دانشکده ریاضی و علوم کامپیوتر

پایان‌نامه برای دریافت درجه کارشناسی ارشد در رشته علوم تصمیم و مهندسی دانش گرایش

یک رویکرد مبتنی بر بهینه‌سازی چند هدفه برای مسئله زمان‌بندی دروس دانشگاهی

استادان راهنما

دکتر محمود امین طوسی و دکتر مرتضی جعفرزاده

استاد مشاور

دکتر علیرضا قدسی

پژوهشگر:

عیسی رسائی زاده

بهمن ۱۳۹۸

باسمه تعالی شماره:

صورتجلسه دفاع از پایان‌نامه کارشناسی ارشد تاریخ:

با تلاوت آیاتی چند ازکلام الله مجید جلسه دفاع از پایان‌نامه آقای /خانم **عیسی رسائی زاده** دانشجوی رشته علوم تصمیم و مهندسی دانش به شماره دانشجویی ۹۶۱۳۱۳۷۰۱۳ با عنوان:

یک رویکرد مبتنی بر بهینه‌سازی چند هدفه برای مسئله زمان‌بندی دروس دانشگاهی

در ساعت مورخه در محل دانشکده ریاضی و علوم کامپیوتر تشکیل گردید . پس از استماع گزارش ارائه شده توسط دانشجو و استاد راهنما، هیات داوران و حاضران سئوالاتی را مطرح و آقای / خانم عیسی رسائی زاده به دفاع از موضوع پرداخت وبه سئوالات آنها پاسخ گفت. سپس پایان‌نامه توسط هیات داوران مورد ارزشیابی قرار گرفت و نمره برابر درجه برای آن تعیین گردید .

به‌این ترتیب ضمن تصویب پایان‌نامه مزبور از این تاریخ آقای/ خانم عیسی رسائی زاده به عنوان کارشناس ارشد در رشته علوم تصمیم و مهندسی دانش شناخته می‌شود .

ردیف	نام و نام خانوادگی	سمت	امضـــــــــــا
۱	دکتر محمود امین طوسی	استاد راهنما	
۲	دکتر مرتضی جعفرزاده	استاد راهنما	
۳	دکتر علیرضا قدسی	استاد مشاور	
۴	دکتر مهدی زعفرانیه	استاد داور	
۵	دکتر یاسر علیزاده ثانی	نماینده تحصیلات تکمیلی	

مدیر گروه: دکتر محمدعلی پرتانیان

رونوشت:

۱. معاون محترم آموزشی و تحصیلات تکمیلی دانشگاه جهت اطلاع
۲. معاون محترم پژوهشی دانشگاه جهت اطلاع
۳. آموزش دانشکده ریاضی و علوم کامپیوتر جهت درج در پرونده دانشجو

سوگند نامه دانش آموختگان دانشگاه حکیم سبزواری

به نام خداوند جان و خرد کزین برتر اندیشه بر نگذرد

اینک که به خواست آفریدگار پاک، کوشش خویش و بهره گیری از دانش استادان و سرمایه‌های مادی و معنوی این مرز و بوم، توشه‌ای از دانش و خرد گردآورده‌ام، در پیشگاه خداوند بزرگ سوگند یاد می‌کنم که در به کارگیری دانش خویش، همواره بر راه راست ودرست گام بردارم. خداوند بزرگ، شما شاهدان، دانشجویان و دیگر حاضران را به عنوان داورانی امین گواه می گیرم که ازهمه دانش و توان خود برای گسترش مرزهای دانش بهره گیرم و از هیچ کوششی برای تبدیل جهان به جایی بهتر برای زیستن، دریغ نورزم. پیمان می‌بندم که همواره کرامت انسانی را در نظر داشته باشم و همنوعان خود را در هر زمان و مکان تا سر حد امکان یاری دهم. سوگند می خورم که در به کارگیری دانش خویش به کاری که با راه و رسم انسانی، آیین پرهیزگاری، شرافت و اصول اخلاقی برخاسته از ادیان بزرگ الهی، به ویژه دین مبین اسلام، مباینت دارد دست نیازم. همچنین در سایه اصول جهان شمول انسانی و اسلامی، پیمان می‌بندم از هیچ کوششی برای آبادانی و سرافرازی میهن و هم میهنانم فروگذاری نکنم و خداوند بزرگ را به یاری طلبم تا همواره در پیشگاه او و در برابر وجدان بیدار خویش و ملت سرافراز، بر این پیمان تا ابد استوار بمانم.

نام و نام خانوادگی: عیسی رسائی زاده

تاریخ و امضا:

تأییدیه‌ی صحت و اصالت نتایج

باسمه تعالی

اینجانب عیسی رسائی زاده به شماره دانشجویی ۹۶۱۳۱۳۷۰۱۳ دانشجوی رشته علوم تصمیم و مهندسی دانش مقطع تحصیلی کارشناسی ارشد تأیید می‌نمایم که کلیه‌ی نتایج این پایان‌نامه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. درصورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسؤولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح (اعم از اداری و قضایی) به عهده ی اینجانب خواهد بود و دانشگاه هیچ گونه مسؤولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی: عیسی رسائی زاده

تاریخ و امضا:

مجوز بهره برداری از پایان‌نامه

بهره برداری از این پایان‌نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می‌شود، بلامانع است:

☐ بهره برداری از این پایان‌نامه برای همگان بلامانع است.

☐ بهره برداری از این پایان‌نامه با اخذ مجوز از استاد راهنما، بلامانع است.

☐ بهره برداری از این پایان‌نامه تا تاریخ ..... ممنوع است.

استادان راهنما: دکتر محمود امین طوسی

دکتر مرتضی جعفرزاده

تاریخ و امضا:

: تقدیم به

همسر

و

پدر و مادرم

سپاس خداوندگار حکیم را که با لطف بی کران خود، آدمی را زیور عقل آراست.

در آغاز وظیفه خود می دانم از زحمات بی دریغ استادان راهنمای خود، جناب آقای دکتر محمود امین طوسی و جناب آقای دکتر مرتضی جعفرزاده، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی‌های ارزنده ایشان، این مجموعه به انجام نمی رسید.

از جناب آقای دکتر علیرضا قدسی که زحمت مطالعه و مشاوره این رساله را تقبل فرمودند و در آماده‌سازی این رساله، به نحو احسن اینجانب را مورد راهنمایی قرار دادند، کمال امتنان را دارم.

از جناب آقای دکتر جلال دلقندی و جناب آقای دکتر مسعود خزاعی فدافن ریاست محترم دانشکده فنی امام خمینی(ره) سبزوار تشکر می‌کنم.

در پایان از خانواده عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، بهترین پشتیبان من بودند، تشکر می‌کنم.

عیسی رسائی زاده

بهمن ۱۳۹۸

فهرست مطالب

فهرست تصاویر ج

فهرست جداول ه

چکیده ۱

پیش‌گفتار ۲

فصل ۱: مروری بر پیشینه موضوع و پیش‌نیازها ۴

۱-۱ ادبیات موضوع ۴

۱-۲ آشنایی با روش حریصانه ۶

۱-۳ خوشه‌بندی ۸

۱-۳-۱ الگوریتم خوشه‌بندی K-میانگین ۸

فصل ۲: یک رویکرد مبتنی بر بهینه‌سازی چند هدفه برای مسئله زمان‌بندی دروس دانشگاهی ۱۴

۲-۱ مسائل عملیاتی، تاکتیکی و استراتژیک در زمان‌بندی دروس دانشگاهی ۱۴

۲-۲ بیان مسئله زمان‌بندی دانشگاهی ۱۷

۲-۲-۱ مدل سازی مسئله ۱۸

۲-۲-۱-۱ مدل کیفیت ۱۹

۲-۲-۱-۲ مدل برنامه‌ریزی اتاق‌ها ۲۱

۲-۲-۱-۳ مدل بازه‌های تدریس ۲۳

۲-۲-۱-۴ مدل برنامه‌ریزی اتاق‌ها در مقابل مدل کیفیت ۲۴

۲-۲-۱-۵ مدل بازه‌های تدریس در مقابل مدل کیفیت ۲۴

۲-۲-۱-۶ مدل بازه‌های تدریس در مقابل مدل برنامه‌ریزی اتاق‌ها ۲۵

۲-۲-۲ روش اپسیلون-محدودیت ۲۶
۲-۲-۳ روش حل ۲۶

فصل ۳: حل مسئله زمان‌بندی امتحانات با استفاده از روش ابتکاری و حریصانه ۲۸
۳-۱ مقدمه ۲۸
۳-۲ محدودیت‌های سخت در مسئله زمان‌بندی امتحانات ۳۰
۳-۳ محدودیت نرم در مسئله زمان‌بندی امتحانات ۳۰
۳-۴ محدودیت‌های لحاظ شده ۳۰
۳-۴-۱ پارامترهای مسئله زمان‌بندی امتحانات ۳۱
۳-۵ الگوریتم روش پیشنهادی ۳۴
۳-۵-۱ الگوریتم تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه ۳۵
۳-۵-۲ الگوریتم تولید گروه‌بندی اولیه دروس ۳۶
۳-۵-۳ الگوریتم تولید برنامه نهایی امتحانات ۳۸
۳-۵-۴ مثال کاربردی الگورتیم‌های ارائه شده در روش ابتکاری ۴۰
۳-۶ پژوهش‌های آتی ۴۵
۳-۶-۱ آشنایی با قوانین انجمنی ۴۶
۳-۷ مثال عملی برنامه ۴۶

فهرست منابع ۵۳

پیوست آ: کد برنامه زمان‌بندی امتحانات با استفاده از روش ابتکاری و حریصانه ۵۵

واژه‌نامه فارسی به انگلیسی ۷۶

واژه‌نامه انگلیسی به فارسی ۷۷

فهرست تصاویر

۱-۱ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : اولین مرحله اجرا ۹

۱-۲ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : دومین مرحله اجرا ۱۰

۱-۳ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : سومین مرحله اجرا ۱۰

۱-۴ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : چهارمین مرحله اجرا ۱۱

۱-۵ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : پنجمین مرحله اجرا ۱۱

۱-۶ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : ششمین مرحله اجرا ۱۲

۱-۷ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : هفتمین مرحله اجرا ۱۲

۱-۸ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : هشتمین مرحله اجرا ۱۳

۱-۹ خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : نهمین مرحله اجرا ۱۳

۱-۲ نمودار تغییرات در سطوح مختلف مسئله زمان‌بندی ۱۵

۳-۱ نمونه‌ای از گراف ماتریس مجاورت شامل ۱۰ درس در مثال ۳-۵-۱ . . . . . ۴۵

۳-۲ فلوچارت روش قوانین انجمنی (روش Apriori Association Rule Mining) ۴۷

۳-۳ گراف انتخاب واحدها در مثال ۳-۷-۱ . مجموعه C، دروس و مجموعه S، دانشجویان هستند. یال‌های جهت دار از دانشجویان به دروس، نشان‌دهنده این است دانشجو درس مربوطه را انتخاب واحد نموده‌است ۴۸

۳-۴ گراف ماتریس مجاورت در مثال ۳-۷-۱. راس‌های همجوار، نشان‌دهنده این است این دو راس به طور هم زمان توسط دانشجو انتخاب واحد شده است. ۴۸
۳-۵ گراف گروه‌بندی اولیه دروس در مثال ۳-۷-۱ . . . . . ۴۹

فهرست جداول

۱-۱ مقایسه نتایج حل دو مثال خرد کردن پول با روش حریصانه با سکه‌های ۵۰، ۲۵، ۱۲، ۱۰، ۵ و یک تومانی ۷

۱-۲ نمادگذاری‌های مسئله زمان‌بندی دروس دانشگاهی ۱۸

۲-۲ نمونه اندیس‌های انتخاب شده ۲۳

۱-۳ واژه‌ها و اصطلاحات الگوریتم ۳۱

۲-۳ نقض محدودیت‌ها ۳۲

۳-۳ پارامترهای مسئله زمان‌بندی امتحانات ۳۳

۴-۳ لیست درس‌ها در مثال (۳-۵-۱) ۴۱

۵-۳ گروه‌بندی اولیه دروس در مثال (۳-۵-۱) با استفاده از الگوریتم (۳-۲) ۴۲

۶-۳ برنامه نهایی امتحانات در مثال (۳-۵-۱) با استفاده از الگوریتم (۳-۳) ۴۳

۷-۳ بخشی از تداخل برنامه امتحانات در دو نوبت امتحانی یک روز در مثال (۳-۵-۱) ۴۴

۸-۳ گروه‌بندی اولیه دروس در مثال ۳-۷-۱ . مجموعه C، دروس و مجموعه S، دانشجویان هستند. ۴۹

۹-۳ میزان تداخل گروه‌ها از گروه‌بندی اولیه دروس در مثال ۳-۷-۱ . . . . . ۴۹

۱۰-۳ نمونه برنامه امتحانی دروس **در حالت۱** در مثال ۳-۷-۱ . مجموعه C دروس، مجموعه S دانشجویان و G گروه‌بندی اولیه دروس هستند. ۵۰

۱۱-۳ نمونه برنامه امتحانی دروس **در حالت۲** در مثال ۳-۷-۱ . مجموعه C دروس، مجموعه S دانشجویان و G گروه‌بندی اولیه دروس هستند. ۵۱

۱۲-۳ نمونه برنامه امتحانی دروس **در حالت۳** در مثال ۳-۷-۱ . مجموعه C دروس، مجموعه S دانشجویان و G گروه‌بندی اولیه دروس هستند. ۵۱

۳-۱۳ نمونه برنامه امتحانی دروس **در حالت۴** در مثال ۳-۷-۱ . مجموعه C دروس، مجموعه S دانشجویان و G گروه‌بندی اولیه دروس هستند. ۵۲

آ-۱ متغیرهای مورد استفاده در کد برنامه زمان‌بندی امتحانات با استفاده از روش ابتکاری و حریصانه۵۶

دانشگاه حکیم سبزواری

فرم چکیده ی پایان‌نامه ی دوره ی تحصیلات تکمیلی

مدیریت تحصیلات تکمیلی

نام خانوادگی دانشجو:رسائی زاده	نام:عیسی	ش. دانشجویی:۹۶۱۳۱۳۷۰۱۳
استادان راهنما: دکتر محمود امین طوسی و دکتر مرتضی جعفرزاده		
استاد مشاور: دکتر علیرضا قدسی		
دانشکده ریاضی و علوم کامپیوتر	رشته: علوم تصمیم و مهندسی دانش	گرایش:
مقطع: کارشناسی ارشد	تاریخ دفاع:بهمن ۱۳۹۸	تعداد صفحات: ۷۸
عنوان پایان‌نامه: یک رویکرد مبتنی بر بهینه‌سازی چند هدفه برای مسئله زمان‌بندی دروس دانشگاهی		
کلید واژه‌ها:جدول زمان‌بندی،برنامه‌ریزی چند هدفه، برنامه‌ریزی عدد صحیح		

چکیده: موضوع برنامه زمان‌بندی یکی از مباحث مهمی است که در زمینه‌های مختلفی از جمله برنامه درسی، امتحانات درسی، حمل و نقل اتوبوس، شیفت کارکنان ادارات، شیفت کارگران کارخانه و شبکه ریلی و قطار مورد استفاده قرار می‌گیرد. در این پایان نامه، یک رویکرد مبتنی بر بهینه‌سازی چند هدفی برای مسئله برنامه‌ریزی دروس دانشگاه با استفاده از برنامه‌نویسی عدد صحیح مختلط ارائه شده است. برنامه درسی عبارت است از اختصاص جلسات کلاس به اتاق‌ها و استادان به طوری که برخی از محدودیت‌ها برآورده شوند. محدودیت‌ها به دو دسته سخت و نرم تقسیم می‌شوند. رعایت محدودیت‌های سخت الزامی و عدم رعایت محدودیت‌های نرم باعث کاهش کیفیت و کارایی برنامه نهایی و در نتیجه رضایت استادان و دانشجویان می‌شود. در مسئله برنامه‌ریزی درسی، هدف این است که جلسه‌های کلاس با حداکثر پوشش به منافع استادان و دانشجویان در خصوص بازه‌زمانی، برنامه زمانی منسجم و جمع و جور اختصاص یابد.

پیش‌گفتار

مسئله زمان‌بندی و تخصیص، در دنیای امروز با توجه به محدودیت‌های منابع مورد نیاز و تعدد درخواست‌های مطرح شده از طرف ذینفعان یکی از مسائل مهم تحقیقاتی و کاربردی است که به طور سنتی به عنوان یک مسئله عملیاتی مورد مطالعه قرار گرفته است. این مسئله در زمینه‌های متفاوت مانند زمان‌بندی برنامه حرکت قطارها و هواپیماها، تخصیص شیفت کاری به کارکنان، زمان‌بندی درس‌ها، زمان‌بندی امتحان‌ها،حرکت اتوبوس و حمل و نقل و موارد مشابه کاربرد دارد.

مسئله تخصیص از جمله مسائلی است که از دیرباز مورد توجه پژوهشگران بوده و با روش‌ها و الگوریتم‌های مختلف حل شده است ولی بهینه نبودن آن‌ها به طور مطلق انکار ناپذیر است و هنوز هم تلاش می‌شود مسئله را به روش‌های جدیدتر و بهینه‌تر و یا به‌روزآوری الگوریتم‌های ارائه شده قبلی،حل نمایند. از جمله این الگوریتم‌ها می‌توان به الگوریتم ژنتیک، رنگ آمیزی گراف،جستجوی ممنوع،اجتماع مورچگان اشاره نمود. انواع مختلف مسئله زمان‌بندی از نظر پیچیدگی محاسباتی به کلاس مسائل $NP-hard$ تعلق دارند. برای حل این مسائل، باید درس‌ها یا کارکنان را با توجه به محدودیت‌های سخت و نرم به بازه‌های زمانی نسبت داد. برای محدودیت‌های سخت امکان تخطی از آن وجود ندارد و باید حتماً رعایت شوند ولی برای محدودیت‌های نرم یک تابع جریمه در نظر می‌گیریم که هدف کمینه کردن این جریمه‌هاست که وابستگی به روش استفاده شده ندارد و رعایت هرچه بیشتر محدودیت‌های نرم جزء اهداف است.

کارتر و لاپورته تعریفی از زمان‌بندی درس‌ها را ارائه نمود." مسئله چند بعدی که در آن دانشجویان، معلمان (یا اعضای هیئت علمی) به دوره‌ها، بخش‌ها یا کلاس‌ها اختصاص می‌یابند. رویدادها (جلسه‌های فردی بین دانش آموزان و معلمان) به کلاس‌ها و زمان‌ها اختصاص می‌یابد " [۱].

در مسئله زمان‌بندی درس‌ها، هدف اختصاص جلسه‌ها به اتاق‌ها و زمان‌بندی آن‌ها و ایجاد جدول زمانی با حداکثر پوشش خواسته‌های مورد علاقه استادان و دانشجویان از لحاظ برنامه کلاسی منسجم است. پیش از این می‌توان دو مسئله مهم دیگر را حل کرد: چه اتاق‌هایی برای برگزاری جلسه‌های درس لازم داریم و در چه بازه‌های تدریس؟ این تصمیمات ممکن است تأثیر زیادی بر جدول زمانی حاصل داشته باشد. روش‌های سنتی

تولید جدول زمانی، ممکن است راه حل‌های خوبی که بتوانند به راحتی با تغییر ورودی و شرایط سازگار شوند را تولید نکنند. تغییرات دیررس با ایجاد حداقل تغییرات، نیاز مهمی در بسیاری از کاربردهای عملی زمان‌بندی است.سازمان‌ها به سمت استفاده بهینه از منابع(فضای آموزشی، زمان، استادان و ...) و کسب رضایت‌مندی در نتیجه اجرای آن برنامه‌ریزی، حرکت می‌کنند.

در این پایان نامه تلاش شده است مسئله را با تابع چند هدفه با استفاده از برنامه‌ریزی عدد صحیح حل نماید.

در فصل اول این پایان نامه به مرور ادبیات موضوع و برخی پیش نیازهای لازم پرداخته و سپس در فصل دوم یک رویکرد مبتنی بر بهینه‌سازی چند هدفه برای مسئله زمان‌بندی دروس دانشگاهی داریم. در پایان و در فصل سوم مسئله زمان‌بندی امتحانات را با استفاده از روش ابتکاری و حریصانه حل نموده و نتایج ان را بر روی مثالی کاربردی مورد بررسی قرار می‌دهیم. مراجع اصلی که در نگارش این پایان نامه مورد استفاده قرار گرفته‌اند به شرح زیر هستند :

1- Lindahl M, Mason AJ, Stidsen T, Sørensen M. A strategic view of University timetabling. European Journal of Operational Research. 2018 Apr 1;266(1):35-45.

2- Babaei H, Karimpour J, Hadidi A. A survey of approaches for university course timetabling problem. Computers Industrial Engineering. 2015 Aug 1;86:43-59.

فصل ۱

مروری بر پیشینه موضوع و پیش‌نیازها

۱-۱ ادبیات موضوع

پیرامون مسئله زمان‌بندی، پژوهش‌های گوناگونی صورت پذیرفته است که اهمیت مسئله و کاربردهای متعدد آن را نشان می‌دهد. سویتا [۲] و همکاران یک سیستم عصبی مورچه و یک مدل دو هدفه را برای حل مسئله زمان‌بندی درس‌های دانشگاهی ارائه دادند [۲]. امین طوسی و حدادنیا روشی مبتنی بر منطق فازی برای جداسازی دانشجویان کلاس در برنامه‌ریزی کلاسی را بررسی نمودند [۳]. چن [۳] و شیح [۴] دو نوع متفاوت از الگوریتم بهینه‌سازی ازدحام ذرات را بر روی این مسئله به کار بردند [۴]. نورمی و کینگسیک الگوریتم تکاملی را به منظور حل مسئله برنامه‌ریزی دوره‌های آموزشی ارائه دادند [۵]. سوگانتو [۵] یک الگوریتم ژنتیک برای زمان‌بندی دوره آموزشی دانشگاه ارائه داد که با استفاده از یک الگوریتم حریصانه، جواب‌های شدنی که همه محدودیت‌های سخت را ارضاء می‌کند، را به دست می‌آورد. [۶]. سسچیا [۶] و همکارانش مسئله زمان‌بندی دوره درس مبتنی بر ثبت‌نام را با استفاده از الگوریتم تبرید شبیه‌سازی‌شده حل نمودند [۷]. دسکالاکی [۷] و بیرباس [۸] با استفاده از روش برنامه‌ریزی صفر و یک، جدول زمان‌بندی درس‌ها را توسعه دادند [۸]. هدف آن‌ها، به حداقل رساندن یک تابع هزینه خطی بود. با این هدف می‌توان رضایت ترجیحات بیان شده در مورد دوره‌های آموزشی یا روزهای هفته یا حتی کلاس‌های درس در دوره‌های مشخص را مدنظر قرار داد. علاوه بر این، با تعریف مناسب از ضرایب هزینه در تابع هدف، می‌توان فضای جواب را کاهش داد. آولا و واسیل الگوریتم شاخه و کران را برای مسئله مربوط به زمان‌بندی دروس دانشگاهی به کار بردند [۹].

آلیوکوب [۹] و شرالیبا [۱۰] با بهره‌گیری از فرآیند آموزشی دانشگاه کویت،یک مدل برنامه‌ریزی عدد صحیح

---

[۲] Socha [۳] Chen [۴] Shih [۵] Suyanto [۶] ceschia [۷] Daskalaki [۸] Birbas
[۹] Al-Yakoob [۱۰] Sherali

مختلط را برای مسئله جدول زمان‌بندی درس‌ها ارائه دادند که قادر است زمان‌بندی کلاس‌ها را که دارای تعدادی از ویژگی‌های مورد نظر در ارتباط با به حداقل رساندن کلاس‌ها، ارائه الگوهای خوب و افزایش جریان ترافیك باشد، طراحی کند [۱۰].

علاوه بر این، در هنگام ایجاد بخش‌هایی از دوره‌ها، سیاست‌های جنسیتی را مورد توجه قرار داده‌اند. این مدل با وظایف ترکیبی و دشوار مانند حل و فصل اختلافات کلاس، برخورد با پارکینگ و ترافیک و اطمینان از بهره‌وری کارآمد از امکانات و منابع انسانی همراه است. نوتیگر [۱] و همکارانش یک مدل با استفاده از الگوریتم اجتماع مورچگان ارائه دادند که از دو ماتریس متمایز و ساده به منظور بهبود همگرایی استفاده می‌کند [۱۱].

وانگ [۲] و همکارانش، یک مدل چند هدفه بهینه‌سازی برای فواصل عزیمت همزمان برای چندین خط اتوبوس توسعه دادند که از یک الگوریتم ژنتیک استفاده کردند. روش پیشنهادی آن‌ها شامل سه هدف بود: هدف اول به حداکثر رساندن سود عملکرد اتوبوس بود. هدف دوم به حداقل رساندن زمان انتظار مسافر بود و هدف سوم این بود که هزینه مسافران را به حداقل برساند. زمان انتظار انتقال برای چندین خط اتوبوس بسیار مهم بود و زمان انتظار طولانی برای انتقال، رضایت مسافران را کاهش می‌داد. بنابراین زمان انتظار انتقال به عنوان یک هدف واحد در نظر گرفته شد. علاوه بر این در مدل آن‌ها یک تابع ارزیابی با استفاده از الگوریتم ژنتیک برای تخمین اینکه آیا سه عملکرد هدف به یک بهینه سراسری رسیده است،استفاده شده است [۱۲].

جدول زمان‌بندی قطار نه تنها برای مسافران بلکه برای اپراتورهای قطار نقش مهمی در مدیریت قطار دارد. در بازار حمل و نقل پویا، جدول زمانی قطار یک پل ضروری است که ارائه دهنده خدمات را با تقاضای حمل و نقل متصل می‌کند. با این حال، در عملیات فعلی، برنامه ریزی قطار بدون در نظر گرفتن تقاضای مسافر می‌تواند از مزیت‌های رقابتی راه آهن در بازار حمل و نقل بکاهد و بیشتر منجر به نارضایتی مسافر شود. بنابراین روند برنامه‌ریزی قطارها بایستی به گونه‌ای باشد که نیازهای مسافران را پوشش دهد.

کورمن [۳] و همکارانش، [۱۳] مسئله زمان‌بندی قطارها را با تقاضای مسافر، به طور خاص تصمیم‌گیری در مورد برنامه توقف قطار بر اساس جفت‌های مختلف تقاضای مسافر مبدا- مقصد مورد بررسی قرار دادند. آن‌ها با در نظر گرفتن شاخص‌های توقف به عنوان متغیرهای مهم تصمیم‌گیری، یک مدل برنامه‌ریزی خطی عدد صحیح آمیخته ارائه کردند که هدف آن‌ها به حداقل رساندن زمان کل سفر قطار و به حداکثر رساندن تعداد مسافران است.

ژائوپنگ [۴] و همکارانش، یک روش جدید برای زمان‌بندی حمل و نقل عمومی با هدف بهینه‌سازی کیفیت خدمات ارائه شده به مسافران و استفاده بهینه از منابع محدود حمل و نقل عمومی پیشنهاد نمودند. در روش پیشنهادی آن‌ها، تصمیم‌گیری مبتنی بر دو متغیر فازی متشکل از رضایت مسافر از کیفیت خدمات ارائه شده حمل و نقل عمومی و استفاده بهینه از ظرفیت وسیله نقلیه است که در مقایسه با جدول زمانی اصلی، جدول زمانی بهینه

---

[۱] Nothegger [۲] Feng [۳] Corman [۴] zhang

سازی شده می‌تواند هم زمان با توازن در رضایت مسافر و بار وسیله نقلیه را ایجاد و برای هر بازه زمانی سرویس ارائه دهد [۱۴].

