مجله مدل‌سازی پیشرفته ریاضی
سال ۱۴۰۱، دوره ۱۲، شماره ۱، ص ۹۰-۱۰۷

شاپا چاپی: ۲۲۵۱-۸۰۸۸
DOI: 10.22055/JAMM.2022.39456.1990

یک مدل برنامه‌ریزی چندهدفه برای مساله برنامه‌ریزی دروس دانشگاهی

سید علیرضا حسینی دهمیری [a] [۱]، محمد عارف صمدی [b]

[a,b] گروه ریاضی، دانشکده علوم ریاضی، دانشگاه ولی عصر (عج) رفسنجان، رفسنجان، ایران

دبیر مسئول: سهراب عفتی

تاریخ دریافت: ۱۴۰۰/۹/۲۳ تاریخ پذیرش: ۱۴۰۰/۱۲/۲۱

چکیده: برنامه‌ریزی دروس دانشگاهی یکی از دغدغه‌های مدیران گروه در شروع هر ترم تحصیلی است. محدودیت‌های فراوان در رابطه با دروس، کلاس، اساتید و گروه‌های دانشجویی باعث شده است که پیچیدگی محاسباتی این مساله بسیار بالا بوده و در گروه مسائل NP–سخت قرار گیرد. در این مقاله سعی شده است این مساله با روی‌کرد برنامه‌ریزی چندهدفه صفر و یک مورد بررسی قرار گرفته و با تعریف حداقل متغیرهای مورد نیاز، مدل ریاضی مساله نوشته شود. روی‌کرد انتخابی برای برخورد با مساله روی‌کرد مبتنی بر فعالیت است تا تعداد متغیرهای لازم برای مدل‌سازی آن در سطح قابل قبولی باشد.

واژه‌های کلیدی: مسائل برنامه‌ریزی چندهدفه، مساله برنامه‌ریزی دروس دانشگاه، برنامه‌ریزی صفر و یک.

رده‌بندی ریاضی: 90C10; 90C29

۱ مقدمه

زمان‌بندی دروس از شاخه‌های مهم و پرکاربرد مساله عمومی زمان‌بندی است و در گروه مسائل بهینه‌سازی ترکیبیاتی قرار دارد. در واقع زمان‌بندی یک فرایند تصمیم‌گیری است که در آن به توزیع مجموعه‌ای از فعالیت‌ها، در بازه‌های زمانی معین پرداخته می‌شود تا ضمن رعایت قوانین حاکم بر مساله، اهداف خاص متناسب با آن، برآورده شود. در بین مسائل زمان‌بندی، مساله زمان‌بندی دروس دانشگاهی بیش‌تر از بقیه مورد توجه قرار گرفته است.

امروزه یکی از مشکلات برای مدیران گروه در دانشگاه‌های کشور، برنامه‌ریزی دروس در ابتدای هر نیم‌سال تحصیلی است. در اکثر موارد این کار به‌صورت دستی انجام می‌شود و نتیجه حاصل از سطح رضایت‌مندی بالایی برخوردار نیست. این مساله دارای پیچیدگی‌های خاص و چالش‌های فراوان است که اغلب ناشی از رعایت هم‌زمان ضابطه‌های گروه و قوانین آموزشی در کنار محدودیت منابع است. رشد سریع اندازه مساله با افزایش تعداد دروس باعث می‌شود که حل این مساله با چالش‌های فراوانی روبرو شود و مساله در زمان چندجمله‌ای قابل حل نبوده و در گروه مسائل NP–سخت قرار بگیرد [۳، ۲۵].

اگر بخواهیم یک تعریف کلی از مساله زمان‌بندی دروس بیاوریم، می‌توان گفت که این مساله قراردادن مجموعه‌ای از درس‌ها در تعداد ثابتی از بازه‌های زمانی یک مجموعه از کلاس‌ها است به‌نحوی که مغایرتی با قوانین آموزشی دانشگاه نداشته باشد. واضح است که هر دانشگاه

[۱] نویسنده مسئول مقاله
رایانامه: dehmiry@vru.ac.ir (A. H. Dehmiry), samadi.ma@vru.ac.ir (M. A. Samadi)

به‌دلایل مختلفی مانند سیاست‌های تعیین‌شده توسط موسسه و سیستم آموزشی کشور یا منطقه‌ی مربوطه، مساله زمان‌بندی منحصر به فرد خود را دارد و در نتیجه نیازمند به مدل‌سازی اختصاصی برای حل این مساله خواهد بود.

با نگاه به پیشینه برخورد با این مساله در می‌یابیم که پژوهش‌گران، روی‌کردهای مختلفی را برای حل این مساله آزموده‌اند. شاید بتوان گفت که مقاله‌ی گاتلیب در سال ۱۹۶۳ [۱۴]، یکی از اولین تلاش‌های انجام‌شده برای بررسی این مساله است. از آن زمان به بعد پژوهش‌گران متعددی با روی‌کردهای متفاوت به مطالعه و حل این مساله پرداخته‌اند. در ادامه به معرفی شش گروه از روی‌کردهای برخورد با مساله‌ی برنامه‌ریزی دروس که تاکنون مورد استفاده قرار گرفته‌اند، می‌پردازیم.

گروه اول را می‌توان تکنیک‌های مبتنی بر تحقیق در عملیات نام نهاد که از روش‌هایی چون برنامه‌ریزی عدد صحیح [۵]، برنامه‌ریزی عدد صحیح آمیخته [۱۷] و مساله رنگ‌آمیزی گراف [۶] برای حل مساله بهره گرفته‌اند. این تکنیک‌ها در رسیدن به جواب‌های شدنی عملکرد قابل قبولی دارند، اما جواب‌های به‌دست‌آمده در مقایسه با سایر روی‌کردها دارای کیفیت پایین‌تری‌اند [۷]. علاوه بر آن، این تکنیک‌ها در حل مسایل با حجم بالا کارایی مناسبی ندارند [۸].

گروه دوم به روی‌کردهای فراابتکاری مبتنی بر راه حل واحد شهرت دارند که روش جستجوی تابو [۱۸]، روش شبیه‌سازی تبرید [۱۲] و روند جستجوی محلی تکراری [۲۲] از این دسته‌اند. اگرچه این روی‌کردها در دستیابی به جواب‌های با کیفیت بالا، قوی‌اند، اما تنظیم پارامترهای مورد نیاز برای این‌گونه روش‌ها در زمان حل، از مشکلات آن‌ها محسوب می‌شود [۱۳].

گروه بعدی روی‌کردهای فراابتکاری مبتنی بر جمعیت‌اند که الگوریتم‌های ژنتیک [۳]، بهینه‌سازی اجتماع مورچگان [۱۹]، بهینه‌سازی ازدحام ذرات [۹]، ازدحام هوشمندانه ماهی‌ها [۲۴]، الگوریتم اجتماع زنبور عسل [۲۱] و جستجوی محلی مبتنی بر جمعیت [۱] در این گروه قرار می‌گیرند. الگوریتم‌های این گروه در مقایسه با سایرین از نظر یافتن جواب‌های شدنی برتری دارند. با این حال، یکی از اشکالات این روی‌کردها زمان محاسباتی مورد نیاز، برای یافتن جواب‌های با کیفیت خوب است [۲۳].

روی‌کردهای فوق اکتشافی [۱۶]، روی‌کردهای چندهدفه [۱۵] و روی‌کردهای ترکیبی [۲] نیز در سال‌های اخیر برای حل مساله زمان‌بندی دروس مورد استفاده قرار گرفته است. برای آشنایی با نقاط ضعف و قوت این روی‌کردها، خواننده‌ی علاقمند را به [۱۰] ارجاع می‌دهیم.

در این مقاله سعی شده است که از رویکرد برنامه‌ریزی صفر و یک برای حل مدل نهایی استفاده شود. از آن‌جایی که تعداد متغیرهای لازم برای مدل‌کردن مساله در حالت معمولی نسبتا زیاد است، از دسته‌بندی اطلاعات برای کاهش حجم متغیرهای تصمیم استفاده‌شده است. در بخش بعد، جزئیات این دسته‌بندی به‌طور اجمالی آورده شده است.

۲ شناخت مساله

برای مدل‌سازی مساله و انتخاب درست متغیرهای تصمیم لازم است تا جزئیات مساله را بهتر بشناسیم. در ابتدا محدودیت‌های موجود در برنامه‌ریزی دروس آورده شده است. اگر چه ممکن است این محدودیت‌ها برای دانشگاه‌های مختلف متفاوت باشد، اما محدودیت‌های معمول که در برنامه‌ریزی دروس دانشگاه با آن‌ها روبروییم، عبارت‌اند از:

۱. همه‌ی دروس درخواستی توسط مدیر گروه، باید ارائه‌شده باشد؛

۲. تعداد کلاس‌های اختصاص‌یافته به یک بازه‌ی زمانی خاص، نباید از تعداد کلاس‌های در دسترس در آن بازه‌ی زمانی بیش‌تر باشد؛

۳. هنگامی که مدرس یک درس خاص در دسترس نیست، برای وی کلاسی در نظر گرفته نشود؛

۴. هنگامی که یک کلاس خاص اشغال است، در آن بازه‌ی زمانی برای آن کلاس، درسی برنامه‌ریزی نشود؛

۵. ظرفیت کلاس اختصاص داده‌شده برای یک درس نباید از تعداد دانشجویان ثبت‌نام‌شده در آن درس کم‌تر باشد؛

۶. اگر برای ارائه یک درس، زمان یا مکان خاصی به‌تشخیص مدیر گروه مناسب نباشد، در آن زمان‌ها و مکان‌ها درسی ارائه نشود؛

۷. یک استاد در آن واحد نمی‌تواند در بیش از یک کلاس حضور داشته باشد؛

۸. در یک بازه‌ی زمانی حداکثر یک درس برای یک ورودی از دانشجویان اختصاص داده شود؛

۹. زمان ارائه‌ی بعضی از دروس نسبت به دروس دیگر دارای شرایط خاص است. مثلاً بعضی از دروس باید در یک روز واحد ارائه شوند و در نقطه‌ی مقابل دروسی وجود دارند که نباید در یک روز واحد ارائه شوند؛

