مهندسی صنایع و مدیریت شریف، (تابستان ۱۴۰۲)
دوره‌ی ۳۹، شماره‌ی ۱، صص. ۱۵۵-۱۶۷، (پژوهشی)

Original Article

زمان‌بندی خودکار دروس دانشگاهی با استفاده از رویکرد ابرابتکاری

مریم فتوتی (کارشناس‌ارشد)
سیدهادی میرقادری* (استادیار)
مسلم علی‌محمدلو (دانشیار)
دانشکده‌ی اقتصاد، مدیریت و علوم اجتماعی، دانشگاه شیراز

این مقاله به مدل‌سازی و حل مسئله‌ی جدول‌بندی زمانی دروس دانشگاهی می‌پردازد. در مدل‌سازی ریاضی این مسئله، دو قسمتی بودن برخی دروس ۳ و ۴ واحدی در طول هفته و همچنین تعلق برخی کلاس‌ها به برخی گروه‌های آموزشی در نظر گرفته شد. برای حل این مسئله، یک الگوریتم ابرابتکاری براساس الگوریتم رقابت استعماری توسعه داده شد که شامل ۹ ابتکاری سطح پایین است و در آن با پنج استراتژی نحوه‌ی تخصیص مشخص می‌شود. الگوریتم ابرابتکاریِ پیشنهادی با داده‌های واقعی از دانشگاه شیراز آزمون شد. نتایج نشان داد که این الگوریتم قادر به تولید ۱۰ جدول زمانی متفاوت طی ۱۷ ساعت اجرا و بدون دخالت انسان است. بهترین جدول زمانی تولید شده به وسیله‌ی ابرابتکاری توسعه داده شده قادر است حدود ۱۱ درصد بهره‌برداری از کلاس را افزایش داده و زمان انتظار دانشجویان برای شروع کلاس بعدی را به طور متوسط حدود ۱ ساعت در هفته کاهش داده است.

واژگان کلیدی: درس، جدول زمانی، ابرابتکاری، فرابتکاری، الگوریتم.

mrymfotovvati@gmail.com
mirghaderi@shirazu.ac.ir
mslmaml@shirazu.ac.ir

۱. مقدمه

آموزش یکی از فرایندهای اصلی هر دانشگاه است و نیازمند زمان‌بندی مناسب برای انجام آن است. ایجاد یک جدول زمانی که تمامی قوانین و نیازهای عملیاتی و همچنین تمایلات و الزامات اساتید و دانشجویان یک دانشگاه را برآورده سازد، در ادبیات علمی به «مسئله‌ی جدول‌بندی زمانی دروس دانشگاهی[۱]» شهرت یافته است و از وظایف مهم و بی‌نهایت دشوار کارشناسان آموزش است.[۱] این مسئله را می‌توان به صورت زمان‌بندیِ تخصیص رویدادهای تدریس در زمان و اتاق معین با ملاحظه‌ی محدودیت‌های ذی‌نفعان دانشگاه ـ نظیر دانشجویان، استادان و گروه‌های آموزشی ـ تعریف کرد. محدودیت‌ها می‌توانند سخت (ملزم به رعایت) یا نرم (رعایت بدون الزام ولی ترجیحی) باشند.[۲] با توجه به پیچیدگی این مسئله، آن را جزء مسئله‌های NP-hard دسته‌بندی کرده‌اند[۳] و از آن‌جا که دانشگاه‌های متفاوت، الزامات و قوانین متفاوتی دارند، در طول سال‌ها الگوریتم‌های متفاوتی برای گونه‌های مختلف این مسئله توسعه یافته است.[۴] به عنوان مثال، حسینی ده‌میری و صمدی، با توجه به مقررات آموزشی و آیین‌نامه‌های دانشگاه ولی عصر (عج) رفسنجان اقدام به طراحی مدل برنامه‌ریزی چندهدفه‌ی صفر و ۱ برای مسئله‌ی برنامه‌ریزی دروس دانشگاهی کردند.[۵] علیرضایی و همکاران نیز مدل برنامه‌ریزی عدد صحیح خطی دو مرحله‌یی برای حل مسئله‌ی برنامه‌ریزی دروس دانشکده ریاضی دانشگاه علم و صنعت توسعه دادند که بهره‌وری استفاده از کلاس‌ها را نسبت به حالت برنامه‌ریزی دستی افزایش و انحراف از اهداف را کاهش می‌داد.[۶] بیگی و بصیری در مدل‌سازی ریاضی مسئله، زمان‌بندی دوهفته‌یی را لحاظ کردند که در آن دروس سه واحدی به صورت هر هفته جلسه‌ی ثابت، یا یک هفته در میان، جلسه‌ی متغیر دارند. آنها یک مدل برنامه‌ریزی چندهدفه طراحی کردند.[۷]

در اغلب دانشگاه‌ها برنامه‌ی فعلی، تنها تکرار برنامه‌ی سال‌های قبل با اندکی تغییرات است[۱] اما با افزایش تغییرات محیطی و برنامه‌های درسی، از جمله بین‌المللی شدن برنامه‌های درسی،[۸] این امر دیگر اجرایی نیست و باید به تکنولوژی و روش‌های جدید متوسل شد.[۱]

چن و همکاران روش‌های حل مسئله‌ی جدول‌بندی زمانی دروس دانشگاهی

* نویسنده مسئول
تاریخ: دریافت ۱۴۰۰/۱۲/۱۵، اصلاحیه ۱۴۰۱/۳/۱۶، پذیرش ۱۴۰۱/۳/۲۴.
DOI:10.24200/J65.2022.57866.2212

استناد به این مقاله:
فتوتی، مریم، میرقادری، سید هادی، و علی‌محمدلو، مسلم، ۱۴۰۲. زمان‌بندی خودکار دروس دانشگاهی با استفاده از رویکرد ابرابتکاری. مهندسی صنایع و مدیریت شریف، ۳۹(۱)، صص. ۱۵۵-۱۶۷.

را در شش گروه طبقه‌بندی کرده‌اند: ۱. روش‌های مبتنی بر پژوهش عملیاتی؛ ۲. فرااِبتکاری‌های تک‌جوابی؛ ۳. فرااِبتکاری‌های مبتنی بر جمعیت؛ ۴. اِبرابتکاری‌ها؛ ۵. رهیافت‌های چندهدفه یا چندمعیاره؛ ۶. رهیافت‌های ترکیبی.[۹]

برای حل مسائل زمان‌بندی می‌توان از الگوریتم‌های دقیق یا تقریبی استفاده کرد. استفاده از الگوریتم‌های دقیق برای حل این‌گونه مسائل در اندازه‌های واقعی امکان‌پذیر نیست؛ زیرا زمان اجرای این دسته از الگوریتم‌ها با رشد اندازه مسئله به صورت نمایی افزایش می‌یابد.[۳] بنابراین از الگوریتم‌های تقریبی مانند «الگوریتم‌های ابتکاری و فرااِبتکاری[۲]» استفاده می‌شود. الگوریتم‌های ابتکاری معروف به الگوریتم‌های جست‌وجو، روش‌های غیرقطعی هستند که می‌توانند جوابی از فضای جست‌وجو را در مدت زمان معقول پیدا کنند؛ جواب پیدا شده شدنی است ولی الزاماً بهینه نیست.[۱۰] الگوریتم‌های فرااِبتکاری نیز با کاوش فضای راه‌حل سعی در پیدا کردن جواب بهینه دارند. مشهورترین نوع این الگوریتم‌ها، الگوریتم ژنتیک است که در تدوین جدول زمانی دروس نیز کاربرد دارد.[۱۱] از طرفی این الگوریتم‌ها جست‌وجوی محلی مؤثری در اطراف جواب بهینه انجام نمی‌دهند که این خود منجر به احتمال از دست دادن جواب‌های خوب می‌شود.[۱۲] از طرفی جست‌وجوی محلی، وابسته به ناحیه‌ی جست‌وجو و در نتیجه مسئله است، یافتن جواب بهینه نیازمند رهیافتی است که این وابستگی را کاهش دهد که این خود منجر به ایجاد رهیافت جدید اَبَرابتکاری[۳] شده است.[۱۳] در واقع رهیافت ابرابتکاری شامل چند جست‌وجوی محلی است که هرکدام با توجه به ناحیه‌ی جست‌وجو اعمال می‌شود و در نتیجه انعطاف‌پذیری روش حل را به طور قابل توجهی افزایش می‌دهد.[۱۴] بنابراین برای رفع این مشکل در این مقاله از یک روش ابرابتکاری به منظور ایجاد جدول زمانی مناسب برای دروس دانشکده‌ی مربوطه استفاده شده است.