یکی از کاربردهای مسئله زمان‌بندی، مسئله زمان‌بندی امتحانات است که در فصل سوم به آن می‌پردازیم. برای حل این مسئله از الگوریتم ابتکاری و حریصانه استفاده می‌کنیم که در ادامه برخی از پیش‌نیازهای آن امده است.

۱-۲ آشنایی با روش حریصانه

عنوان این روش از شخصیت اسکروج گرفته شده است. اسکروج به جای آن که به آینده فکر کند، تنها انگیزه هر روز او به دست آوردن طلای بیشتر بود. الگوریتم حریصانه [۱] نیز مانند شیوه کار اسکروج است. الگوریتم حریصانه اغلب برای حل مسئله بهینه‌سازی استفاده می‌شود. این الگوریتم بهترین انتخاب را در هر مرحله با توجه به شرایط مسئله انجام می‌دهد به امید آن که با ادامهٔ همین روند، بهینه‌سازی حاصل شود. این نکته که بهترین انتخاب فعلی، ما را به جواب بهینه می‌رساند باید اثبات شود. در شیوه حریصانه در هر مرحله عنصری که بر مبنای معیاری معین «بهترین» به نظر می‌رسد، بدون توجه به انتخاب‌هایی که قبلاً انجام شده یا در آینده انجام خواهد شد، انتخاب می‌شود. الگوریتم‌حریصانه اغلب راه‌حل‌های ساده‌ای هستند و مسئله به نمونه‌های کوچک‌تر تقسیم نمی‌شود.

در بسیاری از موارد مسئله از ما می‌خواهد که یک مقدار بهینه را حداقل یا حداکثر کنیم یا این که صورت مسئله مستقیماً از ما نمی‌خواهد که چیزی را حداقل یا حداکثر کنیم اما می‌توان مسئله را تبدیل به یک چنین مسئله‌ای کرد. در اکثر مسائل از این دست ما باید تعدادی انتخاب انجام دهیم و مسئله در تعدادی مرحله حل شود. بازهٔ وسیعی از این مسائل توسط الگوریتم‌های حریصانه قابل حل هستند. الگوریتم‌های حریصانه در هر مرحله در نظر می‌گیرند که کدام انتخاب در حال حاضر بهترین وضعیت را برای ما رقم می‌زند و آن را انجام می‌دهند بدون در نظر گرفتن این که انجام این حرکت در آینده ممکن است به ضرر ما تمام شود. بنابراین الگوریتم حریصانه همیشه جواب درست را به دست نمی‌آورد. این دسته از الگوریتم‌ها در علوم رایانه کاربرد وسیعی دارند. مراحل الگوریتم حریصانه به صورت زیر است :

۱. کار با یک مجموعه تهی شروع شده و به ترتیبی خاص عناصری به مجموعه اضافه می‌شوند. هر دور تکرار الگوریتم شامل مراحل زیر است:

---

[۱] Greedy

(آ) روال انتخاب: این روال عنصر بعدی را طبق یک معیار حریصانه انتخاب می‌کند. این انتخاب یک شرط بهینه را در همان زمان بر آورده می‌سازد.

(ب) تحقیق عملی بودن: در این مرحله مشخص می‌شود که آیا مجموعه جدید به دست آمده، برای رسیدن به جواب عملی است یا خیر؟

(ج) تحقیق جواب: مشخص می‌سازد که آیا مجموعه جدید، نمونه مورد نظر را حل کرده است یا خیر

یکی از مسائل پرکاربرد که توسط روش حریصانه حل شده، مسئله خرد کردن پول با کمترین تعداد سکه است. در این مسئله هدف، خرد کردن پول با کمترین تعداد سکه است. روش کار را که مرتب سازی نزولی سکه‌ها از با ارزش‌ترین به کم ارزش‌ترین و انتخاب آن‌ها است، با ذکر دو مثال ارائه می‌دهیم.

مثال ۱-۲-۱. فرض کنید می‌خواهیم ۳۶ تومان را با سکه‌های ۵۰ تومانی، ۲۵ تومانی، ۱۲ تومانی، ۱۰ تومانی، ۵ تومانی و یک تومانی پرداخت کنیم به طوری که کمترین تعداد سکه را بپردازیم. بر اساس الگوریتم ارائه شده، جواب مسئله یک سکه ۲۵، یک سکه ۱۰ تومانی و یک سکه ۱ تومانی می‌دهد که جواب بهینه آن هم است.

مثال ۱-۲-۲. فرض کنید می‌خواهیم ۱۶ تومان را با سکه‌های ۵۰ تومانی، ۲۵ تومانی، ۱۲ تومانی، ۱۰ تومانی، ۵ تومانی و یک تومانی پرداخت کنیم به طوری که کمترین تعداد سکه را بپردازیم. بر اساس الگوریتم ارائه شده، جواب مسئله یک سکه ۱۲، چهار سکه یک تومانی می‌دهد که جواب بهینه یک سکه ۱۰، یک سکه ۵ و یک سکه ۱ تومانی است.

همان طور که در مثال (۱-۲-۲) مشاهده نمودید، روش حریصانه برای حل این مسئله لزوماً جواب بهینه را بدست نمی‌آورد. [۱۵].

جدول ۱-۱: مقایسه نتایج حل دو مثال خرد کردن پول با روش حریصانه با سکه‌های ۵۰، ۲۵، ۱۲، ۱۰، ۵ و یک تومانی

مثال	مبلغ مورد نظر	جواب با استفاده از روش حل مسئله	جواب بهینه مسئله
(۱-۲-۱)	۳۶	{۲۵،۱۰،۱}	{۲۵،۱۰،۱}
(۲-۲-۱)	۱۶	{۱۲،۱،۱،۱،۱}	{۱۰،۵،۱}

در فرآیند یادگیری ماشین [۱] که زیرمجموعه‌ای از هوش مصنوعی است، طبقه‌بندی داده‌ها اهمیت زیادی دارد و اگر داده‌های نامناسب به سیستم وارد شوند، خروجی نیز نامناسب خواهد بود. در واقع الگوریتم از روی داده‌ها،

[۱] Machine Learning

یادگیری را انجام می‌دهد. داده‌ها را می‌توان دسته‌بندی و یا خوشه‌بندی کنیم. با توجه به اهمیت موضوع، در ادامه به بیان تعریف دسته‌بندی و خوشه‌بندی و تفاوت‌های بین آن‌ها می‌پردازیم.

۱-۳ خوشه‌بندی

ابتدا تفاوت بین دسته‌بندی و خوشه‌بندی را بیان می‌کنیم [۱۶]. در فرآیندهای یادگیری ماشین دو نوع یادگیری با نظارت [۱] (دسته‌بندی [۲]) و یادگیری بدون نظارت [۳] (خوشه‌بندی [۴]) را داریم. در یادگیری با نظارت، ابتدا یادگیری با بخشی از داده‌ها برچسب دار صورت گرفته و سپس بر اساس مدل، داده‌های جدید بدون برچسب را پیش‌بینی می‌کنیم. در یادگیری بدون نظارت، داده‌های برچسب نداشته و امکان یادگیری وجود ندارد. بنابراین مدل با استفاده از خوشه‌بندی داده‌ها، ارتباطات و الگوهای موجود در آن‌ها را کشف می‌کند. برای مثال، یک سیستم یادگیری ماشین بدون نظارت قادر به تشخیص دو نوع میوه سیب و انبه از یکدیگر بوده ولی اما نمی‌توان نام آن‌ها را به صورت جداگانه روی هر دسته مشخص کند. از جمله الگوریتم‌های خوشه‌بندی می‌توان به DB Scan، OPTICS و K Means اشاره نمود. ویژگی [۵]، اساس یادگیری ماشین و داده‌کاوی است. به طور مثال اگر بخواهیم خودروها را خوشه‌بندی کنیم برای هر خودرو می‌توان طول، ارتفاع، وزن، میزان مصرف سوخت و سایر موارد را مطرح نمود. به هر یک از این مقدارها یک ویژگی گفته می‌شود. گاهی داده‌ها دو بعدی و گاهی دارای بعد بیشتری می‌توانند باشند. در ادامه به بیان یکی از الگوریتم‌های خوشه‌بندی می‌پردازیم.

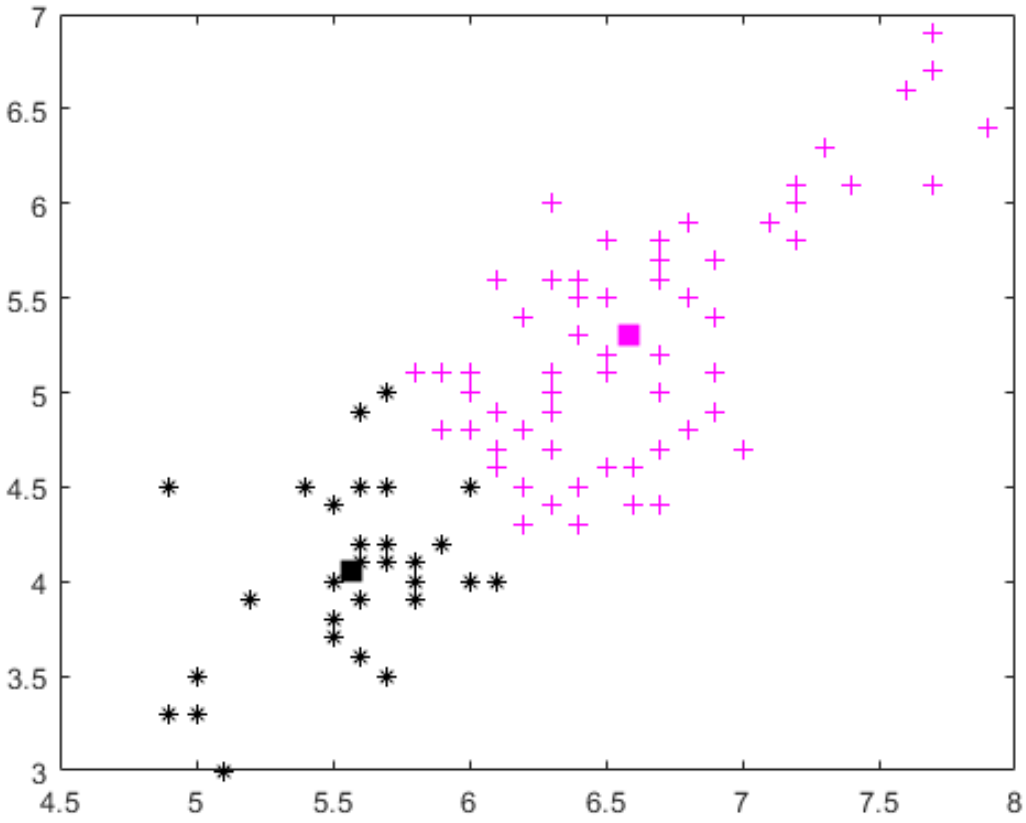
۱-۳-۱ الگوریتم خوشه‌بندی K-میانگین

الگوریتم K-میانگین [۶]، ساده‌ترین الگوریتمِ خوشه‌بندی و از جمله الگوریتم‌های کاربردی در این زمینه است. در این الگوریتم، ابتدا تعدادی نقطه را به صورت دلخواه به عنوان مرکز خوشه در نظر می‌گیریم. سپس با استفاده از میزان فاصله داده‌ها با مرکزهای خوشه، هر کدام از داده‌ها را به یک مرکز خوشه نسبت می‌دهیم. سپس با گرفتن میانگین هر خوشه، می‌توان مرکزهای خوشه را جا به جا کنیم. مرحله قبل را می‌توان تا زمانی که مرکزهای خوشه جا به جا نشوند، ادامه داد. تعیین مناسب مرکزهای خوشه اولیه و تعیین تعداد خوشه‌ها، در میزان بهینه بودن روش تاثیرگذار است.

مثال ۱-۳-۱. یکی از مجموعه داده‌های پرکاربرد، مجموعه گل‌های زنبق [۷] است که توسط آقای فیشر [۸] جمع‌آوری گردیده است و با عنوان مجموعه Fisheriris شناخته می‌شود. این مجموعه داده، شامل طول کاسبرگ [۹] ، عرض

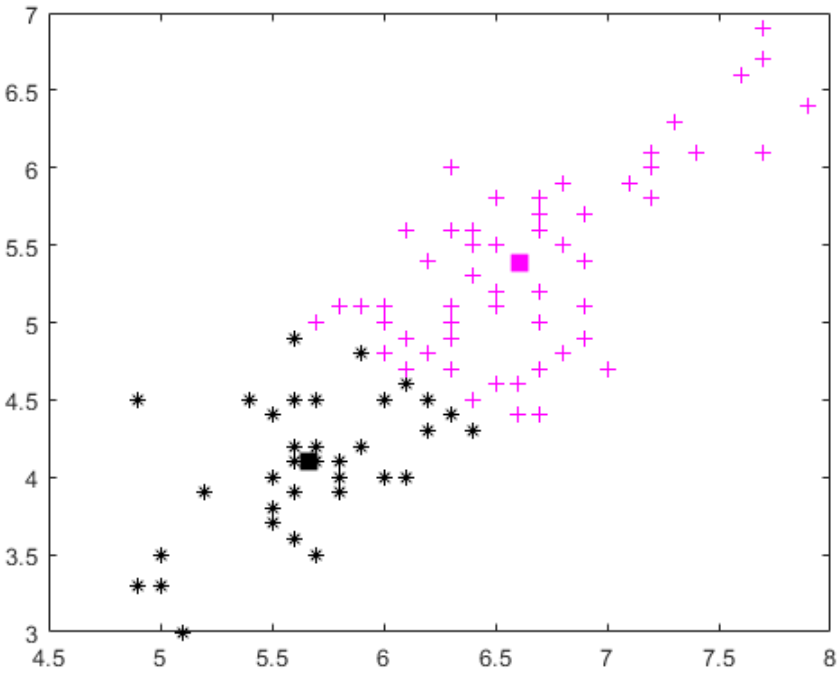
[۱] Supervised Learning [۲] Classification [۳] Unsupervised Learning [۴] Clustering [۵] Feature
[۶] K-means [۷] Irises [۸] Fisher [۹] Sepal Length

کاسبرگ [۱] ، طول گلبرگ [۲] و طول گلبرگ [۳] بر حسب واحد سانتی‌متر است. این گل‌ها در سه کلاس سِتوسا (محدوده ۱ تا ۵۰) [۴] ، ، وِرسی‌کالِر (محدوده ۵۱ تا ۱۰۰) [۵] و ویرجینیکا (محدوده ۱۰۱ تا ۱۵۰) [۶] تقسیم‌بندی می‌شوند. در این مثال، دو دسته وِرسی‌کالِر و ویرجینیکا که در بازه ۵۱ تا ۱۵۰ هستند بر حسب ویژگی طول کاسبرگ و طول گلبرگ با روش K-میانگین خوشه‌بندی می‌کنیم. نتیجه هر مرحله اجرا را در شکل‌های (۱-۱) تا (۱-۷) آمده است. با توجه به این که نتایج اجرای الگوریتم K-میانگین در مرحله‌های ۶ و ۷ بدون تغییر است، بنابراین خوشه‌بندی تغییری نمی‌کند و نیاز به ادامه اجرای الگوریتم K-میانگین نیست.

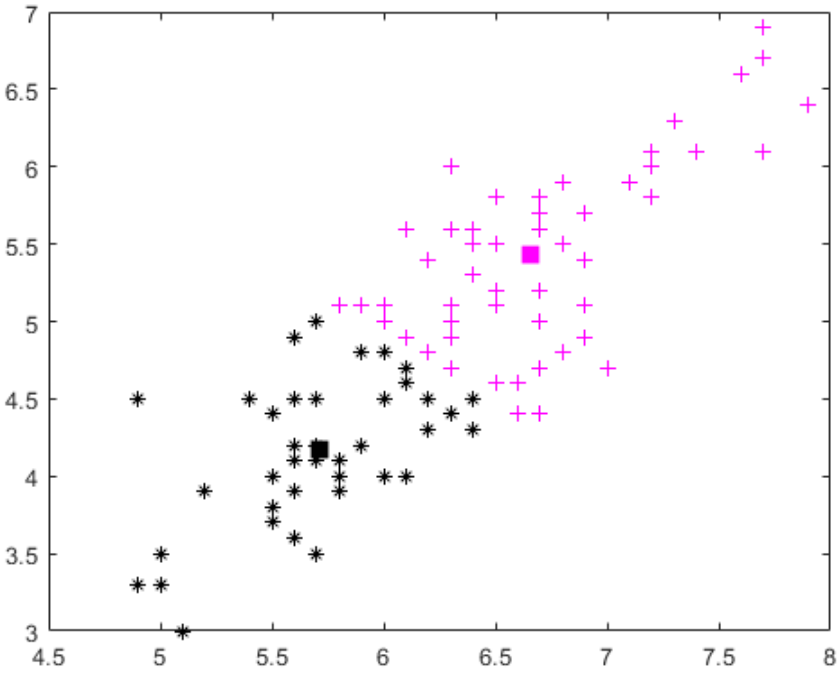


شکل ۱-۱: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : اولین مرحله اجرا

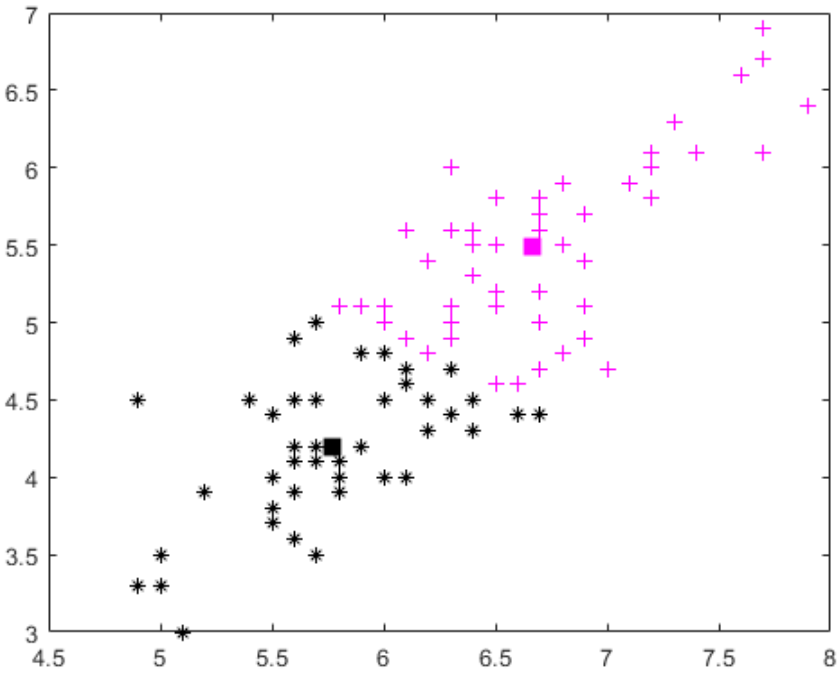
[۱]Sepal Width [۲]Petal Length [۳]Petal Length [۴]Setosa [۵]Versicolor [۶]Virginica



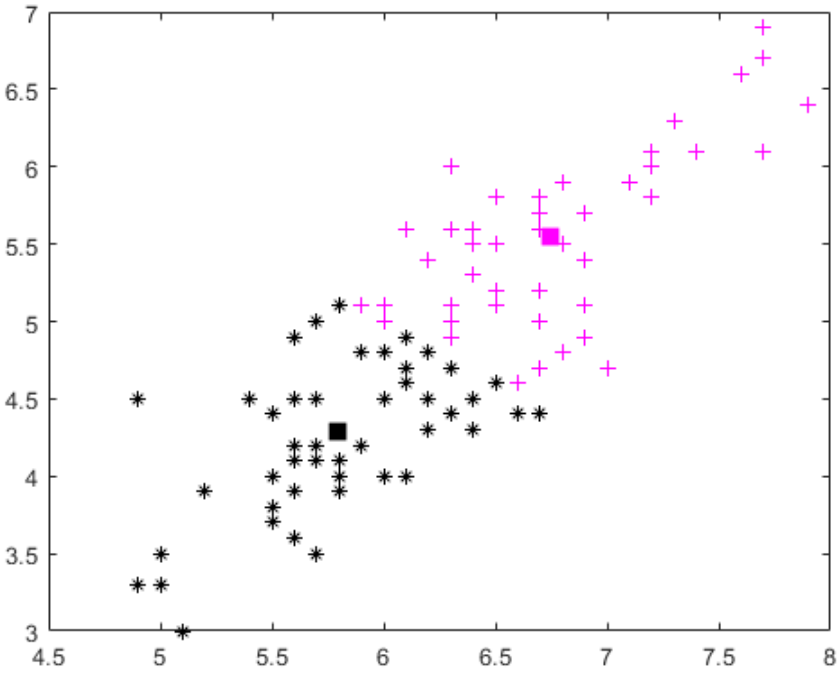
شکل ۱-۲: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : دومین مرحله اجرا



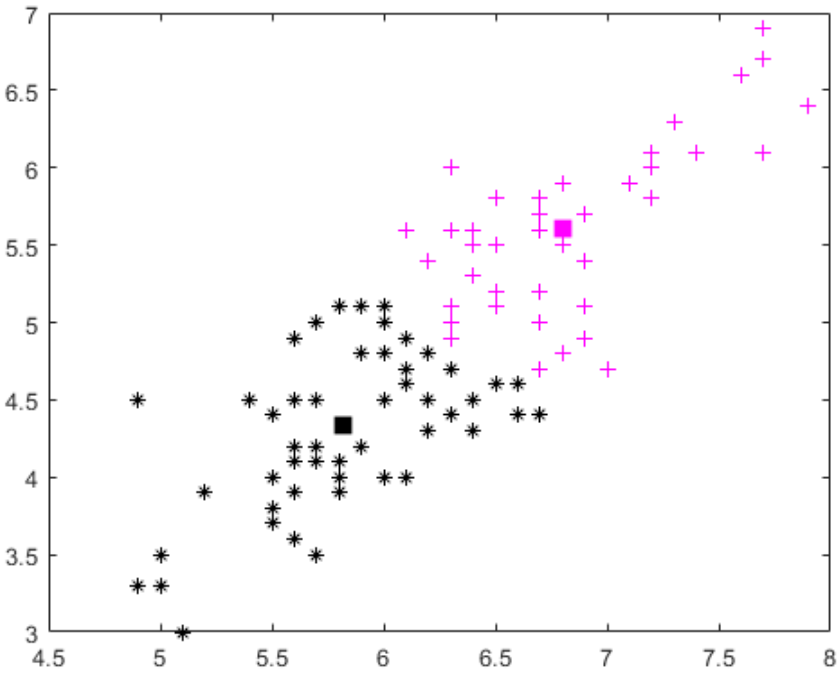
شکل ۱-۳: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : سومین مرحله اجرا



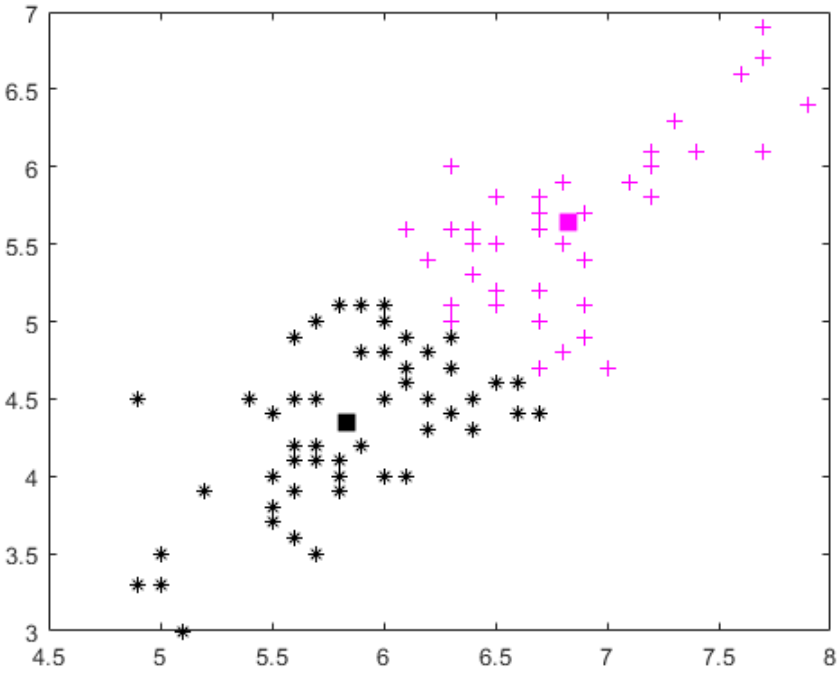
شکل ۱-۴: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : چهارمین مرحله اجرا



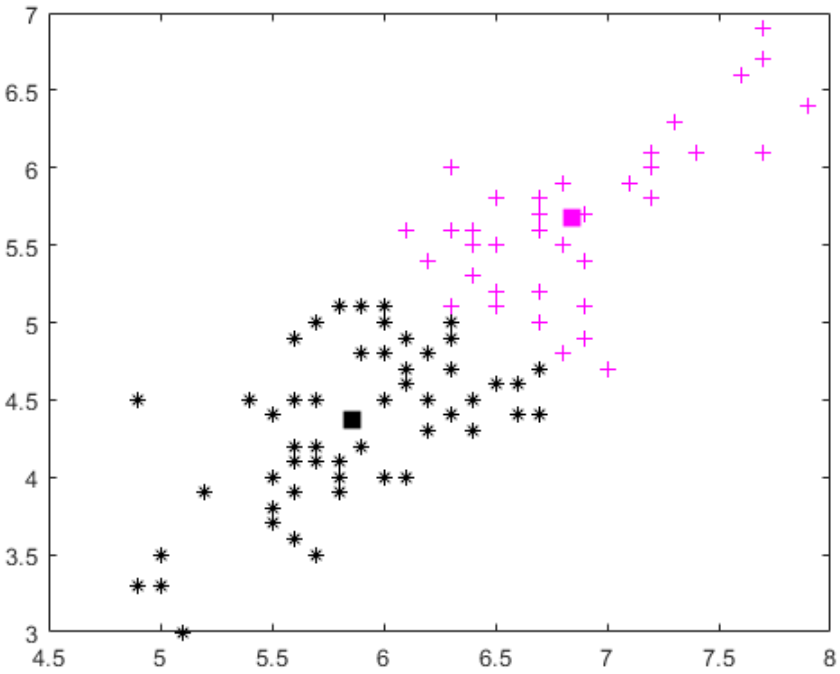
شکل ۱-۵: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : پنجمین مرحله اجرا



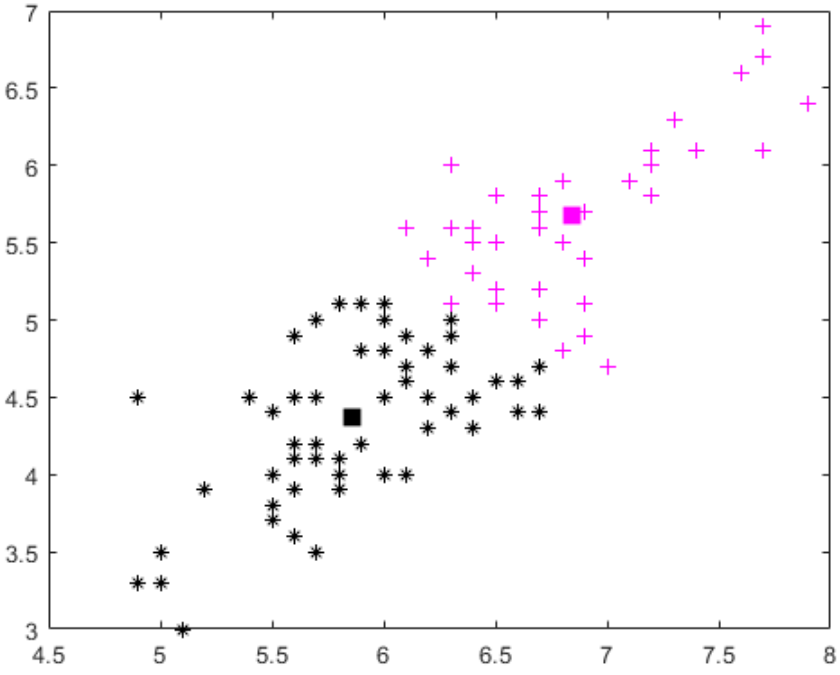
شکل ۱-۶: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : ششمین مرحله اجرا