۱۰. برای دروسی که به‌صورت چندجلسه‌ای در هفته ارائه می‌شوند، رعایت فاصله حداقل یک روز بین جلسات، الزامی است؛

۱۱. زمان و مکان دروس سرویسی در برنامه ثابت است. چون این دروس توسط گروه‌های دیگر درخواست می‌شوند. بنابراین این دسته از دروس نباید در برنامه‌ریزی دخالت داده شوند؛

۱۲. میزان ساعات تدریس هر استاد باید در بازه‌ی کران پایین و کران بالای تدریس قرار گیرد؛

۱۳. هر استاد نباید به‌طور متوالی بیش از ۴ ساعت تدریس در یک روز داشته باشد؛

۱۴. ترجیح اساتید برای زمان تدریس در برنامه‌ریزی در نظر گرفته شود. به‌عنوان مثال اگر یکی از بازه‌های زمانی از نگاه مدرس خوش‌آیندتر نسبت به یک بازه‌ی زمانی دیگر باشد، در صورت امکان این موضوع در برنامه‌ریزی دخالت داده شود؛

۱۵. برنامه‌ریزی باید به‌گونه‌ای باشد که دانشجویان در انتخاب واحد، کم‌ترین مشکل تداخل را داشته باشند؛

۱۶. بهتر است بعضی از دروس که نیاز به‌تمرکز بیش‌تری دارند، در بازه‌هایی قرار گیرند که دانشجویان دارای انرژی بیشتری‌اند؛

با نگاه دقیق‌تر می‌توان دریافت که محدودیت‌های مساله به دو دسته انعطاف‌ناپذیر (سخت) و انعطاف‌پذیر (نرم) قابل‌تقسیم‌اند. محدودیت انعطاف‌ناپذیر به محدودیت‌هایی اطلاق می‌شود که نقض آن‌ها عملا امکان‌پذیر نیست. در مقابل، محدودیت‌های انعطاف‌پذیر قابلیت نقض شدن را دارند ولی نقض هر کدام از آن‌ها میزان معینی ناخوش‌آیندی را به‌همراه دارد. برای کنترل میزان ناخوش‌آیندی، توابع هدف مساله بر اساس محدودیت‌های گروه دوم نوشته می‌شود.

۱.۲ اطلاعات لازم برای مدل‌سازی مساله

با توجه به وجود ساختارهای متفاوت در دروس دانشگاه، جزئیاتی که در خصوص هر درس باید از مدیر گروه دریافت شود، نسبتاً زیاد است و در نگاه اول ممکن است زائد به نظر برسد. اما این جزئیات کمک می‌کند تا متغیرهای مساله زیاد نشوند. هر کدام از ستون‌های در نظر گرفته‌شده، در دسته‌بندی دروسی که دارای شرایط خاص‌اند، قابل استفاده است. این دسته‌بندی‌ها برای مدل‌کردن محدودیت‌ها در بخش مربوط به مدل‌سازی مساله مورد استفاده قرار می‌گیرند. اطلاعاتی که در روند مدل‌سازی به آن احتیاج داریم، به سه دسته تقسیم می‌شوند: اطلاعات دروس، اطلاعات اساتید، اطلاعات کلاس‌های درس.
جزئیات مربوط به اطلاعات اساتید، دروس و کلاس‌ها که باید از کاربر گرفته شوند، به‌ترتیب در جداول ۱، ۲ و ۳ آمده است.

جدول ۱: جدول اطلاعات اساتید.

$Lcode$	شامل تعدادی از کاراکترها که بوسیله آن اساتید را از یکدیگر متمایز می‌کنیم.	کد استاد
$Lname$	نام و نام خانوادگی استاد در این ستون قرار می‌گیرد.	نام استاد
Lj	جدولی است که توسط آن استاد مربوطه زمان‌های آزاد خود را برحسب مشغولیت‌ها و علایق خود، به گروه ارائه می‌دهد در این جدول زمان‌هایی که مدرس نمی‌خواهد تدریس داشته باشد را با صفر و بقیه زمان‌ها را برحسب میزان علاقه خود عددی در بازه‌ی ۱ تا ۵ قرار می‌دهد.	جدول اولویت تدریس
$LTmin$	حداقل ساعاتی که استاد باید تدریس کند. این حداقل بر اساس فاکتورهای مانند مسئولیت اجرایی داشتن، سابقه تدریس و مرتبه علمی مشخص می‌شود.	حداقل ساعات تدریس
$LTmax$	حداکثر ساعاتی که استاد می‌تواند تدریس کند. این حداکثر بر اساس حق‌التدریس یا هیأت علمی بودن استاد مشخص می‌شود.	حداکثر ساعات تدریس
LGE	هیئت علمی یا حق‌التدریس بودن استاد را مشخص می‌کند.	گروه‌بندی استاد
$LGdegree$	عددی است که با توجه به مرتبه علمی، سابقه کار، مدیر اجرایی بودن و غیره برای اساتید لحاظ می‌شود. از این ستون در اولویت بخشی تکمیل برنامه‌ی درسی اساتید در ادامه استفاده می‌شود.	درجه استاد
$Lmas$	این ستون مقادیر ۱ یا ۰ را بسته به‌این‌که عضو هیأت علمی مسئولیت اجرایی دارد یا نه اتخاذ می‌کند.	مسئولیت اجرایی
$Lmah$	اطلاعات مربوط به زمان‌هایی که هیأت علمی حق داشتن کلاس را ندارد.	محدودیت تدریس

۳ دسته‌بندی اطلاعات ورودی

در بیش‌تر روی‌کردهای استفاده‌شده برای حل مساله برنامه‌ریزی دروس، تعداد متغیرهای استفاده‌شده در مدل‌سازی مساله بسیار زیادند. این امر باعث می‌شود که حل مساله با چالش‌های بزرگی روبرو شود. به‌طور کلی می‌توان مدل‌های موجود برای این مساله را به دو گروه مدل‌های

جدول ۲: جدول اطلاعات درس.

$Ccode$	شامل تعدادی از کاراکترها که به‌وسیله آن درس‌ها را از یکدیگر متمایز می‌کنیم.	کد درس
$Cname$	اسمی است که به درس اختصاص داده‌شده است.	نام درس
$Cunit$	شامل عددی است که نسبت به حجم و اهمیت هر درس مشخص می‌گردد.	تعداد واحد درس
$Cteacher$	کد استادی که این درس را ارائه می‌دهد در این ستون قرار می‌گیرد.	کد استاد
$Cgroup$	این ستون مشخص کننده‌ی این است که این درس مربوط به دانشجوی کدام ورودی از کدام مقطع تحصیلی است	گروه دانشجویی
$Cservice$	مقدار این سنون مشخص‌کننده‌ی سرویسی بودن یا نبودن درس است. دروس سرویسی به دروسی اطلاق می‌شوند که توسط گروه‌های دیگر ارائه می‌شوند و زمان ارائه و استاد آن ثابت است.	سرویسی بودن درس
CTS	اطلاعات مربوط به زمان ارائه درس سرویسی در این ستون قرار می‌گیرد.	زمان ارائه درس سرویسی
CPS	اطلاعات مربوط به محل ارائه درس سرویسی در این ستون قرار می‌گیرد.	مکان ارائه درس سرویسی
$Cdegree$	عددی است که براساس میزان تمرکز لازم برای یادگیری درس تعیین می‌گردد.	درجه اهمیت درس
$Cnum$	تعداد کسانی که درس را در ثبت‌نام مقدماتی اخذ کرده‌اند.	تعداد متقاضی
$Ctype$	نوع کلاس بسته به ماهیت درس می‌تواند مقادیر کارگاه، آزمایشگاه، کلاس و بدون نیاز به کلاس را داشته باشد.	نوع کلاس مورد نیاز
$CTNON$	زمان‌هایی که این درس نباید ارائه شود. این زمان‌ها به‌صورت یک جدول گرفته می‌شود.	زمان‌های ممنوعه
$CGTP$	مقدار این سنون نظری یا عملی بودن درس را مشخص می‌کند.	نوع درس
CFY	دروسی که با این درس در یک روز باید ارائه شوند.	همزمانی
CFN	دروسی که با این درس در یک روز نباید ارائه شوند.	عدم تداخل زمانی
CGY	دروسی که با این درس به‌صورت متوالی در یک روز باید ارائه شوند.	رعایت توالی روزانه
CGN	دروسی که با این درس به‌صورت متوالی در یک روز نباید ارائه شوند.	رعایت گسستگی
CH	دروسی که با این درس به‌صورت یکی در نوبت صبح و دیگری در نوبت عصر باید ارائه شوند.	صبح و عصر بودن جلسات
CK	دروسی که نیاز است حداقل فاصله زمانی یک روز در زمان ارائه با این درس رعایت شود.	رعایت فاصله زمانی بین جلسات
$CPar$	دروسی که نیاز به ساعت یکسان در دو روز با این درس دارند.	متقارن بودن
$CPre$	دروسی که نیاز به جدول زمان‌بندی یکی پس از دیگری با این درس دارند.	رعایت توالی هفتگی

جدول ۳: جدول اطلاعات کلاس.

$Rcode$	شامل تعدادی از کاراکترها که بوسیله آن کلاس‌ها را از یکدیگر متمایز می‌کنیم.	کد کلاس
$Rname$	شامل اسمی است که توسط دانشکده به کلاس اختصاص داده‌شده است.	نام کلاس
$Rcapacity$	عددی که تعداد صندلی‌های موجود در کلاس را نشان می‌دهد.	ظرفیت کلاس
$Rtype$	شامل کارگاه، آزمایشگاه، کلاس کوچک، کلاس متوسط، کلاس بزرگ و کلاس خیلی بزرگ است.	نوع کلاس
$Raccess$	ماتریسی که توسط آن بازه‌های زمانی که کلاس در دسترس است، مشخص می‌شود.	ساعات دسترسی

مبتنی بر محدودیت و مدل‌های مبتنی بر فعالیت تقسیم کرد. در گروه اول تعداد متغیرهای به‌کار گرفته‌شده نسبت به گروه دوم بیش‌تر است. به‌همین دلیل مدل‌های گروه دوم در عمل کاربردی‌ترند. برای جزئیات بیش‌تر خواننده را به مراجع [۴، ۱۱، ۲۰] ارجاع می‌دهیم. در این مقاله برای مدل‌سازی مساله از مدل‌سازی مبتنی بر فعالیت استفاده‌شده است. در رویکرد مدل‌سازی مبتنی بر فعالیت، لازم است که داده‌های مساله را به گروه‌های کوچک‌تری تقسیم کنیم تا بتوان با تعداد متغیر کم‌تری مدل‌سازی را انجام داد. برای این منظور در ابتدا داده‌های استفاده‌شده در این مدل، به‌شرح گفته‌شده در جدول ۴، نام‌گذاری می‌شوند.