در روش ابرابتکاری، هدف بهره‌برداری از نقاط قوت ابتکاری‌ها و فرااِبتکاری‌ها به منظور پوشش ضعف‌های احتمالی آنهاست تا بهترین جواب‌ها نسبت به عملکرد انفرادی هرکدام به دست آید. البته باید در نظر داشت که ابرابتکاری‌ها، مستقیماً روی فضای جواب کار نمی‌کنند در حالی که ابتکاری‌ها و فرااِبتکاری‌ها چنین عملکردی دارند.[۱۵]

ابرابتکاری‌ها از طریق کارکردن در فضای ابتکاری به جای فضای جواب باعث ایجاد جواب‌های بهتر می‌شوند. ساختار هر ابرابتکاری شامل تعدادی ابتکاری سطح پایین و یک یا دو ابتکاری سطح بالاست. ابتکاری‌های سطح پایین به تولید یا بهبود جواب اولیه و در هر مرحله ابتکاری‌های سطح بالا به انتخاب بهترین ابتکاری‌های سطح پایین برای تولید یا بهبود جواب توسط آنها می‌پردازند. اغلب از ابتکاری‌های رنگ‌آمیزی گراف برای تولید جواب و ابتکاری‌های انتقال، تعویض و غیره برای بهبود جواب در سطح پایین و از الگوریتم‌های فرااِبتکاری در سطح بالا استفاده می‌شود.[۱۶]

کو و بورک[۱۷] الگوریتم‌های مختلف سطح بالایی را برای جست‌وجوی فضای ابتکاری در مسئله‌ی جدول‌بندی زمانی دروس استفاده کردند. این ساختارها توسط یازده مسئله‌ی الگو آزموده شد. الگوریتم جست‌وجوی همسایگی متغیر بهترین نتیجه در بین الگوریتم‌های دیگر ایجاد می‌کرد. بنابراین، با به کارگیری ساختار ابرابتکاری مبتنی بر الگوریتم جست‌وجوی همسایگی متغیر، برای جست‌وجوی فضای ابتکاری در حل مسئله‌ی جدول‌بندی دروس، از ابتکاری‌های رنگ‌آمیزی گراف برای تولید جواب در سطح پایین استفاده کردند. کالندر و همکاران[۱۸] ساختار ابرابتکاری را برای حل مسئله‌ی جدول‌بندی زمانی دروس دانشگاه یدی‌تپه ترکیه به کار گرفتند. در ساختار ابرابتکاری از ترکیب روش گرادیان حریصانه برای انتخاب ابتکاری و الگوریتم شبیه‌سازی تبرید (معیار متروپلیس) برای پذیرش حرکت با احتمالی خاص استفاده شده است. نتایج حاصله حاکی از آن است که روش گرادیان حریصانه از عملکرد بهتری نسبت به روش‌های تصادفی ساده، حریصانه و تابع انتخاب برای انتخاب ابتکاری برخوردار بوده است.

سوریا الکاراز و همکاران[۱۹] ابتدا در سال ۲۰۱۴ یک ابرابتکاری مبتنی بر الگوریتم جست‌وجوی محلی تکراری را برای حل این مسئله توسعه دادند که در آن از یک عملگر تصادفی ساده به عنوان ابتکاری سطح بالا و از پنج ابتکاری سطح پایین برای بهبود جواب استفاده شد. این ابرابتکاری قادر به تولید یک جدول زمانی مناسب برای داده‌های مسابقه‌ی جهانی جدول‌بندی زمانی ۲۰۰۷ بود. سپس آنها در سال ۲۰۱۶ از یک ساختار ابرابتکاری ترکیبی شامل الگوریتم جست‌وجوی محلی تکرارشونده با استراتژی حذف و اضافه به عنوان ابتکاری سطح بالا و ابتکاری‌های تصادفی ساده و الگوریتم تپه‌نوردی به عنوان ابتکاری‌های سطح پایین برای حل این مسئله استفاده کردند. در این ابرابتکاری، بخشی از جواب اولیه‌ی ایجاد شده توسط الگوریتم جست‌وجوی محلی تکرارشونده انتخاب شده و با استراتژی حذف و اضافه تغییر داده می‌شود؛ سپس ابتکاری‌های سطح پایین، این جواب را بهبود می‌دهند. این روش نتایج اثربخش‌تری نسبت به تحقیق قبلی ایجاد می‌کند.[۲۰] آنها همچنین در سال ۲۰۱۷ ساختار ابرابتکاری دیگری را به کار بردند و طی آن، یک‌بار از الگوریتم جست‌وجوی محلی تکرارشونده به عنوان ابتکاری سطح بالا و از یک الگوریتم تصادفی ساده و یک الگوریتم بهبود دهنده به عنوان ابتکاری‌های سطح پایین استفاده کردند و بار دیگر در همان ساختار به جای الگوریتم بهبوددهنده از الگوریتم کاهش همسایگی متغیر بهره بردند. نتایج به دست آمده از هر دو آزمون نشان داد که ساختار دوم نسبت به ساختار اول از اثربخشی بیشتری برخوردار بوده است.[۲۱]

در بسیاری از پژوهش‌های حوزه‌ی جدول‌بندی زمانی دروس دانشگاهی، تنها به شکل استاندارد مسئله پرداخته شده و در تعداد کمی مدل‌سازی موارد واقعی بررسی شده است. با این‌حال براساس بررسی‌های نویسندگان، مسئله‌ی دوجلسه‌یی بودن دروس سه واحدی در طول هفته، تعریف ارجحیت الگوهای زمانی خاص برای برگزاری این دروس و همچنین تقسیم کلاس‌های درسی بین بخش‌های آموزشی در دانشکده و ترجیح برگزاری دروس هر بخش در کلاس‌های متعلق به خود در محدودیت‌های مسائل دیگر وجود ندارد. از طرفی توانمندی الگوریتم فرااِبتکاری رقابت استعماری تاکنون در ساختار هیچ نوع ابرابتکاری مورد استفاده قرار نگرفته است.

در ساختار ابرابتکاری‌ها، برای ابتکاری سطح بالا از الگوریتم‌های فرااِبتکاری استفاده می‌شود که یکی از این الگوریتم‌ها، الگوریتم رقابت استعماری است. این الگوریتم توسط آتش‌پزگرگری و لوکاس در سال ۲۰۰۷ توسعه داده شد[۲۲] که از پدیده‌ی اجتماعی استعمارگری الهام گرفته شده است. الگوریتم رقابت استعماری همچون سایر الگوریتم‌های تکاملی با ایجاد یک جمعیت اولیه از جواب‌ها (کشورها) شروع به کار می‌کند. بدین منظور از بین کشورهای جهان، به طور تصادفی تعدادی از بهترین کشورها به عنوان استعمارگر و مابقی آنها به عنوان مستعمره‌ی کشورهای استعمارگر انتخاب می‌شود. تعداد کشورهای مستعمره‌ی یک استعمارگر براساس قدرت آن تعیین می‌شود، به گونه‌یی که هرچه قدرت استعمارگر بیشتر باشد تعداد مستعمره‌هایش نیز بیشتر می‌شود.