شکل ۱-۷: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : هفتمین مرحله اجرا



شکل ۱-۸: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : هشتمین مرحله اجرا



شکل ۱-۹: خوشه‌بندی مجموعه گل‌های زنبق با روش K-میانگین در مثال ۱-۳-۱ : نهمین مرحله اجرا

فصل ۲

یک رویکرد مبتنی بر بهینه‌سازی چند هدفه برای مسئله زمان‌بندی دروس دانشگاهی

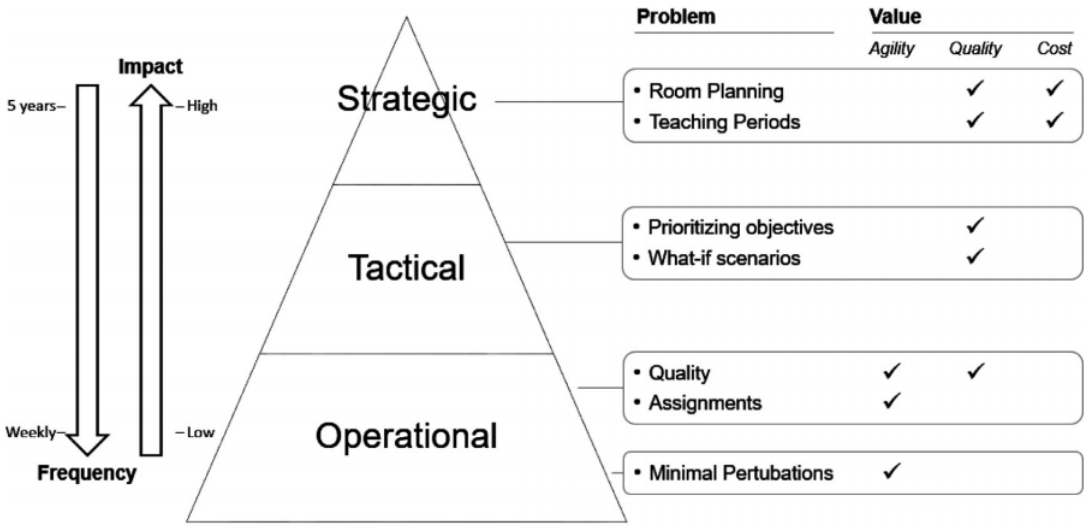
هدف مسئله زمان‌بندی دروس دانشگاهی، اختصاص دادن جلسه‌های درسی به اتاق‌ها و بازه‌های زمانی و ایجاد زمان‌بندی درس‌ها (برنامه کلاسی) با کیفیت بالا برای دانش آموزان و کارکنان (استادان) است [۱۷] . باید برای دو مسئله مهم ابتدا تصیم‌گیری کرد:

- چه اتاق‌هایی برای برگزاری جلسه‌های درس لازم داریم؟
- چه بازه‌های تدریس (بازه‌های زمانی) لازم است؟

این تصمیم‌ها تأثیر زیادی بر زمان‌بندی حاصل خواهد داشت که به ندرت تغییر داشته و مورد بررسی قرار می‌گیرد. این فصل بر حل این دو مسئله مهم تمرکز دارد و تأثیر این تصمیم‌ها بر کیفیت زمان‌بندی درس‌ها، بررسی می‌شود. هم چنین رابطه و تفاوت بین مسائل عملیاتی، تاکتیکی و استراتژیک مسئله زمان‌بندی بررسی می‌گردد. پایه فرمول‌بندی درس‌ها بر اساس دوره‌های آموزشی بوده و بر اساس سه مدل بهینه‌سازی چند هدفه مدل سازی شده است.

۲-۱ مسائل عملیاتی، تاکتیکی و استراتژیک در زمان‌بندی دروس دانشگاهی

سه دیدگاه کلی (استراتژیک، تاکتیکی و عملیاتی) مرتبط به مسئله زمان‌بندی درس‌ها در شکل ۲-۱ آمده‌است. لیست مسئله‌های مرتبط با سه دیدگاه مذکور که در ادامه مورد بررسی قرار خواهند گرفت به شرح زیر است :



شکل ۲-۱: نمودار تغییرات در سطوح مختلف مسئله زمان‌بندی

۱. استراتژیک [۱]

- برنامه‌ریزی اتاق‌ها [۲]
- بازه‌های تدریس [۳]

۲. تاکتیکی [۴]

- اولویت‌بندی اهداف [۵]
- سناریوی چه اقدامی انجام شود؟ [۶]

۳. عملیاتی [۷]

- کیفیت [۸]
- تخصیص [۹]
- تداخل حداقلی [۱۰]

هر یک از این مسائل به صورت عملیاتی، تاکتیکی یا استراتژیک طبقه‌بندی می‌شوند. مسائل به ترتیب استراتژیک، تاکتیکی و در نهایت عملیاتی حل می‌گردند. مسائل حل شده استراتژیک بر مسائل تاکتیکی تاثیر می‌گذارند و حل مسائل عملیاتی، منجر به برنامه کلاسی نهایی می‌شود. مسائل در هر سطح، در شکل ۲-۱ نشان داده شده است. هر یک از مسئله‌های مطرح شده، با سه ویژگی زیر توصیف می‌شوند:

---

[۱]Strategic [۲]Room Planning [۳]Teaching Periods [۴]Tactical [۵]Prioritizing Objectives
[۶]What-if Scenarios [۷]Operational [۸]Quality [۹]Assignments [۱۰]Minimal Pertubations

۱. چابکی [۱] : توانایی تطبیق با تغییرات و سرعت عمل تطبیق با تغییرات زمان‌بندی درس‌ها است.

۲. کیفیت [۲] : زمان‌بندی درس‌ها که شرایط کاری خوب برای کارکنان و دانشجویان ایجاد کند.

۳. هزینه [۳] : زمان‌بندی درس‌ها که استفاده موثر از منابع داشته باشد.

مسائل عملیاتی، شامل مسائلی هستند که حل آن‌ها، در برنامه نهایی کلاسی تاثیرگذار خواهد بود. برای مثال، می‌توان به تصمیم‌گیری در مورد این که جلسه‌ها در چه اتاق و چه بازه زمانی برگزار شود، می‌پردازد و مسئله اصلی هر ترم است. دو موضوع مرتبط، یکی مسئله تخصیص است که هدف یک زمان‌بندی شدنی درس‌ها [۴] و بدون تداخل [۵] است. دومین مسئله از مسائل استراتژیک، مسئله معیار کیفیت است. هدف مسئله کیفیت، ایجاد زمان‌بندی مطلوب درس‌ها برای دانشجویان و استادان است. محدودیت‌های نرم برای مسئله معیار کیفیت استفاده می‌شود و باید نرخ جریمه ناشی از نقض محدودیت‌های نرم را به حداقل رساند.

مسائل تاکتیکی، شامل تصمیم‌گیری‌هایی است که باید قبل از ایجاد زمان‌بندی درس‌ها اتخاذ شود. یکی از این مسائل، اولویت‌بندی اهداف است که بین معیار کیفیت و چگونگی تقسیم بازه‌های زمانی کدام را ترجیح می‌دهیم. این تصمیم‌گیری، بر کیفیت دوره تحصیلی به دلیل نداشتن اتاق مناسب و در نتیجه تخصیص در یک بازه زمانی نامطلوب، تاثیر می‌گذارد.

مسائل استراتژیک، شامل تصمیمات درازمدت است که بر برنامه کلاسی و سازمان تأثیر زیادی می‌گذارند. برای مثال می‌توان از این طریق اجازه داد که تدریس در روز بعد با بازه‌های زمانی اضافه برنامه‌ریزی شود. یکی از مسائل استراتژیک برنامه‌ریزی اتاق‌ها است. افزودن بازه‌های زمانی اضافی، تأثیر زیادی در زمان‌بندی استادان و دانشجویان دارد و باید اغلب توسط مدیریت ارشد تصمیم‌گیری شود. به دلیل گران بودن ساختمان‌ها و اتاق‌ها، هزینه‌های بسیار زیادی را بر دانشگاه تحمیل می‌کند تا جایی که اغلب اتاق‌های استفاده نشده به دفتر مسئولان تخصیص و یا به اجاره داده می‌شوند. با این حال، مسئله استراتژیک را نمی‌توان سال به سال تغییر داد. بنابراین، مهم است که اتاق‌های جدید مطابق ظرفیت واقعی مورد نیاز در نظر گرفته شود. در صورت نیاز به اتاق‌ها در ترم بعد، اتاق‌های مورد نیاز را نباید تا چند سال اجاره داد. این مسئله، مسئله برنامه‌ریزی اتاق‌ها نامیده می‌شود.

در مقایسه با مسئله عملیاتی و تاکتیکی، مسئله استراتژیک کمتر مورد توجه بوده است. مسئله برنامه‌ریزی اتاق در ابتدا توسط فیزانو [۶] و سوانسون [۷] [۱۸] مورد بررسی قرار گرفت. آن‌ها از یک مدل تخصیص [۸] برای پیدا کردن حداقل تعداد اتاق‌های لازم برای ایجاد یک برنامه قابل اجرا استفاده کردند. این کار به طور تکراری با حذف اتاق‌ها انجام می‌شود تا زمانی که مسئله حل شود. بیروتی و همکارانش [۹] چگونگی تاثیر اتاق‌های موجود بر روی برنامه‌ریزی کلاسی را مورد تجزیه و تحلیل قرار داده [۱۹] و سپس نشان دادند که چگونه می‌توان مسئله بهینه‌سازی

[۱]Agility [۲]Quality [۳]Cost [۴]feasible [۵]Conflict-free [۶]Fizzano [۷]Swanson [۸]Assignments [۹]Beyrouthy et al.

اتاق را به حداکثر رساند. با این وجود، محدودیت‌های جدول زمانی مانند برنامه‌های درسی دانشجویان و استادان را در نظر نگرفتند که می‌تواند یک برنامه کلاسی ایجاد شده را غیرممکن کند. مسئله دوره‌های تدریس، مسئله انتخاب تعداد بازه‌های زمانی است که قبلا بررسی نشده است. هدف این فصل، ایجاد بینش جدید در مورد این است که چگونه تصمیم‌های استراتژیک بر برنامه زمانی تاثیر می‌گذارد. دو تصمیم استراتژیک کدام اتاق‌ها باید استفاده شوند و چند بازه زمانی باید وجود داشته باشد، در نظر گرفته شده است. هم چنین تحلیل این که چگونه این تصمیم‌ها با استفاده از بهینه‌سازی دو هدفه و برنامه‌ریزی عدد صحیح آمیخته بر کیفیت برنامه زمانی تاثیر می‌گذارد، بررسی می‌شود.

در ابتدا مفروضات اصلی مسئله بیان شده و سپس به مدل سازی ریاضی آن خواهیم پرداخت.

۲-۲ بیان مسئله زمان‌بندی دانشگاهی

در مسئله زمان‌بندی درس‌ها بر پایه دوره آموزشی، موارد زیر را مفروض می‌گیریم:

۱. مجموعه‌ای از درس‌ها که باید زمان‌بندی شوند

۲. تعداد جلسه‌های مورد نیاز هر درس

۳. بازه‌های زمانی و اتاق که باید به هر جلسه به صورت بدون تداخل تخصیص پیدا کند

۴. دو جلسه از یک درس نباید به طور هم زمان برگزار شود

۵. جلسه‌های یک درس باید در اتاق مشخص برگزار شود

۶. هر اتاق ظرفیت مشخصی دارد

۷. در هر اتاق در یک بازه زمانی فقط یک درس برگزار شود

۸. ظرفیت اتاق باید جوابگوی تعداد دانشجویان باشد

۹. یک استاد در بازه زمانی فقط می‌تواند یک درس را تدریس کند

۱۰. برای هر درسی یک استاد در نظر گرفته شود

۱۱. دوره‌های آموزشی شامل تعدادی درس‌ها هستند که نمی‌توانند در بازه زمانی مشابه برنامه‌ریزی شوند

جدول ۲-۱: نمادگذاری‌های مسئله زمان‌بندی دروس دانشگاهی

مجموعه ها	توضیح
C	مجموعه درس‌ها
CU	دوره‌های تحصیلی
P	مجموعه بازه‌های زمانی در کل هفته
D	مجموعه بازه‌های زمانی به ازاء هر روز هفته $D = \{P_{Mo}, P_{Tu}, P_{We}, P_{Th}, P_{Fr}\}$
R	مجموعه اتاق‌ها
T	مجموعه استادان
$S = \{cap(r) : r \in R\} \cup \{۰\}$	مجموعه غیر تکراری ظرفیت اتاق‌ها با احتساب صفر
$S_{\geq s} = \{s' \in S : s' \geq s\}$	مجموعه غیر تکراری ظرفیت اتاق‌های بزرگتر یا مساوی s $(s \in S)$
$C_{\geq s} = \{c \in C : dem(c) \geq s\}$	لیست درس‌های که تعداد دانشجوی آن‌ها بزرگتر یا مساوی ظرفیت s است $(s \in S)$
$R_{\geq s} = \{r \in R : cap(r) \geq s\}$	لیست اتاق‌هایی که ظرفیت آن‌ها بزرگتر یا مساوی ظرفیت s است $(s \in S)$
پارامترها	**توضیح**
$l(c)$	تعداد جلسه‌های مورد نیاز برای یک درس $(c \in C)$.
$mnd(c)$	حداقل تعداد روز لازم در هفته برای برگزاری جلسه‌ها یک درس $(c \in C)$
$dem(c)$	تعداد دانشجویان ثبت نامی برای یک درس $(c \in C)$
$cap(r)$	ظرفیت یک اتاق $(r \in R)$
متغیرهای تصمیم	**توضیح**
$x_{c,p} \in \mathbb{B}$	از نوع باینری، اگر درس $c \in C$ در بازه زمانی $p \in P$ قرار گیرد، $true$ است
$v_{cu,p} \in [۰, ۱]$	اگر دوره آموزشی $cu \in CU$ در بازه زمانی $p \in P$ تداخل داشت ، مقدار جریمه یک است
$q_{cu,p} \in [۰, ۱]$	اگر دوره آموزشی $cu \in CU$ در بازه زمانی $p \in P$ قرار گیرد، مقدار آن یک است
$z_{c,d} \in [۰, ۱]$	اگر درس $c \in C$ در روز $d \in D$ قرار گیرد، مقدار آن یک است
$w_c \in \mathbb{R}^+ (c \in C)$	جریمه این که تعداد روزهای تخصیص یافته به جلسه‌های درس از حداقل آن بیشتر باشد
$r_s \in \mathbb{Z}^+ (s \in S)$	تعداد اتاق‌ها با ظرفیت s که نیاز داریم
$r_s^+ \in \mathbb{Z}^+ (s \in S)$	تعداد اتاق‌ها با ظرفیت s که وجود دارند
$t_p \in \mathbb{B} (p \in P)$	متغیر باینری (t_p) مقدار $true$ است اگر بازه زمانی p مجاز باشد

۲-۲-۱ مدل سازی مسئله

فرمول اصلی شامل چهار معیار کیفی مختلف (محدودیت‌های نرم) است: ظرفیت اتاق [۱]، بیان می‌کند که ظرفیت اتاق باید بتواند تمامی تعداد مورد نیاز دانشجویان را در نظر بگیرد؛ پایداری اتاق [۲]، بیان می‌کند که تمام جلسه‌های یک درس باید در یک اتاق برنامه‌ریزی شود؛ حداقل روزهای تدریس [۳]، جریمه‌ای در نظر گرفته می‌شود

[۱] Room Capacity [۲] Room Stability [۳] Minimum Working Days

که اگر در تخصیص بازه‌های زمانی برای جلسه‌های یک درس، تعداد روزهای حداقلی لحاظ نشود به ازاء هر روز استفاده نشده، جریمه در نظر گرفته شود؛ فشردگی دوره آموزشی [۱]، برای حالتی که اگر بازه زمانی تخصیص یافته به جلسه‌های یک دوره آموزشی در کنار هم نباشد، جریمه در نظر گرفته می‌شود. نماد گذاری‌های مورد نیاز برای مدل سازی مسئله در جدول ۲-۱ آمده است.

۲-۲-۱-۱ مدل کیفیت

مسئله کیفیت[۲]، مسئله استاندارد در زمان‌بندی درس‌ها است که هدف تولید زمان‌بندی شدنی با کیفیت بالا [۳] است. در این مدل، تعدادی محدودیت‌های نرم تعریف شده و کیفیت بر اساس جریمه‌های ناشی از عدم رعایت محدودیت‌های نرم سنجیده می‌شود. در مدل کیفیت، محدودیت نرم ظرفیت اتاق را به صورت محدودیت سخت در نظر گرفته و به منظور کاهش پیچیدگی محاسبات، معیار پایداری اتاق را مد نظر نمی‌گیریم. این مدل با استفاده از روابط (۲-۱آ) - (۲-۱م) بیان می‌شود که نمادهای مربوط در جدول ۲-۱ تعریف شده‌اند.

تابع بهینه‌سازی استفاده شده تک هدفه است. متغیر $x_{c,p}$ (۲-۱ط) از نوع باینری بوده و اگر درس $c \in C$ در بازه زمانی $p \in P$ تخصیص یافته باشد، دارای مقدار یک خواهد بود. محدودیت (۲-۱ج) تضمین می‌دهد که برای همه جلسه‌های یک درس، بازه زمانی تخصیص یافته است. محدودیت (۲-۱ب) تضمین می‌دهد که ظرفیت کلاس‌های تخصیص یافته به درس‌ها جوابگوی تعداد دانشجوی آن درس است. محدودیت (۲-۱ح) تضمین می‌دهد که برای استادان در هر بازه زمانی حداکثر یک جلسه تخصیص یافته است. متغیر $v_{cu,p}$ (۲-۱م)، اگر دوره آموزشی $cu \in CU$ در بازه زمانی $p \in P$ دارای تداخل ذکر شده در بخش ۲-۲ باشد، دارای مقدار یک خواهد بود که توسط محدودیت (۲-۱ز) تضمین می‌شود. متغیر مثبت $q_{cu,p}$ (۲-۱ل)، اگر دوره آموزشی $cu \in CU$ در بازه زمانی $p \in P$ باشد، دارای مقدار یک خواهد بود که توسط محدودیت (۲-۱و) تضمین می‌شود. دو جمله رابطه (۲-۱آ) بدین صورت محاسبه می‌شود: برای محاسبه حداقل روزهای تدریس متغیر مثبت $z_{c,d}$ را در نظر می‌گیریم که اگر درس $c \in C$ در روز $d \in D$ باشد، مقدار آن یک خواهد بود که این محاسبه، توسط محدودیت (۲-۱د) تضمین می‌شود و در نتیجه تخطی از آن (بیشتر بودن تعداد روزهای تخصیص یافته به جلسه‌ها درس از حداقل آن)، متغیر $w_c \in \mathbb{R}^+$ (۲-۱ی) دارای مقدار یک خواهد بود که همان جریمه مدنظر است و محاسبه آن توسط محدودیت (۲-۱ه) تضمین می‌گردد.

---

[۱] Curriculum Compactness [۲] Quality Problem [۳] feasible Timetable

$$\min f_{qual} = \sum_{c \in C} ۵.w_c + \sum_{cu \in CU, p \in P} ۲.v_{cu,p} \qquad (۲\text{-}۱آ)$$

$s.t.$

$$\sum_{c \in C_{\geq s}} x_{c,p} \leq |R_{\geq s}| \qquad \forall s \in S, p \in P \qquad (۲\text{-}۱ب)$$

$$\sum_{p \in P} x_{c,p} = L(c) \qquad \forall c \in C \qquad (۲\text{-}۱ج)$$

$$\sum_{p \in d} x_{c,p} - z_{c,d} \geq ۰ \qquad \forall c \in C, d \in D \qquad (۲\text{-}۱د)$$

$$\sum_{d \in D} z_{c,d} + w_c \geq mnd(c) \qquad \forall c \in C \qquad (۲\text{-}۱ه)$$

$$\sum_{c \in CU} x_{c,p} - q_{cu,p} = ۰ \qquad \forall cu \in CU, p \in P \qquad (۲\text{-}۱و)$$

$$- q_{cu,p-۱} + q_{cu,p} - q_{cu,p+۱} - v_{cu,p} \leq ۰ \qquad \forall cu \in CU, p \in P \qquad (۲\text{-}۱ز)$$

$$\sum_{c \in C(t)} x_{c,p} \leq ۱ \qquad \forall t \in T, p \in P \qquad (۲\text{-}۱ح)$$

$$x_{c,p} \in \mathbb{B} \qquad \forall c \in C, p \in P \qquad (۲\text{-}۱ط)$$

$$w_c \in \mathbb{R}^+ \qquad \forall c \in C \qquad (۲\text{-}۱ی)$$

$$z_{c,d} \in [۰, ۱] \qquad \forall c \in C, d \in P_d \qquad (۲\text{-}۱ک)$$

$$q_{cu,p} \in [۰, ۱] \qquad \forall cu \in CU, p \in P \qquad (۲\text{-}۱ل)$$

$$v_{cu,p} \in [۰, ۱] \qquad \forall cu \in CU, p \in P \qquad (۲\text{-}۱م)$$

۲-۲-۱-۲ مدل برنامه‌ریزی اتاق‌ها

مسئله برنامه‌ریزی اتاق‌ها[۱]، مسئله حداقل کردن تعداد اتاق‌های لازم برای پاسخگویی به ظرفیت مورد نیاز و تخصیص اتاق به جلسه‌های درس‌ها است. این مدل با استفاده از روابط (۲-۲آ) - (۲-۲ج) بیان می‌شود.

$$\min f_{seats} = \sum_{s \in S} s.r_s \qquad (۲-۲آ)$$

$s.t.$

$$|P| \sum_{s \in S_{\geq s}} r_s \geq \sum_{c \in C_{\geq s}} l(c) \qquad \forall s \in S \qquad (۲-۲ب)$$

$$r_s \in \mathbb{Z}^+ \qquad \forall s \in S \qquad (۲-۲ج)$$

مجموعه تعداد اتاق‌های مورد نیاز با ظرفیت s، با متغیر مثبت r_s ($r_s \in \mathbb{Z}^+$) بیان می‌شود. ظرفیت یک اتاق با اندازه فیزیکی و هزینه اتاق ارتباط مستقیم دارد. بنابراین هدف، به حداقل رساندن تعداد کل صندلی‌ها است که در شرط (۲-۲آ) آمده‌است. معیار بهره‌برداری استاندارد استفاده شده، توسط بیروتی و همکارانش [۲] [۱۹] تعریف شده است. معیار ۲-۳ ، میانگین نسبت تعداد صندلی‌های اشغال شده به تعداد بازه‌های زمانی مورد نیاز را محاسبه می‌کند. با حداقل نمودن تعداد صندلی‌ها، بازدهی حداکثر می‌شود.

$$Utiliztion = \frac{\sum_{c \in C} l(c).dem(c)}{|P| \sum_{s \in S} s.r_s} \qquad (۲-۳)$$

تصمیم‌گیری در مورد ظرفیت اتاق‌های مورد نیاز، در این مدل انجام می‌شود. به طور مثال اگر S مجموعه ظرفیت اتاق‌ها باشد، ظرفیتی مناسب یک درس است که مساوی با ظرفیت مورد نیاز جلسه‌های آن درس و یا کمترین اختلاف را داشته باشد. هر چه اختلاف ظرفیت مورد نیاز جلسه‌های درس با ظرفیت اتاق تخصیص یافته بیشتر باشد، اتلاف اتاق‌ها بیشتر می‌شود. بنابراین مجموعه S باید شامل تمامی ظرفیت‌های مورد نیاز درس‌ها باشد. در عمل، برای مدیر تفاوت بین اتاق با ظرفیت ۲۲ و ۲۵ صندلی اهمیت ندارد. برای کوچکتر نمودن مجموعه S، پارامتر با مقدار دلخواه δ را تعریف می‌کنیم. مجموعه S می‌توان به دو روش زیر تولید شود:

$$S_{optimal} = \{dem(c) : c \in C\} \qquad (۲-۴)$$

[۱] Room Planning Problem [۲] Beyrouthy et al.

$$S_{approx} = \{\delta.\lceil \frac{dem(c)}{\delta} \rceil : c \in C\} \qquad (۵-۲)$$

در حالت بهینه (معیار ۲-۴) [۱]، برای ظرفیت مورد نیاز جلسه‌های درس(تعداد دانشجویان ثبت نامی برای یک درس) [۲]، اتاق با همان ظرفیت وجود دارد. در حالت تقریبی (معیار ۲-۵) [۳]، برای ظرفیت مورد نیاز جلسه‌های درس(تعداد دانشجویان ثبت نامی برای یک درس) اتاق با همان ظرفیت وجود ندارد که در نتیجه ظرفیت اتاق‌های استفاده شده باید بزرگتر از ظرفیت مورد نیاز جلسه‌های درس با کمترین اختلاف و مضرب پارامتر δ باشند. بزرگ‌ترین ظرفیت اتاق بر اساس بیشترین تعداد دانشجویان درس‌ها مضربی از پارامتر δ است.

مثال ۲-۲-۱. برای نمونه، اگر δ را ۳ و $dem(c)$ را ۳۷ در نظر بگیریم، S_{approx} برابر با ۳۹ خواهد شد:

$$\begin{cases} S_{approx} = \{\delta.\lceil \frac{dem(c)}{\delta} \rceil : c \in C\} \\ dem(c) = ۳۷ \\ \delta = ۳ \end{cases} \Rightarrow \left| \begin{array}{ll} S_{approx} & = \{۳.\lceil \frac{۳۷}{۳} \rceil\} \\ & = \{۳.\lceil ۱۲/۳۳ \rceil\} \\ & = \{۳ * ۱۳\} \\ & = ۳۹ \end{array} \right.$$

زمانی که تصمیم به استفاده از یک اتاق را داریم، می‌توان جلسه‌های متعددی را برای همان اتاق در بازه‌های زمانی ($|P|$) در نظر گرفت. شرط داشتن اتاق کافی برای برنامه‌ریزی تمام جلسه‌های درس‌ها توسط محدودیت (۲-۲ب) تضمین می‌شود. روش برنامه‌ریزی عدد صحیح آمیخته زیر توسط بیروتی و همکارانش با استفاده از الگوریتم حریصانه، در زمان خطی بهینه‌سازی می‌شود:

۱. مرتب سازی جلسه‌ها برحسب ظرفیت مورد نیاز ($dem(c)$)

۲. تا زمانی که به جلسه‌ای اتاق تخصیص نیافته ادامه بده:

از بین جلسه‌های تخصیص نیافته، جلسه‌ای که بزرگترین ظرفیت مورد نیاز دارد را انتخاب و اتاق و بازه زمانی را به آن جلسه اختصاص بده. اگر اتاق با ظرفیت همسان آن جلسه نیست، نزدیک‌ترین اتاق با کمترین اختلاف را به آن جلسه تخصیص دهید.

