



دانشگاه صنعتی شاهرود

دانشکده علوم ریاضی

گروه ریاضی کاربردی

پایان نامه

برای دریافت درجه کارشناسی ارشد در رشته
ریاضی کاربردی، گرایش تحقیق در عملیات

عنوان

جدول زمان بندی دروس

استاد راهنما

دکتر جعفر فتحعلی

استاد مشاور

دکتر محسن اصیلی

پژوهشگر

زهره سادات رسولی

۹۲/۹/۳۰

نام خانوادگی دانشجوی: رسولی

نام: زهره سادات

عنوان: جدول زمان بندی دروس

استاد راهنما: دکتر جعفر فتحعلی

استاد مشاور: دکتر محسن اصیلی

مقطع تحصیلی: کارشناسی ارشد رشته: ریاضی کاربردی گرایش: تحقیق در عملیات

دانشگاه: دانشگاه صنعتی شاهرود

تاریخ فارغ التحصیلی: ۹۲/۹/۳۰

تعداد صفحات: ۷۷

واژگان کلیدی: ۱- جدول زمان بندی ۲- جست و جوی ممنوع ۳- ابتکاری ۴- همسایگی

چکیده

مسئله‌ی زمان بندی دروس یکی از مهمترین، کاربردی ترین و زمان برترین مسایلی است که در تمام موسسات آموزشی (بصورت سالیانه یا دو بار در یک سال تحصیلی) رخ می دهد. در ابتدا جدول زمان بندی را بصورت دستی تهیه می کردند، که در خیلی از موسسات آموزشی هنوز این امر رایج می باشد. این روش به زمان زیادی نیاز دارد؛ علاوه براین ممکن است جواب حاصل تنها یک جواب شدنی باشد، نه یک جواب بهینه. در این پایان نامه، یک مدل برنامه ریزی برای حل مسئله‌ی زمان بندی درسی ارائه شده است. برای بهبود یافتن جدول زمان بندی و برطرف کردن نیازهای اساتید و دانشجویان، در این مدل محدودیت های سخت و نرم بیشتری نسبت به مدل های قبلی در نظر گرفته شده است. برای حل این مدل از سه روش جست و جوی محلی و جست و جوی محلی تصادفی و جست و جوی ممنوع استفاده می کنیم. برای بهبود این روش ها راهکارهایی نیز پیشنهاد شده است. در پایان برای بررسی کارایی این برنامه که با استفاده از نرم افزار متلب نوشته شده است، این برنامه روی داده های دانشکده‌ی ریاضی دانشگاه صنعتی شاهرود پیاده سازی شده است.

تقدیم بہ

ہمراہان، ہمیشگی

پدر و مادر م

تعهد نامه

اینجانب **زهره سادات رسولی** دانشجوی دوره کارشناسی ارشد رشته ریاضی کاربردی گرایش تحقیق در عملیات دانشکده علوم ریاضی دانشگاه صنعتی شاهرود نویسنده پایان نامه **جدول زمان بندی دروس تحت راهنمایی دکتر 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 پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

۹۳/۱۰/۱۶



دانشگاه علمی کاربردی

مدیریت تحصیلات تکمیلی

فرم شماره (۶)

باسمه تعالی

شماره:
تاریخ:
ویرایش:

فرم صورت جلسه دفاع از پایان نامه تحصیلی دوره کارشناسی ارشد

با تأییدات خداوند متعال و با استعانت از حضرت ولی عصر (عج) نتیجه ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم زهره سادات رسولی رشته ریاضی کاربردی گرایش تحقیق در عملیات تحت عنوان جدول زمان بندی دروس که در تاریخ ۹۲/۹/۳۰ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می گردد:

☐ قبول (با درجه: عالی)	☐ امتیاز (۱۹/۶)	☐ دفاع مجدد	☐ مردود
---	--	------------------------------------	--------------------------------

۲- بسیار خوب (۱۸/۹۹ - ۱۸)

۱- عالی (۲۰ - ۱۹)

۴- قابل قبول (۱۵/۹۹ - ۱۴)

۳- خوب (۱۷/۹۹ - ۱۶)

۵- نمره کمتر از ۱۴ غیر قابل قبول

عضو هیأت داوران	نام و نام خانوادگی	مرتبه علمی	امضاء
۱- استاد راهنما	دکتر جعفر فتحعلی	دانشیار	
۲- استاد مشاور	دکتر محسن اصیلی	استادیار	
۳- نماینده شورای تحصیلات تکمیلی	دکتر احمد زیره	دانشیار	
۴- استاد ممتحن	دکتر علیرضا ناظمی	استادیار	
۵- استاد ممتحن	دکتر براتاله غزنوی	استادیار	

رئیس دانشکده: دکتر احمد زیره

امضاء



سپاس گزارى...ٲ

سپاس خداوندگار حكيم را كه با لطف بى كران خود، آدمى را زيور عقل آراست. در آغاز وظيفه خود مى دانم از زحمات بى دريغ استاد راهنماى خود، جناب آقاى دكتر جعفر فتحعلى، صميمانه تشكر و قدردانى كنم كه قطعاً بدون راهنمايى هاى ارزنده ايشان، اين مجموعه به انجام نمى رسيد. از جناب آقاى دكتر محسن اصيلى كه زحمت مطالعه و مشاوره اين رساله را تقبل فرمودند و در آماده سازى اين رساله، به نحو احسن اينجانب را مورد راهنمايى قرار دادند، كمال امتنان را دارم.

زهره سادات رسولى

٩٢/٩/٣٠

فهرست مطالب

۱	تعاریف مقدماتی و توصیف مساله	۱
۱	۱.۱ مقدمه	۱
۱	۲.۱ تعاریف اولیه	۱
۴	۳.۱ توصیف مساله	۴
۶	۴.۱ روش‌های فراابتکاری برای حل مسائل بهینه سازی	۶
۱۴	۲ یک مدل برنامه‌ریزی عدد صحیح برای مساله‌ی برنامه‌ریزی دروس	۱۴
۱۸	۳ یک مدل غیرخطی برای مساله‌ی جدول زمانبندی و حل آن با جستجوی ممنوع	۱۸
۱۸	۱.۳ روش جست‌وجوی ممنوع	۱۸
۱۸	۱.۱.۳ مقدمه	۱۸
۲۰	۲.۱.۳ قالبی برای جستجوی ممنوع ساده	۲۰
۲۳	۲.۳ تشریح مدل	۲۳
۲۳	۱.۲.۳ تعیین محدودیت‌ها	۲۳
۲۵	۲.۲.۳ مدل‌بندی	۲۵
۲۹	۳.۲.۳ روش حل	۲۹
۳۶	۳.۳ نتایج	۳۶
۳۹	۴ یک مدل غیرخطی برای مساله‌ی زمانبندی دروس و به‌کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن	۳۹
۳۹	۱.۴ مقدمه	۳۹
۳۹	۲.۴ محدودیت‌ها و پارامترها	۳۹
۴۱	۳.۴ فرمول‌بندی مساله	۴۱
۴۳	۴.۴ روش‌های حل	۴۳
۴۳	۱.۴.۴ خواص الگوریتم جست‌وجوی ممنوع مورد استفاده	۴۳

۴۶	جست‌وجوی موضعی:	۲.۴.۴
۴۷	روش‌های پیشنهادی برای بهبود روش‌های حل	۵.۴
۴۸	پیاده‌سازی روش‌های حل روی چند مساله با داده‌های تصادفی	۶.۴
۵۲	پیاده‌سازی روش برای مساله‌ی زمانبندی دروس دانشکده‌ی ریاضی شاهرود	۷.۴
۵۶	توصیف داده‌های موجود	۱.۷.۴
۵۹	نتیجه‌گیری	۸.۴

۶۳	آ کد Matlab
۶۳	بخشی از کدنویسی روش جست‌وجوی ممنوع

۷۵	مراجع
----	-------

فصل ۱

تعاریف مقدماتی و توصیف مساله

۱.۱ مقدمه

رویکرد علمی به مساله‌ی برنامه‌ریزی عملیات و زمان‌بندی، ریشه در انقلاب صنعتی و تلاش‌های گانت^۱ دارد. در اغلب مسایل زمان‌بندی، تنوع حالات مختلف مساله و ترکیب‌های حاصل از این تنوع، طراحان را با تعداد زیادی از راه‌حل‌ها روبرو می‌سازد. انتخاب بهترین راه‌حل از میان این طیف گسترده، بدون استفاده از رویکردهای برنامه‌ریزی ریاضی غیرممکن است. در این فصل به بیان تعاریف اولیه و توصیف مساله‌ی جدول زمان‌بندی^۲ می‌پردازیم و همچنین اشاره‌ی مختصری به تاریخچه و کارهای گذشته‌ای که در این زمینه انجام شده است، می‌شود.

۲.۱ تعاریف اولیه

مساله‌ی زمان‌بندی از جمله مسائلی است که در زمینه‌های گوناگون، از قبیل آموزش، حمل‌ونقل و ... به کار برده می‌شود. به عنوان مثال می‌توان به موارد زیر اشاره کرد؛
زمان‌بندی شیف کاری پرستاران یک بیمارستان
زمان‌بندی حرکت قطارها و هواپیماها
زمان‌بندی ارایه‌ی دروس و امتحانات برای مراکز آموزشی
زمان‌بندی کارگران یک کارخانه در شیف‌های مختلف در طول شبانه‌روز
مساله‌ی زمان‌بندی حدوداً از سال ۱۹۶۰، نظر محققین را به خود جلب کرده است و کار در این زمینه بصورت گسترده همچنان ادامه دارد. این مساله در مواردی که منابع بسیار زیاد و نامحدود هستند (یعنی اندازه‌ی مساله بزرگ است) و همچنین در مواردی که جدول مربوطه نیاز به بروزرسانی دارد، از اهمیت ویژه‌ای برخوردار

^۱Henry Gunt

^۲Timetabling

است، زیرا تهیهی جدول زمانبندی مذکور امری مشکل و زمانبر است. قبل از اینکه به معرفی مسالهی زمانبندی بپردازیم، ابتدا تعاریفی را ارائه می‌کنیم که برای درک بهتر مطالب، مفید می‌باشند.

تعریف ۱.۲.۱. مسالهی حمل و نقل

در حالت کلی، یک مسالهی حمل و نقل با اطلاعات زیر شناخته می‌شود:

۱. مجموعه‌ای از m مبدا یا نقطه‌ی عرضه که وظیفه‌ی ارسال را انجام می‌دهد. مبدا i حداکثر می‌تواند s_i واحد عرضه کند.

۲. مجموعه‌ای از n مقصد یا نقطه‌ی تقاضا که وظیفه‌ی دریافت را انجام می‌دهد. مقصد j باید حداقل d_j واحد دریافت کند.

۳. به هر واحدی که از مبدا i به مقصد j ارسال صورت می‌گیرد، هزینه‌ای معادل c_{ij} تعلق می‌گیرد. هدف در مسالهی حمل و نقل، مینیم کردن هزینه است.

فرض کنید

x_{ij} : تعداد واحدهای ارسالی از مبدا i به مقصد j

در این صورت، فرمول کلی یک مسالهی حمل و نقل، عبارت است از:

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} \leq s_i \quad (i = 1, \dots, m) \\ & \sum_{i=1}^m x_{ij} \geq d_j \quad (j = 1, \dots, n) \\ & x_{ij} \geq 0 \quad (i = 1, \dots, m), (j = 1, \dots, n) \end{aligned} \quad (1.1)$$

مساله‌ای با قیود (۱.۱) و تابع هدف ماکسیم‌سازی نیز یک مسالهی حمل و نقل است که در این حالت هدف ماکسیم کردن سود می‌باشد.

زمانی که $\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$ ، یعنی عرضه‌ی کل برابر تقاضای کل باشد، مساله را مساله‌ی حمل و نقل متعادل می‌نامیم.

تعریف ۲.۲.۱. مسالهی تخصیص

مساله‌ی تخصیص یک حالت خاص از مساله‌ی حمل و نقل است. به بیان دیگر یک مساله‌ی تخصیص، عبارت

است از یک مساله‌ی حمل‌ونقل متعادل، که میزان عرضه و تقاضا در کلیه‌ی مبداها و مقصدها یک است. در یک مساله‌ی تخصیص، هر نقطه‌ی مبدا (عرضه) باید تنها به یک نقطه‌ی مقصد (تقاضا)، اختصاص یابد و هر نقطه‌ی مقصد (تقاضا) باید به یک نقطه‌ی مبدا (عرضه) اختصاص یابد. در این مساله نیز هدف مینیم کردن هزینه می‌باشد. صورت کلی این مساله به صورت زیر است؛

$$\begin{aligned} \min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ s.t. \\ \sum_{j=1}^n x_{ij} = 1 \quad (i = 1, \dots, m) \\ \sum_{i=1}^m x_{ij} = 1 \quad (j = 1, \dots, n) \\ x_{ij} \geq 0 \quad (i = 1, \dots, m), (j = 1, \dots, n) \end{aligned} \quad (2.1)$$

محدودیت $\sum_{j=1}^n x_{ij} = 1$ ، به‌ازای هر نقطه‌ی مبدا (عرضه) تضمین می‌کند هر مبدا به یک نقطه‌ی عرضه اختصاص یابد و محدودیت $\sum_{i=1}^m x_{ij} = 1$ ، به‌ازای هر نقطه‌ی (تقاضا، تضمین می‌کند هر نقطه‌ی مقصد (تقاضا) دقیقاً به یک نقطه‌ی مبدا (عرضه) اختصاص یابد.

در حالت کلی مساله‌ی زمان‌بندی، به عنوان یک مساله‌ی تخصیص در نظر گرفته می‌شود. زیپنگ لویی^۳ و جین کائو^۴ در [۱۶]، جدول زمان‌بندی را مجموعه‌ای از جلسات بیان کرده‌اند که باید به تعدادی بازه‌ی زمانی و کلاس که تابع یک گروه از محدودیت‌هاست، اختصاص داده شوند. لویس^۵ و همکاران [۹]، مساله‌ی زمان‌بندی را بصورت زیر تعریف کرده‌اند؛ تخصیص منابع و امکانات داده شده با در نظر گرفتن تقریباً تمام مجموعه‌ی محدودیت‌ها و اهداف مطلوبی که از قبل تعیین شده است.

برک^۶ و همکاران [۲]، یک تعریف کلی از جدول زمان‌بندی را بصورت زیر بیان می‌کنند؛

مساله‌ی جدول زمان‌بندی مساله‌ای با چهار پارامتر می‌باشد:

یک مجموعه‌ی متناهی از زمان‌ها، یک مجموعه‌ی متناهی از منابع، یک مجموعه‌ی متناهی از جلسات، بطوری که یک‌سری از محدودیت‌ها تا حد امکان رعایت شوند.

از میان انواع گسترده‌ی جدول زمان‌بندی، جدول زمان‌بندی آموزشی بصورت گسترده‌ای از دیدگاه علمی مورد مطالعه قرار می‌گیرد. این مساله یکی از مهمترین، کاربردی‌ترین و زمان برترین مسائلی است که در تمام موسسات آموزشی رخ می‌دهد. همان‌طور که قبلاً گفته شد، مساله‌ی جدول زمان‌بندی آموزشی می‌تواند

^۳ Zhipeng Lu

^۴ Jin-Kao

^۵ Luis

^۶ Burk

بصورت یک مساله‌ی تخصیص تعریف شود که هدف تخصیص یک‌سری منابع (دروس و امتحانات و ...) و یا مسولیت‌های شغلی (مانند اساتید) به تعداد محدودی از بازه‌های زمانی است، بصورتی که یک‌سری از محدودیت‌ها و اهداف در نظر گرفته شوند. هدف ما در این پایان‌نامه تمرکز روی مساله‌ی زمان‌بندی دروس در دانشگاه است.

۳.۱ توصیف مساله

مساله‌ی جدول زمان‌بندی دروس دانشگاهی یک مساله‌ی NP-سخت^۷ است که با تعداد زیادی درس، استاد، کلاس و بازه‌های زمانی سروکار دارد. بنابراین دستیابی به جواب در یک زمان معقول امکان‌پذیر نخواهد بود. در این مساله پس از مشخص شدن دروس، اساتید، کلاس‌ها و بازه‌های زمانی باید به تخصیص دروس به بازه‌های زمانی پرداخت به‌طوری که مجموعه‌ای از محدودیت‌ها رعایت شوند و اهداف مطلوب نیز برآورده شوند. سرانجام باید جدولی به‌دست آید که مشخص کند هر درس در چه ساعتی توسط استاد مربوطه ارائه شود.

مساله‌ی جدول زمان‌بندی آموزشی کاربردهای متفاوتی از جمله در مدارس، دانشگاه‌ها و موسسات آموزشی دارد. این مسائل از جهاتی به هم شبیه‌اند و از جهاتی با هم متفاوتند. به عنوان مثال در همه‌ی این مسائل اساتید تنها می‌توانند به یک گروه از دانشجویان، در یک زمان و در یک کلاس، درس دهند ولی در جدول زمان‌بندی مربوط به دانشگاه‌ها حق انتخاب برای دانشجویان وجود دارد در حالی که در جدول زمان‌بندی مربوط به مدارس این امکان وجود ندارد. جدول زمان‌بندی مربوط به دانشگاه‌ها از اندازه‌ی بزرگ‌تر نسبت به مدارس برخوردار است و در نتیجه حل آن مشکل‌تر از مسائل مدارس می‌باشد.

به‌طور کلی مسائل زمان‌بندی دانشگاهی به دو دسته‌ی عمده تقسیم می‌شوند:

- **مسائل جدول زمان‌بندی دروس:** جدول زمان‌بندی دروس درگیر این مساله می‌باشد که دانشجویان، متقاضی یک‌سری دروس (دروس تئوری، سمینار، آزمایشگاه، دروس خودآموز و غیره) می‌باشند که هدف مینیم کردن ناسازگاری بین آن‌ها و برقراری اهداف و محدودیت‌هایی می‌باشد که توسط موسسه‌ی آموزشی مربوطه مطلوب است.

- **مساله‌ی زمان‌بندی امتحانات:** اکثر موسسات آموزشی باید مجموعه‌ای از امتحانات را در پایان هر ترم یا هر سال تحصیلی زمان‌بندی کنند. به شکل ساده‌تر این مساله می‌تواند به عنوان یک مساله‌ی تخصیص مطرح شود که هدف تخصیص یک مجموعه از امتحانات به تعداد ثابتی از بازه‌های زمانی است، به طوری که هیچ دانشجویی بیش از یک امتحان در یک زمان خاص نداشته باشد. از طرفی

^۷NP-hard (حل نشدنی در زمان چندجمله‌ای (زمان معقول) برحسب اندازه‌ی ورودی مساله)

یکسری محدودیت‌ها و اهداف دیگر نیز باید رعایت شوند که با توجه به مراکز آموزشی، این اهداف متفاوت می‌باشند.

مساله‌ی زمان‌بندی آموزشی معمولاً براساس یکی از دو معیار زیر ایجاد می‌شود:

۱- جدول زمان‌بندی براساس ثبت‌نام

۲- جدول زمان‌بندی براساس برنامه‌ی درسی (هفتگی)

تفاوت این دو دسته این است که در جدول زمان‌بندی براساس پیش ثبت‌نام، تخصیص دروس براساس اطلاعات پایه‌ی دانشجویان انجام می‌گیرد، در حالی که در جدول زمان‌بندی مبتنی بر برنامه‌ی درسی، این تخصیص براساس برنامه‌های آموزشی منتشر شده توسط دانشگاه انجام می‌شود.

به خاطر کاربرد زیادی که این مساله دارد، محققان زیادی مساله‌ی زمان‌بندی آموزشی (درسی و امتحانی) را مطالعه کرده‌اند و مدل‌ها و الگوریتم‌های مختلفی برای حل آن نیز ارائه کرده‌اند. به عنوان مثال میر حسینی^۸ در [۱۲] یک مدل برنامه‌ریزی عدد صحیح را برای حل جدول زمان‌بندی داده‌های دانشگاه صنعتی شاهرود ارائه داده است. آلوارز والدس^۹ در [۱۴] الگوریتمی برای حل مساله‌ی زمان‌بندی مدارس اسپانیایی ارائه کرده است که در آن هدف پیدا کردن یک جواب شدنی برای کلاس‌ها و معلمان است. دریس^{۱۰} و همکاران [۴]، یک روند حل برای مساله‌ی جدول زمان‌بندی دروس دانشگاهی بیان کرده‌اند. الگوریتم ارائه شده توسط آن‌ها، برای یک مثال واقعی پیاده‌سازی شده است که در کمتر از ۳۳ دقیقه جدولی مناسب برای دانشگاه به‌دست می‌آورد. همچنین کارتر^{۱۱} در [۱۰] و [۱۱]، بسیاری از روش‌های به کار رفته تا سال ۱۹۸۶ را که در رابطه با مساله‌ی جدول زمان‌بندی دروس به کار رفته است، بیان کرده است.

همان‌طور که بیان شد برای به‌دست آوردن جدول زمان‌بندی، باید مجموعه‌ای از محدودیت‌ها رعایت شوند. در حالت کلی محدودیت‌های مطرح شده در این مساله به دو دسته تقسیم می‌شوند:

محدودیت‌های سخت^{۱۲}: برای به‌دست آوردن یک جدول زمان‌بندی، یک‌سری از محدودیت‌ها وجود دارند که رعایت آن‌ها الزامی است. در صورتی که یکی از این محدودیت‌ها رعایت نشود، جدول حاصل نشدنی است. به این دسته از محدودیت‌ها، محدودیت‌های سخت گویند.

محدودیت‌های نرم^{۱۳}: رعایت این دسته از محدودیت‌ها میزان کارایی و مطلوبیت جدول را مشخص می‌کند. به عبارت دیگر رعایت این دسته از محدودیت‌ها الزامی نیست، اما هر چه بیشتر این محدودیت‌ها رعایت شوند، جدول زمان‌بندی مطلوب‌تر است. البته در موارد واقعی رعایت تمام محدودیت‌های نرم مشکل و در خیلی از موارد غیرممکن است.