جدول ۴: دسته‌بندی داده‌های مساله.

C	مجموعه دروس ارائه‌شده در ترم
R	مجموعه کلاس‌های موجود
L	مجموعه اساتید
S	مجموعه گروه‌های دانشجویی
T	مجموعه بازه‌های زمانی هفتگی ($\{۰, ۱, \ldots, ۲۴\}$)

لازم به توضیح است که در این جا S مجموعه گروه‌های دانشجویی است که شامل کاردانی ترم اول تا چهارم $(A_۱, \ldots, A_۴)$، کارشناسی ترم اول تا هشتم $(B_۱, \ldots, B_۸)$، کارشناسی ارشد ترم اول تا چهارم $(M_۱, \ldots, M_۴)$ و دکترا ترم اول تا چهارم $(D_۱, \ldots, D_۴)$ است. برای این مجموعه یک گروه متفرقه (O) را نیز در نظر می‌گیریم تا دروسی که خارج از نمودار[۲] تحصیلی ارائه می‌شوند، در این

[2] chart

گروه‌بندی قرار داده شوند. علاوه بر آن برای T در هر روز هفته (شنبه تا چهارشنبه) ۵ بازه‌ی زمانی دوساعته در نظر گرفته‌شده است و این بازه‌های زمانی از ۰ تا ۲۴ اندیس‌گذاری شده‌اند. در اینجا لازم است دروس ارائه‌شده در ترم را به دسته‌بندی‌های گفته‌شده در جدول ۵ تقسیم کنیم. این تقسیم‌بندی در نوشتن محدودیت‌های مساله به‌کار گرفته می‌شود. نوع کلاس مورد نیاز برای دروس متفاوت است. بعضی از دروس

جدول ۵: جدول گروه‌بندی دروس.

C_t	مجموعه دروس نظری در C_t قرار می‌گیرند. این مجموعه را می‌توان به‌کمک ستون $CGTP$ مقداردهی کرد.
C_p	مجموعه دروس علمی در C_p قرار می‌گیرند. این مجموعه را می‌توان به‌کمک ستون $CGTP$ مقداردهی کرد.
C_r	دروسی که نیاز به کلاسی از نوع کلاس‌های قیدشده در ستون $Rtype$ دارند. برای تعیین این درس، برنامه می‌تواند از ستون‌های $Cnum$ و $Ctype$ استفاده کند. دروس سرویسی که کلاس آن‌ها توسط گروه درخواست‌دهنده تأمین می‌شوند در این شمارش قرار نمی‌گیرند.
C_l	مجموعه دروسی که توسط مدرس $l \in L$ تدریس می‌شود. این مجموعه توسط ستون $Cteacher$ مقداردهی می‌شود.
C_s	مجموعه دروسی که دانشجویان گروه $s \in S$ آن درس را انتخاب می‌کنند. مقادیر این مجموعه توسط ستون $CGgroup$ و $CGmaghta$ و $CGvorodi$ مشخص می‌شوند.
C_{ser}	مجموعه‌ی دروس سرویسی که زمان ارائه آن‌ها توسط گروه پیشنهاددهنده تعیین‌شده است.

برای ارائه نیاز به کلاس درس دارند و بعضی دیگر در کارگاه یا آزمایشگاه ارائه می‌شوند. حتی کلاس‌ها می‌توانند بر اساس ظرفیت کلاس به دسته‌های کوچک، متوسط، بزرگ و خیلی بزرگ تقسیم شوند. بنابراین R را مجموعه

$$\{ \text{کارگاه، آزمایشگاه، کلاس خیلی بزرگ، کلاس بزرگ، کلاس متوسط، کلاس کوچک} \}$$

تعریف می‌کنیم و آن‌ها را با نمادهای اختصاری زیر نمایش می‌دهیم:

$$R = \{S, M, L, XL, A, K\}.$$

متغیرهای متناظر با تعداد کلاس‌های در دسترس در جدول ۶ آمده است. بازه‌های زمانی نامطلوب نیز به دسته‌های بیان‌شده در جدول ۷ تقسیم می‌شوند. برای برنامه‌ریزی دروس به مجموعه‌های دیگری نیاز است که در جدول ۸ به‌معرفی آن‌ها می‌پردازیم. فرض کنیم c_i و c_j

جدول ۶: تعداد کلاس‌های در دسترس بر اساس ظرفیت و نوع آن‌ها.

نوع کلاس	ظرفیت کلاس	تعداد کلاس از این نوع
کلاس کوچک	۶ − ۲۰ صندلی	N_S
کلاس متوسط	۲۱ − ۳۵ صندلی	N_M
کلاس بزرگ	۳۶ − ۵۰ صندلی	N_L
کلاس خیلی‌بزرگ	۵۱ − ۶۵ صندلی	N_{XL}
گارگاه	ظرفیت اسمی کارگاه (n_K)	N_K
آزمایشگاه	ظرفیت اسمی آزمایشگاه (n_A)	N_A

جدول ۷: جدول دسته‌بندی بازه‌های زمانی نامطلوب.

T_s	مجموعه‌های زمان‌های نامطلوب برای ارائه دروس $c \in C_s$ دروس سرویسی ارائه‌شده برای یک گروه دانشجویی خاص نباید با بقیه دروس مربوط به این گروه دانشجویی تداخل داشته باشند.
T_l	زمان‌هایی که مدرس $l \in L$ در دسترس نیست که بر اساس مقادیر ستون $Lmah$ تعیین می‌شوند.
T_c	زمان‌هایی که برای ارائه درس $c \in C$ مناسب نیست که مقدار آن بر اساس ستون $CTNON$ مشخص می‌شود.

دو درس از مجموعه دروس C باشند.

جدول ۸: جدول دسته‌بندی‌های تکمیلی لازم برای مدل‌سازی.

F	مجموعه درس‌هایی به‌صورت (c_i, c_j) که باید حتماً در یک روز ارائه شوند.
F'	مجموعه درس‌هایی به‌صورت (c_i, c_j) که نباید در یک روز ارائه شوند.
G	مجموعه درس‌هایی به‌صورت (c_i, c_j) که باید درس c_j بلافاصله بعد از درس c_i در همان روز در برنامه قرار گیرد.
G'	مجموعه درس‌هایی به‌صورت (c_i, c_j)که نباید به‌صورت متوالی در یک روز قرار بگیرند.
H	مجموعه درس‌هایی به‌صورت (c_i, c_j)که یکی در نوبت صبح و دیگری در نوبت عصر قرار می‌گیرد.
K	مجموعه درس‌هایی به‌صورت (c_i, c_j)که نیاز دارند حداقل فاصله زمانی یک روز در بین زمان ارائه آن‌ها لحاظ شود.
Par	مجموعه درس‌هایی به‌صورت (c_i, c_j)که نیاز به ساعت یکسان در دو روز دارند.
Pre	مجموعه درس‌هایی به‌صورت (c_i, c_j)که نیاز دارند یکی پس از دیگری برنامه‌ریزی شوند.
$LNON$	مجموعه اساتیدی که پست اجرایی ندارند یا حق‌التدریس نیستند که به‌کمک ستون‌های $Lmas$ و LGE مشخص می‌گردد.
$Lday$	مجموعه روزهایی که استاد l در دسترس است. این ستون مجموعه‌ای از اعداد ۰ تا ۴ را می‌تواند بگیرد که بر اساس ستون Lj مشخص می‌شود.
N	جدولی است که در آن در هر بازه‌ی زمانی تعداد کلاس‌ها در دسترس از نوع $r \in R$ را مشخص می‌کند. بنابراین N یک جدول ۶×۲۵ خانه‌ای خواهد بود.
NCS_{max}	جدولی است که در آن ماکزیمم تعداد دروسی که می‌تواند برای یک گروه دانشجویی در هر روز ارائه شود توسط مدیر گروه تعیین می‌شود.

۴ مدل‌سازی مساله برنامه‌ریزی دروس

در این بخش با معرفی یک متغیر تصمیم مناسب به مدل‌سازی مساله برنامه‌ریزی دروس پرداخته‌شده است. ابتدا مجموعه محدودیت‌های مساله که شامل قوانین آموزشی، ساختار برنامه‌ریزی و محدودیت منابع است، به‌تفصیل آورده‌شده و پس از توضیح مختصری در خصوص قید داده‌شده، ضابطه‌ی آن نوشته خواهد شد.

