاتی است که این دو پارامتر را برای هر گراف بصورت جداگانه تخمین می‌زند (کران الگوریتمی^۲). در حقیقت در رویکرد اول کران بدست آمده برای مجموع رنگی یا قدرت گراف به نمونه مسئله مورد بررسی وابستگی ندارد و مستقل از آن است ولی در رویکرد دوم این وابستگی بسیار مشهود است. در این فصل سعی شده‌است به هر دو رویکرد در مواجهه با مسئله رنگ آمیزی مجموع در حد امکان پرداخته می‌شود.

لازم به ذکر است در این قسمت به بیان اثبات قضایایی که طولانی و مفصل‌تر هستند، پرداخته نشده است. برای بررسی جزئیات اثبات این قضایا، به مرجع مربوطه مراجعه شود.

۱.۳ کران تحلیلی

رنگ آمیزی بهینه c از گراف G را در نظر بگیرید که در آن $s = \max\{c(v_i) : 1 \leq i \leq n\}$ بدیهی است که متناظر با این رنگ آمیزی افراز C_1, C_2, \dots, C_s از رئوس G وجود دارد بطوری که $C_i = c^{-1}(\{i\})$. در این صورت

$$\sum G = \sum (G, c) = \sum_{i=1}^s i \cdot |C_i|$$

با نماد گذاری فوق، گزاره زیر خصوصیت‌های مهم رنگ آمیزی بهینه c را بیان می‌کند.

گزاره ۱.۱.۳ [۱۵] رنگ آمیزی بهینه c از گراف n راسی G دارای دو ویژگی زیر است:

$$|C_1| \geq |C_2| \geq \dots \geq |C_s| \quad (\text{آ})$$

ب) برای هر دو رنگ $i < j$ و هر راس $v \in C_j$ راسی مثل $w \in C_i$ وجود دارد بطوریکه با v مجاور است.

$$\sum (G \setminus C_1) = \sum G - n \quad (\text{ج})$$

¹Theoretical Bound

²Algorithmic Bound

$$s(G \setminus C_1) = s(G) - 1 \quad (د)$$

اکنون چند کران برای مجموع رنگی گراف دلخواه مورد بررسی قرار می‌گیرد بطوریکه در آنها پارامترهای مختلفی از گراف از جمله تعداد راس‌ها (n) ، تعداد یال‌ها (m) ، قدرت گراف، عدد رنگی، عدد استقلال مورد نظر است.

قضیه ۱.۱.۳ [۲۸] هر گراف G دارای کران‌های زیر برای مجموع رنگی است:

$$n \leq \sum(G) \leq m + n \quad (۱)$$

$$\lceil \sqrt{\lambda m} \rceil \leq \sum(G) \leq \lfloor \frac{2(m+1)}{3} \rfloor \quad (۲)$$

$$s(G) \frac{(s(G)+1)}{4} \leq \sum(G) \leq n \frac{\chi(G)+1}{4} \quad (۳)$$

$$p = \lfloor \frac{n}{\alpha(G)} \rfloor \quad \text{که در آن } (p+1)(n - p \frac{\alpha(G)}{4}) \leq \sum(G) \quad (۴)$$

رابطه فوق برای گراف‌های دو بخشی کامل و گراف کامل بصورت تساوی برقرار است.

کران برای $s(G)$ نیز نقش مهمی در طراحی الگوریتم دقیق برای حل مسئله رنگ آمیزی مجموع دارد. بعنوان مثال با استفاده از قسمت سوم قضیه (۱.۱.۳) بدست می‌آید:

نتیجه ۱.۱.۳ برای هر گراف G

$$s(G) \leq \lfloor \sqrt{n \cdot (\chi(G) + 1)} \rfloor$$

همچنین برای قدرت راسی گراف مشابه قضیه بروکس، قضیه زیر وجود دارد:

قضیه ۲.۱.۳ [۱۵] برای هر گراف همبند G

$$s(G) \leq \Delta(G) + 1 \quad (۱.۳)$$

بعلاوه تنها برای گراف‌های کامل و دور فرد رابطه زیر برقرار است

$$s(G) = \Delta(G) + 1$$

کران بالای دیگری برای $s(G)$ در [۱۵] معرفی شده که بر اساس عدد رنگ آمیزی است. این مفهوم برای اولین بار در سال ۱۹۶۶ معرفی شد [۱۲].

تعریف ۱.۱.۳ فرض کنید G یک گراف باشد و $V(G) = \{v_1, v_2, \dots, v_n\}$ ، عدد رنگ آمیزی G که با نماد $col(G)$ نمایش داده می‌شود، به صورت زیر تعریف می‌گردد:

$$col(G) = \min_{\pi} \max_i \{d_{G_{\pi(i)}}(v_{\pi(i)})\}$$

که در آن، π جایگشت روی رؤس G است و $G_{\pi(i)}$ ، زیرگراف القایی از G است که توسط رأس‌های $v_{\pi(1)}, v_{\pi(2)}, \dots, v_{\pi(i)}$ القا می‌شود.

به عبارت ساده‌تر می‌توان گفت که عدد رنگ‌آمیزی گراف G برابر است با کوچکترین عدد طبیعی مانند d ، به طوری که برای هر رابطه‌ی ترتیب خطی $<$ ، از مجموعه رؤس گراف G ، درجه‌ی برگشتی هر رأس u از گراف G ، یعنی $|\{v : v < u, uv \in E(G)\}|$ ، اکیداً کمتر از d باشد. منظور از درجه‌ی برگشتی رأس u ، تعداد رؤس مجاور با آن که در رابطه‌ی ترتیب خطی صدق می‌کنند، می‌باشد. پارامتر دیگری که به طور غیرمنتظره‌ای با $col(G)$ ارتباط دارد عدد زکرس-ویلف^۳ است که در سال ۱۹۶۸ معرفی شده است و آنرا با $\omega(G)$ نمایش می‌دهند. دو سال بعد از معرفی $\omega(G)$ ، اثبات شد که برای هر گراف دلخواه G ، $\omega(G)$ و $col(G)$ با هم برابرند.

قضیه ۳.۱.۳. [۳۲] برای هر گراف دلخواه G داریم :

$$col(G) = 1 + \max_{H \subseteq G} \delta(H) \quad (۲.۲)$$

که در آن H زیرگراف القایی از G است.

با توجه به (۲.۲) کاملاً واضح است که $col(G) \leq \Delta(G) + 1$. بعلاوه داریم $\chi(G) \leq col(G)$. زیرا بنا به تعریف عدد رنگ‌آمیزی بدیهی است که هر گراف G را می‌توان با $col(G)$ تعداد رنگ بصورت موجه رنگ‌آمیزی کرد. این ایده‌ی رنگ‌آمیزی در واقع همان ایده‌ی رنگ‌آمیزی برای یافتن کران بالای مناسب برای $\chi(G)$ می‌باشد. اگر رأس‌های گراف را به ترتیبی دلخواه در نظر بگیریم کنیم، با توجه به اینکه هر رأس گراف مانند v ، حداکثر با $\Delta(G)$ رأس مجاور است، پس هنگام رنگ‌آمیزی v ، حداقل یک رنگ در اختیار خواهیم داشت که در همسایگی رأس v ظاهر نشده است و بنابراین می‌توان با $\Delta(G) + 1$ رنگ، یک رنگ‌آمیزی مجاز برای گراف G ارائه نمود. لذا نامساوی $\chi(G) \leq col(G)$ بهبودیافته‌ی رابطه‌ی $\chi(G) \leq \Delta(G) + 1$ است.

همان‌طور که قبلاً ذکر شد، برای هر گراف G داریم $\chi(G) \leq col(G)$. اما نامساوی $s(G) \leq col(G)$ همواره برقرار نیست. به عنوان مثال، تقریباً اغلب درخت‌ها قدرتی بیشتر از ۲ دارند، درحالی که عدد رنگ‌آمیزی آنها همواره ۲ است.

قضیه ۴.۱.۳. [۱۵] برای هر گراف دلخواه G داریم :

$$s(G) \leq \left\lceil \frac{col(G) + \Delta(G)}{2} \right\rceil \quad (۳.۳)$$

در [۲۳] نشان داده شده است کران بالا (۳.۳) برای $s(G)$ ، دقیق است. اگر در رابطه (۳.۳) گراف

³Szekeres-Wilf

G را درخت فرض کنیم، آنگاه رابطه مذکور بصورت زیر خواهد بود: .

$$s(T) \leq \lceil \frac{\Delta(T)}{2} \rceil + 1$$

این کران برای درخت‌ها می‌تواند وضعیت بهتری نیز داشته باشد.

گزاره ۲.۱.۳. [۱۵] برای هر درخت T با قطر $d(T)$ داریم:

$$s(T) \leq \lceil \frac{\min(d(T), \Delta(T))}{2} \rceil + 1$$

از آنجایی که $\chi(G) \leq col(G)$ نویسندگان [۱۵] کران بهتری نسبت به کران رابطه (۳.۳) برای قدرت راسی گراف حدس زده‌اند.

حدس ۲.۱.۳. [۱۵] برای هر گراف دلخواه مانند G داریم:

$$s(G) \leq \left\lceil \frac{\chi(G) + \Delta(G)}{2} \right\rceil \quad (۴.۳)$$

از آنجایی که عدد رنگ آمیزی هر درخت با عدد رنگی آن برابر است و مقدار دو می‌باشد، لذا حدس فوق برای درخت‌ها برقرار است درحالی‌که حتی برای گراف‌های دوبخشی هنوز ثابت نشده‌است. در [۲۳] نشان داده شده‌است که کران (۳.۳) برای برخی درخت‌ها دقیق است.

قضیه ۵.۱.۳. [۲۳] برای هر عدد صحیح مثبت k درختی مثل T_k وجود دارد بطوری که $s(T_k) = k$ و $\Delta(T_k) = 2k - 2$.

اثبات قضیه (۵.۱.۳) بصورت ساختاری است. به این مفهوم که در [۲۳] روشی برای ساختن درخت T_k ارائه شده‌است. این قضیه باعث بهبود اثبات ساختاری نتیجه‌ای مشابه در [۳۰] مبنی بر وجود درختی با قدرت راسی k و بیشترین درجه $\frac{k^2}{4}$ شده‌است. این بخش را با دو قضیه‌ی زیر که اولی در مورد ارتباط بین عدد رنگی و قدرت گراف و دومی در ارتباط با قدرت یالی گراف است، به پایان می‌بریم.

قضیه ۶.۱.۳. [۲۳] برای هر عدد صحیح $k \geq 2$ و $t \geq 0$ گراف G_k^t وجود دارد بطوریکه $\chi(G_k^t) = k$ و $s(G_k^t) = k + t$.

مشابه تعریف مجموع رنگی یالی، $\Sigma' G$ ، قدرت یالی گراف G را می‌توان بصورت $s'(G) = s(L(G))$ تعریف کرد. همچنین برای قدرت یالی گراف مشابه قضیه ویزینگ، قضیه زیر وجود دارد:

قضیه ۷.۱.۳. [۱۵] برای هر گراف همبند داریم

$$s'(G) \leq \Delta(G) + 1 \quad (۵.۳)$$

۲.۳ کران الگوریتمی

همانطور که قبلاً بیان شد، کران الگوریتمی برای مجموع رنگی یا قدرت گراف، کرانی است که بر اساس اجرای یک الگوریتم و روش حل برای گراف دلخواه بدست می‌آید در واقع این نوع کران از گرافی به گراف دیگر ممکن است تفاوت داشته باشد و برخلاف کران تحلیلی، نمی‌توان آنرا به سایر پارامترهای گراف مرتبط کرد. بنابراین در این قسمت روش‌های مختلف کران‌های مختلفی را منجر می‌شود. به عنوان مثال با ذکر منبع به دسته‌ای از آنها اشاره می‌کنیم.

دسته‌ای از آنها به تولید کران پایین برای مسئله رنگ آمیزی مجموع اختصاص دارد که از آنها با عنوان روش‌های فرا ابتکاری یاد می‌شود که از برخی از آنها عبارتند از: الگوریتم مورچگان (ANT) [۱۰]، روش جستجوی بهینه محلی بر اساس تغییر همسایگی (MDS(5)+LS) [۲۱] و روش تجزیه به خوشه‌ها (EXCLIQUE) [۵۰]. از طرف دیگر روش‌های ابتکاری و فرا ابتکاری دیگری وجود دارد که منجر به تولید کران بالا برای مسئله رنگ آمیزی مجموع می‌شوند. از جمله از آنها می‌توان به روش الگوریتم ژنتیک به همراه محدودیت‌های ابتکاری جانشین (GA) [۱۱]، روش MDS(5)+LS [۲۱] که برای کران پایین هم بکار گرفته می‌شود، روش جستجوی محلی با مکانیزم خروج از بهینه محلی BLS [۶]، روش جستجوی ممنوع EXSCOL [۵۱] که بر اساس استخراج مجموعه‌های مستقل عمل می‌کند.

کران بالا در این مقالات اغلب با یافتن یک رنگ آمیزی مجاز از یک الگوریتم رنگ آمیزی بدست می‌آید. از طرف دیگر کران پایین بر اساس استخراج زیر گراف‌های جزئی (تعریف ۱.۲.۳) از گراف اصلی است که مجموع رنگی برای آنها مشخص است. به عنوان مثال استخراج این زیر گراف‌های جزئی با تجزیه گراف اصلی به گراف‌های دوبخشی یا درخت‌ها یا خوشه‌ها [۳۹] امکان پذیر است.

تعریف ۱.۲.۳. یک زیر گراف جزئی $G' = (V, E')$ از گراف $G = (V, E)$ زیر گرافی است که در آن $E' \subseteq E$.

۱.۲.۳ افراز به خوشه‌ها

فرض کنید مجموعه رئوس V از گراف G به مجموعه‌های V_1, V_2, \dots, V_l افراز شده اند بطوری که هر زیر گراف $G[V_i]$ برای $i = 1, \dots, l$ یک خوشه است. بدیهی است گراف $G' = G[V_1] \cup G[V_2] \cup \dots \cup G[V_l]$ یک زیر گراف جزئی از G است. مجموع رنگی این زیر گراف برابر است با

$$\sum G' = \sum_{i=1}^l \frac{|V_i|(|V_i| + 1)}{2}$$

طبق تعریف گراف جزئی، هر رنگ آمیزی مجاز برای G ، برای G' هم مجاز است. لذا مجموع رنگی G' یک کران پایین برای مجموع رنگی G است، یعنی $\sum G' \leq \sum G$. حال مسئله این است که چگونه

گراف اصلی به خوشه‌ها افراز شود بطوریکه این کران پایین، بیشترین مقدار ممکن باشد. این مسئله یک مسئله بهینه‌سازی است که در [۵۰]، از آن با عنوان مسئله افزار به خوشه‌ها برای مسئله رنگ آمیزی مجموع کمینه^۴ ($PCMSCP$) یاد شده است. این مسئله ارتباط نزدیکی به مسئله بهینه‌سازی افراز به خوشه‌ها^۵ (PCP) دارد ولی لزوماً جواب بهینه یکسانی ندارد. مسئله $PCMSCP$ همانند PCP یک مسئله NP -سخت است [۳۹].

تعریف ۲.۲.۳. مسئله افراز به خوشه‌ها (PCP).

گراف G با مجموعه رئوس V را در نظر بگیرید. فرض کنید V به مجموعه‌های V_1, V_2, \dots, V_l افراز شود بطوریکه $G[V_i]$ برای $i = 1, \dots, l$ یک خوشه باشد. هدف در این مسئله کمترین تعداد افراز به خوشه است. به عبارت دیگر هدف در این مسئله یافتن حداقل تعداد افراز رئوس G است بطوری که گراف القایی متناظر با هر افراز خوشه باشد.

بعنوان مثال گراف شکل (۱۱.۳) را در نظر بگیرید. دو گراف جزئی حاصل از تجزیه این گراف به خوشه‌ها ایجاد شده که در شکل‌های (۱۰.۳ب) و (۱۰.۳ج) نمایش داده شده‌اند که در آن $\Lambda_1 = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$ و $\Lambda_2 = \{\{1, 3, 7, 9, 6, 4\}, \{2\}, \{8\}, \{5\}\}$ ترکیبی از سه خوشه است که مجموع رنگی برای آن برابر است با $3 \times \frac{3 \times 4}{2} = 18$ در حالیکه Λ_2 ترکیبی از چهار خوشه است و مجموع رنگی متناظر آن برابر است با $3 + \frac{6 \times 7}{2} = 24$. لذا تجزیه Λ_2 تجزیه بهتری است، زیرا کران پایین بالاتری را برای مجموع رنگی G ایجاد می‌کند، در حالی که Λ_1 برای PCP افراز بهینه است. افراز $\Lambda = \{\{v_1\}, \{v_2\}, \dots, \{v_n\}\}$ یک تجزیه بدیهی به خوشه‌ها برای هر گراف دلخواه G است که منجر به حصول کران پایین بدیهی $\sum G \geq n$ می‌شود. بدیهی است بدست آوردن جواب $PCMSCP$ برای گراف‌هایی با عدد خوشه ۲ ($\omega(G) = 2$) معادل با مسئله یافتن تطابق بیشینه^۶ برای آن گراف است که این مسئله در زمان چند جمله‌ای حل می‌شود [۴۲].

اکنون سوالی که مطرح است این است که بیشترین مقدار $PCMSCP$ چه قدر می‌تواند باشد؟ یعنی به چه میزان جواب بهینه مسئله مذکور (که یک کران پایین برای مسئله رنگ آمیزی مجموع کمینه محسوب می‌شود) می‌تواند به جواب بهینه این مسئله نزدیک شود؟ با گزاره زیر، جواب این سوال تا حدی مشخص می‌شود. اهمیت یافتن پاسخ این سوال زمانی مشخص می‌شود که در فرایند جستجوی رنگ آمیزی‌ها با استفاده از بررسی افرازهای مختلف رئوس گراف به خوشه‌ها، می‌توان از آن به عنوان شرط توقف استفاده کرد.

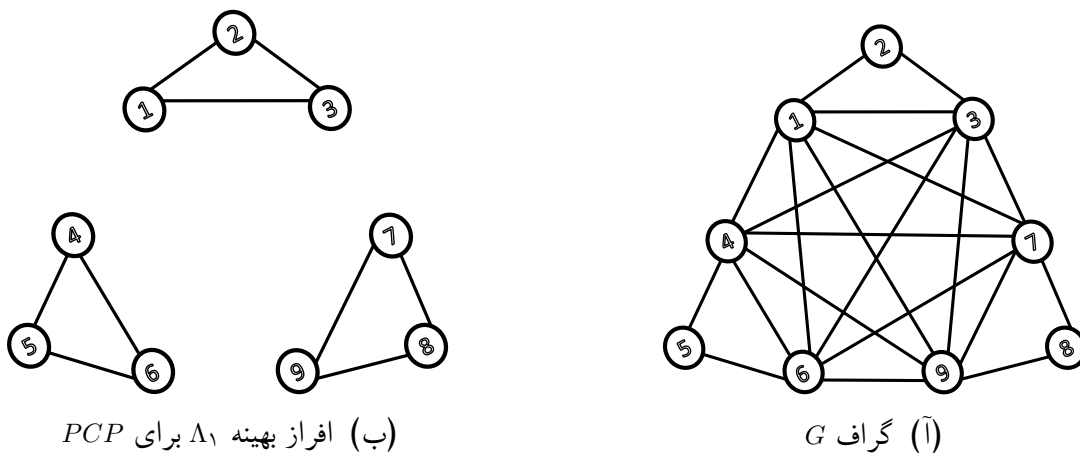
گزاره ۱.۲.۳. [۳۹] فرض کنید ω اندازه بزرگترین خوشه از G باشد و $r = n - \omega \lfloor \frac{n}{\omega} \rfloor$ در این صورت یک کران بالا برای $PCMSCP$ بصورت زیر است:

$$\sum_{Clique} G \leq \frac{\omega(\omega + 1)}{2} \lfloor \frac{n}{\omega} \rfloor + \frac{r(r + 1)}{2}$$

^۴Partition into Cliques for Minimum Sum Coloring Problem (PCMSCP)

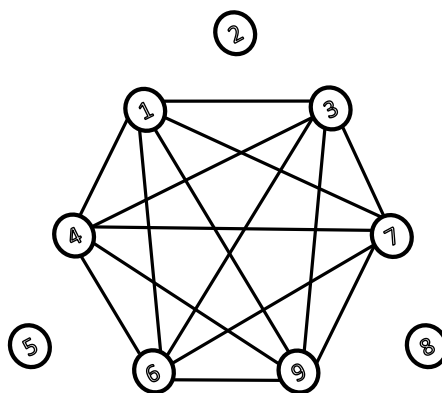
^۵Partition into Cliques Problem

^۶Maximum Matching Problem



(ب) افراز بهینه Λ_1 برای PCP

(ا) گراف G



(ج) افراز بهینه Λ_2 برای PCMSCP

شکل ۱۰.۳: دو نمونه زیر گراف جزئی با تجزیه به خوشه برای گراف G

که در آن، $\sum_{Clique} G$ مقدار جواب بهینه PCMSCP است.

روش یافتن زیر گراف جزئی (تجزیه به خوشه) که در این قسمت مطرح شد، منجر به محاسبه کران پایین برای مسئله رنگ آمیزی مجموع کمینه خواهد شد. روشن است که هر روشی که یک رنگ آمیزی مجاز را برای گراف G محاسبه کند یک کران بالا برای مسئله رنگ آمیزی مجموع کمینه خواهد بود. ولی با اندک توجهی متوجه می شویم از روش بیان شده می توان در یافتن کران بالا برای مسئله رنگ آمیزی مجموع کمینه استفاده کرد. در واقع هر افراز رئوس گراف G به خوشه‌ها معادل با افراز رئوس گراف مکمل (\bar{G}) به مجموعه‌های مستقل است. در حقیقت ما با افراز رئوس گراف \bar{G} به خوشه‌ها، به افراز رئوس G به مجموعه‌های مستقل (کلاس‌های رنگی) دست می یابیم و در نتیجه به یک رنگ آمیزی مجاز G می رسیم و در نهایت به یک کران بالا برای مسئله رنگ آمیزی مجموع کمینه حاصل می شود.

به عنوان مثال گراف شکل (۱۰.۳) را در نظر بگیرید. با محاسبه گراف مکمل متناظر با G ملاحظه می شود $\Lambda = \{\{2, 8, 5\}, \{1\}, \{3\}, \{4\}, \{6\}, \{7\}, \{9\}\}$ یک افراز رئوس \bar{G} به خوشه‌ها است. لذا

این افراز می‌تواند رنگ آمیزی مجازی مثل c برای گراف G القا کند که به ازای آن

$$26 = \sum G \leq \sum_c G = 3 + (2 + 3 + 4 + 5 + 6 + 7) = 30.$$

بنابراین برای این گراف، نامساوی زیر برقرار است:

$$24 \leq 26 = \sum G \leq 30$$

همچنین طبق توضیحات فوق یک کران برای مسئله رنگ آمیزی مجموع بصورت زیر است:

$$\sum_{Clique} G \leq \sum G \leq \sum (G, c)$$

که در آن c رنگ آمیزی بهینه‌ای است که از حل $PCMSCP$ روی گراف \bar{G} بدست آمده است.