---

[۱] $S_{optimal}$ [۲] dem(c) [۳] S_{approx}

۲-۲-۱-۳ مدل بازه‌های تدریس

مسئله بازه‌های تدریس[۱]، مسئله حداقل کردن تعداد بازه‌های زمانی برای پاسخگویی به ظرفیت مورد نیاز جلسه‌های درس جهت تخصیص است. یکی دیگر از عوامل مهم در برنامه کلاسی بهینه، تعداد بازه‌های زمانی در دسترس است که می‌توان برای تدریس استفاده نمود. این مدل با استفاده از روابط (۲-۶آ) - (۲-۶د) بیان می‌شود.

متغیر باینری t_p در صورتی که بازه زمانی p مجاز باشد، دارای مقدار یک خواهد بود (۲-۶د). تعداد کل بازه‌های زمانی استفاده شده توسط تابع (۲-۶آ) به دست می‌آید. محدودیت (۲-۶ب)، تضمین می‌کند که بازه‌های زمانی و اتاق با ظرفیت مناسب برای جلسه‌ها داشته باشد. در عمل، زمان‌بندی‌های انتخاب شده (به استثناء بازه‌های زمانی خالی وسط روز که به دلیل ناهار، همایش و غیره است)، متوالی هستند. محدودیت (۲-۶ج)، تضمین می‌کند که بازه زمانی t_{p+1} تنها زمانی می‌تواند مورد استفاده قرار گیرد که بازه زمانی t_p (بازه زمانی قبلی) نیز استفاده شود. نمونه‌ای از اندیس‌های انتخاب شده که پشت سر هم هستند در جدول ۲-۲ آمده است.

$$\min f_{time} = \sum_{p \in P} t_p \qquad (۲-۶آ)$$

$s.t.$

$$|R_{\geq s}| \sum_{p \in P} t_p \geq \sum_{c \in C_{\geq s}} l(c) \qquad \forall s \in S \qquad (۲-۶ب)$$

$$t_{p+1} - t_p \leq 0 \qquad \forall p \in P \qquad (۲-۶ج)$$

$$t_p \in \mathbb{B} \qquad \forall p \in P \qquad (۲-۶د)$$

جدول ۲-۲: نمونه اندیس‌های انتخاب شده

شنبه	یکشنبه	دوشنبه	سه شنبه	چهارشنبه
۱	۲	۳	۴	۵
۶	۷	۸	۹	۱۰
۱۱	۱۲	۱۳	۱۴	۱۵
۱۶	۱۷	۱۸	۱۹	۲۰

[۱] Teaching Periods Problem

۲-۲-۱-۴ مدل برنامه‌ریزی اتاق‌ها در مقابل مدل کیفیت

برنامه کلاسی تولید شده توسط مدل برنامه‌ریزی اتاق‌ها، به دلیل عدم امکان سنجی وجود بازه زمانی گاهی نشدنی است. در مدل برنامه‌ریزی اتاق‌ها در مقابل مدل کیفیت، کنترل تخصیص بازه‌های زمانی نیست و در نتیجه امکان دارد برای جلسه‌های درس‌ها، اتاق داشته باشیم در حالی که بازه زمانی (و یا بازه زمانی مناسب جهت داشتن زمان‌بندی درس‌ها با کیفیت برای استادان و دانشجویان) فراهم نشود. مدل کیفیت (۲-۲-۱-۱) ، با توجه به در نظر گرفتن اتاق‌های موجود و در دسترس، محدود می‌شود. مدل کیفیت بدون در نظر گرفتن لیست اتاق‌های در دسترس می‌تواند به برنامه زمان‌بندی غیر قابل پذیرش منجر شود. بنابراین باید تابع دو هدفه ترکیبی مدل کیفیت و مدل برنامه‌ریزی اتاق‌ها را داشته باشیم. این مدل با استفاده از روابط (۲-۷آ) - (۲-۷و) بیان می‌شود.

حلقه اتصال دو مدل، محدودیت (۲-۷ج) است که جایگزین محدودیت (۲-۱ب) شده و با اضافه شدن متغیر r_s^+ ، تضمین می‌کند جلسه‌های تخصیص یافته درس‌ها به بازه‌های زمانی بیشتر از اتاق‌های در دسترس نشود. محدودیت (۲-۷د)، تضمین می‌کند که تعداد اتاق‌های تخصیص یافته با ظرفیت s با تعداد اتاق‌های در دسترس با ظرفیت s در نهایت برابری کند.

$$\min f_{seats}: \quad f_{seats} = \sum_{s \in S} s.r_s \tag{۲-۷آ}$$

$$f_{qual}: \quad f_{qual} = \sum_{c \in C} ۵.w_c + \sum_{cu \in CU, p \in P} ۲.v_{cu,p} \tag{۲-۷ب}$$

$s.t.$

$$\sum_{c \in C_{\geq s}} x_{c,p} - r_s^+ \leq ۰ \qquad \forall s \in S, p \in P \tag{۲-۷ج}$$

$$r_s^+ - \sum_{s' \in S_s} r_{s'} = ۰ \qquad \forall s \in S \tag{۲-۷د}$$

$$\text{(۲-۱م)} - \text{(۲-۱ج)} \tag{۲-۷ه}$$

$$\text{(۲-۲ج)} - \text{(۲-۲ب)} \tag{۲-۷و}$$

۲-۲-۱-۵ مدل بازه‌های تدریس در مقابل مدل کیفیت

مدل بازه‌های تدریس، با هدف حداقل نمودن بازه‌های زمانی، توجهی به ویژگی زمان‌بندی درس‌ها با کیفیت برای استادان و دانشجویان (هدف مسئله کیفیت) ندارد. بنابراین با ترکیب دو مدل بازه‌های تدریس و مدل کیفیت، انتظار می‌رود آزادی عملی بیشتری در انتخاب بازه‌های زمانی نسبت به مدل بازه‌های تدریس داشته باشیم. حلقه

اتصال دو مدل، محدودیت (۲-۸ج) است که جایگزین محدودیت (۲-۶ب) شده و تضمین می‌کند جلسه‌ها فقط به بازه‌های زمانی در دسترس، تخصیص پیدا کند. این مدل با استفاده از روابط (۲-۸آ) - (۲-۸ه) بیان می‌شود.

$$\min f_{time} = \sum_{p \in P} t_p \qquad (۲-۸آ)$$

$$f_{qual}: \quad f_{qual} = \sum_{c \in C} ۵.w_c + \sum_{cu \in CU, p \in P} ۲.v_{cu,p} \qquad (۲-۸ب)$$

$s.t.$

$$x_{c,p} - t_p \leq ۰ \qquad \forall p \in P, c \in C \qquad (۲-۸ج)$$

$$(۲-۱ب) - (۲-۱م) \qquad (۲-۸د)$$

$$(۲-۶ج) - (۲-۶د) \qquad (۲-۸ه)$$

۲-۲-۱-۶ مدل بازه‌های تدریس در مقابل مدل برنامه‌ریزی اتاق‌ها

زمانی که دانشگاه‌ها تمایل به افزایش تعداد درس‌ها، بالاتر از ظرفیت فعلی (شامل اتاق‌ها و بازه‌های زمانی) دارند، می‌توانند تعداد و ظرفیت اتاق‌ها را افزایش دهند تا در نتیجه آن ظرفیت پذیرش هر بازه زمانی افزایش یابد. مدل دو هدفه مدل بازه‌های تدریس در مقابل مدل برنامه‌ریزی اتاق‌ها، با ترکیب سه مدل کیفیت، مدل برنامه‌ریزی اتاق‌ها و مدل بازه‌های تدریس ایجاد می‌شود.این مدل با استفاده از روابط (۲-۹آ) - (۲-۹و) بیان می‌شود.

$$\min \quad f_{seats}: \quad f_{seats} = \sum_{s \in S} s.r_s \qquad (۲-۹آ)$$

$$min \quad f_{time} = \sum_{p \in P} t_p \qquad (۲-۹ب)$$

$$s.t. \quad (۲-۱ج) - (۲-۱م) \qquad (۲-۹ج)$$

$$(۲-۶ج) - (۲-۶د) \qquad (۲-۹د)$$

$$(۲-۷ج) - (۲-۷د) \qquad (۲-۹ه)$$

$$(۲-۸ج) \qquad (۲-۹و)$$

۲-۲-۲ روش اپسیلون-محدودیت

در این بخش به معرفی روش اپسیلون-محدودیت [۱] می‌پردازیم که یک روش شناخته شده برای حل مسائل چند هدفه است. این روش، در هر مرحله یکی از هدف‌های تابع چند هدفه را انتخاب و به تابع تک هدفه تبدیل و سایر هدف‌ها با در نظر گرفتن یک حد بالا و پایین برای هر کدام به عنوان محدودیت در نظر گرفته می‌شوند و با روش برنامه‌ریزی عدد صحیح حل می‌شود. در نتیجه در هر مرحله یک جواب را داریم. مشکل روش اپسیلون-محدودیت در تخمین درست ε است. روش اپسیلون-محدودیت را در رابطه‌های (۲-۱۰آ)-(۲-۱۰ب) آمده است.

$$min_{x \in X} \quad f_j \tag{۲-۱۰آ}$$

$$subject\ to\ f_k \leq \varepsilon_k \quad k = ۱, ۲, ..., p \quad k \neq j \quad \& \quad where\ \varepsilon \in \mathbb{R}^p \tag{۲-۱۰ب}$$

بهترین جواب حاصل حل تابع چند هدفه به صورت تک هدفه، انتخاب می‌شود.

۲-۲-۳ روش حل

در این بخش، روش‌های حل سه مدل دو هدفه را ارائه شده توسط مقاله مرجع [۱۷] . حل مدل کیفیت مشکل بوده و از ۲۱ نمونه مجموعه آزمایشی، ۴ نمونه آن به صورت بهینه حل نشده است. یکی از روش‌های متداول حل مدل دو هدفه، اپسیلون-محدودیت است. مقادیر f_{seats} و f_{time}، فقط عدد صحیح گسسته بوده و مقدار f_{seats}، به ظرفیت اتاق‌ها وابسته است. نتیجه آزمایش‌ها نشان داده است فاصله بین مقدار کران پایین و کران بالا دو هدف (f_{seats} و f_{time}) نسبتا کم و به مقدارهای مجموعه گسسته موجود (f_{seats} و f_{time}) محدود است. بنابراین با استفاده از روش اپسیلون-محدودیت و با تبدیل تابع هدف اول به محدودیت، تابع هدف دوم را کمینه می‌کنیم. در الگوریتم (۲-۱) ، یکی از تابع‌های (f_{seats} یا f_{time}) تابع هدف f_x و دیگری f_y بوده و دارای مقدارهای گسسته هستند. ابتدا کران پایین f_y را در متغیر min_y قرار داده و پس از اضافه کردن محدودیت $f_y = min_y$ به محدودیت‌ها، کران پایین f_x را به max_x نسبت داده و محدودیت اضافه شده $f_y = min_y$ را حذف می‌کند. کران پایین f_x را در متغیر ε قرار داده و محدودیت $f_x \leq \varepsilon$ را به محدودیت‌ها اضافه می‌کند. به صورت تکراری مقدار Δ افزایش می‌یابد. کران‌های تابع‌های هدف، برای شرط پایان اجرای الگوریتم استفاده شده است. به دلیل حل نشدن مسائل استراتژیک به طور قطعی، برای هر بار تکرار برنامه ۱۵ دقیقه در نظر گرفته شده است. همچنین $\delta = ۲۵$ در نظر گرفته شده است.

[۱] $\varepsilon - constraint$

الگوریتم ۲-۱ الگوریتم $\varepsilon - constraintmethod - f_x vs. f_y$.

Parameters: Δ {//$Discrezation\ of\ f_x$ }
$min_y \leftarrow Minimize(f_x)$ {//$Lower\ limit\ on\ f_y$ }
$Add\ constraint:\ f_y = min_y$
$max_x \leftarrow Minimize(f_x)$ {//$Upper\ limit\ on\ f_x$ }
$Remove\ constraint:\ f_y = min_y$
$\varepsilon \leftarrow Minimize(f_x)$
$Add\ \varepsilon - constraint:\ f_x \leq \varepsilon$
repeat
    $(\hat{f}_x, \hat{f}_y) \leftarrow Minimize(f_y)$ {//$FindPareto - solution$ }
    $\varepsilon \leftarrow \varepsilon + \Delta$
    $Update\ \varepsilon - constraint:\ f_x \leq \varepsilon$
until { $\varepsilon > max_x\ or\ \hat{f}_y < min_y$ } {//$Only\ continue\ inside\ limits$ }

علاوه بر مسئله زمان‌بندی دروس دانشگاهی که در این فصل مدل سازی آن ارائه شد، مسئله مهم دیگر که در مراکز آموزشی حائز اهمیت است، مسئله زمان‌بندی امتحانات است که این مسئله در فصل بعد مطرح و الگوریتمی برای حل آن ارائه می‌دهیم.

فصل ۳

حل مسئله زمان‌بندی امتحانات با استفاده از روش ابتکاری و حریصانه

در کنار مسئله زمان‌بندی دروس که به نحوه تخصیص دروس به استادان و بازه‌های زمانی و اتاق‌ها می‌پردازد، مسئله زمان‌بندی امتحانات نیز دیگر مسئله مهم برای تمامی مراکز آموزشی است که هدف آن تخصیص امتحانات به بازه‌های زمانی با هدف برگزاری هرچه بهتر امتحانات و کسب حداکثر رضایت دانشجویان و استادان و حفظ انسجام برنامه حضور استادان و همچنین داشتن فاصله مناسب بین امتحانات هر دانشجو است.

در این فصل روشی برای حل مسئله جدول زمان‌بندی امتحانات دانشگاهی با استفاده از روش حریصانه و روش ابتکاری مورد بررسی قرار می‌گیرد. هدف اصلی این روش همانند سایر الگوریتم‌های موجود ([۷، ۸])، نداشتن نقض محدودیت سخت و کمینه نمودن نقض محدودیت نرم می‌باشد.

۳-۱ مقدمه

به صورت معمول و در روال دستی، روند تخصیص دروس به بازه‌های زمانی امتحانات بر اساس چارت درسی ترم-رشته است. بنابراین زمان برگزاری امتحانات دروس یک ترم هر رشته نباید با دروس همان ترم آن رشته تداخل داشته و باید فاصله مناسب بین روزهای امتحانی رعایت گردد. در این حالت اگر دانشجویی درسی از ترم گذشته را انتخاب نماید احتمال این که با چند امتحان در یک روز ولی به صورت غیر هم زمان و یا در شرایط سخت‌تر چند امتحان در یک ساعت امتحانی به طور هم زمان برگزار شود، وجود خواهد داشت.

با این فرض که در حالت عادی اکثر دانشجویان بر اساس چارت درسی ترم-رشته انتخاب واحد می‌نمایند،

در این فصل روشی برای برنامه‌ریزی ارائه خواهد شد.

در این روش، ابتدا درس‌ها بر اساس داده‌های ورودی که انتخاب واحدهای دانشجویان است با روش ابتکاری گروه‌بندی می‌شوند. هر گروه از دروس، شامل درس‌هایی هستند که امکان برگزاری به صورت هم زمان به نحوی که برای دانشجویان مشکل چند امتحان در یک بازه زمانی به وجود نیاید، وجود دارد. در مرحله بعد، با توجه به شرایط مرکز آموزشی (تعداد روزهای امتحانی، بیشینه ظرفیت برگزاری امتحان و تعداد گروه‌های درسی) در مورد نحوه اختصاص بازه‌های زمانی به دروس، مبتنی بر روش حریصانه با هدف کمینه نمودن نقض محدودیت‌ها و بیشینه نمودن میزان رضایت دانشجویان تصمیم‌گیری می‌شود. در روش حریصانه، لیست میزان نقض محدودیت‌ها هر دو گروه درسی به صورت صعودی مرتب می‌شود که مبنای روال انتخاب برای اختصاص به بازه‌های زمانی خواهد بود. در ادامه فصل، مباحث زیر را خواهیم داشت :

- **محدودیت‌های سخت در مسئله زمان‌بندی امتحانات :** در این بخش، به بیان محدودیت‌های سخت در نظر گرفته شده برای مسئله زمان‌بندی امتحانات در روش پیشنهادی می‌پردازیم. این بخش به دلیل این که در یک برنامه زمان‌بندی امتحانات شدنی و قابل اجرا، محدودیت‌های سخت باید لحاظ شود، اهمیت خواهد داشت.

- **محدودیت نرم در مسئله زمان‌بندی امتحانات :** در این بخش، به بیان محدودیت‌های نرم در نظر گرفته شده برای مسئله زمان‌بندی امتحانات در روش پیشنهادی می‌پردازیم. این بخش به دلیل این که در میزان کیفیت و در نتیجه آن میزان رضایت دانشجویان از برنامه زمان‌بندی امتحانات، به میزان پوشش محدودیت‌های نرم بستگی دارد، اهمیت خواهد داشت.

- **محدودیت‌های لحاظ شده :** در این بخش، واژه‌ها و اصطلاحات الگوریتم روش پیشنهادی با همراه توضیح مربوط به آن واژه بیان شده است که در واقع الفبای روش پیشنهادی محسوب می‌شود و برای درک روش پیشنهادی الزامی است.

- **پارامترهای مسئله زمان‌بندی امتحانات :** در این بخش، پارامترهای روش پیشنهادی برای مسئله زمان‌بندی امتحانات بیان می‌شود. به دلیل این که مواردی مانند بیشینه ظرفیت برگزاری امتحانات، تعداد روزهای برگزاری امتحانات، بیشینه ظرفیت مورد نیاز برای گروه‌های درسی، تعداد کل دانشجویان و سایر پارامترهای بیان شده در نحوه برنامه‌ریزی امتحانات تاثیرگذار خواهند بود و حالت‌های متفاوتی را در مسئله زمان‌بندی امتحانات به وجود می‌آورد. بنابراین نحوه برخورد و روش حل هر حالت با حالت‌های دیگر متفاوت خواهد بود.

- **الگوریتم روش پیشنهادی :** در این بخش، الگوریتم‌های حل مسئله زمان‌بندی امتحانات در روش پیشنهادی به طور کامل همراه با جزئیات کامل مطرح می‌شود.

۳-۲ محدودیت‌های سخت در مسئله زمان‌بندی امتحانات

محدودیت سخت و نرم تعریف شده در زمان‌بندی امتحانات، از دانشگاهی(مدرسه) به دانشگاه(مدرسه) دیگر و حتی از دانشکده‌ای به دانشکده‌ای دیگر در همان دانشگاه می‌تواند متفاوت و وابسته به دیدگاه مسئولین آن مرکز آموزشی، امکانات موجود در دسترس آنها و سایر موارد است. بنابراین طراحی و ارائه روشی جامع که بتواند همه خواسته‌ها را پوشش دهد غیرممکن است. محدودیت‌های سخت در این پژوهش همانند روش‌های متداول استفاده شده برای حل مسئله زمان‌بندی امتحانات است. در ادامه به بیان بخشی از آنها می پردازیم.

۱. یک دانشجو نباید دو یا چند امتحان در یک نوبت امتحانی داشته باشد.

۲. برنامه‌ریزی امتحانات طوری تنظیم شود تا دانشجو در یک روز سه امتحان نداشته باشد.

۳. تعداد دانشجویان انتخاب شده در هر بازه زمانی نباید بیشتر از ظرفیت سالن امتحانات باشد.

۴. دروسی که نمونه سوال امتحانی یکسان دارند باید در یک بازه زمانی قرار بگیرند.

۳-۳ محدودیت نرم در مسئله زمان‌بندی امتحانات

محدودیت‌های نرم زمان‌بندی امتحانات به شرح ذیل هستند :

۱. برنامه‌ریزی امتحانات طوری تنظیم شود تا دانشجو در یک روز چند امتحان نداشته باشند.

۲. در صورتی که دانشجویی دو امتحان در یک روز داشت ترجیحا در ساعات اول و آخر همان روز باشد و فاصله‌ای بین آنها باشد.

۳-۴ محدودیت‌های لحاظ شده

تعریف واژه‌ها و اصطلاحات مورد نیاز در الگوریتم در جدول ۳-۱ آمده است.

جدول ۳-۱: واژه‌ها و اصطلاحات الگوریتم

ردیف	اصطلاح	توضیح
۱	انتخاب واحد دانشجویان	با ساختار دو ستون (کد درس و کد دانشجو) به عنوان ورودی
۲	تعداد روزهای امتحانی	تعداد روزهای امتحانی جهت تخصیص
۳	ظرفیت پذیرش نوبت امتحانی	حداکثر ظرفیت پذیرش نوبت امتحانی
۴	لیست فراوانی تکرار دروس	فراوانی تعداد تکرار هر درس(تعداد انتخاب شده توسط دانشجویان)
۵	لیست انتخاب واحد تفکیکی دانشجو	مجموعه انتخاب واحد دانشجویان به تفکیک هر دانشجو
۶	تعداد دروس	تعداد دروس انتخابی دانشجویان
۷	ماتریس مجاورت دروس	ماتریس مجاورت دروس شامل صفر و یک
۸	فراوانی دروس	فراوانی دروس (با هدف مشخص شدن میزان دانشجویان مشترک بین هر دو درس)
۹	گروه‌بندی اولیه دروس	گروه‌بندی اولیه دروس با ساختار دو ستون دارای کد درس و کد گروه
۱۰	تعداد گروه‌های گروه‌بندی اولیه دروس	بر اساس گروه‌بندی اولیه دروس
۱۱	فراوانی ماتریس مجاورت دروس	میزان اشتراک‌های دروس با هم به تفکیک هر درس
۱۲	برنامه امتحانی دانشجویان	ذخیره‌سازی برنامه امتحانی دانشجویان
۱۳	میزان نقض محدودیت‌ها	میزان نقض محدودیت‌ها بر اساس نوع نقض آن (بر اساس موارد ذکر شده در جدول ۳-۲) به ازاء هر دانشجو
۱۴	برنامه نهایی امتحانات	گروه بندی نهایی دروس با ساختار دارای دو ستون کد درس و کد گروه
۱۵	بیشینه تعداد بازه زمانی گروه‌ها	بیشینه تعداد بازه زمانی مورد نیاز یک گروه در گروه‌بندی اولیه دروس

الگوریتم مورد استفاده را می‌توان ترکیبی از روش ابتکاری و روش حریصانه در نظر گرفت. ابتدا با روشی ابتکاری اقدام به گروه‌بندی دروس بر حسب انتخاب واحدهای انجام شده توسط دانشجویان می‌کنیم. نتیجه این گروه‌بندی (همان خوشه‌بندی است)، گروه‌هایی از دروس است که می‌توانند به صورت هم زمان در یک بازه زمانی برگزار گردند و هیچ دانشجویی در آن بازه زمانی دو یا چند امتحانی نباشد. در مرحله بعد با روش حریصانه، چیدمان گروه‌ها را با هدف داشتن کمترین نقض محدودیت نرم (که همان چند امتحانی در یک روز است) بررسی می‌کنیم.

۳-۴-۱ پارامترهای مسئله زمان‌بندی امتحانات

پارامترهای تاثیرگذار در مسئله زمان‌بندی امتحانات به شرح زیر است :

۱. تعداد روزهای امتحانی

جدول ۳-۲: نقض محدودیت‌ها

کد نقض	شرح
۱	دانشجو چند امتحان در یک بازه زمانی داشته باشد.
۲	دانشجو سه امتحان در هر سه بازه زمانی همان روز داشته‌باشد.
۳	دانشجو دو امتحان در بازه زمانی اول و دوم همان روز داشته‌باشد.
۴	دانشجو دو امتحان در بازه زمانی دوم و سوم همان روز داشته‌باشد.
۵	دانشجو دو امتحان در بازه زمانی اول و سوم همان روز داشته‌باشد.

۲. ظرفیت پذیرش نوبت امتحانی

۳. تعداد گروه‌های اولیه دروس

۴. بیشینه ظرفیت یک درس (بیشینه ظرفیت مورد نیاز برای یک درس)

۵. تعداد بازه‌های زمانی گروه‌ها(تعداد بازه‌های زمانی مورد نیاز برای گروه‌ها تعیین شده بر اساس ظرفیت پذیرش نوبت امتحانی)

۶. بیشینه بازه زمانی یک گروه(بیشینه بازه زمانی مورد نیاز برای یک گروه تعیین شده بر اساس ظرفیت پذیرش نوبت امتحانی)

۷. جمع ظرفیت مورد نیاز همه گروه‌ها

توضیح تمامی حالت‌های برگزاری امتحانات ذکر شده جدول ۳-۳ به شرح زیر است.

۱. در این حالت یک نوبت امتحانی هر روز را به یک گروه به نحوی اختصاص می‌دهیم که هیچ گونه نقض محدودیت سخت و نرم نداشته باشیم.

۲. در این حالت دو یا سه نوبت امتحانی هر روز را به یک گروه اختصاص که در نتیجه هیچ گونه نقض محدودیت سخت و نرم نخواهیم داشت. در این حالت فرض بر این است که حداکثر بازه‌های امتحانی مورد نیاز با توجه به ظرفیت پذیرش نوبت امتحانی، سه بازه است.

۳. در این حالت بازه‌های زمانی کافی برای تخصیص گروه‌ها را داریم اما برعکس حالت‌های قبلی(حالت ۱ و ۲) در هر روز ممکن است دو یا سه گروه را داشته باشیم که در نتیجه فقط نقض محدودیت نرم را خواهیم داشت. ضمنا در این حالت تعداد روزهای امتحانی بزرگتر یا مساوی با تعداد گروه‌های اولیه دروس است و بیشینه بازه زمانی یک گروه می‌تواند بیشتر از ۳ بازه زمانی باشد.

جدول ۳-۳: پارامترهای مسئله زمان‌بندی امتحانات

کد حالت	شرط اول	شرط دوم
۱	تعداد روزهای امتحانی=>تعداد گروه‌های اولیه دروس	ظرفیت پذیرش نوبت امتحانی=>بیشینه ظرفیت یک گروه
۲	تعداد روزهای امتحانی=>تعداد گروه‌های اولیه دروس	ظرفیت پذیرش نوبت امتحانی<بیشینه ظرفیت یک گروه=<ظرفیت پذیرش نوبت امتحانی×۳
۳	تعداد روزهای امتحانی=>تعداد گروه‌های اولیه دروس	تعداد روزهای امتحانی<تعداد بازه‌های زمانی گروه‌ها=<تعداد روزهای امتحانی×۳
۴	تعداد روزهای امتحانی<تعداد بازه‌های زمانی گروه‌ها=<تعداد روزهای امتحانی×۳	ظرفیت پذیرش نوبت امتحانی=>بیشینه ظرفیت یک گروه
۵	تعداد روزهای امتحانی<تعداد بازه‌های زمانی گروه‌ها=<تعداد روزهای امتحانی×۳	۱<بیشینه بازه زمانی یک گروه=<۳
۶	تعداد روزهای امتحانی<تعداد بازه‌های زمانی گروه‌ها=<تعداد روزهای امتحانی×۳	بیشینه بازه زمانی یک گروه>۳
۷	ظرفیت پذیرش نوبت امتحانی<جمع ظرفیت مورد نیاز همه گروه‌ها=<ظرفیت پذیرش نوبت امتحانی×تعداد روزهای امتحانی×۳	تعداد بازه‌های زمانی گروه‌ها>تعداد روزهای امتحانی×۳
۸	جمع ظرفیت مورد نیاز همه گروه‌ها>ظرفیت پذیرش نوبت امتحانی×تعداد روزهای امتحانی×۳	-

کد حالت	نقض محدودیت سخت	نقض محدودیت نرم	بررسی شده
۱	ندارد	ندارد	شده
۲	ندارد	ندارد	شده
۳	ندارد	دارد	نشده
۴	ندارد	دارد	شده
۵	ندارد	دارد	شده
۶	ندارد	دارد	نشده
۷	دارد	دارد	نشده
۸	-	-	نشده

۴. در این حالت بازه‌های زمانی کافی برای تخصیص گروه‌ها را داریم اما با توجه به ظرفیت پذیرش نوبت امتحانی هر گروه فقط به یک بازه زمانی اختصاص داده خواهد شد که در نتیجه فقط نقض محدودیت نرم را خواهیم داشت.