فرض کنیم $X_{c,t}$ متغیر صفر و یکی باشد که اگر درس c در زمان t ارائه شود مقدار یک و در غیر این صورت مقدار صفر را اختیار کند. با تعریف این متغیر مجموعه قیود مساله به‌صورت زیر فرمول‌بندی می‌شوند:

۱. محدودیت ارائه اجباری همه‌ی دروس در ترم:
هر درسی که تعداد حد نصاب ثبت‌نام مقدماتی را به‌تشخیص مدیر گروه داشته باشد، باید در برنامه گنجانده شود. برای اعمال این محدودیت کافی است جمع کلیه‌ی متغیرهای $X_{c,t}$ به‌ازای c ثابت را در کل هفته محاسبه کنیم. اگر این مقدار دقیقاً یک باشد به این معنی است که درس فقط یک‌بار در هفته ارائه‌شده است.

$$\sum_{t \in T} X_{c,t} = ۱, \quad \forall c \in C.$$

۲. محدودیت تعداد کلاس‌های اختصاص‌یافته در زمان t:
در این محدودیت کنترل می‌شود که تعداد کلاس تخصیص‌یافته از نوع $r \in R$ در زمان t از تعداد کل کلاس‌های در دسترس در این بازه‌ی زمانی تجاوز نکند.

$$\sum_{c \in C_r} X_{c,t} \leqslant N_{r,t}, \quad \forall r \in R, \quad \forall t \in T.$$

۳. محدودیت بازه‌های زمانی که مدرس l در دسترس نیست:
در این محدودیت زمان‌هایی که دسترسی به مدرس l وجود ندارد (یعنی $t \in T_l$)، برای وی درسی در برنامه در نظر گرفته نخواهد شد.

$$\sum_{t \in T_l} \sum_{c \in C_l} X_{c,t} = ۰, \quad \forall l \in L.$$

۴. محدودیت بازه‌های زمانی که برای ارائه درس $c \in C$ مناسب نیست:
این محدودیت به‌کمک ستون T_c اجازه‌ی تعریف درس c را در بازه‌ی زمانی $t \in T_c$ نمی‌دهد.

$$\sum_{t \in T_c} X_{c,t} = ۰, \ \ \forall c \in C.$$

۵. محدودیت عدم اختصاص بیش از یک درس به یک مدرس در یک بازه‌ی زمانی:
این محدودیت تعداد کلاس‌های اختصاص‌یافته به مدرس l را در یک بازه‌ی زمانی کنترل می‌کند و اجازه نمی‌دهد که بیش از یک درس در یک بازه‌ی زمانی به مدرس l اختصاص یابد.

$$\sum_{c \in C_l} X_{c,t} \leqslant ۱, \ \ \forall t \in T, \ \ \forall l \in L.$$

۶. محدودیت عدم اختصاص بیش از یک درس در یک بازه‌ی زمانی برای یک گروه دانشجویی خاص:
توسط این محدودیت تعداد دروس اختصاص‌یافته برای یک گروه دانشجویی خاص در یک بازه‌ی زمانی کنترل می‌شود. این امر باعث می‌شود که دانشجویان یک گروه دانشجویی خاص بتوانند دروسی را که مربوط به نمودار تحصیلی آن‌ها است، بدون تداخل انتخاب کنند.

$$\sum_{c \in C_s} X_{c,t} \leqslant ۱, \ \ \forall t \in T, \ \ \forall s \in S.$$

۷. محدودیت ارائه دروس مجموعه F در یک روز واحد:
با اضافه‌کردن قید زیر می‌توان این امر را برای دسته دروسی که در مجموعه F قرار دارند، لحاظ کرد:

$$\sum_{k_۱=۰}^{۴} (X_{c_i,۵k_۲+k_۱} - X_{c_j,۵k_۲+k_۱}) = ۰, \ \ \forall (c_i, c_j) \in F, \ \ k_۲ \in \{۰, \ldots, ۴\}.$$

۸. محدودیت عدم ارائه دروس مجموعه F' در یک روز واحد:
برای اعمال این محدودیت می‌توان از قید زیر برای دروس متعلق به مجموعه F' استفاده کرد:

$$\sum_{k_۱=۰}^{۴} (X_{c_i,۵k_۲+k_۱} + X_{c_j,۵k_۲+k_۱}) \leqslant ۱, \ \ \forall (c_i, c_j) \in F', \ \ k_۲ \in \{۰, \ldots, ۴\}.$$

۹. محدودیت متوالی بودن دروس مجموعه G در برنامه:

برای اعمال این محدودیت از فرمول زیر استفاده می‌کنیم:

$$X_{c_i,۵k_۲+k_۱} - X_{c_j,۵k_۲+k_۱+۱} = ۰, \ \ \forall (c_i, c_j) \in G,$$
$$k_۱ \in \{۰, \ldots, ۴\}, \ \ k_۲ \in \{۰, \ldots, ۴\}.$$

۱۰. محدودیت عدم توالی دروس مجموعه G' در برنامه:
در مقابل قید دهم بعضی از کلاس‌ها نباید به‌صورت متوالی در برنامه زمانی قرار بگیرند. برای این رشته از دروس که در مجموعه G' گردآوری‌شده‌اند، از قید زیر استفاده می‌کنیم:

$$X_{c_i,۵k_۲+k_۱} + X_{c_j,۵k_۲+k_۱+۱} \leqslant ۰, \ \ \forall (c_i, c_j) \in G',$$
$$k_۱ \in \{۰, \ldots, ۴\}, \ \ k_۲ \in \{۰, \ldots, ۴\}.$$

۱۱. برای برنامه‌ریزی بعضی از دروس الزام قرارگیری یک درس در صبح و دیگری در عصر همان روز وجود دارد. این دسته از دروس در مجموعه H قرار دارند و با قید زیر می‌توان آن را اعمال کرد:

$$\sum_{k_۱=۰}^{۱} X_{c_i,۵k_۲+k_۱} = \sum_{k_۱=۲}^{۴} X_{c_j,۵k_۲+k_۱}, \quad \forall (c_i, c_j) \in H, \ \ k_۲ \in \{۰, \ldots, ۴\}.$$

۱۲. بعضی از دروس نیاز به برنامه‌ریزی در یک ساعت یکسان ولی در دو روز متفاوت دارند. این مجموعه دروس را قبلاً توسط مجموعه Par مشخص کرده‌ایم. برای اعمال این قید به‌صورت زیر عمل می‌کنیم:

$$X_{c_i,۵k_۲+k_۱} - \sum_{k=۰,k\neq k_۲}^{۴} X_{c_j,۵k+k_۱} = ۰, \quad \forall (c_i,c_j) \in Par,\ \ k_۱,k_۲ \in \{۰,\ldots,۴\}.$$

۱۳. دروس مجموعه Pre باید به‌صورتی برنامه‌ریزی شوند که یکی پس از دیگری در برنامه قرار بگیرند. برای اعمال این محدودیت به‌صورت زیر عمل می‌کنیم:
مجموعه T_t را به‌صورت $T_t = T - \{۰, ۱, \ldots, t\}$ تعریف می‌کنیم.

$$X_{c_i,t} - \sum_{t\in T_t} X_{c_j,t} \leqslant ۰,\ \ \forall (c_i,c_j) \in Pre,\ \ \forall t \in T.$$

۱۴. دروس مجموعه K دارای این محدودیت در برنامه‌ریزی‌اند که الزام وجود یک روز فاصله بین آن‌ها وجود دارد. بنابراین برای این مجموعه از دروس، قید زیر را به مجموعه قیود مساله اضافه می‌کنیم: فرض کنیم $t' = ۵[\frac{t}{۵}]$ و

$$\bar{T_t} = \{t \in T | t' - ۵ \leqslant t \leqslant t' + ۱۰\},$$

که در آن نماد $[.]$ بیان‌گر جزء صحیح است. با این تعریف، قید مربوطه به‌صورت زیر نوشته می‌شود:

$$X_{c_i,t} + \sum_{t\in \bar{T_t}} X_{c_j,t} \leqslant ۱,\ \ \forall (c_i,c_j) \in K,\ \ \forall t \in T.$$

۱۵. دروس سرویسی باید حتماً در زمان تعیین‌شده توسط گروه سرویس‌دهنده ارائه شوند. برای این منظور می‌توان از قید زیر استفاده کرد:

$$X_{c_i,t_{c_i}} = ۱,\ \ \forall c_i \in C_{ser}.$$

که در آن t_{c_i} زمان تعیین‌شده برای ارائه درس سرویسی c_i است.

۱۶. بعضی از دروس به‌صورت سرویسی ارائه می‌شوند و زمان اختصاص‌یافته به این درس‌ها از طرف گروه ارائه‌دهنده تعیین می‌گردد. بنابراین زمان ارائه این دروس امکان تغییر ندارد. فرض کنیم مجموعه T_s زمان‌های ارائه دروس سرویسی برای گروه دانشجویی $s \in S$ باشد. بنابراین کافی است برای این گروه دانشجویی در این زمان‌ها درسی در برنامه قرار نگیرد تا امکان اخذ درس سرویسی توسط دانشجویان این گروه وجود داشته باشد. محدودیت زیر می‌تواند پاسخگوی این مطلب باشد:

$$\sum_{c\in C_s} X_{c,t} = ۱,\ \ \forall t \in T_s.$$

۱۷. حداقل و حداکثر تعداد واحد درسی که مدرس l در هفته تدریس می‌کند، باید رعایت شود. این مقادیر در ستون‌های $LTmin$ و $LTmax$ برای هر $l \in L$ از ورودی دریافت‌شده است.

$$LTmin_l \leqslant \sum_{t\in T}\sum_{c\in C_l} Cunit\ X_{c,t} \leqslant LTmax_l, \quad ,\forall l \in L.$$

۱۸. به‌منظور بالا رفتن کیفیت تدریس، هر استاد نباید به‌طور متوالی بیش‌تر از ۴ ساعت تدریس در روز داشته باشد. چون این قید فقط برای زمان عصر موضوعیت پیدا می‌کند، می‌توان آن را به‌صورت زیر کنترل کرد:

$$\sum_{c\in C_l} (X_{c,۵i+۲} + X_{c,۵i+۳} + X_{c,۵i+۴}) \leqslant ۲,\ \ i = \{۰, ۱, \ldots, ۴\}.$$