فصل ۴

برخی از تعمیم‌های مجموع رنگی و کاربرد آن

۱.۴ مقدمه

در این فصل دو تعریف جدید برای گراف G که تعمیمی از مفهوم مجموع رنگی است، معرفی می‌شود. هدف اصلی در این فصل ارزیابی نتایج مشابه آنچه که در [۱] در مورد مجموع رنگی آمده است، می‌باشد. ارایه این دو مفهوم که عبارتند از مجموع رنگی جمعی و مجموع رنگی تفاضلی، منجر به دست یابی به نتایج جدید مشابه [۱] می‌شود. از جمله آنها می‌توان به شرایط عدم وجود همریختی بین دو گراف اشاره کرد. علاوه بر این یافتن کران پایین برای عدد رنگی کسری از جمله نتایج دیگری از ارائه این دو مفهوم است.

لذا در ابتدا چند تعریف مورد استفاده در این فصل، یاد آوری می‌شود. بیان این تعاریف برای فهم نتایجی که در این فصل مورد بررسی قرار می‌گیرد، ضروری به نظر می‌رسد. مفاهیمی چون همریختی^۱ بین گراف‌ها، گراف راس ترا یا^۲ و یال ترا یا^۳، گراف کنسر^۴ و عدد رنگی کسری^۵. از جمله مفاهیمی هستند که در این فصل بررسی می‌کنیم. کلیه تعاریف از منبع [۷] انتخاب شده است.

تعریف ۱.۱.۴. با فرض اینکه G و H دو گراف باشند، یک همریختی مانند f از گراف G به H ، یک تابع مانند $f: V(G) \rightarrow V(H)$ است، بطوریکه حافظ مجاورت باشد. یعنی برای هر u و v در G داشته باشیم:

$$uv \in E(G) \implies f(u)f(v) \in E(H)$$

مجموعه‌ی همه‌ی همریختی‌های از G به H را با $\text{Hom}(G, H)$ نشان داده می‌شود. بعلاوه دو گراف G و H را یکریخت^۶ می‌نامیم هرگاه تابع یک‌به‌یک و پوشای $f: V(G) \rightarrow V(H)$ وجود داشته باشد، به طوری که برای هر u و v در $V(G)$ رابطه زیر وجود داشته باشد

$$uv \in E(G) \iff f(u)f(v) \in E(H)$$

در این صورت f را یک یکریختی^۷ از G به H است. یک یکریختی $f: V(G) \rightarrow V(G)$ ، خودریختی^۸ روی G است. مجموعه‌ی همه‌ی خودریختی‌های روی $V(G)$ را با $\text{Aut}(G)$ نشان می‌دهیم. $\text{Aut}(G)$ ناتهی و البته متناهی می‌باشد زیرا تعداد رئوس، متناهی است.

تعریف ۲.۱.۴. گراف G راس ترا یا است هرگاه به ازای هر $u, v \in V(G)$ یک تابع مانند $f \in \text{Aut}(G)$ موجود باشد به طوری که $u = f(v)$.

¹Homomorphism

²Vertex Transitive

³Edge Transitive

⁴Kneser Graph

⁵Fractional Coloring Number

⁶Isomorphic

⁷Isomorphism

⁸Automorphism

تعریف ۳.۱.۴. گراف G یال ترا یا است هرگاه گراف خط متناظر با آن راس ترا یا باشد.

گراف راس ترا یا لزوماً منتظم است. گراف کامل k_n و گراف کنسر $KG(m, n)$ که در ادامه تعریف می‌شود، نمونه‌هایی از گراف راس ترا یا و یال ترا یا هستند.

تعریف ۴.۱.۴. گراف کنسر، که آن را با $KG(m, n)$ نشان می‌دهیم و $m \geq 2n$ گرافی است که رئوس آن تمام زیرمجموعه‌های n عضوی از مجموعه $\{1, 2, \dots, m\}$ هستند. در این حالت دو رأس در صورتی با هم مجاورند هرگاه زیر مجموعه‌های n عضوی متناظر آنها هیچ اشتراکی با هم نداشته باشند.

بنابراین تعداد رئوس برابر است با $\binom{m}{n}$ و از آنجایی که $KG(m, n)$ یک گراف $(m-n)$ -منتظم می‌باشد، لذا تعداد یال‌های آن برابر است با

$$\frac{\binom{m}{n} \cdot \binom{m-n}{n}}{2}$$

گراف کامل k_m را می‌توان به صورت گراف کنسر $KG(m, 1)$ نشان داد. همچنین در [۳۳] ثابت شده است

$$\chi(KG(m, n)) = m - 2n + 2$$

براحتی می‌توان نشان داد، عدد رنگی گراف G که در فصل اول معرفی شد، با مقدار زیر برابر است [۷]:

$$\chi(G) = \min \{r \mid \exists f : G \rightarrow K_r \text{ هم‌ریختی یک}\}$$

تعریف ۵.۱.۴. عدد رنگی کسری گراف G که با $\chi_f(G)$ نشان می‌دهیم، برابر است با:

$$\chi_f(G) = \inf \left\{ \frac{m}{n} \mid \exists f : G \rightarrow KG(m, n) \text{ هم‌ریختی یک} \right\}$$

بعد از ارائه‌ی این تعریف، ثابت شد که اینفیمم در این تعریف حاصل می‌شود [۴۴] و می‌توان لفظ اینفیمم را با مینیمم تعویض کرد.

$$\chi_f(G) = \min \left\{ \frac{m}{n} \mid \exists f : G \rightarrow KG(m, n) \text{ هم‌ریختی یک} \right\}$$

با توجه به دو تعریف فوق می‌توان دریافت، نقشی که گراف کامل برای عدد رنگی رأسی دارد، گراف کنسر برای عدد رنگی کسری ایفا می‌کند. به وضوح می‌توان مشاهده کرد $\chi_f(G) \leq \chi(G)$ و علاوه در [۴۴] نشان داده شده است که $\frac{\chi(G)}{\chi_f(G)}$ می‌تواند به اندازه‌ی دلخواه بزرگ باشد و این یعنی اینکه $\chi_f(G)$ می‌تواند خیلی کوچکتر از $\chi(G)$ باشد. بنابراین گراف‌هایی وجود دارند که در آنها عدد رنگی کران بالای خوبی برای عدد رنگی کسری نیست.

۲.۴ مجموع رنگی تفاضلی و مجموع رنگی جمعی

تعریف ۱.۲.۴ [۴۵] فرض کنید $c: V \rightarrow I$ یک رنگ آمیزی مجاز برای رئوس G باشد. رنگ آمیزی مجموع تفاضلی^۹ و رنگ آمیزی مجموع جمعی^{۱۰} متناظر با c به ترتیب با نمادهای $\sum_c D(G)$ و $\sum_c S(G)$ نمایش می‌دهیم و بصورت زیر تعریف می‌کنیم

$$\sum_c D(G) = \sum_{(u,v) \in E(G)} |c(u) - c(v)| \quad (۱.۴)$$

$$\sum_c S(G) = \sum_{(u,v) \in E(G)} (c(u) + c(v)) \quad (۲.۴)$$

تعریف ۲.۲.۴ فرض کنید C مجموعه تمام رنگ آمیزی‌های مجاز برای G باشند. مجموع رنگی تفاضلی^{۱۱} و مجموع رنگی جمعی^{۱۲} را به ترتیب با نمادهای $\sum D(G)$ و $\sum S(G)$ نمایش می‌دهیم و بصورت زیر تعریف می‌کنیم

$$\sum D(G) = \min \left\{ \sum_c D(G) \mid c \in C \right\} \quad (۳.۴)$$

$$\sum S(G) = \min \left\{ \sum_c S(G) \mid c \in C \right\} \quad (۴.۴)$$

بعلاوه کمترین تعداد رنگِ رنگ آمیزی مجاز c برای آنکه $\sum_c D(G) = \sum D(G)$ را با $ds(G)$ نشان می‌دهیم. مشابه کمترین تعداد رنگِ رنگ آمیزی مجاز c برای آنکه $\sum_c S(G) = \sum S(G)$ را با $ss(G)$ نمایش می‌دهیم.

بدیهی است که $\chi(G) \leq ds(G)$ و حتی مواردی وجود دارد که به ازای آنها نامساوی اخیر بصورت اکید برقرار است (شکل ۱.۴) را ببینید. همچنین در حالت کلی داریم $ss(G) \neq s(G)$ (شکل ۲.۴ را ببینید).

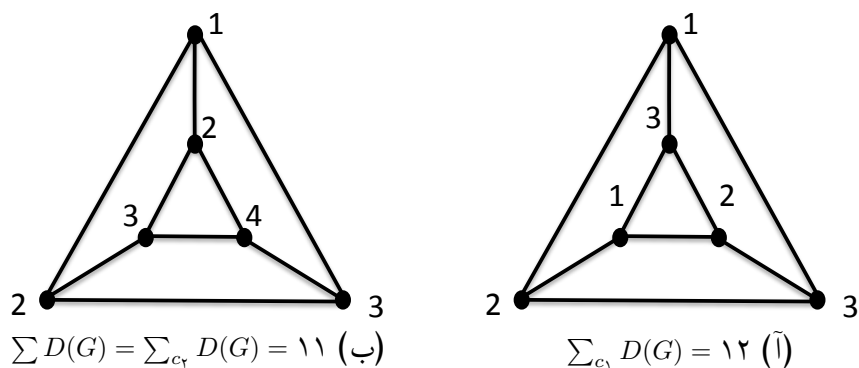
اکنون فرض کنید f یک رنگ آمیزی مجاز دلخواه برای رئوس G باشد و همچنین برای هر راس

⁹difference sum coloring

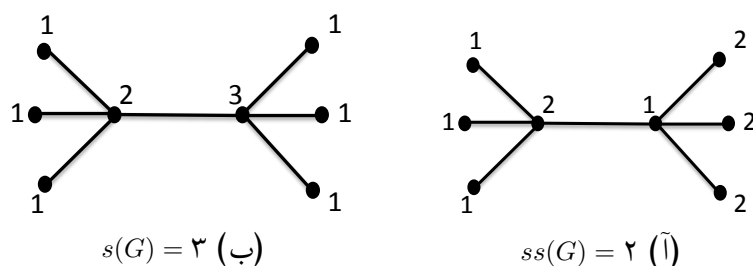
¹⁰sum sum coloring

¹¹difference chromatic sum

¹²sum chromatic sum



شکل ۱.۴: مجموع رنگی تفاضلی با $ds(G) = 4$ و $\chi(G) = 3$



شکل ۲.۴: $ss(G) \neq s(G)$

$v \in V$ ، $d(v)$ درجه راس v باشد. لذا داریم:

$$\sum_{(u,v) \in E(G)} (f(u) + f(v)) = \sum_{v \in V(G)} d(v)f(v)$$

$$\Rightarrow \delta \sum_{v \in V(G)} f(v) \leq \sum_{(u,v) \in E(G)} (f(u) + f(v)) \leq \Delta \sum_{v \in V(G)} f(v) \quad (5.4)$$

رابطه (۵.۴) برای هر رنگ آمیزی مجاز مثل f برقرار است. لذا داریم

$$\delta \min_f \left\{ \sum_{v \in V(G)} f(v) \right\} \leq \min_f \left\{ \sum_{(u,v) \in E(G)} (f(u) + f(v)) \right\} \leq \Delta \min_f \left\{ \sum_{v \in V(G)} f(v) \right\}$$

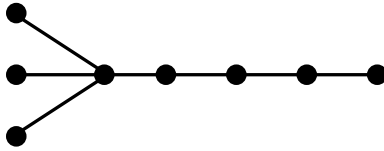
$$\Rightarrow \delta \sum G \leq \sum S(G) \leq \Delta \sum G \quad (6.4)$$

لذا رابطه (۶.۴) ما را به کران زیر برای مجموع رنگی می‌رساند.

$$\left\lceil \frac{\sum S(G)}{\Delta} \right\rceil \leq \sum G \leq \left\lfloor \frac{\sum S(G)}{\delta} \right\rfloor \quad (7.4)$$

همانطور که ملاحظه می‌شود، اگر در گرافی $\delta \ll \Delta$ (مثلا شکل ۳.۴)، کران (۷.۴) برای مجموع رنگی کران خوبی نیست ($6 \leq \sum G = 11 \leq 21$). ولی بدیهی است برای گراف‌هایی که در آنها

مقادیر δ و Δ بهم نزدیک باشند کران (۷.۴) دقیقتر خواهد بود و در حالتی که $\Delta = \delta = k$ (یعنی برای گراف‌های k -منتظم) گزاره زیر را داریم.



شکل ۳.۴: $\Delta(G) = 4, \delta(G) = 1$

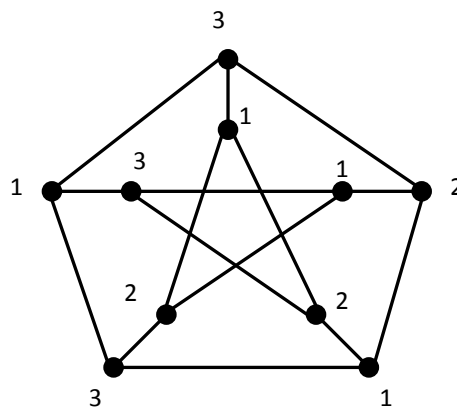
گزاره ۱.۲.۴. برای هر گراف k -منتظم G داریم:

$$\sum S(G) = k \sum G$$

و بعلاوه $ss(G) = s(G)$.

بعنوان مثال برای گراف پترسن P (شکل ۴.۴) داریم $\sum(P) = 19$ و $\sum S(G) = 3 \times 19 = 57$. بنابراین بنابه گزاره فوق برای هر گراف k -منتظم G رابطه زیر برقرار است.

$$\frac{\sum G}{\sum S(G)} = \frac{1}{k} = \frac{|V(G)|}{2|E(G)|} \quad (۸.۴)$$



شکل ۴.۴: گراف پترسن

طبق رابطه (۸.۴) نسبت مجموع رنگی و مجموع رنگی جمعی برای گراف‌های منتظم همواره مقدار ثابتی است.

اکنون سوالی که در این قسمت مطرح است آن است که رابطه اخیر برای گراف‌های غیر منتظم چگونه است؟ آیا می‌توان گرافی مثل G یافت که در آن کسر $\frac{\sum G}{\sum S(G)}$ از کسر $\frac{|V(G)|}{2|E(G)|}$ بیشتر باشد؟ (حدس)

۳.۲.۴. مثلاً برای گراف شکل (۲.۴) داریم:

$$\frac{\sum G}{\sum S(G)} = \frac{11}{21} \leq \frac{4}{7} = \frac{|V(G)|}{2|E(G)|}$$

یا سوال دیگر اینکه آیا عکس رابطه مربوط به گزاره (۸.۴) برقرار است؟ یعنی اگر برای گرافی رابطه (۸.۴) برقرار باشد، آیا می‌توان گفت آن گراف منتظم است؟ (حدس ۴.۲.۴)

حدس ۳.۲.۴. برای هر گراف G داریم:

$$\frac{\sum G}{\sum S(G)} \leq \frac{|V(G)|}{2|E(G)|}$$

حدس ۴.۲.۴. گراف G ، k -منتظم است اگر و تنها اگر،

$$\sum S(G) = k \sum G$$

حدس‌های اخیر را می‌توان در حدس زیر خلاصه کرد.

حدس ۵.۲.۴. برای هر گراف G داریم:

$$\frac{\sum G}{\sum S(G)} \leq \frac{|V(G)|}{2|E(G)|}$$

و تساوی فوق فقط زمانی اتفاق می‌افتد که گراف G ، منتظم باشد.

۳.۴ شرایط لازم برای وجود همریختی بین دو گراف

یکی از مسائل مهم و مورد توجه در نظریه گراف، مسئله همریختی بین گراف‌هاست که با مفاهیم بسیار دیگری از نظریه گراف در ارتباط است. در حالت کلی، پاسخ به این سوال که آیا همریختی بین دو گراف G و H وجود دارد یا خیر؟، یک مسأله‌ی NP -کامل است. بنابراین بدست آوردن شرایط لازم برای وجود چنین نگاشت‌هایی می‌تواند بسیار حائز اهمیت باشد.

به‌عنوان مثال می‌دانیم که اگر $f: G \rightarrow H$ یک همریختی باشد آنگاه $\chi(G) \leq \chi(H)$. لذا شرط $\chi(G) \leq \chi(H)$ یک شرط لازم برای وجود همریختی از گراف G به گراف H است. مثلاً از گراف پترسن به گراف C_4 همریختی وجود ندارد. بنابراین هرچه بتوان دامنه شروط لازم برای وجود همریختی بین دو گراف را بیشتر گسترش داد، می‌توان دسته وسیع‌تری از گراف‌هایی که بین آنها اساساً همریختی وجود ندارد را بهتر تشخیص داد.

در این قسمت قصد داریم برخی شرایط لازم برای وجود همریختی بین دو گراف را که مبتنی بر دو

مفهوم بیان شده در بخش (۲.۴) است را بیان کنیم. در ارتباط با مفهوم مجموع رنگی، قضیه زیر شرط لازم برای وجود همریختی را از گراف G به یک گراف راس H ، معرفی می‌کند.

قضیه ۱.۳.۴ [۱] فرض کنید G و H دو گراف باشند به طوری که H یک گراف راس H باشد. اگر $f: G \rightarrow H$ یک همریختی بین G و H باشد، آنگاه:

$$\frac{\sum(G)}{|G|} \leq \frac{\sum(H)}{|H|}$$

طبق قضیه فوق اگر همریختی مثل f از G به گراف راس H موجود باشد، باید داشته باشیم:

$$\frac{\sum(G)}{|G|} \leq \frac{\sum(H)}{|H|}$$

. بنابراین اگر رابطه اخیر برقرار نباشد، همریختی ای هم از G به H موجود نخواهد بود. در این قضیه شرط وجود همریختی به مفهوم مجموع رنگی مرتبط شده است. ما در دو قضیه بعد دو شرط وجود همریختی از گراف G به گراف H را با مفاهیم جدید مجموع رنگی تفاضلی و جمعی مرتبط کرده‌ایم.

قضیه ۲.۳.۴ فرض کنیم G و H دو گراف باشند به طوری که H یک گراف H باشد. اگر $f: G \rightarrow H$ یک همریختی بین G و H باشد، آنگاه

$$\frac{\sum D(G)}{|E(G)|} \leq \frac{\sum D(H)}{|E(H)|}. \quad (۹.۴)$$

برهان. فرض کنید $\{\sum_1, \dots, \sum_n\} = \text{Aut}(H)$. همچنین فرض کنید G_1, G_2, \dots, G_n n تایی یکرخت با G باشند. قرار دهید $\tilde{G} = \bigcup_{i=1}^n G_i$. بعلاوه فرض کنید $\tilde{f}: \tilde{G} \rightarrow H$ بطوری که \tilde{f} به G_i برابر باشد با $\sum_i \circ f$ ، یعنی $\tilde{f}|_{G_i} = \sum_i \circ f$. واضح است که \tilde{f} یک همریختی است بطوری که هر یال $e = ab \in E(H)$ داریم $|\tilde{f}^{-1}(e)| = \frac{n|E(G)|}{|E(H)|}$. فرض کنید c یک رنگ آمیزی از رئوس H باشد بطوری که $\sum_c D(H) = \sum D(H)$. برای هر راس $v \in V(\tilde{G})$ قرار دهید $\tilde{c}(v) = c(\tilde{f}(v))$. واضح است که \tilde{c} یک رنگ آمیزی مجاز برای رئوس \tilde{G} است. بنابراین

$$\sum_{\tilde{c}} D(\tilde{G}) = \frac{n|E(G)|}{|E(H)|} \sum_c D(H).$$

لذا اندیسی مثل i وجود دارد بطوری که \tilde{c} به G_i رنگ آمیزی مجازی مثل c' خواهد بود بطوری که

$$\sum_{c'} D(G) \leq \frac{1}{n} \sum_{\tilde{c}} D(\tilde{G}) = \frac{|E(G)|}{|E(H)|} \sum_c D(H)$$

از طرف دیگر می‌دانیم $\sum D(G) \leq \sum_c D(G)$ بنابراین

$$\sum D(G) \leq \frac{|E(G)|}{|E(H)|} \sum_c D(H)$$

و این اثبات را کامل می‌کند. \square

قضیه ۳.۳.۴. فرض کنیم G و H دو گراف باشند به طوری که H یک گراف یال ترایا باشد. اگر $f: G \rightarrow H$ یک همریختی بین G و H باشد، آنگاه

$$\frac{\sum S(G)}{|E(G)|} \leq \frac{\sum S(H)}{|E(H)|}. \quad (۱۰.۴)$$

برهان. با استدلالی مشابه، همانند آنچه در اثبات قضیه (۲.۳.۴) آمد، می‌توان این قضیه را ثابت کرد. \square

روابط (۹.۴) و (۱۰.۴) دو شرط لازم برای وجود همریختی از گراف G به گراف یال ترایای H هستند.

نتیجه ۱.۳.۴. هیچ همریختی از گراف پترسن به C_5 وجود ندارد.

برهان. توجه کنید گراف C_5 یک گراف یال ترایاست. $\sum D(C_5) = 6$, $\sum S(C_5) = 18$. همچنین $\sum D(P) = 18$, $\sum S(P) = 57$. با اطلاعات فوق از آنجا که نامساوی (۱۰.۴) نقض می‌شود، لذا قضیه (۳.۳.۴) نتیجه می‌دهد هیچ همریختی از پترسن به C_5 وجود ندارد. \square

۴.۴ کران بالا برای $\sum D(KG(m, n))$ و $\sum S(KG(m, n))$

همانطور که در بخش اول این فصل بیان شد، گراف کنسر نقش مهمی در تعریف عدد رنگ آمیزی کسری دارد و از آنجا که بخش بعدی به ارایه کران پایین برای عدد رنگی کسری اختصاص دارد، لذا این بخش را به ارائه یک کران بالا برای $\sum D(KG(m, n))$ و $\sum S(KG(m, n))$ اختصاص داده‌ایم. یافتن کران بالا منوط به ارائه رنگ آمیزی مجازی برای گراف $KG(m, n)$ است. هر رنگ آمیزی مجاز در واقع یک افراز برای رئوس گراف به مجموعه‌های مستقل است.

ذکر این نکته ضروری است که کلیه ی محاسباتی که در این قسمت ملاحظه می‌شود، با دستورات نرم افزار میپل^{۱۳} نسخه ۰۰.۱۴ انجام شده و قابل بررسی است.