پس از تقسیم کشورهای مستعمره بین استعمارگران، امپراطوری‌ها شکل می‌گیرد و مستعمره‌ها شروع به حرکت به سوی استعمارگر خود می‌کنند. قدرت یک امپراطوری تابعی از دو عامل قدرت استعمارگر و قدرت‌مستعمره‌های آن امپراطوری است. رقابت بزرگی میان امپراطوری‌ها شکل می‌گیرد. هر امپراطوری چنانچه نتواند در این رقابت پیروز شود یا نتواند قدرتش را افزایش دهد (یا دست‌کم از کاهش قدرتش جلوگیری

کند) از صحنه‌ی رقابت حذف می‌شود. این رقابت استعمارگرایانه به تدریج موجب قدرتمندتر شدن امپراطوری‌های قوی و کاهش قدرت امپراطوری‌های ضعیف می‌شود. در نهایت امپراطوری‌های ضعیف قدرت خود را از دست داده و سقوط خواهند کرد. حرکت مستعمره‌ها به سمت امپراطوری‌ها در طول رقابت و سازوکار سقوط امپراطوری‌های ضعیف منجر به این می‌شود که تمامی مستعمره‌ها در نهایت به سمت یک استعمارگر واحد همگرا شوند و کلیه امپراطوری‌ها از بین رفته و تنها یک امپراطوری پابرجا می‌ماند. در حالتی که الگوریتم رقابت استعماری به عنوان یک الگوریتم فرابتکاری در نظر گرفته شود هر کشور نمایانگر یک جدول زمانی است اما در این تحقیق در الگوریتم ابرابتکاری رقابت استعماری هر کشور نمایانگر لیستی از ابتکاری‌های سطح پایین است.

هدف پژوهش حاضر، از یک سو مدل‌سازی کلیه‌ی محدودیت‌های مربوط به دروس جهت تهیه جدول زمانی دروس از جمله دوقسمتی بودن دروس سه واحدی و چهارواحدی و نیز محدودیت نرم تخصیص اولیه‌ی کلاس‌ها به بخش‌های آموزشی است. از سوی دیگر توسعه یک روش ابرابتکاری مبتنی بر الگوریتم رقابت استعماری برای حل مسئله‌ی جدول‌زمانی دروس دانشگاهی است.

نوآوری این تحقیق نسبت به سایر تحقیق‌ها، استفاده از یک رویکرد ابرابتکاری مبتنی بر الگوریتم رقابت استعماری است که تاکنون از این الگوریتم در یک رویکرد ابرابتکاری استفاده نشده است. بنابراین، سؤالات این تحقیق عبارت‌اند از: «مدل ریاضی جدول‌بندی دروس دانشگاهی مبتنی بر واقعیات موجود چگونه است؟» و همچنین «آیا برای حل این مسئله، عملکرد الگوریتم رقابت استعماری در یک ساختار ابرابتکاری نسبت به حالت برنامه‌ریزی دستی، باعث بهبود می‌شود؟»

۲. تعریف مسئله

مسئله‌ی جدول‌بندی زمانی دروس دانشگاهی یکی از مسائل کلاسیک و مشهور در حوزه‌ی مسائل بهینه‌سازی است[۲۳] و پس از ثبت‌نام دانشجویان مطرح می‌شود. این مسئله شامل تخصیص زمان و مکان به دروس تعریف شده است. هدف نیز عبارت است از اطمینان یافتن از عدم تداخل دروس هر دانشجو.[۲۴]

مثال عددی این تحقیق و محدودیت‌ها و تابع هدف آن مربوط به دروس شش بخش آموزشی دانشکده اقتصاد، مدیریت و علوم اجتماعی دانشگاه شیراز شامل ۳۵۰ درس است. در این پژوهش برای تجزیه و تحلیل داده‌ها از نرم‌افزار اکسل و متلب استفاده شده است. این تحقیق در پنج مرحله به شرح نمودار ۱ انجام شده است.

۱.۲. محدودیت‌های سخت و نرم

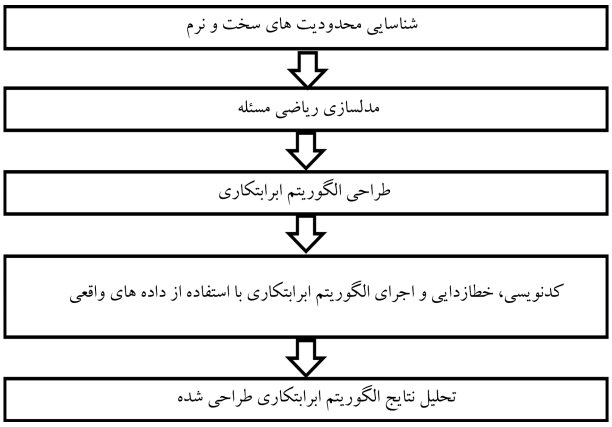
در مسئله‌ی جدول‌بندی زمانی دروس محدودیت‌ها به دو دسته‌ی سخت و نرم تقسیم‌بندی می‌شوند. محدودیت‌های سخت، محدودیت‌هایی هستند که تخطی از آنها مجاز نیست و در صورت تخطی جواب‌های تولید شده (جداول زمانی) موجه نیست. در این مسئله محدودیت‌های سخت عبارت‌اند از:

۱. هر درس به اندازه مدت زمان تعیین شده برای آن اتاق بیفتد؛

۲. هر درس تنها در یک کلاس برگزار شود؛

۳. هر درس براساس الگوی برگزاری آن اتفاق بیفتد؛

۴. هر درس در کلاسی با ظرفیت کافی برای دانشجویان، آن درس را برگزار کند؛



نمودار ۱. مراحل انجام پژوهش.

۵. تداخل زمانی در دروس یک گروه آموزشی وجود نداشته باشد؛

۶. دروس یک استاد، دارای تداخل زمانی نباشد؛

۷. تعداد کلاس‌های برگزار شده در هر اسلات از تعداد کلاس‌های در دسترس بیشتر نباشد؛

۸. هر درس متناسب با تعداد واحد آن در هفته برگزار شود؛

۹. هر کلاس حداکثر به یک درس در یک بازه زمانی معین تخصیص یابد.

محدودیت‌های نرم از اهمیت کم‌تری نسبت به محدودیت‌های سخت برخوردارند به طوری که تخطی از تعدادی از این محدودیت‌ها گاهی اجتناب‌ناپذیر است. هر جواب موجه برای جدول زمانی، توسط یک تابع جریمه ارزیابی می‌شود. این تابع میزان تخطی جدول زمانی از محدودیت‌های نرم را محاسبه می‌کند. این محدودیت‌ها باعث می‌شوند که برگزاری درس در برخی زمان‌ها و در برخی مکان‌ها موجب محاسبه‌ی جریمه توسط تابع هدف شود. در این پژوهش برای استفاده از برخی الگوهای برگزاری دروس نیز جریمه در نظر گرفته شد که این الگوها در ادامه توضیح داده می‌شود. دسته‌بندی محدودیت‌های نرم در این پژوهش عبارت است از:

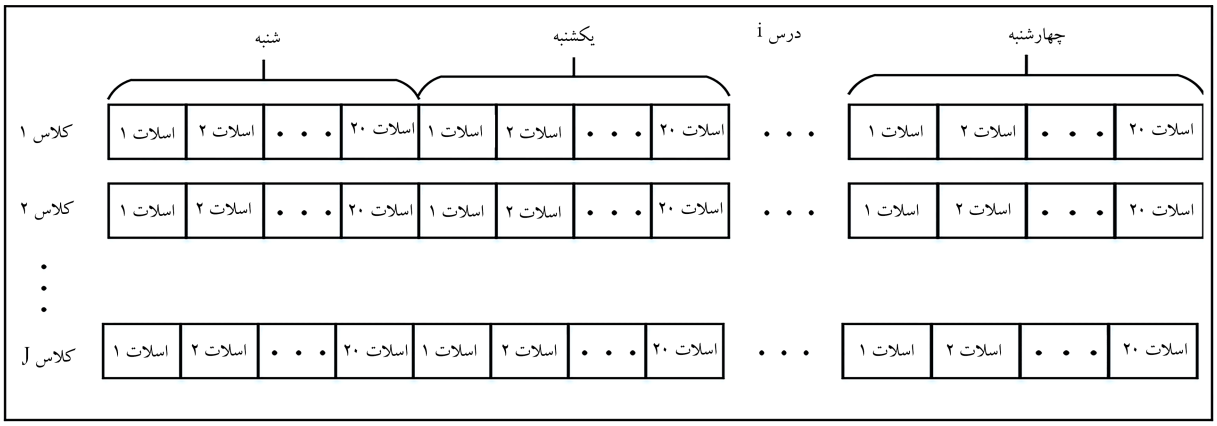
۱. کل جریمه‌ی کاربرد اسلات‌های مختلف برای برگزاری دروس کمینه شود؛

۲. کل جریمه‌ی استفاده از الگوهای برگزاری دروس کمینه شود؛

۳. کل جریمه‌ی استفاده از کلاس‌های مختلف جهت برگزاری دروس کمینه شود.