^۸Mirhassani

^۹Alvarez-Valdes

^{۱۰}Deris

^{۱۱}Carter

^{۱۲}Hard Constraints

^{۱۳}Soft Constraints

با مطالعه روی مقالاتی که در دهه‌ی اخیر به عمل آورده‌ایم، تعدادی محدودیت سخت و نرم متداول را برای دروس دانشگاهی جمع‌آوری کرده‌ایم که در جداول ۱.۱ و ۲.۱ آن‌ها را بیان کرده‌ایم.

برخی از محدودیت‌ها توسط برخی از محققین به عنوان محدودیت‌های سخت در نظر گرفته شده است، در حالی که همین محدودیت‌ها توسط دیگران به عنوان محدودیت‌های نرم در نظر گرفته شده است. برای مثال محدودیت H_q ، توسط آلاداگ^{۱۴} [۱] به عنوان محدودیت نرم در نظر گرفته شده، در حالی که همین محدودیت در مقاله‌ی لویی و جین کایو [۱۶] به عنوان محدودیت سخت در نظر گرفته شده است. همچنین محدودیت S_m توسط لویس [۹]، به عنوان یک محدودیت سخت در نظر گرفته شده است در حالی که توسط پانگچاروئنا^{۱۵} [۱۳] و دریس [۴] به عنوان یک محدودیت نرم در نظر گرفته شده است.

۴.۱ روش‌های فراابتکاری برای حل مسائل بهینه سازی

بهینه‌سازی یک فعالیت مهم و تعیین‌کننده در طراحی ساختاری است. طراحان زمانی قادر خواهند بود طرح‌های بهتری تولید کنند که بتوانند با روش‌های بهینه‌سازی در صرف زمان و هزینه طراحی صرفه‌جویی نمایند. بسیاری از مسائل بهینه‌سازی در مهندسی، طبیعتاً پیچیده‌تر و مشکل‌تر از آن هستند که با روش‌های مرسوم بهینه‌سازی نظیر روش برنامه‌ریزی ریاضی و نظایر آن قابل حل باشند. به‌طور کلی مسائل بهینه‌سازی از نظر نوع متغیرهای تصمیم می‌تواند به دو گروه مسائل گسسته^{۱۶} و مسائلی با متغیرهای پیوسته^{۱۷} تقسیم‌بندی شود. در عمل ممکن است مسائلی به صورت مخلوط^{۱۸} تعریف شود که همزمان دارای متغیرهای گسسته و پیوسته باشد. هنگامی که متغیرهای تصمیم در مسائل بهینه‌سازی گسسته باشند، مساله‌ی یافتن جواب‌های بهینه، بهینه‌سازی ترکیبی^{۱۹} نامیده می‌شود. مثالی ساده از مساله بهینه‌سازی ترکیبی، مساله فروشنده دوره گرد می‌باشد، سفر بین n شهر طوری که از همه شهرها تنها یک بار بگذریم و در نهایت به نقطه شروع بازگردیم. هدف، پیدا کردن کوتاه‌ترین مسیر است.

در زمینه‌ی مسائل پیوسته، روش‌های کلاسیک بسیاری برای بهینه‌سازی سراسری ارایه شده است، اما این تکنیک‌ها در صورتی که تابع هدف دارای ویژگی‌های مشخصی (نظیر محدب بودن^{۲۰}) نباشد، معمولاً کارایی ندارند. روش‌های حل مسائل بهینه‌سازی پیوسته به دو دسته‌ی روش‌های بهینه‌سازی خطی و غیرخطی تقسیم‌بندی می‌شوند. روش‌های حل بهینه‌سازی ترکیبی را نیز می‌توان به دو قسمت الگوریتم‌های دقیق^{۲۱} و

^{۱۴} Aladag

^{۱۵} Pongcharoena

^{۱۶} Discrete

^{۱۷} Continuous

^{۱۸} Mixed Problem

^{۱۹} Combinational Optimization

^{۲۰} Convexity

^{۲۱} Exact Algorithms

جدول ۱.۱: محدودیت‌های سختی که به‌صورت گسترده توسط محققین در مقالات مختلف استفاده شده‌اند.

H_1	هر جلسه از یک درس باید در یک زمان متمایز و در یک کلاس متمایز زمان‌بندی شود.
H_2	در یک کلاس درس و در یک زمان، بیش از یک درس نمی‌تواند ارائه شود.
H_3	دروسی که در یک برنامه‌ی درسی واحد قرار دارند، نباید در یک بازه‌ی زمانی زمان‌بندی شوند.
H_4	هیچ استادی نمی‌تواند هم‌زمان دو درس را تدریس کند.
H_5	هیچ کلاسی نمی‌تواند به‌طور هم‌زمان به دو درس اختصاص داده شود.
H_6	کلاس‌های تخصیص داده شده به دروس باید متناسب با نوع آن‌ها باشد.
H_7	در یک زمان و در یک روز تنها به تعداد کلاس‌های درس موجود می‌تواند درس ارائه شود.
H_8	به هر درس باید به تعداد جلسات مورد نیاز، بازه‌ی زمانی اختصاص داده شود.
H_9	در بازه‌های زمانی که اساتید در دسترس نیستند، نباید به آن‌ها درسی تخصیص داده شود.

جدول ۲.۱: محدودیت‌های نرمی که به صورت گسترده توسط محققین در مقالات مختلف استفاده شده‌اند.

S_1	بین دو جلسه از یک درس یک روز فاصله وجود داشته باشد.
S_2	جدول حاصل باید تا حد امکان برای دانشجویان فشرده باشد.
S_3	تعداد دانشجویان هر درس نباید بیشتر از ظرفیت کلاس تخصیص داده شده باشد.
S_4	جلسات یک درس باید در کمترین روزهای کاری گسترده شود.
S_5	در جدول زمان‌بندی اساتید نباید فاصله وجود داشته باشد.
S_6	ظرفیت آموزشی هر استاد باید رعایت شود.
S_7	برخی دروس باید به بازه‌های زمانی و یا به روزهای خاصی اختصاص یابند.
S_8	برخی دروس باید در ساعت خاصی ارائه شوند.

الگوریتم‌های ابتکاری^{۲۲} تقسیم‌بندی کرد. بهینه‌سازی غیرخطی پیوسته و روش‌های تقریبی مسائل گسسته، جزو مسائل سخت محسوب می‌شوند. الگوریتم‌های دقیق برای حل مسائل بهینه‌سازی ترکیبی نظیر شاخه و کران، روش‌هایی هستند که رسیدن به جواب بهینه را تضمین می‌کنند. روش‌های دقیق، معمولاً برای حل مسائل با اندازه‌های کوچک یا متوسط می‌توانند به کار گرفته شوند. این روش‌ها رسیدن به جواب بهینه را تضمین می‌کنند اما برای حل مسائل واقعی که معمولاً با ابعاد بزرگ و پیچیده هستند، امکان استفاده از روش‌های دقیق وجود ندارد. بنابراین غالباً از روش‌های ابتکاری برای حل مسائل بهینه‌سازی ترکیبی بزرگ استفاده می‌شود.

بیش‌تر مسائل بهینه‌سازی ترکیبی، در کلاس مسائل NP-سخت قرار می‌گیرند. بنابراین برای حل آن‌ها باید از روش‌های تقریبی که جواب‌های نزدیک به بهینه را در زمانی قابل قبول به دست می‌آورند، استفاده کرد. الگوریتم‌هایی از این دست الگوریتم‌های ابتکاری نام دارند. الگوریتم‌های ابتکاری معمولاً به دنبال رسیدن به جواب‌های خوب، یعنی جواب‌های نزدیک به بهینه در زمان محاسباتی قابل قبول هستند ولی قادر به تضمین بهینه بودن جواب‌های به دست آمده نیستند. روش‌های ابتکاری مختلفی برای حل مسائل بهینه‌سازی ترکیبی ابداع شده است که در آن‌ها جواب نزدیک به بهینه تولید می‌شود. مشکل این روش‌ها، این است که اکثر آن‌ها فقط در مورد یک مسأله‌ی به خصوص قابل کاربرد هستند.

روش جست‌وجوی موضعی^{۲۳}، یک روش ابتکاری است که از یک جواب اولیه شروع می‌کند و در هر تکرار جواب فعلی را با یک همسایه جایگزین می‌کند. در این روش، جواب همسایه‌ای که جواب فعلی می‌شود باید مقدار تابع هدف را بهبود بخشد و مقدار آن از تمامی همسایه‌های شدنی بهتر باشد. جست‌وجوی زمانی پایان می‌یابد که همسایه‌های کاندید، بدتر از جواب فعلی باشند و این بدان معناست که بهینه‌ی محلی یافت شده است.

یکی از عیب‌های الگوریتم‌های ابتکاری، این است که این روش‌ها نسبت به جواب اولیه بسیار حساس هستند. به این معنا که با شروع از جواب‌های اولیه‌ی مختلف معمولاً جواب‌های نهایی متفاوتی به دست می‌آید. اشکال دیگر این روش‌ها تولید یک جواب یا تعداد محدودی از جواب‌ها و توقف آن‌ها در بهینه‌ی محلی با کیفیت پایین است. الگوریتم‌های فراابتکاری، برای حل این مشکلات و عیب‌های روش‌های ابتکاری پیشنهاد شده‌اند. الگوریتم‌های فراابتکاری، با معرفی اصول سیستماتیک برای خارج شدن از بهینه‌ی محلی یا جلوگیری از قرار گرفتن در بهینه‌های محلی با این مشکل برخورد می‌کنند. ویژگی مشترک الگوریتم‌های فراابتکاری استفاده از مکانیزم‌های خروج از بهینه‌ی محلی است.

این اصول سیستماتیک به‌طور کلی به چهار دسته تقسیم می‌شوند:

- شروع از جواب‌های اولیه‌ی متفاوت به صورت تکرارشونده: این استراتژی در الگوریتم‌هایی نظیر جست‌وجوی محلی چند شروعی، جست‌وجوی محلی تکرارشونده^{۲۴} و . . . به کار برده می‌شود.

^{۲۲}Heuristic Algorithms

^{۲۳}Local Search

^{۲۴}Iterative Local Search

- **پذیرش جواب‌های بدون بهبود:** این رویکرد امکان اجرای حرکت‌هایی را فراهم می‌سازد که جواب فعلی را بدتر می‌کنند. به وسیله‌ی این رویکرد، فرار از یک بهینه‌ی محلی امکان‌پذیر می‌شود. تبرید شبیه‌سازی شده و جستجوی ممنوعه^{۲۵}، متداولترین الگوریتم‌هایی هستند که از این استراتژی استفاده می‌کنند.
- **تغییر همسایگی:** این رویکرد شامل تغییر در ساختار همسایگی در طول جستجو است. به طور مثال، این رویکرد در استراتژی جستجوی همسایه‌ی متغیر، مورد استفاده قرار می‌گیرد.
- **تغییر تابع هدف یا داده‌های ورودی مساله:** در این رویکرد، مساله در اثر ایجاد یک به هم ریختگی در داده‌های ورودی مساله، تابع هدف و یا محدودیت‌های آن، تغییر می‌یابد، به این امید که جواب‌های بهتری برای مساله به دست آید. این رویکرد در جستجوی محلی هدایت شده مورد استفاده قرار می‌گیرد.

الگوریتم‌های فراابتکاری

یک الگوریتم فراابتکاری، الگوریتمی است که می‌تواند با تغییرهای کم، برای مسائل بهینه‌سازی مختلف به کار رود. استفاده از الگوریتم‌های فراابتکاری به‌طور قابل ملاحظه‌ای توانایی یافتن جواب‌های با کیفیت بالا را برای مسائل بهینه‌سازی ترکیبی افزایش می‌دهد. به عبارت دیگر، یک الگوریتم فراابتکاری یک روش ابتکاری است که قادر به جست‌وجوی فضای جواب برای یافتن جواب با کیفیت بالاست. روش‌های فراابتکاری دارای ویژگی‌های مشترک زیر هستند:

- این روش‌ها همگی تا حدودی احتمالی هستند. چنین رویکردی این امکان را می‌دهد که از قرار گرفتن الگوریتم در دام بهینه‌ی محلی جلوگیری شود.
 - این الگوریتم‌ها معمولاً برای مسائل گسسته ارایه شده‌اند، اما امکان کاربرد در مسائل پیوسته را نیز دارند.
 - این مسائل معمولاً از مفاهیم زیست‌شناسی، رفتارشناسی جانوری و فیزیک الهام گرفته شده‌اند.
 - یکی از عیب‌های مشترک در این روش‌ها، دشوار بودن تنظیم و تطبیق پارامترهاست.
- از معیارهای مختلفی برای طبقه‌بندی الگوریتم‌های فراابتکاری استفاده می‌شود که مواردی از آن‌ها در زیر آمده است.

- **مبتنی بر یک جواب و مبتنی بر جمعیت:** الگوریتم‌های مبتنی بر یک جواب، در حین فرآیند جست‌وجو یک جواب را تغییر می‌دهند. در حالی که در الگوریتم‌های مبتنی بر جمعیت، در حین جست‌وجو یک

^{۲۵} Tabu Search

جمعیت از جواب‌ها در نظر گرفته می‌شوند. ویژگی این دو نوع الگوریتم تکمیل‌کننده‌ی یکدیگر هستند. روش‌های فراابتکاری مبتنی بر یک جواب، این توانایی را دارند که بر روی مناطق محلی جست‌وجو تمرکز کنند. در مقابل الگوریتم‌های فراابتکاری مبتنی بر جمعیت می‌توانند جست‌وجو را به مناطق مختلفی از فضای جواب هدایت کنند. در جدول ۳.۱ فهرست کامل اسامی این الگوریتم‌ها آورده شده است.

- **الگوریتم‌های الهام گرفته از طبیعت و بدون الهام از طبیعت:** بسیاری از الگوریتم‌های فراابتکاری از طبیعت الهام گرفته‌اند. به‌عنوان مثال الگوریتم‌های تکاملی که از زیست‌شناسی الهام گرفته‌اند و یا الگوریتم مورچه‌ها، کلونی زنبورها و بهینه‌سازی گروهي ذرات که از هوش جمعی در گونه‌های مختلف موجودات الهام گرفته‌اند و تبرید شبیه‌سازی شده که از فیزیک الهام گرفته است.

- **الگوریتم‌های با حافظه و بدون حافظه:** برخی از الگوریتم‌های فراابتکاری فاقد حافظه هستند. به این معنا که این نوع الگوریتم‌ها از اطلاعات به‌دست آمده در حین جست‌وجو استفاده نمی‌کنند. بعضی از نمونه‌های این دسته از الگوریتم‌ها عبارت‌اند از: روش جست‌وجوی انطباقی حریصانه و تبرید شبیه‌سازی شده. این در حالی است که برخی از الگوریتم‌های فراابتکاری نظیر جست‌وجوی ممنوعه از حافظه استفاده می‌کنند. این حافظه اطلاعات به‌دست آمده در طول جست‌وجو را در خود ذخیره می‌کند.

- **الگوریتم‌های قطعی و احتمالی:** در الگوریتم‌های قطعی، مساله با استفاده از تصمیم‌های قطعی حل می‌شود اما در الگوریتم‌های فراابتکاری احتمالی یک سری قوانین احتمالی در طول جست‌وجو مورد استفاده قرار می‌گیرد. در الگوریتم‌های قطعی با شروع از یک جواب اولیه ی یکسان به یک جواب نهایی مشابه می‌رسیم در حالی که در الگوریتم احتمالی ممکن است از یک جواب اولیه‌ی مشابه به جواب‌های نهایی مختلفی برسیم. روش جست‌وجوی ممنوعه یک الگوریتم فراابتکاری قطعی و روش تبرید شبیه‌سازی شده یک الگوریتم احتمالی است.

- **الگوریتم‌های حریصانه و تکرارشونده:** الگوریتم‌های تکرارشونده با یک جواب (یا جمعیتی از جواب‌ها) کامل شروع می‌کنند و آن را در هر تکرار با استفاده از برخی عملگرهای جست‌وجو تغییر می‌دهند. الگوریتم‌های حریصانه با یک جواب خالی شروع و در هر گام، یک متغیر تصمیم از مساله را مقداردهی می‌کنند تا این که یک جواب کامل به‌دست آید. بیشتر الگوریتم‌های فراابتکاری تکرارشونده هستند.

انتخاب استراتژی حل

برای برخی از مسائل رایج، روش‌های حل بسیاری وجود دارد. برای انتخاب بهتر رویکرد حل و تعیین بهتر اهداف باید روش‌های حل موجود برای مساله‌ی مورد نظرمان را مرور کنیم. پس از مرور روش‌های حل مساله

جدول ۳.۱: فهرست الگوریتم‌های مبتنی بر یک جواب و الگوریتم‌های مبتنی بر جمعیت

الگوریتم‌های فراابتکاری مبتنی بر یک جواب	الگوریتم‌های فراابتکاری مبتنی بر یک جمعیت
جست‌وجوی محلی هدایت‌شده ^آ	بهینه‌سازی کلونی مورچگان ^ب
روش جست‌وجوی انطباقی حریصانه ^ج	کلونی زنبورها ^د
جست‌وجوی محلی تکرارشونده	الگوریتم ژنتیک ^ه
تبرید شبیه‌سازی شده ^و	برنامه‌ریزی تکاملی ^ز
جست‌وجوی ممنوعه ^ح	استراتژی تکاملی ^ط
روش‌های هموارسازی ^ی	جست‌وجوی پراکنشی ^{یا}
روش‌های نوسانی ^{یب}	بهینه‌سازی گروهی ذرات ^{یج}

- ^آ Guided Local Search
^ب Ant Colony Optimization
^ج Greedy Adaptive Search Procedure
^د Bee Colony
^ه Genetic Algorithm
^و Simulated Annealing
^ز Evolutionary Programming
^ح Tabu Search
^ط Evolutionary Strategies
^ی Smoothing Methods
^{یا} Scatter Search
^{یب} Noisy Methods
^{یج} Particle Swarm Optimization

باید قوت‌ها و ضعف‌های آن‌ها به‌طور مثال کارایی، استفاده از حافظه و . . . را تعیین کنیم. در این گام، پس از بررسی روش‌های حل موجود برای مساله‌ی مورد نظر، باید ضرورت استفاده از روش‌های فراابتکاری مشخص شود. در واقع، در این گام باید تصمیم‌گیری شود که آیا استفاده از الگوریتم‌های فراابتکاری ضرورتی دارد یا خیر؟ به‌طور کلی، اگر مساله در ابعاد بزرگ نباشد و پیچیدگی بالایی نداشته باشد، آنگاه ما نیاز به یک الگوریتم دقیق داریم. در غیر این صورت، یعنی هنگامی که ابعاد مساله بزرگ و پیچیده باشد، نمی‌توان از الگوریتم‌های دقیق استفاده کنیم و باید الگوریتم‌های ابتکاری و فراابتکاری را به‌کار ببریم.

مساله‌ی زمان‌بندی دروس دانشگاهی معمولاً در ابعاد (دروس، اساتید، بازه‌های زمانی و ...) گسترده است. بنابراین پیچیدگی زمانی این نوع مسائل معمولاً نمی‌تواند چندجمله‌ای باشد و از نوع نمائی است. لذا برای حل این مساله اغلب از روش‌های ابتکاری یا فراابتکاری استفاده می‌شود.

از جمله مقالاتی که در آن از الگوریتم جستجوی ممنوع برای حل مساله‌ی جدول زمان‌بندی دروس استفاده شده است می‌توان به مقاله‌ی آلاداگ و همکاران [۱]، اشاره کرد. آنها در این مقاله کارایی الگوریتم جستجوی ممنوع را با به‌کارگیری دو ساختار همسایگی جدید، مورد بررسی قرار می‌دهند. دیمپوئولو [۲۶] [۵]، یک کاربرد جدید از الگوریتم ترکیبی گره‌بندی ژنتیک را در رابطه با یک مساله‌ی جدول زمان‌بندی دانشگاهی بیان می‌کند. در [۳]، یک الگوریتم چند هدفه‌ی ژنتیک برای حل مساله‌ی جدول زمان‌بندی مربوط به یک موسسه‌ی آموزشی بیان می‌شود. در این مقاله هر جواب مساله‌ی جدول زمان‌بندی، به عنوان یک کروموزوم از نوع ماتریس در نظر گرفته می‌شود و بر اساس عملگرهای ژنتیک خاص فرض شده و جهش‌های توسعه‌یافته، جمعیت دوم کروموزوم‌ها با بعدی ثابت، ایجاد می‌شوند. اسلوانی [۱۵]، از ترکیب دو روش جستجوی ممنوع و ژنتیک برای حل مساله‌ی جدول زمان‌بندی دروس دانشگاهی استفاده کرد.

در مقالات [۷] و [۸]، از روش شبیه‌سازی تبریدی برای حل مساله‌ی زمان‌بندی استفاده شده است.

فصل ۲

یک مدل برنامه‌ریزی عدد صحیح برای مسالهی برنامه‌ریزی دروس

میرحسینی در [۱۲] با عنوان "مدل برنامه‌ریزی عدد صحیح برای مسالهی برنامه‌ریزی درسی" یک مدل برنامه‌ریزی عدد صحیح برای حل مسالهی زمانبندی ارایه کرده است. انعطاف‌پذیری و قدرت برنامه‌ریزی عدد صحیح، این امکان را به ما می‌دهد که مدل را در جهت‌های مختلفی گسترش دهیم و نیازها و محدودیت‌های خود را اعمال کنیم. میرحسینی در این مقاله فرمول‌بندی جدیدی از مسالهی برنامه‌ریزی دانشگاهی ارایه کرده است.

در این مقاله بیان می‌شود که مسالهی جدول زمانبندی دروس شامل زمانبندی تعدادی کلاس درسی بین مدرسان و دانشجویان در یک مجموعه از ساعات‌های آموزشی است که باید یکسری از محدودیت‌ها را نیز ارضا نماید. این کار به تعداد دانشجویان، مدرسان، کلاس‌ها و ساعات‌های آموزشی وابسته است. یک زمانبندی بهینه بگونه‌ای است که در هر ساعت هیچ استاد، دانشجو یا کلاسی بیش از یک بار مورد استفاده قرار نگیرد.