ما در اینجا مجموعه‌های مستقل که قرار است رئوس $KG(m, n)$ را افراز کنند را با A_i نمایش داده و برای هر $i = 1, 2, \dots, m - n + 1$ آنها را بصورت زیر تعریف می‌کنیم:

$$A_i = \{A \subseteq [m], |A| = n \mid A \text{ شامل } i \text{ باشد ولی شامل } 1, 2, \dots, i-1 \text{ نباشد}\} \quad (۱۱.۴)$$

که در آن منظور از $[m]$ مجموعه $\{1, 2, \dots, m\}$ است. شامل i باشد ولی شامل $1, 2, \dots, i-1$ نباشد بنا بر توضیحات فوق ابتدا باید ثابت کنیم مجموعه‌های تعریف شده A_i برای $i = 1, 2, \dots, m-n+1$ و $m \geq 2n$ اولاً مستقل هستند و ثانیاً رئوس $KG(m, n)$ را افزاز می‌کنند.

گزاره ۱.۴.۴. فرض کنید $m \geq 2n$ و مجموعه‌های A_i برای $i = 1, 2, \dots, m-n+1$ بر طبق رابطه (۱۱.۴) تعریف شده باشند، در این صورت برای گراف $KG(m, n)$ داریم:

(آ) برای هر $i = 1, 2, \dots, m-n+1$ ها مجموعه‌های مستقل هستند.

(ب) برای هر s و t بطوریکه $1 \leq s \neq t \leq m-n+1$ داریم $A_s \cap A_t = \emptyset$.

(ج) $V(KG(m, n)) = \bigcup_{i=1}^{m-n+1} A_i$

برهان. (آ) از آنجایی که A_i شامل تمام زیر مجموعه‌هایی است که عنصر i را دارند لذا برای هر A و B در A_i داریم $A \cap B = \emptyset$. بنا بر این طبق تعریف گراف کنسر مجموعه‌های A و B نمی‌توانند مجاور باشند. پس برای هر $i = 1, 2, \dots, m-n+1$ مجموعه‌ای مستقل است.

(ب) بدون از دست دادن کلیت فرض کنید $s < t$. به برهان خلف فرض کنید $B \in A_s \cap A_t$. بنابه تعریف مجموعه‌های A_s و A_t ، زیر مجموعه n عضوی B شامل عنصر s است ($B \in A_s$) و هم شامل عنصر s نیست ($B \in A_t$) و این تناقض است.

(ج) برای هر زیر مجموعه n عضوی A در $V(KG(m, n))$ فرض کنید t کوچکترین عضو A باشد. لذا $A \in A_t$. پس $V(KG(m, n)) = \bigcup_{i=1}^{m-n+1} A_i$ □

گزاره فوق بیان می‌کند مجموعه‌های A_i برای $i = 1, \dots, m-n+1$ مجموعه‌های مستقلی هستند که رئوس گراف $KG(m, n)$ را افزاز می‌کنند.

گزاره ۲.۴.۴. خانواده مجموعه‌های $\{A_i\}_{i=1}^{m-n+1}$ دارای خواص زیر هستند. برای $i = 1, 2, \dots, m-n+1$ داریم

$$|A_i| = \binom{m-i}{n-1} \quad (۱)$$

$$|A_{i+1}| < |A_i| \quad (۲)$$

(۳) برای هر راس $v \notin A_i$ درجه v نسبت به مجموعه A_i عبارتست از

$$\binom{m-n-i}{n-1}$$

از قسمت سوم گزاره (۲.۴.۴) نتیجه می‌شود

$$m - n - i \leq n - 1 \implies i \leq m - 2n + 1 \quad (۱۲.۴)$$

رابطه (۱۲.۴) نشان می‌دهد بین مجموعه‌های A_i و A_j برای $m - 2n + 2 \leq i, j \leq m - n + 1$ یالی وجود ندارد. لذا می‌توان افزایش نتیجه شده از گزاره (۱.۴.۴) به مجموعه‌های مستقل را کوچکتر در نظر گرفت و بصورت

$$A_1, A_2, \dots, A_{m-2n+1}, B_{m-2n+2}$$

بیان کرد که در آن

$$B_{m-2n+2} = \bigcup_{i=m-2n+2}^{m-n+1} A_i.$$

یک مجموعه مستقل است.

بنابراین این $m - 2n + 2$ مجموعه مستقل یک کلاس رنگی مجاز برای رئوس $KG(m, n)$ با $\chi(KG(m, n)) =$ رنگ است. اکنون رنگ آمیزی c را برای $KG(m, n)$ بصورت زیر در نظر می‌گیریم. ۱ را به A_1 ، رنگ ۲ را به A_2 و ... و رنگ $m - 2n + 2$ را به B_{m-2n+2} اختصاص می‌دهیم. حال با این رنگ آمیزی می‌خواهیم کران بالایی برای $\sum D(KG(m, n))$ بیابیم. برای این منظور نیاز به رابطه‌ای داریم که در ادامه آنرا بیان می‌کنیم. فرض کنید S مجموعه مستقل دلخواهی از گراف دلخواه G باشد. در اینصورت رابطه (۱۳.۴) نتیجه بدیهی تعریف مجموع رنگی تفاضلی است.

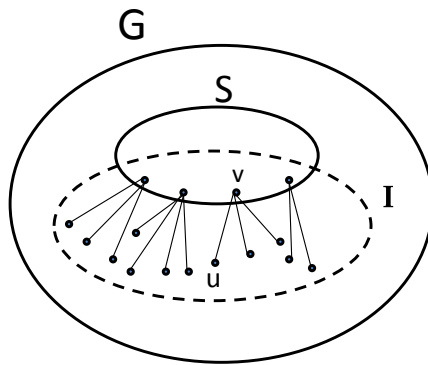
$$\sum D(G) \leq \sum_{c'} D(G) = \sum_{(u,v) \in I} |c'(u) - c'(v)| + \sum D(G \setminus S) \quad (۱۳.۴)$$

که در آن I مجموعه تمام یال‌های بین S و $G \setminus S$ است و

$$c'(v) = \begin{cases} \min \{c(u) | u \in N(v)\} + 1 & v \in S \\ c(v) & v \notin S \end{cases}$$

و هر رنگ آمیزی مجازی برای G می‌تواند باشد. برای درک بیشتر، شکل (۵.۴) را ملاحظه بفرمایید. حال در رابطه (۱۳.۴) قرار دهید $G = KG(m, n)$ و $S = A_1$. عبارت $\sum_{(u,v) \in I} |c'(u) - c'(v)|$ به طور دقیق قابل محاسبه است اگر رنگ آمیزی c را رنگ آمیزی بیان شده برای A_i ها که در ابتدای این بخش معرفی شد در نظر بگیریم.

$$\sum D(KG(m, n)) \leq \sum D(KG(m-1, n)) + \binom{m-n-1}{n-1} \left[\binom{m}{n+1} - \frac{n-1}{n+1} \binom{2n-1}{n} \right] \quad (۱۴.۴)$$



شکل ۵.۴: S یک مجموعه مستقل دلخواه از G و I مجموعه یال‌های بین S و $G \setminus S$

همچنین با جایگذاری گراف $KG(m-1, n)$ بجای $KG(m, n)$ در رابطه (۱۴.۴) بعد از $m-2n$ مرحله داریم

$$\begin{aligned} \sum D(KG(m, n)) &\leq \frac{m+1}{2n+1} \binom{m-n-1}{n-1} \binom{m}{n+1} - \frac{n-1}{n+1} \binom{2n-1}{n} \left[\binom{m-n}{n} - 1 \right] \\ &\quad - \binom{2n}{n+1} + \sum D(KG(2n, n)) \\ &= \frac{m+1}{2n+1} \binom{m-n-1}{n-1} \binom{m}{n+1} - \frac{n-1}{n+1} \binom{2n-1}{n} \left[\binom{m-n}{n} - 1 \right] \\ &\quad - \binom{2n}{n+1} + \frac{1}{2} \binom{2n}{n}. \end{aligned}$$

عبارت $\frac{1}{2} \binom{2n}{n} - \binom{2n}{n+1}$ برای $n \in \mathbb{N}$ یک عبارت نامثبت است زیرا

$$\begin{aligned} \frac{1}{2} \binom{2n}{n} - \binom{2n}{n+1} &= \frac{(2n)!}{2 \times n!n!} - \frac{(2n)!}{(n-1)!(n+1)!} = \frac{(2n)!(n+1)}{2 \times n!(n+1)!} - \frac{(2n)!(2n)}{2 \times (n)!(n+1)!} \\ &= \frac{(2n)!(1-n)}{2 \times (n)!(n+1)!} \leq 0. \end{aligned}$$

بنابراین

$$\sum D(KG(m, n)) \leq \frac{m+1}{2n+1} \binom{m-n-1}{n-1} \binom{m}{n+1}. \quad (15.4)$$

بعلاوه چون گراف $KG(m, n)$ یک گراف $(m-n)$ -منتظم است لذا

$$\sum S(KG(m, n)) \leq \binom{m-n}{n} \sum_c KG(m, n). \quad (16.4)$$

$$\begin{aligned} \sum_c KG(m, n) &= |A_1| + 2|A_2| + \dots + (m - 2n + 1)|A_{m-2n+1}| + (m - 2n + 2)|B_{m-2n+2}| \\ &= \binom{m}{n} \left(\frac{m+1}{n+1} - \frac{n-1}{2n+2} \frac{\binom{2n}{n}}{\binom{m}{n}} \right) \end{aligned}$$

که این مقدار همان مقداری است که در [۱] بعنوان مقدار دقیق مجموع رنگی گراف کنسر حدس زده شده است. نامساوی (۱۶.۴) نتیجه می‌دهد

$$\sum S(KG(m, n)) \leq \binom{m-n}{n} \binom{m}{n} \left(\frac{m+1}{n+1} - \frac{n-1}{2n+2} \frac{\binom{2n}{n}}{\binom{m}{n}} \right) \quad (17.4)$$

روابط (۱۷.۴) و (۱۵.۴) به ترتیب دو کران بالا برای مجموع رنگی تفاضلی و مجموع رنگی جمعی هستند که در بخش بعد مورد استفاده قرار می‌گیرند.

۵.۴ کران پایین برای عدد رنگی کسری

همانطور که قبلا بیان شد نقشی که گراف کامل برای عدد رنگی رأسی دارد، گراف کنسر برای عدد رنگی کسری ایفا می‌کند. در این بخش، کاربردی از قضیه (۲.۳.۴) و (۳.۳.۴) را با بکارگیری و استفاده از این ارتباط، جهت یافتن کران پایینی برای عدد رنگی کسری برای هر گراف G ، ارائه می‌کنیم. در [۱] کران پایینی برای عدد رنگی کسری بعنوان نتیجه ای از قضیه (۱.۳.۴) ارائه شده است.

قضیه ۱.۵.۴. [۱] برای هر گراف G نامساوی زیر برقرار است

$$\sum G < \chi_f(G) |V(G)|$$

اکنون در ارتباط با مفاهیم مجموع رنگی تفاضلی و جمعی دو قضیه زیر را داریم.

قضیه ۲.۵.۴. برای هر گراف G نامساوی زیر برقرار است

$$\sum D(G) < \chi_f(G) |E(G)| \quad (18.4)$$

برهان. فرض کنید $\chi_f(G) = \frac{m}{n}$ و $Hom(G, KG(m, n)) \neq \emptyset$. بنا به رابطه $|E(KG(m, n))| =$

$\frac{(m-n)(m)}{۳}$ و نامساوی (۱۵.۴) و قضیه (۲.۳.۴) می توان نتیجه گرفت

$$\begin{aligned} \sum D(G) &\leq \sum D(K(m,n)) \frac{|E(G)|}{|E(KG(m,n))|} \\ &\leq \frac{۲n(m+۱)}{(n+۱)(۲n+۱)} |E(G)| \\ &\leq \frac{m+۱}{n+۱} |E(G)| \\ &< \frac{m}{n} |E(G)| = \chi_f(G) |E(G)| \end{aligned}$$

□

قضیه ۳.۵.۴. برای هر گراف G نامساوی زیر برقرار است

$$\sum S(G) < ۲\chi_f(G) |E(G)|. \quad (۱۹.۴)$$

برهان. اثبات این قضیه مشابه قضیه قبل بر اساس نامساوی (۱۷.۴) و قضیه (۳.۳.۴) است. □

اکنون فرض کنید G یک گراف یال انتقالی باشد و $\omega(G)$ اندازه بزرگترین خوشه از G باشد. از آنجایی که همواره یک همریختی از $K_{\omega(G)}$ به G وجود دارد، قضیه (۲.۳.۴) نتیجه می دهد $\sum D(G) \geq \frac{\omega(G)+۱}{۳} |E(G)|$. همچنین بطور مشابه می توان بدست آورد $\sum S(G) \geq (\omega(G)+۱) |E(G)|$. در قضیه بعد به ارائه کران بالای دیگری برای مجموع رنگی تفاضلی و جمعی مبتنی بر عدد رنگی گراف می پردازیم.

قضیه ۴.۵.۴. برای هر گراف G داریم

$$\sum D(G) \leq \frac{(\chi(G)+۱)}{۳} |E(G)| \quad (\bar{A})$$

$$\sum S(G) \leq (\chi(G) - ۱) |E(G)| \quad (B)$$

برهان. (A) فرض کنید $f: G \rightarrow K_n$ یک همریختی باشد بطوریکه $\chi(G) = n$. از قضیه (۲.۳.۴) نتیجه می شود $\sum D(G) \leq \frac{|E(G)|}{|E(K_n)|} \sum D(K_n)$. به سادگی قابل بررسی است که $|\sum D(K_n)| = \frac{n(n-۱)(n+۱)}{۶}$ بنابراین داریم

$$\sum D(G) \leq |E(G)| \frac{(n+۱)}{۳}$$

و این اثبات را تمام می کند.

□

(B) اثبات مشابه حالت (A) است.

بنابر قضایای (۲.۳.۴) و (۳.۳.۴) به ترتیب دو کران پایین مختلف برای عدد $\chi_f(G)$ بدست آمد.

$$\chi_f(G) > \frac{\sum D(G)}{|E(G)|} \bullet$$

$$\chi_f(G) > \frac{\sum S(G)}{|E(G)|} \cdot$$

در صورت درست بودن حدس (۴.۲.۴) انتظار می‌رود $\frac{\sum S(G)}{|E(G)|}$ کران پایین بهتری نسبت به کران پایین بیان شده در [۱] برای $\chi_f(G)$ باشد.

فصل ۵

روش جستجوی همسایگی متغیر
برای مسئله رنگ آمیزی مجموع

همانطور که در فصل (۲) بررسی شد، مسئله رنگ آمیزی مجموع یک مسئله NP -سخت است و بنابر روال معمول و منطقی در مواجهه با این گونه مسائل، در این فصل به ارائه روش حل برای این مسئله به منظور یافتن جواب‌های نزدیک بهینه می‌پردازیم. همانند سایر مسائل NP روش‌های مختلفی برای رویارویی با این مسئله ارائه شده‌است. نمونه‌هایی از این روش‌ها که برای حل تقریبی مسئله رنگ آمیزی مجموع مورد توجه قرار گرفته‌است را در انتهای فصل (۳) بیان کردیم. روش جستجوی همسایگی متغیر^۱ از جمله روش‌هایی است که از آن‌ها با عنوان روش‌های فرا ابتکاری^۲ یاد می‌شود. این روش برای دسته وسیع و گسترده‌ای از مسائل بهینه سازی پیچیده مورد توجه قرار گرفته‌است. مسئله رنگ آمیزی گراف‌ها هم از این موضوع مستثنی نبوده و در مقالات مختلفی به این مسئله با دید و رویکرد جستجوی همسایگی متغیر نگریسته شده‌است.

بنابر توضیحات بیان شده در این فصل می‌خواهیم به ارائه روش جستجوی همسایگی متغیر برای حل مسئله رنگ آمیزی مجموع بپردازیم. همانطور که از نام مشخص است این روش بسیار به نوع همسایگی مفروض برای مسئله متکی است. بنابراین قبل از هر موضوعی باید فضای موجه مسئله را معرفی کنیم تا بتوانیم بر اساس آن همسایگی‌ها و در نهایت روش جستجو را توضیح دهیم. لذا قبل از ورود به موضوع اصلی این فصل ابتدا به معرفی مدل غیر خطی عدد صحیح مسئله رنگ آمیزی مجموع می‌پردازیم و سپس سایر مطالب را به ترتیب جلو می‌بریم. لازم به ذکر است کلیه گزاره‌ها، قضایای و مطالب این فصل که منبع آن ذکر نشده‌است، از مرجع [۴۶] اتخاذ شده‌اند.

۱.۵ مدل ریاضی مسئله MSC

در مدل سازی مسایل رنگ آمیزی گراف بصورت فرم برنامه ریزی عدد صحیح اغلب از متغیرهای صفر و یک x_{uj} استفاده می‌شود. این متغیر به معنای آنست که آیا رنگ j ام به راس u ام تخصیص داده شده است یا خیر؟ همچنین برای اینکه تخصیص اعداد به راس‌ها مجاز باشد باید متغیرهای فوق در دسته قیود $x_{uj} + x_{vj} \leq 1$ برای هر یال $(u, v) \in E$ و برای هر $1 \leq j \leq n$ بایست صدق کند. به عنوان مثال مدل ریاضی زیر فقط فضای شدنی مسئله رنگ آمیزی گراف را مشخص می‌کند [۳۷]:

$$x_{uj} + x_{vj} \leq 1 \quad : \forall (u, v) \in E, \forall j$$

$$\sum_j x_{uj} = 1 \quad : u = 1, 2, \dots, n$$

$$x_{uj} \in \{0, 1\}$$

همانطور که ملاحظه می‌شود تعداد متغیرها و قیود در این نوع مدل ریاضی زیاد است. مثلاً برای گراف پترسن (شکل ۴.۴) با ۱۰ راس و ۱۵ یال، مدل ریاضی فوق دارای ۱۶۰ محدودیت و ۱۰۰

¹Variable Neighborhood Search

²Metaheuristic Methods

متغیر است.

بدیهی است که روش حل هر مسئله بهینه‌سازی بسیار به این موضوع وابسته است که ما روی چه نوع مدل ریاضی صحبت می‌کنیم. لذا در این قسمت مدل ریاضی دیگری از مسئله *MSC* با تعداد متغیرها و محدودیت‌های کمتر ولی غیر خطی، ارائه می‌کنیم که بر روش حلی که در ادامه بیان می‌شود، تاثیر گذار است.

۱.۱.۵ مدل ریاضی غیر خطی عدد صحیح مسئله *MSC*

گراف ساده بدون دور و جهت $G = (V, E)$ با مجموعه رئوس $\{v_1, v_2, \dots, v_n\}$ را در نظر بگیرید. اکنون برای هر $i = 1, \dots, n$ متغیر x_{v_i} یا به اختصار x_i را رنگ راس i ام تعریف کنید.

بنابراین مدل برنامه ریزی غیر خطی عدد صحیح مسئله *MSC* بصورت زیر است:

$$\begin{aligned} & \text{Min} \quad \sum_{i=1}^n x_i \\ & \text{s.t.} \\ & |x_i - x_j| \geq 1 \quad : \forall (i, j) \in E \\ & 1 \leq x_i \leq n, \text{ integer} \quad : i = 1, 2, \dots, n \end{aligned}$$

با تغییر متغیر مناسب و حذف قيود زاید خواهیم داشت:

$$\begin{aligned} P) \quad & \text{Min} \quad \sum_{i=1}^n x_i \\ & \text{s.t.} \\ & |x_i - x_j| \geq 1 \quad : \forall (i, j) \in E \quad (1.5) \\ & x_i \geq 0, \text{ integer} \quad : i = 1, 2, \dots, n \quad (2.5) \end{aligned}$$

هر بردار n تایی مثل $X = (x_1, x_2, \dots, x_n)$ که در محدودیت های (۱.۵) و (۲.۵) صدق کند، یک جواب شدنی یا یک رنگ آمیزی مجاز برای گراف G یا به اختصار یک جواب مسئله *MSC* است.

۲.۱.۵ افزایش فضای شدنی

در مسئله P فرض کنید نماد قدر مطلق را از محدودیت‌های (۱.۵) حذف کنیم، داریم:

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^n x_i \\ \text{s.t.} \quad & \\ & x_i - x_j \geq 1 \quad : \forall (i, j) \in E \end{aligned} \quad (3.5)$$

$$x_i \geq 0, \text{integer} \quad : i = 1, 2, \dots, n \quad (4.5)$$

صورت ماتریسی مسئله فوق به شکل زیر است:

$$\begin{aligned} \text{ILPA)} \quad \text{Min} \quad & 1 \cdot X^t \\ \text{s.t.} \quad & \end{aligned}$$

$$AX \geq 1 \quad (5.5)$$

$$X \geq 0, \text{integer} \quad (6.5)$$

که در آن $A_{m \times n}$ ماتریسی است که سطرهای آن متناظر با یال‌ها و ستون‌های آن متناظر با راس‌های G است. در واقع با حذف علامت قدر مطلق از محدودیت‌های (۱.۵)، بسیاری از جواب‌های مسئله P نادیده گرفته می‌شود و فقط جواب‌هایی که در آنها $x_i > x_j$ برای یال $(i, j) \in E$ در نظر گرفته می‌شود. اکنون D را گراف جهت‌گذاری شده‌ای در نظر بگیرید که از G به صورت زیر بدست می‌آید. جهت یال (i, j) از j به i است، هرگاه $x_j < x_i$. بنابراین اگر در ماتریس A ، سطر r ام متناظر با یال e_r و ستون s ام متناظر با راس v_s باشد، آنگاه درآیه a_{rs} از آن بصورت زیر خواهد بود:

$$a_{rs} = \begin{cases} 1 & \text{اگر } v_s \text{ راس انتهایی یال جهت دار } e_r \text{ باشد} \\ -1 & \text{اگر } v_s \text{ راس ابتدایی یال جهت دار } e_r \text{ باشد} \\ 0 & \text{در غیر اینصورت} \end{cases}$$

همانطور که ملاحظه می‌شود ماتریس A همان ترانزاده ماتریس وقوع گراف^۳ جهت‌دار D است. بدیهی است که متناظر با هر جهت‌گذاری D برای گراف G یک ماتریس منحصر به فرد مثل A وجود دارد و برعکس.

از طرف دیگر سیستم $AX \geq 1, X \geq 0$ به دلیل یکپارچگی کلی بودن^۴ ماتریس ضرایب A و عدد صحیح بودن بردار سمت راست، همواره دارای جواب راسی صحیح و در نتیجه جواب بهینه صحیح

³Incidence Matrix

⁴Totally Unimodularity

است [۴۱] و لذا شرط صحیح بودن برای مسئله ILP_A را می‌توان در نظر نگرفت (آزاد سازی^۵ محدودیت مربوط به متغیر) و مسئله برنامه ریزی خطی زیر را مورد بررسی قرار داد.

$$LP_A) \quad Min \quad \mathbf{1} \cdot X^t$$

$$s.t.$$

$$AX \geq \mathbf{1} \quad (۷.۵)$$

$$X \geq \mathbf{0} \quad (۸.۵)$$

حال اگر فرض کنید S فضای شدنی مسئله P و S_A فضای شدنی مسئله LP_A باشد، بدیهی است که $S_A \subseteq S$ و لذا $z^* \leq z_A^*$ (و z^* به ترتیب مقدار بهینه مسئله P و LP_A است) بنابراین با حل مسئله LP_A که یک برنامه ریزی خطی است، می‌توان یک کران بالا برای مسئله اصلی بدست آورد. البته این کران به ماتریس A و به تبع آن به جهت‌گذاری متناظر D بستگی دارد. اکنون بنا بر مطالب بیان شده، به ارایه نتایجی در قالب چند گزاره می‌پردازیم که در اینجا مفید به نظر می‌رسند.