۲.۲. مدل‌سازی ریاضی

مدل ریاضی جدول‌بندی زمانی دروس می‌تواند دید بهتری نسبت به مسئله‌ی مربوطه ارائه دهد و پیچیدگی‌های موجود در آن را تا حدی تشریح کند. به منظور مدل‌سازی ریاضی این مسئله، هر روز کاری (از ساعت ۸:۰۰ تا ۱۸:۰۰) در هفته به صورت بیست اسلات ۳۰ دقیقه‌یی در نظر گرفته شده که دروس می‌توانند به برخی از این اسلات‌های زمانی تخصیص یابند و مابقی اسلات‌های زمانی غیرقابل استفاده‌اند. اسلات‌های زمانی قابل استفاده را می‌توان در اتاق‌های از پیش تعیین شده برای دروس، توسط درس‌ها پر کرد. جلسات درسی را می‌توان به صورت یک‌بار یا دو بار در هفته برگزار کرد. الگوی برگزاری کلاس‌های دو قسمتی را می‌توان در قالب یکی از شش الگوی شنبه/دوشنبه، یکشنبه/سه‌شنبه، دوشنبه/چهارشنبه، شنبه/سه‌شنبه، یکشنبه/چهارشنبه و شنبه/چهارشنبه در نظر گرفت. در نظر گرفتن این الگوهای



شکل ۱. حالت کلی اسلات‌های زمانی در مسئله‌ی تحقیق.

زمانی در تشکیل کلاس‌های دوقسمتی، یکی از تفاوت‌های نگاه به این مسئله نسبت به مدل‌های کلاسیک است. دوره‌ی تکرار در اکثر مسائل کلاسیک، دو هفته‌یی است در حالی که در مسئله‌ی حاضر براساس واقعیت موجود، یک هفته‌یی در نظر گرفته شد. شکل ۱ حالت کلی اسلات‌های زمانی در یک روز را به صورت تصویری نمایش می‌دهد.

تمامی محدودیت‌های سخت ارائه شده در مدل ریاضی مربوطه براساس الزامات تدوین یک مدل ریاضی مسئله‌ی جدول‌بندی زمانی دروس است و محدودیت‌های نرم مربوط به ترجیحات برگزاری دروس در دانشکده مربوطه است. همچنین پیش‌فرض‌های مدل ریاضی تدوین شده به شرح زیر است:

۱. جریمه‌های تخطی از محدودیت‌های نرم با هم قابل تجمیع‌اند؛

۲. وزن جریمه‌های «تخطی از محدودیت‌های نرم» در تابع هدف برابرند؛

۳. دوره‌ی تکرار برنامه درسی هفتگی است؛

۴. روزهای کاری شامل روزهای شنبه تا چهارشنبه است؛

۵. تعطیلات در مدل ریاضی لحاظ نشده است؛

۶. ظرفیت کلاس‌ها ثابت است؛

۷. امکان برگزاری کلاس زودتر از ۸ صبح و بعد از ساعت ۱۸ وجود ندارد؛

۸. کلاس‌های درس قابل ادغام نیستند؛

۹. تعداد دانشجویان هر کلاس درس ثابت و از قبل مشخص است؛

۱۰. دروس هر گروه درسی از قبل مشخص است؛

۱۱. طول هر کلاس درس ثابت و از قبل مشخص است و امکان طولانی‌تر شدن آن وجود ندارد.

۱.۲.۲. پارامترهای مسئله

Y_i: نوع درس i (یک قسمتی =۰ و دو قسمتی =۱)؛

U_i: تعداد واحد درس i؛

G_i: گروه درسی مربوط به درس i؛

N_i: تعداد دانشجویان درس i؛

P_i: الگوی برگزاری درس i (ماتریس سطری صفر و ۱ به تعداد اسلات‌های موجود)؛

C_k: جریمه‌ی برگزاری کلاس در اسلات زمانی k؛

C_{lk}: جریمه برگزاری کلاس برای استاد l در اسلات k؛

C'_{pi}: جریمه استفاده از الگوی P برای درس i؛

C''_{ij}: جریمه برگزاری درس i در کلاس j؛

A_j: کاربرد کلاس j (دروس خاص =۱، سایر دروس =۰)؛

V_j: ظرفیت کلاس j؛

L_l: مجموعه دروس مربوط به استاد l؛

T_i: مدت زمان درس i در هفته به دقیقه $(i = ۱, \ldots, t)$؛

$Z_۱$: کل جریمه برگزاری درس برای اساتید در اسلات‌های مختلف؛

$Z_۲$: کل جریمه برگزاری درس در اسلات‌های مختلف؛

$Z_۳$: کل جریمه برگزاری درس براساس الگوها؛

$Z_۴$: کل جریمه برگزاری درس در کلاس‌های مختلف؛

- **متغیرهای تصمیم**

X_{ijk}: وضعیت برگزاری درس i در مکان j و در اسلات زمانی k (۱= برگزار می‌شود، ۰= برگزار نمی‌شود.)

- **اندیس‌ها**

i: اندیس درس $(i = ۱, \ldots, I)$؛

j: اندیس کلاس $(j = ۱, \ldots, J)$؛

k: اندیس اسلات زمانی $(k = ۱, \ldots, K)$؛

l: اندیس استاد درس؛

$$MinZ = Z_۱ + Z_۲ + Z_۳ + Z_۴$$

$$s.t.:$$

$$\forall\, i \in L_i: \quad \sum_{j=۱}^{J} \sum_{k=۱}^{K} ۳۰ \times X_{ijk} = T_i \qquad (۱)$$

$$\forall\, i, j: \quad \sum_{k=۱}^{K} X_{ijk} = \begin{cases} ۰ \\ ۲ \times U_i \end{cases} \qquad (۲)$$

$$\forall\, (Y_i = ۱): \quad \sum_{k=۱}^{K} \overrightarrow{P_i} \times \overrightarrow{X_i} = ۲ \times U_i \qquad (۳)$$

$$\forall\, X_{ijk}: \quad N_i \times X_{ijk} \le V_j \times X_{ijk} \qquad (۴)$$

$$\forall\, i, i' \in G_i\,, \forall k: \quad X_{ijk} = \begin{cases} ۰ \\ ۱ - X_{i'jk} \end{cases} \qquad (۵)$$

$$\forall i, i' \in L_l, \forall k: \quad X_{ijk} \times X_{i'jk} = 0 \tag{۶}$$

$$\forall k: \quad \sum_i \sum_j X_{ijk} \leq J \tag{۷}$$

$$\forall i: \quad \sum_j \sum_k X_{ijk} = 2 \times U_i \tag{۸}$$

$$\forall j, k: \quad \sum_i X_{ijk} = \begin{cases} 0 \\ 1 \end{cases} \tag{۹}$$

$$\forall i \in L_i: \quad Z_1 = \sum_{j=1}^{J} \sum_{k=1}^{K} C_{lk} \times X_{ijk} \tag{۱۰}$$

$$\forall X_{ijk}: \quad Z_2 = \sum_i \sum_j \sum_k C_k \times X_{ijk} \tag{۱۱}$$

$$\forall X_{ijk}: \quad Z_3 = \sum_i \sum_j \sum_k C_{pi} \times X_{ijk} \tag{۱۲}$$

$$\forall X_{ijk}: \quad Z_4 = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} C_{ij} \times X_{ijk} \tag{۱۳}$$

محدودیت‌های سخت:

- محدودیت اول تضمین می‌کند که هر درس به اندازه مدت زمان تعیین شده برای آن اتاق بیفتد؛
- محدودیت دوم تضمین می‌کند که هر درس تنها در یک کلاس برگزار شود؛
- محدودیت سوم تضمین می‌کند هر درس براساس الگوی برگزاری آن اتاق بیفتد؛
- محدودیت چهارم تضمین می‌کند هر درس در کلاسی اتاق بیافتد که ظرفیت کافی برای دانشجویان آن درس را داشته باشد؛
- محدودیت پنجم عدم تداخل زمانی دروس یک گروه آموزشی را تضمین می‌کند؛
- محدودیت ششم تضمین می‌کند که دروس یک استاد، دارای تداخل زمانی نباشد؛
- محدودیت هفتم تضمین می‌کند تعداد کلاس‌های برگزار شده در هر اسلات از تعداد کلاس‌های در دسترس بیشتر نباشد؛
- محدودیت هشتم تضمین می‌کند که هر درس متناسب با تعداد واحد آن در هفته برگزار شود؛
- محدودیت نهم تضمین می‌کند هرکلاس به حداکثر یک درس در یک بازه زمانی معین تخصیص یابد.