۵. در این حالت بازه‌های زمانی کافی برای تخصیص گروه‌ها را داریم اما با توجه به ظرفیت پذیرش نوبت امتحانی هر گروه به دو یا سه بازه زمانی اختصاص داده خواهد شد که در نتیجه فقط نقض محدودیت نرم را خواهیم داشت. ضمنا در این حالت تعداد روزهای امتحانی کمتر از تعداد گروه‌های اولیه دروس است و بیشینه بازه زمانی یک گروه نهایتا محدود به ۳ بازه زمانی است.

۶. در این حالت بازه‌های زمانی کافی برای تخصیص گروه‌ها را داریم اما با توجه به ظرفیت پذیرش نوبت امتحانی هر گروه به دو یا سه بازه زمانی اختصاص داده خواهد شد که در نتیجه فقط نقض محدودیت نرم را خواهیم داشت. ضمنا در این حالت تعداد روزهای امتحانی کمتر از تعداد گروه‌های اولیه دروس است و بیشینه بازه زمانی یک گروه بیشتر از ۳ بازه زمانی است.

۷. در این حالت ظرفیت کل پیش بینی شده (ظرفیت پذیرش نوبت امتحانی×تعداد روزهای امتحانی×۳) جوابگوی جمع ظرفیت مورد نیاز همه گروه‌ها است ولی برخلاف حالت‌های بالا باید دو یا چند گروه را در یک بازه زمانی تخصیص دهیم که در نتیجه هم نقض محدودیت سخت و هم نقض محدویت نرم را خواهیم داشت.

۸. در این حالت چون کل ظرفیت برگزاری امتحانات از ظرفیت مورد نیاز کمتر است، برگزاری امتحانات با پیش فرض‌های فعلی(تعداد روزهای تعریف شده و ظرفیت هر نوبت امتحانی) امکان پذیر نیست.

۳-۵ الگوریتم روش پیشنهادی

در الگوریتم پیش شرط‌های لازم برای بررسی هر یک از حالت‌های بررسی شده در جدول (۳-۳) را به طور کامل بررسی می‌نماییم تا از خطاهای احتمالی جلوگیری شود. ضمنا در برنامه به طور پیش فرض برای هر روز امتحانی سه نوبت امتحانی در نظر گرفته شده است.

لیست الگوریتم‌ها به شرح زیر است که در زیربخش‌های (۳-۵-۱)-(۳-۵-۳) آنها را بررسی می‌کنیم.

۱. الگوریتم تولید گروه‌بندی اولیه دروس(الگوریتم ۳-۲)

۲. الگوریتم تولید برنامه نهایی امتحانات(الگوریتم‌های ۳-۳)

الگوریتم ۳-۱ تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه

ورودی: انتخاب واحد دانشجویان، تعداد روزهای امتحانی، ظرفیت پذیرش نوبت امتحانی
خروجی: برنامه نهایی امتحانات، میزان نقض محدودیت‌ها، لیست برنامه امتحانی دانشجویان
۱: الگوریتم تولید گروه‌بندی اولیه دروس (الگوریتم ۳-۲)
۲: الگوریتم تولید برنامه نهایی امتحانات (الگوریتم ۳-۳)

۳. الگوریتم تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه(الگوریتم ۳-۱)

توضیح ورودی‌ها و خروجی‌های استفاده شده الگوریتم‌ها، در جدول ۳-۱ آمده‌است.

۳-۵-۱ الگوریتم تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه

ورودی و خروجی الگوریتم تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه به شرح زیر است : **ورودی الگوریتم**

۱. انتخاب واحد دانشجویان

۲. تعداد روزهای امتحانی

۳. ظرفیت پذیرش نوبت امتحانی

خروجی الگوریتم

۱. برنامه نهایی امتحانات

۲. میزان نقض محدودیت‌ها

۳. لیست برنامه امتحانی دانشجویان

این تابع اصلی برنامه است و هدف آن بررسی جریان انجام فراخوانی توابع موجود در برنامه است. این الگوریتم بر اساس توابع معرفی شده در الگوریتم (۳-۲) (بخش ۳-۵-۲) و الگوریتم (۳-۳) (بخش ۳-۵-۳) برنامه نهایی امتحانات دانشجویان را تولید می‌کند. این الگوریتم در (۳-۱) آمده است.

۳-۵-۲ الگوریتم تولید گروه‌بندی اولیه دروس

ورودی و خروجی الگوریتم گروه‌بندی اولیه دروس به شرح زیر است :

ورودی الگوریتم

۱. انتخاب واحد دانشجویان

خروجی الگوریتم

۱. لیست فراوانی تکرار دروس

۲. لیست انتخاب واحد تفکیکی دانشجو

۳. تعداد دروس

۴. ماتریس مجاورت دروس

۵. فراوانی دروس

۶. گروه‌بندی اولیه دروس

۷. فراوانی گروه‌بندی اولیه دروس

تنها داده ورودی الگوریتم گروه‌بندی اولیه دروس، انتخاب واحد دانشجویان شامل شماره دانشجویی و شماره درس انتخاب واحد شده توسط دانشجو است. دانشجو می‌تواند یک درس یا چندین درس را انتخاب واحد کرده باشد. هدف اصلی الگوریتم، گروه‌بندی اولیه دروس به نحوی که دروس هر گروه، دانشجوی مشترک وجود نداشته باشند. برای رسیدن به این هدف باید سه مرحله را داشته باشیم :

مرحله اول

در مرحله اول لیست انتخاب واحد تفکیکی دانشجو، لیست فراوانی تکرار دروس و تعداد دروس را به عنوان بخشی از خروجی‌های الگوریتم بر اساس ورودی الگوریتم (انتخاب واحدهای دانشجویان) ایجاد می‌نماییم.

مرحله دوم

در مرحله دوم، ماتریس مجاورت دروس و فراوانی دروس را بر اساس لیست انتخاب واحد تفکیکی دانشجو و تعداد دروس بدست می‌آوریم. ماتریس مجاورت دروس و فراوانی دروس به منظور تشخیص روابط حاکم میان هر دو درس تشکیل می‌شود به نحوی که یک بودن مقدار متناظر دو درس در ماتریس مجاورت به این معنی است

الگوریتم ۳-۲ تولید گروه‌بندی اولیه دروس

ورودی: انتخاب واحد دانشجویان

خروجی: لیست فراوانی تکرار دروس، لیست انتخاب واحد تفکیکی دانشجو، تعداد دروس، ماتریس مجاورت دروس، فراوانی دروس، گروه‌بندی اولیه دروس، فراوانی گروه‌بندی اولیه دروس

۱: لیست انتخاب واحد تفکیکی دانشجو، لیست فراوانی تکرار دروس و تعداد دروس را بر اساس انتخاب واحد دانشجویان تشکیل می‌دهیم.

۲: به ازاء هر دانشجو و بر اساس لیست انتخاب واحد تفکیکی دانشجو، مقدار متناظر هر دو درس انتخاب شده توسط دانشجو(یان) در ماتریس مجاورت دروس دارای مقدار یک شود.

۳: به ازاء هر دانشجو و بر اساس لیست انتخاب واحد تفکیکی دانشجو، مقدار فعلی متناظر هر دو درس انتخاب شده به طور هم زمان در فراوانی دروس یک واحد افزایش یابد.

۴: تا زمانی که به تمامی دروس گروهی تخصیص نیافته است گام‌های ۴ تا ۸ را تکرار می‌کنیم.

۵: در هر بار اجرای حلقه، فقط یک درس که در صدر لیست فراوانی تکرار دروس قرار دارد را بررسی می‌کنیم.

۶: برای هر درس انتخابی از اول لیست فراوانی تکرار دروس، کمینه شماره گروه مجاز را به آن درس در گروه‌بندی اولیه دروس اختصاص می‌دهیم.

۷: روز اختصاص داده شده به آن درس را برای کلیه درس‌هایی که با آن درس در ماتریس مجاورت دروس همسایگی دارند را غیرفعال می‌کنیم.

۸: دروسی که با آن درس همسایگی ندارند را در مراحل بعد الگوریتم بررسی می‌نماییم.

۹: فقط رکورد اول که همان کد درس پردازش شده است را از لیست فراوانی تکرار دروس حذف می‌نماییم.

که دو درس مدنظر دانشجوی مشترک دارند. به بیانی دیگر، حداقل یک دانشجو هر دو درس را هم زمان انتخاب واحد کرده است.

مرحله سوم

در نهایت و در مرحله سوم، گروه‌بندی اولیه دروس و فراوانی گروه‌بندی اولیه دروس را بدست می‌آوریم. در نتیجه گروه‌بندی اولیه دروس، نقض محدودیت نرم و سخت را نخواهیم داشت. با روشی ابتکاری به هر درس، شماره گروه(خوشه) از شماره یک و بالاتر اختصاص دهیم. در این روش ابتکاری، ترتیب انتخاب دروس برحسب تعداد دانشجوی انتخاب واحد کرده برای آن درس به صورت حریصانه و نزولی است. ابتدا به هر درس یک شماره گروه را اختصاص می‌دهیم. شماره گروه جاری را برای سایر درس‌هایی که با درس جاری ارتباط داشته (به عبارت دیگر، هر جفت از درس‌ها که به طور هم زمان توسط دانشجو(یان) انتخاب واحد شده باشند)، حذف می‌کنیم. این روند را تا زمانی که به همه درس‌ها، شماره گروه اختصاص پیدا نکرده است ادامه می‌یابد. الگوریتم تولید گروه‌بندی اولیه دروس در (۳-۲) آمده است.

۳-۵-۳ الگوریتم تولید برنامه نهایی امتحانات

ورودی الگوریتم

۱. تعداد روزهای امتحانی

۲. تعداد دروس

۳. گروه‌بندی اولیه دروس

۴. ظرفیت پذیرش نوبت امتحانی

۵. ماتریس مجاورت دروس

۶. فراوانی دروس

۷. لیست فراوانی تکرار دروس

۸. فراوانی ماتریس مجاورت دروس

خروجی الگوریتم

۱. برنامه نهایی امتحانات

۲. میزان نقض محدودیت‌ها

۳. لیست برنامه امتحانی دانشجویان

وظیفه اصلی این الگوریتم، تولید برنامه نهایی امتحانات، میزان نقض محدودیت‌ها و لیست برنامه امتحانی دانشجویان است که برای حالت‌های ۱، ۲، ۴و۵ از جدول (۳-۳) پیاده‌سازی شده‌است. توضیح الگوریتم به شرح ذیل است.

برای حالت ۱، برنامه نهایی امتحانات را برابر با گروه‌بندی اولیه دروس قرار می‌دهیم. در این حالت با توجه به این که تعداد روزهای امتحانی جوابگوی تعداد گروه‌بندی اولیه و هم چنین ظرفیت پذیرش نوبت امتحانی جوابگوی حداکثر ظرفیت مورد نیاز برای یک گروه هستند، به هر گروه فقط نوبت اول امتحانی هر روز را اختصاص می‌دهیم.

الگوریتم ۳-۳ تولید برنامه نهایی امتحانات

ورودی: تعداد روزهای امتحانی، تعداد دروس، گروه‌بندی اولیه دروس، ظرفیت پذیرش نوبت امتحانی، ماتریس مجاورت دروس،فراوانی دروس، لیست فراوانی تکرار دروس، فراوانی ماتریس مجاورت دروس

خروجی: برنامه نهایی امتحانات، میزان نقض محدودیت‌ها، لیست برنامه امتحانی دانشجویان

۱: تولید برنامه نهایی امتحانات بر اساس حالت‌های ۱، ۲، ۴ و ۵ جدول (۳-۳)

۱. **برای حالت ۱**، برنامه نهایی امتحانات را مشابه با گروه‌بندی اولیه دروس، در یک روز و فقط در یک بازه زمانی قرار دهید.

۲. **برای حالت ۲**، برنامه نهایی امتحانات مشابه با گروه‌بندی اولیه دروس، در یک روز و در دو یا سه بازه زمانی با توجه به ظرفیت پذیرش امتحانی و ظرفیت مورد نیاز هر گروه قرار دهید.

۳. **برای حالت ۴**، میزان نقض محدودیت نرم تخصیص چند گروه، در یک روز را محاسبه و بر اساس روش حریصانه و صعودی مرتب سازی نمایید. سپس گروه‌ها را فقط در یک بازه زمانی با کمترین میزان نقض تخصیص دهید.

۴. **برای حالت ۵**، گروه‌ها با تعداد مورد نیاز ۳ بازه زمانی را به یک روز تخصیص داده و سپس میزان نقض محدودیت نرم تخصیص چند گروه، در یک روز را محاسبه و بر اساس روش حریصانه و صعودی مرتب سازی نمایید. سپس گروه‌ها را در دو یا سه بازه زمانی با کمترین میزان نقض تخصیص دهید.

۶: محاسبه میزان نقض محدودیت‌ها به ازاء هر دانشجو بر اساس جدول (۳-۲)

۷: تولید برنامه امتحانی هر دانشجو بر اساس برنامه نهایی امتحانات

برای حالت ۲، مشابه حالت ۱ است با این تفاوت که چون ظرفیت پذیرش نوبت امتحانی جوابگوی حداکثر ظرفیت مورد نیاز برای یک گروه نیست، باید دروس آن گروه در دو یا سه نوبت امتحانی با توجه به ظرفیت پذیرش نوبت امتحانی همان روز توزیع شود.

برای حالت ۴، تعداد کل بازه‌های زمانی مورد نیاز از تعداد روزهای امتحانی بیشتر و ظرفیت پذیرش نوبت امتحانی جوابگوی حداکثر ظرفیت مورد نیاز برای گروه‌ها است، به هر گروه فقط یک بازه زمانی را تخصیص خواهیم داد. در این حالت تلاش بر این است نقض محدودیت نرم را با توجه به این که در یک روز دو یا سه گروه متفاوت را داریم، به حداقل کاهش دهیم. ترتیب انتخاب درس‌ها، بر اساس لیست مرتب شده فراوانی دروس به صورت حریصانه و صعودی از میزان اشتراک بین گروه‌ها را است تا کمترین تداخل چیدمان گروه‌ها را داشته باشیم.

برای حالت ۵، ابتدا گروه‌هایی که به سه نوبت امتحانی نیاز دارند را به یک روز کامل اختصاص داده می‌شود. در نتیجه در این گروه‌ها نقض محدودیت نرم را نخواهیم داشت. در این حالت تلاش بر این است نقض محدودیت

نرم را با توجه به این که در یک روز دو یا سه گروه متفاوت را داریم، به حداقل کاهش دهیم. ترتیب انتخاب درس‌ها، بر اساس لیست مرتب شده فراوانی دروس به صورت حریصانه و صعودی از میزان اشتراک بین گروه‌ها را است تا کمترین تداخل چیدمان گروه‌ها را داشته باشیم.

در ادامه، انجام محاسبات میزان نقض محدودیت‌ها بر اساس ساختار ذکر شده در جدول ۳-۲ صورت می‌پذیرد. در این ساختار ۵ نوع نقض محدودیت لحاظ شده است. جزء در حالت ۷ ذکر شده در جدول (۳-۳) (لیست حالت‌های کلی در برنامه) که نقض محدودیت سخت داریم، در سایر حالت‌ها با توجه به گروه‌بندی انجام شده در مراحل قبل هیچ گونه نقض محدودیت سخت را نخواهیم داشت. در جدول (۳-۳) (لیست حالت‌های کلی در برنامه)، وجود یا عدم وجود نقض محدودیت در دو نوع سخت و نرم بیان شده است. الگوریتم تولید برنامه نهایی امتحانات در (۳-۳) آمده است.

۳-۵-۴ مثال کاربردی الگورتیم‌های ارائه شده در روش ابتکاری

در این بخش، به ارائه مثال کاربردی برای الگورتیم‌های ارائه شده در روش ابتکاری می‌پردازیم.

مثال ۳-۵-۱. در این مثال، زمان‌بندی امتحانات بر اساس انتخاب واحدهای صورت گرفته ۱۰۴۷ دانشجو و برای ۲۰۷ درس انجام شده و نتایج آن را مشاهده می‌نمایید. تعداد روزهای امتحانی، ۱۵ و ظرفیت پذیرش نوبت امتحانی ۴۵۰ نفر در نظر گرفته شده است. لیست درس‌ها، در جدول ۳-۴ آمده است. ابتدا ماتریس مجاورت با استفاده از الگوریتم (۳-۲) بدست می‌آوریم. نمونه‌ای از گراف ماتریس مجاورت در شکل (۳-۱) آمده است. گروه‌بندی اولیه دروس با استفاده از الگوریتم (۳-۲) بدست می‌آید و دروس به ۱۸ گروه تقسیم می‌شود. دانشجویان هر گروه، فقط یک درس باید انتخاب واحد کرده باشند. لیست دروس به همراه شماره گروه‌بندی اولیه در جدول (۳-۵) آمده است. برنامه نهایی امتحانات با استفاده از الگوریتم (۳-۳) بدست می‌آید. برنامه نهایی امتحانات در جدول (۳-۶) آمده است. با توجه به این که تعداد روزهای امتحانی، ۱۵ و گروه‌بندی اولیه دروس ۱۸ و ظرفیت پذیرش نوبت امتحانی ۴۵۰ نفر است، دروس در برنامه نهایی امتحانات در ۲۳ نوبت امتحانی برگزار می‌شود که در نتیجه فقط نقض محدودیت نرم (دو امتحان در دو نوبت امتحانی یک روز) را خواهیم داشت. از بین ۱۰۴۷ دانشجو، ۶۸ نفر، دو امتحان در دو نوبت امتحانی یک روز خواهند داشت. بخشی از تداخل برنامه امتحانات را در جدول (۳-۷) امده است.

جدول ۳-۴: لیست درس‌ها در مثال (۳-۵-۱)

کد درس	درس
۱	اجرای سازه‌های آبی
۲	اجرای سازه‌های بتنی
۳	اجرای سازه‌های بتنی (۲)
۴	اجزاء ساختمان
۵	اجزاء ماشین (۳)
۶	اجزای ماشین
۷	اخلاق اسلامی (مبانی و مفاهیم)
۸	اخلاق مهندسی
۹	استاتیک
۱۰	اصول سرپرستی
۱۱	اصول مدیریت ساخت
۱۲	اقتصاد(۱) (خرد)
۱۳	اقتصاد(۲) (کلان)
۱۴	الکترومغناطیس
۱۵	الکترونیک خودرو
۱۶	الکترونیک صنعتی
۱۷	الکترونیک عمومی
۱۸	اندیشه اسلامی(۱) (مبدأ و معاد)
۱۹	اندیشه اسلامی(۲) (نبوت و امامت)
۲۰	انقلاب اسلامی ایران
۲۱	ایستایی
۲۲	ایستایی (۱)
۲۳	ایستایی (۲)
۲۴	ایمنی در برق
۲۵	ایمنی کارگاه
۲۶	آشنایی با زلزله و اثرآن برسازه‌ها
۲۷	آشنایی با فرهنگ و ارزش های دفاع مقدس
۲۸	آشنایی با معماری اسلامی (۱)
۲۹	آمار و احتمالات
۳۰	آمار و احتمالات مهندسی
۳۱	آئین نامه های ساختمانی
۳۲	بتن پیش ساخته و پیش تنیده
۳۳	بررسی سیستم‌های قدرت ۱
۳۴	بررسی سیستم‌های قدرت۲
۳۵	برق تاسیسات
۳۶	برنامه سازی پیشرفته (۲)
۳۷	برنامه سازی سیستم
۳۸	برنامه‌نویسی رایانه
۳۹	برنامه‌نویسی کامپیوتر
۴۰	برنامه‌نویسی ماشین‌های کنترل عددی
۴۱	برنامه‌نویسی مبتنی بروب
۴۲	پایگاه داده‌ها
۴۳	پرسپکتیو
۴۴	تاریخ تحلیلی صدر اسلام
۴۵	تاسیسات الکتریکی و پروژه
۴۶	تاسیسات تهویه مطبوع
۴۷	تجزیه و تحلیل سیگنال‌ها و سیستم‌ها
۴۸	تجهیزات پست و نیروگاه
۴۹	تحلیل رفتار قطعات بتنی
۵۰	تحلیل مدارهای الکترونیکی
۵۱	تحلیل مدارهای الکتریکی
۵۲	تحلیل مقدماتی سازه‌ها
۵۳	ترمودینامیک
۵۴	ترمودینامیک (۲)
۵۵	تعمیرو نگهداری ساختمان
۵۶	تفسیر موضوعی قرآن
۵۷	تکنولوژی انتقال قدرت اتوماتیک
۵۸	تکنولوژی انتقال قدرت معمولی
۵۹	تکنولوژی سوخت رسانی موتورهای بنزینی
۶۰	تکنولوژی سوخت و سوخت رسانی گازی و کارگاه
۶۱	تکنولوژی سیستم‌های الکتریکی و الکترونیکی خودرو
۶۲	تکنولوژی سیستم‌های هدایت و کنترل خودرو
۶۳	تکنولوژی عایق ها و فشارقوی
۶۴	تکنولوژی مالتی پلکس
۶۵	تکنولوژی مولد قدرت
۶۶	تکنولوژی و بازرسی جوش و کارگاه
۶۷	تکنولوژی و کارگاه قالب بندی و آرماتور
۶۸	تکنولوژی وکارگاه قالب سازی
۶۹	تکنیک پالس
۷۰	تمرین های معماری (۱)
۷۱	تمرین های معماری (۲)
۷۲	تنظیم شرایط محیطی (۱)
۷۳	تولید و نیروگاه
۷۴	حسابداری دولتی(۱)
۷۵	حسابداری شرکت‌ها (۲)
۷۶	حسابداری صنعتی (۲)
۷۷	حسابداری مالی
۷۸	حسابداری مالیاتی
۷۹	حسابرسی (۱)
۸۰	حفاظت ماشین‌های الکتریکی
۸۱	خطوط انتقال مخابراتی
۸۲	دانش خانواده و جمعیت
۸۳	درک و بیان معماری (۲)
۸۴	دستگاه‌های برودتی
۸۵	دستگاه‌های تهویه مطبوع
۸۶	دینامیک
۸۷	ذخیره و بازیابی اطلاعات
۸۸	راهسازی و روسازی
۸۹	رسم فنی و نقشه کشی ساختمان
۹۰	رله و حفاظت
۹۱	روسازی‌های بتنی و آسفالتی
۹۲	روستا (۱)
۹۳	روشنایی فنی
۹۴	روش‌های آماری
۹۵	روش‌های تعمیر و نگهداری ساختمان
۹۶	روش‌های مرمت ابنیه
۹۷	ریاضی ۲
۹۸	ریاضی عمومی
۹۹	ریاضی عمومی (۱)
۱۰۰	ریاضی عمومی (۲)
۱۰۱	ریاضی کاربردی
۱۰۲	ریاضی گسسته
۱۰۳	ریاضیات مهندسی
۱۰۴	زبان فارسی
۱۰۵	زبان تخصصی
۱۰۶	زبان تخصصی نرم افزار
۱۰۷	زبان خارجی
۱۰۸	زبان فنی
۱۰۹	زبان فنی و کاتالوگ خوانی
۱۱۰	زبان ماشین و اسمبلی
۱۱۱	ساختمان داده ها
۱۱۲	ساختمان‌های بتن آرمه
۱۱۳	ساختمان‌های فولادی
۱۱۴	سخت افزار کامپیوتر (۲)
۱۱۵	سرپرستی سازمان
۱۱۶	سیستم عامل شبکه
۱۱۷	سیستم‌های تلویزیون
۱۱۸	سیستم‌های کنترل خطی
۱۱۹	سیستم‌های مخابراتی
۱۲۰	سیستمهای کنترل خطی
۱۲۱	شبکه‌های کامپیوتری
۱۲۲	شبکه‌های محلی کامپیوتری
۱۲۳	شیوه ارائه نوشتاری و گفتاری
۱۲۴	طراحی اجزاء ماشین (۱)
۱۲۵	طراحی الگوریتم‌ها
۱۲۶	طراحی به کمک کامپیوتر (۲)
۱۲۷	طراحی خطوط انتقال انرژی و پروژه
۱۲۸	طراحی معماری (۱)
۱۲۹	طراحی معماری و شهرسازی
۱۳۰	طرح قالب‌های بتنی
۱۳۱	عایق‌ها و فشار قوی
۱۳۲	عناصر و جزئیات ساختمان
۱۳۳	عناصر و جزئیات ساختمانی (۱)
۱۳۴	فناوری‌های نوین ساختمان
۱۳۵	فیزیک الکتریسیته و مغناطیس
۱۳۶	فیزیک حرارت
۱۳۷	فیزیک مکانیک
۱۳۸	قانون کار و اقتصاد صنعتی
۱۳۹	قراردادها و مبانی حقوقی
۱۴۰	کارآفرینی
۱۴۱	کاربرد رایانه در اجرای ساختمان‌های بتنی
۱۴۲	کاربرد رایانه در برق
۱۴۳	کاربرد نرم افزارهای رایانه ای در معماری
۱۴۴	کاربردکامپیوتر در حسابداری(۲)
۱۴۵	کاربردکامپیوتر در حسابداری(۳)
۱۴۶	کاربردکامپیوتر در ساختمان
۱۴۷	کنترل صنعتی
۱۴۸	کنترل کننده های صنعتی
۱۴۹	گرافیک کامپیوتری (۱)
۱۵۰	ماشین های الکتریکی
۱۵۱	ماشین‌های الکتریکی ۳
۱۵۲	ماشین های الکتریکی سه فاز
۱۵۳	ماشین های الکتریکی مخصوص
۱۵۴	ماشین های مخصوص
۱۵۵	ماشینهای الکتریکی ۲
۱۵۶	ماشینهای سنکرون
۱۵۷	مباحث ویژه
۱۵۸	مباحث ویژه در الکترونیک
۱۵۹	مبانی دیجیتال
۱۶۰	مبانی سیستم‌های قدرت
۱۶۱	مبانی مدارهای منطقی
۱۶۲	مبانی مهندسی نرم افزار
۱۶۳	متره و برآورد
۱۶۴	محاسبات ساختمان های بتنی
۱۶۵	محاسبات ساختمان‌های فلزی
۱۶۶	محاسبات عددی
۱۶۷	محیط های چندرسانه ای
۱۶۸	مدارمجتمع خطی
۱۶۹	مدارهای الکترونیکی
۱۷۰	مدارهای الکتریکی ۲
۱۷۱	مدارهای مجتمع خطی
۱۷۲	مدارهای مخابراتی
۱۷۳	مدارهای منطقی
۱۷۴	مدیریت مالی
۱۷۵	مصالح ساختمانی و تکنولوژی بتن
۱۷۶	معادلات دیفرانسل
۱۷۷	معادلات دیفرانسیل
۱۷۸	مقاومت مصالح
۱۷۹	مقاومت مصالح (۱)
۱۸۰	مقاومت مصالح (۲)
۱۸۱	مقاومت مصالح و آزمایشگاه
۱۸۲	مقررات ساختمان‌های بتنی
۱۸۳	مقررات ملی ساختمان
۱۸۴	مکانیک خاک و پی سازی
۱۸۵	مکانیک خاک و پی سازی و آزمایشگاه
۱۸۶	مکانیک خاک و مهندسی پی
۱۸۷	مکانیک ساختمان
۱۸۸	مکانیک سیالات
۱۸۹	مکانیک سیالات و هیدرولیک
۱۹۰	مکانیک کاربردی
۱۹۱	منابع تغذیه
۱۹۲	مهندسی اینترنت
۱۹۳	مهندسی نرم افزار
۱۹۴	موتورهای القائی
۱۹۵	میکروکنترولر
۱۹۶	نحوه اجرای تاسیسات برقی ساختمان
۱۹۷	نحوه اجرای تاسیسات مکانیکی ساختمان
۱۹۸	نقشه برداری
۱۹۹	نقشه کشی تخصصی
۲۰۰	نقشه کشی ساختمان های بتنی
۲۰۱	نقشه کشی و نقشه خوانی سازه‌های بتنی
۲۰۲	نگهداری راه و بینه
۲۰۳	هیدرولوژی
۲۰۴	هیدرولیک ماشین آلات سنگین
۲۰۵	هیدرولیک و پنوماتیک
۲۰۶	هیدرولیک و نیوماتیک
۲۰۷	وسایل اندازه گیری و کنترل تبرید و تهویه