گزاره ۱.۱.۵. فرض کنید D جهت‌گذاری و A ماتریس وقوع متناظر با آن برای گراف $G = (V, E)$ با n راس و m یال باشد، در اینصورت مجموعه $S_A = \{X \in \mathbb{R}^n | AX \geq \mathbf{1}, X \geq \mathbf{0}\}$ ناتهی است، اگر و فقط اگر جهت‌گذاری D شامل دور جهت دار نباشد.

برهان. فرض کنید $X = (x_1, x_2, \dots, x_n)^t \in S_A$ ، در اینصورت جهت‌گذاری D دور جهت دار ندارد. زیرا اگر به برهان خلف فرض کنیم دور جهت داری مثل $c = (v_1, v_2, \dots, v_{t-1}, v_t = v_1)$ داشته باشد، آنگاه چون X در S_A قرار دارد، بنابراین خواهیم داشت:

$$x_{v_1} < x_{v_2} < x_{v_3} < \dots < x_{v_{t-1}} < x_{v_t} = x_{v_1}$$

که تناقض است. بنابراین D شامل دور جهت داری نیست. برعکس، فرض کنید D جهت‌گذاری بدون دور جهت دار متناظر با A باشد. نشان می‌دهیم مجموعه S_A ناتهی است. زیرا در غیر اینصورت مسئله LP_A نشدنی خواهد بود و لذا دوگان آن یعنی DLP_A که به صورت زیر است، نامتناهی یا نشدنی خواهد بود [۵].

$$DLP_A) \quad Max \quad \mathbf{1} \cdot Y$$

$$s.t.$$

$$YA \leq \mathbf{1} \quad (۹.۵)$$

$$Y \geq \mathbf{0} \quad (۱۰.۵)$$

^۵relaxation

اما فضای شدنی مسئله فوق دارای جواب بدیهی $(0, \dots, 0) = Y = (y_1, y_2, \dots, y_m)$ است و لذا DLP_A نمی‌تواند نشدنی باشد پس نامتناهی است. بنابراین جهت شدنی راسی دور شونده ای مثل $d \in \mathbb{R}^m$ وجود دارد بطوریکه $d \neq 0, d \geq 0, d \leq 0$ [۵]. بدون از دست دادن کلیت فرض کنید d_1, d_2, \dots, d_s مولفه‌های مثبت بردار d و e_1, e_2, \dots, e_s یال‌های متناظر با آنها باشند. و همچنین فرض کنید p طولانی‌ترین مسیر جهت دار که زیر گراف القا شده توسط یال‌های e_1, e_2, \dots, e_s باشد. به دلیل بیشینه بودن طول مسیر p ، راس انتهایی آن نمی‌تواند یال خروجی داشته باشد و از بین یال‌های ورودی به این راس یالی وجود دارد که مولفه متناظر با آن در d مثبت است. این مطلب با شدنی بودن سیستم $dA \leq 0, d \geq 0, d \neq 0$ در تناقض است. بنابراین نتیجه می‌شود مجموعه S_A ناتهی است. \square

گزاره ۲.۱.۵. اگر D_1 و D_2 دو جهت‌گذاری متمایز و A_1 و A_2 دو ماتریس وقوع متناظر با این جهت‌گذاری‌ها برای گراف G باشند، آنگاه $S_{A_1} \cap S_{A_2} = \emptyset$

برهان. بنابه نحوه جهت‌گذاری گراف بدیهی است. \square

از دو گزاره فوق نتیجه می‌شود که برای گراف G اگر D_1, D_2, \dots, D_t تمام جهت‌گذاری‌های بدون دور و A_1, A_2, \dots, A_t ماتریس‌های وقوع متناظر با این جهت‌گذاری‌ها باشد، آنگاه S_{A_i} ‌ها مجموعه S را افزای می‌کند، یعنی:

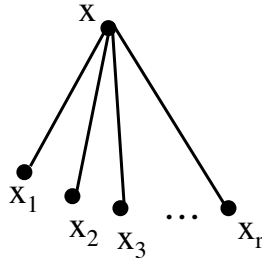
$$i \neq j : S_{A_i} \cap S_{A_j} = \emptyset \quad (1)$$

$$S = \bigcup_{i=1}^t S_{A_i} \quad (2)$$

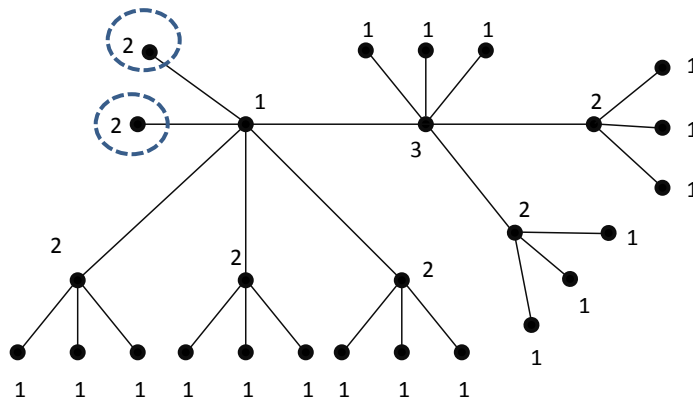
در واقع با این روش فضای موجه یک مسئله NP -کامل به تعدادی زیر مسئله برنامه ریزی خطی شکسته شد ولی تعداد این زیر مسئله‌ها بسیار زیاد است ($t \in o(2^{|E|})$) و با استفاده از روش‌های ابتکاری می‌توان تعدادی از این زیر مسئله‌ها را که باعث بهبود جواب بهینه نمی‌شوند، مورد بررسی قرار نداد. به عنوان مثال می‌توان در مورد رنگ برگ‌ها در رنگ آمیزی بهینه گزاره زیر را بیان کرد.

گزاره ۳.۱.۵. در رنگ آمیزی بهینه برای مسئله MSC رنگ برگ‌های مجاور با یک راس، باهم برابر و حداکثر دو است.

برهان. همانطور که در شکل (۱.۵) ملاحظه می‌شود، اولاً بدیهی است که رنگ رئوس x_1, x_2, \dots, x_r که مجاور راس x هستند، در رنگ آمیزی بهینه باید برابر باشند. (رنگ برگ‌های مجاور با یک راس، در جواب بهینه دارای یک وضعیت و مقدار هستند). همچنین فرض کنید $c : V \rightarrow I$ تابع رنگ آمیزی بهینه باشد و راسی مثل x_1 موجود باشد که $c(x_1) \geq 3$. از طرف دیگر $c(x) \neq c(x_1)$ ، یعنی $c(x) < c(x_1)$ یا $c(x) > c(x_1)$. با اندکی بررسی می‌توان در هر دو حالت رنگ آمیزی جدیدی معرفی کرد که دارای مجموع رنگ کمتری نسبت به c است و این با بهینه بودن c تناقض دارد. بنابراین رنگ بهینه رئوس تنها با هم برابرند و بعلاوه حداکثر دو می‌باشند. \square



شکل ۱.۵: برگ‌های مجاور با راس x در رنگ آمیزی بهینه وضعیت یکسانی دارند



شکل ۲.۵: برگ‌هایی که حتما باید دارای رنگ ۲ در رنگ آمیزی بهینه باشند

شکل (۲.۵) مثالی از یک درخت به همراه رنگ آمیزی مجموع بهینه آن است که در آن رنگ بعضی از برگ‌ها حتما باید ۲ باشد (با یک بررسی ساده این موضوع قابل تحقیق است). با توجه مطالب بیان شده در این قسمت، جهت یافتن جواب قابل قبولی برای مسئله P نیاز به حل زیر مسئله های خطی LP_A است. با استفاده از گزاره زیر می‌توان جواب بهینه مسئله LP_A را بدون درگیر شدن با سیستم $AX \geq 1$ و $X \geq 0$ بدست آورد.

گزاره ۴.۱.۵. گراف $G = (V, E)$ را در نظر بگیرید. فرض کنید D یک جهت گذاری از G بدون دور جهتدار و A ماتریس متناظر با آن باشد. در اینصورت $X^* = (x_1, \dots, x_n)$ جواب بهینه منحصر به فرد مسئله LP_A خواهد بود اگر و فقط اگر x_i (رنگ راس v_i)، برابر طول بزرگترین مسیر جهتدار منتهی به v_i باشد.

برهان. به برهان خلف فرض کنید X^* جواب بهینه مسئله LP_A نباشد. لذا فرض کنید جواب دیگری مثل $Y = (y_1, \dots, y_n)$ موجود باشد بطوری که

$$\sum_{i=1}^n y_i < \sum_{i=1}^n x_i$$

بدیهی است که حداقل راسی مثل v_l موجود است بطوری که در آن $y_l < x_l = k$. بنا به فرض

$$p = (v_1 v_2 \dots v_{k+1} = v_l)$$

بزرگترین مسیر جهتدار منتهی به راس v_l با طول k متشکل از $k+1$ راس است. از طرف دیگر، چون $Y^* \in S_A$ پس داریم:

$$0 \leq y_1 < y_2 < \dots < y_{k+1} = y_l < x_l = k$$

یعنی بازه $[0, k)$ دارای $k+1$ عدد صحیح متمایز است و این امکان ندارد. بنابراین X^* جواب بهینه مسئله LP_A است.

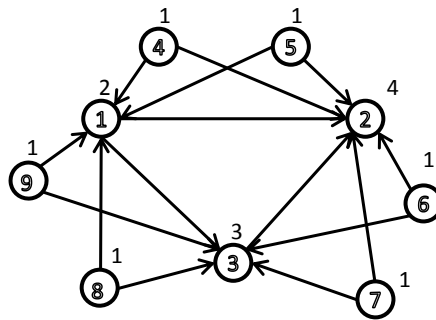
برای اثبات منحصر به فرد بودن فرض کنید $X' \neq X^*$ موجود است بطوری که $1.X' = 1.X^*$. بنابراین بدون از دست دادن کلیت، راسی مثل v_s موجود است بطوریکه $x'_s < x_s$. مجدداً با استفاده از استدلال ابتدای اثبات نتیجه می‌شود که این موضوع امکان ندارد. لذا منحصر به فردی جواب X^* اثبات می‌شود.

انکون فرض کنید مسئله LP_A دارای جواب بهینه منحصر به فرد $X^* = (x_1, \dots, x_n)$ باشد. به دلخواه راسی مثل v_l با رنگ x_l را در نظر بگیرید. در گراف جهت گذاری شده، مجموعه رئوسی را در نظر بگیرید که قبل از v_l هستند. بدیهی است که در این مجموعه حداقل راسی مثل v_{l-1} با رنگ $x_l - 1$ وجود دارد. زیرا در غیر این صورت X^* جواب بهینه نخواهد بود. با همین استدلال راسی مثل v_{l-2} قبل از v_l با رنگ $x_l - 2$ موجود است. لذا مسیر $p = (v_1 v_2 \dots v_{l-1} v_l)$ موجود است که در آن رنگ راس v_1 صفر است و راسی قبل از آن وجود ندارد. بدیهی است که این مسیر، بزرگترین مسیر جهت

□

دار منتهی به v_l با طول x_l است.

برای ایجاد درک بهتر از گزاره اخیر شکل (۳.۵) را ملاحظه کنید. اعدادی که داخل و روی رئوس نوشته شده اند به ترتیب شماره و رنگ رئوس مورد نظر هستند. در این گراف طبق جهت گذاری داده شده، رنگ آمیزی بهینه متناظر با آن ارایه شده است. لازم به ذکر است بدون از دست دادن کلیت بازه مربوط به انتخاب رنگ ها بجای $0, 1, \dots, n-1$ از بازه $1, 2, \dots, n$ انتخاب شده است.



شکل ۳.۵: جهت گذاری مفروض و رنگ آمیزی بهینه متناظر با آن

۲.۵ معرفی همسایگی‌های N_k

همانطور که قبلا بیان شد، صحبت از جستجوی فضای شدنی، بدون اطلاع داشتن از این که جستجو بر اساس چه همسایگی اتفاق می افتد، بی معناست. در واقع حل یک مسئله ترکیباتی ارتباط تنگاتنگی با ساختار همسایگی تعریف شده برای آن دارد. تعریف نقطه بهینه محلی و بهینه سراسری در تعریف زیر آمده است.

تعریف ۱.۲.۵ [۴۲] مسئله کمینه سازی (F, c) با فضای شدنی F و تابع هزینه c را در نظر بگیرید. همچنین فرض کنید N همسایگی تعریف شده برای این مسئله باشد. جواب شدنی $f \in F$ را یک **بهینه محلی** نسبت به همسایگی N می نامیم، هرگاه داشته باشیم:

$$c(f) \leq c(x) \quad : \forall x \in N(f)$$

همچنین جواب موجه f برای مسئله (F, c) **بهینه سراسری** است، هرگاه داشته باشیم:

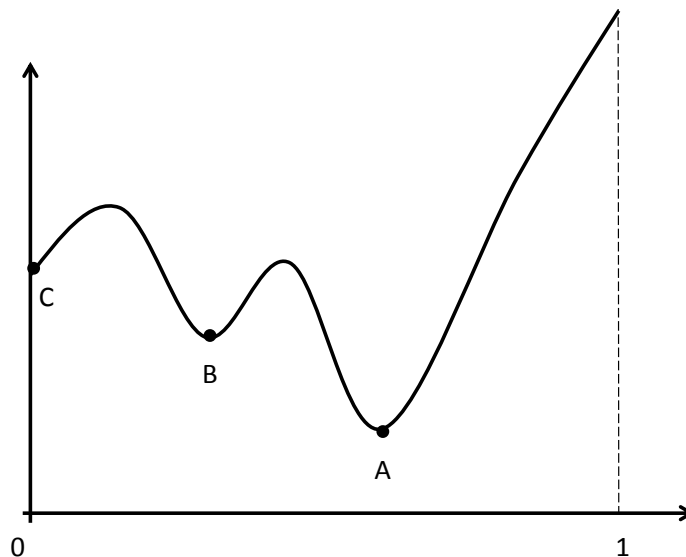
$$c(f) \leq c(x) \quad : \forall x \in F$$

به عنوان نمونه فرض کنید می خواهیم نقطه کمینه تابع نمایش داده شده در شکل (۴.۵) را در بازه $[0, 1]$ بدست بیاوریم. معروف ترین همسایگی برای بهینه سازی یک بعدی، همسایگی با فاصله

اقلیدسی است که برای هر $\epsilon > 0$ بصورت زیر تعریف می شود:

$$N_\epsilon(f) = \{x | x \in [0, 1]; |x - f| < \epsilon\}$$

اگر ϵ به قدر کافی کوچک انتخاب شود، هر سه نقطه A و B و C می توانند نسبت به همسایگی های



شکل ۴.۵: مسئله بهینه سازی یک بعدی با همسایگی فاصله اقلیدسی

متناظر بهینه محلی باشند، در حالیکه از بین آنها فقط نقطه A بهینه سراسری است. اکنون اگر ϵ را که همان شعاع همسایگی $N_\epsilon(\cdot)$ است، بزرگتر مساوی با ۱ انتخاب شود، نقطه A یک بهینه محلی خواهد بود که سراسری هم هست. بنابراین انتخاب همسایگی در جواب خروجی مسئله تاثیر گذار است. به عنوان نمونه دیگر مسئله یافتن مینیمم درخت فراگیر^۶ برای گراف G که دارای یالهای وزن دار است را در نظر بگیرید. هدف این مسئله، یافتن درخت فراگیر با کمترین هزینه (مجموع وزن یالها) است. در این مسئله فرض کنید T یک درخت فراگیر برای گراف زمینه G باشد. در [۴۲] همسایگی برای این مسئله بصورت زیر معرفی شده است:

$$N(T) = \{ \text{می آیند بدست } T \text{ از یال یک حذف و یال یک افزودن از که درخت هایی تمام مجموعه} \}$$

بازای همسایگی فوق هر بهینه محلی مسئله MST یک بهینه سراسری است [۴۲]. در این نوع همسایگی ها که از آن با عنوان همسایگی کامل^۷ یاد می شود، هر بهینه محلی بهینه سراسری هست. در حقیقت درستی و صحت برخی الگوریتم هایی که جواب دقیق را برای مسئله MST بدست می دادند،

^۶Minimum Spanning Tree

^۷Exact Neighborhood

مثل الگوریتم‌های کروسکال^۸ و پریم^۹، به دلیل این خاصیت مهم همسایگی $N(T)$ است [۲۵]. همسایگی $N_\epsilon(\cdot)$ نمونه‌ای از همسایگی است که برای $\epsilon > 0$ به قدر کافی کوچک ناکامل است. ولی همین همسایگی برای $\epsilon \geq 1$ کامل است. بنابراین توضیحات فوق، طراحی و تعریف مناسب همسایگی برای یک مسئله بهینه‌سازی، در فرایند طراحی الگوریتم حل آن تاثیر بسزایی دارد.

۱.۲.۵ همسایگی برای مسئله MSC

برای مسئله MSC همسایگی‌های مختلفی در مقالات معرفی شده است. برای مثال رجوع کنید به [۱۹، ۲۱، ۲۴، ۶]. در اغلب آنها همسایگی‌های تعریف شده مبتنی بر مجموعه‌های مستقلی است که از آن به عنوان کلاس رنگی یاد می‌شود. ولی همسایگی تعریف شده در اینجا مبتنی بر مدل بندی است که در بخش (۱.۵) بیان شد. همانطور که در بخش (۱.۵) اشاره شد هر بردار n تایی مثل $X = (x_1, x_2, \dots, x_n)$ که در محدودیت‌های (۱.۵) و (۲.۵) صدق کند، یک جواب شدنی یا یک رنگ آمیزی مجاز برای گراف G یا به اختصار یک جواب مسئله MSC است. بنابراین همسایگی‌هایی که در ادامه معرفی می‌شوند، بایست حول این بردارهای n تایی تعریف شوند. فرض کنید X یک رنگ آمیزی مجاز برای مسئله P و S فضای شدنی آن باشد. همسایگی $N_k(X)$ را برای $k = 1, \dots, n$ بصورت زیر تعریف می‌کنیم.

تعریف ۲.۲.۵. فرض کنید X یک جواب شدنی (رنگ آمیزی مجاز) برای مسئله P با فضای شدنی S باشد. در این صورت مجموعه $N_k(X)$ را همسایگی k ام بردار X می‌نامیم و بصورت زیر تعریف می‌کنیم:

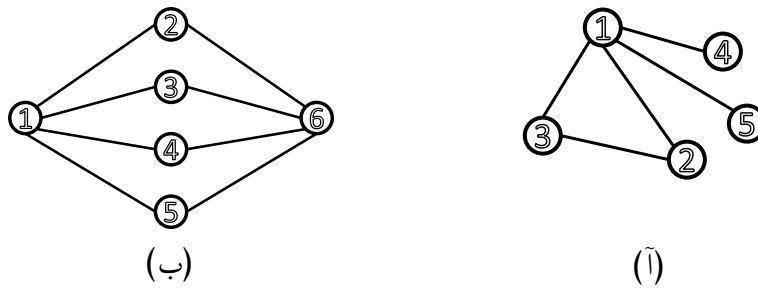
$$N_k(X) = \{Y \in S \mid Y \text{ حداکثر } k \text{ مولفه } X \text{ از مولفه } k \text{ در حداکثر } k \text{ مولفه}\} \quad (۱۱.۵)$$

در حقیقت $N_k(X)$ شامل رنگ آمیزی‌هایی از مجموعه S است که در آنها حداکثر رنگ k راس بیشتر از X است. به عنوان مثال گراف شکل (۱۵.۵) را در نظر بگیرید. در این گراف اعدادی که داخل رئوس نوشته شده اند شماره رئوس هستند. جواب $X = (1, 2, 3, 2, 2)$ عضوی از مجموعه S است.^{۱۰} بدیهی است که $N_1(X) = \{X\}$ و از طرفی $Y = (2, 1, 3, 1, 1) \in N_1(X)$ یک بهینه محلی نسبت به همسایگی N_1 و همچنین بهینه سراسری است. با اندک بررسی نتیجه می‌شود که همسایگی N_1 برای این گراف یک همسایگی کامل است (یعنی هر بهینه محلی نسبت به N_1 ، بهینه سراسری است). ولی برای گراف شکل (۵.۵) اینطور نیست. در واقع برای این گراف، $X = (1, 2, 2, 2, 2, 1)$ یک بهینه محلی است (نسبت به N_1) که سراسری نیست. زیرا جواب

^۸Kruskal

^۹Prime

^{۱۰}بدون از دست دادن کلیت، رنگ آمیزی رئوس، بجای عدد صفر از عدد ۱ آغاز شده‌است.



شکل ۵.۵: همسایگی $N_1(X)$

همانطور که می‌دانیم در مسئله رنگ آمیزی مجموع بر خلاف مسئله رنگ آمیزی معمولی، افزایش رنگ رئوس می‌تواند باعث بهبود جواب بهینه (کاهش مجموع رنگ‌ها) شود و همین موضوع، انگیزه اصلی ما برای معرفی همسایگی N_k بصورت آنچه در رابطه (۱۱.۵) آمده، شد. پرواضح است که اندازه همسایگی N_k با افزایش k بزرگ و بزرگتر می‌شود. یعنی برای یک جواب X داریم:

$$N_0(X) \subseteq N_1(X) \subseteq \dots \subseteq N_k(X)$$

بنابراین بدیهی است که اگر X یک جواب بهینه محلی نسبت به همسایگی N_k باشد، آنگاه X نسبت به همسایگی‌های N_i برای $0 \leq i \leq k-1$ نیز بهینه محلی است. لذا برای یک X داده شده، یافتن کمترین مقدار k که بازای آن مطمئن باشیم همسایگی $N_k(X)$ کامل باشد، حائز اهمیت است. بدیهی است که بیشترین مقدار k حداکثر $|V| - 1$ است. در واقع همسایگی N_k برای $k = |V| - 1$ کامل است و این اصلاً کران خوبی برای k نیست. زیرا یافتن یک جواب بهینه محلی نسبت به این همسایگی، برای گراف‌هایی با اندازه نه چندان بزرگ هم، کار ساده‌ای نیست و محاسبات زیادی را تحمیل می‌کند. محاسباتی که شاید چندان ضروری هم نباشد. لذا برای ما مقدور نیست که در افزایش اندازه همسایگی‌ها محدودیتی نداشته باشیم. هر چه بتوان فضای مربوط به همسایگی N_k را بدون نیاز به اندازه همسایگی‌ها سریعتر بررسی کنیم، فرایند جستجو دارای سرعت بیشتری خواهد بود. برای انجام این مهم، در ادامه مفهومی جدید در گراف بنام نگهدارندگی را معرفی می‌کنیم که در تسریع فرایند جستجو و بهبود جواب‌ها بسیار راه‌گشاست.