محدودیت‌های نرم:

- معادله دهم کل جریمه مربوط به زمان تخصیص یافته اساتید برای برگزاری دروس را محاسبه می‌کند؛
- معادله یازدهم کل جریمه به کارگیری اسلات‌های مختلف برای برگزاری دروس را محاسبه می‌کند؛
- معادله دوازدهم کل جریمه استفاده از الگوهای برگزاری دروس را محاسبه می‌کند؛
- معادله سیزدهم کل جریمه استفاده از کلاس‌های مختلف برای برگزاری دروس را محاسبه می‌کند؛
- تابع هدف عبارت است از «کمینه کردن میزان تخطی از محدودیت‌های نرم».

۳.۲. طراحی الگوریتم ابرابتکاری

با توجه به پیچیده بودن مسئله‌ی جدول‌بندی زمانی دروس، الگوریتم ابتکاری سطح بالا بایستی قابلیت‌های فرار از بن‌بست‌های برنامه‌ریزی را داشته باشد و راهکارهای جایگزین برای این موضوع داشته باشد. از طرفی باید بتواند از بهینه‌های محلی بگریزد و به جواب بهینه سراسری نزدیک شود. این ویژگی‌ها در الگوریتم رقابت استعماری ارائه شده توسط محققان ایرانی، آتش‌پزگرگری و لوکاس[۲۲] وجود دارد. با این الگوریتم می‌توان چند برنامه ایجاد کرد و آنها را از لحاظ انطباق با شاخص‌های ارزیابی برنامه مورد سنجش قرار داد و بهترین برنامه را انتخاب کرد.

ازآن‌جاکه پیشتر هیچ محققی از این الگوریتم به عنوان ابتکاری سطح بالا استفاده نکرده است، سفارشی‌سازی الگوریتم رقابت استعماری برای مسئله‌ی جدول‌بندی زمانی دروس به عنوان یک ابتکاری سطح بالا، فرایند دشواری داشت. در این تحقیق به منظور انطباق این الگوریتم به عنوان ابتکاری سطح بالا تغییراتی در آن داده شده که به شرح زیر است:

- الگوریتم رقابت استعماری کلاسیک ابتدا جواب‌ها را به صورت کامل تولید می‌کند و طی مراحلی آنها را تکامل می‌دهد ولی در الگوریتم رقابت استعماری طراحی شده برای مسئله تعدادی جدول زمانی با به کارگیری عملگرها و مفاهیم الگوریتم رقابت استعماری به مرور ساخته می‌شوند؛
- الگوریتم رقابت استعماری کلاسیک برای جذب مستعمره‌ها به استعمارگر خود از سازوکار ساده‌یی نظیر جایگزینی یک متغیر در مستعمره با مقدار آن متغیر در استعمارگر استفاده می‌کند ولی در الگوریتم رقابت استعماری طراحی شده، فرایند جذب مستعمره به استعمارگر در دو سطح عملیاتی و استراتژیک انجام می‌گیرد. در سطح عملیاتی، ابتکاریِ به کار گرفته شده توسط استعمارگر برای تولید جواب مرحله iام عیناً توسط مستعمره نیز به کار گرفته می‌شود. در سطح استراتژیک، استراتژی تولید جواب استعمارگر توسط مستعمره تقلید می‌شود؛
- در الگوریتم رقابت استعماری کلاسیک، مشخصات امپراطوری‌ها یکسان است و تفاوت‌ها به صورت تصادفی ایجاد می‌شود. در الگوریتم رقابت استعماری طراحی شده، به منظور تولید جواب‌های متفاوت و به عبارتی افزایش واریانس بین جدول‌های زمانی تولید شده، شاخصی به نام «تمرکز بر هزینه» تعریف شد که مقدار آن بین صفر و ۱ است و برای هر امپراطوری مقداری متفاوت است. این امر به این دلیل است که تمرکز بر هزینه در زمان‌بندی زیاد باشد، ممکن است در مراحلی به بن‌بست برنامه‌ریزی منجر شود و کلیه اسلات‌های زمانی یک درس را غیرقابل انتخاب سازد و در نقطه‌ی مقابل چنانچه تمرکز بر هزینه حداقل باشد، بقای برنامه و برنامه‌ریزی کل دروس بیشتر می‌شود؛
- الگوریتم رقابت استعماری کلاسیک تک‌هدفه است ولی الگوریتم طراحی شده، دارای دو هدف هزینه و بقاست. هزینه بایستی کاهش یابد و بقا بایستی افزایش یابد. در هر تخصیص مقدار این دو هدف محاسبه و مبنای رقابت درون امپراطوری و بین استعمارگرها می‌شود.

به منظور ساده‌سازی درک برنامه، شبه‌کد ساده شده ابرابتکاری پیشنهادی در نمودار ۲ نشان داده شده است.

در ابرابتکاری توسعه داده شده، استعمارگر هر امپراطوری در هر مرحله مناسب‌ترین درس و بهترین اسلات زمان ـ مکان را انتخاب و طبق آن قسمتی از جدول زمانی را

۱- تنظیمات اولیه الگوریتم: تعداد امپراتوری‌ها برابر پنج
۲- به تعداد دروس انجام بده
۳- به تعداد امپراتوری‌ها انجام بده
۳-۱- اگر درس/ امتحان محدودیت‌دار وجود دارد، استراتژی را یک قرار بده وگرنه استراتژی را به صورت تصادفی بین ۲ تا ۵ انتخاب کن.
۳-۲- بهترین ابتکاری را برای استعمارگر امپراتوری i پیدا کن.
۳-۳- بهترین ابتکاری را برای استعمارگر امپراتوری i اعمال کن.
۳-۴- به تعداد مستعمره‌های امپراتوری i انجام بده.
۳-۴-۱- عدد تصادفی R را تولید کن.
۳-۴-۲- اگر R کوچکتر از «نسبت تشابه ابتکاری» بود، ابتکاری استعمارگر را به این مستعمره اعمال کن.
۳-۴-۳- اگر R بزرگتر از «نسبت تشابه ابتکاری» و کوچکتر از «نسبت تشابه استراتژیک» بود، استراتژی استعمارگر را به مستعمره اعمال کن.
۳-۴-۴- اگر R بزرگتر از مقدار بند قبل بود، استراتژی جدیدی را به مستعمره اعمال کن.
۳-۵- میزان هزینه و بقای استعمارگر و مستعمره‌هایش را محاسبه کن.
۳-۶- رقابت درون امپراتوری i را برقرار کن و اگر مستعمره‌ای برتر از استعمارگر بود، جایگزین آن استعمارگر شود.
۴- رقابت بین امپراتوری‌ها را برقرار کن، امپراتوری غالب، یک مستعمره از ضعیف‌ترین امپراتوری جدا می‌کند و به خود الحاق می‌کند.

نمودار ۲. شبه کد الگوریتم ابرابتکاری پیشنهادی.

تکمیل می‌کند. برای انتخاب مناسب‌ترین درس از بین دروس باقیمانده، ابتکاری‌های نه‌گانه‌ی زیر به کار گرفته شد:

۱. درسی که مکان و زمان آن از قبل به صورت اجباری و بدون حق انتخاب مشخص شده‌اند، ابتدا تخصیص داده شود؛

۲. درسی که مکان یا زمان آن از قبل به صورت اجباری محدود شده است، ابتدا تخصیص داده شود؛

۳. درسی که بیش‌ترین محدودیت را دارد، ابتدا تخصیص داده شود؛

۴. درسی که تعداد اسلات کم‌تری برای آن باقی مانده، ابتدا تخصیص داده شود؛

۵. درسی که تعداد دانشجویان بیش‌تری دارد، ابتدا تخصیص داده شود؛

۶. درسی که طول زمانی بیشتری در هفته لازم دارد، ابتدا تخصیص داده شود؛

۷. درسی که دارای دروس هم‌گروه بیشتری است، ابتدا تخصیص داده شود؛

۸. درسی که استاد آن تعداد بیشتری درس دارد، ابتدا تخصیص داده شود؛

۹. درس استادان هیات علمی، ابتدا تخصیص داده شود.