مدل ریاضی باید وضعیت درست و دقیق مدرسان، دانشجویان، درس‌ها و کلاس‌ها و هم چنین خواسته‌های دانشکده و محدودیت‌ها را نشان دهد. در این مقاله متغیرهای تصمیم به صورت زیر تعریف شده‌اند:

$$x_{lcdh} = \begin{cases} 1 & \text{اگر استاد } l \text{ درس } c \text{ را در روز } d \text{ و در زمان } h \text{ تدریس کند} \\ 0 & \text{در غیر این صورت} \end{cases} \quad (1.2)$$

تقدم برای استفاده از ساعات آموزشی به واسطه‌ی پارامتر $p_h \geq 0$ مشخص می‌شود که هر چه مقدار آن بیشتر باشد، تمایل کمتر برای تدریس در آن ساعت را نشان می‌دهد. تعداد دفعاتی که محدودیت‌های نرم نقض می‌شوند به ترتیب توسط متغیرهای V_{cd} و U_{dht}

($d = 1, \dots, D, h = 1, \dots, H, t = 1, \dots, T, c = 1, \dots, C$) اندازه‌گیری می‌شود و به ترتیب با

پارامترهای ثابت q, r مقدار جریمه شده را در تابع هدف وارد می‌کنیم. یعنی به ازای هر دفعه نقض شدن، q یا r واحد جریمه می‌شویم. تابع هدف، نشدنی بودن محدودیت‌های نرم و جریمه‌ی ساعات با ترجیح کمتر را مینیمم می‌کند. مدل بیان‌شده در این مقاله به‌صورت زیر است:

$$\min Z = \sum_{lcdh} X_{lcdh} + \sum_{lcdh} p_h X_{lcdh} + \sum_{dht} q U_{dht} + \sum_{cd} r V_{cd} \quad (2.2)$$

اولین محدودیت سخت در نظر گرفته شده در این مدل این است که تعداد جلسات تخصیص داده شده به هر درس باید برابر با تعداد جلسات مورد نیاز برای آن درس باشد. تعداد جلسات مورد نیاز برای درس c با cs_c نشان داده شده است. این محدودیت به این معنی است که برای هر درس c باید حاصل جمع متغیرهای تصمیم در همه‌ی روزها و بازه‌های زمانی و به ازای تمامی اساتید برابر تعداد جلسات مورد نیاز برای آن درس باشد.

$$\sum_{ldh} X_{lcdh} = cs_c \quad \forall c$$

محدودیت سخت دوم تضمین می‌کند که در هر ساعت، هر استاد بیش‌تر از یک درس تدریس نکند. در واقع باید حاصل جمع متغیرهای تصمیم به ازای تمام دروس در یک بازه‌ی زمانی برای هر استاد، حداکثر یک باشد.

$$\sum_c X_{lcdh} \leq 1 \quad \forall l, d, h$$

محدودیت سخت بعدی تضمین می‌کند که تعداد دروس اختصاص داده شده به یک بازه‌ی زمانی از تعداد کلاس‌های آن نوع بیش‌تر نشود. در این‌جا تعداد کلاس‌های از نوع rt با $Rooms(rt)$ نشان داده شده است. به عبارت دیگر، به ازای هر بازه‌ی زمانی در هر روز و به ازای هر نوع کلاس نباید حاصل جمع متغیرهای تصمیم بر روی تمام اساتید و دروسی که به آن نوع کلاس احتیاج دارند، از تعداد آن نوع کلاس بیشتر شود.

$$\sum_{lc, CRT(c)=RoomType(rt)} X_{lcdh} \leq Rooms(rt) \quad \forall h, d, rt$$

همه‌ی دروسی که در یک نیم‌سال یا ترم ارایه می‌شوند، ممکن است توسط دانشجویان به‌صورت هم‌زمان انتخاب شوند. بنابراین لازم است در روزها و زمان‌های مختلف برنامه‌ریزی شوند. نقض این قانون با U_{dht} در این محدودیت مطرح شده است. این یک محدودیت نرم است.

$$\sum_{lc} x_{lcdh} - U_{dht} \leq 1 \quad \forall d, h, t$$

جدول ۱.۲: مجموعه‌ها و شاخص‌های موجود در مدل‌بندی

مجموعه‌ها (شناسه‌ها)	شاخص‌ها	شرح
دروس	c	دروس موجود
اساتید	l	اساتید موجود
روزها	d	روزهای کاری
بازه‌های زمانی	h	زمان‌های مجاز موجود در یک روز
ترم‌ها	t	ترم‌ها یا نیم‌سال‌های تعریف شده
کلاس‌های درس	rt	انواع کلاس‌ها

با توجه به وجود U_{dht} در تابع هدف و به دلیل مینیمم‌سازی کردن تابع هدف، سعی می‌شود U_{dht} کمترین مقدار ممکن را اختیار کند. با توجه به این محدودیت با کم بودن مقدار U_{dht} ، مقدار X_{lcdh} نیز کاهش می‌یابد. بنابراین دروس در روزها و زمان‌های مختلفی برنامه‌ریزی می‌شوند.

در محدودیت نرم دوم سعی می‌شود که یک روز فاصله بین دو جلسه از یک درس وجود داشته باشد. تجاوز از این قانون با V_{cd} اندازه گرفته می‌شود. در این محدودیت نیز مانند محدودیت قبل، از آنجا که V_{cd} در تابع هدف وجود دارد، سعی می‌شود مقدار V_{cd} تا حد امکان کاهش پیدا کند. هر چه مقدار V_{cd} کمتر باشد، طبق این رابطه مقدار حاصل جمع متغیرهای تصمیم بر روی تمام بازه‌های زمانی و تمام اساتید به ازای هر درس در هر روز و روز بعدش نیز کاهش می‌یابد. بنابراین با رعایت این محدودیت بین دو جلسه از یک درس حداقل یک روز فاصله ایجاد می‌شود.

$$\sum_{lh} X_{lcdh} + \sum_{lh} X_{lc,d+1h} - V_{cd} \leq 1 \quad \forall c, d$$

در ادامه‌ی این مقاله توضیح داده شده است که فرموله کردن این مدل با استفاده از نرم‌افزار *AIMMS* بسیار ساده است. *AIMMS* یک سیستم مدلینگ است که به ما اجازه می‌دهد مدل‌های خطی پیچیده را که با ۱۰۰۰ متغیر و محدودیت درگیر است، در راهی مختصر و موثر تنظیم کنیم. برای استفاده از این نرم‌افزار متغیرهای مطرح شده می‌توانند گسسته یا پیوسته باشند. برای حل مسأله‌ی بالا با استفاده از این نرم‌افزار، با استفاده از مواردی که در ادامه شرح داده می‌شوند این کار صورت می‌گیرد.

این آزمایش همچنین روی یک مسأله‌ی جدول زمانبندی واقعی انجام شده است. مشخصات این مسأله در جدول ۲.۲ زیر نمایش داده شده است.

هدف از حل اینگونه مسائل این است که بتوان یک جدول زمانبندی بدون تداخل به دست آوریم. جدول خروجی نباید هیچگونه تداخلی داشته باشد. این مسأله توسط نرم‌افزار *AIMMS* که بر روی یک کامپیوتر *pentium 4* با $1/84\text{GHZ}$ سرعت و 256MB حافظه نصب شده، حل شده است. جدول زمانبندی بهینه

جدول ۲.۲: داده‌های مثال واقعی

تعریف	تعداد	نام پارامتر
تمام وقت و نیمه وقت	۲۵	اساتید
شامل دروس از قبل تخصیص داده شده	۷۵	دروس
-	۸	تعداد ترم‌ها
به‌جز جمعه بعد از ظهر	۷	تعداد روزهای کاری
شامل آزمایشگاه‌ها	۳	انواع کلاس‌ها
به‌جز جمعه‌ها که دو ساعت در روز است	۶	بازه‌های زمانی
۱۰۸۲ دودویی	۱۲۲۶	متغیرها
-	۹۴۴	محدودیت‌ها

با مقدار تابع هدف صفر) در کمتر از ۱۰ ثانیه به‌دست آمده است.

فصل ۳

یک مدل غیرخطی برای مساله‌ی جدول زمانبندی و حل آن با جستجوی ممنوع

در این فصل به بیان مقاله‌ی لویی و جان کائو [۱۶]، می‌پردازیم که در آن از یک مدل غیرخطی برای حل مساله‌ی جدول زمانبندی دروس استفاده شده است. در این مقاله روش فراابتکاری جستجوی ممنوع برای حل این مساله به کار گرفته شده است. بنابراین در ابتدای این فصل به بیان توضیح مختصری از روش جستجوی ممنوع می‌پردازیم و در ادامه نحوه‌ی مدل‌بندی و روش حلی که در این مقاله مورد استفاده قرار گرفته است را شرح می‌دهیم.

۱.۳ روش جست‌وجوی ممنوع

۱.۱.۳ مقدمه

روش جست‌وجوی ممنوع^۱ (TS) برای اولین بار توسط گلوور^۲ [۶]، در سال ۱۹۸۶ پیشنهاد شده است. واژه‌ی تابو از تنگان زبان مردم جزایر پلینزی در اقیانوس آرام گرفته شده است. این واژه به معنای شی مشخصی است که به دلیل قداست نباید آن را لمس کرد. بر اساس واژه‌نامه‌ی وبستر، امروزه این واژه در معنای ممنوعیت ایجاد شده به دلیل فرهنگ اجتماعی برای ایجاد اقدام حفاظتی یا ممنوعیت چیزی که دارای ریسک است، به کار می‌رود. معنای اخیر واژه‌ی تابو با تکنیک جست‌وجوی ممنوعه کاملاً سازگار است. ریسکی که در الگوریتم جست‌وجوی ممنوع از آن اجتناب می‌شود، خطر مسیرهای نامناسب است. در جست‌وجوی ممنوعه، تابوها توسط یک حافظه‌ی جمعی که در طول زمان تغییر می‌کند، ذخیره می‌شود و برای مدتی مورد انتخاب قرار نمی‌گیرد. به این صورت ارتباط اساسی میان تابو و جست‌وجوی ممنوع برقرار می‌شود. روش جست‌وجوی ممنوع به‌طور قابل ملاحظه‌ای توانایی ما را در حل مسائل علمی افزایش می‌دهد.

^۱Tabu Search

^۲Glover

کاربردهای کنونی این روش در زمینه‌های برنامه‌ریزی منابع، ارتباطات مخابراتی، تحلیل مالی، زمانبندی، توزیع انرژی، مهندسی مولکولی، مدیریت مواد زائد، اکتشاف مواد معدنی و بسیاری از موارد دیگر است. در سال‌های اخیر، مجله‌های مختلفی در طیف گسترده‌ای از زمینه‌ها مقاله‌ها و مطالعه‌هایی را چاپ می‌کنند که موفقیت روش جست‌وجوی ممنوع را در بسیاری از مسائل نشان می‌دهد.

در این روش عملکرد حافظه‌ی انسان شبیه سازی شده است. حافظه‌ی انسان با به کارگیری ساختمانی موثر و در عین حال ساده از اطلاعات آنچه را در قبل رویت شده ذخیره می‌کند. این مرکز هم‌چنین فهرستی از حرکات منع شده را تنظیم می‌کند و این فهرست همواره براساس آخرین جست‌وجوها منظم می‌شود. این روش از انجام هرگونه عملیات مجدد و تکراری جلوگیری می‌کند.

رویکردی که گلوور با نام جست‌وجوی ممنوع در سال ۱۹۸۶ ارائه کرد، در واقع به جست‌وجوی محلی اجازه می‌دهد تا بر بهینه‌ی محلی غلبه کند. اصول و مبانی این روش به جست‌وجوی محلی اجازه می‌دهد که در هر جایی که به بهینه‌ی محلی رسید با استفاده از حافظه‌ها که لیست ممنوعه نام دارند از بهینه‌ی محلی خارج و هم‌چنین از برگشت به آن‌ها جلوگیری شود.

هدف از این روش آن است که، بخش‌هایی از مجموعه‌ی جواب که پیش از این بررسی نشده است مدنظر قرار گیرد. بدین منظور حرکت به جواب‌هایی که اخیراً جست‌وجو شده، ممنوع خواهد بود.

ساختار کلی روش جست‌وجوی ممنوع به این صورت است که ابتدا یک جواب اولیه‌ی تصادفی انتخاب می‌شود؛ سپس برای جواب مربوط بر اساس یک معیار خاص مجموعه‌ای از جواب‌های همسایه‌ی امکان‌پذیر در نظر گرفته می‌شود. تعریف ساختار همسایگی به مقدار زیادی به تعریف فضای جست‌جو بستگی دارد. در هر گام از این روش یک انتقال محلی بر روی جواب فعلی x به کار می‌رود و جواب‌های همسایه‌ی $N(x)$ (همسایه‌ی x) را در فضای جست‌جو تولید می‌کند. در واقع $N(x)$ زیر مجموعه‌ای از فضای جست‌جو است که به صورت زیر تعریف می‌شود:

$$N(x) = \{ \}$$

در حالت کلی برای هر مسأله‌ی مشخص، هر چه تعداد تعریف‌های ممکن برای فضای جست‌جو بیشتر باشد، ساختارهای همسایگی بیشتری وجود خواهد داشت. در واقع ممکن است ساختارهای همسایگی مختلفی برای یک تعریف از فضای جواب مفروض وجود داشته باشد. در گام بعد، پس از ارزیابی جواب‌های همسایه‌ی تعیین شده، بهترین آن‌ها انتخاب می‌شود و جابجایی از جواب جاری به جواب همسایه‌ی انتخابی صورت می‌گیرد. این فرآیند به همین ترتیب تکرار می‌شود تا زمانی که شرط خاتمه تحقق یابد.

در روش جست‌وجوی ممنوع فهرستی وجود دارد که جابجایی‌های منع شده را نگهداری می‌کند و به فهرست ممنوعه معروف است. کاربرد اصلی این لیست، پرهیز از همگرا شدن به جواب‌های بهینه‌ی محلی است. به عبارت دیگر به کمک فهرست ممنوعه، جابجایی به جواب‌هایی که اخیراً جست‌وجو شده‌اند ممنوع خواهد

شد. با این کار بخش‌هایی از مجموعه‌ی جواب که پیش از این مورد بررسی قرار نگرفته‌اند، مدنظر خواهد بود. در روش جست‌وجوی ممنوع بعد از هر جابجایی، فهرست ممنوعه به‌روز می‌شود به نحوی که جابجایی جدید به آن فهرست اضافه می‌شود. در واقع یک TS ساده را می‌توان ترکیبی از یک حافظه‌ی کوتاه‌مدت با جست‌وجوی موضعی (LS) دانست.

مفهوم اساسی در TS ، مجاز دانستن جواب‌هایی است که در تابع هدف بهبود ایجاد نمی‌کنند، اما ممکن است ما را به سمت جواب سراسری راهنمایی کنند، با این شرط که در لیست جواب‌های ممنوع قرار نداشته باشند.

معیار توقف

به طور نظری، جست‌جو می‌تواند برای همیشه ادامه پیدا کند، مگر اینکه مقدار بهینه‌ی مساله از قبل مشخص شده باشد. در عمل، به طور واضح و مشخص جست‌جو باید در نقطه‌ای متوقف شود. رایج‌ترین معیارهای توقفی که در TS مورد استفاده قرار می‌گیرند، به صورت زیر است:

- پس از تعداد معینی تکرار
 - پس از تعداد تکرارهای معینی که هیچ بهبودی در تابع هدف ایجاد نشود (معیار توقفی که در اکثر پیاده‌سازی‌ها استفاده می‌شود).
 - هنگامی که تابع هدف به مقدار آستانه‌ی از پیش تعیین شده برسد.
- در روش‌های پیچیده‌تر جست‌جوی ممنوع، جست‌جو معمولاً پس از اتمام چندین مرحله متوقف می‌شود، که هر مرحله بر اساس یکی از مراحل بالا متوقف می‌شود.

۲.۱.۳ قالبی برای جست‌جوی ممنوع ساده

هم اکنون که به سطح قابل قبولی از آشنایی با مفاهیم رسیدیم، می‌توانیم قالبی برای TS ارائه دهیم. فرض کنید به دنبال کمینه‌سازی تابع $f(x)$ هستیم. مفاهیم اصلی روش جست‌وجوی ممنوع را می‌توان به صورت زیر بیان کرد؛

علائم و نمادها

- x : جواب فعلی
- x^* : بهترین جواب یافت شده
- f : مقدار تابع هدف برای x

• f^* : مقدار تابع هدف برای x^*

• $N(x)$: همسایگی x

• T : لیست ممنوعه

آماده‌سازی

تولید جواب اولیه x_0 .

قرار دادن $x = x_0, f^* = f(x), x^* = x_0, T = \phi$

جستجو

تا زمانی که معیار توقف دیده نشده است، مراحل زیر را انجام بده:

• انتخاب x از میان مینیمم $f(x')$ که $x' \in N(x)$

• اگر $f(x) < f^*$ ، آنگاه قرار بده $f^* = f(x), x^* = x$.

• ذخیره‌سازی تابو برای حرکت فعلی در T (حذف قدیمی‌ترین ورودی در صورت لزوم).

شکل ۱.۳ نمودار جریان روند معرفی شده در بالا را به صورت گرافیکی نمایش می‌دهد.

دو راهکار برای بهبود روش جستجوی ممنوع

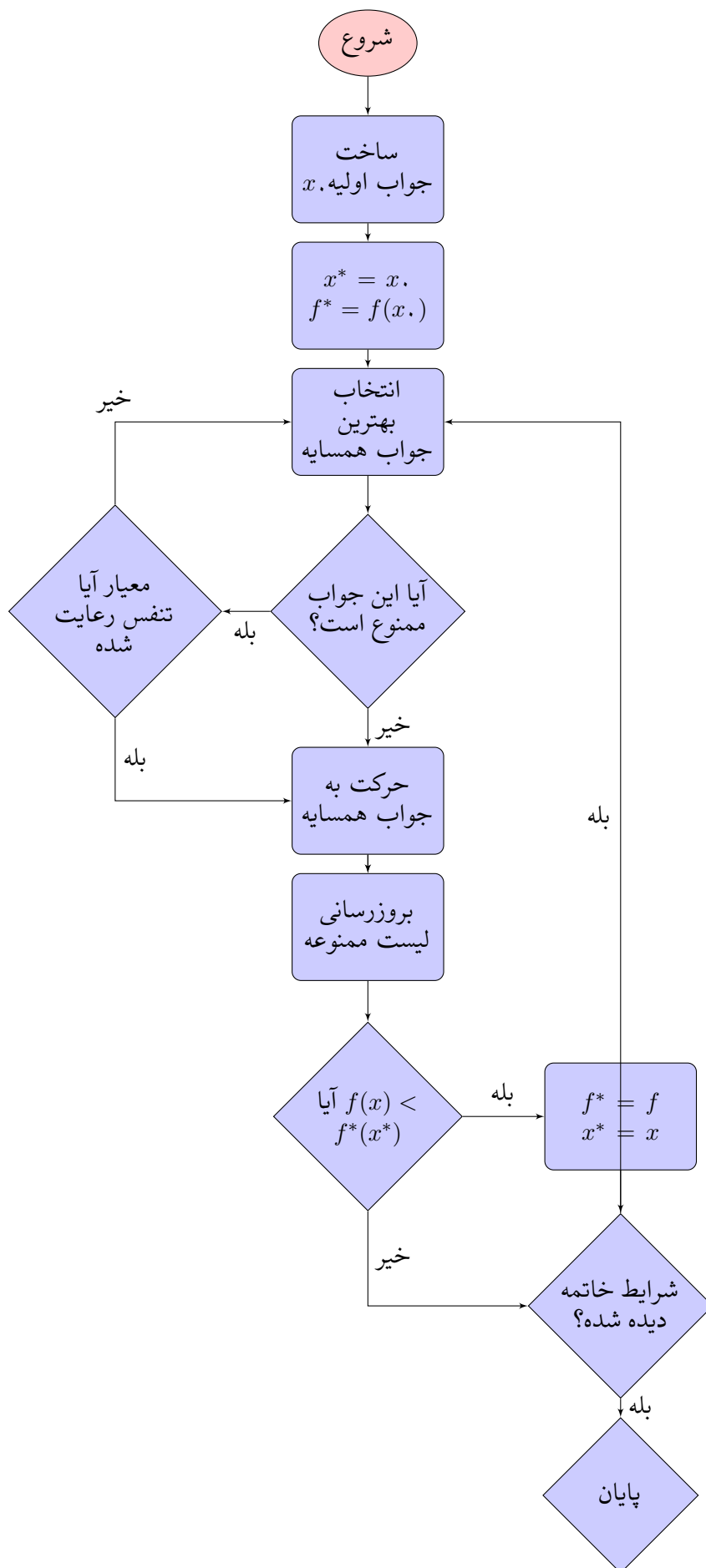
استفاده از دو ابتکار تقویت^۳ و تنوع^۴ در روش جست‌وجوی ممنوع، باعث می‌شود محدوده‌ی گسترده‌تری از فضای جواب جست‌وجو شود.

تقویت: تقویت در روش جست‌وجوی ممنوع به مکانیسمی اطلاق می‌شود که براساس آن جواب‌ها و حرکت‌هایی که باعث رسیدن به جواب‌های خوب شده‌اند، تقویت می‌شوند. به عبارت بهتر این استراتژی، به بازگشت به جواب‌های نخبه و جست‌وجوی بیشتر در محدوده‌ی آن‌ها، اشاره دارد.

تنوع: یکی از مهم‌ترین مشکلات روش‌های مبتنی بر جست‌وجوی موضعی، گرایش آن‌ها به محلی بودن است. این روش‌ها تمایل دارند تا بیش‌تر در یک منطقه‌ی محدود به جست‌وجو بپردازند. در واقع در این روش‌ها گرچه ممکن است جواب خوبی پیدا شود، اما ممکن است تمام فضای جواب، جست‌وجو نشود و جواب به دست آمده بهینه‌ی سراسری نباشد. تنوع در روش جست‌وجوی ممنوع، مکانیسمی است که تلاش می‌کند این مشکل را با انتقال جست‌وجو به ناحیه‌های کمتر مورد بررسی قرار گرفته و یا قرار نگرفته حل کند. دو روش عمده برای اعمال مکانیسم تنوع وجود دارد:

^۳ Intensification

^۴ Diversification



۱- شروع دوباره: جست‌وجو را از نقاطی که کمتر مورد استفاده قرار گرفته‌اند و یا از جواب بهینه‌ای که پیدا شده است، مجدداً شروع می‌کنیم.