جدول ۳-۵: گروه‌بندی اولیه دروس در مثال (۳-۵-۱) با استفاده از الگوریتم (۳-۲)

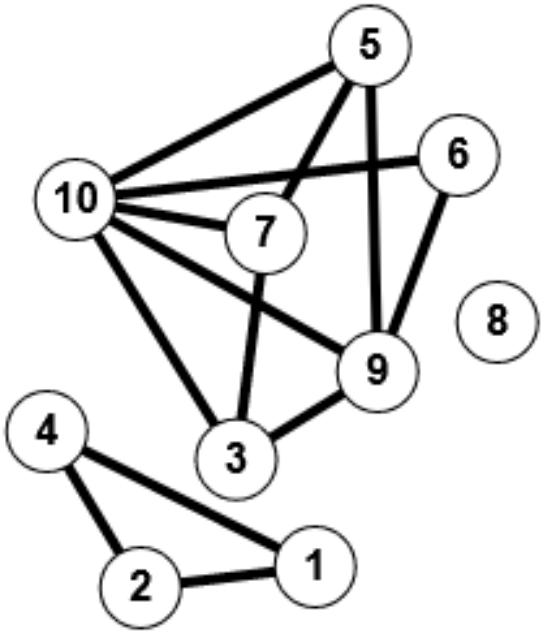
درس	گروه
اجزای ماشین	۱
انقلاب اسلامی ایران	۱
برنامه سازی پیشرفته (۲)	۱
برنامه نویسی کامپیوتر	۱
رسم فنی ونقشه کشی ساختمان	۱
رله و حفاظت	۱
روشهای آماری	۱
زبان تخصصی نرم افزار	۱
عایق ها و فشار قوی	۱
کارآفرینی	۱
مقررات ساختمانهای بتنی	۱
منابع تغذیه	۱
مهندسی نرم افزار	۱
نحوه اجرای تاسیسات مکانیکی ساختمان	۱
نقشه برداری	۱
نقشه کشی ساختمان های بتنی	۱
هیدرولیک ماشین آلات سنگین	۱
وسایل اندازه گیری و کنترل تبرید و تهویه	۱
تفسیر موضوعی قرآن	۲
تکنولوژی سوخت رسانی موتورهای بنزینی	۲
تکنولوژی سوخت و سوخت رسانی گازی و کارگاه	۲
دانش خانواده و جمعیت	۲
روستا (۱)	۲
ریاضی گسسته	۲
ریاضیات مهندسی	۲
مکانیک خاک و پی سازی و آزمایشگاه	۲
نحوه اجرای تاسیسات برقی ساختمان	۲
هیدرولیک و نیوماتیک	۲
الکترومغناطیس	۳
برنامه نویسی رایانه	۳
تحلیل رفتار قطعات بتنی	۳
تکنولوژی مولد قدرت	۳
زبان فنی	۳
زبان فنی وکاتالوگ خوانی	۳
طراحی به کمک کامپیوتر (۲)	۳
طراحی معماری و شهرسازی	۳
قانون کار و اقتصاد صنعتی	۳
گرافیک کامپیوتری (۱)	۳
محاسبات عددی	۳
مکانیک سیالات و هیدرولیک	۳
اجرای سازه های بتنی (۲)	۴
اندیشه اسلامی(۲) (نبوت و امامت)	۴
ایستایی (۲)	۴
تکنولوژی مالتی پلکس	۴
تمرین های معماری (۱)	۴
حفاظت ماشینهای الکتریکی	۴
روشهای تعمیر و نگهداری ساختمان	۴
زبان فارسی	۴
شبکه های محلی کامپیوتری	۴
کاربردکامپیوتر در حسابداری(۲)	۴
ماشین های الکتریکی	۴
ماشین های الکتریکی سه فاز	۴
مباحث ویژه در الکترونیک	۴
مدارهای منطقی	۴
مهندسی اینترنت	۴
اندیشه اسلامی(۱) (مبدأ و معاد)	۵
تاریخ تحلیلی صدر اسلام	۵
تاسیسات الکتریکی و پروژه	۵
تکنولوژی سیستم های هدایت و کنترل خودرو	۵
درک و بیان معماری (۲)	۵
راهسازی و روسازی	۵
سیستم های تلویزیون	۵
طراحی الگوریتمها	۵
کاربردکامپیوتر در ساختمان	۵
مقاومت مصالح و آزمایشگاه	۵
مکانیک خاک و مهندسی پی	۵
مکانیک کاربردی	۵
اصول سرپرستی	۶
ایستایی	۶
برنامه سازی سیستم	۶
تکنولوژی وکارگاه قالب سازی	۶
دستگاههای تهویه مطبوع	۶
زبان تخصصی	۶
ساختمان داده ها	۶
سرپرستی سازمان	۶
سیستم عامل شبکه	۶
قراردادها و مبانی حقوقی	۶
متره و برآورد	۶
مقاومت مصالح (۲)	۶
مکانیک ساختمان	۶
موتورهای القائی	۶
میکروکنترولر	۶
اجزاء ساختمان	۷
اخلاق مهندسی	۷
اقتصاد(۱) (خرد)	۷
الکترونیک صنعتی	۷
برنامه نویسی ماشین های کنترل عددی	۷
ترمودینامیک	۷
دستگاههای برودتی	۷
ریاضی ۲	۷
زبان خارجی	۷
طراحی اجزاء ماشین (۱)	۷
طرح قالب های بتنی	۷
کاربرد نرم افزارهای رایانه ای در معماری	۷
ماشین های الکتریکی ۳	۷
محیط های چندرسانه ای	۷
مدارهای الکترونیکی	۷
اخلاق اسلامی (مبانی و مفاهیم)	۸
استاتیک	۸
اقتصاد(۲) (کلان)	۸
ایستایی (۱)	۸
آمار و احتمالات مهندسی	۸
تحلیل مقدماتی سازه ها	۸
ذخیره و بازیابی اطلاعات	۸
سیستم های مخابراتی	۸
سیستمهای کنترل خطی	۸
ماشین های مخصوص	۸
مبانی سیستم های قدرت	۸
مبانی مدارهای منطقی	۸
مدارهای مخابراتی	۸
معادلات دیفرانسیل	۸
مقاومت مصالح (۱)	۸
هیدرولوژی	۸
اجرای سازه های آبی	۹
تاسیسات تهویه مطبوع	۹
تحلیل مدارهای الکترونیکی	۹
ترمودینامیک (۲)	۹
تکنولوژی عایق ها و فشارقوی	۹
تمرین های معماری (۲)	۹
تولید و نیروگاه	۹
خطوط انتقال مخابراتی	۹
ریاضی عمومی	۹
ریاضی عمومی (۲)	۹
عناصر و جزئیات ساختمان	۹
عناصر و جزئیات ساختمانی (۱)	۹
ماشین های الکتریکی مخصوص	۹
ماشینهای الکتریکی ۲	۹
ماشینهای سنکرون	۹
مقررات ملی ساختمان	۹
مکانیک خاک و پی سازی	۹
آشنایی با فرهنگ و ارزش های دفاع مقدس	۱۰
برق تاسیسات	۱۰
تجزیه و تحلیل سیگنالها و سیستمها	۱۰
تکنیک پالس	۱۰
دینامیک	۱۰
ساختمانهای فولادی	۱۰
سیستم های کنترل خطی	۱۰
فیزیک الکتریسیته و مغناطیس	۱۰
فیزیک مکانیک	۱۰
کنترل کننده های صنعتی	۱۰
مدیریت مالی	۱۰
مصالح ساختمانی و تکنولوژی بتن	۱۰
معادلات دیفرانسل	۱۰
اصول مدیریت ساخت	۱۱
الکترونیک خودرو	۱۱
پرسپکتیو	۱۱
تکنولوژی سیستم های الکتریکی و الکترونیکی خودرو	۱۱
تکنولوژی و بازرسی جوش و کارگاه	۱۱
حسابرسی (۱)	۱۱
ریاضی عمومی (۱)	۱۱
ریاضی کاربردی	۱۱
طراحی خطوط انتقال انرژی و پروژه	۱۱
محاسبات ساختمان های فلزی	۱۱
مدارمجتمع خطی	۱۱
مدارهای الکتریکی ۲	۱۱
نقشه کشی و نقشه خوانی سازه های بتنی	۱۱
آشنایی با زلزله و اثرآن برسازه ها	۱۲
بتن پیش ساخته و پیش تنیده	۱۲
بررسی سیستم های قدرت ۱	۱۲
بررسی سیستم های قدرت۲	۱۲
تحلیل مدارهای الکتریکی	۱۲
تنظیم شرایط محیطی (۱)	۱۲
حسابداری شرکتهای (۲)	۱۲
زبان ماشین و اسمبلی	۱۲
فناوریهای نوین ساختمان	۱۲
مبانی مهندسی نرم افزار	۱۲
مدارهای مجتمع خطی	۱۲
مقاومت مصالح	۱۲
نقشه کشی تخصصی	۱۲
اجرای سازه های بتنی	۱۳
اجزاء ماشین (۳)	۱۳
الکترونیک عمومی	۱۳
تجهیزات پست و نیروگاه	۱۳
حسابداری مالیاتی	۱۳
شبکه های کامپیوتری	۱۳
شیوه ارائه نوشتاری و گفتاری	۱۳
طراحی معماری (۱)	۱۳
فیزیک حرارت	۱۳
کاربرد رایانه در اجرای ساختمانهای بتنی	۱۳
آشنایی با معماری اسلامی (۱)	۱۴
تعمیرو نگهداری ساختمان	۱۴
تکنولوژی انتقال قدرت اتوماتیک	۱۴
تکنولوژی انتقال قدرت معمولی	۱۴
حسابداری صنعتی (۲)	۱۴
ساختمانهای بتن آرمه	۱۴
کنترل صنعتی	۱۴
مباحث ویژه	۱۴
ایمنی در برق	۱۵
حسابداری دولتی(۱)	۱۵
حسابداری مالی	۱۵
روشنایی فنی	۱۵
سخت افزار کامپیوتر (۲)	۱۵
محاسبات ساختمان های بتنی	۱۵
مکانیک سیالات	۱۵
نگهداری راه و بینه	۱۵
ایمنی کارگاه	۱۶
پایگاه داده ها	۱۶
تکنولوژی و کارگاه قالب بندی و آرماتور	۱۶
کاربرد رایانه در برق	۱۶
کاربردکامپیوتر در حسابداری(۳)	۱۶
هیدرولیک و پنوماتیک	۱۶
آئین نامه های ساختمانی	۱۷
برنامه نویسی مبتنی بروب	۱۷
روشهای مرمت ابنیه	۱۷
مبانی دیجیتال	۱۷
آمار و احتمالات	۱۸
روسازی های بتنی و آسفالتی	۱۸

جدول ۳-۶: برنامه نهایی امتحانات در مثال (۳-۵-۱) با استفاده از الگوریتم (۳-۳)

درس	نوبت و روز	درس	نوبت و روز	درس	نوبت و روز	درس	نوبت و روز	درس	نوبت و روز
اجرای سازه های آبی	نوبت اول روز ۱	طراحی اجزاء ماشین (۱)	نوبت اول روز ۳	تکنولوژی مالتی پلکس	نوبت اول روز ۶	تکنولوژی و کارگاه قالب بندی و آرماتور	نوبت دوم روز ۸	نقشه کشی تخصصی	نوبت اول روز ۱۲
تاسیسات تهویه مطبوع	نوبت اول روز ۱	طرح قالب های بتنی	نوبت اول روز ۳	تمرین های معماری (۱)	نوبت اول روز ۶	کاربرد رایانه در برق	نوبت دوم روز ۸	اصول مدیریت ساخت	نوبت اول روز ۱۳
تحلیل مدارهای الکترونیکی	نوبت اول روز ۱	کاربرد نرم افزارهای رایانه ای در معماری	نوبت اول روز ۳	حفاظت ماشینهای الکتریکی	نوبت اول روز ۶	کاربردکامپیوتر در حسابداری(۳)	نوبت دوم روز ۸	الکترونیک خودرو	نوبت اول روز ۱۳
ترمودینامیک (۲)	نوبت اول روز ۱	ماشین های الکتریکی ۳	نوبت اول روز ۳	روشهای تعمیر و نگهداری ساختمان	نوبت اول روز ۶	هیدرولیک و پنوماتیک	نوبت دوم روز ۸	پرسپکتیو	نوبت اول روز ۱۳
تکنولوژی عایق ها و فشارقوی	نوبت اول روز ۱	محیط های چندرسانه ای	نوبت اول روز ۳	زبان فارسی	نوبت اول روز ۶	ایمنی در برق	نوبت اول روز ۹	تکنولوژی سیستم های الکتریکی و الکترونیکی خودرو	نوبت اول روز ۱۳
تمرین های معماری (۲)	نوبت اول روز ۱	مدارهای الکترونیکی	نوبت دوم روز ۳	شبکه های محلی کامپیوتری	نوبت اول روز ۶	حسابداری دولتی(۱)	نوبت اول روز ۹	تکنولوژی و بازرسی جوش و کارگاه	نوبت اول روز ۱۳
تولید و نیروگاه	نوبت اول روز ۱	اصول سرپرستی	نوبت اول روز ۴	کاربردکامپیوتر در حسابداری(۲)	نوبت اول روز ۶	حسابداری مالی	نوبت اول روز ۹	حسابرسی (۱)	نوبت اول روز ۱۳
خطوط انتقال مخابراتی	نوبت اول روز ۱	ایستایی	نوبت اول روز ۴	ماشین های الکتریکی	نوبت اول روز ۶	روشنایی فنی	نوبت اول روز ۹	ریاضی عمومی (۱)	نوبت اول روز ۱۳
ریاضی عمومی	نوبت اول روز ۱	برنامه سازی سیستم	نوبت اول روز ۴	ماشین های الکتریکی سه فاز	نوبت اول روز ۶	سخت افزار کامپیوتر (۲)	نوبت اول روز ۹	ریاضی کاربردی	نوبت اول روز ۱۳
ریاضی عمومی (۲)	نوبت اول روز ۱	تکنولوژی وکارگاه قالب سازی	نوبت اول روز ۴	مباحث ویژه در الکترونیک	نوبت اول روز ۶	محاسبات ساختمان های بتنی	نوبت اول روز ۹	طراحی خطوط انتقال انرژی و پروژه	نوبت اول روز ۱۳
عناصر و جزئیات ساختمان	نوبت اول روز ۱	دستگاههای تهویه مطبوع	نوبت اول روز ۴	مدارهای منطقی	نوبت اول روز ۶	مکانیک سیالات	نوبت اول روز ۹	محاسبات ساختمان های فلزی	نوبت اول روز ۱۳
عناصر و جزئیات ساختمانی (۱)	نوبت اول روز ۱	زبان تخصصی	نوبت اول روز ۴	مهندسی اینترنت	نوبت اول روز ۶	نگهداری راه و بینه	نوبت اول روز ۹	مدارمجتمع خطی	نوبت اول روز ۱۳
ماشین های الکتریکی مخصوص	نوبت اول روز ۱	ساختمان داده ها	نوبت اول روز ۴	آمار و احتمالات	نوبت دوم روز ۶	آشنایی با معماری اسلامی (۱)	نوبت اول روز ۱۰	مدارهای الکتریکی ۲	نوبت اول روز ۱۳
ماشینهای الکتریکی ۲	نوبت اول روز ۱	سرپرستی سازمان	نوبت اول روز ۴	روسازی های بتنی و آسفالتی	نوبت دوم روز ۶	تعمیرو نگهداری ساختمان	نوبت اول روز ۱۰	نقشه کشی و نقشه خوانی سازه های بتنی	نوبت اول روز ۱۳
ماشینهای سنکرون	نوبت اول روز ۱	سیستم عامل شبکه	نوبت اول روز ۴	الکترومغناطیس	نوبت اول روز ۷	تکنولوژی انتقال قدرت اتوماتیک	نوبت اول روز ۱۰	آشنایی با فرهنگ و ارزش های دفاع مقدس	نوبت اول روز ۱۴
مقررات ملی ساختمان	نوبت اول روز ۱	قراردادها و مبانی حقوقی	نوبت اول روز ۴	برنامه نویسی رایانه	نوبت اول روز ۷	تکنولوژی انتقال قدرت معمولی	نوبت اول روز ۱۰	برق تاسیسات	نوبت اول روز ۱۴
مکانیک خاک و پی سازی	نوبت دوم روز ۱	متره و برآورد	نوبت اول روز ۴	تحلیل رفتار قطعات بتنی	نوبت اول روز ۷	حسابداری صنعتی (۲)	نوبت اول روز ۱۰	تجزیه و تحلیل سیگنالها و سیستمها	نوبت اول روز ۱۴
اخلاق اسلامی (مبانی و مفاهیم)	نوبت اول روز ۲	مقاومت مصالح (۲)	نوبت اول روز ۴	تکنولوژی مولد قدرت	نوبت اول روز ۷	ساختمانهای بتن آرمه	نوبت اول روز ۱۰	تکنیک پالس	نوبت اول روز ۱۴
استاتیک	نوبت اول روز ۲	مکانیک ساختمان	نوبت اول روز ۴	زبان فنی	نوبت اول روز ۷	کنترل صنعتی	نوبت اول روز ۱۰	دینامیک	نوبت اول روز ۱۴
اقتصاد(۲) (کلان)	نوبت اول روز ۲	موتورهای القائی	نوبت اول روز ۴	زبان فنی وکاتالوگ خوانی	نوبت اول روز ۷	مباحث ویژه	نوبت اول روز ۱۰	ساختمانهای فولادی	نوبت اول روز ۱۴
ایستایی (۱)	نوبت اول روز ۲	میکروکنترولر	نوبت دوم روز ۴	طراحی به کمک کامپیوتر (۲)	نوبت اول روز ۷	اجرای سازه های بتنی	نوبت اول روز ۱۱	سیستم های کنترل خطی	نوبت اول روز ۱۴
آمار و احتمالات مهندسی	نوبت اول روز ۲	اجرای ماشین	نوبت اول روز ۵	طراحی معماری و شهرسازی	نوبت اول روز ۷	اجزاء ماشین (۳)	نوبت اول روز ۱۱	فیزیک الکتریسیته و مغناطیس	نوبت اول روز ۱۴
تحلیل مقدماتی سازه ها	نوبت اول روز ۲	انقلاب اسلامی ایران	نوبت اول روز ۵	قانون کار و اقتصاد صنعتی	نوبت اول روز ۷	الکترونیک عمومی	نوبت اول روز ۱۱	فیزیک مکانیک	نوبت اول روز ۱۴
ذخیره و بازیابی اطلاعات	نوبت اول روز ۲	برنامه سازی پیشرفته (۲)	نوبت اول روز ۵	گرافیک کامپیوتری (۱)	نوبت اول روز ۷	تجهیزات پست و نیروگاه	نوبت اول روز ۱۱	کنترل کننده های صنعتی	نوبت اول روز ۱۴
سیستم های مخابراتی	نوبت اول روز ۲	برنامه نویسی کامپیوتر	نوبت اول روز ۵	محاسبات عددی	نوبت اول روز ۷	حسابداری مالیاتی	نوبت اول روز ۱۱	مدیریت مالی	نوبت اول روز ۱۴
سیستمهای کنترل خطی	نوبت اول روز ۲	رسم فنی ونقشه کشی ساختمان	نوبت اول روز ۵	مکانیک سیالات و هیدرولیک	نوبت اول روز ۷	شبکه های کامپیوتری	نوبت اول روز ۱۱	مصالح ساختمانی و تکنولوژی بتن	نوبت اول روز ۱۴
ماشین های مخصوص	نوبت اول روز ۲	رله و حفاظت	نوبت اول روز ۵	آئین نامه های ساختمانی	نوبت دوم روز ۷	شیوه ارائه نوشتاری و گفتاری	نوبت اول روز ۱۱	معادلات دیفرانسل	نوبت اول روز ۱۴
مبانی سیستم های قدرت	نوبت اول روز ۲	روشهای آماری	نوبت اول روز ۵	برنامه نویسی مبتنی بروب	نوبت دوم روز ۷	طراحی معماری (۱)	نوبت اول روز ۱۱	اندیشه اسلامی(۱) (مبدأ و معاد)	نوبت اول روز ۱۵
مبانی مدارهای منطقی	نوبت اول روز ۲	زبان تخصصی نرم افزار	نوبت اول روز ۵	روشهای مرمت ابنیه	نوبت دوم روز ۷	فیزیک حرارت	نوبت اول روز ۱۱	تاریخ تحلیلی صدر اسلام	نوبت اول روز ۱۵
مدارهای مخابراتی	نوبت اول روز ۲	عایق ها و فشار قوی	نوبت اول روز ۵	مبانی دیجیتال	نوبت دوم روز ۷	کاربرد رایانه در اجرای ساختمانهای بتنی	نوبت اول روز ۱۱	تاسیسات الکتریکی و پروژه	نوبت اول روز ۱۵
معادلات دیفرانسیل	نوبت اول روز ۲	کارآفرینی	نوبت اول روز ۵	تفسیر موضوعی قرآن	نوبت اول روز ۸	آشنایی با زلزله و اثرآن برسازه ها	نوبت اول روز ۱۲	تکنولوژی سیستم های هدایت و کنترل خودرو	نوبت اول روز ۱۵
مقاومت مصالح (۱)	نوبت دوم روز ۲	مقررات ساختمانهای بتنی	نوبت اول روز ۵	تکنولوژی سوخت رسانی موتورهای بنزینی	نوبت اول روز ۸	بتن پیش ساخته و پیش تنیده	نوبت اول روز ۱۲	درک و بیان معماری (۲)	نوبت اول روز ۱۵
هیدرولوژی	نوبت دوم روز ۲	منابع تغذیه	نوبت اول روز ۵	تکنولوژی سوخت و سوخت رسانی گازی و کارگاه	نوبت اول روز ۸	بررسی سیستم های قدرت ۱	نوبت اول روز ۱۲	راهسازی و روسازی	نوبت اول روز ۱۵
اجزاء ساختمان	نوبت اول روز ۳	مهندسی نرم افزار	نوبت اول روز ۵	دانش خانواده و جمعیت	نوبت اول روز ۸	بررسی سیستم های قدرت۲	نوبت اول روز ۱۲	سیستم های تلویزیون	نوبت اول روز ۱۵
اخلاق مهندسی	نوبت اول روز ۳	نحوه اجرای تاسیسات مکانیکی ساختمان	نوبت اول روز ۵	روستا (۱)	نوبت اول روز ۸	تحلیل مدارهای الکتریکی	نوبت اول روز ۱۲	طراحی الگوریتمها	نوبت اول روز ۱۵
اقتصاد(۱) (خرد)	نوبت اول روز ۳	نقشه برداری	نوبت اول روز ۵	ریاضی گسسته	نوبت اول روز ۸	تنظیم شرایط محیطی (۱)	نوبت اول روز ۱۲	کاربردکامپیوتر در ساختمان	نوبت اول روز ۱۵
الکترونیک صنعتی	نوبت اول روز ۳	نقشه کشی ساختمان های بتنی	نوبت دوم روز ۵	ریاضیات مهندسی	نوبت اول روز ۸	حسابداری شرکتهای (۲)	نوبت اول روز ۱۲	مقاومت مصالح و آزمایشگاه	نوبت اول روز ۱۵
برنامه نویسی ماشین های کنترل عددی	نوبت اول روز ۳	هیدرولیک ماشین آلات سنگین	نوبت دوم روز ۵	مکانیک خاک و پی سازی و آزمایشگاه	نوبت اول روز ۸	زبان ماشین و اسمبلی	نوبت اول روز ۱۲	مکانیک خاک و مهندسی پی	نوبت اول روز ۱۵
ترمودینامیک	نوبت اول روز ۳	وسایل اندازه گیری و کنترل تبرید و تهویه	نوبت دوم روز ۵	نحوه اجرای تاسیسات برقی ساختمان	نوبت اول روز ۸	فناوریهای نوین ساختمان	نوبت اول روز ۱۲	مکانیک کاربردی	نوبت اول روز ۱۵
دستگاههای برودتی	نوبت اول روز ۳	اجرای سازه های بتنی (۲)	نوبت اول روز ۶	هیدرولیک و نیوماتیک	نوبت اول روز ۸	مبانی مهندسی نرم افزار	نوبت اول روز ۱۲		
ریاضی ۲	نوبت اول روز ۳	اندیشه اسلامی(۲) (نبوت و امامت)	نوبت اول روز ۶	ایمنی کارگاه	نوبت دوم روز ۸	مدارهای مجتمع خطی	نوبت اول روز ۱۲		
زبان خارجی	نوبت اول روز ۳	ایستایی (۲)	نوبت اول روز ۶	پایگاه داده ها	نوبت دوم روز ۸	مقاومت مصالح	نوبت اول روز ۱۲		

جدول ۳-۷: بخشی از تداخل برنامه امتحانات در دو نوبت امتحانی یک روز در مثال (۳-۵-۱)

کد دانشجو	روز ششم نوبت اول	روز ششم نوبت دوم	روز هفتم نوبت اول	روز هفتم نوبت دوم	روز هشتم نوبت اول	روز هشتم نوبت دوم
۵۲۲	روش‌های تعمیر و نگهداری ساختمان	روسازی‌های بتنی و آسفالتی				
۵۷۳	شبکه‌های محلی کامپیوتری	آمار و احتمالات				
۷۶۱	زبان فارسی	آمار و احتمالات				
۷۶۶	شبکه‌های محلی کامپیوتری	آمار و احتمالات				
۷۷۱	شبکه‌های محلی کامپیوتری	آمار و احتمالات				
۷۷۳	شبکه‌های محلی کامپیوتری	آمار و احتمالات				
۳۲			زبان فنی	مبانی دیجیتال		
۱۴۱			زبان فنی	مبانی دیجیتال		
۲۹۰			زبان فنی	برنامه‌نویسی مبتنی بر وب		
۲۹۳			زبان فنی	برنامه‌نویسی مبتنی بر وب		
۳۵۴			زبان فنی	آئین نامه‌های ساختمانی		
۴۲۰			زبان فنی	مبانی دیجیتال		
۵۳۴			مکانیک سیالات و هیدرولیک	روش‌های مرمت ابنیه		
۱۵۷					دانش خانواده و جمعیت	پایگاه داده‌ها
۱۶۳					دانش خانواده و جمعیت	کاربردکامپیوتر در حسابداری(۳)
۳۴۰					دانش خانواده و جمعیت	هیدرولیک و پنوماتیک
۳۴۸					دانش خانواده و جمعیت	تکنولوژی و کارگاه قالب بندی و آرماتور
۳۶۰					دانش خانواده و جمعیت	تکنولوژی و کارگاه قالب بندی و آرماتور
۳۶۳					دانش خانواده و جمعیت	تکنولوژی و کارگاه قالب بندی و آرماتور
۳۶۷					دانش خانواده و جمعیت	تکنولوژی و کارگاه قالب بندی و آرماتور
۴۲۴					دانش خانواده و جمعیت	کاربرد رایانه در برق
۴۸۰					تفسیر موضوعی قرآن	ایمنی کارگاه
۴۸۱					تفسیر موضوعی قرآن	ایمنی کارگاه
۵۱۷					تفسیر موضوعی قرآن	ایمنی کارگاه
۵۱۸					تفسیر موضوعی قرآن	ایمنی کارگاه



شکل ۳-۱: نمونه‌ای از گراف ماتریس مجاورت شامل ۱۰ درس در مثال ۳-۵-۱

۳-۶ پژوهش‌های آتی

در این بخش، روش قوانین انجمنی [۱] برای حل مسئله زمان‌بندی امتحانات پیشنهاد می‌شود. قوانین انجمنی از جمله روش‌های حوزه داده‌کاوی است که روابط و وابستگی‌های متقابل بین مجموعه بزرگی از اقلام داده‌ای را نشان می‌دهند. هدف از گروه‌بندی اولیه دروس این است که دانشجویان در یک گروه دو یا چند درس مشترک نداشته باشند و هیچ دانشجویی دارای دو یا چند امتحان در یک بازه زمانی نباشد و نقض محدودیت نرم و سخت را نداشته باشیم. بنابراین با توجه به این که روش قوانین انجمنی برای تحلیل سبد خرید مشتریان(مثال ۳-۶-۱) کاربرد دارد، می‌توان به فرآیند انتخاب واحد دانشجویان به این دید نگاه کرد که دانشجویان(مشتریان) و دروس(کالاها) هستند. بنابراین می‌توان با تشخیص الگوهای پرتکرار موجود در انتخاب واحدها، مجموعه دروسی که با هم بیشتر انتخاب واحد شده‌اند در گروه‌های متفاوت قرار بگیرند تا از ایجاد تداخل سخت جلوگیری نماییم. فلوچارت این روش(تصویر ۳-۲) را مشاهده می‌کنید. این روش برای گروه‌بندی اولیه دروس پیشنهاد می‌شود.