۳.۵ نگهدارندگی و کاهندگی

در فرایند جستجوی فضای موجه به جواب‌هایی (بردارهای رنگ آمیزی) بر می‌خوریم که در آنها کاهش برخی از رئوس مستلزم افزایش رنگ برخی دیگر از رئوس است. این رئوس که از کاهش رنگ سایر رئوس ممانعت به عمل می‌آورند، رئوسی هستند که به آنها رئوس (مجموعه) نگهدارنده

می‌گوییم.

هدف اصلی از ارایه چنین تعریفی در اینجا آن است که در واقع با شناسایی این رئوس بتوان به کاهش تابع هدف و بهبود جواب کمک کرد. البته گاهی اوقات این افزایش و کاهش رنگ رئوس، در نهایت منجر به افزایش تابع هدف می‌شود و این خوب نیست. در اینجا است که باید سراغ رئوس یا مجموعه‌های نگهدارنده‌ای برویم که افزایش رنگ آنها منجر به کاهش تابع هدف می‌شود. برای این منظور در قسمت بعد این مفهوم را تعریف می‌کنیم و توضیح خواهیم داد که چگونه این مفهوم به افزایش سرعت پیمایش فضای همسایگی‌ها کمک خواهد کرد.

۱.۳.۵ مجموعه‌های نگهدارنده و کاهنده

تعریف ۱.۳.۵. راس نگهدارنده. فرض کنید X بردار رنگ آمیزی مجاز برای گراف G باشد. راس u یک راس نگهدارنده است هرگاه وجود داشته باشد $v \in N(u)$ بطوریکه:

$$\text{الف) } X(u) < X(v)$$

ب) برای هر راس $u \neq w \in N(v)$ داشته باشیم $X(w) \neq X(u)$.

در این حالت می‌گوییم راس u راس v را نگه داشته‌است و آنرا با نماد $v \nearrow u$ نمایش می‌دهیم.

تعریف ۲.۳.۵. مجموعه k -نگهدارنده. بردار رنگ آمیزی مجاز X و زیر مجموعه k عضوی $S = \{t_1, \dots, t_k\}$ از مجموعه رئوس گراف G را در نظر بگیرید. گوییم S ، k -نگهدارنده است هرگاه راسی مثل $v \in N(S)$ ^{۱۱} و اندیس r ای $(1 \leq r \leq k)$ موجود باشد بطوریکه:

$$\text{الف) } X(t_r) < X(v)$$

ب) برای هر راس $t_r \neq w \in N(v) \setminus S$ داشته باشیم $X(w) \neq X(t_r)$

بدیهی است که هر راس نگهدارنده خود یک مجموعه ۱-نگهدارنده است. برای روشن تر شدن این مفاهیم جدید گراف شکل (۵.۵) و جواب $X = (1, 2, 2, 2, 2, 1)$ را در نظر بگیرید. همانطور که ملاحظه می‌شود در این حالت هیچ کدام از رئوس این گراف نگهدارنده نیستند یا بعبارت دیگر هیچ مجموعه ۱-نگهدارنده ندارد در حالیکه مجموعه $H = \{1, 6\}$ یک مجموعه ۲-نگهدارنده است.

تعریف ۳.۳.۵. مجموعه k -کاهنده (کاهش دهنده). بردار رنگ آمیزی مجاز X و زیر مجموعه k عضوی S از مجموعه رئوس گراف G را در نظر بگیرید. گوییم S ، k -کاهنده است هرگاه اولاً k -نگهدارنده باشد و ثانیاً ترتیبی از افزایش رنگ رئوس S موجود باشد بطوریکه منجر به کاهش مجموع رنگ رئوس شود.

تعاریف فوق ما را به نتایجی نسبتاً بدیهی در قالب دو گزاره راهنمایی می‌کند که ارتباط این مفاهیم را با روش جستجوی همسایگی متغیر مبتنی بر همسایگی‌های N_k بیشتر مشخص می‌کند.

^{۱۱} منظور از $N(S)$ رئوس مجاور مجموعه S است.

گزاره ۱.۳.۵. جواب X نسبت به همسایگی N_k بهینه محلی است، اگر و فقط اگر گراف G دارای هیچ مجموعه k - کاهنده‌ای نسبت به X نباشد.

گزاره ۲.۳.۵. اگر جواب X نسبت به هر $k \geq 1$ دارای هیچ مجموعه k - کاهنده‌ای نباشد، آنگاه جواب X بهینه سراسری است.

به عنوان مثال گراف شکل (۵.۵ب) بازای بردار رنگ آمیزی $X = (1, 2, 2, 2, 2, 1)$ ، دارای هیچ راس نگهدارنده‌ای نیست به تبع آن دارای هیچ مجموعه ۱-کاهنده‌ای هم نخواهد بود. بنابراین این جواب یک بهینه محلی نسبت به N_1 است ولی مجموعه $H = \{1, 6\}$ یک مجموعه ۲-کاهنده است و بنابر گزاره (۲.۳.۵) X جواب بهینه سراسری نخواهد بود.

همچنین بنا بر تعاریف مجموعه‌های نگهدارنده و کاهنده بدیهی است که اگر گراف G نسبت به جواب X دارای هیچ مجموعه k -نگهدارنده نباشد، آنگاه شامل هیچ مجموعه k -کاهنده‌ای هم نخواهد بود و لذا بنا بر گزاره (۱.۳.۵) جواب X نسبت به همسایگی N_k یک بهینه محلی است.

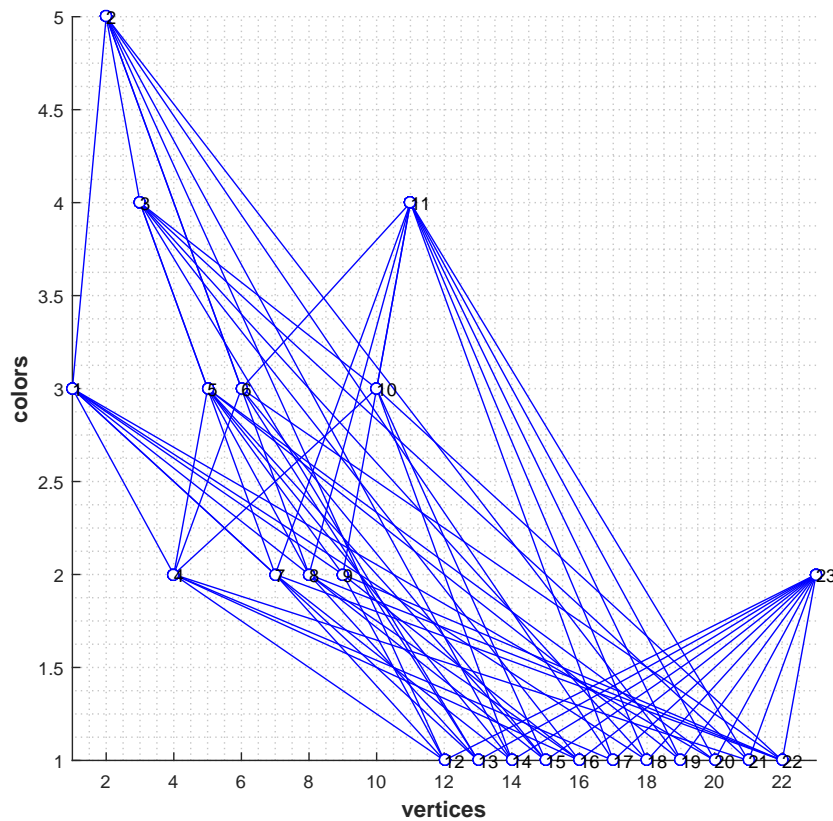
مسئله یافتن مجموعه‌های کاهنده با کوچکترین اندازه از جمله چالش‌های مهم در این فصل است. زیرا با افزایش k و به تبع آن افزایش اندازه مجموعه‌ها ممکن است موجب تحمیل محاسبات غیر ضروری و در نتیجه موجب کاهش سرعت روش مورد نظر شود. متأسفانه یافتن مجموعه‌های کاهنده به اتفاق ترتیب افزایش رنگ رئوس (که سبب کاهش مجموع رنگ‌های رئوس شود) کار ساده‌ای نیست. زیرا در این حالت وضعیت‌های مختلفی ایجاد می‌شود که عدم بررسی آنها ممکن است ما را به جواب‌های نامناسبی هدایت کند.

۲.۳.۵ ارتباط مجموعه‌های نگهدارنده و کاهنده با مسئله MSC

بنابر توضیحاتی که در مورد مفهوم نگهدارندگی و کاهندگی بیان شد ممکن است این سوال پیش بیاید که این مفاهیم اساساً چه ارتباطی با جستجوی همسایگی می‌توانند داشته باشند یا بعبارت دیگر، چگونه این مفاهیم به بهبود عملکرد فرایند جستجو کمک می‌کند؟ قبل از آنکه در صدد پاسخ به این سوال براییم، مثالی را جهت روشن تر شدن موضوع بیان می‌کنیم.

ساختمان صد طبقه متروکه‌ای را در نظر بگیرید که قرار است توسط یک گروه تخریب کار ساختمان با مواد منفجره محدود ویران شود. طبیعی است که برای بالابردن بازدهی کار و استفاده کمتر از مواد منفجره، بهتر است این مواد در نقاطی از طبقات ساختمان کار گذاشته شوند که بیشترین ریزش را ایجاد کنند. علاوه بر اینکه مواد منفجره در چه مکانی کار گذاشته شوند، ترتیب ریزش بعد از هر انفجار هم بر ادامه فرایند تخریب بسیار موثر است. حتی اگر در مرحله‌ای ریزشی صورت نگیرد، شناسایی مکان‌های انفجاری که به سست تر شدن ساختمان کمک کنند تا در مرحله بعد ریزش صورت گیرد، مفید است (ایجاد آشفستگی).

اکنون مسئله MSC را در نظر بگیرید. گراف مورد نظر همراه با یک رنگ آمیزی مجاز آن همانند یک ساختمان چند طبقه است بطوری که بعضی از رئوس آن روی بعضی راس یا رئوس دیگر سوار هستند یا بعبارتی بعضی از رئوس نقش نگهدارندگی را برای سایر رئوس دارند. در واقع اعضای مجموعه‌های



شکل ۶.۵: گراف myciel4 با ۲۳ راس و نمودار دو بعدی رنگ راس متناظر با رنگ آمیزی بهینه

کاهنده، همان مکان‌هایی هستند که قرار است مواد منفجره در آن‌ها کار گذاشته شوند. یافتن کمترین تعداد مکان‌های انفجار برای بیشترین میزان تخریب، مثل یافتن یک مجموعه k -کاهنده با کمترین مقدار k نسبت به یک بردار رنگ آمیزی X است بطوریکه بیشترین کاهش را در مجموع رنگ رئوس گراف ایجاد کند. احتمالاً با توضیحات فوق ارتباط بین مفاهیم نگهدارندگی، کاهندگی و کاهش تابع هدف بیشتر مشخص شده است. شکل (۶.۵) مثالی از یک گراف با رنگ آمیزی مجازی برای آن است که در یک نمودار دو بعدی (محور افقی اندیس رئوس و محور عمودی رنگ رئوس) نمایش داده شده است. همانطور که از روی شکل پیداست هر راس با رنگ مشخص، حداقل با یک راس در تمامی رنگ‌های کمتر مجاور است.

۳.۳.۵ ترتیب ریزش

بنابر توضیحات قبل، یافتن روشی جهت استخراج کوچکترین مجموعه‌ی کاهنده با بیشترین کاهش دهنده در میزان تابع هدف، در زمان مناسب کار مهمی است که در این قسمت قصد توضیح آنرا داریم. برای این منظور بنا بر تعریف (۳.۳.۵) هر مجموعه کاهنده قبل از هر چیز باید نگهدارنده باشد. لذا ابتدا باید یک مجموعه نگهدارنده ایجاد کنیم. برای این کار هم از مفهوم نگهدارندگی راسی کمک می‌گیریم

و مجموعه نگهدارنده را از روی رئوس نگهدارنده محاسبه می‌کنیم. پس قبل از هر کاری ابتدا بایست نگهدارنده بودن یا نبودن هر راس گراف نسبت به راس یا رئوس دیگر بررسی شود و تعیین شود کدام راس توسط کدام راس یا رئوس دیگر نگهداشته شده‌است. در اینجا است که به ترتیب‌های مختلفی از رئوس برخورد می‌کنیم که در این ترتیب‌ها هر راس، راس یا رئوس بعدی در ترتیب را نگهداشته‌است. با حذف راس ابتدایی در این ترتیب (ها)، رنگ سایر رئوس ترتیب بصورت سلسله وار کم می‌شود و اصطلاحاً ریزش می‌کند. بعد از تعیین رنگ راس یا رئوس حذف شده، در شرایط مساعد، تابع هدف کاهش پیدا خواهد کرد. این‌گونه ترتیب از رئوس را **ترتیب ریزش می‌نامیم**.

در روند یافتن مجموعه‌های کاهنده این موضوع بسیار اهمیت دارد که ما چه رئوسی را انتخاب کنیم بطوری که بعد از حذف آنها بیشترین ریزش^{۱۲} برای سایر رئوس ایجاد شود. در حقیقت برای ساختن مجموعه‌های کاهنده از رئوسی استفاده می‌شود که دارای بیشترین میزان نگهدارندگی باشد. برای این منظور نیازمند تعریفی روی گراف G هستیم که رابطه نگهدارندگی رئوس را روی آن بتوان بهتر دید. لذا **گراف جهتدار بدون دور^{۱۳} D** را از روی G بصورت زیر تعریف می‌کنیم.

تعریف ۴.۳.۵. گراف $G = (V(G), E(G))$ و رنگ آمیزی مجاز X از آن را در نظر بگیرید. گراف $D = (V(D), E(D), W)$ بصورت زیر تعریف می‌کنیم:

$$V(D) = V(G) \quad (۱)$$

$$E(D) = \{(a, b) \in E(G) \mid a, b \in V(D) \& a \nearrow b\} \quad (۲)$$

$$\forall e = (a, b) \in E(D), w_e = x_b - x_a \quad (۳)$$

که در آن x_a مولفه ای از بردار X متناظر با راس a و w_e مولفه‌ای از بردار W متناظر با یال $e = (a, b)$ است و W بردار وزن یال‌های D است.

با تعریف فوق بدیهی است گراف D یک گراف جهتدار بدون دور و یال چندگانه، طوقه است. روشن است که بین یافتن طولانی‌ترین مسیر جهتدار D نسبت به بردار وزنی تعریف شده W ، و بهترین ترتیب ریزش ارتباط تنگاتنگی وجود دارد.

اگرچه مسئله یافتن مسیر بیشینه در گراف‌ها بطور کلی یک مسئله NP -سخت است ولی همین مسئله برای گراف‌های جهتدار بون دور، خطی است ($O(n)$) [۸]. یعنی الگوریتم با زمان خطی نسبت به اندازه ورودی برای یافتن مسیر بیشینه در گراف‌های جهتدار بدون دور وجود دارد. دلیل این امر به خاطر ویژگی خاصی این گراف‌ها است که از آن با عنوان مرتب سازی توپولوژیکی^{۱۴} یاد می‌شود. مرتب سازی توپولوژیکی برای گراف جهتدار بدون دوری مثل D عبارت است از یک ترتیب از رئوس D در یک خط راست بطوریکه برای هر یال جهت دار (u, v) ، راس u قبل از راس v ظاهر شود.

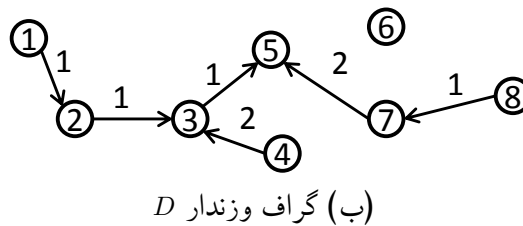
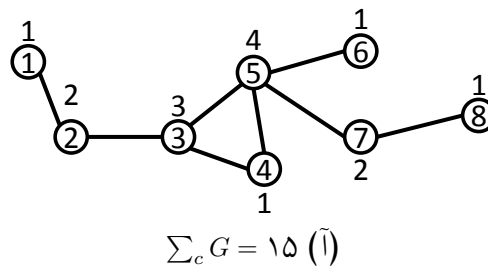
^{۱۲} منظور از ریزش، همان کاهش رنگ رئوس است.

^{۱۳}Directed Acyclic Graph

^{۱۴}Topological Sorting

بدیهی است که مرتب سازی توپولوژیکی زمانی برای یک گراف امکان پذیر است که آن گراف یک گراف جهت دار بدون دور باشد [۸].

در شکل (۷.۵) موضوع ترتیب ریزش مورد بررسی قرار گرفته است. اعدادی که داخل و روی هر یک از راس‌ها قرار گرفته‌اند به ترتیب، شماره و رنگ هر یک از رئوس هستند. در شکل (۷.۵) گراف G به همراه یک ۴-رنگ آمیزی $X = (1, 2, 3, 1, 4, 1, 2, 1)$ با مجموع رنگ ۱۵ نشان داده شده است. همچنین گراف وزندار و جهتدار بدون دور D ، مطابق با تعریف (۴.۳.۵)، در شکل (۷.۵) قابل رویت است. در این شکل سه مسیر جهتدار $P_1 : 1 - 2 - 3 - 5$ ، $P_2 : 4 - 3 - 5$ و



شکل ۷.۵: ترتیب ریزش

$5 - 7 - 8 : P_3$ با طول بیشینه ۳ وجود دارد. مثلاً اگر P_1 را انتخاب و راس ابتدایی آن را موقتاً حذف کنیم، ملاحظه می‌کنیم که رنگ رئوس ۲، ۳، ۵ به ترتیبی که در مسیر آمده است، کاهش می‌یابد. در نهایت رنگ راس ابتدایی مسیر، یعنی راس ۱ را مجدداً تعیین می‌کنیم. اکنون با این تغییرات به جواب $Y = (2, 1, 2, 1, 3, 1, 2, 1)$ می‌رسیم که دارای دو واحد کاهش در تابع هدف نسبت به جواب X است.

بنابراین با انتخاب راس ۱ که در واقع یک مجموعه ۱-نگهدارنده (۱-کاهنده) است، مقدار تابع هدف کمتر شد. اکنون می‌توان همین فرایند را روی جواب جدید اعمال کرد تا آنجا که دیگر بازای جواب مورد نظر، مجموعه ۱-کاهنده‌ای موجود نباشد. در اینجا است که باید اندازه همسایگی را بزرگتر کنیم (افزایش k). این کار تازمانی ادامه پیدا می‌کند که همسایگی‌ها به بزرگترین اندازه خود رسیده باشند ($k = k_{max}$). توجه داشته باشید که صحبت از مفهوم نگهدارندگی (یا کاهندگی) بدون در نظر گرفتن یک بردار رنگ آمیزی، بی‌معناست.

بنابر توضیحات فوق با یافتن ترتیب‌های ریزش خوب (که موجب کاهش تابع هدف شوند)، می‌توان فرایند یافتن جواب بهینه محلی را سرعت بخشید. در حقیقت یافتن ترتیب ریزش مناسب دو فایده مهم دارد: اول اینکه می‌توان از نخستین راس آن در تکمیل مجموعه کاهنده استفاده کرد و ثانیاً ترتیب پیمایش رئوس، بعد از حذف رئوس مجموعه کاهنده را مشخص می‌کند.

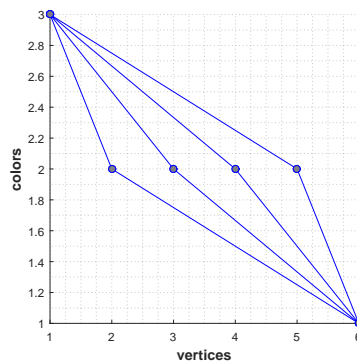
وقتی k به بیشترین مقدار خود رسیده باشد و ترتیب ریزش خوب (کاهش دهنده) موجود نباشد، الگوریتم در یک بهینه محلی نسبت به همسایگی N_k قرار گرفته است (یا در تله بهینه محلی گیر افتاده است). در اغلب روش‌هایی که برای مسائل بهینه سازی طراحی می‌شود، برای خروج از چنین حالتی از مکانیسمی استفاده می‌شود که از آن با عنوان خروج از بهینه محلی یاد می‌کنند. (نمونه‌ای از این تکنیک در مورد مسئله رنگ آمیزی مجموع را می‌توانید در مرجع [۶] ملاحظه بفرمایید.) بنابر توضیحات فوق در قسمت بعدی قصد داریم مکانیسمی برای خروج از بهینه محلی نسبت به همسایگی‌های N_k معرفی کنیم.

۴.۳.۵ مکانیسم آشفستگی برای خروج از بهینه محلی

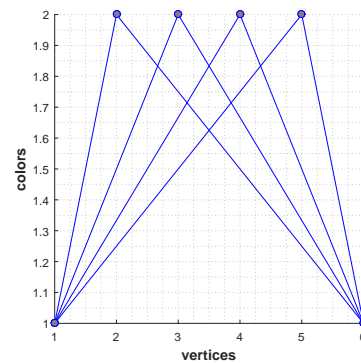
حالتی را در نظر بگیرید که یا اصلاً ترتیب ریزشی وجود ندارد یا اگر هم وجود داشته باشد، منجر به کاهش تابع هدف نمی‌شود. غالباً در این حالات افزایش میزان k هم مثر ثمر نیست و بجز تحمیل بار محاسبات زمانبر، نتیجه‌ای ندارد. لذا باید با ایجاد کمترین تغییراتی در گراف، رئوس نگهدارنده ایجاد کنیم تا به تبع آن ترتیب‌های ریزش مناسبی ایجاد شوند. برای خروج از این گونه حالات از روشی استفاده می‌کنیم که آنرا **مکانیزم آشفستگی** نام نهاده‌ایم.

در واقع در این حالات گراف D متناظر، گراف تهی یا گرافی با چگالی پایین است. در این موارد الگوریتم جستجوی محلی بین دو یا چند جواب بهینه محلی حرکت می‌کند و الگوریتم در یک ایستایی و رکود گرفتار می‌شود. برای خروج از چنین حالتی باید با تغییری که در بردار X می‌دهیم، زمینه را برای ایجاد ترتیب‌های ریزش مناسب ایجاد کنیم. برای این منظور راس یا رئوسی را انتخاب می‌کنیم که بیشترین مزاحمت را برای نگهدارنده بودن سایر رئوس موجب می‌شوند. با افزایش رنگ این رئوس سعی در ایجاد رئوس نگهدارنده و به تبع آن ترتیب‌های ریزش داریم تا از رکود و ایستایی الگوریتم جلوگیری کنیم.