۲- هدایت جست‌وجو: قرار دادن مکانیسمی در فرآیند جست‌وجو که کار رعایت پراکندگی را از همان ابتدا در دستور کار قرار دهد.

شکل ۲.۳ نشان‌دهنده‌ی نمودار روش TS بهبود یافته است.

۲.۳ تشریح مدل

جدول زمانبندی که لویی و جین کایو در مقاله‌ی خود در نظر گرفته‌اند، براساس برنامه‌ی درسی تنظیم شده است. جدول زمانبندی دروس دانشگاه بر اساس برنامه‌ی درسی، مجموعه‌ای از جلسات هستند که باید به تعدادی بازه‌ی زمانی و کلاس که تابع یک گروه از محدودیت‌هاست، اختصاص داده شوند. هدف از این مساله، به حداقل رساندن نقض محدودیت‌های نرم در جدول زمانبندی شدنی است.

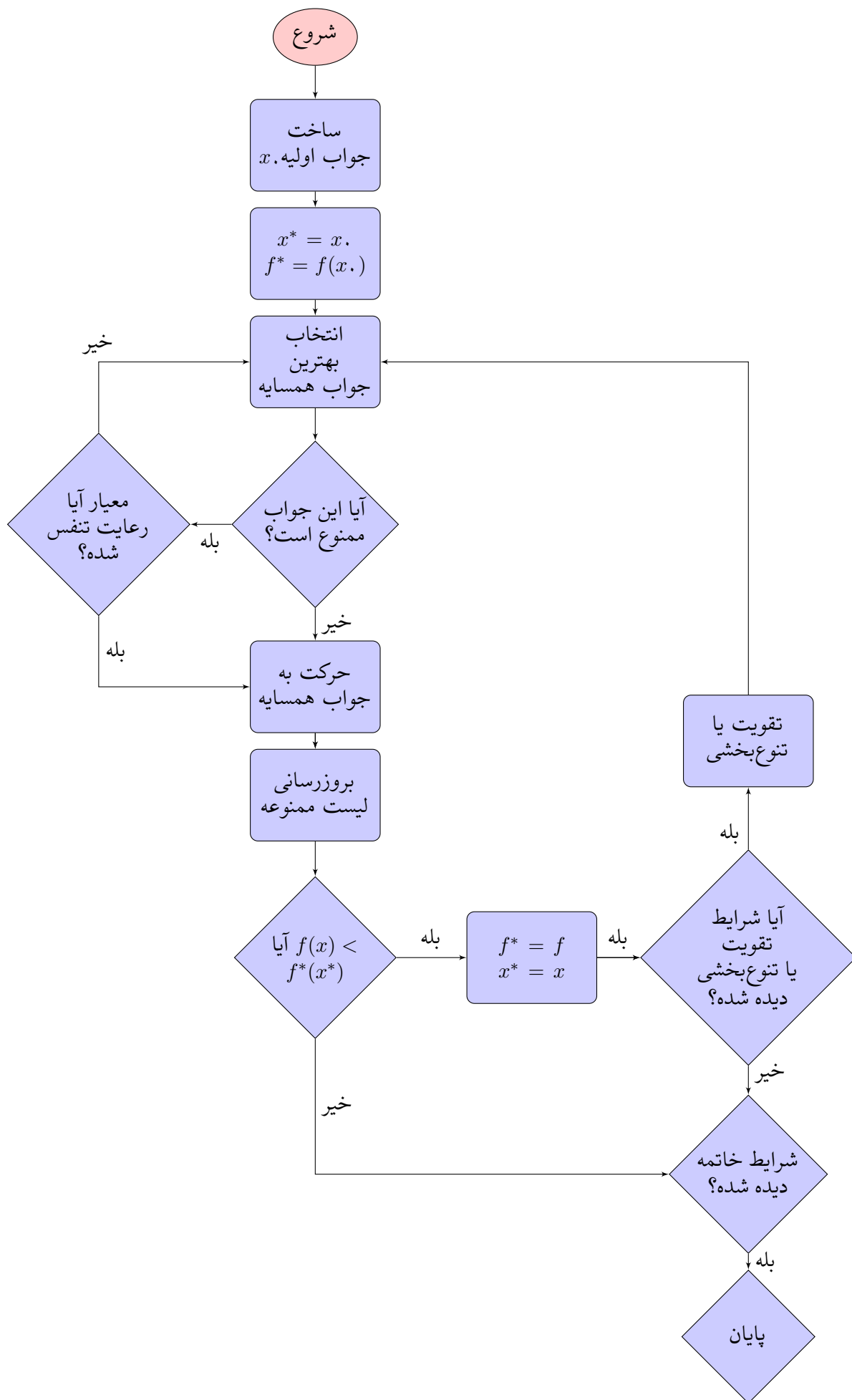
۱.۲.۳ تعیین محدودیت‌ها

محدودیت‌های سختی که در این مساله در نظر گرفته شده‌اند به این ترتیب است:

۱. هر جلسه از یک درس باید در یک زمان متمایز و یک کلاس متمایز زمانبندی شوند.
۲. دو جلسه نمی‌توانند در یک زمان به یک کلاس اختصاص داده شوند.
۳. جلسات دروس در برنامه‌ی درسی واحد یا تدریس شده توسط استاد واحد نمی‌تواند در زمان واحد زمانبندی شوند.
۴. اگر استادی در یک زمان خاص در دسترس نباشد، هیچ جلسه‌ای نباید در آن زمان به این استاد تخصیص داده شود.

محدودیت‌های نرم به‌صورت زیر است:

۵. برای هر جلسه تعداد دانشجویان هر درس نباید بیش‌تر از ظرفیت آن کلاس باشد.
۶. همه‌ی جلسات از یک درس باید در کلاس واحد زمانبندی شود. اگر این کار غیرممکن است باید تعداد کلاس‌های اشغالی به حداقل برسد.
۷. جلسات از یک درس باید در کمترین تعداد روزهای کاری گسترده شوند.
۸. برنامه‌ی درسی دانشجویان باید تا حد امکان فشرده باشد.



۲.۲.۳ مدل‌بندی

مسالهی جدول زمانبندی بر اساس برنامه‌ی درسی، متشکل از n درس $C = \{c_1, \dots, c_n\}$ است که باید در یک گروه p عضوی از بازه‌های زمانی $T = \{t_1, \dots, t_p\}$ و در یک گروه m عضوی از کلاس‌های $R = \{r_1, \dots, r_m\}$ زمانبندی شوند. در این مساله هر درس c_i به تعداد l_i ساعت درسی نیاز دارد. سایر نمادهای مورد نیاز برای مدل‌بندی این مساله، در جدول ۳.۳ نشان داده شده است.

یک جواب برای این مساله با یک ماتریس $p \times m$ نشان داده می‌شود که در آن x_{ij} نشان‌دهنده‌ی برچسب درس اختصاص داده شده در زمان t_i و کلاس r_j است. اگر هیچ درسی به بازه‌ی t_i و r_j اختصاص داده نشود x_{ij} مقدار ۱- می‌پذیرد. این به این معنی است که در ابتدا تمام درایه‌های این ماتریس مقدار ۱- دارند. در ادامه با تخصیص دروس به بازه‌های زمانی و کلاس‌ها، برچسب درس مربوطه جایگزین مقدار ۱- می‌شود. با استفاده از این کار ما تضمین می‌کنیم بیش‌تر از یک درس در یک بازه‌ی زمانی به یک کلاس اختصاص داده نشود. این به این معنی است که محدودیت سخت دوم همواره رعایت می‌شود. تابع هدفی که برای این مدل‌بندی در نظر گرفته شده است، به گونه‌ای است که در آن سعی می‌شود نقض محدودیت‌های نرم مینیمم شود. هر یک از توابع f_1, f_2, f_3, f_4 توابعی هستند که برای محدودیت‌های نرم ۱ تا ۴ در نظر گرفته شده است. مقادیر $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ هم جریمه‌هایی هستند که برای نقض هر کدام از محدودیت‌های نرم در نظر گرفته شده است. در این مقاله این جریمه‌ها به صورت زیر در نظر گرفته شده‌اند:

$$\alpha_1 = 1, \quad \alpha_2 = 2, \quad \alpha_3 = 5, \quad \alpha_4 = 2$$

این مقداردهی نشان‌دهنده‌ی آن است که در این مساله رعایت شدن محدودیت نرم سوم، نسبت به سایر محدودیت‌های نرم در اولویت است. با توجه به مطالب فوق مدل مساله به صورت زیر است:

$$\min f(x) = \sum_{x_{i,j} \in X} \alpha_1 \cdot f_1(x_{i,j}) + \sum_{c_i \in C} \alpha_2 \cdot f_2(c_i) + \sum_{c_i \in C} \alpha_3 \cdot f_3(c_i) + \sum_{x_{i,j} \in X} \alpha_4 \cdot f_4(x_{i,j}) \quad (1.3)$$

s.t

$$(1) \quad \forall c_k \in C \quad \sum_{i=1, \dots, p, j=1, \dots, m} \chi\{x_{ij} = c_k\} = l_k$$

$$(3) \quad \forall x_{i,j}, x_{i,k} \in X, \quad x_{i,j} = c_u, \quad x_{i,k} = c_v, \quad con_{uv} = \bullet$$

جدول ۳.۳: نمادهای مورد استفاده در مدل‌بندی

نمادها	توضیح
n	تعداد دروس
m	تعداد کلاس‌ها
d	تعداد روزهای هفته
h	تعداد بازه‌های زمانی هر روز کاری
p	تعداد دوره‌ها $p = h \times d$
s	تعداد برنامه‌های درسی
C	مجموعه‌ی دروس $C = \{c_1, \dots, c_n\}$
R	مجموعه‌ی کلاس‌ها $R = \{r_1, \dots, r_m\}$
T	مجموعه‌ی بازه‌های زمانی $T = \{t_1, \dots, t_p\}$
CR	مجموعه‌ی برنامه‌های درسی $CR = \{cr_1, \dots, cr_s\}$
Cr_k	k امین برنامه‌ی درسی شامل یک مجموعه از دروس
l_i	تعداد جلسات مورد نیاز درس c_i
l	تعداد کل جلسات $L = \sum_{i=1}^n l_i$
std_i	تعداد دانشجویان خواهان درس i
tc_i	اساتیدی که درس i را می‌آموزند

نمادها	توضیح
md_i	مینیمم تعداد روزهای کاری درس i
cap_j	ظرفیت کلاس r_j
uav_{ij}	اگر درس c_i در زمان t_j در دسترس باشد $uav_{ij} = 0$ در غیر این صورت $uav_{ij} = 1$
con_{ij}	اگر درس c_i و c_j با هم ناسازگار هستند؛ $con_{ij} = \begin{cases} 0 & if(tc_i \neq tc_j) \wedge (\forall cr_q, c_i \in cr_q \vee c_j \in cr_q) \\ 1 & o.w \end{cases}$
x_{ij}	درس تخصیص داده شده در زمان t_i در کلاس t_j
$nr_i(X)$	تعداد کلاس‌های اشغال شده با درس c_i برای راه‌حل کاندید X $nr_i(X) = \sum_{j=1}^m \sigma_{ij}(X)$ $\sigma_{ij} = \begin{cases} 1 & if \forall x_{kj} \in X, \quad x_{kj} = c_i \\ 0 & o.w \end{cases}$
$nd_i(X)$	تعداد روزهای کاری که درس c_i در راه‌حل کاندید X می‌گیرد $nd_i(X) = \sum_{j=1}^d \beta_{ij}(X)$ $\beta_{ij} = \begin{cases} 1 & if \forall x_{uv} \in X, \quad x_{uv} = c_i \wedge [u/h] = j \\ 0 & o.w \end{cases}$
$app_{k,i}(X)$	$app_{k,i}(x) = \begin{cases} 1 & if \forall x_{ij} \in X, \quad x_{ij} = c_u \quad c_u \in cr_k \\ 0 & o.w \end{cases}$

$$(۴) \quad \forall x_{i,j} = c_k \in X \quad uav_{i,j} = \bullet$$

$$(۵) \quad \forall x_{i,j} = c_k \in X, \quad f_1(x_{i,j}) = \begin{cases} std_k - cap_j & if\ std_k > cap_j \\ 0.w & o.w \end{cases}$$

$$(۶) \quad \forall c_i \in C \quad f_2(c_i) = nr_i(x) - 1$$

$$(۷) \quad \forall c_i \in C \quad f_3(x_{c_i}) = \begin{cases} md_i - nd_i(x), & if\ nd_i(x) < md_i \\ 0.w & o.w \end{cases}$$

$$(۸) \quad \forall x_{i,j} = c_k \in X, \quad f_4(x_{i,j}) = \sum_{cr_q \in CR} \chi\{c_k \in cr_q\}.iso_{q,i}$$

که در آن

$$iso_{q,i}(x) = \begin{cases} 1 & if\ (i \bmod h = 1 \vee app_{q,i-1}(x) = \bullet) \wedge (i \bmod h = \bullet \vee app_{q,q+1}(x) = \bullet) \\ 0.w & o.w \end{cases}$$

در محدودیت اول χ تابع مشخصه است. این تابع ارزش یک می گیرد اگر گزاره ی داده شده درست باشد و در غیر این صورت مقدار صفر می گیرد. با در نظر گرفتن این تابع تضمین می شود که به هر درس به تعداد جلسات مورد نیاز آن درس، بازه ی درس و کلاس تخصیص داده شود. همانطور که گفته شد با اجرای روش پیشنهادی محدودیت دوم همیشه رعایت می شود.

محدودیت سوم بیان می کند که اگر دو درس به یک بازه ی یکسان تخصیص داده شوند، نباید با هم ناسازگاری داشته باشند. ناسازگاری به این معنی است که نباید دارای استاد مشترک باشند و هر دو متعلق به یک برنامه ی درسی باشند (با توجه به تعریف تابع con_{ij} در جدول ۳.۳).

با توجه به تعریف تابع usv_{ij} واضح است که محدودیت چهارم، در دسترس بودن اساتید را تضمین می کند. محدودیت نرم پنج ایجاب می کند چنانچه تعداد دانشجویان یک درس از ظرفیت کلاس تخصیص داده شده به آن درس بیشتر باشد، تابع f_1 مقداری برابر تفاضل تعداد دانشجویان با ظرفیت کلاس خواهد گرفت و در تابع هدف جریمه خواهد شد. اما اگر ظرفیت کلاس از تعداد دانشجویان بیشتر نشود، f_1 مقدار صفر خواهد گرفت و در نتیجه در تابع هدف جریمه نخواهد شد. در محدودیت شش چنانچه تعداد کلاس های اشغال

شده توسط یک درس بیشتر از یک کلاس باشد، تابع f_2 به تعداد کلاس‌های بیشتر از یکی مقدار می‌گیرد. هر چه مقدار f_2 (تعداد کلاس‌های اشغالی بیشتر از یکی) بیشتر شود در نتیجه مقدار تابع هدف هم بیشتر می‌شود.

با رعایت محدودیت هفتم اگر تعداد روزهای اختصاص داده شده به یک درس از مینیمم روزهای کاری تعیین شده برای آن درس بیشتر باشد، تابع f_3 مقداری برابر تفاضل این دو مقدار می‌گیرد و در غیر این صورت مقدار صفر خواهد گرفت.

محدودیت هشتم به ازای هر درس c_k که در برنامه‌ی درسی cr_q قرار دارد، و در جواب x به بازه‌ی زمانی i (به ازای $i = 1, \dots, p$) تخصیص داده شده است، بررسی می‌شود. چنانچه هیچ درس دیگری از برنامه‌ی درسی cr_q در بازه‌ی زمانی قبل و بعد i در یک روز برگزار نشود، تابع یک واحد به جریمه‌ی این جواب اضافه می‌شود.

برای توضیح تابع $iso_{q,i}$ برای مثال فرض کنیم ما در هر روز ۴ بازه‌ی زمانی داریم و به‌طور کلی ۳ روز کاری در هفته داریم. بنابراین $h = 4$ و $d = 3$. حال چنانچه درس c_k به یک بازه‌ی زمانی از روز دوم تخصیص داده شود، باید بررسی شود درس دیگری از این برنامه‌ی درسی به بازه‌های قبل یا بعد از این بازه‌ی زمانی در یک روز اختصاص داده شده یا خیر. بازه‌های زمانی ۵، ۶، ۷، ۸ در روز دوم کاری قرار گرفته‌اند. اگر درس c_k به بازه‌ی زمانی $i = 5$ تخصیص داده شده باشد، بنابراین داریم $i \bmod h = 1$. بنابر رابطه‌ی تابع $iso_{q,i}$ حال باید بررسی شود که در بازه‌ی زمانی $i + 1$ ، درسی درسی از این برنامه‌ی درسی برگزار می‌شود یا نه. این کار توسط تابع $app_{k,i}$ انجام می‌شود. چنانچه درسی به این بازه تخصیص داده نشده بود یک واحد به مقدار f_4 و در واقع یک واحد به میزان جریمه‌ی این جواب افزوده می‌شود. اگر $i = 6$ یا $i = 7$ آن‌گاه $i \bmod h \neq 0$ و $i \bmod h \neq 1$. بنابراین باید دو بازه‌ی زمانی $i + 1$ و $i - 1$ مورد بررسی قرار گیرند. اگر $i = 8$ آن‌گاه $i \bmod h = 0$. در نتیجه فقط بازه‌ی زمانی $i - 1$ برای این کار بررسی می‌شود.

۳.۲.۳ روش حل

الگوریتم پیشنهادی در این مقاله برای حل این مدل، دارای سه چارچوب کلی است:

- جواب اولیه
- تشدید
- تنوع

مرحله‌ی مقداردهی اولیه براساس یک جدول زمانبندی شدنی با استفاده از یک روش حریمانه^۵ به دست می‌آید. سپس مراحل تشدید و تنوع، برای کاهش تعداد نقض محدودیت‌های نرم به کار گرفته می‌شود.

^۵Greedy

جواب اولیه

همانطور که گفته شد، در این مقاله برای یافتن یک جواب اولیه‌ی شدنی از روش حریصانه استفاده شده است. هدف این مرحله از الگوریتم تولید یک جواب اولیه است به‌طوری که تمام محدودیت‌های سخت در آن رعایت شوند. ایده‌ی اصلی روش حریصانه این است که از میان تمامی جواب‌ها، اولین جوابی که شدنی است به عنوان جواب اولیه در نظر گرفته می‌شود. در این مساله، کار با یک جدول زمانبندی خالی آغاز می‌شود و با گذاشتن جلسات مناسب در جدول زمانبندی دروس تخصیص داده می‌شوند. در هر مرحله دو عملیات مجزا اجرا می‌شود. ابتدا انتخاب جلسه‌ی اختصاص داده نشده سپس یافتن یک جفت کلاس و بازه‌ی زمانی برای این جلسه. در ابتکار انتخاب جلسه، درس‌هایی که در بازه‌های زمانی کمتری در دسترس هستند و همچنین تعداد زیاد جلسات اختصاص داده نشده، در اولویت هستند.

حل مساله به وسیله‌ی الگوریتم جست‌وجوی ممنوع

الگوریتم جست‌وجوی ممنوع تطبیقی (ATS^6)، با استفاده از دو ساختار همسایگی به کار انداخته می‌شود. به‌طور دقیق‌تر این روش با یکی از همسایگی‌ها آغاز می‌شود. هنگامی که جست‌وجو تمام شد، با بهترین بهینه‌ی محلی حاصل از این همسایگی، مجدداً الگوریتم راه‌اندازی می‌شود با این تفاوت که این بار از همسایگی دوم استفاده می‌شود.

ساختار همسایگی

● **همسایگی N_1 :** همسایگی N_1 از همه‌ی حرکت‌های شدنی توسط مبادله‌ی ساده γ تشکیل شده است. این حرکت شامل تبادل جفت زمان و کلاس تخصیص یافته به دو جلسه از دروس مختلف است. برای دو درس مختلف $x_{i,j} = \alpha$ و $x_{i',j'} = \alpha'$ روش مبادله‌ی ساده شامل تخصیص ارزش $x_{i,j}$ به $x_{i',j'}$ و برعکس است.

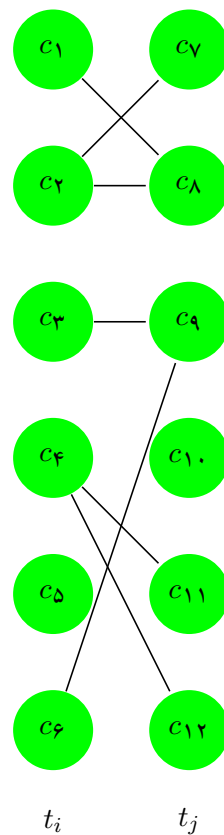
● **همسایگی N_2 :** همسایگی N_2 شامل همه‌ی حرکت‌های شدنی مبادله‌ی گروهی α است. اگر تنها روی درس‌ها و ناسازگاری‌ها (تداخل‌ها) تمرکز کنیم، هر نمونه مساله را می‌توان همانند یک گراف G در نظر گرفت که در آن گره‌ها دروس هستند و یال‌های اتصال دروس، اساتید و دانشجویان مشترک هستند. فرآیند مبادله‌ی گروهی، یک تخصیص جدید شدنی توسط مبادله‌ی برچسب‌های تخصیص یافته به دروس وابسته به یک یا دو گروه مشخص، ایجاد می‌کند. برای مثال به شکل ۱.۳ توجه کنید:

با توجه به این شکل، درس c_1 با درس c_8 دارای استاد یا دانشجوی مشترک هستند. این به این معناست که برای مثال اگر بخواهیم درس c_1 را از زمان t_i به زمان t_j منتقل کنیم باید در نظر داشته باشیم که

⁶ Adaptive Tabu Search

⁷ Simple Swap

⁸ Kempe Swap



شکل ۱.۳: تخصیص دروس به دو بازه‌ی زمانی t_i و t_j

چون درس c_1 با درس c_8 دارای استاد یا درس مشترک هستند، نمی‌توانند باهم در یک بازه‌ی زمانی قرار بگیرند. به همین ترتیب درس c_8 با c_2 و درس c_2 با c_7 ، استاد یا درس مشترک دارند. بنابراین دروس c_1, c_2, c_7, c_8 با هم در یک گروه قرار می‌گیرند. با همین استنباط گروه‌های دیگر به ترتیب زیر هستند:

$$k_a = \{c_1, c_2, c_7, c_8\} \quad k_b = \{c_3, c_6, c_9\} \quad k_c = \{c_4, c_{11}, c_{12}\} \quad k_d = \{c_5\} \quad k_e = \{c_{10}\}$$

در این روش می‌توانیم یک یا چند مجموعه را انتخاب کنیم و بازه‌های زمانی اعضای آن‌ها را با هم، تعویض کنیم. باید دقت داشت که نباید مبادله‌ای انجام داد که شدنی بودن را از بین ببرد. در این مثال می‌توان با مبادله‌ی دو گروه k_c و k_b (شکل ۲.۳)، یک جواب جدید شدنی به‌دست آورد. یکی دیگر از جواب‌های شدنی در این مثال مبادله‌ی دروس در گروه k_a است. باید در نظر داشت که بعد از انجام مبادله، باید بررسی شود که تعداد دروس تخصیص داده شده به یک بازه‌های زمانی از تعداد کلاس‌ها بیشتر نباشد. مثلاً در این مثال نمی‌توان گروه k_b را به‌تنهایی مبادله کرد، زیرا در این صورت تعداد دروس تخصیص داده شده از تعداد

کلاس‌ها در یک بازه‌ی زمانی بیشتر می‌شود.