به منظور تولید جواب‌های متنوع، هر امپراطوری متمرکز بر یک استراتژی برای بهره‌گیری از ابتکاری‌های سطح پایین است. استراتژی‌های پنج‌گانه که هریک در دستور کار یک استعمارگر قرار دارد عبارت‌اند از:

۱. استراتژی اولویت‌دهی به دروس محدودیت‌دار: این استراتژی شامل به کارگیری ابتکاری‌های سطح پایین شماره ۱، ۲، ۳ و ۴ دروس است. یعنی دروسی که به صورت کامل یا نسبی از قبل تخصیص داده شده‌اند، دروس محدودیت‌دار و دروسی که اسلات‌های کمی برای آنها باقی مانده است؛

۲. استراتژی اولویت‌دهی به هزینه: از بین نتایج اجرای ابتکاری‌های سطح پایین، آن ابتکاری انتخاب می‌شود که نسبت به بقیه هزینه‌ی کم‌تری به جدول زمانی اضافه می‌کند؛

۳. استراتژی تعیین اولویت براساس هزینه و بقا: از بین نتایج اجرای ابتکاری‌های سطح پایین، یک ابتکاری انتخاب می‌شود به گونه‌یی که ترکیب هزینه و بقای بهتری داشته باشد. نحوه‌ی ترکیب شدن دو شاخص هزینه و بقا، براساس مقدار شاخص «تمرکز بر هزینه» امپراطوری است که برای هر امپراطوری متفاوت است؛

۴. استراتژی تعیین اولویت براساس اضطرار: از بین نتایج اجرای ابتکاری‌های سطح پایین، یک ابتکاری انتخاب می‌شود به گونه‌یی که شاخص اضطرار بالاتری داشته باشد؛

۵. استراتژی تعیین اولویت به صورت تصادفی: از بین ابتکاری‌های سطح پایین، یک ابتکاری به تصادف انتخاب می‌شود.

پس از مشخص شدن درس بعدی، باید آن را در جدول زمانی قرار داد. برای تعیین محل درس در بین اسلات‌های ممکن، چهار ابتکاری طراحی شده‌اند که غربال‌گری اسلات‌ها و انتخاب از بین اسلات‌های باقیمانده را انجام می‌دهند. این ابتکاری‌ها عبارت‌اند از:

۱. کم‌ترین هزینه← بهترین انطباق ← اولین اسلات ممکن: براساس این ابتکاری، ابتدا از بین اسلات‌های قابل تخصیص به درس، اسلات‌هایی که کم‌ترین هزینه را دارند گزینش می‌شوند و سپس از بین آنها، اسلات‌هایی که مربوط به کلاس‌هایی با ظرفیت نزدیک به تعداد دانشجویان درس هستند، انتخاب شده و از بین آنها، اولین اسلات ممکن انتخاب می‌شود؛

۲. کم‌ترین هزینه ← بهترین انطباق ← آخرین اسلات ممکن: گام‌های این ابتکاری شبیه ابتکاری شماره ۱ است با این تفاوت که درگام آخر، آخرین اسلات ممکن انتخاب می‌شود؛

۳. بهترین انطباق ← کم‌ترین هزینه ←اولین اسلات ممکن: براساس این ابتکاری، ابتدا از بین اسلات‌های قابل تخصیص به درس، اسلات‌هایی که مربوط به کلاس‌هایی با بهترین انطباق بین ظرفیت آنها و تعداد دانشجویان درس است گزینش می‌شود و سپس بین آنها، اسلات‌هایی که کم‌ترین هزینه را دارند انتخاب می‌شود و درانتها اولین اسلات از بین اسلات‌های باقیمانده ممکن انتخاب می‌شود؛

۴. بهترین انطباق ← کم‌ترین هزینه ← آخرین اسلات ممکن: این ابتکاری نیز شبیه ابتکاری شماره ۳ است با این تفاوت که در انتها، آخرین اسلات ممکن انتخاب می‌شود.

۴.۲. تنظیمات الگوریتم ابرابتکاری

از آن‌جا که در این تحقیق از الگوریتم رقابت استعماری به عنوان ابتکاری سطح بالا استفاده شده، پارامترهای این الگوریتم باید مشخص شوند. به طور معمول این کار با روش آزمون و خطا یا روش طراحی آزمایشات[۴] انجام می‌شود. از آن‌جا که ابرابتکاری قرار است برای اندازه‌های متفاوت مسئله قادر به تولید جواب باشد لذا پارامترهای آن بایستی طوری باشند که نیاز به تنظیم توسط کاربر نداشته باشند و خود الگوریتم آنها را تنظیم کند. پارامتر ξ براساس مقاله اصلی الگوریتم رقابت استعماری، برابر ۰/۱ قرارگرفته است. بر این اساس شاخص هزینه‌ی هر امپراطوری برابر هزینه‌ی استعمارگر به علاوه یک‌دهم متوسط هزینه‌ی مستعمره‌ها می‌شود. در ابرابتکاری طراحی شده، علاوه بر هزینه، شاخص بقا نیز به همین شکل برای امپراطوری‌ها تعریف شده است. به منظور تغییر پویای پارامتر ξ، ابتدا مقدار پارامتر برابر ۰/۱ قرارگرفت و براساس فرمول ۱، در دور بعد مقدار آن تغییر می‌یابد. در عمل، اگر رقابت بین امپراطوری‌ها منجر به جابه‌جایی یک مستعمره می‌شود، این مقدار کاهش می‌یابد و برعکس اگر رقابت مؤثر بین امپراطوری‌های صورت نگیرد، این مقدار کاهش می‌یابد. افزایش مقدار ξ منجر به تحریک رقابت بین استعمارگران می‌شود و بالعکس.

$$\xi_{new} = \xi_{old}^{signal} \quad signal = \begin{cases} \text{اگر رقابت مؤثر نباشد } ۰/۹ \\ \text{اگر رقابت مؤثر باشد } ۱/۱ \end{cases} \qquad (۱۴)$$

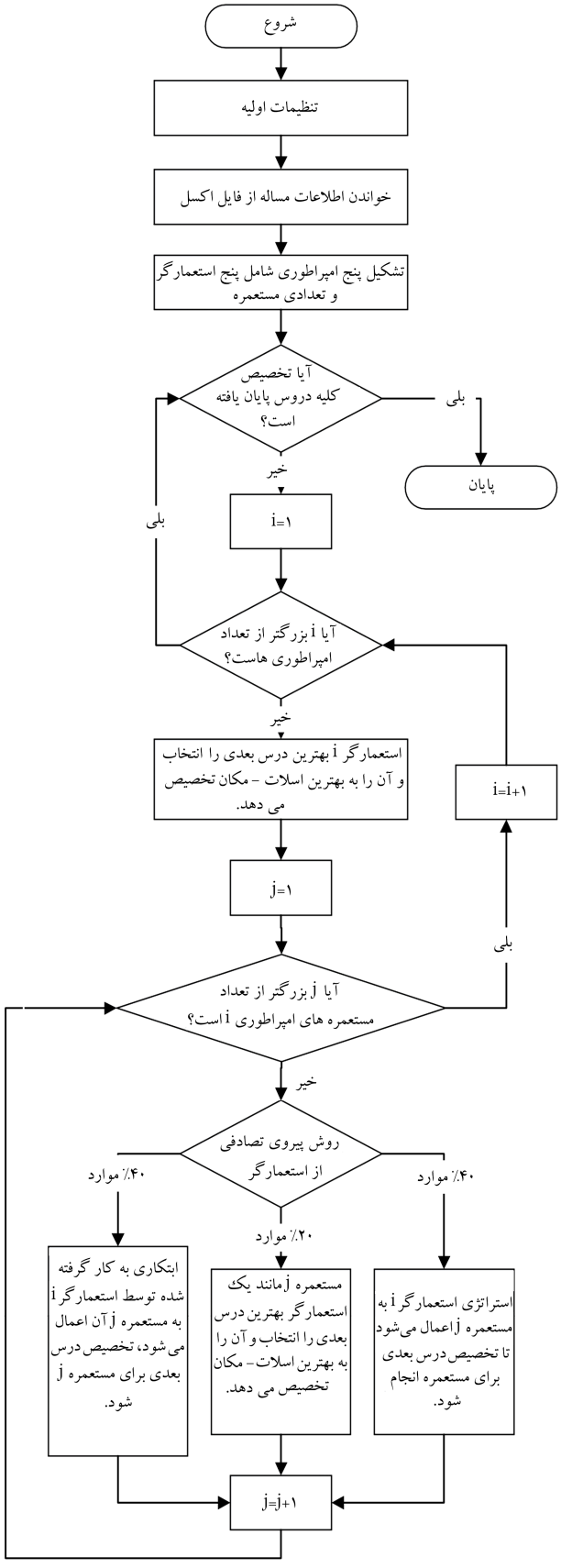
براساس فرمول ۱۴، مقدار ξ به طور پویا توسط عملکرد خود الگوریتم تنظیم می‌شود و در هر صورت مقدار آن بین صفر و ۱ باقی می‌ماند.