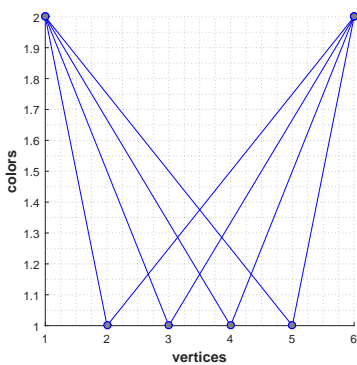
بطور کلی، در هر مرحله از روی جواب فعلی برای کاهش تابع هدف، به دنبال یافتن ترتیب‌های ریزش مناسب جهت تسریع در فرایند جستجو محلی هستیم. اگر این ترتیب‌ها موجود بودند آنها را روی جواب فعلی اعمال کرده و جواب جدید را بدست می‌آوریم، در غیر اینصورت با اعمال مکانیزم آشفستگی فوق سعی در ایجاد ترتیب ریزش مناسب در گراف می‌کنیم و همین روال را تا رسیدن به شرایط توقف ادامه می‌دهیم. شکل (۸.۵) چگونگی عملکرد این مکانیسم را توضیح می‌دهد. در این شکل گراف G به همراه رنگ آمیزی مجاز X در یک نمودار دوبعدی (محور افقی اندیس و محور عمودی رنگ رئوس است) ترسیم شده است تا رابطه نگهدارنده بودن رئوس مشخص‌تر شود. در این مثال فرض کنید $k_{max} = 1$ و فرآیند جستجو در جواب $X = (1, 2, 2, 2, 2, 1)$ متوقف شده است (شکل ۱۰.۵). در این حالت نه راس نگهدارنده‌ای و به تبع آن نه مجموعه ۱- کاهنده‌ای نسبت به X وجود دارد. لذا بنابر گزاره (۱۰.۳.۵) نتیجه می‌شود که جواب X نسبت به همسایگی N_1 بهینه محلی است. برای خروج از این وضعیت (خروج از بهینه محلی) باید رئوسی که مزاحم نگهدارنده بودن سایر رئوس هستند را شناسایی کرده و رنگ آنها را افزایش دهیم (شکل ۸.۵ ب). در اینجا رئوس ۱ و ۶ رئوس



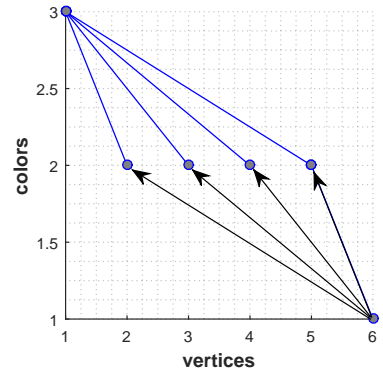
(ب) افزایش رنگ راس ۳ - ایجاد آشفتگی



(آ) جواب بهینه محلی نسبت به N_1



(د) جواب بهینه سراسری



(ج) تبدیل راس ۶ به یک راس نگهدارنده

شکل ۸.۵: توضیح مکانیسم ایجاد آشفتگی

مزاحم هستند. لذا با افزایش رنگ یکی از آنها (مثلا ۱) رئوس نگهدارنده ایجاد می شود (شکل ۸.۵ ج). اکنون اگر روش جستجوی محلی با همسایگی N_1 را روی جواب $Y = (3, 2, 2, 2, 2, 1)$ بکار ببریم، به جواب بهینه $X = (2, 1, 1, 1, 1, 2)$ دست خواهیم یافت که جواب بهینه سراسری است (شکل ۸.۵ د).

۴.۵ روش جستجوی همسایگی متغیر برای مسئله MSC نسبت به همسایگی های N_k

یکی از پر کاربردترین الگوریتم های فرا ابتکاری در حل مسایل ترکیبیاتی، روش جستجوی همسایگی متغیر (VNS)^{۱۵} است که در سال ۱۹۹۷ توسط هنسن^{۱۶} و ملادنویچ^{۱۷} ارائه شد [۱۸]. ایده اصلی این

¹⁵Variable Neighborhood Search

¹⁶Pierre Hansen

¹⁷Nenad Meladenovic

روش مبتنی بر تغییر همسایگی بر اساس اسلوب معین در فرایند جستجو است. ایده تغییر همسایگی در این روش، برای خروج از بهینه محلی است که بر سه اصل زیر استوار است:

(الف) یک جواب کمینه محلی نسبت به ساختار همسایگی خاص لزوماً نسبت به همسایگی دیگر کمینه محلی نیست.

(ب) یک جواب کمینه سراسری، یک جواب کمینه محلی نسبت به هر ساختار همسایگی است.

(ج) در بسیاری از مسایل بهینه‌سازی، جواب‌های کمینه محلی نسبت به یک یا چند ساختار همسایگی نسبتاً به هم نزدیکند.

در روش VNS دائماً همسایگی بزرگتری جستجو می‌شود. اگر در این جستجو جواب بهتری یافت شود جستجو حول جواب جدید از ابتدا آغاز می‌شود.

روش VNS شامل انواع مختلفی است که در هنگام پیاده‌سازی این روش بکار گرفته می‌شوند. از جمله می‌توان به روش کاهش همسایگی متغیر (VND)^{۱۸}، روش جستجوی همسایگی متغیر کاهش یافته (RVNS)^{۱۹}، روش جستجوی همسایگی متغیر مورب (SVNS)^{۲۰}، روش جستجوی همسایگی متغیر اصلی (BVNS)^{۲۱} و ... اشاره کرد. (خواننده علاقمند می‌تواند برای مطالعه بیشتر در این زمینه به [۳۸، ۱۸] مراجعه کند.) اتخاذ هر کدام از روش‌های بیان شده برای یک مسئله خاص می‌تواند در کیفیت جواب‌های مسئله یاد شده موثر باشد.

۱.۴.۵ جستجوی همسایگی متغیر مورب

همیشه اینطور نیست که جواب‌های بهینه محلی در یک جا جمع شده و بهم نزدیک باشند. گاهی اوقات در برخی مسایل جواب‌های نزدیک بهینه از یکدیگر یا از جواب فعلی فاصله دارند. این اتفاق در مورد مسائل رنگ آمیزی گراف‌ها که دارای فضای شدنی بزرگی هستند، محتمل تر است. برای پیمایش این چنین جواب‌هایی بزرگتر کردن همسایگی کار مناسبی نیست زیرا ممکن است اطلاعات مربوط به جواب بهینه محلی فعلی از دست برود. برای رفع این مشکل روش VNS اصلاح می‌شود که از آن به عنوان روش جستجوی همسایگی متغیر مورب (SVNS) یاد می‌کنند. در این رساله از این نوع روش جستجوی همسایگی متغیر روی مسئله MSC استفاده شده است. ساختار کلی این روش برای مسئله رنگ آمیزی مجموع در الگوریتم (۱) آورده شده است.

در این روش $\sum(X'')$ (مقدار تابع هدف جواب فعلی) با عبارت $\sum(X'') - \alpha\rho(X, X'')$ جایگزین می‌شود که در آن $\rho(X, X'')$ فاصله بین X و X'' و α یک پارامتر است. تابع ρ یک متر بین جواب‌هاست. به عنوان مثال در فضای پیوسته می‌تواند تابع فاصله اقلیدسی یا در فضای گسسته

¹⁸Variable Neighborhood Descent

¹⁹Reduced VNS

²⁰Skewed VNS

²¹Basic VNS

الگوریتم ۱ ساختار کلی روش جستجوی همسایگی مورب (SVNS)

- مقدار دهی اولیه ماتریس مجاورت گراف G ، همسایگی های N_k برای $k = 1, \dots, k_{max}$
 جواب ابتدایی X به همراه مقدار آن یعنی $\sum(X)$
 قرار دهید $X_{opt} \leftarrow X$
 شرط (یا شرایط) پایانی به همراه مقداری برای پارامتر α در نظر بگیرید.
 ۱: تازمانیکه شرط توقف برقرار نشده انجام دهید :
 ۲: قرار دهید $k \leftarrow 1$:
 ۳: تازمانیکه $k \leq k_{max}$ انجام دهید :
 ۴: آ. (لرزش یا آشفتگی) :
 جواب تصادفی X' را از k امین همسایگی X انتخاب کنید. ($X' \in N_k(X)$).
 ۵: ب. (جستجوی محلی) :
 یک روش جستجوی محلی را روی جواب X' اعمال کرده و جواب بهینه محلی X'' را بدست بیاورید.
 ۶: ج. (بهبود یافتن جواب) :
 ۷: اگر X'' از X_{opt} بهتر بود، یعنی $\sum(X'') < \sum(X_{opt})$ آنگاه :
 ۸: $X_{opt} \leftarrow X''$
 ۹: $\sum(X_{opt}) \leftarrow \sum(X'')$
 ۱۰: پایان شرط
 ۱۱: د. (جابجایی جواب) :
 ۱۲: اگر $\sum(X'') - \alpha\rho(X, X'') < \sum(X)$ آنگاه :
 ۱۳: $X \leftarrow X''$
 ۱۴: $k \leftarrow 1$
 ۱۵: وگرنه
 ۱۶: $k \leftarrow k + 1$
 ۱۷: پایان شرط
 ۱۸: پایان حلقه
 ۱۹: پایان حلقه
 ۲۰: جواب X_{opt} را چاپ کنید.

برای بردارهای بولی فاصله همینگ در نظر گرفته شود. ما در اینجا برای مسئله مذکور از نرم $p=2$ و از متری که توسط این نرم القا می شود برای $p=1, 2$ استفاده می کنیم. می دانیم p -نرم دسته ای از نرم های برداری است که با نماد $\|\cdot\|_p$ نمایش داده و بصورت زیر تعریف می شود:

$$\|X\|_p = (|x_1|^p + |x_2|^p + \dots + |x_n|^p)^{\frac{1}{p}} \quad p \geq 1, X \in \mathbb{R}^n$$

همچنین فاصله هر دو بردار n تایی X و Y بر اساس این نرم بصورت زیر تعریف می شود:

$$\|(X, Y)\|_p = (|x_1 - y_1|^p + |x_2 - y_2|^p + \dots + |x_n - y_n|^p)^{\frac{1}{p}}$$

الگوریتم (۱) که ساختار کلی روش جستجوی همسایگی متغیر مورب را برای مسئله رنگ آمیزی مجموع توضیح می‌دهد، شامل چهار مرحله اصلی است. این چهار مرحله عبارتند از: لرزش یا آشفتگی، جستجوی محلی، بهبود جواب و جابجایی به جواب جدید. معمولاً مرحله دوم (جستجوی محلی) زمان بیشتری را نسبت به سایر مراحل به خود اختصاص می‌دهد. این روش با یک رنگ آمیزی تصادفی اما مجاز شروع می‌شود. همچنین مقدار k_{max} بیشترین مقدار افزایش اندازه همسایگی‌ها، به همراه سایر شرایط توقف باید در ابتدا مشخص شود.

در مرحله نخست (خط ۴) یکی از دو استراتژی لرزش یا آشفتگی با احتمال p انتخاب می‌شود. در مرحله جستجوی محلی (خط ۵) از روش کاهش همسایگی متغیر (*VND*) یا همسایگی متغیر کاهش یافته (*RVNS*) باز هم بصورت تصادفی استفاده می‌کنیم. مراحل کلی این روش‌ها نیز به ترتیب در الگوریتم‌های (۲) و (۳) نشان داده شده‌اند. سپس روش *SVNS* بررسی می‌کند که آیا بهبودی در جواب اتفاق افتاده است یا خیر (خط ۶-۹)؟ در خطوط ۱۱ الی ۱۷، الگوریتم تصمیم می‌گیرد که آیا جابجایی به جواب جدید انجام بشود یا اینکه مقدار k افزایش یابد؟ *SVNS* گام‌های مذکور را تا رسیدن به حداکثر تعداد تکرار از آخرین بهبود واقع شده انجام می‌دهد. همانطور که قبلاً بیان

الگوریتم ۲ روش کاهش همسایگی متغیر (*VND*)

- مقدار دهی اولیه همسایگی‌های N_k برای $k = 1, \dots, l_{max}$
 جواب ابتدایی X به همراه مقدار آن یعنی $\sum(X)$
 خروجی: جواب X با مقدار $\sum(X)$
- ۱: تازمانیکه بهبود در جواب موجود است انجام دهید:
 - ۲: قرار دهید $l \leftarrow 1$
 - ۳: تازمانیکه $l \leq l_{max}$ انجام دهید:
 - ۴: (جستجو در همسایگی فعلی):
 - ۵: بهترین جواب X' در همسایگی X را پیدا کنید ($X' \in N_l(X)$).
(بهبود و جابجایی):
 - ۶: اگر $\sum(X') < \sum(X)$ آنگاه:
 - ۷: $X \leftarrow X'$
 - ۸: $l \leftarrow 1$
 - ۹: وگرنه
 - ۱۰: $l \leftarrow l + 1$
 - ۱۱: پایان شرط
 - ۱۲: پایان حلقه
 - ۱۳: پایان حلقه

شد، بیشترین محاسبات در این روش مربوط به مرحله جستجوی محلی است. برای بهبود در کیفیت جواب‌ها و همچنین افزایش سرعت فاز جستجوی محلی، از مفهوم نگهدارنگی و کاهندگی جهت یافتن ترتیب‌های ریزش مفید (بخش ۳.۳.۵) کمک گرفته شده است. همچنین برای خروج از بهینه محلی و بهبود در کیفیت جواب‌ها از مکانیسم توضیح داده شده در بخش (۴.۳.۵) استفاده شده است. برای درک بهتر عملکرد روش *SVNS*، خروجی الگوریتم مذکور را در چند گام متوالی منتهی به

الگوریتم ۳ روش همسایگی متغیر کاهش یافته (RVND)

مقدار دهی اولیه همسایگی های N_k برای $k = 1, \dots, l_{max}$

جواب ابتدایی X به همراه مقدار آن یعنی $\sum(X)$

شرط (یا شرایط) توقف را تعیین کنید.

خروجی: جواب X با مقدار $\sum(X)$

۱: تازمانیکه شرایط توقف محقق نشده است انجام دهید:

۲: قرار دهید $l \leftarrow 1$

۳: تازمانیکه $l \leq l_{max}$ انجام دهید:

۴: (لرزش):

جواب X' را بصورت تصادفی از $N_l(X)$ انتخاب کنید.

۵: (بهبود و جابجایی):

۶: اگر $\sum(X') < \sum(X)$ آنگاه:

۷: $X \leftarrow X'$

۸: $l \leftarrow 1$

۹: وگرنه

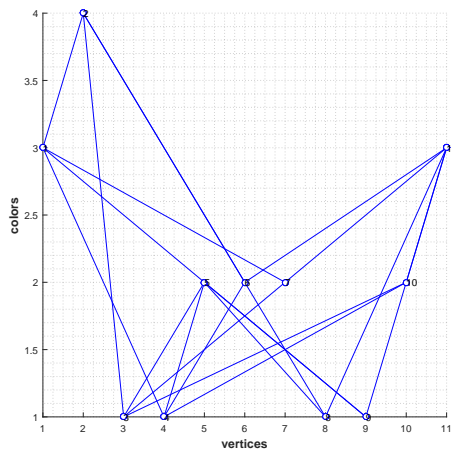
۱۰: $l \leftarrow l + 1$

۱۱: پایان شرط

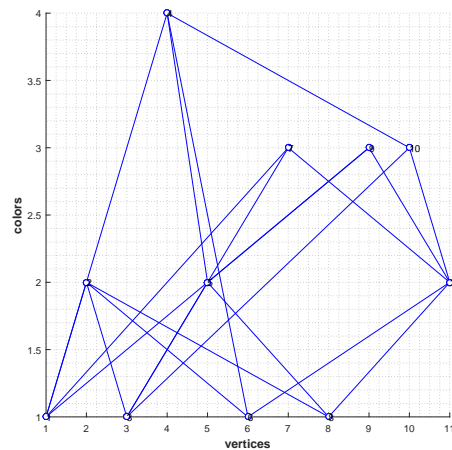
۱۲: پایان حلقه

۱۳: پایان حلقه

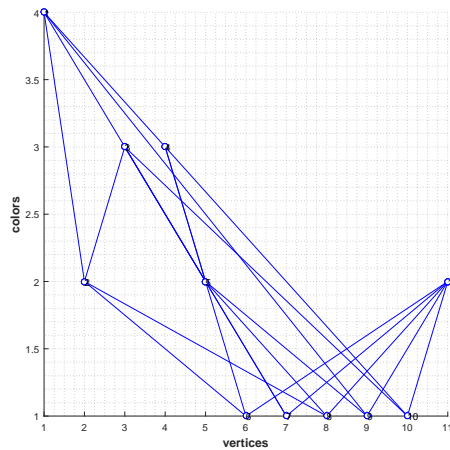
جواب بهینه برای سه نمونه مسئله $myciel^3$ ، $myciel^4$ و $queen_{5 \times 5}$ با مرتبه های ۱۱، ۲۳ و ۲۵ به ترتیب در شکل های (۹.۵)، (۱۰.۵) و (۱۱.۵) نمایش داده شده است.



(ب) رنگ آمیزی غیر بهینه



(آ) رنگ آمیزی غیر بهینه



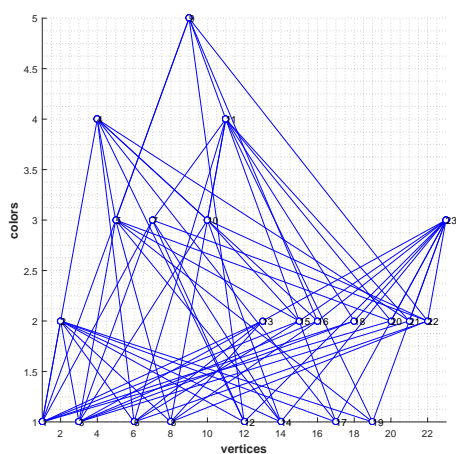
(ج) نمایش گراف در رنگ آمیزی بهینه

شکل ۹.۵: خروجی عملکرد $SVNS$ روی مسئله $myciel_3$ در سه گام متوالی منتهی به جواب بهینه

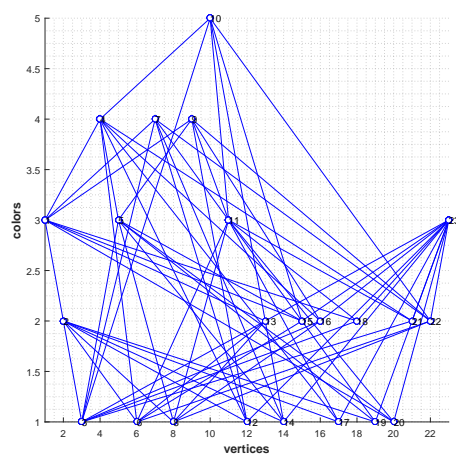
۵.۵ تعمیم روش $SVNS$ برای مسئله GCP

در این بخش قصد داریم روش بیان شده در این فصل را برای مسئله رنگ آمیزی معمولی (GCP) بکار ببریم. همانطور که می‌دانیم جواب بهینه مسئله رنگ آمیزی معمولی با مسئله رنگ آمیزی معمولی می‌تواند بسیار متفاوت باشد. بنابراین از این جهت ممکن است روش‌هایی که برای حل یکی از این دو مسئله ارائه می‌شود، برای دیگری کارایی نداشته باشد. اما ما در این بخش قصد داریم از روش ارائه شده برای حل مسئله رنگ آمیزی مجموع استفاده کرده و آنرا برای مسئله رنگ آمیزی معمولی بکار ببریم. برای این منظور جهت یاد آوری اجازه دهید به مسئله رنگ آمیزی معمولی اشاره‌ای داشته باشیم.

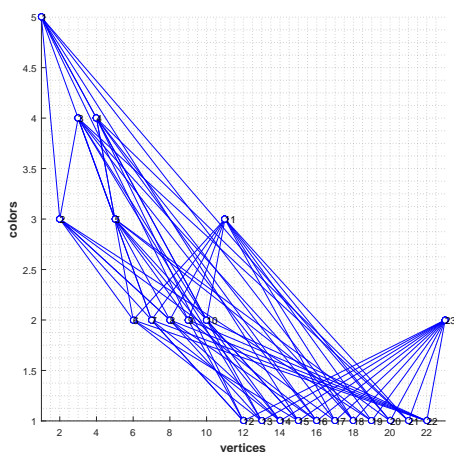
گراف ساده G را در نظر بگیرید. هدف در مسئله رنگ آمیزی معمولی گراف، یافتن حداقل تعداد رنگ برای تخصیص به رئوس است بصورتی که این تخصیص‌ها موجه (مجاز) باشند. مدل ریاضی



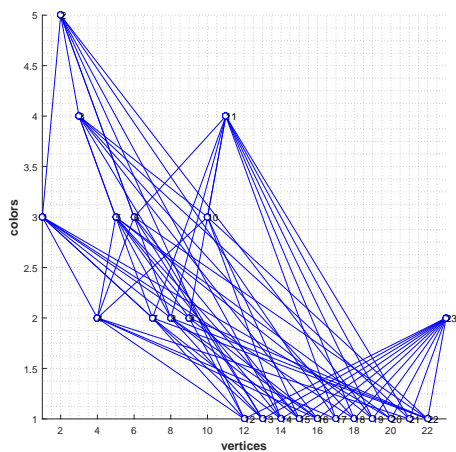
(ب) رنگ آمیزی غیر بهینه



(ا) رنگ آمیزی غیر بهینه



(د) نمایش گراف در رنگ آمیزی بهینه



(ج) رنگ آمیزی غیر بهینه

شکل ۱۰.۵: خروجی عملکرد *SVNS* روی مسئله *myciel*^۴ در چهار گام متوالی منتهی به جواب بهینه

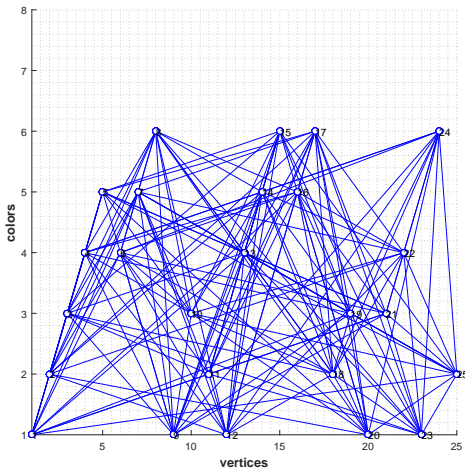
این مسئله بر اساس نمادگذاری‌های بخش (۱.۵) بصورت زیر خواهد بود:

$$\text{Min} \quad \max_{1 \leq i \leq n} x_i$$

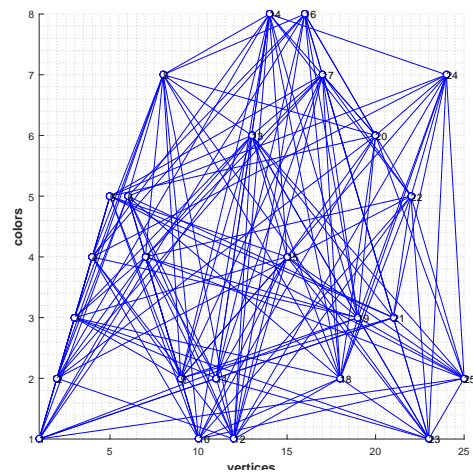
s.t.