ارزیابی همسایگی‌ها

از آنجایی که برای بررسی شدنی بودن همسایگی N_1 تنها به محاسبه‌ی هزینه‌ی مربوط به زمان (Δf_p) نیاز داریم ولی در همسایگی N_2 باید علاوه بر هزینه‌ی مربوط به زمان، هزینه‌ی مربوط به کلاس (Δf_r) نیز محاسبه شود. برای محاسبه‌ی هزینه‌ی مربوط به کلاس از الگوریتم تطبیق استفاده می‌شود. این الگوریتم بررسی می‌کند که تعداد دروس تخصیص داده شده از تعداد کلاس‌های موجود بیشتر نباشد. با توجه به هزینه‌ی بالای استفاده از همسایگی دوم، استفاده از این همسایگی برای همه‌ی حرکت‌ها مقرون به صرفه نمی‌باشد. لذا با قرار دادن شرطی، تنها برای تعدادی از حرکت‌های این همسایگی هزینه‌ی مربوط به کلاس را محاسبه می‌کنیم. در حقیقت از هزینه‌ی مربوط به زمان برای بررسی مقرون به صرفه بودن حرکت استفاده می‌شود. به این ترتیب که اگر $\Delta f_p < \tau$ ، سپس الگوریتم تطبیق برای تخصیص کلاس‌ها و به دست آوردن هزینه‌ی مربوط به کلاس، به کار گرفته می‌شود. τ در این جا مقداری است که با توجه به شرایط مساله تعیین می‌شود (در این مقاله $\tau = 2$ در نظر گرفته شده است).

شرط توقف

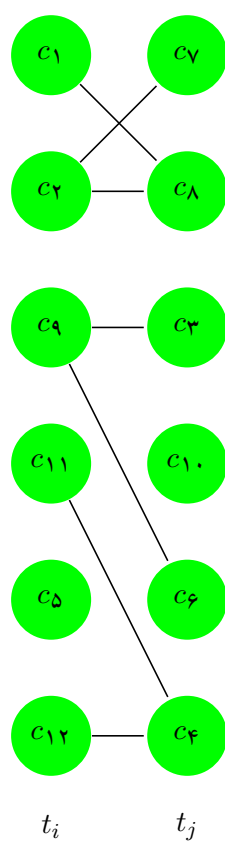
روش جست‌وجوی ممنوع در این مقاله زمانی متوقف می‌شود که یک جواب نتواند در تعداد θ داده شده از حرکت‌ها که آن را عمق روش جست‌وجوی ممنوع می‌دانیم، بهبود یابد.

مدیریت لیست ممنوعه

همان‌طور که گفته شد در روش جست‌وجوی ممنوع، یک لیست ممنوعه وجود دارد که مانع بازگشت به راه‌حلی می‌شود که قبلاً مشاهده شده است. در این مقاله، زمانی که یک جلسه از یک موقعیت (زوج کلاس و زمان) به موقعیت دیگر (توسط همسایگی N_1) منتقل می‌شود و یا از یک زمان به زمان دیگر (توسط همسایگی N_2) می‌رود، در چند تکرار بعدی نمی‌تواند به موقعیت قبلی خود بازگردد. برای مثال در همسایگی N_1 اگر یک جلسه از درس c_i از موقعیت (t_j, r_k) منتقل شود، حرکت هر جلسه از درس c_i به موقعیت (t_j, r_k) ممنوع خواهد بود. از سوی دیگر در همسایگی N_2 ، اگر یک جلسه از درس c_i ، از زمان t_j به زمان t_k منتقل شود، در چند تکرار بعدی که توسط تصرف ممنوع درس c_i مشخص می‌شود نباید هر جلسه‌ی c_i به t_j اختصاص داده شود.

تصرف ممنوع درس c_i ($tt(c_i)$) براساس مقدار تابع هدف جواب فعلی و میزان ناسازگاری درس c_i ، تنظیم می‌شود که به صورت زیر نمایش داده می‌شود:

$$tt(c_i) = f + \phi \quad (2.3)$$



شکل ۲.۳: نحوه‌ی تخصیص دروس به بازه‌های زمانی t_i و t_j بعد از اعمال همسایگی N_2

بخش اول این تابع را به این صورت می‌توان توضیح داد که یک جواب با جریمه‌ی بالا برای محدودیت‌های نرم، برای فرار از بهینه‌ی محلی باید دارای مقدار تصرف ممنوع بالاتری باشد. بنابراین با توجه به رابطه‌ی ۲.۳ اگر مقدار تابع هدف یک جواب زیاد باشد، به همان مقدار تصرف ممنوع هم بالا می‌رود. در این رابطه ϕ یک مقدار بین $[0, 1]$ می‌گیرد. بخش دوم تابع ۲.۳، به منظور جریمه کردن یک حرکت است که بیش از پیش تکرار می‌شود. در این فرمول ϕ برابر است با نسبت دروس ناسازگار با درس c_i به کل دروس. این منطقی به نظر می‌رسد که انتقال درسی که ناسازگاری بیشتری دارد نسبت به درسی که ناسازگاری کمتری دارد، ریسک بیشتری دارد.

جست‌وجوی ممنوع تطبیقی

روش جست‌وجوی ممنوع می‌تواند با زمان CPU کوتاه و بلند مورد استفاده قرار گیرد. به طور کلی، زمان CPU بالا موجب دست‌یافتن به نتایج بهتر می‌شود. با این حال اگر زمان محاسبه‌ی کلی محدود باشد، برای بهبود یافتن جواب در زمان کم روش جست‌وجوی ممنوع با عمق کم را با برخی اپراتورهای تنوع قوی ترکیب می‌کنیم. اپراتورهای تنوع ما را در فرار از بهینه‌ی محلی و حرکت به سوی مناطق جدید امیدوارکننده در فضای جست‌وجو راهنمایی می‌کنند.

در این مقاله وقتی که یک جواب نمی‌تواند با استفاده‌ی بیشتر از الگوریتم جست‌وجوی ممنوع بهبود یابد، از یک عملگر اغتشاش برای بازسازی راه‌حل بهینه‌ی به دست آمده تاکنون استفاده می‌شود. به طور کلی اگر اغتشاش خیلی قوی باشد ممکن است همانند یک راه‌اندازی مجدد تصادفی عمل کند. از سویی دیگر اگر اغتشاش خیلی کوچک باشد، جست‌وجو در یک منطقه‌ی کوچک محدود خواهد شد و در یک بهینه‌ی محلی سقوط می‌کند.

در این مقاله به منظور استفاده از عملگر اغتشاش، q جلسه‌ای که دارای جریمه‌های بالایی هستند (تعداد محدودیت‌های نرم بیشتری را نقض می‌کنند) انتخاب می‌شوند (در این مقاله به طور تجربی $q = 30$ در نظر گرفته شده است). سپس با انتخاب یک عدد تصادفی از میان همسایگی‌ها، این حرکت را روی حداقل یکی از این جلسات انجام می‌دهیم. تعداد کل حرکت‌های اغتشاش، قدرت اغتشاش نامیده می‌شود و با η نمایش داده می‌شود.

به عبارت دیگر هنگامی که یک فاز از روش جست‌وجوی ممنوع خاتمه یافت، در جواب به دست آمده از این روش، q جلسه‌ای که بیشترین جریمه‌ها را دارند، پیدا می‌شود. سپس η جلسه از میان این q جلسه انتخاب می‌شود. در این انتخاب جلسه‌ی k براساس توزیع احتمال زیر به دست می‌آید:

$$p(k) \propto k^{-Q} \quad (3.3)$$

که در فرمول ۳.۳، Q یک عدد حقیقی مثبت است (در این مقاله به طور تجربی این مقدار برابر ۴ در نظر گرفته شده است). پس از آن η حرکت شدنی از حرکت‌های مبادله‌ای ساده یا مبادله‌ای گروهی به صورت تصادفی و پی‌درپی انتخاب می‌شود به طوری که هر یک شامل حداقل یکی از η جلسه‌ی انتخاب شده باشند.

جدول ۴.۳: الگوریتم جست و جوی ممنوع تطبیقی

```

1: Input:  $I$ : an instance of CB-CTT
2: Output:  $X^*$ : the best solution found so far
3: % Initialization: lines 6–8
4: % Intensification: lines 11–17
5: % Diversification: lines 10, 23
6:  $X_0 \leftarrow$  feasible initial solution
7:  $\xi \leftarrow 0, \theta \leftarrow \theta_0, \eta \leftarrow \eta_{\min}$ 
8:  $X^* \leftarrow TS(X_0, \theta)$ 
9: repeat
10:    $X' \leftarrow \text{Perturb}(X^*, \eta)$ 
      % perturb  $X^*$  with strength  $\eta$ , get  $X'$ 
11:    $X^* \leftarrow TS(X', \theta)$ 
12:   if  $f(X') \leq f(X^*) + 2$  then
13:     repeat
14:        $\theta \leftarrow (1 + \mu) \cdot \theta$ 
      % gradually increase the depth of TS
15:      $X^* \leftarrow TS(X', \theta)$ 
16:     until no better solution is obtained
17:   end if
18:   if  $f(X') < f(X^*)$  then
19:      $X^* \leftarrow X'$ 
      % accept  $X'$  as the best solution found so far
20:      $\theta \leftarrow \theta_0, \eta \leftarrow \eta_{\min}$ 
21:   else
22:      $\theta \leftarrow \theta_0, \xi \leftarrow \xi + 1$ 
23:      $\eta \leftarrow \max\{\eta_{\min} + \lambda \cdot \xi, \eta_{\max}\}$ 
24:   end if
25: until (stop condition is met)

```

در این مقاله مقدار η به طور تجربی در بازه $[\eta_{\min}, \eta_{\max}]$ ، ($\eta_{\min} = 4, \eta_{\max} = 15$) در نظر گرفته شده است. در این الگوریتم مقدار اولیه ξ برابر صفر در نظر گرفته شده است. با توجه به الگوریتم، در ابتدای جست و جوی یک جست و جوی ممنوع با عمق کوتاه ($\theta = 10$) اجرا می شود. هنگامی که این روش نتواند بهترین جواب به دست آمده را بهبود بخشد، اپراتور اغتشاش با قدرت ضعیف ($\eta = \eta_{\min}$) به بهترین جواب که تاکنون به دست آمده است (این عمل در خط ۱۰ الگوریتم که در شکل ۴.۳ نشان داده شده است). سپس جواب به دست آمده از این عمل (x') به عنوان جواب اولیه روش جست و جوی ممنوع در نظر می شود و دوباره این روش با عمق کم پیاده سازی می شود. اگر جواب به دست آمده توسط این روش قابل قبول باشد، به عبارت دیگر اگر مقدار تابع هدف این جواب، به مقدار تابع هدف که تاکنون به دست آمده نزدیک باشد ($f \leq f_{best}$)، به تدریج به مقدار عمق این روش (θ) افزوده می شود ($\theta = (1 + \mu) \cdot \theta$) تا یک جست و جوی بیش از پیش فشرده اطراف این جواب انجام شود. این کار تا زمانی ادامه پیدا می کند که دیگر هیچ بهبودی در جواب حاصل نشود. در مرحله بعد، اگر مقدار تابع هدف جواب به دست آمده از اعمال اپراتورهای تشدید و تنوع (x') از مقدار تابع هدف به دست آمده از روش جست و جوی ممنوع اولیه (x^*) کمتر باشد، این جواب جایگزین جواب اولیه می شود (دستورات ۱۸ و ۱۹ الگوریتم) و مقدار عمق جست و جوی ممنوع

جدول ۵.۳: پارامترهای استفاده شده در روش ATS

پارامترها	توضیح	مقدار
θ	عمق اولیه‌ی (طول) لیست ممنوعه	۱۰
μ	سرعت افزایش θ	۰/۶
θ	عمق لیست ممنوعه	$\theta = (1 + \mu) \cdot \theta$
ξ	فازهای بهبود نیافته‌ی TS	$\xi = \xi + 1$
η_{min}	قدرت اغتشاش اولیه	۴
η_{max}	حداکثر قدرت اغتشاش	۱۵
η	قدرت اغتشاش	$\max\{\eta_{min} + \lambda \cdot \xi, \eta_{max}\}$
λ	به‌روز رسانی فاکتور η	۰/۳
q	تعداد کل دروس برای اغتشاش	۳۰
ϕ	فاکتور اساسی برای انتخاب دروس اغتشاش	۴
τ	فاکتوری برای بررسی حرکت	۲

(θ) و مقدار قدرت اغتشاش (η) به کمترین مقدار خود برمیگردند. در غیراین صورت مقدار عمق به مینیمم مقدار خود بازمی‌گردد و به مقدار ξ یک واحد اضافه می‌گردد. برای به‌روز رسانی مقدار η در این حالت از فرمول $\eta \leftarrow \max\{\eta_{min} + \lambda \cdot \xi, \eta_{max}\}$ استفاده شده است. این مراحل تا زمانی ادامه پیدا می‌کنند تا شروط توقف محقق شود.

۳.۳ نتایج

همان‌طور که بیان شد، در این مقاله سعی شده با به‌کارگیری تغییراتی باعث بهبود روش جست‌وجوی ممنوع شود. این تغییرات شامل استفاده از ساختار همسایگی متفاوت، به‌کارگیری رابطه‌ی مناسب برای تعیین عمق لیست ممنوعه و استفاده از روش‌های مختلف برای اعمال مکانیسم‌های تشدید و تنوع است.

جدول ۶.۳: نتایج روش ATS روی ۱۰ مساله

مساله	n	p	m	f_{min}	f_{ave}	σ	$Iter$	$Time(s)$
مساله ۱	۴۶	۲۰	۱۲	۲۲۴	۲۲۹/۵	۱/۸	۱۵/۵۸۶	۱۸۹
مساله ۲	۵۲	۲۰	۱۲	۱۶	۱۷/۱	۱/۰	۳۵/۲۷۱	۱۸۲
مساله ۳	۵۶	۲۰	۱۳	۷۴	۸۲/۹	۴/۱	۲۰/۵۴۹	۱۶۰
مساله ۴	۵۵	۲۵	۱۰	۷۴	۸۹/۴	۶/۱	۳۷/۳۴۶	۲۰۸

روش پیشنهادی (ATS)، روی چند مساله پیاده‌سازی شده است. پارامترهای مورد استفاده در این روش در جدول ۵.۳ نشان داده شده است. مقادیر به‌دست آمده از این پیاده‌سازی در جدول ۶.۳ آورده شده است. در این جدول شاخص‌ها به این صورت هستند؛

- n : تعداد دروس
- p : تعداد بازه‌های زمانی
- m : تعداد کلاس‌ها
- f_{min} : بهترین مقدار تابع هدف
- f_{ave} : میانگین تابع هدف‌ها
- σ : انحراف استاندارد
- $Iter$: تعداد تکرار
- $Time$: زمان مورد نیاز برای پیدا کردن بهترین راه‌حل

در جدول ۷.۳ روش پیشنهادی با روش TS معمولی با هم مقایسه شده‌اند. در این مقاله به زمان مورد نیاز برای به‌دست آوردن بهترین مقدار از روش TS اشاره‌ای نشده است. بنابراین نمی‌توان مقایسه‌ای عادلانه‌ای بین این دو روش انجام داد. اما با توجه به نتایج به‌دست آمده مشخص می‌شود که، در اکثر موارد روش پیشنهادی به جوابی بهتر از روش TS معمولی رسیده است. در یک مورد هم هر دو روش به مقدار برابر رسیده‌اند.

جدول ۷.۳: مقایسه‌ی توابع هدف روش ATS با روش TS

مساله	ATS	TS
مساله ۱	۲۲۴	۲۳۰
مساله ۲	۱۶	۱۶
مساله ۳	۷۴	۸۲
مساله ۴	۷۴	۹۲

فصل ۴

یک مدل غیرخطی برای مسالهی زمانبندی دروس و به کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن

۱.۴ مقدمه

در این فصل در ابتدا نحوه‌ی مدل‌بندی مسالهی جدول زمانبندی دروس که بر اساس محدودیت‌های دانشگاه صنعتی شاهرود، تنظیم شده است، بیان می‌شود. در این مدل محدودیت‌های سخت و نرم بیشتری نسبت به مدل‌بندی میرحسینی در نظر گرفته شده است. در این پایان‌نامه جدول زمانبندی بر اساس برنامه‌ی درسی تنظیم شده است. جدول زمانبندی دروس دانشگاه بر اساس برنامه‌ی درسی، مجموعه‌ای از جلسات است که باید به تعدادی بازه‌ی زمانی و تعدادی کلاس، که تابع یک گروه از محدودیت‌ها هستند اختصاص داده شوند. هر جلسه بیانگر این است که چه درس توسط چه استادی ارائه می‌شود. در ادامه به بیان روش‌های حلی که برای این مساله در نظر گرفته شده و همچنین پیشنهاداتی جهت بهبود یافتن این روش‌ها می‌پردازیم. در انتها نتایج حاصل از به کارگیری روش‌های پیشنهادی روی چند مثال تصادفی نشان داده می‌شوند.

۲.۴ محدودیت‌ها و پارامترها

محدودیت‌های سختی که برای این مساله در نظر گرفته شده است به صورت زیر است:

- ۱- برای هر درس باید به تعداد ساعات مورد نیاز آن درس، بازه‌ی زمانی و کلاس تخصیص داده شود.
- ۲- جلساتی که استاد یا درس مشترک (منظور دروسی است که در یک برنامه‌ی درسی قرار گرفته‌اند) دارند، نباید در یک زمان برگزار شوند.
- ۳- نباید در ساعاتی که اساتید در دسترس نیستند، به آن‌ها جلسه‌ای تخصیص داد.

۴. یک مدل غیرخطی برای مسالهی زمانبندی دروس و به کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۴۰

جدول ۱.۴: نمادهای مورد استفاده در مدل بندی

نمادها	توضیح
cs_i	تعداد ساعات مورد نیاز درس i ام
s_r	مجموعه‌ی r ام شامل جلسات مشترک
d_i	روز کاری i ام
D	تعداد روزهای کاری
std_i	تعداد دانشجویان جلسه‌ی i ام
cap_j	ظرفیت کلاس r_j ام
N_k	مجموعه جلسات که در بازه‌ی زمانی k در دسترس نیستند.
cr_i	i امین برنامه‌ی درسی شامل یک مجموعه از جلسات

- ۴- دو جلسه نمی‌تواند در یک زمان به یک کلاس اختصاص داده شوند.
- محدودیت‌های نرم که رعایت هر چه بیشتر آن‌ها مورد نیاز است، به این صورت می‌باشد:
- ۵- دو جلسه از یک درس نباید در یک روز یا در دو روز متوالی قرار گیرند.
- ۶- مطلوب است که برنامه‌درسی دانشجویان هر ترم، در تعداد روز کمتری بسته شود.
- ۷- برای هر جلسه تعداد دانشجویان هر درس نباید بیش‌تر از ظرفیت آن کلاس باشد.
- نمادهای مورد استفاده در جدول ۱.۴ ارایه شده است.