۱۹. محدودیت مربوط به اختصاص کلاس مناسب به درس بر اساس ظرفیت کلاس و تعداد دانشجو در ثبت‌نام مقدماتی:
تعداد کلاس‌ها که شامل کلاس کوچک، کلاس متوسط، کلاس بزرگ، کلاس خیلی بزرگ، کارگاه‌ها و آزمایشگاه‌ها می‌شدند، به‌تفکیک در پارامترهای $N_A, N_K, N_{XL}, N_L, N_M, N_S,$ مشخص‌شده است. بنابراین باید کنترل کرد که تعداد کلاس‌های اختصاص داده‌شده برای هر بازه‌ی زمانی از مقادیر فوق تجاوز نکند. بدین منظور از تابع قدرمطلق و جزء صحیح استفاده‌شده است. به‌راحتی می‌توان کنترل کرد که تابع $f(x,a,b) = [max\{\frac{-۱}{۲(b-a)}|۲x-a-b| + \frac{۳}{۲}, ۰\}]$ در صورتی که مقدار ورودیش، یعنی x، در بازه‌ی $[a,b]$ باشد، مقدار یک و در غیر این صورت مقدارصفر را برمی‌گرداند. از این تابع می‌توان برای تعیین تعداد کلاس‌های مورد نیاز از هر نوع در زمان $t \in T$ استفاده کرد. به‌عنوان مثال تعداد کلاس‌های کوچک مورد نیاز در زمان t با فرمول

$$\sum_{c \in C} f((C.num)_c, ۶, ۲۰)\ X_{c,t}$$

قابل محاسبه است و این عدد نباید از N_S تجاوز کند. بنابراین قید

$$\sum_{c \in C_r} f((C.num)_c, ۶, ۲۰)\ X_{c,t} \leqslant N_{S,t},\ \ \forall t \in T,$$

را به مساله اضافه می‌کنیم. به‌طور مشابه باید قیود زیر به مساله اضافه شوند:

$$\sum_{c \in C_r} f((C.num)_c, ۲۱, ۳۵)\ X_{c,t} \leqslant N_{M,t},\ \ \forall t \in T,$$
$$\sum_{c \in C_r} f((C.num)_c, ۳۶, ۵۰)\ X_{c,t} \leqslant N_{L,t},\ \ \forall t \in T,$$
$$\sum_{c \in C_r} f((C.num)_c, ۵۱, ۶۵)\ X_{c,t} \leqslant N_{XL,t},\ \ \forall t \in T,$$
$$\sum_{c \in C_r} f((C.num)_c, ۱, n_K)\ X_{c,t} \leqslant N_{K,t},\ \ \forall t \in T,$$
$$\sum_{c \in C_r} f((C.num)_c, ۱, n_A)\ X_{c,t} \leqslant N_{A,t},\ \ \forall t \in T.$$

۲۰. محدودیت قرارگیری برنامه‌ی اساتید در ۴ روز هفته:
به‌دلیل وجود شرط ۴۰ ساعت حضور فیزیکی استاد در دانشگاه، معمولاً معاونت آموزشی اصرار دارد که برنامه‌ی اساتید در ۴ روز هفته ریخته شود. این محدودیت باید برای اعضای هیات علمی لحاظ شود ولی برای اساتید حق‌التدریس چنین شرطی نداریم.
برای هر استاد $l \in L$ مجموعه $LDay$ را مجموعه‌ی روزهایی که استاد l در دسترس است، تعریف می‌کنیم. این ستون می‌تواند توسط مقادیر ستون Lj مقداردهی شود. در حقیقت کافی است جمع مقادیر مربوط به هر سطر Lj را محاسبه کنیم و در صورت غیرصفر بودن، آن روز را برای استاد $l \in L$ جزء روزهای در دسترس قرار دهیم. هم‌چنین می‌توان مجموعه اساتیدی را که حق‌التدریس نیستند، به‌کمک ستون LGE مشخص کرد. فرض کنیم این مجموعه را $LNON$ بنامیم. حال می‌توان محدودیت چهار روز حضور فیزیکی را به‌صورت زیر به مساله اضافه کرد:

$$\sum_{c \in C_l} \sum_{j=۰}^{۴} X_{c,۵i+j} \geqslant ۱,\ \ \forall l \in LNON,\ \ \forall i \in LDay.$$

۲۱. تعداد دروسی که برای یک گروه دانشجویی خاص ارائه می‌شوند از حداکثری که مدیر گروه در مجموعه NCS_{max} تعیین کرده است تجاوز نکند. این شرط به‌کمک قید زیر قابل اعمال خواهد بود:

$$\sum_{c \in C_s} \sum_{j=۰}^{۴} X_{c,۵i+j} \leqslant (NCS_{max})_s,\ \ \forall s \in S,\ \ \forall i = \{۰, \ldots, ۴\}.$$

توابع هدف برای این مدل برابر است با:

۱. بیشینه کردن میزان رضایت اساتید از برنامه‌ی هفتگی:
هر یک از اساتید در ستون Lj جدولی از میزان اولویت زمانی تدریس را مشخص می‌کند. این ستون با مقادیری بین ۰ تا ۵ برای هر بازه‌ی زمانی، بسته‌به نظر استاد تکمیل می‌شود. حال کافی است ضرب این ستون را در دروسی که برای مدرس $l \in L$ اختصاص‌یافته، محاسبه و جمع کل آن را حداکثر کنیم. بنابراین برای این هدف می‌توان از تابع زیر استفاده کرد:

$$\max \sum_{l \in L} \sum_{c \in C_l} \sum_{t \in T} (LGdegree)_l \, (Lj)_t \, X_{c,t}.$$

۲. بیشینه کردن رعایت اولویت زمان ارائه دروس نظری:
بهتر است دروس نظری در زمان صبح زمان‌بندی شوند. دلیل این امر این است که بازدهی یادگیری دروس نظری در صبح بیش‌تر از عصر است.

$$\max \sum_{c \in C_t} \sum_{t \in T_{morn}} (Cdegree)_c \, X_{c,t},$$

$$T_{morn} = \{t \in T | t - ۵[\frac{t}{۵}] \in \{۰, ۱\}\}.$$

۳. بیشینه کردن رعایت اولویت زمان ارائه دروس عملی:
بهتر است دروس عملی در زمان عصر زمان‌بندی شوند. دلیل این امر این است که بازدهی یادگیری دروس عملی در عصر بیش‌تر از صبح است.

$$\max \sum_{c \in C_p} \sum_{t \in T_{even}} (Cdegree)_c \, X_{c,t},$$

$$T_{even} = \{t \in T | t - ۵[\frac{t}{۵}] \in \{۲, ۳, ۴\}\}.$$

۵ نتایج عددی

مدل ارائه‌شده در بخش قبل را با مثال زیر امتحان می‌کنیم.
در این مساله ۵۶ جلسه برای برنامه‌ریزی در نظر گرفته‌شده است. لازم به ذکر است که دروس چهار واحدی که دارای حل تمرین بوده‌اند به‌صورت سه جلسه مجزا، دروس چهار واحدی بدون حل تمرین به‌صورت دو جلسه و دروس سه واحدی نیز به‌صورت دو جلسه در هفته تعریف‌شده است. در این مثال ده استاد و هشت کلاس (شامل دو کلاس بزرگ، دو کلاس متوسط، دو کلاس کوچک، یک کارگاه و یک آزمایشگاه) برای مساله در نظر گرفته‌شده‌اند. اطلاعات مربوط به دروس، اساتید و کلاس‌ها در جداول ۹، ۱۰، ۱۱ و ۱۲ آورده‌شده است.

جدول ۹: جدول اطلاعات دروس.