$$|x_i - x_j| \geq 1 \quad : \forall (i, j) \in E$$

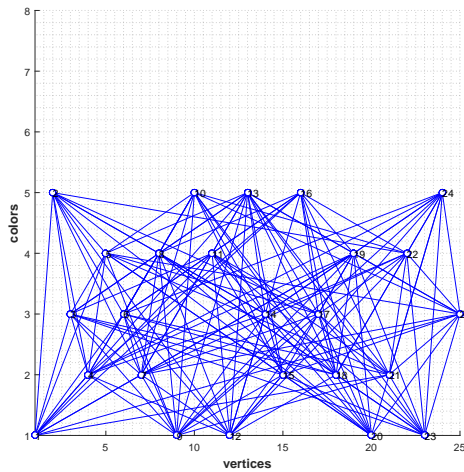
$$x_i \geq 0, \text{ integer} \quad : i = 1, 2, \dots, n$$



(ب) رنگ آمیزی غیر بهینه



(ا) رنگ آمیزی غیر بهینه



(ج) نمایش گراف در رنگ آمیزی بهینه

شکل ۱۱.۵: خروجی عملکرد SVNS روی مسئله 5×5 queen در سه گام متوالی منتهی به جواب بهینه

همانطور که ملاحظه می شود فضای شدنی این مدل همان فضای شدنی مدل بخش (۱۰.۵) است و تفاوت فقط در تابع هدف است. با فرض $y = \max_{1 \leq i \leq n} x_i$ مدل فوق بصورت زیر در می آید:

$$\text{Min } y$$

s.t.

$$|x_i - x_j| \geq 1 \quad : \forall (i, j) \in E$$

$$y - x_i \geq 0 \quad : i = 1, 2, \dots, n$$

$$x_i \geq 0, \text{ integer} \quad : i = 1, 2, \dots, n$$

همچنین با تغییر متغیر $y = Y - 1$ داریم :

$$\begin{aligned}
 P_g) \quad & \text{Min} \quad Y - 1 \\
 & \text{s.t.} \\
 & |x_i - x_j| \geq 1 \quad : \forall (i, j) \in E \\
 & Y - x_i \geq 1 \quad : i = 1, 2, \dots, n \\
 & x_i \geq 0, \text{integer} \quad : i = 1, 2, \dots, n
 \end{aligned}$$

بنابر توضیحات فوق مقدار جواب بهینه مسئله P_g ، عدد رنگی گراف G خواهد بود. اکنون می‌خواهیم روش ارائه شده در بخش (۴.۵) برای یافتن رنگ آمیزی مجموع را برای یافتن عدد رنگی تعمیم دهیم. برای این منظور گراف ساده G با مجموعه رئوس (v_1, v_2, \dots, v_n) را در نظر بگیرید. گراف G' را از روی G بصورت زیر بسازید. راس v_{n+1} را به G اضافه کرده و آن را به تمام رئوس G وصل کنید. عبارت دیگر قرار دهید $G' = G + v_{n+1}$. لذا مدل ریاضی مسئله رنگ آمیزی مجموع برای گراف G' بصورت زیر است:

$$\begin{aligned}
 \text{Min} \quad & \sum_{i=1}^{n+1} x_i \\
 & \text{s.t.} \\
 & |x_i - x_j| \geq 1 \quad : \forall (i, j) \in E(G') \\
 & x_i \geq 0, \text{integer} \quad : i = 1, 2, \dots, n+1
 \end{aligned}$$

اکنون با تغییری کوچک در تابع هدف مدل فوق، مسئله P_M را بصورت زیر در نظر بگیرید:

$$\begin{aligned}
 P_M) \quad & \text{Min} \quad \sum_{i=1}^n x_i + Mx_{n+1} \\
 & \text{s.t.} \\
 & |x_i - x_j| \geq 1 \quad : \forall (i, j) \in E(G') \\
 & x_i \geq 0, \text{integer} \quad : i = 1, 2, \dots, n+1
 \end{aligned}$$

که در آن M عددی مثبت و به قدر کافی بزرگ است و x_{n+1} رنگ راس v_{n+1} است. واضح است که جواب بهینه مسئله P_M یک کران پایین برای مسئله P_g است (زیرا فضای شدنی مسئله P_M شامل فضای شدنی مسئله P_g است). اکنون می‌خواهیم جوابی نزدیک بهینه برای مسئله P_g بدست بیاوریم. برای این منظور روش SVNS را روی مسئله P_M (برای M به قدر کافی بزرگ) اجرا می‌کنیم. با اندک تغییری در روش مذکور (SVNS تغییر یافته) می‌توان به جوابی دست یافت که برای مسئله P_g هم کاربرد داشته باشد. در واقع در هرگام از اجرای این روش SVNS روی گراف G' ، باید طوری

عمل کنیم که رنگ راس v_{n+1} از راس تمام رئوس دیگر بیشتر باشد. برای این منظور می‌توان از مکانیزم آشفته‌گی روی راس v_{n+1} استفاده کرد. با این عمل جواب خروجی برای مسئله P_g هم موجه خواهد بود.

بدلیل وجود ضریب M در تابع هدف مسئله P_M ، الگوریتم $SVNS$ سراغ مینیمم کردن متغیر x_{n+1} می‌رود. از طرف دیگر کاهش در متغیر x_{n+1} نمی‌تواند به میزانی باشد که از رنگ سایر رئوس کمتر شود (زیرا خودمان اینطور فرض کردیم). لذا الگوریتم روی مسئله P_M دقیقاً همان کاری را می‌کند که ما انتظار داریم از حل مسئله P_g که همان کمینه کردن تعداد رنگ رئوس بکار رفته است، حاصل شود.

بنابر مطالب اخیر نتیجه می‌شود اگر روش $SVNS$ تغییر یافته روی مسئله P_M اجرا شود، خروجی آن جوابی است که برای مسئله P_g هم شدنی است. لذا یک کران بالا برای مسئله P_g که همان محاسبه عدد رنگی گراف (مسئله رنگ آمیزی معمولی) است، بدست می‌آید. البته صحبت در مورد کارایی این روش زمانی امکان پذیر است که بتوان آنرا با سایر روش‌های موجود مقایسه کرد که لازمه آن پیاده سازی در قالب یک برنامه کامپیوتری است. این موضوع می‌تواند در ادامه کار این رساله پیشنهاد گردد.

فصل ۶

پیاده سازی و نتایج محاسباتی

عموما بعد از بیان روش برای حل یک مسئله، مبحث پیاده سازی و بررسی کارایی آن روش، بسیار مهم است. زیرا هر چند ممکن است روشی در ابتدا برای حل یک مسئله بسیار مناسب به نظر برسد، ولی تا زمانی که آن روش توسط رایانه پیاده سازی و توسط یک برنامه رایانه ای روی چند نمونه مسئله اجرا نشده است، نمی توان عملا از کارایی آن سخنی به میان آورد. اینکه در فرایند پیاده سازی روش حل یک مسئله از چه نرم افزار یا زبان برنامه نویسی یا ساختمان داده ای یا ... استفاده شود از جمله موضوعات و چالش هایی است که ما هنگام پیاده سازی یک روش با آن روبرو هستیم. در حقیقت عدم انتخاب درست ساختمان داده یا نحوه کدنویسی و زبان برنامه نویسی مناسب می تواند منجر به تولید حجم محاسبات زمانبر شود و یک الگوریتم سریع را به یک الگوریتم بسیار کند بدل کند به طوری که فضای و زمان بیش از اندازه ای از حافظه و پردازشگر را درگیر خود می کند^۱.

۱.۶ نتایج تجربی

ما در این فصل قصد داریم به بررسی کارایی روش ارائه شده در فصل (۵) پردازیم و عملکرد این روش را در یافتن رنگ آمیزی مجموع روی گراف های ساده مورد مطالعه قرار بدهیم. برای این منظور از نمونه مسائل موجود در کتابخانه ای بنام COLOR^۲ استفاده می کنیم. ۲۳ مورد از نمونه گراف های معروف این کتابخانه که در مقالات متعدد بیشترین تکرار را داشته و روش های دیگر روی آنها با یکدیگر مقایسه شده اند، در جدول (۱.۶) آمده است.

روش SVNS با استفاده از نرم افزار متلب^۳ پیاده سازی شده و روی رایانه ای با مشخصات معلوم^۴ اجرا شده است. از آنجایی که SVNS دارای گام های تصادفی است، لذا نتایج بدست آمده حاصل ۱۰ اجرای مستقل است تا بتوان دید بهتری از عملکرد روش ارائه شده داشت. نتایج بدست آمده به همراه مشخصات گراف ها در جدول (۱.۶) ارائه شده است. این مشخصات شامل تعداد رئوس ($|V|$)، تعداد یال ها ($|E|$)، چگالی ($d = \frac{2mn}{n-1}$)، مجموع رنگی (Σ) همچنین قدرت راسی گراف (k) به ترتیب در ستون های ۱ الی ۶ جدول مذکور بیان شده است.

در ستون های ۷ الی ۹ این جدول جزئیات مربوط به نتایج بعد از اجرای الگوریتم SVNS است. بهترین میزان تابع هدف (Σ_*)، تعداد رنگ های بکار رفته (k_*)، میانگین اجراهای مختلف ($Avg.$) به همراه انحراف معیار جواب ها ($Std.$) جزئیاتی است که در این ستون ها ارائه شده است. بنابر نتایج بیان شده در جدول (۱.۶)، الگوریتم SVNS در ۱۹ مورد از ۲۳ مورد موفق به یافتن مجموع رنگی گراف شده است (جواب دقیق) در حالیکه در یافتن مجموع رنگی برای سایر ۴ مورد دیگر بصورت تقریبی عمل کرده است همچنین در جدول (۱.۶) نتایج مقایسه ای با پنج روش موجود در مقالات تست شده روی مسائل اشاره شده در یک دهه گذشته، نشان داده شده است. این روش ها که بصورت مخفف نمایش

^۱البته این موضوع به معنای بهینه بودن نحوه کد نویسی کدهای موجود در پیوست نیست.

^۲<http://mat.gsia.cmu.edu/COLOR02/>

^۳MATLAB (R2015a)

^۴2.7 GHz CPU and 4 Gb RAM

جدول ۱۰۶: نتایج محاسباتی حاصل از اجرای الگوریتم SVNS روی ۲۳ نمونه گراف از کتابخانه COLOR

Instances	مشخصات گراف					SVNS		
	$ V $	$ E $	d	Σ	k	$\Sigma_*(k_*)$	Avg.	Std.
myciel3	11	20	0.36	21	4	21 (4)	21.0	0.0
myciel4	23	71	0.28	45	5	45 (5)	45.0	0.0
myciel5	47	236	0.22	93	6	93 (6)	93.0	0.0
myciel6	95	755	0.17	189	7	189 (7)	189.0	0.0
myciel7	191	2360	0.13	381	8	381 (8)	381.4	0.52
anna	138	986	0.05	276	11	276 (11)	276.3	0.48
david	87	812	0.11	237	11	237 (11)	238.6	0.84
huck	74	602	0.11	243	11	243 (11)	243.0	0.0
jean	80	508	0.08	217	10	217 (10)	217.1	0.31
homer	561	1629	0.01	-	10	1163 (13)	1168.5	3.5
queen5.5	25	160	0.53	75	5	75 (5)	75.0	0.0
queen6.6	36	290	0.46	138	7	138 (8)	138.2	0.42
queen7.7	49	476	0.4	196	7	196 (7)	197.4	4.4
queen8.8	64	728	0.36	291	9	294 (9)	301	5.33
games120	120	638	0.09	443	9	443 (9)	445.3	1.56
miles250	128	387	0.05	325	8	329 (8)	333.2	2.1
miles500	128	1170	0.11	≤ 709	20	719(20)	728.2	7.03
mug88-1	88	146	0.04	178	4	178 (4)	178.0	0.0
mug88-25	88	146	0.04	178	4	178 (4)	178.0	0.0
mug100-1	100	166	0.03	202	4	202 (4)	202.0	0.0
mug100-25	100	166	0.03	202	4	202 (4)	202.0	0.0
2-insertion-3	37	72	0.11	62	4	62 (4)	62.0	0.0
3-insertion-3	56	110	0.07	92	4	92 (4)	92.0	0.0

داده شده‌اند، عبارتند از: EXSCOL (یک روش ابتکاری) [۵۱]، MRLF (الگوریتم حریمانه مبتنی بر روش شناخته شده RLF برای رنگ آمیزی معمولی گراف) [۳۱]، MDS5 [۲۱]، PGA (الگوریتم ژنتیک موازی) [۲۷] و HLS (الگوریتم جستجوی محلی ترکیبی) [۱۱]. مبنای مقایسه بر اساس کمتر بودن مقدار تابع هدف یعنی همان مجموع رنگ رئوس، است. علامت '-' در برخی از درایه‌های جدول (۲۰۶) نشان دهنده‌ای عدم دسترسی به اطلاعات مورد نظر است.

نتایج مربوط به روش SVNS در ستون انتهایی جدول (۲۰۶) آمده‌است. قسمت‌هایی که با قلم ضخیم‌تر نمایش داده شده‌اند، مواردی هستند که بازای آنها روش SVNS موفق به یافتن جواب دقیق می‌شود. همانطور که ملاحظه می‌شود فقط در سه مورد (queen8.8-miles250-miles500) جواب دقیق بدست نیامده‌است. البته در این موارد هم روش SVNS از برخی از سایر روش‌ها بهتر عمل می‌کند.

Names	Σ	EXCOL [51]	MRLF [31]	MDS5 [21]	PGA [27]	HLS [11]	SVNS
myciel3	21	21	21	21	21	21	21
myciel4	45	45	45	45	45	45	45
myciel5	93	93	93	93	93	93	93
myciel6	189	189	189	189	189	189	189
myciel7	381	381	381	381	382	381	381
anna	276	283	277	276	281	-	276
david	237	237	241	237	243	-	237
huck	243	243	244	243	243	243	243
jean	217	217	217	217	217	-	217
queen5.5	75	75	75	75	75	-	75
queen6.6	138	138	138	138	138	138	138
queen7.7	196	196	196	196	196	-	196
queen8.8	291	291	303	291	302	-	294
games120	443	443	446	443	460	446	443
miles250	325	328	334	325	347	343	329
miles500	≤ 709	709	715	712	762	755	719
mug88-1	178	-	-	178	-	-	178
mug88-25	178	-	-	178	-	-	178
mug100-1	202	-	-	202	-	-	202
mug100-25	202	-	-	202	-	-	202
2-insertion-3	62	-	-	62	-	-	62
3-insertion-3	92	-	-	92	-	-	92

جدول ۲.۶: مقایسه نتایج بین روش SVNS و پنج روش دیگر روی مجموعه‌ای از نمونه مسائل COLOR02

۲.۶ نتیجه‌گیری و پیشنهادات

علیرغم وجود مدل‌های متنوع رنگ‌آمیزی گراف در مقالات متعدد در زمینه علوم مختلف، همچنان رنگ‌آمیزی کلاسیک یکی از مدل‌های پرکاربرد است. از جمله مسائل رنگ‌آمیزی که بسیار به مدل کلاسیک نزدیک است، مسئله رنگ‌آمیزی مجموع است. تلاش این رساله مطالعه روی وجوه مختلف این مسئله NP-سخت است. بطور کلی از زمان معرفی این مسئله در دهه ۱۹۸۰ میلادی، دو دیدگاه در برخورد با این مسئله وجود داشته است. رویکرد تحلیلی و رویکرد الگوریتمی. موضوع تعمیم این مسئله نیز از جمله موضوعاتی است که از دیدگاه تحلیلی به آن پرداخته شده است. با اینکه رویکرد اصلی این رساله در بررسی این مسئله بیشتر متوجه دیدگاه دوم یعنی الگوریتمی است، اما سعی شده در پرداختن به مسئله مذکور از دیدگاه تحلیلی غفلتی صورت نگیرد.

در راستای بررسی مسئله با رویکرد دوم، یکی از روش‌های فراابتکاری برای این مسئله بعنوان یک مسئله NP-سخت، روش جستجوی همسایگی متغیر است. این روش بر اساس تغییر در همسایگی‌هایی انجام می‌شود که باید قبلاً بر مبنای مدل فضای شدنی تعریف شده باشند. با تغییری که در مدل ریاضی مسئله رنگ‌آمیزی مجموع قرار دادیم، قطعاً نوع همسایگی و پیمایش فضای جستجو به منظور یافتن جواب بهینه نیز دستخوش تغییر قرار می‌گیرد. لذا روش جستجوی همسایگی متغیری که در این رساله ارائه شده بر اساس این همسایگی‌های جدید است.

اما از چالش‌های مهم و اساسی هنگام پیاده‌سازی روش‌های ابتکاری و فرا ابتکاری برای مسائل NP-سخت، یکی افزایش حجم محاسبات هنگام افزایش حجم همسایگی‌ها و دیگری افتادن در تله بهینه محلی و مکانیزم خروج از آن است. در مواجهه با چالش اول از مفهوم جدیدی بنام نگهدارندگی راسی و مجموعه‌ای استفاده شده است تا در فرایند جستجو از بررسی جواب‌هایی که موجب بهبود جواب بهینه نمی‌شوند، اجتناب شود. در مورد چالش دوم هم از یک مکانیزم ایجاد آشفتنگی در جواب‌ها جهت ممانعت ورود به بهینه محلی یا خروج از آن استفاده شده است.

ایجاد ارتباط بین روش ارائه شده در این رساله برای مسئله مجموع رنگی با مسئله رنگ‌آمیزی معمولی از جمله مباحث مورد مطالعه در این رساله است. در نهایت نیز کارایی روش ارائه شده روی چند نمونه گراف مورد بررسی و مقایسه قرار گرفت.

ارائه یک روش ابتکاری برای حل مسئله رنگ‌آمیزی مجموع چندگانه با اعمال تغییراتی روی روش ارائه شده فعلی، می‌تواند بعنوان یک پیشنهاد در ادامه پژوهش فعلی تلقی گردد. همچنین با توجه به رویکرد تحلیلی برای مسائل رنگ‌آمیزی مجموع تفاسلی و جمعی (که در فصل (۴) معرفی شده‌اند)، بررسی و طراحی الگوریتم برای آن‌ها مشابه روش ارائه شده برای مسئله رنگ‌آمیزی مجموع، می‌تواند از دیگر موضوعاتی باشد که در ادامه این رساله انجام خواهد گرفت.

مراجع

- [1] M. Alishahi and A. Taherkhani, *A note on chromatic sum*, Tech. Report arXiv:0802.1936, Feb 2008.
- [2] K. Appel and W. Haken, *Every planar map is four colorable. part i: Discharging*, Illinois Journal of Mathematics **21** (1977), no. 3, 429–490.
- [3] K. Appel, W. Haken, and J. Koch, *Every planar map is four colorable. part ii: Reducibility*, Illinois Journal of Mathematics **21** (1977), no. 3, 491–567.
- [4] Amotz Bar-Noy, Mihir Bellare, Magnús M Halldórsson, Hadas Shachnai, and Tami Tamir, *On chromatic sums and distributed resource allocation*, Information and Computation **140** (1998), no. 2, 183 – 202.
- [5] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear programming and network flows*, Wiley-Interscience, 2004.
- [6] U. Benlic and J. K. Hao, *A study of breakout local search for the minimum sum coloring problem*, pp. 128–137, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [7] M. Bondy and U. S. R. Murty, *Graph theory with applications*, Elsevier, New York, 1982.
- [8] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to algorithms*, 2nd ed., McGraw-Hill Higher Education, 2001.
- [9] P. Crescenzi and V. Kann, *How to find the best approximation results – a follow up to garey and johnson*, 1998, pp. 1–7.

- [10] S. M. Douiri and S. Elbernoussi, *A new ant colony optimization algorithm for the lower bound of sum coloring problem*, Journal of Mathematical Modelling and Algorithms **11** (2012), no. 2, 181–192.
- [11] S.M. Douiri and S. Elbernoussi, *New algorithm for the sum coloring problem*, International Journal of Contemporary Mathematical Sciences **6** (2011), no. 10, 453–463.
- [12] P. Erdos and A. Hajnal, *On chromatic number of graphs and set systems*, Acta Mathematica Hungarica **17** (1966), no. 1-2, 61–99.
- [13] P. Erdos, E. Kubicka, and A. J. Schwenk, *Graphs that require many colors to achieve their chromatic sum*, **71** (1990), 17–28.
- [14] K. Giaro, R. Janczewski, M. Kubale, and M. Malafiejski, *A 27/26-approximation algorithm for the chromatic sum coloring of bipartite graphs*, pp. 135–145, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
- [15] H. Hajiabolhassan, M.L. Mehrabadi, and R. Tusserkani, *Minimal coloring and strength of graphs*, Discrete Mathematics **215** (2000), no. 1, 265 – 270.
- [16] M. M. Halldorsson, G. Kortsarz, and H. Shachnai, *Minimizing average completion of dedicated tasks and interval graphs*, pp. 114–126, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.
- [17] M. M. Halldorsson and G. Kortsarz, *Multicoloring planar graphs and partial k -trees*, pp. 73–84, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.
- [18] P. Hansen and N. Mladenovic, *Variable neighborhood search: Principles and applications*, European Journal of Operations Research **130** (2001a), 449–467.
- [19] J.K. Hao and Q. Wu, *Improving the extraction and expansion method for large graph coloring*, Discrete Applied Mathematics **160** (2012), no. 16–17, 2397 – 2407.
- [20] P. J. Heawood, *Map color theorem*, The Quarterly Journal of Pure and Applied Mathematics. **24** (1890), 332–338.
- [21] A. Helmar and M. Chiarandini, *A local search heuristic for chromatic sum*, 2011, pp. 1–10.

-
- [22] T.R. Jensen and B. Toft, *Graph coloring problems*, Wiley, New York, 1995.
- [23] T. Jiang and D. B. West, *Coloring of trees with minimum sum of colors*, Journal of Graph Theory **32** (1999), no. 4, 354–358.
- [24] Y. Jin and J. K. Hao, *Hybrid evolutionary search for the minimum sum coloring problem of graphs*, Information Sciences **352** (2016), no. C, 15–34.
- [25] D. Jungnickel, *Graphs, networks and algorithms*, 3rd ed., Springer Publishing Company, Incorporated, 2007.
- [26] A. B. Kempe, *On the geographical problem of the four colours*, American Journal of Mathematics **2** (1879), no. 3, 193–200.
- [27] Z. Kokosiński and K. Kwarciany, *On sum coloring of graphs with parallel genetic algorithms*, pp. 211–219, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [28] M. Kubale, *Contemporary mathematics 352: Graph colorings*, American Mathematical Society, 2004.
- [29] E. Kubicka, *The chromatic sum of a graph*, Ph.D. thesis, Western Michigan University, Michigan, 1989.
- [30] E. Kubicka and A. J. Schwenk, *An introduction to chromatic sums*, Proceedings of the 17th Conference on ACM Annual Computer Science Conference (New York, NY, USA), ACM, 1989, pp. 39–45.
- [31] Y. Li, C. Lucet, A. Moukrim, and K. Sghiouer, *Greedy algorithms for the minimum sum coloring problem*, Logistique et transports (Sousse, Tunisia), Mar 2009, pp. LT–027.
- [32] D.R. Lick and A.T. White, *k-degenerate graphs*, Canadian Journal of Mathematics **22** (1970), 1082–1096.
- [33] L. Lovasz, *Kneser’s conjecture, chromatic number, and homotopy*, Journal of Combinatorial Theory, Series A **25** (1978), no. 3, 319 – 324.
- [34] M. Malafiejski, *The complexity of the chromatic sum problem on cubic planar graphs and regular graphs*, Electronic Notes in Discrete Mathematics **8** (2001), 62.