در این مساله متغیرها به صورت زیر تعریف می‌شوند:

$$x_{ijk} = \begin{cases} 1 & \text{اگر جلسه‌ی } l_i \text{ در کلاس } r_j \text{ و در زمان } t_k \text{ برگزار شود} \\ 0 & \text{در غیراینصورت} \end{cases} \quad (۱.۴)$$

۳.۴ فرمول‌بندی مساله

تابع هدف این مدل از حاصل جمع سه تابع تشکیل شده است که هر کدام از این تابع‌ها، نشان‌دهنده‌ی یکی از محدودیت‌های نرم است. مینیم کردن این تابع هدف باعث می‌شود مقادیر f_1, f_2, f_3 و در نتیجه نقض محدودیت‌های نرم به حداقل برسد. مقادیر α_1 و α_2 و α_3 در تابع هدف، جریمه‌های مربوط به هر یک از محدودیت‌های نرم است این مقادیر میزان اولویت محدودیت‌های نرم را مشخص می‌کند. هر چقدر این ضریب جریمه کوچک‌تر باشد، بدین معنی است که اولویت برای رعایت محدودیت مربوطه پایین‌تر است. مدل‌بندی مساله به صورت زیر است:

$$\min f(x) = \alpha_1 \sum_i f_1(x) + \alpha_2 \sum_q f_2(x) + \alpha_3 \sum_{x_{ijk}=1} f_3(x) \quad (۲.۴)$$

s.t

$$(۱) \quad \sum_j \sum_k x_{ijk} = cs_i \quad \forall i$$

$$(۲) \quad \sum_{i \in sm} \sum_j x_{ijk} \leq 1 \quad \forall k, m$$

$$(۳) \quad \sum_{i \in N_k} \sum_j x_{ijk} = 0 \quad \forall k$$

$$(۴) \quad \sum_i x_{ijk} \leq 1 \quad \forall j, k$$

که در آن توابع زیر محدودیت‌های نرم را تشکیل می‌دهند:

$$(۵) \quad f_1(x) = \sum_i \sum_{h=1}^D \chi \left(\sum_j \sum_{k \in d_h} x_{ijk} + \sum_j \sum_{k \in d_{h+1}} x_{ijk} > 1 \right)$$

$$(۶) \quad f_2(x) = \sum_q \sum_h \chi \left(\sum_{i \in cr_q} \sum_j \sum_{k \in d_h} x_{ijk} = 1 \right)$$

۴. یک مدل غیرخطی برای مسالهی زمانبندی دروس و به کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۴۲

$$(۷) \quad \forall x_{ijk} = 1 \quad f_3(x) = \begin{cases} std_i - cap_j & std_i < cap_j \\ 0 & \text{در غیراینصورت} \end{cases}$$

در این مدل‌بندی x تابع مشخصه است که ارزش یک می‌گیرد اگر گزاره‌ی داده شده درست باشد و ارزش صفر می‌گیرد اگر اشتباه باشد. در محدودیت اول تضمین می‌شود که به هر درس، به تعداد ساعات مورد نیاز بازه‌ی زمانی و کلاس تخصیص داده شود. در محدودیت دوم بیان می‌شود دروسی که با هم نقطه‌ی اشتراک دارند (دارای استاد مشترک هستند یا در یک برنامه‌ی درسی قرار دارند)، نباید به یک بازه‌ی زمانی یکسان تخصیص داده شوند.

همان‌طور که در جدول ۱.۴ بیان شده است، N_k شامل جلساتی است که در بازه‌ی زمانی k در دسترس نیستند. محدودیت سوم تضمین می‌کند دروسی که به ازای هر k در N_k قرار دارند، نباید به این بازه‌ی زمانی اختصاص داده شوند. با رعایت محدودیت چهارم، در یک بازه‌ی زمانی و در یک کلاس حداکثر یک جلسه برگزار می‌شود.

در محدودیت پنجم چنانچه تعداد جلسات تخصیص داده شده به یک درس در دو روز متوالی بیشتر از یک باشد، تابع مقدار یک می‌گیرد. در نتیجه یک واحد به مقدار f_1 اضافه می‌شود و به خاطر نقض محدودیت نرم در تابع هدف جریمه می‌شود. در محدودیت ششم، چنانچه از یک برنامه‌ی درسی در یک روز فقط یک درس ارائه شود، تابع f_2 جریمه می‌شود. محدودیت هفتم هم تضمین می‌کند چنانچه تعداد دانشجویان تخصیص داده شده به یک کلاس از ظرفیت آن بیشتر باشد، تابع f_3 به اندازه‌ی این اختلاف مقدار می‌گیرد و بنابراین در تابع هدف جریمه می‌شود.

برای استفاده از این مدل ابتدا باید اطلاعات مورد نیاز جمع‌آوری شوند. اطلاعاتی از قبیل:

- دروسی که باید ارائه شوند
- هر درس توسط چه استادی ارائه می‌شود
- تعداد ساعات مورد نیاز برای هر درس
- بازه‌های زمانی موجود
- روزهای کاری
- تعیین روزها و ساعاتی که اساتید در دسترس نیستند
- تعیین برنامه‌ی درسی برای دانشجویان هر ترم
- تعداد کلاس‌های موجود
- ظرفیت هر کلاس

۴.۴ روش های حل

روش های ابتکاری و فوق ابتکاری زیادی برای حل مدل های ریاضی وجود دارند. همان طور که در فصل اول گفته شد مشکل روش های ابتکاری توقف آن ها در بهینه ی محلی است که الگوریتم های فراابتکاری با معرفی اصول سیستماتیک با این مشکل برخورد می کنند. ما در این فصل برای حل این مدل ریاضی از روش ابتکاری جست و جوی محلی و روش فراابتکاری جست و جوی ممنوع استفاده می کنیم، سپس با مقایسه ی نتایج به دست آمده بررسی می شود که آیا اصول به کار رفته در روش جست و جوی ممنوع کارایی لازم را داشته است یا نه.

۱.۴.۴ خواص الگوریتم جست و جوی ممنوع مورد استفاده

عناصری از الگوریتم جست و جوی ممنوع در ارتباط با مساله ی جدول زمان بندی در این جا تعریف می شوند:

جواب x : هر جواب x یک ماتریس است که در ابتدا تمام عناصرش ۱- هستند. در این مدل بندی جلسات تخصیص داده می شوند. به عبارت دیگر هر جلسه که نشانگر ترکیب یک درس با یک استاد است (این جلسات از قبل مشخص شده است)، به بازه های زمانی و کلاس های موجود تخصیص داده می شوند. برای مثال درس c_1 با استاد l_1 تشکیل جلسه ی i_1 را می دهد و به همین ترتیب بقیه ی دروس به همراه اساتید مورد نظر کاربر، تشکیل بقیه ی جلسات را می دهند.. مولفه های این ماتریس با x_{ijk} نمایش داده می شوند که طبق رابطه ۱.۴ مقدار می گیرند. به عبارت دیگر اگر جلسه ی i به کلاس j و در بازه ی زمانی k تخصیص داده شود، متغیر x_{ijk} مقدار ۱ می گیرد و در غیر این صورت مقدار ۰ خواهد گرفت. برای درک بهتر، این ماتریس در جدول ۲.۴ نشان داده شده است. سطرهای این ماتریس را جلسات تشکیل می دهند و ستون های این ماتریس نشان دهنده ی تمام ترکیبات ممکن بین بازه های زمانی و کلاس های موجود است. تعداد کل ستون های این ماتریس برابر است با حاصل ضرب تعداد بازه های زمانی در تعداد کلاس ها. برای مثال فرض کنید که ۵ جلسه داریم $(i_5, i_4, \dots, i_1)$ ، و تعداد بازه های زمانی برابر ۳ و تعداد کلاس ها نیز ۳ باشد. بنابراین تعداد سطرهای این ماتریس برابر ۵ و تعداد ستون های آن ۹ است.

جواب اولیه: در روش جست و جوی ممنوع معمولاً الگوریتم با یک جواب تصادفی آغاز می شود و حرکت متوالی به سمت یکی از همسایگی ها انجام می شود تا سرانجام به یک جواب مناسب برسد. در این روش اگر این جواب اولیه ی تصادفی به صورت مناسب انتخاب نشده باشد، کارایی الگوریتم پایین می آید. این یکی از ایرادهای این الگوریتم است. ما در این فصل با یافتن یک جواب اولیه ی شدنی (جوابی که تمام محدودیت های سخت در آن رعایت شده باشند) سعی در رفع این مشکل داریم. این جواب اولیه بدین صورت به دست می آید:

ما برای به دست آوردن جواب اولیه ی شدنی، از روش حریصانه استفاده می کنیم. برای این کار ابتدا جلسه ای

۴. یک مدل غیرخطی برای مسالهی زمانبندی دروس و به کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۴۴

جلسه (کلاس-زمان)	$j_1 k_1$	$j_2 k_1$	$j_3 k_1$	$j_1 k_2$	$j_2 k_2$	$j_3 k_2$	$j_1 k_3$	$j_2 k_3$	$j_3 k_3$
i_1	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
i_2	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
i_3	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
i_4	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱
i_5	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱	-۱

جدول ۲.۴: ماتریس نشان‌دهنده‌ی جواب اولیه

که در تعداد بازه‌های زمانی کمتری در دسترس می‌باشد را به اولین بازه‌ی زمانی در دسترس تخصیص می‌دهیم. جلساتی که با این جلسه در یک برنامه‌ی درسی قرار دارند یا دارای استاد مشترک هستند نباید به این بازه اختصاص داده شوند. در شکل ۲.۴ پس از اینکه جلسه‌ی i که کمتر در دسترس است را یافتیم، اولین بازه‌ی که این جلسه در دسترس است را به آن اختصاص می‌دهیم (به آن درایه مقدار ۱ می‌دهیم). سپس به تمام درایه‌هایی که در آن ستون قرار گرفته‌اند، مقدار ۰ تخصیص می‌دهیم. با این کار تضمین می‌شود که در یک بازه‌ی زمانی و در یک زمان بیشتر از یک جلسه تخصیص داده نشود (محدودیت سخت چهارم). این کار را برای بقیه‌ی جلسات نیز تکرار می‌کنیم. باید به این نکته نیز توجه داشت که برای رعایت محدودیت سخت اول، حاصل جمع درایه‌های هر ستون باید برابر تعداد ساعات مورد نیاز برای آن درس باشد. برای رعایت محدودیت سخت سوم، باید در نظر داشت که در ساعاتی که اساتید در دسترس نیستند به آن‌ها جلسه‌ای اختصاص داده نشود. همچنین جلساتی که با هم استاد یا درس مشترک دارند، نباید در یک بازه‌ی زمانی برگزار شوند (محدودیت سخت دوم). نمونه‌ای از این اعمال در جدول ۲.۴ نشان داده شده است.

تابع هدف: در این تابع هدف، با جریمه کردن نقض محدودیت‌های نرم، تلاش می‌شود تا حدامکان از رعایت نشدن این محدودیت‌ها جلوگیری شود. جریمه‌های $\alpha_1, \alpha_2, \alpha_3$ ، با توجه به نیاز دانشگاه تنظیم می‌شود. از آن‌جا که برای ما رعایت محدودیت نرم ۵ از دو محدودیت نرم دیگر مهم‌تر است، لذا این مقادیر را به صورت زیر در نظر می‌گیریم:

$$\alpha_1 = 5, \alpha_2 = 1, \alpha_3 = 1$$

ساختار همسایگی‌ها: دو همسایگی توسط حرکت‌های زیر پیشنهاد شده‌اند:

همسایگی ساده: در این نوع همسایگی، بازه‌ی زمانی تخصیص داده شده به یک جلسه در صورتی که شدنی بودن جواب از بین نرود، به یک بازه‌ی زمانی خالی منتقل می‌شود.

جلسه (کلاس-زمان)	$j_1 k_1$	$j_2 k_1$	$j_3 k_1$	$j_1 k_2$	$j_2 k_2$	$j_3 k_2$	$j_1 k_3$	$j_2 k_3$	$j_3 k_3$
i_1	۰	۰	۱	۰	۰	۰	۰	۰	۱
i_2	۱	۰	۰	۰	۰	۰	۰	۰	۰
i_3	۰	۱	۰	۰	۰	۰	۰	۰	۰
i_4	۰	۰	۰	۰	۱	۰	۰	۰	۰
i_5	۰	۰	۰	۰	۰	۰	۰	۱	۰

جدول ۳.۴: ماتریس نشان‌دهنده‌ی جواب اولیه‌ی شدنی

همسایگی مبادله‌ای: در همسایگی مبادله‌ای، بازه‌ی زمانی و کلاس تخصیص داده شده به دو درس مختلف، در صورت شدنی ماندن جواب با یکدیگر مبادله می‌شوند.

همسایگی‌های پیشنهادی: در بخش قبل چگونگی کار همسایگی ساده و مبادله‌ای توضیح داده شد. در این مقاله ما دو نوع ساختار همسایگی جدید که ترکیبی از این دو حرکت است پیشنهاد می‌کنیم. در حرکت ساده فقط یک درس می‌تواند در جدول زمانبندی تغییر کند. در حرکت مبادله‌ای دو درس می‌توانند با هم مبادله شوند. بنابراین حرکت مبادله‌ای تاثیر تنوع بیشتری از حرکت ساده دارد. هدف از ساختار همسایگی پیشنهادی، ترکیب این دو نوع حرکت است.

اولین ساختار همسایگی به این صورت است که، بعد از یافتن جواب اولیه‌ی شدنی، الگوریتم با استفاده از حرکت ساده آغاز خواهد شد. اگر در تعداد تکراری که از قبل تعیین شده، جواب بهبود نیافت این بار الگوریتم با بهترین جواب به‌دست آمده تاکنون و حرکت مبادله‌ای ادامه پیدا می‌کند.

در دومین ساختار همسایگی پیشنهادی هر دو حرکت ساده و مبادله‌ای به‌طور هم‌زمان در تعیین جواب فعلی دخالت دارند. این روش به این صورت است که در هر تکرار بهترین جواب تولیدشده توسط هر کدام از این دو حرکت با هم مقایسه می‌شوند. بعد از مقایسه، جوابی که مقدار تابع هدفش کمتر است به‌عنوان جواب بعدی در نظر گرفته می‌شود.

لیست ممنوعه: همان‌طور که بیان شد، در روش جست‌و‌جوی ممنوع فهرستی وجود دارد که جابه‌جایی‌های منع شده در آن نگهداری می‌شوند که به فهرست تابو معروف است. کاربرد اصلی این لیست پرهیز از همگرا شدن به جواب‌های بهینه‌ی محلی است. به عبارت دیگر به کمک فهرست تابو جابه‌جایی توسط حرکت‌هایی که اخیراً انجام شده‌اند، ممنوع خواهد شد. در این روش در همسایگی ساده، زمانی که یک جلسه (برای مثال i_1) از یک بازه‌ی زمانی (t_1) به یک بازه‌ی زمانی خالی منتقل می‌شود، تا چند تکرار بعد (به اندازه‌ی

عمق (TS) نباید به بازهی زمانی t_1 بازگردد. برای همسایگی N_2 نیز زمانی که جلسه ی i_1 با بازهی زمانی t_1 و کلاس r_1 ، با جلسه ی i_2 در بازهی زمانی t_2 و کلاس r_2 مبادله می شوند، تا چند تکرار بعد نمی توانند به جایگاه قبلی خود بازگردند. یکی از مشخصه های مهم لیست ممنوع عمق آن است. این بدین معنی است که وقتی حرکتی وارد این لیست شد، تا چند تکرار بعد انجام این حرکت ممنوع می باشد. اگر عمق این روش بالا باشد حرکت های بیشتری مورد بررسی قرار می گیرد، اما این باعث نمی شود لزوماً جواب بهتری حاصل شود. در حالت کلی مقدار عمق این روش می تواند ثابت یا متغیر باشد. در این جا برای هرچه بهتر شدن جواب، ما عمق لیست ممنوعه را به صورت زیر در نظر می گیریم:

$$T = \frac{1}{3} \times n$$

معیار تنفس: اگر یک حرکت منجر به جوابی شود که مقدار تابع هدفش بهتر از مقداری باشد که تاکنون به دست آمده است، حتی اگر این حرکت ممنوع باشد ممنوعیت این جواب نادیده گرفته می شود و جواب به دست آمده از این حرکت به عنوان جواب بعدی در نظر گرفته می شود.

انتخاب بهترین همسایه: پس از یافتن کلیه ی حرکت های شدنی، یک حرکت با بهترین مقدار تابع هدف انتخاب می شود. اگر این حرکت در لیست ممنوع بود بررسی می شود که آیا معیار تنفس در آن رعایت شده است یا خیر. اگر معیار تنفس در آن رعایت شده بود این حرکت پذیرفته می شود. در غیر این صورت حرکت دیگری با بهترین مقدار تابع هدف مورد بررسی قرار می گیرد. بدین ترتیب بهترین حرکت قابل قبول انتخاب می شود و جواب بعدی، جواب تولید شده توسط این حرکت است.

شرایط توقف: در این روش، دو شرط به عنوان شروط خاتمه ی الگوریتم در نظر گرفته شده است:

- اگر جوابی به دست آید که مقدار تابع هدف آن صفر باشد، الگوریتم متوقف می شود.
- اگر به تعداد تکرارهایی که از قبل مشخص شده رسیدیم، الگوریتم متوقف می شود.

این خواص به عنوان خواص پایه ی روش جست و جوی ممنوع ما در نظر گرفته شده است که از این روش با نام روش جست و جوی ممنوع ساده یاد می کنیم.

۲.۴.۴ جست و جوی موضعی:

روش جست و جوی موضعی یک روال جست و جوی تکرار شونده است که از یک جواب شدنی شروع می کند. سپس با استفاده از همسایگی های تعریف شده به سمت راه حل های مجاور حرکت می کند. حرکت به سمت راه حل های مجاور در صورتی انجام می شود که راه حل جدید به جوابی بهتر یا برابر، نسبت به راه حل قبلی برسد. جواب اولیه و ساختار همسایگی ها همانند ساختار همسایگی روش جست و جوی ممنوع در نظر گرفته شده است. در این روش ما از دو راه برای انتخاب همسایگی استفاده می کنیم. در روش اول پس

از پیدا شدن تمام همسایگی‌های شدنی، از بین آن‌ها همسایگی را که منجر به جواب بهتری نسبت به بقیه‌ی همسایگی‌ها می‌شود، انتخاب می‌کنیم و به‌عنوان حرکت بعدی از آن استفاده می‌کنیم. در روش بعدی از میان تمام همسایگی‌ها، بدون در نظر گرفتن مقدار تابع هدف و به تصادف یکی را انتخاب می‌کنیم و جواب حاصل از این حرکت به‌عنوان جواب بعدی روش در نظر گرفته می‌شود. در ادامه با تغییراتی که در بعضی خواص این الگوریتم‌ها می‌دهیم، سعی می‌شود این روش را تا حد زیادی بهبود بخشیم.

۵.۴ روش‌های پیشنهادی برای بهبود روش‌های حل

- **بهبود جواب اولیه:** به‌عنوان اولین تغییر جهت بهبود بخشیدن به جواب شدنی اولیه تا حد قابل توجهی، از ابتکار تعویض تخصیص دروسی که در یک برنامه‌ی درسی قرار گرفته‌اند استفاده می‌کنیم. با این کار تنها محدودیت سخت سوم را برای شدنی باقی ماندن جواب بررسی می‌کنیم. زیرا با این کار رعایت محدودیت‌های اول و دوم و چهارم به مخاطره نمی‌افتد. برای این کار مکان دروسی که در یک برنامه‌ی درسی قرار گرفته‌اند را با هم تعویض می‌کنیم و هر بار مقدار تابع هدف را برای جواب جدید محاسبه می‌کنیم و از میان همه‌ی آن‌ها بهترین جواب را انتخاب می‌کنیم. با این کار به تعداد برنامه‌های درسی در یک زمان، مبادله‌ی تخصیص جلسات را داریم. بنابراین مقدار تابع هدف تا حد زیادی تغییر می‌کند.
- **تغییر در ساختار همسایگی ساده:** در دومین روش برای بهبود بخشیدن به الگوریتم پایه، در قسمت ساختار همسایگی (همسایگی ساده) تغییر ایجاد می‌کنیم. در این همسایگی به‌جای یافتن یک بازه‌ی زمانی خالی و تخصیص دادن همه‌ی دروس به همین یک بازه‌ی زمانی خالی و مقایسه‌ی آن‌ها برای یافتن بهترین جواب، تمام بازه‌های زمانی موجود در جدول زمانبندی را یافته و دروس را به تمام این بازه‌های زمانی اختصاص می‌دهیم و بهترین جواب را پیدا می‌کنیم. همان‌طور که مشخص است این کار زمان برنامه را تا حدود قابل توجهی افزایش می‌دهد اما منجر به ایجاد جواب‌هایی می‌شود که نسبت به جواب‌های به‌دست آمده از روش‌های دیگر، بسیار بهتر است.
- در روش بعدی با ترکیب دو روش قبلی تلاش می‌شود جواب بهینه را در مدت زمان کمتری به‌دست آوریم.

با توجه به مطالب ذکر شده، ما برای حل این مدل ریاضی از سه روش جست‌وجوی محلی ساده، جست‌جوی محلی تصادفی و جست‌جوی ممنوع استفاده می‌کنیم. در هر کدام از این روش‌ها با به‌کار بردن ابتکاراتی که

۴. یک مدل غیرخطی برای مساله‌ی زمانبندی دروس و به‌کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۴۸

جدول ۴.۴: داده‌های مسائل تصادفی

مسائل	تعداد دروس	تعداد کلاس‌ها	تعداد بازه‌های زمانی در هر روز	تعداد روزهای کاری
مساله‌ی ۱	۵	۲	۲	۳
مساله‌ی ۲	۱۰	۲	۴	۳
مساله‌ی ۳	۱۵	۳	۴	۴
مساله‌ی ۴	۲۰	۳	۳	۵
مساله‌ی ۵	۲۵	۳	۳	۵

در بالا بیان شد، سعی می‌شود روش‌ها را برای به دست آوردن تابع هدفی بهتر آماده کرد. این روش‌ها با استفاده از نرم افزار متلب ۲۰۱۲ کدنویسی شده است. در ادامه نتایج حاصل از پیاده‌سازی این روش‌ها روی چند مساله‌ی تصادفی که اطلاعات آن در جدول ۴.۴ آورده شده است، بیان شده است.