در ابرابتکاری طراحی شده پارامتر تعداد امپراطوری‌ها برابر تعداد استراتژی‌ها یعنی ۵ قرارگرفت زیرا هر امپراطوری تمرکز ویژه‌یی بر اجرای یک استراتژی دارد و مستعمره‌های خود را براساس آن به سمت تکامل پیش می‌برد. تعداد کشورها نیز باید مضربی از ۵ باشد و هرچه تعداد کشورها بیشتر باشد، برنامه‌های نهایی بیشتری به دست خواهد آمد و احتمال برخورد با بن‌بست برنامه‌ریزی نیز کم‌تر می‌شود ولی زمان بیشتری برای اجرای الگوریتم نیاز است.

۵.۲. برنامه‌نویسی رایانه‌یی و اجرای ابرابتکاری

ابرابتکاری طراحی شده برای حل مسئله‌ی جدول‌بندی زمانی دروس شامل خواندن اطلاعات از روی فایل اکسل، قراردهی یک به یک دروس در جدول زمانی و ارزیابی مستمر این فرایند است؛ فلوچارت ساده شده‌ی الگوریتم ابرابتکاری در نمودار ۳ آمده است.

ورودی‌های بکار گرفته شده جهت آزمودن الگوریتم، داده‌های نیمسال اول تحصیلی ۹۸-۹۷ دانشکده اقتصاد، مدیریت و علوم اجتماعی دانشگاه شیراز است. از آن‌جا که دسترسی به ترجیحات زمانی اعلام شده توسط اساتید امکان‌پذیر نبود، ساعات کلاس آنها در این نیم‌سال به عنوان ترجیح اول با جریمه‌ی صفر و سایر ساعات هفته با هزینه‌ی ۴ به عنوان ترجیح زمانی اساتید در نظرگرفته شد. اطلاعات دروس شامل واحد، نوع، مدرس(ها)، هم‌زمان و هم‌مکان و هم‌استاد بودن آنها از آمار نیمسال استخراج شد. گروه‌های درسی نیز از برنامه درسی مقاطع و رشته‌های



نمودار ۳. فلوچارت ساده شده الگوریتم ابرابتکاری پیشنهادی.

جدول ۱. زمان اجرای ابرابتکاری پیشنهادی.

ردیف	تعداد کشورها	زمان مورد نیاز
۱	۱۰	۱۷ ساعت
۲	۳۰	۲ روز و ۳ ساعت
۳	۵۰	۳ روز و ۱۲ ساعت

مختلف استخراج و با استفاده از داده‌های مربوط به دانشجویان مشترک، دروس گروه‌بندی شدند.

اطلاعات مربوط به دروس از سیستم اتوماسیون آموزشی دانشگاه شیراز استخراج شد و اطلاعات مربوط به کلاس‌ها شامل ظرفیت، بخش مربوطه، خاص بودن، مخصوص تحصیلات تکمیلی بودن آنها از کارشناسان آموزش بخش‌ها گرفته شد. به منظور رهایی از بهینه‌سازی بخشی و همچنین در نظر گرفتن نظم فعلی تخصیص کلاس‌های درس به بخش‌ها، برای اختصاص کلاس یک بخش به سایر بخش‌ها جریمه‌یی در نظرگرفته شده که برای کلاس‌های مخصوص تحصیلات تکمیلی (نظیر سالن کنفرانس بخش‌ها) این هزینه بی‌نهایت است یعنی تخصیص آن به سایر بخش‌ها امکان‌پذیر نیست و برای کلاس‌های دیگر، این هزینه ۴ است. چنانچه سیاست کلی دانشکده، عدم واگذاری کلاس‌ها به بخش‌ها و برنامه‌ریزی یکپارچه توسط معاونت آموزشی باشد می‌توان این جریمه را صفر در نظر گرفت.

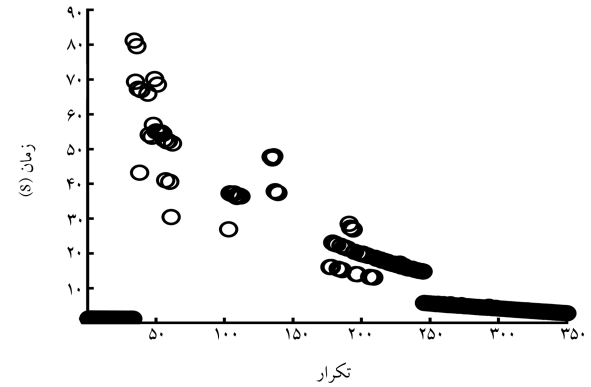
در این تحقیق ابتدا مفاهیم الگوریتم رقابت استعماری بر مسئله‌ی جدول‌بندی زمانی دروس تطبیق داده شد و سپس از آن به عنوان ابتکاری سطح بالا در قالب یک ابرابتکاری استفاده شد. در نهایت پیاده‌سازی نرم‌افزاری الگوریتم ابرابتکاری منجر به تعریف ۱۱ تابع مجزا در نرم‌افزار متلب شامل حدوداً ۲۰۰۰ خط برنامه شد. به منظور پیاده‌سازی الگوریتم ابرابتکاری طراحی شده، از نرم‌افزار متلب نسخه b۲۰۱۸R و برای آماده‌سازی ورودی‌ها از نرم‌افزار مایکروسافت اکسل نسخه ۲۰۱۶ Professional plus استفاده شد. اجرای الگوریتم روی رایانه‌ی خانگی با پردازنده‌ی i۵ – ۳۴۵۰ با سرعت ۳٫۵۰ – ۳٫۱۰ GHZ و حافظه‌ی موقت (RAM) ۸ گیگابایت مجهز به سیستم عامل ۶۴ بیتی ویندوز ۱۰ نسخه ۲۰۱۵ Enterprise انجام شد. زمان اجرای الگوریتم ابرابتکاری طراحی شده، براساس تعداد کشورها متفاوت است. جدول ۱ تغییر زمان تقریبی مورد نیاز برای الگوریتم براساس تعداد کشورها را نشان می‌دهد.

۳. نتایج و بحث

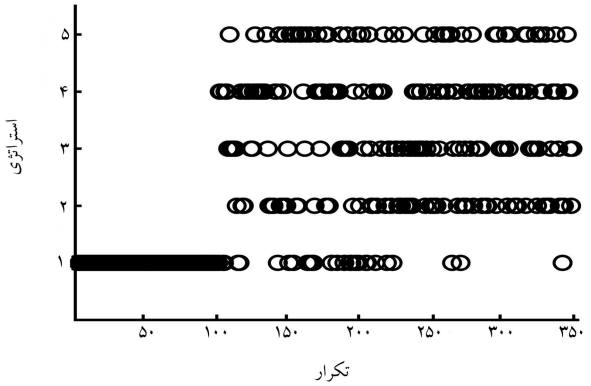
پس از اجرای الگوریتم ابرابتکاری روی مجموعه داده‌های واقعی، عملکرد زمانی و رفتاری الگوریتم در استفاده از ابتکاری‌های سطح پایین، استراتژی‌ها و ابتکاری‌های انتخاب اسلات مورد بررسی قرارگرفت. نتایج به دست آمده از فرایند بهترین جدول زمانی تولید شده توسط الگوریتم ابرابتکاری طراحی شده در ادامه در دو قسمت فرایند و نتایج اجرای الگوریتم ابرابتکاری تشریح می‌شود. در این نمودارها، هر تکرار، به تخصیص یک درس مربوط می‌شود.