[۱] Association Rule Learning

۳-۶-۱ آشنایی با قوانین انجمنی

قوانین انجمنی وابستگی‌ها و روابط متقابل بین مجموعه‌های بزرگ از اقلام داده‌ای را نشان می‌دهد. یافتن این قوانینی در حوزه‌های مختلف مورد توجه است. به عنوان مثال در حوزه پزشکی و تشخیص تقلب در تراکنش‌های مالی اشاره نمود. الگوریتم اپریوری [۱] می‌تواند تعداد اقلامی که نیاز به بررسی آن‌ها است را کاهش دهد.

مثال ۳-۶-۱. یک مثال پرکاربرد کشف قوانین انجمنی، تحلیل سبد خرید است. در این فرایند با توجه به اقلام خرید مشتریان در سبد خرید دارند، رفتار و عادات و خرید مشتریان مورد تحلیل قرار می‌گیرد. برای مثال احتمال این که مشتریانی که برای خرید خامه به فروشگاه آمده‌اند عسل نیز خریداری کنند مشخص است. این روش، اقلام فاقد تکرار مجموعه را بدست می‌آورد. در این حالت اگر { نوشابه } فاقد تکرار است، {نوشابه، ساندویچ} هم، نادر باشند. بنابراین، برای یکی کردن لیست مجموعه اقلام محبوب، نیازی به در نظر گرفتن نوشابه، ساندویچ و یا هیچ یک از دیگر مجموعه اقلام حاوی نوشابه، نخواهد بود.

با استفاده از اصل اپریوری، هرس کردن تعداد مجموعه اقلام که باید بررسی شوند امکان پذیر است. یکی از محدودیت‌های روش قوانین، انجمنی هزینه محاسباتی حتی اگر الگوریتم اپریوری تعداد اقلام کاندید برای بررسی را کاهش دهد بالا است. یک راهکار جایگزین، کاهش تعداد مقایسه‌ها با استفاده از ساختارهای پیشرفته داده مانند جدول‌های هش برای مرتب‌سازی اقلام کاندید به شیوه موثرتر است. فلوچارت مربوط به این روش(نمودار ۳-۲) را مشاهده می‌کنید.

۳-۷ مثال عملی برنامه

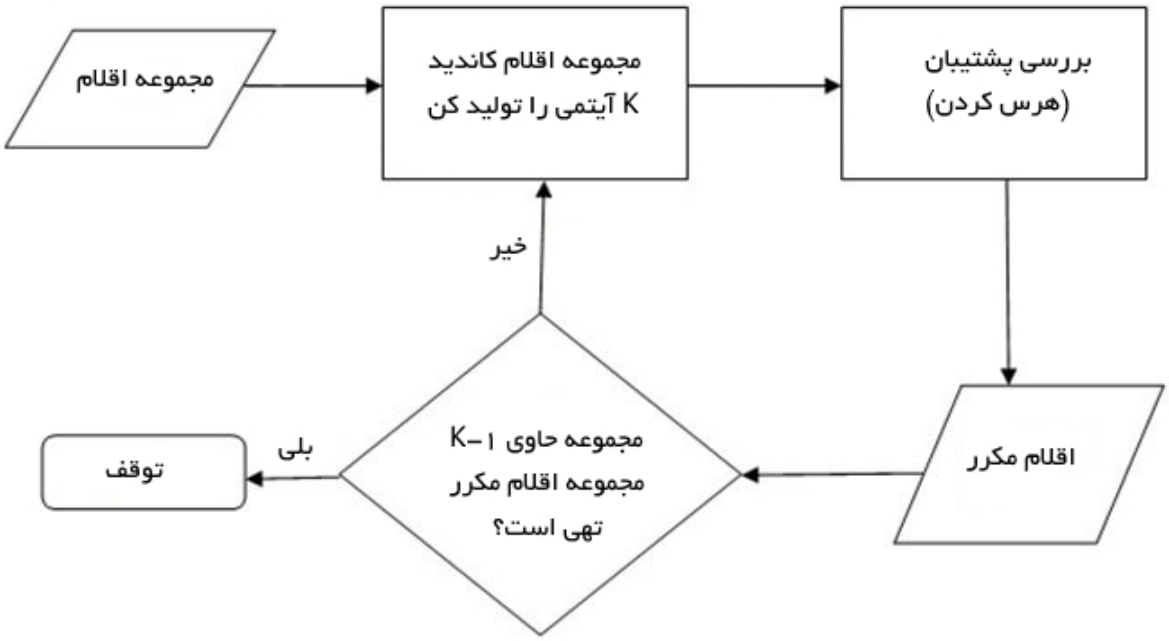
در این بخش به معرفی یک مثال عملی برای روش زمان‌بندی امتحانات ارائه شده در این فصل می‌پردازیم.

مثال ۳-۷-۱. با فرض این که لیست انتخاب واحدهای ۱۲ دانشجو از بین ۱۰ درس را داشته باشیم، مسئله زمان‌بندی امتحانات را با استفاده از روش ابتکاری و حریصانه حل می‌نماییم. ابتدا گراف انتخاب واحدها را تشکیل می‌دهیم که در شکل ۳-۳ ، گراف انتخاب واحدها را مشاهده می‌نمایید. مجموعه C، دروس و مجموعه S، دانشجویان هستند. یال‌های جهت دار از دانشجویان به دروس، نشان دهنده این است دانشجویی درس مربوطه را انتخاب واحد نموده‌است.

در مرحله بعد، ماتریس مجاورت دروس را تشکیل می‌دهیم. ماتریس مجاورت دروس را مشاهده می‌نمایید.

---

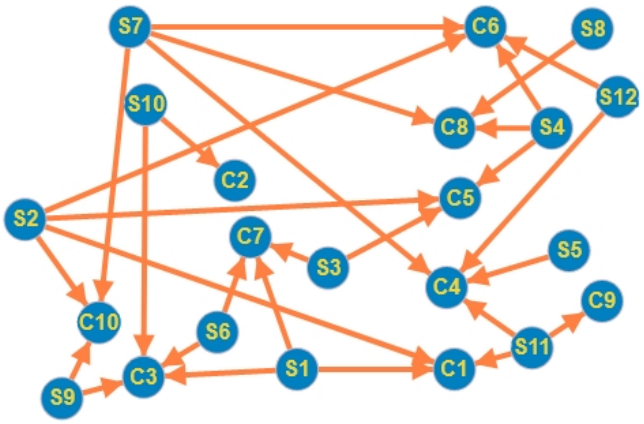
[۱] Apriori Association Rule Mining



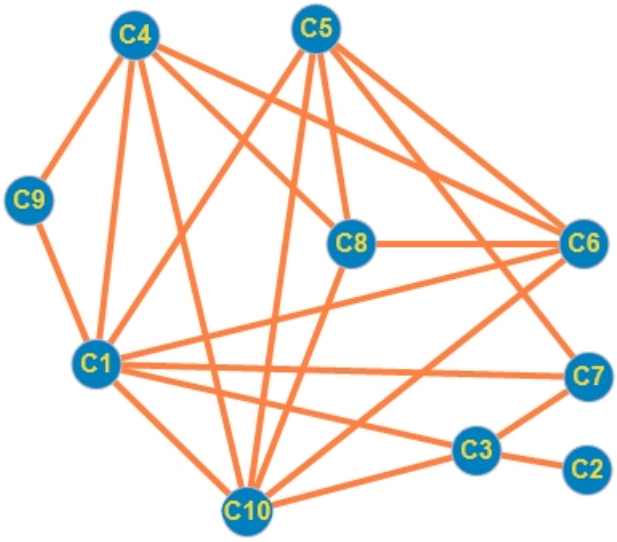
شکل ۳-۲: فلوچارت روش قوانین انجمنی (روش Apriori Association Rule Mining)

	$C۱$	$C۲$	$C۳$	$C۴$	$C۵$	$C۶$	$C۷$	$C۸$	$C۹$	$C۱۰$
$C۱$	۰	۰	۱	۱	۱	۱	۱	۰	۱	۱
$C۲$	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰
$C۳$	۱	۱	۰	۰	۰	۰	۱	۰	۰	۱
$C۴$	۱	۰	۰	۰	۰	۱	۰	۱	۱	۱
$C۵$	۱	۰	۰	۰	۰	۱	۱	۱	۰	۱
$C۶$	۱	۰	۰	۱	۱	۰	۰	۱	۰	۱
$C۷$	۱	۰	۱	۰	۱	۰	۰	۰	۰	۰
$C۸$	۰	۰	۰	۱	۱	۱	۰	۰	۰	۱
$C۹$	۱	۰	۰	۱	۰	۰	۰	۰	۰	۰
$C۱۰$	۱	۰	۱	۱	۱	۱	۰	۱	۰	۰

گراف ماتریس مجاورت را در شکل ۳-۴ مشاهده می‌نمایید. مجموعه C، دروس هستند. راس‌های همجوار، نشان‌دهنده این است این دو درس به طور هم زمان توسط دانشجو(یان) انتخاب واحد شده‌اند.



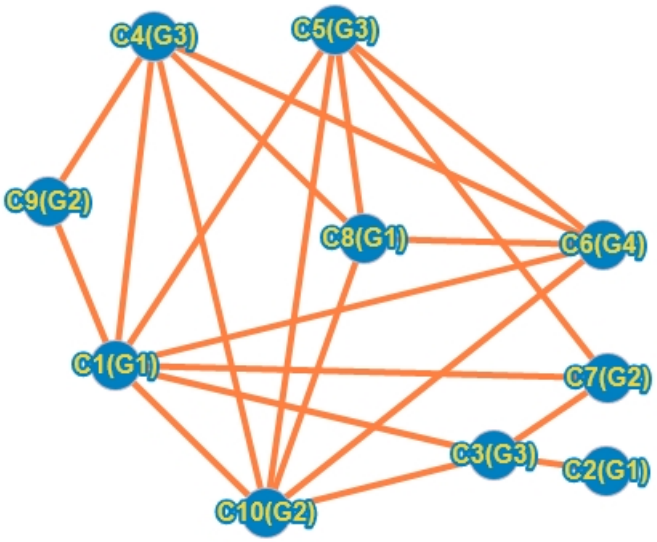
شکل ۳-۳: گراف انتخاب واحدها در مثال ۳-۷-۱ . مجموعه C، دروس و مجموعه S، دانشجویان هستند. یال‌های جهت دار از دانشجویان به دروس، نشان‌دهنده این است دانشجو درس مربوطه را انتخاب واحد نموده‌است



شکل ۳-۴: گراف ماتریس مجاورت در مثال ۳-۷-۱. راس‌های همجوار، نشان‌دهنده این است این دو راس به طور هم زمان توسط دانشجو انتخاب واحد شده است.

در این مرحله، با استفاده از تابع تولید گروه‌بندی اولیه دروس گروه‌بندی اولیه دروس انجام می‌شود. نتیجه گروه‌بندی اولیه دروس، تقسیم بندی دروس به ۴ گروه است که در شکل ۳-۵ مشاهده می‌نمایید.

نتایج گروه‌بندی اولیه دروس را در جدول ۳-۸ مشاهده می‌کنید. میزان تداخل گروه‌ها از گروه‌بندی اولیه دروس(جدول ۳-۸) در مثال ۳-۷-۱ در جدول ۳-۹ در دو گروه دانشجویان دو امتحانی و دانشجویان سه امتحانی آمده است.



شکل ۳-۵: گراف گروه‌بندی اولیه دروس در مثال ۳-۷-۱

جدول ۳-۸: گروه‌بندی اولیه دروس در مثال ۳-۷-۱ . مجموعه C، دروس و مجموعه S، دانشجویان هستند.

گروه	دروس(لیست دانشجویان)
G۱	$\{\overbrace{S۱, S۲, S۱۱}^{C۱}, \overbrace{S۱۰}^{C۲}, \overbrace{S۴, S۷, S۸}^{C۸}\}$
G۲	$\{\overbrace{S۱, S۳, S۶}^{C۷}, \overbrace{S۱۱}^{C۹}, \overbrace{S۲, S۷, S۹}^{C۱۰}\}$
G۳	$\{\overbrace{S۱, S۶, S۹, S۱۰}^{C۳}, \overbrace{S۵, S۷, S۱۱, S۱۲}^{C۴}, \overbrace{S۲, S۳, S۴}^{C۵}\}$
G۴	$\{\overbrace{S۲, S۴, S۷, S۱۲}^{C۶}\}$

جدول ۳-۹: میزان تداخل گروه‌ها از گروه‌بندی اولیه دروس در مثال ۳-۷-۱

گروه‌ها	دانشجویان دو امتحانی	دانشجویان سه امتحانی
$\{G۱, G۲\}$	$\{S۱, S۲, S۷, S۱۱\}$	-
$\{G۱, G۳\}$	$\{S۱, S۲, S۴, S۷, S۱۰, S۱۱\}$	-
$\{G۱, G۴\}$	$\{S۲, S۴, S۷\}$	-
$\{G۲, G۳\}$	$\{S۱, S۲, S۳, S۶, S۷, S۹, S۱۱\}$	-
$\{G۲, G۴\}$	$\{S۲, S۷\}$	-
$\{G۳, G۴\}$	$\{S۲, S۴, S۷, S۱۲\}$	-
$\{G۱, G۲, G۳\}$	$\{S۳, S۴, S۶, S۹, S۱۰\}$	$\{S۱, S۲, S۷, S۱۱\}$
$\{G۱, G۲, G۴\}$	$\{S۱, S۴, S۱۱\}$	$\{S۲, S۷\}$
$\{G۱, G۳, G۴\}$	$\{S۱, S۱۰, S۱۱, S۱۲\}$	$\{S۲, S۴, S۷\}$
$\{G۲, G۳, G۴\}$	$\{S۱, S۳, S۴, S۶, S۹, S۱۱, S۱۲\}$	$\{S۲, S۷\}$

تمامی انتخاب‌ها در حالت‌های زیر در مثال ۳-۷-۱، با توجه به جدول میزان تداخل گروه‌ها از گروه‌بندی اولیه دروس(جدول ۳-۹) با هدف نداشتن و یا حداقل نمودن نقض محدودیت سخت و نرم و با در نظر گرفتن دو فرض این که در هر روز سه نوبت برگزاری امتحان(بازه زمانی) داریم و ظرفیت پذیرش نوبت امتحانی برابر با تعداد کل دانشجویان در نظر می‌گیریم.

در حالت۱، اگر تعداد روزهای امتحانی بزرگتر یا مساوی تعداد گروه‌ها(عدد ۴) در نظر بگیریم، هیچ گونه نقض محدودیت نداریم. یک نمونه برنامه امتحانی دروس را در جدول ۳-۱۰ مشاهده می‌کنید.

در حالت۲، اگر تعداد روزهای امتحانی ۳ و کوچکتر از تعداد گروه‌ها(عدد ۴) در نظر بگیریم، گروه‌های $G۱$ و $G۳$ را در دو روز و گروه‌های $\{G۲, G۴\}$ را در یک روز در دو بازه زمانی همان روز برگزار می‌کنیم. در نتیجه نقض محدودیت نرم را داریم. یک نمونه برنامه امتحانی دروس را در جدول ۳-۱۱ مشاهده می‌کنید.

در حالت۳، اگر تعداد روزهای امتحانی ۲ و کوچکتر از تعداد گروه‌ها(عدد ۴) در نظر بگیریم، گروه $G۳$ را در یک روز و گروه های $\{G۱, G۲, G۴\}$ را در یک روز در سه نوبت امتحانی همان روز برگزار می‌کنیم. در نتیجه نقض محدودیت نرم را داریم. یک نمونه برنامه امتحانی دروس را در جدول ۳-۱۲ مشاهده می‌کنید.

در حالت۴، اگر تعداد روزهای امتحانی ۱ و کوچکتر از تعداد گروه‌ها(عدد ۴) در نظر بگیریم، گروه‌های $G۱$ و $G۳$ را در دو بازه زمانی به صورت جداگانه و گروه‌های $\{G۲, G۴\}$ را در یک بازه زمانی به صورت هم زمان برگزار می‌کنیم. در نتیجه نقض محدودیت سخت و نرم را داریم. یک نمونه برنامه امتحانی دروس را در جدول ۳-۱۳ مشاهده می‌کنید.

جدول ۳-۱۰: نمونه برنامه امتحانی دروس **در حالت۱** در مثال ۳-۷-۱ . مجموعه C دروس، مجموعه S دانشجویان و G گروه‌بندی اولیه دروس هستند.

روز	بازه زمانی۱	بازه زمانی۲	بازه زمانی۳
۱	$\overbrace{\{\overbrace{S۱, S۲, S۱۱}^{C۱}, \overbrace{S۱۰}^{C۲}, \overbrace{S۴, S۷, S۸}^{C۸}\}}^{G۱}$	-	-
۲	$\overbrace{\{\overbrace{S۱, S۳, S۶}^{C۷}, \overbrace{S۱۱}^{C۹}, \overbrace{S۲, S۷, S۹}^{C۱۰}\}}^{G۲}$	-	-
۳	$\overbrace{\{\overbrace{S۱, S۶, S۹, S۱۰}^{C۳}, \overbrace{S۵, S۷, S۱۱, S۱۲}^{C۴}, \overbrace{S۲, S۳, S۴}^{C۵}\}}^{G۳}$	-	-
۴	$\overbrace{\{\overbrace{S۲, S۴, S۷, S۱۲}^{C۶}\}}^{G۴}$	-	-

جدول ۳-۱۱: نمونه برنامه امتحانی دروس **در حالت۲** در مثال ۳-۷-۱ . مجموعه C دروس، مجموعه S دانشجویان و G گروه‌بندی اولیه دروس هستند.

روز	بازه زمانی۱	بازه زمانی۲	بازه زمانی۳
۱	$\overbrace{\{\overbrace{S۱, S۳, S۶}^{C۷}, \overbrace{S۱۱}^{C۹}, \overbrace{S۲, S۷, S۹}^{C۱۰}\}}^{G۲}$	$\overbrace{\{\overbrace{S۲, S۴, S۷, S۱۲}^{C۶}\}}^{G۴}$	-
۲	$\overbrace{\{\overbrace{S۱, S۲, S۱۱}^{C۱}, \overbrace{S۱۰}^{C۲}, \overbrace{S۴, S۷, S۸}^{C۸}\}}^{G۱}$	-	-
۳	$\overbrace{\{\overbrace{S۱, S۶, S۹, S۱۰}^{C۳}, \overbrace{S۵, S۷, S۱۱, S۱۲}^{C۴}, \overbrace{S۲, S۳, S۴}^{C۵}\}}^{G۳}$	-	-
۴	-	-	-

جدول ۳-۱۲: نمونه برنامه امتحانی دروس **در حالت۳** در مثال ۳-۷-۱ . مجموعه C دروس، مجموعه S دانشجویان و G گروه‌بندی اولیه دروس هستند.

روز	بازه زمانی۱	بازه زمانی۲	بازه زمانی۳
۱	$\overbrace{\{\overbrace{S۱, S۲, S۱۱}^{C۱}, \overbrace{S۱۰}^{C۲}, \overbrace{S۴, S۷, S۸}^{C۸}\}}^{G۱}$	$\overbrace{\{\overbrace{S۱, S۳, S۶}^{C۷}, \overbrace{S۱۱}^{C۹}, \overbrace{S۲, S۷, S۹}^{C۱۰}\}}^{G۲}$	$\overbrace{\{\overbrace{S۲, S۴, S۷, S۱۲}^{C۶}\}}^{G۴}$
۲	$\overbrace{\{\overbrace{S۱, S۶, S۹, S۱۰}^{C۳}, \overbrace{S۵, S۷, S۱۱, S۱۲}^{C۴}, \overbrace{S۲, S۳, S۴}^{C۵}\}}^{G۳}$	-	-
۳	-	-	-
۴	-	-	-

جدول ۳-۱۳: نمونه برنامه امتحانی دروس **در حالت۴** در مثال ۳-۷-۱ . مجموعه C دروس، مجموعه S دانشجویان و G گروه‌بندی اولیه دروس هستند.

روز	بازه زمانی۱	بازه زمانی۲	بازه زمانی۳
۱	$\overbrace{\{\overbrace{S۱, S۲, S۱۱}^{C۱}, \overbrace{S۱۰}^{C۲}, \overbrace{S۴, S۷, S۸}^{C۸}\}}^{G۱}$	$\overbrace{\{\overbrace{S۱, S۶, S۹, S۱۰}^{C۳}, \overbrace{S۵, S۷, S۱۱, S۱۲}^{C۴}, \overbrace{S۲, S۳, S۴}^{C۵}\}}^{G۳}$	$\overbrace{\{\overbrace{S۲, S۴, S۷, S۱۲}^{C۶}\}}^{G۴}$، $\overbrace{\{\overbrace{S۱, S۳, S۶}^{C۷}, \overbrace{S۱۱}^{C۹}, \overbrace{S۲, S۷, S۹}^{C۱۰}\}}^{G۲}$
۲	-	-	-
۳	-	-	-
۴	-	-	-

فهرست منابع

[1] Carter, Michael W and Laporte, Gilbert. Recent developments in practical course timetabling, 1997.

[2] Socha, Krzysztof, Sampels, Michael, and Manfrin, Max. Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. in *Workshops on Applications of Evolutionary Computation*, pp. 334–345. Springer, 2003.

[3] Amintoosi, Mahmood and Haddadnia, Javad. Feature selection in a fuzzy student sectioning algorithm. in *International Conference on the Practice and Theory of Automated Timetabling*, pp. 147–160. Springer, 2004.

[4] Chen, Ruey-Maw and Shih, Hsiao-Fang. Solving university course timetabling problems using constriction particle swarm optimization with local search. *Algorithms*, 6(2):227–244, 2013.

[5] Nurmi, Kimmo and Kyngäs, Jari. A conversion scheme for turning a curriculum-based timetabling problem into a school timetabling problem. in *Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008)*, 2008.

[6] Suyanto, Suyanto. An informed genetic algorithm for university course and student timetabling problems. in *Proceedings of the 10th international conference on Artifical intelligence and soft computing: Part II*, pp. 229–236. Springer-Verlag, 2010.

[7] Ceschia, Sara, Di Gaspero, Luca, and Schaerf, Andrea. Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem. *Computers & Operations Research*, 39(7):1615–1624, 2012.

[8] Daskalaki, Sophia and Birbas, Theodore. Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160(1):106–120, 2005.

[9] Avella, Pasquale and Vasil'Ev, Igor. A computational study of a cutting plane algorithm for university course timetabling. *Journal of Scheduling*, 8(6):497–514, 2005.

[10] Al-Yakoob, Salem M and Sherali, Hanif D. A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations. *European Journal of Operational Research*, 180(3):1028–1044, 2007.

[11] Nothegger, Clemens, Mayer, Alfred, Chwatal, Andreas, and Raidl, Günther R. Solving the post enrolment course timetabling problem by ant colony optimization. *Annals of Operations Research*, 194(1):325–339, 2012.

[12] Feng, Jiaxiao, Ye, Zhirui, Wang, Chao, Chang, Cheng, Xu, Mingtao, and Sun, Cuicui. Optimising departure intervals for multiple bus lines with a multi-objective model. *IET Intelligent Transport Systems*, 12(8):809–818, 2018.

[13] Hao, Weining, Meng, Lingyun, Corman, Francesco, Long, Sihui, and Jiang, Xi. A train timetabling and stop planning optimization model with passenger demand. in *RailNorrköping 2019. 8th International Conference on Railway Operations Modelling and Analysis (ICROMA), Norrköping, Sweden, June 17th–20th, 2019*, no. 069, pp. 390–406. Linköping University Electronic Press, 2019.

[14] Zhang, Yanan, Hu, Qinghua, Meng, Zhaopeng, and Ralescu, Anca. Fuzzy dynamic timetable scheduling for public transit. *Fuzzy Sets and Systems*, 2019.

[۱۵] تامس، کورمن، چارلز، لیزرسان، رونالد، دیویست، و کلیفورد، اشتفان. مقدمه ای بر الگوریتم ها. ترجمه‌ی روانخواه، مهدی، عالمی، محمود، اسدی،محسن، شبنم، جعفرخانی،، شادی، لنگری،، و الهام، صدر،. نشر درخشش، مشهد، ویرایش سوم، تابستان ۱۳۸۳.

[۱۶] هان، ژیاوی، کمبر، میشلین، و پی، ژان. داده کاوی(مفاهیم و تکنیک ها). ترجمه‌ی اسماعیلی، مهدی. نشر نیاز دانش، تهران، ویرایش سوم، ۱۳۹۴.

[17] Lindahl, Michael, Mason, Andrew J, Stidsen, Thomas, and Sørensen, Matias. A strategic view of university timetabling. *European Journal of Operational Research*, 266(1):35–45, 2018.

[18] Fizzano, Perry and Swanson, Steven. Scheduling classes on a college campus. *Computational optimization and applications*, 16(3):279–294, 2000.

[19] Beyrouthy, Camille, Burke, Edmund K, Landa-Silva, Dario, McCollum, Barry, McMullan, Paul, and Parkes, Andrew J. Improving the room-size profiles of university teaching space. in *Dagstuhl Seminar on" Cutting, Packing, Layout and Space Allocation," March. Citeseer*. Citeseer, 2007.