$Code$	$Cname$	$Cunit$	$Cteacher$	$Cgroup$	$Cservice$	CTS	CPS	$Cdegree$	$Cnum$	$Ctype$	$CTNON$	$CGTP$	CK	$CPar$	$CPre$
۰۱	ریاضی عمومی ۱ ج ۱	۲	۰۰۱	B_1	Yes	۰	۰۰۰۲	۵	۳۵	M		T			
۰۲	ریاضی عمومی ۱ ج ۲	۲	۰۰۱	B_1	Yes	۱۰	۰۰۰۲	۵	۳۵	M		T			
۰۳	ریاضی عمومی ۱ ج ۲	۱/۳۳	۰۰۱	B_1	Yes	۲۰	۰۰۰۲	۵	۳۵	M		T			
۰۴	مبانی علوم ریاضی ج ۱	۱/۵	۰۰۳	B_1	Yes	۱	۰۰۰۱	۴	۱۹	S		T			
۰۵	مبانی علوم ریاضی ج ۲	۱/۵	۰۰۳	B_1	Yes	۱۱	۰۰۰۱	۴	۱۹	S		T			
۰۶	مبانی برنامه‌نویسی ج ۱	۱/۵	۰۰۴	B_1	No	No	No	۵	۴۰	L	۱۲	T	۰۷	۰۷	
۰۷	مبانی برنامه‌نویسی ج ۲	۱/۵	۰۰۴	B_1	No	No	No	۵	۴۰	L	۱۲	T	۰۶	۰۶	
۰۸	نظریه محاسبات ج ۱	۱/۵	۰۰۵	B_7	No	No	No	۴	۲۰	S	۱۲	T	۰۹	۰۹	
۰۹	نظریه محاسبات ج ۲	۱/۵	۰۰۵	B_7	No	No	No	۴	۲۰	S	۱۲	T	۰۸	۰۸	
۱۰	ساختمان داده ج ۱	۲	۰۰۴	B_5	No	No	No	۴	۳۰	M	۱۲	T	۱۱, ۱۲	۱۱, ۱۲	
۱۱	ساختمان داده ج ۲	۲	۰۰۴	B_5	No	No	No	۴	۳۰	M	۱۲	T	۱۰, ۱۲	۱۰, ۱۲	
۱۲	ساختمان داده ج ۳	۱/۳۳	۰۰۴	B_5	No	No	No	۴	۳۰	M	۱۲	T	۱۰, ۱۱	۱۰, ۱۱	
۱۳	برنامه‌نویسی پیشرفته ج ۱	۱/۵	۰۰۸	B_7	No	No	No	۴	۳۰	M	۱۲	T	۱۴	۱۴	
۱۴	برنامه‌نویسی پیشرفته ج ۲	۱/۵	۰۰۸	B_7	No	No	No	۴	۳۰	M	۱۲	T	۱۳	۱۳	
۱۵	اصول سیستم‌های عامل ج ۱	۲	۰۱۰	B_5	No	No	No	۴	۴۴	L	۱۲	T	۱۶	۱۶	
۱۶	اصول سیستم‌های عامل ج ۲	۲	۰۱۰	B_5	No	No	No	۴	۴۴	L	۱۲	T	۱۵	۱۵	
۱۷	مبانی نظریه محاسبه ج ۱	۱/۵	۰۰۵	B_3	No	No	No	۵	۲۸	M	۱۲	T	۱۸	۱۸	
۱۸	مبانی نظریه محاسبه ج ۲	۱/۵	۰۰۵	B_3	No	No	No	۵	۲۸	M	۱۲	T	۱۷	۱۷	
۱۹	مبانی ترکیبات ج ۱	۲	۰۰۱	B_1	No	No	No	۵	۳۷	L	۱۲	T	۲۰, ۲۱	۲۰, ۲۱	
۲۰	مبانی ترکیبات ج ۲	۲	۰۰۱	B_1	No	No	No	۵	۳۷	L	۱۲	T	۱۹, ۲۱	۱۹, ۲۱	
۲۱	مبانی ترکیبات ج ۳	۱/۳۳	۰۰۱	B_1	No	No	No	۵	۳۷	L	۱۲	T	۱۹, ۲۰	۱۹, ۲۰	
۲۲	مبانی منطق ج ۱	۱/۵	۰۰۹	B_3	No	No	No	۴	۲۰	S	۱۲	T	۲۳	۲۳	
۲۳	مبانی منطق ج ۲	۱/۵	۰۰۹	B_3	No	No	No	۴	۲۰	S	۱۲	T	۲۲	۲۲	
۲۴	طراحی الگوریتم ج ۱	۱/۵	۰۰۶	B_5	No	No	No	۳	۲۵	M	۱۲	T	۲۵	۲۵	
۲۵	طراحی الگوریتم ج ۲	۱/۵	۰۰۶	B_5	No	No	No	۳	۲۵	M	۱۲	T	۲۴	۲۴	
۲۶	هوش مصنوعی ج ۱	۱/۵	۰۰۷	B_7	No	No	No	۴	۱۴	S	۲۱	T	۲۷	۲۷	
۲۷	هوش مصنوعی ج ۲	۱/۵	۰۰۷	B_7	No	No	No	۴	۱۴	S	۱۲	T	۲۶	۲۶	
۲۸	تربیت بدنی ۱	۱	۰۱۱	B_1	Yes	۳	No	۱	۳۵	K		P			
۲۹	اندیشه اسلامی ۱	۲	۰۱۲	B_1	Yes	۲	No	۲	۵۰	L		T			
۳۰	اندیشه اسلامی ۲	۲	۰۱۳	B_3	Yes	۷	No	۲	۴۵	L		T			
۳۱	پایگاه داده ج ۱	۱/۵	۰۰۳	B_7	No	No	No	۴	۳۲	M	۱۲	T	۳۲	۳۲	
۳۲	پایگاه داده ج ۲	۱/۵	۰۰۳	B_7	No	No	No	۴	۳۲	M	۱۲	T	۳۱	۳۱	
۳۳	پردازش تصویر ج ۱	۳	۰۰۷	M_3	No	No	No	۵	۱۱	S	۶, ۱۲	T	۳۴	۳۴	
۳۴	پردازش تصویر ج ۲	۳	۰۰۷	M_3	No	No	No	۵	۱۱	S	۶, ۱۲	T	۳۳	۳۳	
۳۵	هوش مصنوعی پیشرفته ج ۱	۳	۰۰۶	M_1	No	No	No	۵	۸	S	۶, ۱۲	T	۳۶	۳۶	
۳۶	هوش مصنوعی پیشرفته ج ۲	۳	۰۰۶	M_1	No	No	No	۵	۸	S	۶, ۱۲	T	۳۵	۳۵	
۳۷	نظریه زبان طبیعی ج۱	۳	۰۰۳	M_1	No	No	No	۵	۱۰	S	۶, ۱۲	T	۳۸	۳۸	
۳۸	نظریه زبان طبیعی ج۲	۳	۰۰۳	M_1	No	No	No	۵	۱۰	S	۶, ۱۲	T	۳۷	۳۷	
۳۹	مبانی اقتصاد	۲	۰۱۴	B_5	Yes	۲۲	No	۲	۳۳	M		T			
۴۰	ریاضی عمومی ۲ ج ۱	۲	۰۰۲	B_3	Yes	۰	۰۰۰۴	۵	۳۷	L		T			
۴۱	ریاضی عمومی ۲ ج ۲	۲	۰۰۲	B_3	Yes	۱۰	۰۰۰۴	۵	۳۷	L		T			
۴۲	ریاضی عمومی ۲ ج ۳	۱/۳۳	۰۰۲	B_3	Yes	۲۰	۰۰۰۴	۵	۳۷	L		T			
۴۳	مبانی احتمال ج ۱	۱/۵	۰۱۵	B_1	Yes	۵	۰۰۰۲	۳	۳۵	M		T			
۴۴	مبانی احتمال ج ۲	۱/۵	۰۱۵	B_1	Yes	۱۵	۰۰۰۲	۳	۳۵	M		T			
۴۵	تفسیر قرآن	۲	۰۱۲	B_5	Yes	۲	No	۲	۴۰	L		T			
۴۶	کارگاه کامپیوتر	۲	۰۰۵	B_3	No	No	No	۳	۴۰	K	۱۲	P			
۴۷	آزمایشگاه کامپیوتر	۱	۰۰۷	B_5	No	No	No	۳	۳۵	A	۱۲	P			
۴۸	مباحثی در علوم کامپیوتر ج ۱	۱/۵	۰۰۸	B_7	No	No	No	۵	۳۱	M	۱۲	T	۴۹	۴۹	
۴۹	مباحثی در علوم کامپیوتر ج ۲	۱/۵	۰۰۸	B_7	No	No	No	۵	۳۱	M	۱۲	T	۴۸	۴۸	
۵۰	کامپایلر ج ۱	۱/۵	۰۰۵	B_3	No	No	No	۳	۳۴	M	۱۲	T	۵۱	۵۱	
۵۱	کامپایلر ج ۲	۱/۵	۰۰۵	B_3	No	No	No	۳	۳۴	M	۱۲	T	۵۰	۵۰	
۵۲	تفسیر نهج البلاغه	۲	۰۱۵	B_5	Yes	۸	No	۱	۴۵	L		T			
۵۳	زبان تخصصی	۲	۰۱۰	B_7	No	No	No	۱	۲۲	M	۱۲	T			
۵۴	دانش خانواده و جمعیت	۲	۰۱۶	B_5	Yes	۱۸	۰۰۰۲	۱		M		T			
۵۵	منطق ج ۱	۳	۰۱۰	M_3	No	No	No	۳	۱۰	S	۶, ۱۲	T	۵۶	۵۶	
۵۶	منطق ج ۲	۳	۰۱۰	M_3	No	No	No	۳	۱۰	S	۶, ۱۲	T	۵۵	۵۵	

علاوه بر اطلاعات فوق مدیر گروه تصمیم دارد تا در هر روز از هفته، حداکثر سه درس برای هر ورودی در دوره کارشناسی و حداکثر دو درس برای هر ورودی در دوره کارشناسی ارشد در نظر بگیرد. بنابراین در این مثال برای مولفه‌های متناظر در متغیر NCS_{max}، به‌ترتیب

جدول ۱۰: جدول اطلاعات اساتید.

$Lcode$	$Lname$	Lj					$LTmin$	$LTmax$	LGE	$LGdegree$	$Lmas$	$Lmah$				
۰۰۱	علی ایزدپناه	۰	۲	۳	۵	۴	۶	۱۵	He	۳	Yes	۱	۱	۳	۰	۴
		۱	۱	۰	۱	۳						۴	۰	۳	۰	۳
		۱	۲	۲	۰	۰						۳	۰	۰	۱	۱
		۰	۰	۰	۴	۳						۱	۵	۳	۰	۴
		۰	۰	۳	۲	۵						۰	۰	۰	۰	۰
۰۰۲	محمد رضا خسرویان	۱	۱	۲	۴	۳	۰	۱۸	Ha	۲	No	۱	۲	۲	۳	۲
		۰	۰	۰	۳	۲						۱	۱	۳	۴	۵
		۰	۰	۳	۵	۱						۱	۰	۳	۱	۱
		۰	۰	۰	۰	۰						۰	۰	۰	۰	۰
		۱	۱	۱	۴	۳						۱	۲	۵	۳	۴
۰۰۳	آرش عطاخواه	۰	۰	۰	۵	۴	۱۰	۱۸	He	۳	No	۲	۲	۲	۲	۲
		۰	۱	۱	۴	۱						۱	۱	۲	۳	۴
		۱	۲	۲	۴	۳						۱	۰	۳	۴	۳
		۲	۲	۱	۳	۳						۱	۱	۲	۵	۱
		۰	۰	۰	۲	۲						۰	۰	۰	۵	۴
۰۰۴	سعید قاسمی	۰	۰	۰	۰	۰	۱۰	۱۸	He	۵	No	۲	۲	۲	۰	۵
		۱	۲	۳	۵	۵						۲	۲	۲	۵	۵
		۰	۱	۱	۳	۲						۰	۱	۰	۲	۴
		۰	۰	۲	۵	۵						۲	۲	۴	۲	۴
		۳	۲	۲	۵	۱						۲	۲	۲	۵	۵
۰۰۵	هادی فاضل	۱	۲	۳	۴	۵	۱۰	۱۸	He	۲	No	۱	۱	۲	۱	۵
		۰	۰	۰	۱	۲						۰	۱	۰	۱	۳
		۱	۲	۲	۴	۴						۱	۰	۲	۲	۳
		۲	۴	۳	۵	۳						۱	۱	۵	۳	۳
		۱	۱	۳	۳	۴						۰	۱	۲	۳	۴
۰۰۶	روح الله یوسفی	۵	۲	۰	۱	۲	۱۰	۱۸	He	۱	No	۲	۲	۵	۳	۱
		۲	۳	۳	۵	۲						۲	۲	۱	۲	۴
		۰	۰	۰	۱	۲						۳	۴	۱	۴	۴
		۱	۱	۲	۴	۳						۳	۳	۱	۲	۴
		۵	۵	۴	۳	۲						۱	۲	۱	۲	۵
۰۰۷	ناصر سلطانی	۱	۲	۴	۵	۵	۵	۱۸	He	۵	Yes	۰	۱	۲	۰	۴
		۵	۴	۴	۲	۳						۲	۳	۳	۰	۳
		۱	۴	۴	۲	۳						۴	۵	۰	۰	۲
		۰	۴	۳	۳	۵						۲	۴	۳	۰	۵
		۱	۲	۳	۰	۱						۰	۲	۰	۰	۳
۰۰۸	محمد رضا دورابی	۱	۱	۱	۳	۲	۵	۱۸	He	۴	Yes	۰	۱	۲	۰	۲
		۴	۲	۲	۱	۴						۱	۳	۵	۰	۲
		۵	۳	۱	۵	۲						۰	۳	۰	۰	۵
		۳	۴	۱	۴	۲						۱	۳	۳	۰	۲
		۵	۰	۰	۱	۱						۱	۲	۰	۰	۵
۰۰۹	فاضل فروزان	۳	۴	۴	۳	۳	۰	۱۸	Ha	۴	No	۱	۱	۱	۵	۳
		۵	۰	۰	۰	۰						۳	۳	۱	۵	۳
		۳	۵	۱	۱	۲						۱	۱	۱	۱	۲
		۵	۵	۳	۲	۱						۰	۲	۰	۳	۴
		۵	۲	۲	۰	۰						۲	۴	۵	۱	۱
۰۱۰	فاطمه عبدالهی	۱	۲	۲	۱	۵	۱۰	۱۸	He	۵	No	۲	۲	۱	۰	۵
		۲	۱	۳	۳	۱						۱	۴	۱	۱	۵
		۵	۰	۴	۴	۱						۲	۱	۰	۱	۳
		۵	۴	۳	۵	۰						۲	۰	۱	۰	۳
		۱	۵	۳	۲	۰						۱	۳	۳	۱	۳