۶.۴ پیاده‌سازی روش‌های حل روی چند مساله با داده‌های تصادفی

در جدول ۵.۴ نتایج حاصل از به‌کارگیری روش جستجوی محلی با انتخاب ساده نشان داده شده است. این روش با به کار گیری چهار راهکار جستجوی محلی ساده (روش اول)، جستجوی محلی با بهبود در جواب اولیه (روش دوم)، جستجوی محلی با تغییر در ساختار همسایگی ساده (روش سوم) و جستجوی محلی ترکیبی (روش چهارم) بررسی شده است. با مقایسه‌ی نتایج به‌دست آمده در مساله‌ی اول با تعداد ۵ درس، مشاهده می‌شود که دو روش ۳ و ۴ نتایج بهتری نسبت به دو روش دیگر تولید می‌کنند. زمان مورد نیاز دو روش با هم برابر است. در مورد دوم نیز این دو روش کارایی بهتری از خود نشان داده‌اند. در مساله‌ی سوم، بهترین مقدار از روش ۴ به‌دست می‌آید. علاوه بر این، این مقدار در مدت زمان کوتاهی توسط این روش به دست آمده است. در مساله چهارم، روش سوم نسبت به سه روش دیگر مقدار تابع هدف بهتری را ایجاد کرده است. البته مقدار به‌دست آمده از روش ۴ نیز نزدیک به این جواب است و در مدت زمان کمتری نسبت

جدول ۵.۴: جدول نتایج روش جستجوی محلی روی چند مثال تصادفی

تعداد دروس	مقدار تابع هدف				مدت زمان اجرا (ثانیه)			
	روش ۱	روش ۲	روش ۳	روش ۴	روش ۱	روش ۲	روش ۳	روش ۴
مساله‌ی ۱	۱۴	۱۴	۷	۷	۰/۲۸	۰/۴۰	۰/۳۰	۰/۳۰
مساله‌ی ۲	۲۲	۲۲	۲	۲	۱/۴۹	۱/۸۵	۷/۰۵	۸/۰۲
مساله‌ی ۳	۴۲	۴۲	۱	۰	۶/۶۶	۱/۹۴	۶۰/۳۹	۴۶/۹۵
مساله‌ی ۴	۳۴	۲۹	۱	۳	۲۰/۳۸	۲۲/۲۶	۸۹/۲۴	۶۱/۲۱
مساله‌ی ۵	۳۹	۲۳	۳	۱۵	۴۹/۶۶	۵۵/۱۵	۱۳۵/۹۸	۷۲/۱۵

به روش ۳ به دست آمده است. در مساله‌ی آخر روش ۳ کارکرد بهتری نسبت به سایر روش‌ها داشته است.

در جدول ۶.۴ نتایج به‌کارگیری جستجوی محلی با انتخاب تصادفی نشان داده شده است. در دو مساله‌ی اول و دوم، باز هم روش‌های ۳ و ۴ به مقادیر بهتری نسبت به دو روش اول رسیده‌اند. روش ۴ در مدت زمان کمتری به بهترین جواب رسیده است. در مساله‌ی سوم نیز روش ۴ نسبت به سه روش دیگر به جواب بهتری (نزدیک به جواب بهینه) رسیده است. در مساله‌ی چهارم هم روش ۳ و ۴ به مقدار تابع هدف بهتری رسیده‌اند. مدت زمان مورد نیاز برای رسیدن به این جواب نیز در هر دو روش تقریباً یکسان است. در مساله‌ی پنجم روش ۴ به جوابی بهتر نسبت به دیگر روش‌ها رسیده است. زمان رسیدن به این جواب از روش ۳ کمتر بوده است.

در جدول ۷.۴ نتایج پیاده‌سازی جستجوی ممنوع روی همان مثال‌های قبلی بیان شده است. در مساله‌ی اول، روش ۴ در مدت زمان کمتری به بهترین جواب نسبت به سه روش دیگر، رسیده است. در مساله‌ی دوم دو روش ۳ و ۴ به مقدار تابع هدف یکسانی رسیده است. در مورد سوم دو روش ۳ و ۴ به مقدار بهینه رسیده‌اند. با این تفاوت که روش ۴ در مدت زمان بسیار کمتری به این جواب رسیده است. در مساله‌ی چهارم، روش

۴. یک مدل غیرخطی برای مسالهی زمانبندی دروس و به‌کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۵۰

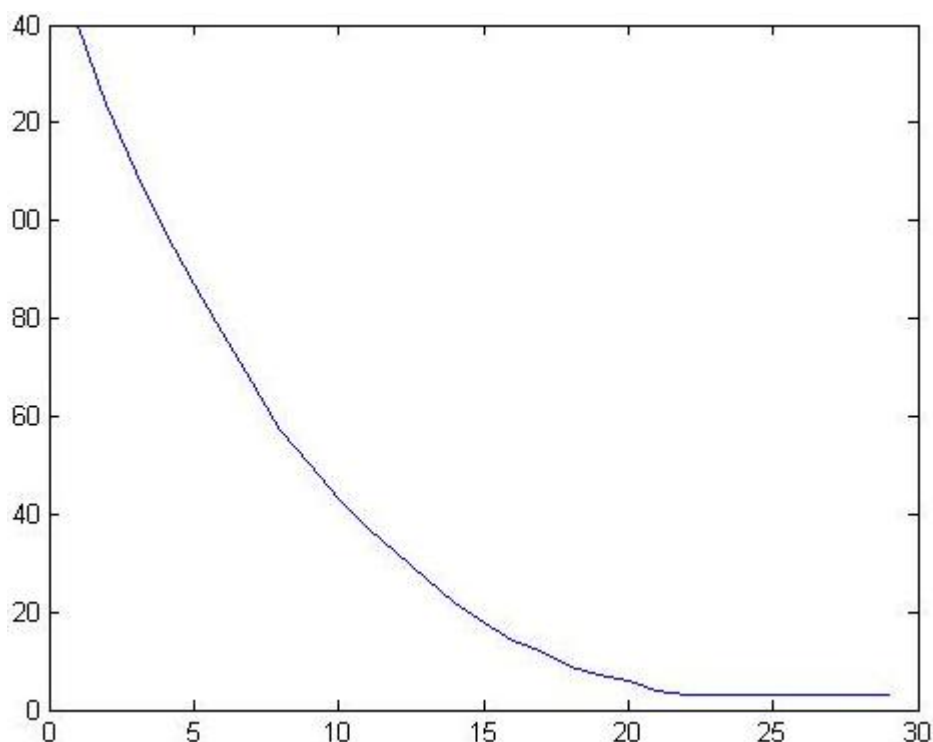
جدول ۶.۴: جدول نتایج روش جستجوی محلی تصادفی روی چند مثال تصادفی

مسائل	مقدار تابع هدف				مدت زمان اجرا (ثانیه)			
	روش ۱	روش ۲	روش ۳	روش ۴	روش ۱	روش ۲	روش ۳	روش ۴
مسالهی ۱	۱۴	۱۴	۷	۷	۰/۰۵	۰/۰۷	۰/۸۸	۰/۸۴
مسالهی ۲	۱۷	۱۷	۴	۴	۸/۵۰	۸/۲۱	۳۰/۲۹	۲۸
مسالهی ۳	۴۲	۳۷	۷	۲	۲۶/۷۴	۴۰/۵۲	۱۰۰/۹۵	۱۲۰
مسالهی ۴	۳۶	۲۴	۱۲	۱۲	۹۴/۵۰	۸۰/۳۹	۱۸۰/۶۶	۱۸۴/۴۵
مسالهی ۵	۳۱	۲۳	۲۳	۲۰	۱۹۶/۲۷	۱۷۷/۲۷	۲۶۰	۲۳۰/۰۶

جدول ۷.۴: جدول نتایج روش جستجوی ممنوع روی چند مثال تصادفی

مسائل	مقدار تابع هدف				مدت زمان اجرا (ثانیه)			
	روش ۱	روش ۲	روش ۳	روش ۴	روش ۱	روش ۲	روش ۳	روش ۴
مساله‌ی ۱	۱۴	۱۴	۱۱	۷	۰/۰۶۲	۰/۱۰	۰/۴۶	۰/۳۹
مساله‌ی ۲	۱۷	۱۷	۲	۲	۵/۰۵	۷/۲۹	۶/۲۷	۸/۱۹
مساله‌ی ۳	۳۷	۳۷	۰	۰	۱۳/۰۳	۲۶/۰۱	۳۰۰/۶۰	۴۳/۱۹
مساله‌ی ۴	۹	۱۰	۰	۱	۱۰۶/۰۵	۱۰۰/۰۷	۱۲۴/۵۳	۸۷/۱۱
مساله‌ی ۵	۱۰	۱۷	۰	۱۰	۳۲۸/۰۴	۱۶۱/۱۱	۲۱۴/۹۷	۸۰/۱۳

۴. یک مدل غیرخطی برای مسالهی زمانبندی دروس و به‌کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۵۲



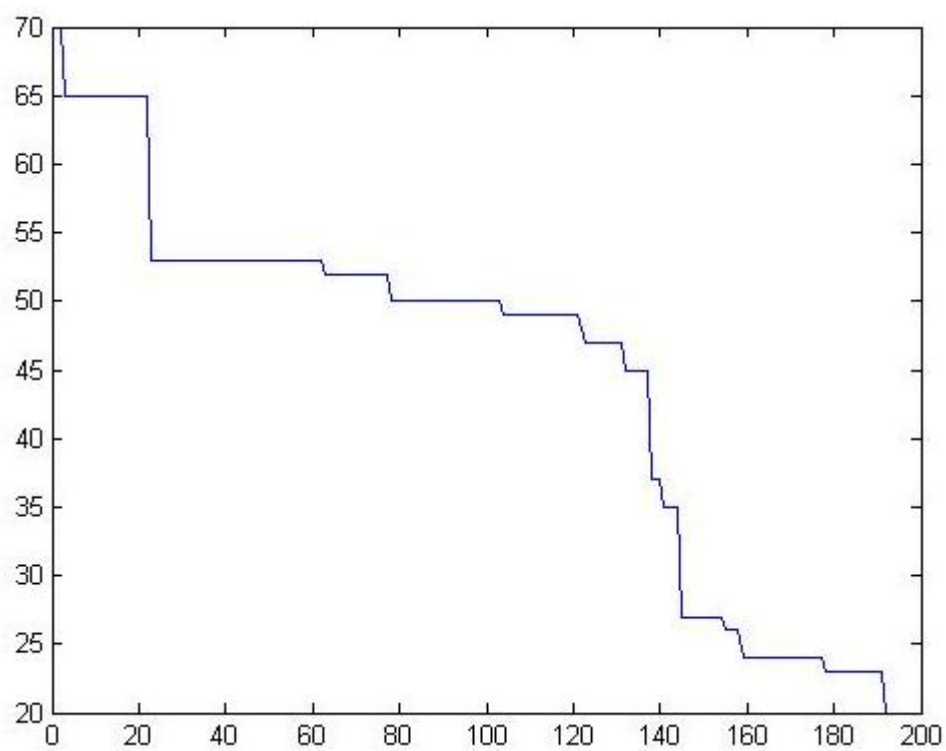
شکل ۱.۴: نمودار به‌دست‌آمده از روش جستجوی محلی

۳ به مقدار بهینه رسیده است. روش ۴ هم جوابی با یک واحد اختلاف نسبت به مقدار بهینه و در مدت زمان بسیار کمتری نسبت به روش ۳، به‌دست آورده است. در مسالهی آخر روش ۳ به جواب بهینه برای این مساله رسیده است.

در شکل‌های ۱.۴ و ۲.۴ و ۳.۴ نمودارهای بهترین تابع هدف بر حسب تعداد تکرار حاصل از پیاده‌سازی این سه روش روی مسالهی ۵ نشان داده شده است.

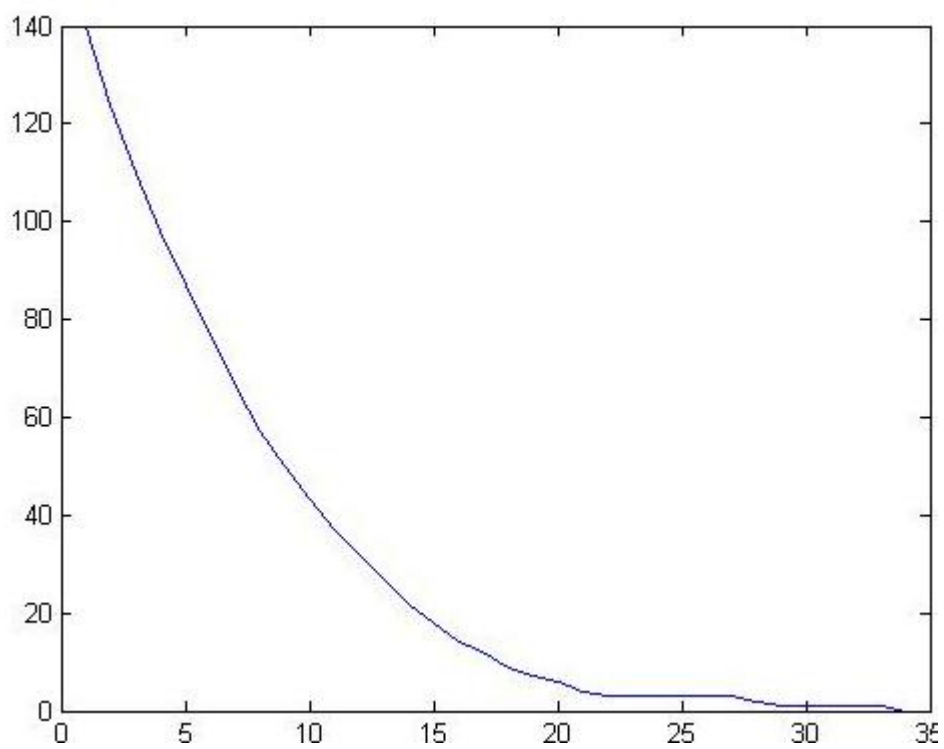
۷.۴ پیاده‌سازی روش برای مسالهی زمانبندی دروس دانشکده‌ی ریاضی شاهرود

در این بخش به توصیف یک مثال واقعی برای بررسی کارایی برنامه‌ی موجود می‌پردازیم. با توجه به نتایج به‌دست آمده از پیاده‌سازی روش‌های جستجوی محلی، روش جستجوی محلی تصادفی و جستجوی ممنوع روی مسایل با داده‌های تصادفی، مشخص شد که روش جستجوی ممنوع از دو روش دیگر کاراتر است. بنابراین ما برای حل مسالهی واقعی از این روش استفاده می‌کنیم. در این مثال واقعی، هدف تهیه‌ی جدول زمانبندی برای دانشکده‌ی ریاضی دانشگاه صنعتی شاهرود می‌باشد. بنابراین لازم است به توصیف داده‌های موجود در این دانشکده بپردازیم.



شکل ۲.۴: نمودار به‌دست‌آمده از روش جست‌جوی محلی تصادفی

۴. یک مدل غیرخطی برای مسالهی زمانبندی دروس و به‌کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۵۴



شکل ۳.۴: نمودار به‌دست‌آمده از روش جستجوی ممنوع

جدول ۸.۴: جدول مقایسه‌ی نتایج پیاده‌سازی سه روش جستجو

تعداد دروس			مقدار تابع هدف			مدت زمان اجرا (ثانیه)		
			روش ۱	روش ۲	روش ۳	روش ۱	روش ۲	روش ۳
مسالهی ۱			۷	۷	۷	۰/۳۰	۰/۸۴	۰/۳۹
مسالهی ۲			۲	۴	۲	۷/۰۵	۲۸	۶/۲۷
مسالهی ۳			۰	۲	۰	۴۶/۹۵	۱۲۰	۴۳/۱۹
مسالهی ۴			۱	۱۲	۰	۸۹/۲۴	۱۸۰/۴۵	۱۲۴/۵۳
مسالهی ۵			۳	۲۰	۰	۱۳۵/۹۸	۲۳۰/۶۶	۲۱۴/۹۷

۴. یک مدل غیرخطی برای مساله‌ی زمانبندی دروس و به‌کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۵۶

جدول ۹.۴: داده‌های مربوط به دانشکده‌ی ریاضی

مقدار اندیس در مثال واقعی	مفهوم اندیس	اندیس
۷۵	تعداد دروس	C
۲۷	تعداد اساتید	L
۵	تعداد روزهای کاری	D
۴	ساعات موجود در یک روز	h

۱.۷.۴ توصیف داده‌های موجود

شرایط و امکانات موجود در این دانشکده در جدول ۹.۴ آمده است؛ اطلاعات دیگر مورد نیاز برای استفاده از این برنامه عبارت‌اند از:

- هر درس توسط کدام استاد تدریس می‌شود
 - هر استاد در کدام روزها نمی‌توانند حضور داشته باشند.
 - هر درس به چند ساعت در هفته نیاز دارد
 - تعیین دروس اختصاصی که برای دانشجویان هر ترم ارائه می‌شود
- همان‌طور که بیان شد، دروسی که نباید با هم به یک بازه‌ی زمانی تخصیص داده شوند شامل دروسی است که دارای استاد مشترک هستند و یا با هم در یک برنامه‌ی درسی قرار گرفته‌اند. برنامه‌های درسی اختصاص داده شده برای دانشجویان هر ترم این دانشکده در نیمسال دوم با توجه به چارت درسی به‌صورت زیر است

- تحقیق در عملیات ۲، آنالیز عددی ۲، توابع مختلط
- سری‌های زمانی، بهینه‌سازی خطی
- برنامه‌نویسی پیشرفته، ریاضی عمومی ۳

جدول ۱۰.۴: دروس اختصاصی و روزهای غیرقابل دسترس اساتید

روزهای غیرقابل دسترس	دروس تخصیص داده شده به استاد	استاد
سه‌شنبه	هندسه دیفرانسیل سراسری	۱
چهارشنبه-یکشنبه ۱۰-۱۲ شنبه ۱۰-۱۲	کنترل بهینه و حساب تغییرات، بهینه‌سازی، نظریه تقریب	۲
چهارشنبه	استنباط ۱، نظریه احتمال (گ۱)	۳
چهارشنبه	استنباط ۲، سری‌های زمانی، روش‌های آماری	۴
یکشنبه-شنبه ۱۶-۱۸ سه‌شنبه ۱۰-۱۲	رمزنگاری، مبانی ترکیبات (گ۲)	۵
چهارشنبه	مباحثی در نظریه فازی گراف، نظریه فازی، مباحثی در مدلسازی گراف، نظریه گراف	۶
چهارشنبه	اقتصاد مهندسی، ریاضی ۲ (گ۲)، آنالیز ریاضی ۳، زبان تخصصی	۷
چهارشنبه-شنبه ۱۶-۱۸	مبانی آنالیز عددی، آنالیز عددی ۱ و ۲، حل عددی معادلات دیفرانسیل جزئی (گ۲)	۸
چهارشنبه-شنبه ۱۶-۱۸	آنالیز تابعی، جبرهای باناخ، جبرهای وقون نیومن	۹
چهارشنبه	جبر جامع، جبر ۲، نظریه حلقه‌ها ۲	۱۰
چهارشنبه	مبانی ماتریس و جبر خطی (گ۲)، جبر ۱، جبر خطی عددی	۱۱
چهارشنبه-یکشنبه ۱۴-۱۶	روش‌های نقطه داخلی، بهینه‌سازی مدل‌های غیرخطی	۱۲
چهارشنبه	تحقیق پیشرفته ۲، کنترل بهینه	۱۳
یکشنبه	آنالیز بهترین تقریب، آنالیز تابعی هندسی، مبانی آنالیز ریاضی، مباحثی در قضایای باناخ	۱۴

۴. یک مدل غیرخطی برای مساله‌ی زمانبندی دروس و به‌کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۵۸

روزهای غیرقابل دسترس	دروس اختصاصی	استاد
یکشنبه	آنالیز حقیقی، آنالیز مختلط، توابع مختلط	۱۵
چهارشنبه	هندسه منیفلد ۱ و ۲، گروه جبرلی ۲	۱۶
شنبه	نظریه گروه‌های متناهی، مبانی ماتریس و جبرخطی (گ ۱)، نظریه گروه‌های خطی	۱۷
چهارشنبه	تحقیق در عملیات ۲، بهینه‌سازی خطی	۱۸
دوشنبه-شنبه ۱۶-۱۴	نظریه مقدماتی اعداد، نظریه اعداد، ریاضی عمومی ۳	۱۹
شنبه	آمار و احتمال ۲، فنون آماری	۲۰
شنبه	حل عددی معادلات دیفرانسیل جزئی، آنالیز عددی پیشرفته، سمینار	۲۱
چهارشنبه-شنبه ۱۶-۱۴	مبانی ترکیبات (گ ۱)، نظریه گراف	۲۲
شنبه-یکشنبه ۱۶-۱۴	احتمال ۱، آنالیز چند متغیره	۲۳
شنبه-یکشنبه ۱۶-۱۴	نظریه احتمال (گ ۲)، آنالیز حقیقی (گ ۲)، نظریه اندازه	۲۴
شنبه	فلسفه علم، تاریخ ریاضیات، مبانی آنالیز مقدماتی	۲۵
یکشنبه	برنامه‌نویسی پیشرفته	۲۶
شنبه	آنالیز ریاضی ۲، ریاضی ۲ (گ ۱)	۲۷

● روش‌های آماری، آمار و احتمال ۲، آنالیز ریاضی ۲

● جبر ۱، مبانی ترکیبات

● استنباط ۱، فنون آماری

● تحقیق پیشرفته ۲، آنالیز عددی پیشرفته

● استنباط ۲، نظریه‌ی احتمال

تعداد اساتید و دروس اختصاصی و روزهایی که اساتید در دسترس نیستند در جدول ۱.۷.۴ نشان داده شده است. در روز دوشنبه ۱۲-۱۰ جلسه‌ی گروه برگزار می‌شود. بنابراین نباید به این بازه‌ی زمانی درسی اختصاص داده شود. با توجه به داده‌های این دانشکده مشخص می‌شود که تعداد ۷۵ درس باید به ۱۳ کلاس و ۲۰ بازه‌ی زمانی در یک هفته اختصاص داده شوند. دروس ۳ واحدی و ۴ واحدی به ۲ جلسه در طول هفته نیاز دارند و دروس ۲ واحدی به یک جلسه در طول هفته نیاز دارند. از میان این ۷۵ درس فقط یک درس دو واحدی است. بنابراین ۷۴ درس به دو جلسه در هفته و یک درس به یک جلسه در هفته نیاز دارد. پارامترهای $\alpha_1, \alpha_2, \alpha_3$ در این دانشکده با توجه به اولویت‌بندی محدودیت‌های نرم در نظر گرفته شده است که مقادیر آن‌ها عبارت‌اند از:

$$\alpha_1 = 1, \quad \alpha_2 = 5, \quad \alpha_3 = 1$$

با توجه به این که داده‌های این مساله واقعی هستند و تعداد آن‌ها نسبتاً زیاد است، بنابراین مدت زمان مورد نیاز برای به‌دست آوردن جدول زمانبندی بهینه بالا می‌رود. جدول ۱۱.۴ نتایج حاصل از پیاده‌سازی روش جست‌جوی ممنوع را روی این مساله‌ی واقعی نشان می‌دهد. جدول ۱۲.۴ هم، نشان‌دهنده‌ی برنامه‌ی هفتگی به‌دست‌آمده است.