-
- [35] M. Malafiejski, *Scheduling multiprocessor tasks to minimize mean flow time (in polish)*, Ph.D. thesis, Gdansk University of Technology, 2002.
- [36] M. Malafiejski, L. Kuszner, and K. Piwakowski, *Preemptive multiprocessor task scheduling to minimize the sum of completion times*, ACS-SCM'02, Miedzyzdroje (2002), 189–196.
- [37] A. Mehrotra and M. A. Trick, *A column generation approach for graph coloring*, INFORMS J. Computing **8** (1996), no. 4, 344–354.
- [38] N. Mladenovic and P. Hansen, *Variable neighborhood search*, Computers and Operations Research **24** (1997), 1097–1100.
- [39] A. Moukrim, K. Sghiouer, C. Lucet, and Y. Li, *Lower bounds for the minimal sum coloring problem*, Electronic Notes in Discrete Mathematics **36** (2010), 663 – 670.
- [40] R. Neapolitan and K. Naimipour, *Foundations of algorithms using c++ pseudocode, third edition*, Jones and Bartlett Publishers, Inc., USA, 2008.
- [41] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, Wiley-Interscience, New York, NY, USA, 1988.
- [42] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.
- [43] M. Sarrafzadeh S. Nicoloso and X. Song, *On the sum coloring problem on interval graphs*, Algorithmica **23** (1999), no. 2, 109–126.
- [44] E. R. Scheinerman and D. H. Ullman, *Fractional graph theory: a rational approach to the theory of graphs*, Dover Publications, 2013.
- [45] S. Rahimi Sharbaf and K. Erfani, *On the sum and difference edge-coloring problem of simple graphs*, Algebraic Structures and Their Applications **4** (2017), no. 1, 33–42.
- [46] S. Rahimi Sharbaf, J. Fathali, and K. Erfani, *New variable neighborhood search method for minimum sum coloring on simple graphs*, Iranian Journal of Numerical Mathematics and Optimization (2017), Accepted.

- [47] K. J. Supowit, *Finding a maximum planar subset of a set of nets in a channel*, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems **6** (2006), no. 1, 93–94.
- [48] T. Szkaliczki, *Routing with minimum wire length in the dogleg-free manhattan model is np-complete.*, SIAM J. Comput. **29** (1999), no. 1, 274–287.
- [49] C. Thomassen, P. Erdos, Y. Alavi, P. J. Malde, and A. J. Schwenk, *Tight bounds on the chromatic sum of a connected graph*, Journal of Graph Theory **13** (1989), no. 3, 353–357.
- [50] Q. Wu and J. K. Hao, *Improved lower bounds for sum coloring via clique decomposition*, CoRR **abs/1303.6761** (2013).
- [51] Q. Wu and J.K. Hao, *An effective heuristic algorithm for sum coloring of graphs*, Computers & Operations Research **39** (2012), no. 7, 1593 – 1600.

پیوست آ

برنامه‌های متلب

کد برنامه کامپیوتری روش SVNS با استفاده از نرم افزار متلب

```

1 function X_opt=SVNS_for_MSCP(X,A,k_max,r_max,itr_max,alfa)
2 Adc=cal_adjecency_list(A);
3 X=update_solution([],X,Adc);
4 [V,I]=holder_vertex_value(X,Adc,0);
5 [O,~,~]=value_of_order_falling(V,I,X,Adc);
6 X=update_solution(O{1},X,Adc);
7 X_opt=X;
8 itr=1;
9 tic
10 while itr < itr_max && toc<=1000
11     k=1;
12     while k<=k_max
13         if rand<0.55 %Generate random solution X1 from N_k(X) or
14             Perturbation Mechanism
15             X1=perturbation1(X,Adc);
16         else
17             X1=Shaking(X,Adc,k);
18         end
19         if rand<0.76 %Executing the local search method
20             with intial solution X1
21             [X11,~]=Local_search1(X1,Adc,r_max);
22         else
23             X11=Local_search2(X1,Adc,r_max);
24         end
25         if sum(X11) < sum(X_opt) %Update the best solution (
26             optimal solution)
27             X_opt=X11;
28             toc
29             sum(X_opt)
30         end
31         d=rou(X,X11);
32         if sum(X11)-alfa*d< sum(X) % Move to another solution or
33             increase the size of neighborhood!!!!
34             X=X11;
35             k=1;
36         else
37             k=k+1;
38         end
39     end
40     itr=itr+1;
41 end
42 end

```

```

1 function Adc=cal_adjecency_list(A)
2 n=size(A,2);
3 Adc=cell(1,n);
4 parfor i=1:n
5     Adc{i}=find(A(i,:)==1);
6 end

```

```
7 end
```

```
1 function A=cal_adjecency_mat(Adc)
2 A=zeros(size(Adc,2));
3     for i=1:size(Adc,2)
4         for j=1:size(Adc{i},2)
5             A(i,Adc{i}(j))=1;
6         end
7     end
8 end
```

```
1 function k=what_miss(v)
2 if iscell(v)
3     v=cell2mat(v);
4 end
5 v=sort(v(v>0));
6 v(2:end+1)=v(1:end);
7 v(1)=0;
8 s=size(v,2);
9 i=2;
10 while i<=s && v(i)-v(i-1)<=1
11     i=i+1;
12 end
13 k=v(i-1)+1;
14 end
```

```
1 function X=update_solution(O,X,Adc)
2 n=size(X,2);
3 itr=0;
4 i=1;
5 ver=1:n;
6 ver(0)=0;
7 v1=ver(ver>0);
8 v=horzcat(O,v1(randperm(size(v1,2)))));
9 while itr < n
10     k=what_miss(X(Adc{v(i)}));
11     if X(v(i))<=k
12         itr=itr+1;
13         i=i+1;
14         continue
15     else
16         X(v(i))=k;
17         i=1;
18         itr=0;
19     end
20 end
21 end
```

```
1 function [V,I]=holder_vertex_value(X,Adc,k)
2 n=size(X,2);
3 V=cell(1,n);
4 I=cell(1,n);
5 parfor t=1:n
6     V1=Adc{t}(X(Adc{t}) > X(t));
7     d=size(V1,2);
```

```

8     if d==0
9         continue
10    end
11    for i=1:d
12        b=Adc{V1(i)}(X(Adc{V1(i)})==X(t));
13        for j=0:k
14            if size(b,2)==j+1
15                V{t}=horzcat(V{t},V1(i));
16                b=setdiff(b,t);
17                I{t}=unique(horzcat(I{t},b));
18            end
19        end
20    end
21 end
22 end

```

```

1 function Y=Shaking(X,Adc,k)
2 a=rand;
3 if a >= 0.75
4     [~,S]=number_of_eight_structure(X,Adc);
5 elseif 0.35<=a && a<0.75
6     [~,S]=value_of_reduction(X,Adc);
7 else
8     [~,S]=number_of_edge_disjoint_2path(X,Adc);
9 end
10 S=S(randperm(k));
11 for i=1:k
12     X(S(i))= what_miss(horzcat(X(Adc{S(i)}),X(S(i))));
13 end
14 Y=X;
15 end

```

```

1 function [O,val,H,X]=traveling_vertex_from(S,I,v,X,Adc)
2 [dist,L,~]=graphshortestpath(-S,v,'Method','Acyclic');
3 t=find(dist==min(dist));
4 L=L{t(1)};
5 O=L;
6 L1=L;
7 val=0;
8 while (size(L1,2)-1)~=0
9     i=size(L1,2);
10    S(L1(i-1),L1(i))=0;
11    [dist,L1,~]=graphshortestpath(-S,v,'Method','Acyclic');
12    t=find(dist==min(dist));
13    L1=L1{t(1)};
14    for j=1:size(L1,2)
15        if ismember(L1(j),0)
16            continue
17        else
18            O=horzcat(O,L1(j));
19        end
20    end
21 end
22 if size(O,2)<=1
23     val=-inf;
24     H=[];

```

```

25     return
26 end
27 H=0(1);
28 for i=1:size(0,2)
29     H=horzcat(H,I{0(i)});
30 end
31 A=cal_adjecency_mat(Adc);
32 A(0(1),:)=0;
33 A(:,0(1))=0;
34 Adc1=cal_adjecency_list(A);
35 for i=2:size(0,2)
36     h=what_miss(X(setdiff(Adc1{0(i)},I{0(i-1)})));
37     val= val + X(0(i)) - h;
38     X(0(i))= h;
39 end
40 for i=1:size(H,2)
41     h=what_miss(X(Adc{H(i)}));
42     val = val + X(H(i)) - h;
43     X(H(i))=h;
44 end
45 end

```

```

1 function [num,I]=number_of_eight_structure(X,Adc)
2 n=size(X,2);
3 add=zeros(1,n);
4 omit=zeros(1,n);
5 for i=1:n
6     a=Adc{i}( X(Adc{i}) > X(i));
7     while size(a,2)
8         b=a(X(a)==X(a(1)));
9         if size(b,2)<=1
10            a=setdiff(a,b);
11            continue
12        end
13        add(i)=add(i) + 1;
14        a=setdiff(a,b);
15    end
16 end
17 for i=1:n
18     a=Adc{i}( X(Adc{i}) > X(i));
19     for j=1:size(a,2)
20         vec=Adc{a(j)}( X(Adc{a(j)})==X(i));
21         if size(vec,2)>=2
22             omit(i) = omit(i) + 1;
23         end
24     end
25 end
26 num=omit-add;
27 [num,I]=sort(num,'descend');
28 end

```

```

1 function [num,I]=number_of_edge_disjoint_2path(X,Adc)
2 n=size(X,2);
3 num=zeros(1,n);
4 for i=1:n
5     a=Adc{i}( X(Adc{i}) > X(i));

```

```

6   for j=1:size(a,2)
7       vec=Adc{a(j)}( X(Adc{a(j)})==X(i));
8       if size(vec,2)>= 3
9           continue
10      else
11          num(i)=num(i)+size(vec,2)-1;
12      end
13
14  end
15 end
16 [num,I]=sort(num,'descend');
17 end

```

```

1  function [V,I]=value_of_reduction(X,Adc)
2  m=max(X);
3  k=1;
4  V=zeros(1,size(X,2));
5  for i=1:m
6      C=find(X==i);
7      for j=1:size(C,2)
8          a=Adc{j}( X(Adc{j}) > X(j));
9          while size(a,2)
10             b=a(X(a)==X(a(1)));
11             kk=(X(b(1))-X(j))*size(b,2);
12             if V(k)<= kk
13                 V(k) = kk ;
14             end
15             a=setdiff(a,b);
16         end
17         k=k+1;
18     end
19 end
20 [V,I]=sort(V,'descend');
21 end

```

```

1  function X=perturbation1(X,Adc)
2  a=rand;
3  if a >= 0.85
4      [num,I]=number_of_eight_structure(X,Adc);
5  elseif 0.55<=a && a<0.85
6      [num,I]=value_of_reduction(X,Adc);
7  else
8      [num,I]=number_of_edge_disjoint_2path(X,Adc);
9  end
10
11 I=I(num>0);
12 r=size(I,2);
13 if r==0
14     return
15 else
16     k=randi([1 floor(r/1)],1);
17     I=I(randperm(k));
18     for i=1:k
19         X(I(i))= what_miss(horzcat(X(Adc{I(i)}),X(I(i))));
20     end
21 end

```

22 end

```
1 function [X,key]=Local_search1(X,Adc,r_max)
2 itr=1;
3 while itr<20
4     [Y,key]=generate_random_sol(X,Adc,r_max);
5     if ~key
6         break
7     end
8     if sum(Y)< sum(X)
9         X=Y;
10        itr=1;
11    else
12        itr=itr+1;
13    end
14 end
15 X=Y;
16 end
```

```
1 function X=Local_search2(X,Adc,r_max)
2 value=1;
3 while value
4     [~,~,value,Z]=generate_random_reducer_set(X,Adc,r_max);
5     X=Z;
6 end
7 X=update_solution([],Z,Adc);
8 end
```

```
1 function [S,O,value,Y]=generate_random_reducer_set(X,Adc,s_max)
2 n=size(X,2);
3 O=cell(1,n);
4 value=0;
5 r=0;
6 S=[];
7 Y=X;
8 while r<=s_max
9     [V,I]=holder_vertex_value(X,Adc,r);
10    [list_0,vec,H,XX]=value_of_order_falling(V,I,X,Adc);
11    if size(vec,2)*size(vec,1)>= 1
12        if rand <0.75
13            S=H{1};
14            O=list_0{1};
15            value=vec(1);
16            Y=XX{1};
17        else
18            t=randi([1,size(vec,2)],1);
19            S=H{t};
20            O=list_0{t};
21            value=vec(t);
22            Y=XX{t};
23        end
24        return
25    else
26        r=r+1;
27    end
```

```
28 end
29 O=cell2mat(O);
30 end
```

```
1 function [Z,key]=generate_random_sol(X,Adc,r_max)
2 key=1;
3 [~,O,value,Z]=generate_random_reducer_set(X,Adc,r_max);
4 Z=update_solution(O,Z,Adc);
5 if value==0 && sum(Z)==sum(X)
6     key=0;
7     return
8 end
9 end
```

واژه‌نامه فارسی به انگلیسی

Relaxation	آزاد سازی
Probabilistic	احتمالی
Polynomial time algorithm	الگوریتم زمان چند جمله‌ای
Non deterministic algorithm	الگوریتم غیر قطعی
Partition	افراز
Global optimum	بهینه سراسری
Local optimum	بهینه محلی
Order of falling	ترتیب ریزش
Matching	تطابق
Neighborhood search	جستجوی همسایگی
Feasible solution	جواب شدنی
Direction	جهت گذاری
Automorphism	خودریختی
Vertex transitive	راس ترایا
Holder vertex	راس نگهدارنده
Optimal coloring	رنگ آمیزی بهینه
Multicoloring	رنگ آمیزی چندگانه
Vertex coloring	رنگ آمیزی راسی
Proper coloring	رنگ آمیزی مجاز
Sum coloring	رنگ آمیزی مجموع
Edge coloring	رنگ آمیزی یالی
Scheduling	زمانبندی
Scheduling and resource allocation	زمانبندی و تخصیص منابع
Fractional chromatic number	عدد رنگی کسری

Coloring number	عدد رنگ آمیزی
Chromatic number	عدد رنگی
Strength of graph	قدرت گراف
Reducibility	کاهش پذیری
Reducing	کاهندگی
Algorithmic bound	کران الگوریتمی
Analytical Bound	کران تحلیلی
Interval graph	گراف بازه ای
Conflict graph	گراف برخورد
Partial graph	گراف جزئی
Directed acyclic graph	گراف جهت دار بدون دور
Kneser graph	گراف کنسر
Chromatic sum	مجموع رنگی
Difference chromatic sum	مجموع رنگی تفاضلی
Sum chromatic sum	مجموع رنگی جمعی
Independent set	مجموعه مستقل
Reducer set	مجموعه کاهنده
Holder set	مجموعه نگهدارنده
Mathematical model	مدل ریاضی
Four color problem	مسئله چهار رنگ
Perturbation mechanism	مکانیزم آشفتگی
Homomorphism	همریختی
Variable neighborhood	همسایگی متغیر
Neighborhood	همسایگی
Perfect neighborhood	همسایگی کامل
Edge transitive	یال ترا یا
Isomorphism	یکریختی

واژه‌نامه انگلیسی به فارسی

Algorithmic bound	کران الگوریتمی
Analytical Bound	کران تحلیلی
Automorphism	خودریختی
Chromatic number	عدد رنگی
Chromatic sum	مجموع رنگی
Coloring number	عدد رنگ آمیزی
Conflict graph	گراف برخورد
Difference chromatic sum	مجموع رنگی تفاضلی
Directed acyclic graph	گراف جهت دار بدون دور
Direction	جهت گذاری
Edge coloring	رنگ آمیزی یالی
Edge transitive	یال ترا یا
Feasible solution	جواب شدنی
Four color problem	مسئله چهار رنگ
Fractional chromatic number	عدد رنگی کسری
Global optimum	بهینه سراسری
Holder set	مجموعه نگهدارنده
Holder vertex	راس نگهدارنده
Homomorphism	همریختی
Independent set	مجموعه مستقل
Interval graph	گراف بازه ای
Isomorphism	یکریختی
Kneser graph	گراف کنسر
Local optimum	بهینه محلی

Matching	تطابق
Multicoloring	رنگ آمیزی چندگانه
Neighborhood	همسایگی
Neighborhood search	جستجوی همسایگی
Non deterministic algorithm	الگوریتم غیر قطعی
Non linear mathematical model	مدل ریاضی غیر خطی
Optimal coloring	رنگ آمیزی بهینه
Order of falling	ترتیب ریزش
Partial graph	گراف جزئی
Partition	افراز
Perfect neighborhood	همسایگی کامل
Perturbation mechanism	مکانیزم آشفتگی
Polynomial time algorithm	الگوریتم زمان چندجمله‌ای
Probabilistic	احتمالی
Proper coloring	رنگ آمیزی مجاز
Reducer set	مجموعه کاهشنده
Reducibility	کاهش پذیری
Reducing	کاهندگی
Relaxation	آزاد سازی
Scheduling	زمانبندی
Scheduling and resource alocation	زمانبندی و تخصیص منابع
Sequencing	توالی
Strength of graph	قدرت گراف
Sum chromatic sum	مجموع رنگی جمعی
Sum coloring	رنگ آمیزی مجموع
Variable neighborhood	همسایگی متغیر
Vertex coloring	رنگ آمیزی راسی
Vertex transitive	راس تراپا

نمایه

آ

آزاد سازی، ۵۹

آشفتگی، ۶۹، ۷۲

ا

افراز به خوشه، ۳۴

الگوریتم زمان چندجمله‌ای، ۲۰

الگوریتم غیر قطعی، ۲۰

اندازه، ۵

ب

برنامه نویسی، ۸۴

بهینه سازی پیوسته، ۱

بهینه سازی گسسته، ۱

بهینه سراسری، ۶۳، ۷۲

بهینه محلی، ۶۳، ۷۲

ت

ترتیب ریزش، ۶۷، ۶۹

تطابق ۳-بعدي، ۲۳

ج

جستجوی همسایگی متغیر، ۵۶، ۷۳

جواب شدنی، ۶۵

جهت گذاری، ۵۹، ۶۰

چ

چگالی، ۸۴

خ

خودریختی، ۴۰

خوشه، ۷، ۵۲

ر

راس نگهدارنده، ۶۷

رنگ آمیزی، ۲

رنگ آمیزی بهینه، ۱۴، ۳۰

رنگ آمیزی چندگانه، ۱۶

رنگ آمیزی راسی، ۲

رنگ آمیزی مجاز، ۶۵

رنگ آمیزی یالی، ۲

رنگ پذیری یالی، ۹

رنگ آمیزی مجموع، ۱۲

رنگپذیری راسی، ۹

ز

زمانبندی و تخصیص منابع، ۱۷

ش

شرایط لازم همریختی، ۴۶

ض

ضرب دکارتی دوگراف، ۸

ط

طراحی VLSI مدار الکتریکی، ۱۷

- ع
عدد رنگ آمیزی، ۳۱
عدد رنگی، ۴۱
عدد رنگی کسری، ۴۱، ۵۱
عدد زکرس-ویلف، ۳۲
- ق
قدرت راسی گراف، ۱۲
- ک
کاهش پذیری، ۲۱
کاهندگی، ۶۶
کران الگوریتمی، ۳۴
کران پایین، ۵۱
کران تحلیلی، ۳۰
- گ
گراف بازه‌ای، ۱۰
گراف برخورد، ۱۷
گراف جزئی، ۳۴
گراف جهتدار بدون دور، ۷۰
گراف راس تراپا، ۴۰
گراف کنسر، ۴۱
گراف مکمل، ۸
گراف یال تراپا، ۴۱
- م
ماتریس وقوع، ۵۸
مجموع رنگی تفاضلی، ۴۲
مجموع رنگی جمعی، ۴۲
مجموعه مستقل، ۸
مجموعه نگهدارنده، ۶۶
مدل ریاضی غیر خطی، ۵۸
مدل ریاضی مسئله MSC ، ۵۶
- مرتبه، ۵
مرحله تصدیق کردن، ۲۰
مرحله حدس زدن، ۲۰
مسائل تصمیم گیری، ۲۰
مسائل NP -کامل، ۲۰
مسئله چهار رنگ، ۲
- ن
نگهدارندگی، ۶۶
نمایش رنگ - راس گراف، ۷۶
- ه
همریختی، ۴۰
همسایگی کامل، ۶۵
همسایگی یک راس، ۶
همسایگی N_k ، ۶۳
- ی
یکریختی، ۴۰
یکهنگه کلی، ۵۸

Abstract

In this thesis, we mainly consider sum coloring problem which is very close to the classic coloring problem. This problem asks for the minimum possible summation of colors assigned properly to the vertices of a given graph. There are generally two different approaches dealing with this problem, which are namely analytic approach and algorithmic approach. To solve this problem, we introduce a meta-heuristic method, categorized as in the second approach, which is called variable neighborhood search and is based on a new structure of neighborhoods. To increase the speed of the neighborhood search process, we present some new definitions of holder vertex and set. Tested on 23 commonly used benchmark instances, our algorithm shows acceptable competitive performance with respect to recently proposed heuristics.

Moreover, the edge-difference chromatic sum and the edge-sum chromatic sum of graphs, as two generalizations of the chromatic sum, are introduced. In this regard, we present some necessary conditions for the existence of homomorphism between two graphs via these parameters. Furthermore, some upper and lower bounds for these parameters in terms of the fractional chromatic number, are introduced.

Keywords: vertex sum coloring, variable neighborhood search, holder set, holder and reducer set, vertex edge-difference chromatic sum, vertex edge-sum chromatic sum, non homomorphism theorems, fractional coloring, kneser graph.



Shahrood University of Technology

Faculty Of Mathematical Sciences

PhD Thesis in: Graph and Combinatorics

**A new approach for solving minimum sum
coloring problem and its generalizations on
simple graphs**

By: Khalil Erfani Heidarnia

Supervisors

Dr. Sadegh Rahimi Sharbaf

Dr. Jafar Fathali

Advisor

Dr. Meysam Alishahi

April 2018