مقادیر سه و دو در نظر گرفته می‌شود. کدنویسی این مدل در محیط متلب نسخه $R۲۰۱۹a$ انجام شده است که صحه‌گذاری آن با حل مثال‌های مختلف و متنوع، همراه با مقایسه و بررسی نتایج به‌دست آمده، صورت گرفته است. روش انتخابی برای حل این مساله‌ی چندهدفه، روش مجموع وزنی است. در اینجا وزن‌های انتخابی برای توابع هدف یکسان در نظر گرفته‌شده است. خروجی مدل در جدول ۱۳ آورده‌شده است. مقادیر توابع هدف در جواب بهینه، به‌ترتیب برابر با ۵۷۸، ۶ و ۱۲۹ است. در صورتی که مدیر گروه با این جواب قانع نشد، می‌توان مساله را با ضرایب دیگر حل کرد.

جدول ۱۱: اطلاعات مربوط به مشخصات کلاس‌ها.

$Rcode$	$Rname$	$Rcapacity$	$Rtype$	$Raccess$
۰۰۰۱	۱۰۱	۲۰	کلاس کوچک	۱ ۰ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۰ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱
۰۰۰۲	۱۰۲	۳۵	کلاس متوسط	۰ ۱ ۱ ۱ ۱ ۰ ۱ ۱ ۱ ۱ ۰ ۰ ۱ ۱ ۱ ۰ ۱ ۱ ۰ ۱ ۰ ۱ ۱ ۱ ۱
۰۰۰۳	۱۰۳	۵۰	کلاس بزرگ	۰ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۰ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۰ ۱ ۱ ۱ ۱
۰۰۰۴	۱۰۴	۵۰	کارگاه	۱ ۱
۰۰۰۵	۱۰۵	۵۰	آزمایشگاه	۱ ۱
۰۰۰۶	۱۰۶	۲۰	کلاس کوچک	۱ ۱
۰۰۰۷	۱۰۷	۳۵	کلاس متوسط	۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۱ ۰ ۱ ۱ ۱ ۱ ۱ ۱
۰۰۰۸	۱۰۸	۵۰	کلاس بزرگ	۱ ۱

از آن‌جا که اولویت اغلب اساتید برای تدریس (سنون L_j در جدول ۱۰) در بازه‌های زمانی صبح بوده است، خروجی مدل از حداکثر ظرفیت تشکیل کلاس در دو بازه‌ی زمانی صبح استفاده کرده است. واضح است که رسیدن به چنین جوابی به‌صورت دستی بسیار دشوار خواهد بود. زمان اجرای برنامه برای این مثال در مجموع ۲٫۷۳۱۲ ثانیه بود که این زمان شامل ساخت ماتریس ضرایب، بردار سمت راست محدودیت‌ها، بردار تابع هدف و حل مساله است. تعداد متغیرهای مورد نیاز در مدل‌سازی این مثال تنها $۵۶ \times ۲۵ = ۱۴۰۰$ عدد است که اگر بدون دسته‌بندی محدودیت‌ها مدل‌سازی انجام می‌گرفت، به تعداد $۵۶ \times ۲۵ \times ۱۰ \times ۸ = ۱۱۲۰۰۰$ متغیر افزایش می‌یافت. در اینجا منظور از عدم دسته‌بندی محدودیت‌ها این است که متغیر تصمیم، متغیر صفر و یکی به‌صورت $X_{c,t,l,r}$ باشد که اگر درس c در زمان t توسط استاد l در کلاس r ارائه شود، مقدار یک و در غیر این صورت مقدار صفر را اختیار کند.

جدول ۱۲: اطلاعات مربوط به متغیر N.

شنبه	$S=۲$ $M=۱$ $L=۱$ $XL=۱$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۱$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۱$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$
یک‌شنبه	$S=۲$ $M=۱$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۱$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$
دوشنبه	$S=۲$ $M=۲$ $L=۱$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$
سه‌شنبه	$S=۲$ $M=۱$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۱$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$
چهارشنبه	$S=۲$ $M=۱$ $L=۱$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$	$S=۲$ $M=۲$ $L=۲$ $XL=۰$ $A=۱$ $K=۱$

۶ نتیجه‌گیری

در این مقاله مساله برنامه‌ریزی دروس دانشگاهی با استفاده از برنامه‌ریزی چندهدفه‌ی صفر و یک مورد بررسی قرار گرفته است. محدودیت‌های اعمال‌شده در این مدل‌سازی با توجه به قوانین آموزشی و آیین‌نامه‌های موجود در دانشگاه ولی عصر(عج) رفسنجان در نظر گرفته‌شده است و بنابراین می‌توان گفت که این مساله به‌نوعی با شرایط بومی سازگارشده است. در اکثر پژوهش‌های انجام‌شده بر روی این مساله، در نظر گرفتن متغیرهای با بیش از دو اندیس باعث‌شده است که پیچیدگی محاسباتی مساله بالا رفته و در نتیجه زمان حل مساله به‌صورت نمایی افزایش یابد. تعداد اندیس‌های در نظر گرفته‌شده برای متغیرها در این مقاله دو است که این مطلب باعث کاهش قابل‌ملاحظه‌ی پیچیدگی محاسباتی و در نتیجه زمان حل مساله شده است.

از مزیت‌های دیگر روش اعمال‌شده در این مقاله می‌توان به پرهیز از تقسیم‌بندی محدودیت‌ها به دو گروه محدودیت‌های انعطاف‌ناپذیر و انعطاف‌پذیر اشاره کرد. ایده‌ی انتقال محدودیت‌های انعطاف‌پذیر به مجموعه‌ی توابع هدف در عین نوآوری باعث افزایش کارایی مساله‌شده است. اگرچه در نظر گرفتن دو اندیس برای متغیرها دارای مزیت‌های فراوانی بوده ولی باعث‌شده است که تقسیم دروس بین اساتید در خلال حل مساله انجام نشود. این تقسیم‌بندی قبل از حل مساله در سطح صفر توسط مدیر گروه انجام می‌شود. این موضوع را می‌توان در پژوهش‌های آتی مورد بررسی قرار داد و مدل مساله را به‌گونه‌ای نوشت که اختصاص استاد درس نیز توسط برنامه انجام شود.

تشکر و قدردانی

نویسندگان مقاله از سردبیر محترم نشریه و هم‌چنین داوران گرامی که زحمت بازنگری مقاله را متقبل و با نظرات و پیشنهادهای خود باعث بهبود مطالب مندرج در مقاله شدند، کمال تشکر و قدردانی را به‌عمل می‌آورند.

جدول ۱۳: نتیجه نهایی مدل پیشنهادی.