پیوست آ

کد برنامه زمان‌بندی امتحانات با استفاده از روش ابتکاری و حریصانه

در این بخش، کد برنامه زمان‌بندی امتحانات با استفاده از روش ابتکاری و حریصانه (فصل ۳) که به زبان برنامه‌نویسی متلب [۲] است، بیان می‌شود. کد برنامه شامل توابع زیر است :

۱. تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه (الگوریتم ۳-۱)

(آ) تابع برنامه (تابع Main)

۲. تولید گروه‌بندی اولیه دروس (الگوریتم ۳-۲)

(آ) r تابع خواندن اطلاعات انتخاب واحدها (تابع Function_ Read)

(ب) تابع تولید ماتریس مجاورت و ماتریس فراوانی دروس (تابع Function_CreateM2MA)

(ج) تابع تولید گروه‌بندی اولیه امتحانات (تابع Function_CreatePreGExam)

۳. تولید برنامه نهایی امتحانات (الگوریتم ۳-۳)

(آ) تابع تولید برنامه نهایی امتحانات (تابع Function_AssignCoursesToExamTimes)

(ب) تابع محاسبه میزان نقض محدودیت سخت و نرم (تابع Function_CalHardConstraintDefect)

ابتدا در بخش اول، متغیرهای استفاده شده در کد برنامه و سپس در بخش دوم کد متلب برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه را بیان می‌کنیم.

---

[۲] MATLAB

جدول آ-۱: متغیرهای مورد استفاده در کد برنامه زمان‌بندی امتحانات با استفاده از روش ابتکاری و حریصانه

ردیف	اصطلاح	متغیر مورد استفاده
۱	انتخاب واحد دانشجویان	$temp...$
۲	تعداد روزهای امتحانی	$NumDay$
۳	ظرفیت پذیرش نوبت امتحانی	$maxSize$
۴	لیست فراوانی تکرار دروس	$m۱$
۵	لیست انتخاب واحد تفکیکی دانشجو	T
۶	تعداد دروس	$NumCourse$
۷	ماتریس مجاورت دروس	ma
۸	فراوانی دروس	$m۲$
۹	گروه‌بندی اولیه دروس	$GPreExam$
۱۰	تعداد گروه‌های گروه‌بندی اولیه دروس	بر اساس گروه‌بندی اولیه دروس
۱۱	میزان نقض محدودیت‌ها	$tad۱$
۱۲	برنامه نهایی امتحانات	$prog۱$

آ-۱ لیست متغیرهای استفاده شده در کد متلب برنامه تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه

لیست متغیرهای استفاده شده در کد متلب برنامه تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه، در جدول (آ-۱) آمده است.

آ-۲ کد متلب برنامه تولید برنامه زمان‌بندی امتحانات بر اساس روش ابتکاری و روش حریصانه

آ-۲-۱ تابع اصلی برنامه

کد تابع برنامه (Main) در شکل زیر آمده است. این تابع، وظیفه کنترل روال فراخوانی توابع را دارد. همچنین دو متغیر تعداد روزهای امتحانی و ظرفیت پذیرش نوبت امتحانی، مقداردهی می‌شوند. در خط ۳ تابع، مجموعه داده انتخاب واحدهای دانشجویان بارگذاری می‌شود. در نهایت میزان تخطی از محدودیت‌های سخت و نرم برنامه نهایی امتحانات چاپ می‌شود.

```
1 clear all;close all;clc;
2 %%  step1
```

```
3  [m1,NumCourse,T,NumSelectedCourse,m11]=Function_Read('temp2
       ');
4  if(NumCourse~=-1)
5      %%  step2
6      [m2,ma]=Function_CreateM2MA(NumCourse,T);
7      if(m2~=-1)
8          %%  step3
9          NumDay=15;
10         maxSize=450;
11         [s,GPreExam] = Function_CreatePreGExam(NumCourse,
               NumDay,ma);
12
13         if(max(GPreExam)<NumDay)
14             NumDay=max(GPreExam);
15         end
16         if(GPreExam~=-1)
17             %%  step4
18             [tad,taDay1,prog1] =
                   Function_CalHardConstraintDefect(T,NumDay,
                   GPreExam);
19             if(max(tad(:,1))==0)
20                 disp 'No hard constraint defect step1';
21                 %%  step5
22                 [GExam]=Function_AssignCoursesToExamTimes(s
                       ,NumDay,GPreExam,maxSize,m1,NumCourse,m2
                       ,NumSelectedCourse,m11);
23                 if(GExam~=-1)
24                     [tad1,taDay1,prog1] =
```

```
                    Function_CalHardConstraintDefect(T,
                    NumDay,GExam);
25                  if(max(tad1(:,1))==0)
26                      disp 'No hard constraint defect
                        step2';
27                      sum(tad1)
28                  else
29                      disp 'Error step2';
30                  end
31              else
32                  disp 'Error step2';
33              end
34          else
35              disp 'Error step1';
36          end
37      end
38  end
39  else
40      disp 'Error';
41  end
```

آ-۲-۲ تابع خواندن اطلاعات انتخاب واحدها

کد تابع خواندن اطلاعات انتخاب واحدها (Function_ Read) در زیر آمده است. در این تابع تبدیل ساختار اولیه داده‌ها از فایل ورودی که انتخاب واحدهای دانشجویان است، به ساختار مورد نیاز برای ادامه برنامه به لیست دروس انتخابی به ازاء هر دانشجو تبدیل می‌شود. نوع فایل ورودی، با پسوند mat. است. سایر خروجی‌های این تابع، تعداد کل دروس انتخاب شده توسط دانشجویان و ماتریس فراوانی دروس که همان میزان تعداد دانشجوی ثبت نامی برای هر درس می‌توان اشاره نمود.

```
function [m1,NumCourse,T,NumSelectedCourse,m11] =
   Function_Read(NameDS)
    try
        T1=load(strcat(NameDS,'.mat'));
        T1=struct2cell(T1);
        for i=1:size(T1{1},1)
            m11(T1{1}(i,2),T1{1}(i,1))=1;
        end
        r=1;
        NumCourse=0;
        T{r,1}(1,1)=T1{1}(1,2);
        for i=2:size(T1{1},1)
            if(T1{1,1}(i,1)==T1{1}(i-1,1))
                T{r,1}(1,size(T{r}(1,:),2)+1)=T1{1}(i,2);
                NumSelectedCourse(r,2)=size(T{r,1},2);
            else
                NumSelectedCourse(r,2)=size(T{r,1},2);
                r=r+1;
                T{r,1}(1,1)=T1{1}(i,2);
            end
            T{r,1}=sort(T{r,1});
            if(max(T{r,1})>NumCourse)
                NumCourse=max(T{r,1});
            end
        end
        m1=zeros(NumCourse,1);
        for i=1:size(T1{1},1)
```

```
27              m1(T1{1}(i,2),1)=m1(T1{1}(i,2),1)+1;
28          end
29          T=T;
30          NumSelectedCourse(:,1)=1:size(T,1);
31      catch
32          m1=-1;
33          NumCourse=-1;
34          T=-1;
35          NumSelectedCourse=-1;
36      end
37  end
```

آ-۲-۳ کد تابع تولید ماتریس مجاورت و ماتریس فراوانی دروس

کد تابع تولید ماتریس مجاورت و ماتریس فراوانی دروس (Function_CreateM2MA) در شکل زیر آمده است.

```
1   function [m2,ma] = Function_CreateM2ma(NumCourse,T)
2       try
3           m2=zeros(NumCourse);
4           ma=zeros(NumCourse);
5           for i=1:size(T,1)
6               if(size(T{i,1}(:,:),2)>1)
7                   for j=1:(size(T{i,1}(:,:),2)-1)
8                       for k=(j+1):size(T{i,1}(:,:),2)
9                           ma(T{i,1}(1,j),T{i,1}(1,k))=1;
10                          m2(T{i,1}(1,j),T{i,1}(1,k))=m2(T{i
                                ,1}(1,j),T{i,1}(1,k))+1;
11
```

```
12                          ma(T{i,1}(1,k),T{i,1}(1,j))=1;
13                          m2(T{i,1}(1,k),T{i,1}(1,j))=m2(T{i
                                ,1}(1,k),T{i,1}(1,j))+1;
14                      end
15                  end
16              end
17          end
18          for i=1:NumCourse
19              m2(i,i)=0;
20              ma(i,i)=0;
21          end
22      catch
23          m2=-1;
24          ma=-1;
25      end
26  end
```

آ-۲-۴ کد تابع تولید گروه‌بندی اولیه امتحانات

کد تابع تولید گروه‌بندی اولیه امتحانات (Function_CreatePreGExam) در شکل زیر آمده است.

```
1   function [s,GPreExam] = Function_CreatePreGExam(NumCourse,
        NumDay,ma)
2       try
3           s(:,1)=1:NumCourse;
4           s(:,2)=sum(ma);
5           s=flipud(sortrows(s,2));
6           limit=zeros(NumCourse,NumDay*3);
7           for i=1:size(limit,2)
```

```
8               limit(:,i)=i;
9           end
10          limitDay=zeros(NumDay*3,3);
11          limitDay(:,1)=1:(NumDay*3);
12          limitDay(:,2)=ceil(NumCourse/(NumDay*3));
13          limitDay(:,3)=ceil(NumCourse/(NumDay*3));
14          for i=1:size(s,1)
15              GPreExam(s(i,1),1)=min(limit(s(i,1),:));
16              ind=ma(s(i,1),:)>0;
17              limitDay(min(limit(s(i,1),:)),3)=limitDay(min(
                    limit(s(i,1),:)),3)-1;
18              limit(ind,min(limit(s(i,1),:)))=1099;
19          end
20      catch
21          GPreExam=-1;
22          s=-1;
23      end
24  end
```

آ-۲-۵ تابع تولید برنامه نهایی امتحانات

کد تابع تولید برنامه نهایی امتحانات (Function_AssignCoursesToExamTimes) در شکل زیر آمده است.

```
1   function [GExam] = Function_AssignCoursesToExamTimes(s,
        NumDay,GPreExam,maxSize,m1,NumCourse,m2,
        NumSelectedCourse,m11)
2       try
3           t_Cday=zeros(NumDay*3,3);
4           t_Cday(:,1)=1:NumDay*3;
```

```
5          t_s1=sortrows(s,1);
6          t_GExam=zeros(max(GPreExam),7);
7          t_GExam(:,1)=1:max(GPreExam);
8          for i=1:NumCourse
9              t_GExam(GPreExam(i),2)=t_GExam(GPreExam(i),2)
                  +1;
10             t_GExam(GPreExam(i),3)=t_GExam(GPreExam(i),3)+
                  m1(i);
11             t_GExam(GPreExam(i),4)=ceil(t_GExam(GPreExam(i)
                  ,3)/maxSize);
12         end
13         if((NumDay>=max(GPreExam))&&(maxSize>=max(t_GExam
              (:,3))))
14             GExam1=GPreExam;
15         elseif((NumDay>=max(GPreExam))&&((maxSize*3)>=max(
              t_GExam(:,3))))
16             [GExam1]=Function_Case2(NumDay,maxSize,
                  NumCourse,t_GExam,GPreExam,m1);
17         elseif((NumDay<max(GPreExam))&&(maxSize>=max(
              t_GExam(:,3))))
18             if(size(t_GExam,1)<=2*NumDay)
19                 A=nchoosek(1:size(t_GExam),2);
20                 A(:,3)=0;
21                 for i=1:size(A,1)
22                     GPreExam1=GPreExam;
23                     ind1=GPreExam(:,1)~=A(i,1);
24                     ind2=GPreExam(:,1)~=A(i,2);
25                     t_m2=m2;
```

```
26                  t_m2(ind1,:)=[];
27                  t_m2(:,ind2)=[];
28                  A(i,3)=mean(t_m2(:));
29                  A(i,4)=sum(t_m2(:));
30                  A(i,5)=size(t_m2,1)+size(t_m2,2);
31                  A(i,6)=A(i,5)-(sum(sum(t_m2)==0)+sum(
                        sum(t_m2')==0));
32              end
33              A=(sortrows(A,3));%flipud
34              n2t=mod(size(t_GExam,1),NumDay);
35              CL=(1:size(t_GExam,1))';
36              GPreExam1=GPreExam;
37              GPreExam1(:,1)=0;
38              for i=1:n2t
39                  x1=A(i,1);
40                  x2=A(i,2);
41                  for j=1:2
42                      ind1=GPreExam(:,1)==A(i,j);
43                      GPreExam1(ind1,1)=i+(j-1)*NumDay;
44                      if(size(find(CL(:,1)==A(i,j)),1)>0)
45                          CL(find(CL(:,1)==A(i,j)),:)=[];
46                      end
47                  end
48                  A(A(:,1)==x1,:)=[];
49                  A(A(:,2)==x1,:)=[];
50                  A(A(:,1)==x2,:)=[];
51                  A(A(:,2)==x2,:)=[];
52              end
```

```
53                  for i=1:size(CL,1)
54                      ind1=GPreExam(:,1)==CL(i,1);
55                      GPreExam1(ind1,1)=i+n2t;
56                  end
57                  GExam1=GPreExam1;
58              end
59          elseif((NumDay<max(GPreExam))&&(maxSize<max(
                t_GExam(:,3))))
60              G2j=zeros(size(t_GExam,1),1);
61              id_G2j=1;
62              G1j=zeros(size(t_GExam,1),1);
63              id_G1j=1;
64              for i=1:size(t_GExam,1)
65                  if(t_GExam(i,4)==2)
66                      G2j(id_G2j,1)=i;
67                      id_G2j=id_G2j+1;
68                  elseif(t_GExam(i,4)==1)
69                      G1j(id_G1j,1)=i;
70                      id_G1j=id_G1j+1;
71                  end
72                  t_GExam(i,5)=fix(t_GExam(i,4)/3);
73                  if((mod(t_GExam(i,4),3)==2))
74                      t_GExam(i,6)=1;
75                  end
76                  if((mod(t_GExam(i,4),3)==1))
77                      t_GExam(i,7)=1;
78                  end
79              end
```

```
80              id_G1j=1;
81              G2j(G2j(:,1)==0)=[];
82              G1j(G1j(:,1)==0)=[];
83              for i=1:size(G2j,1)
84                  for j=1:size(G1j)
85                      TableOverlap(id_G1j,1)=G2j(i,1);
86                      TableOverlap(id_G1j,2)=G1j(j,1);
87                      TableOverlap(id_G1j,3)=m2(G2j(i,1),G1j(
                            j,1));
88                      TableOverlap(id_G1j,4)=2;
89                      TableOverlap(id_G1j,5)=1;
90                      id_G1j=id_G1j+1;
91                  end
92              end
93              TableOverlap=(sortrows(TableOverlap,3));%flipud
94              if((NumDay<max(GPreExam))&&((NumDay*3)>=sum(
                    t_GExam(:,4))))
95                  [GExam1]=Function_Case5(s,NumDay,GPreExam,
                        maxSize,m1,NumCourse,m2,
                        NumSelectedCourse,t_Cday,t_s1,t_GExam,
                        TableOverlap);
96              end
97          end
98          GExam=GExam1;
99          m1(:,2)=1:NumCourse;
100         m1(:,3)=GExam;
101         sd=zeros(max(m1(:,3)),2);
102         sd(:,1)=1:max(m1(:,3));
```

```
103         for i=1:max(m1(:,3))
104             m120=m1;
105             m120(m120(:,3)~=i,:)=[];
106             sd(i,2)=sum(m120(:,1));
107         end
108         for i=1:NumDay
109             if((i<=size(sd,1))&&((i+NumDay)<=size(sd,1)))
110                 if(sd(i,2)<sd(i+NumDay,2))
111                     ind1=GExam==i;
112                     GExam(ind1)=1000+i;
113                     ind2=GExam==(i+NumDay);
114                     GExam(ind2)=1000+i+NumDay;
115
116                     GExam(ind1)=i+NumDay;
117                     GExam(ind2)=i;
118                 end
119             end
120             if(((i+NumDay)<=size(sd,1))&&((i+2*NumDay)<=
                   size(sd,1)))
121                 if(sd(i+NumDay,2)<sd(i+2*NumDay,2))
122                     ind1=GExam==(i+NumDay);
123                     GExam(ind1)=1000+i+NumDay;
124                     ind2=GExam==(i+2*NumDay);
125                     GExam(ind2)=1000+i+2*NumDay;
126
127                     GExam(ind1)=i+2*NumDay;
128                     GExam(ind2)=i+NumDay;
129                 end
```

```
130                 end
131                 if((i<=size(sd,1))&&((i+2*NumDay)<=size(sd,1)))
132                     if(sd(i,2)<sd(i+2*NumDay,2))
133                         ind1=GExam==i;
134                         GExam(ind1)=1000+i;
135                         ind2=GExam==(i+2*NumDay);
136                         GExam(ind2)=1000+i+2*NumDay;
137 
138                         GExam(ind1)=i+2*NumDay;
139                         GExam(ind2)=i;
140                     end
141                 end
142             end
143             m120;
144         catch
145             GExam=-1;
146         end
147 end
148 %
    ..........................................................................

149 function [GExam] = Function_Case2(NumDay,maxSize,NumCourse,
    t_GExam,GPreExam,m1)
150         try
151             GExam1=GPreExam;
152             GExam1(:,1)=0;
153             GPreExam(:,2)=1:size(GPreExam,1);
154             GPreExam(:,3)=m1;
```

```
155         GPreExam=flipud(sortrows(GPreExam,3));
156         for i=1:size(t_GExam,1)
157             maxSize1=maxSize;
158             GPreExam1=GPreExam;
159             ind1=GPreExam(:,1)==t_GExam(i,1);
160             ind2=GPreExam(:,1)~=t_GExam(i,1);
161             GPreExam1(ind2,:)=[];
162             d=i;
163             for j=1:size(GPreExam1,1)
164                 if((maxSize1-m1(GPreExam1(j,2)))>=0)
165                     maxSize1=maxSize1-m1(GPreExam1(j,2));
166                     GExam1(GPreExam1(j,2),1)=d;
167                 else
168                     maxSize1=maxSize;
169                     d=d+NumDay;
170                     maxSize1=maxSize1-m1(GPreExam1(j,2));
171                     GExam1(GPreExam1(j,2),1)=d;
172                 end
173             end
174         end
175         GExam=GExam1;
176     catch
177         GExam=-1;
178     end
179 end
180 %
```

```
181 function [GExam] = Function_Case5(s,NumDay,GPreExam,maxSize
    ,m1,NumCourse,m2,NumSelectedCourse,t_Cday,t_s1,t_GExam,
    TableOverlap)
182     try
183         t_GPreExam=GPreExam;
184         t_GPreExam(:,1)=0;
185 
186         GPreExam(:,2)=1:size(GPreExam,1);
187         GPreExam(:,3)=m1;
188         GPreExam(:,4)=0;
189 
190         NumDay1=1;
191 
192         t_GExam1=flipud(sortrows(t_GExam,4));
193         b=1;
194         while((size(t_GExam1,1)>0)&&(max(t_GExam1(:,4))>1)
            &&(b==1)&&(sum(GPreExam(:,4)==0)>0)&&(NumDay1<=
            NumDay))
195             if((size(t_GExam1,1)>0)&&(t_GExam1(1,4)>1))
196                 GPreExam1=GPreExam;
197                 ind1=GPreExam1(:,1)==t_GExam1(1,1);
198                 ind2=GPreExam1(:,1)~=t_GExam1(1,1);
199                 GPreExam1(ind2,:)=[];
200                 maxSize1=maxSize;
201                 NumDay2=NumDay1;
202                 for i=1:size(GPreExam1,1)
203                     if((maxSize1-GPreExam1(i,3))>=0)
204                         GPreExam(GPreExam1(i,2),4)=NumDay2;
```

```
205                         t_Cday(NumDay2,2)=1;
206                         maxSize1=maxSize1-GPreExam1(i,3);
207                     else
208                         maxSize1=maxSize;
209                         NumDay2=NumDay2+NumDay;
210                         GPreExam(GPreExam1(i,2),4)=NumDay2;
211                         t_Cday(NumDay2,2)=1;
212                         maxSize1=maxSize1-GPreExam1(i,3);
213                     end
214                 end
215                 t_GExam1(1,:)=[];
216                 NumDay1=NumDay1+1;
217             else
218                 if(size(t_GExam,1)==0)
219                     b=0;
220                 end
221             end
222         end
223         t_Cday;
224         t_Cday1=t_Cday;
225         t_Cday1(t_Cday1(:,2)>0,:)=[];
226         t_GExam1=sortrows(t_GExam1,4);
227         for i=1:size(t_GExam1,1)
228             if(t_GExam1(1,4)>1)
229                 ind1=GPreExam(:,1)==t_GExam1(1,1);
230                 ind2=GPreExam(:,1)~=t_GExam1(1,1);
231                 GPreExam2=GPreExam1;
232                 GPreExam2(ind2,:)=[];
```

```
233                 maxSize1=maxSize;
234                 for i=1:size(GPreExam2,1)
235                     if((maxSize1-GPreExam2(i,3))>=0)
236                         maxSize1=maxSize1-GPreExam2(i,3);
237                         GPreExam(GPreExam2(i,2),4)=t_Cday1
                                (1,1);
238                     else
239                         maxSize1=maxSize;
240                         t_Cday1(1,:)=[];
241                         GPreExam(GPreExam2(i,2),4)=t_Cday1
                                (1,1);
242                     end
243                 end
244             elseif(t_GExam1(1,4)==1)
245                 GPreExam1=GPreExam;
246                 ind1=GPreExam1(:,1)==t_GExam1(1,1);
247                 GPreExam(ind1,4)=t_Cday1(1,1);
248                 t_GExam1(1,:)=[];
249                 t_Cday1(1,:)=[];
250             end
251         end
252         GExam=GPreExam(:,4);
253     catch
254         GExam=-1;
255     end
256 end
```

آ-۲-۶ تابع محاسبه میزان نقض محدودیت سخت و نرم

کد تابع محاسبه میزان نقض محدودیت سخت و نرم (Function_CalHardConstraintDefect) در شکل زیر آمده است.

```
function [tad1,taDay1,prog1] =
   Function_CalHardConstraintDefect(T,NumDay,GExam)
    try
        tad=zeros(size(T,1),5);
        for i=1:size(T,1)
            t(1,1:NumDay*3)=0;
            for j=1:size(T{i,1},2)
                prog{i,1}(1,j)=GExam(T{i,1}(1,j));
                t(1,GExam(T{i,1}(1,j)))=t(1,GExam(T{i,1}(1,
                   j)))+1;
            end
            prog{i,1}=sort(prog{i,1});
            tad(i,1)=sum(t>1);
        end
        prog1=prog;
        taDay=zeros(NumDay,4);
        for i=1:size(prog,1)
            for j=1:NumDay
                if((size(find(prog{i,1}(1,:)==j),2)>0)&&(
                   size(find(prog{i,1}(1,:)==(j+NumDay)),2)
                   >0)&&(size(find(prog{i,1}(1,:)==(j+2*
                   NumDay)),2)>0))
                    tad(i,2)=tad(i,2)+1;
                    mod1=mod(i,NumDay);
```

```
20                  if(mod1==0)
21                      mod1=NumDay;
22                  end
23                  taDay(mod1,1)=taDay(mod1,1)+1;
24              elseif((size(find(prog{i,1}(1,:)==j),2)>0)
                &&(size(find(prog{i,1}(1,:)==(j+NumDay))
                ,2)>0)&&(size(find(prog{i,1}(1,:)==(j+2*
                NumDay)),2)==0))
25                  tad(i,3)=tad(i,3)+1;
26                  mod1=mod(i,NumDay);
27                  if(mod1==0)
28                      mod1=NumDay;
29                  end
30                  taDay(mod1,2)=taDay(mod1,2)+1;
31              elseif((size(find(prog{i,1}(1,:)==(j+NumDay
                )),2)>0)&&(size(find(prog{i,1}(1,:)==(j
                +2*NumDay)),2)>0)&&(size(find(prog{i
                ,1}(1,:)==j),2)==0))
32                  tad(i,4)=tad(i,4)+1;
33                  mod1=mod(i,NumDay);
34                  if(mod1==0)
35                      mod1=NumDay;
36                  end
37                  taDay(mod1,3)=taDay(mod1,3)+1;
38              elseif((size(find(prog{i,1}(1,:)==j),2)>0)
                &&(size(find(prog{i,1}(1,:)==(j+2*NumDay
                )),2)>0)&&(size(find(prog{i,1}(1,:)==(j+
                NumDay)),2)==0))
```

```
39                      tad(i,5)=tad(i,5)+1;
40                      mod1=mod(i,NumDay);
41                      if(mod1==0)
42                          mod1=NumDay;
43                      end
44                      taDay(mod1,4)=taDay(mod1,4)+1;
45                  end
46              end
47          end
48          tad1=tad;
49          taDay1=taDay;
50      catch
51          taDay1=-1;
52          tad1=-1;
53          prog1=-1;
54      end
55  end
```

واژه‌نامه فارسی به انگلیسی

اپریوری . . . . . Apriori Rule Mining

استراتژیک . . . . . Strategic

بازه‌های تدریس . . . . . Teaching Periods

بدون تداخل . . . . . Conflict-Free

برنامه‌ریزی اتاق‌ها . . . . . Room Planning

تاکتیکی . . . . . Tactical

جواب شدنی . . . . . Feasible

خوشه‌بندی . . . . . Clustring

دسته‌بندی . . . . . Classification

ظرفیت اتاق . . . . . Room Capacity

قوانین انجمنی . . . . . Association Rule

ویژگی . . . . . Feature

یادگیری با نظارت . . . . . Superviseed Learning

یادگیری بدون نظارت . . . . . Unsuperviseed Learning

یادگیری ماشین . . . . . Machin Learning

واژه‌نامه انگلیسی به فارسی

اپریوری . . . . . Apriori Rule Mining

قوانین انجمنی . . . . . Association Rule

دسته‌بندی . . . . . Classification

خوشه‌بندی . . . . . Clustring

بدون تداخل . . . . . Conflict-Free

جواب شدنی . . . . . Feasible

ویژگی . . . . . Feature

یادگیری ماشین . . . . . Machin Learning

ظرفیت اتاق . . . . . Room Capacity

برنامه‌ریزی اتاق‌ها . . . . . Room Planning

استراتژیک . . . . . Strategic

یادگیری با نظارت . . . . . Superviseed Learning

تاکتیکی . . . . . Tactical

بازه‌های تدریس . . . . . Teaching Periods

یادگیری بدون نظارت . . . . . Unsuperviseed Learning

Hakim Sabzevari University

An Outline of MSc. Thesis

Surname:Rasaeizade	Name:Eisa	Student No.:9613137013
Supervisors: Dr. Mahmood Amintoosi and Dr. Morteza Jafarzadeh		
Advisor: Dr. Alireza Ghodsi		
Faculty of Mathematics and Computer Science	Decision Science and Knowledge Engineering	Operational Research
Title of thesis: An Approach Based on Multi-objective Optimization for University Course Timetabling Problem		
Keywords:Timetabling,Multiple objective programming,Integer programming		

Abstract: The scheduling issue is one of the important topics that are used in various areas such as curriculum, lesson exams, bus transportation, office staff shifts, factory workers shift and rail network and trains. In this thesis, a multi-objective optimization-based approach to the problem of scheduling university courses is presented using mixed integer programming. Curriculum is the allocation of class sessions to roommates and instructors so that some limitations are met. The restrictions are divided into two categories: hard and soft. Adhering to strict mandatory restrictions and non-compliance with soft restrictions will reduce the quality and efficiency of the final program and thus reduce the satisfaction of faculty and students. In the issue of curriculum planning, the goal is to devote classroom sessions with maximum coverage to the interests of faculty and students in a coherent and compact timeframe and class schedule.

Hakim Sabzevari University

Faculty of Mathematics and Computer Science

A Thesis Submitted in Partial Fulfilment of the Requirement for the Degree of Master of Science in Decision Science and Knowledge Engineering

An Approach Based on Multi-objective Optimization for University Course Timetabling Problem

Supervisors:

Dr. Mahmood Amintoosi and Dr. Morteza Jafarzadeh

Advisor:

Dr. Alireza Ghodsi

By:

Eisa Rasaeizade

January 2020