۸.۴ نتیجه‌گیری

با توجه به نتایج به‌دست آمده از پیاده‌سازی سه روش جست‌جوی محلی با انتخاب ساده (روش ۱) و جست‌جوی محلی با انتخاب تصادفی (روش ۲) و جست‌جوی ممنوع (روش ۳) در جدول ۸.۴ و ۱۱.۴، مشخص می‌شود که بطور کلی جواب‌های به‌دست آمده از روش جست‌جوی ممنوع نسبت به دو روش دیگر بهتر است. به‌طور دقیق‌تر مقدار تابع هدف به‌دست آمده از این روش کمتر است و این به معنی آن است که با حفظ شدنی بودن (رعایت محدودیت‌های سخت)، تعداد محدودیت‌های نرم کمتری نقض شده‌اند. همان‌طور که مشاهده می‌شود در بعضی از مسائل این مقدار به صفر رسیده است. این بدین معنی است که تمامی محدودیت‌های سخت و نرم رعایت شده است و این بیانگر کارایی بالای برنامه‌ی موجود است. از میان راهکارهای پیشنهادی

۴. یک مدل غیرخطی برای مسالهی زمانبندی دروس و به کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۶۰

جدول ۱۱.۴: نتایج حاصل از به کارگیری روش جستجوی ممنوع روی داده‌های واقعی

روش	مقدار تابع هدف	زمان (ثانیه)
۱	۷۱	۱۶۵۰
۲	۵۵	۱۷۲۰
۳	۲۰	۳۵۴۰
۴	۹	۳۰۵۰

جهت بهبود بخشیدن به روش‌های فراابتکاری، راهکار تغییر در ساختار همسایگی ساده (روش ۳) موثرتر واقع شد و با ترکیب این راه با راهکار بهبود در جواب اولیه (روش ۴)، رسیدن به جواب بهینه در مدت زمان کمتری محقق می‌شود.

جدول ۱۲.۴: جدول هفتگی به‌دست آمده از پیاده‌سازی روش روی داده‌های دانشکده ریاضی شاهرود

شبه	۸-۱۰	۱۰-۱۲	۲-۴	۴-۶
شبه	استنباط ۱ (ارشد) رمزنگاری (ارشد) مباحثی در نظریه فازی گراف ها (دکتری) ریاضی ۲ گ ۲ سریهای زمانی ۱ آنالیز حقیقی (ارشد) گ ۱ آنالیز تابعی هندسی جبر ۲ جبرهای وفون نیومن دکتری آنالیز عددی ۲ نظریه تقریب (ارشد) مبانی آنالیز ریاضی کنترل بهینه (ارشد)	جبر جامع (ارشد) مبانی ماتریس و جبر خطی گ ۲ تحقیق در عملیات ۲ بهینه سازی مدل های غیر خطی (ارشد) نظریه اعداد مباحثی در فضاها و باناخ (دکتری)	اقتصاد مهندسی مبانی آنالیز عددی آنالیز بهترین تقریب (ارشد) مبانی ترکیبیات گ ۲ جبر خطی عددی ریاضی عمومی ۳ توابع مختلط	آنالیز مختلط (ارشد) نظریه گراف و کاربردهای آن حل عددی معادلات دیفرانسیل معمولی (ارشد) برنامه‌نویسی پیشرفته
۱ شبه	هندسی دیفرانسیل سرتاسری بهینه سازی (ارشد) روشهای نقطه داخلی (دکتری) تحقیق پیشرفته ۲ (ارشد) آنالیز عددی ۱ بهینه سازی خطی ۱ نظریه مقدماتی اعداد احتمال ۱ هندسه منیفلد ۱ (ارشد) نظریه گراف (ارشد)	استنباط ۲ (ارشد) آنالیز تابعی (ارشد) نظریه فلزی (ارشد) جبر جامع (ارشد) هندسه منیفلد ۲ (ارشد) نظریه گروه‌های متناهی زبان تخصصی مبانی ترکیبیات گ ۱ تاریخ ریاضیات آنالیز حقیقی (ارشد) گ ۲ آنالیز ریاضی ۲ سمینار (ارشد)	کنترل بهینه و حساب تغییرات (دکتری) آنالیز ریاضی ۳ روش‌های آماری جبر ۱ جبرهای باناخ دکتری آنالیز عددی پیشرفته ارشد نظریه احتمال (ارشد) گروه و جبر لی ۲ (ارشد) فنون آماری (ارشد) مباحثی در مدل سازی گراف ها (دکتری) نظریه اندازه مبانی آنالیز مقدماتی	آمار و احتمال ۲ نظریه حلقه های ۲ (دکتری)

۴. یک مدل غیرخطی برای مساله‌ی زمانبندی دروس و به‌کارگیری روش جستجوی ممنوع بهبود یافته برای حل آن ۶۲

۲-۴	۱۰-۱۲	۸-۱۰	۴-۶
۲شنبه آنالیز تابعی هندسی مبانی ترکیبیات ۲ تحقیق در عملیات ۲ آنالیز مختلط (ارشد) مبانی ماتریس و جبر خطی ۱ نظریه گراف و کاربردهای آن حل عددی معادلات دیفرانسیل جزئی ارشد نظریه احتمال	۳شنبه جبرهای باناخ (دکتری) احتمال ۱ بهینه سازی مدل های غیرخطی ارشد نظریه حلقه های ۲ دکتری فنون آماری ارشد تاریخ ریاضیات نظریه اعداد آنالیز حقیقی (ارشد) ۲ هندسه منیفلد (ارشد) آنالیز ریاضی ۲ سمینار (ارشد) نظریه گراف (ارشد) کنترل بهینه (ارشد)	۴شنبه هندسی دیفرانسیل سرتاسری حل عددی معادلات دیفرانسیل جزئی ارشد آنالیز چندمتغیره (ارشد) نظریه گروه های خطی (دکتری) ریاضی ۲ نظریه اندازه	۵شنبه اقتصاد مهندسی نظریه گروه های خطی (دکتری) توابع مختلط
۶شنبه کنترل بهینه و حساب تغییرات (دکتری) استنباط ۲ (ارشد) مباحثی در نظریه فازی گراف ها (دکتری) آنالیز عددی ۱ زبان تخصصی جبرهای وفون نیومن دکتری فلسفه علم ریاضی عمومی ۳	۷شنبه استنباط ۱ (ارشد) بهینه سازی (ارشد) آنالیز تابعی (ارشد) سریهای زمانی ۱ روشهای نقطه داخلی (دکتری) آنالیز ریاضی ۳ نظریه مقدماتی اعداد جبر ۲ جبر ۱ حل عددی معادلات دیفرانسیل معمولی (ارشد) آنالیز عددی پیشرفته (ارشد) گروه و جبرلی ۲ (ارشد) مباحثی در مدل سازی گراف ها (دکتری)	۸شنبه مبانی آنالیز عددی نظریه فلزی (ارشد) مبانی ماتریس و جبر خطی ۲ ریاضی ۲ ۲ تحقیق پیشرفته ۲ (ارشد) روش های آماری هندسه منیفلد ۲ (ارشد) بهینه سازی خطی ۱ آمار و احتمال ۲ مبانی ترکیبیات ۱ نظریه احتمال (ارشد) نظریه تقریب (ارشد) مبانی آنالیز مقدماتی	۹شنبه آنالیز تابعی هندسی مبانی ترکیبیات ۲ تحقیق در عملیات ۲ آنالیز مختلط (ارشد) مبانی ماتریس و جبر خطی ۱ نظریه گراف و کاربردهای آن حل عددی معادلات دیفرانسیل جزئی ارشد نظریه احتمال

پیوست کد Matlab

۱.آ بخشی از کدنویسی روش جست وجوی ممنوع

```
tic
MaxIt=110; % Maximum Number of Iterations

TL=100; % Tabu Length

%% Initialization
n= 72;
m= 20;
p= 13;
q= 36;
D=5;
IQ=9;
f=m*p;
s(1)=2;
s(2)=2;
s(3)=2;
s(4)=2;
s(5)=2;
s(6)=2;
```

```
s(7)=2;  
s(8)=2;  
s(9)=2;  
s(10)=2;  
s(11)=2;  
s(12)=2;  
s(13)=2;  
s(14)=2;  
s(15)=2;  
s(16)=2;  
s(17)=2;  
s(18)=2;  
s(19)=2;  
s(20)=2;  
s(21)=2;  
s(22)=2;  
s(23)=2;  
s(24)=2;  
s(25)=2;  
s(26)=2;  
s(27)=2;  
s(28)=2;  
s(29)=2;  
s(30)=2;  
s(31)=2;  
s(32)=2;  
s(33)=2;  
s(34)=2;  
s(35)=2;  
s(36)=2;  
s(37)=2;
```

s(38)=2;

s(39)=2;

s(40)=2;

s(41)=2;

s(42)=2;

s(43)=2;

s(44)=2;

s(45)=2;

s(46)=2;

s(47)=2;

s(48)=2;

s(49)=2;

s(50)=1;

s(51)=2;

s(52)=2;

s(53)=2;

s(54)=2;

s(55)=2;

s(56)=2;

s(57)=2;

s(58)=2;

s(59)=2;

s(60)=2;

s(61)=2;

s(62)=2;

s(63)=2;

s(64)=2;

s(65)=2;

s(66)=2;

s(67)=2;

s(68)=2;

```
s(69)=2;
```

```
s(70)=2;
```

```
s(71)=2;
```

```
s(72)=2;
```

```
N{1}=[26 30 59 36 51 37 44 64 41 55 47 56 68 50 53 69 60 67 ];
```

```
N{2}=[26 30 59 36 51 37 44 64 41 55 47 56 68 50 53 69 60 67 20 29 66 2 9];
```

```
N{3}=[26 30 59 36 51 37 44 64 41 55 47 56 68 50 53 69 60 67 70 29 54 58 9];
```

```
N{4}=[26 30 59 36 51 37 44 64 41 55 47 56 68 50 53 69 60 67 5 46 62 23 20];
```

```
N{5}=[5 21 18 24 65 71 20 29 66 52];
```

```
N{6}=[5 21 18 24 65 71 20 29 66 52 2 9];
```

```
N{7}=[5 21 18 24 65 71 20 29 66 52 16 43];
```

```
N{8}=[5 21 18 24 65 71 20 29 66 52];
```

```
N{9}=[33 54 61];
```

```
N{10}=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 28 29 30 31 32 33 34 35  
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66  
67 68 69 70 71 72];
```

```
N{11}=[33 54 61];
```

```
N{12}=[33 54 61];
```

```
N{13}=[1];
```

```
N{14}=[1 5 21];
```

```
N{15}=[1];
```

```
N{16}=[1];
```

```
N{17}=[2 9 63 3 45 4 15 22 6 11 57 33 7 14 19 27 8 23 38 62 10 39 46 12 34 49 13 35 42 16 43 17 72  
25 58 48 28 31 40 70 ];
```

```
N{18}=[2 9 63 3 45 4 15 22 6 11 57 33 7 14 19 27 8 23 38 62 10 39 46 12 34 49 13 35 42 16 43 17 72  
25 58 48 28 31 40 70 ];
```

```
N{19}=[2 9 63 3 45 4 15 22 6 11 57 33 7 14 19 27 8 23 38 62 10 39 46 12 34 49 13 35 42 16 43 17 72  
25 58 48 28 31 40 70 ];
```

```
N{20}=[2 9 63 3 45 4 15 22 6 11 57 33 7 14 19 27 8 23 38 62 10 39 46 12 34 49 13 35 42 16 43 17 72
```

25 58 48 28 31 40 70] ;

$L\{1\}=[1];$

$L\{2\}=[2 \ 9 \ 63];$

$L\{3\}=[3 \ 45];$

$L\{4\}=[4 \ 15 \ 22];$

$L\{5\}=[5 \ 21];$

$L\{6\}=[6 \ 11 \ 57];$

$L\{7\}=[7 \ 14 \ 19 \ 27];$

$L\{8\}=[8 \ 23 \ 38 \ 62];$

$L\{9\}=[10 \ 39 \ 46];$

$L\{10\}=[12 \ 34 \ 49];$

$L\{11\}=[13 \ 35 \ 42];$

$L\{12\}=[16 \ 43];$

$L\{13\}=[17 \ 72];$

$L\{14\}=[18 \ 24];$

$L\{15\}=[20 \ 29 \ 66];$

$L\{16\}=[25 \ 58 \ 48];$

$L\{17\}=[26 \ 30 \ 59];$

$L\{18\}=[28 \ 31];$

$L\{19\}=[33 \ 54 \ 61];$

$L\{20\}=[36 \ 51];$

$L\{21\}=[37 \ 44 \ 64];$

$L\{22\}=[40 \ 70];$

$L\{23\}=[41 \ 55];$

$L\{24\}=[47 \ 56 \ 68];$

$L\{25\}=[50 \ 53 \ 69];$

$L\{26\}=[52];$

$L\{27\}=[60 \ 67];$

$L\{28\}=[7 \ 50 \ 15 \ 32];$

$L\{29\}=[27 \ 19];$

```
L{30}=[66 28 62];  
L{31}=[36 23 60];  
L{32}=[36 41];  
L{33}=[61 52];  
L{34}=[3 51];  
L{35}=[4 45];  
L{36}=[44 17];
```

```
cr{1}=[7 50 15 32];  
cr{2}=[27 19];  
cr{3}=[66 28 62];  
cr{4}=[36 23 60];  
cr{5}=[36 41];  
cr{6}=[61 52];  
cr{7}=[3 51];  
cr{8}=[4 45];  
cr{9}=[44 17];
```

```
% Create Empty Individual Structure
```

```
empty_individual.Position=[];  
empty_individual.Cost=[];  
empty_individual.ActionIndex=[];
```

```
% Create Initial Solution
```

```
soli.Position=FeasibleSolution(n,m,p,q,D,s,N,L)  
soli.Position
```

```

sol_i.Cost=MyCost(sol_i.Position,n,m,D,p,f,s,cr,IQ)
    sol.Position= goodsol(sol_i,n,m,f,p,s,N,D,cr,q,IQ)
    sol.Cost=MyCost(sol.Position,n,m,D,p,f,s,cr,IQ)
%solposition=FeasibleSolution(n,m,p,q,D,s,N,L)
%solCost= MyCost(solposition,n,m,D,p,f,s,cr,std)

```

```

% Initialize Best Solution Ever Found

```

```

BestSol=sol;

```

```

% Array to Hold Best Costs

```

```

BestCost=zeros(MaxIt,1);

```

```

% Initialize Action Tabu Counters

```

```

BestCost=zeros(MaxIt,1);

```

```

% Initialize Action Tabu Counters

```

```

F{1}=[0];

```

```

AQ=1;

```

```

TC=zeros( 200,1);

```

```

%% Tabu Search Main Loop

```

```

for it=1:MaxIt

```

```

    coo=0;

```

```

    bestnewsol.Cost=inf;

```

```

    SWW=0;

```

```

    MS=0;

```

```

[A,B,sh]=findemptyperiod(sol.Position,m,p,n);

```

```

ActionList=CreatePermActionList(sol.Position,N,n,f,p,L,q,s,A,B,sh);

nAction=numel(ActionList)

% Apply Actions
for i=1:nAction
    RA=100;
    NB=0;

    for j=1:AQ
        if numel(F{j})==numel(ActionList{i})
            if numel(F{j})==4
                if F{j}==ActionList{i}
                    RA=50;
                    NB=1;
                    if TC(j)==0
                        RA=1;
                        break
                    end
                end
            end
        else
            if F{j}(2)==ActionList{i}(2)&F{j}(4)==ActionList{i}(4)&((F{j}(3)==ActionList{i}(3)
                &F{j}(4)==ActionList{i}(4))||(F{j}(3)==ActionList{i}(5)& F{j}(5)==ActionList{i}(3)))

                RA=50;
                NB=1;
                if TC(j)==0
                    RA=1;
                    break
                end
            end
        end
    end
end
end

```


end

if NB==1

break

end

end

end

SWW=1;

newsol.Position=DoAction(sol.Position,ActionList{i});

newsol.Cost=MyCost(newsol.Position,n,m,D,p,f,s,cr,IQ);

newsol.ActionIndex=ActionList{i};

if RA==100 || RA==1 || (RA==50 & newsol.Cost < BestSol.Cost)

coo=coo+1;

if newsol.Cost<bestnewsol.Cost

MS=1;

bestnewsol.Position=newsol.Position;

bestnewsol.Cost=newsol.Cost;

bestnewsol.ActionIndex=newsol.ActionIndex;

end

end

end

% Update Current Solution

if SWW==0

disp('no')

```

        break
    else
        Z0=0;
        sol=bestnewsol;
        for j=1:AQ
            if numel(F{j})==numel(bestnewsol.ActionIndex)
                if F{j}==bestnewsol.ActionIndex
                    Z0=1;
                    break
                end
            end
        end
        if Z0==0
            AQ=AQ+1;
            F{AQ}=bestnewsol.ActionIndex;
        end

% Update Tabu List

for i=1:AQ
    if numel(F{i})==numel(bestnewsol.ActionIndex)
        if F{i}==bestnewsol.ActionIndex

            TC(i)=TL;                % Add To Tabu List
        else

            TC(i)=max(TC(i)-1,0);
        end                        % Reduce Tabu Counter
    end
end
end

```

```
% Update Best Solution Ever Found
if sol.Cost<BestSol.Cost
    BestSol=sol;
end

% Save Best Cost Ever Found
BestCost(it)=BestSol.Cost;

% Show Iteration Information
disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);

% If Global Minimum is Reached
if BestCost(it)==0
    break;
end

end

end

x=1:MaxIt
y=BestCost(x)
axis normal
plot(x,y)
BestCost=BestCost(1:it);

%% Results

disp(' ');
```

```
disp('Best Solution:');
x=BestSol.Position
COST=BestSol.Cost
for e=1:m
    ZM{e}=[];
end
kh=1;
for k=1:m
    for j=kh:k*p
        for i=1:n
            if x(i,j)==1
                ZM{k}=[ZM{k},i];
                break
            end
        end
    end
    kh=kh+13;

end
for b=1:m
    M=ZM{b}
end

toc
```

مراجع

- [1] Aladag, C. H. Hocaoglu, G. and Basaran, M. A. *The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem*, Expert Systems with Applications, 36 (2009) 12349-12356.
- [2] Burke, E. K. and Petrovice, S. *Recent research direction in automated timetabling*, European Journal of Operational Research, 140 (2002) 266-280.
- [3] Carrasco, M. P. and Pato, M. V. *A multiobjective genetic algorithm for the class/ teacher timetabling problem*, Lecture Notes in Computer Science, 2079 (2001) 3-17.
- [4] Deris, S. Omatu, S. and Hiroshi, A. *Timetable planning using the constraint-based reasoning*, Computers and Operations Research, 27 (2000), 819-840.
- [5] Dimopoulou, M. and Miliotis, P. *Implementation of a university course and examination timetabling system*, European Journal of Operational Research, 130 (2001) 202-213.
- [6] Glover, F. *Future paths for integer programming and links to artificial intelligence*, Computers and Operations Research, 13(1986) 533-549.
- [7] Grigorios, N. *Likothanassisa applying evolutionary computation to the school timetabling problem*, Computers and Operations Research, 35 (2008) 1265-1280.
- [8] Kustoch, P. A. *Timetabling competition sa-based heuristics*, International Timetabling Competition Results, 45 (2003) 124-130.
- [9] Luis, E. *A hybrid grouping genetic algorithm for assigning students to preferred*, Expert Systems with Applications, 36 (2009) 7234-7241
- [10] Carter, M. *A survey of practical applications of examination timetabling algorithms*, Operations Research, 34 (1986) 193-202.
- [11] Carter, M. *A lograngion relaxation approach to the classroom assignment*, INFOR, 27 (1986) 230-246.
- [12] MirHassani, S. A. *A computational approach to enhancing course timetabling with integer programming*, Appl. Math. Comput, 175 (2006), 814-822.
- [13] Pongcharoena, P. *Stochastic Optimization Timetabling Tool for university course scheduling*, Int.J.Production Economics, 112 (2008) 903-918.
- [14] Alvarez-Valdes, R. Martin, G. Tamarit, J. M. *Constructing good solution for Spanish school timetabling problem*, Operation Research Society, 47 (1996) 1203-1215.
- [15] Abdullah, S. Turabieh, H. *On the use of multi neighbourhood structure within a tabu-based memetic approach to university timetabling problems*, Information Sciences, 91 (2012) 146-168.
- [16] Lu, Z. and Kao Hao, J. *Adaptive tabu Search for course timetabling*, European Journal of Research, 200 (2010) 235-244.

Surname: Rasuli

Name: Zohre

Title: The Course Timetabling

Supervisor: Dr.Jafar Fathali

Advisor: Dr.Mohsen Asili

Degree: Master of Science

Subject: Applied Mathematics

Field: Operational Research

Technology University of Shahrood

Faculty Of Mathematical Sciences

Date: 92/9/30

Number of pages: 77

Keywords: 1-Timetabling 2- Tabu Search 3-Heuristic 4-Neighbourhood

Abstract

Abstract Here The problem of lessons scheduling is one of the most significant and practical issues which has been done in all educational institutes, annually or twice in an educational year. In the past, timetabling was provided manually; however, still now this is common in a great number of educational institutes. Writing the timetabling manually, needs a lot of time. In addition, the solution may be a possible answer but, not the best one. In this thesis, a model of solving the lessons scheduling problem has mentioned. To optimize the timetabling and to reply needs of professors and students, hard and soft limitation of this model is more than the previous ones. Three methods of local search, stochastic local search and, tabu search have been used to solve this model. Some solutions have been suggested in order to optimize these approaches. Finally for proving the practicality of this program which is written with Matlab, it has been implemented on data related to mathematics department of Shahroud technical university



Technology University of Shahrood
Faculty Of Mathematical Sciences

Dissertation Submitted in Partial
Fulfillment of The Requirements For The
Degree of Master of Science in
Applied Mathematics

The Course Timetabling

Supervisor

Dr.Jafar Fathali

Advisor

Dr.Mohsen Asili

by

Zohre Rasuli

92/9/30