	۸ – ۱۰	۱۰ – ۱۲	۱۳ : ۳۰ – ۱۵ : ۳۰	۱۵ : ۳۰ – ۱۷ : ۳۰	۱۷ : ۳۰ – ۱۹ : ۳۰
شنبه	ریاضی عمومی ۱ ج ۱ (فروزان) (کارشناسی ترم اول) نظریه محاسبات ج ۱ (فاضل) (کارشناسی ترم هفتم) ساختمان داده ج ۳ (قاسمی) (کارشناسی ترم پنجم) پردازش تصویر ج ۱ (سلطانی) (کارشناسی ارشد ترم سوم) هوش مصنوعی پیشرفته ج ۲ (یوسفی) (کارشناسی ارشد ترم اول ریاضی عمومی ۲ ج ۱ (خسرویان) (کارشناسی ترم سوم)	مبانی برنامه‌نویسی ج۱ (قاسمی) (کارشناسی ترم اول) نظریه زبان طبیعی ج ۱ (عطاخواه) (کارشناسی ارشد ترم اول) مباحثی در علوم کامپیوتر ج ۲ (دورابی) (کارشناسی ترم هفتم) کامپایلر ج ۱ (فاضل) (کارشناسی ترم سوم)	اندیشه اسلامی ۱ (اساتید گروه معارف) (کارشناسی ترم اول) تفسیر قرآن (اساتید گروه معارف) (کارشناسی ترم پنجم)	تربیت بدنی ۱ (اساتید گروه تربیت بدنی) (کارشناسی ترم اول)	اصول سیستم‌های عامل ج ۱ (عبدالهی) (کارشناسی ترم پنجم) مبانی ترکیبیات ج ۳ (ایزدپناه) (کارشناسی ترم اول)
یک‌شنبه	برنامه‌نویسی پیشرفته ج ۱ (دورابی) (کارشناسی ترم هفتم) مبانی منطق ج ۱ (فروزان) (کارشناسی ترم سوم) مبانی احتمال ج ۱ (اساتید گروه آمار) (کارشناسی ترم اول)	مبانی نظریه محاسبه ج ۲ (فاضل) (کارشناسی ترم سوم) طراحی الگوریتم ج ۱ (یوسفی) (کارشناسی ترم پنجم) پایگاه داده ج ۱ (عطاخواه) (کارشناسی ترم هفتم)	اندیشه اسلامی ۲ (اساتید گروه معارف) (کارشناسی ترم سوم)	تفسیر نهج‌البلاغه (اساتید گروه معارف) (کارشناسی ترم پنجم) زبان تخصصی (عبدالهی) (کارشناسی ترم هفتم)	آزمایش‌گاه کامپیوتر (سلطانی) (کارشناسی ترم پنجم)
دوشنبه	ریاضی عمومی ۱ ج ۲ (فروزان) (کارشناسی ترم اول) نظریه محاسبات ج ۲ (فاضل) (کارشناسی ترم هفتم) ساختمان داده ج ۲ (قاسمی) (کارشناسی ترم پنجم) هوش مصنوعی پیشرفته ج ۱ (یوسفی) (کارشناسی ارشد ترم اول) ریاضی عمومی ۲ ج ۲ (خسرویان) (کارشناسی ترم سوم) منطق ج ۱ (عبدالهی) (کارشناسی ارشد ترم سوم)	مبانی علوم ریاضی ج ۱ (عطاخواه) (کارشناسی ترم اول) هوش مصنوعی ج ۱ (خسرویان) (کارشناسی ترم هفتم) کامپایلر ج ۲ (فاضل) (کارشناسی ترم سوم)		مبانی ترکیبیات ج ۱ (ایزدپناه) (کارشناسی ترم اول)	
سه‌شنبه	اصول سیستم‌های عامل ج ۲ (عبدالهی) (کارشناسی ترم پنجم) مبانی منطق ج ۲ (فروزان) (کارشناسی ترم سوم) پردازش تصویر ج ۲ (سلطانی) (کارشناسی ارشد ترم سوم) نظریه زبان طبیعی ج ۲ (عطاخواه) (کارشناسی ارشد ترم اول) مبانی احتمال ج ۲ (اساتید گروه آمار) (کارشناسی ترم اول) مباحثی در علوم کامپیوتر ج ۱ (دورابی) (کارشناسی ترم هفتم)	مبانی برنامه‌نویسی ج ۲ (قاسمی) (کارشناسی ترم اول) مبانی نظریه محاسبه ج ۱ (فاضل) (کارشناسی ترم سوم) طراحی الگوریتم ج ۲ (یوسفی) (کارشناسی ترم پنجم) پایگاه داده ج ۲ (عطاخواه) (کارشناسی ترم هفتم)		کارگاه کامپیوتر (فاضل) (کارشناسی ترم سوم) دانش خانواده و جمعیت (اساتید گروه معارف) (کارشناسی ترم پنجم)	
چهارشنبه	ریاضی عمومی ۱ ج ۳ (فروزان) (کارشناسی ترم اول) برنامه‌نویسی پیشرفته ج ۲ (دورابی) (کارشناسی ترم هفتم) ریاضی عمومی ۲ ج ۳ (خسرویان) (کارشناسی ترم سوم)	مبانی علوم ریاضی ج ۲ (عطاخواه) (کارشناسی ترم اول) ساختمان داده ج ۱ (قاسمی) (کارشناسی ترم پنجم) هوش مصنوعی ج ۲ (خسرویان) (کارشناسی ترم هفتم) منطق ج ۲ (عبدالهی) (کارشناسی ارشد ترم سوم)	مبانی ترکیبیات ج ۲ (ایزدپناه) (کارشناسی ترم اول) مبانی اقتصاد (اساتید گروه اقتصاد) (کارشناسی ترم پنجم)		

فهرست منابع

[1] Abuhamdah, A., Ayob, M., Kendall, G., and Sabar, N. R., *Population based local search for university course timetabling problems*, Appl. Intell., **40** (2014), 44-53.

[2] Akkan, C. and Gülcü, A., *A bi-criteria hybrid genetic algorithm with robustness objective for the course timetabling problem*, Comput. Oper. Res., **90** (2018), 22-32.

[3] Assi, M., Halawi, B., and Haraty, R. A., *Genetic algorithm analysis using the graph coloring method for solving the university timetable problem*, Procedia. Comput. Sci., **126** (2018), 899-906.

[4] Aziz, N. L. A. and Aizam, N. A. H., *A survey on the requirements of university course timetabling*, World Acad. Sci. Eng. Technol. Int. J. Math. Comput. Phys. Electr. Comput. Eng., **10** (2016), 236-241.

[5] Bagger, N. C. F., Desaulniers, G., and Desrosiers, J., *Daily course pattern formulation and valid inequalities for the curriculum-based course timetabling problem*, J. Sched., **22** (2019), 155-172.

[6] Budiono, T. A. and Wong, K. W., *A pure graph coloring constructive heuristic in timetabling*, in Proc. Int. Conf. Comput. Inf. Sci. (ICCIS), (2012), 307-312.

[7] Burke, E. K. and Petrovic, S., *Recent research directions in automated timetabling*, Eur. J. Oper. Res., **140** (2002), 266-280.

[8] Burke, E. K., McCollum, B., Meisels, A., Petrovic, S., and Qu, R., *A graph-based hyper-heuristic for educational timetabling problems*, Eur. J. Oper. Res., **176** (2007), 177-192.

[9] Chen, R. M. and Shih, H. F., *Solving university course timetabling problems using constriction particle swarm optimization with local search*, Algorithms, **6** (2013), 227-244.

[10] Chen, M. C., Sze, S. N., Goh, S. L., Sabar, N. R. and Kendall, G., *A survey of university course timetabling problem: perspectives, trends and opportunities*, IEEE. Access., **9** (2021), 106515-106529.

[11] Feng, X., Lee, Y., and Moon, I., *An integer program and ahybrid genetic algorithm for the university timetabling problem*, Optim. Methods. Softw., **32** (2017), 625-649.

[12] Goh, S. L., Kendall, G., and Sabar, N. R., *Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem*, J. Oper. Res. Soc., **70** (2019), 873-888.

[13] Goh, S. L., Kendall, G., Sabar, N. R., and Abdullah, S., *An effective hybrid local search approach for the post enrolment course timetabling problem*, Opsearch., **57** (2020), 1131-1163.

[14] Gotlieb, C., *The construction of class-teacher timetables*, in Proc. IFIP Congress, **62** (1963), 73-77.

[15] Gunawan, A., Ng, K. M., and Poh, K. L., *A hybridized lagrangian relaxation and simulated annealing method for the course timetabling problem*, Comput. Oper. Res., **39** (2012), 3074-3088.

[16] Habashi, S. S., Salama, C., Yousef, A. H., and Fahmy, H. M., *Adaptive diversifying hyper-heuristic based approach for timetabling problems*, in Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON), (2018), 259-266.

[17] Lindahl, M., Mason, A. J., Stidsen, T., and Sørensen, M., *A strategic view of university timetabling*, Eur. J. Oper. Res., **266** (2018), 35-45.

[18] Nagata,Y., *Random partial neighborhood search for the post-enrollment course timetabling problem*, Comput. Oper. Res., **90** (2018), 84-96.

[19] Nothegger, C., Mayer, A., Chwatal, A., and Raidl, G. R., *Solving the post enrolment course timetabling problem by ant colony optimization*, Ann. Oper. Res., **194** (2012), 325-339.

[20] Oktavia, M., Aman, A., and Bakhtiar, T., *Courses timetabling problem by minimizing the number of less preferable time slots*, IOP Conf. Ser.: Mater. Sci. Eng. **166** (2017), 012025.

[21] Sabar, N. R., Ayob, M., Kendall, G., and Qu, R., *A honey-bee mating optimization algorithm for educational timetabling problems*, Eur. J. Oper. Res., **216** (2012), 533-543.

[22] Song, T., Liu, S., Tang, X., Peng, X., and Chen, M., *An iterated local search algorithm for the university course timetabling problem*, Appl. Soft. Comput., **68** (2018), 597-608.

[23] Teoh, C. K., Wibowo, A., and Ngadiman, M. S., *Review of state of the art for metaheuristic techniques in academic scheduling problems*, Artif. Intell. Rev., **44** (2015), 1-21.

[24] Turabieh, H., Abdullah, S., McCollum, B., and McMullan, P., *Fish swarm intelligent algorithm for the course timetabling problem*, in Rough Set and Knowledge Technology, (2010), 588-595.

[25] Wang, B., Geng, Y., and Zhang, Z., *Applying genetic algorithm to university classroom arrangement problem*, J. Phys. Conf. Ser., **1325** (2019), 012157.

A Multiobjective Programming Model for University Course Timetabling Problem

Alireza Hosseini Dehmiry[1, 3] **, Mohammad Aref Samadi**[2]

[1,2] Department of Mathematics, Faculty of Mathematic Science, Vali-e-Asr University of Rafsanjan, Iran

Communicated by: Sohrab Effati

Received: 2021/12/14 Accepted: 2022/3/12

Abstract: The university timetabling problem is one of principal management problem in the beginning of each semester in the colleges. There are many limitations related to classes, teachers and students that increase computational complexity and put the problem in the field of "NP-Hard" class. In this study we tried to solve this problem in terms of zero-one multiobjective programming and devise a mathematical model of the problem using a minimum number of variables. The proposed approach is based on activity in order to reduce the number of variables to modeling the problem in an acceptable level.

Keywords: Multiobjective programming problem, Efficient points, University course timetabling problem, zero-one programming.

©2022 Shahid Chamran University of Ahvaz, Ahvaz, Iran. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonComertial 4.0 International (CC BY-NC 4.0 license) (http://creativecommons.org/licenses/by-nc/4.0/).

[3]Corresponding author.
E-mail addresses: dehmiry@vru.ac.ir (A. H. Dehmiry), samadi.ma@vru.ac.ir (M. A. Samadi).