۱.۳. فرایند اجرای الگوریتم ابرابتکاری

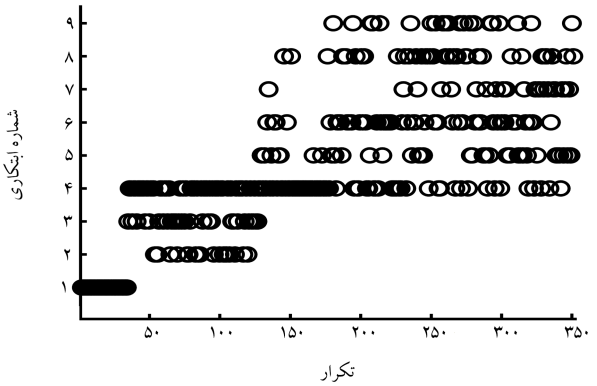
براساس نتایج به دست آمده، زمان انتخاب و جای‌دهی هر درس طی اجرای الگوریتم متفاوت است. در نمودار ۴ تغییر زمان را حین پیشرفت الگوریتم شاهد هستیم.



نمودار ۴. متوسط زمان تخصیص هر درس در هر تکرار.



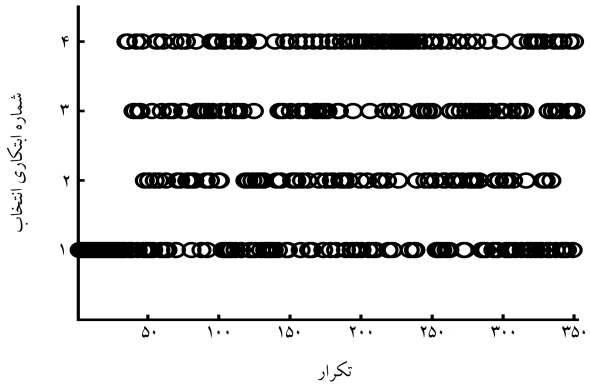
نمودار ۵. ترتیب انتخاب استراتژی‌ها در بهترین جواب ابرابتکاری.



نمودار ۶. ترتیب به کارگیری ابتکاری‌های سطح پایین.

چنان که در نمودار ۴ مشاهده می‌شود زمان تخصیص دروس در چند درس ابتدایی بسیارکم است و علت آن است که ابتدا دروس از قبل تخصیص یافته در جدول زمانی جای داده می‌شوند و چون این دروس گزینه‌های زیادی برای قرارگرفتن در برنامه ندارند، بنابراین محاسبات زیادی هم ندارند و به سرعت تخصیص آنها به اسلات‌ها و مکان‌ها انجام می‌شود ولی در ادامه زمان بیشتری نیاز است تا حالات زیادی بررسی شود و به تدریج که این حالات کم‌تر می‌شود (به دلیل محدود شدن حالات با پیشرفت برنامه) زمان کاهش می‌یابد.

براساس پیشینه‌ی تحقیق، ترتیب استفاده از ابتکاری‌ها بر کیفیت جواب اثرگذار است و هر تحقیق ترتیب خاصی را ارائه می‌دهد. براساس یافته‌های این تحقیق بهترین استراتژی و ابتکاری انتخاب درس و اسلات به شرح نمودارهای ۵ و ۶ است.



نمودار ۷. ترتیب به کارگیری ابتکاری‌های انتخاب.

الگوریتم ابرابتکاری طراحی شده در ابتدا بر تخصیص دروس محدودیت‌دار تمرکز دارد و این اولین استراتژی انتخاب دروس است. به همین دلیل در نمودار ۵، در تکرارهای ابتدایی الگوریتم، میزان استفاده از این استراتژی بالاست. در ادامه نیز هنگامی که درسی دارای محدودیت در تخصیص می‌شود، با این استراتژی در اولویت تخصیص قرار می‌گیرد. استفاده از سایر استراتژی‌ها نیز به صورت تصادفی انجام می‌شود که یکسانی استفاده از آنها در نمودار دیده می‌شود.

نمودار ۶ میزان استفاده ابرابتکاری از ابتکاری‌های سطح پایین انتخاب را نشان می‌دهد. چنان که در شکل مشاهده می‌شود، در ابتدای شروع به کارگیری ابرابتکاری، تخصیص براساس ابتکاری سطح پایین اول انجام می‌شود که همان دروس از پیش تخصیص داده شده هستند. مکان و زمان این دروس ابتدا تنظیم می‌شود در ادامه از آنجا که امپراطوری مورد نظر در ابتدا براساس استراتژی اول جلو می‌رود، از ابتکاری‌های ۲، ۳ و ۴ استفاده می‌کند و پس از آن از سایر ابتکاری‌ها به تصادف براساس استراتژی اتخاذ شده استفاده شده است. ابتکاری شماره ۴ در این مسئله به کارگیری بالایی داشته است. این ابتکاری به اولویت‌دهی به دروسی که اسلات کم‌تری برای آنها باقیمانده مربوط می‌شود در مواقعی از فرایند تخصیص دروس به اسلات‌ها ممکن است این وضعیت برای سایر دروس بوجود آید به همین دلیل به کرات از این ابتکاری استفاده شده است.

نمودار ۷ میزان استفاده از ابتکاری‌های انتخاب اسلات برای قراردهی درس را نشان می‌دهد. از آن‌جا که به کارگیری این ابتکاری‌ها به صورت تصادفی است، میزان استفاده از هر ابتکاری تقریباً برابر است. در ابتدای الگوریتم (تخصیص دروس دارای محدودیت) از آن‌جا که برای انتخاب اسلات گزینه‌یی وجود ندارد، اولین ابتکاری به این کار تخصیص می‌یابد و به همین دلیل در ابتدای الگوریتم میزان استفاده از این ابتکاری بالاست.

۲.۳. نتایج اجرای الگوریتم ابرابتکاری

الگوریتم طراحی شده قادر است تعداد زیادی جدول زمانی ارائه دهد. به منظور ارزیابی این راه‌حل‌ها شباهت آنها با وضع موجود و یکدیگر مورد بررسی قرار می‌گیرد. میزان مشابهت با دو معیار مشابهت زمانی و مشابهت مکانی در نظر گرفته شده است. منظور از مشابهت زمانی، یکسان بودن زمان برگزاری درس و مشابهت مکانی یکسان بودن کلاس درس است. الگوریتم ابرابتکاری طراحی شده ۱۰ برنامه در هربار اجرا تولید می‌کند که هر ۱۰ برنامه متفاوت از یکدیگر هستند. در نهایت متوسط میزان شباهت جدول زمان‌بندی دروس حاصل از الگوریتم با برنامه طراحی شده توسط کارشناسان بخش‌ها به لحاظ زمانی حدود ۱۶ درصد و به لحاظ مکانی حدود ۳۱ درصد است. دلیل مشابهت پایین برنامه‌ی تولید شده توسط نرم‌افزار و برنامه‌ی تولید شده توسط کارشناسان آموزش آن است که نرم‌افزار از الگوهای غیرسنتی برگزاری کلاس‌های دوقسمتی نظیر الگوی شنبه/سه‌شنبه، شنبه /چهارشنبه، یکشنبه/سه‌شنبه و یکشنبه/چهارشنبه نیز استفاده می‌کند و قابلیت بالاتری برای تولید برنامه کلاسی دارد.

منظور از گسستگی در جدول زمانی، وجود زمان‌های خالی بین دروس برنامه‌ریزی شده در یک روز برای یک گروه درسی (دانشجویان دارای دروس مشابه) است. در واقع این گسست به معنای انتظار دانشجویان یک گروه درسی برای شروع درس بعدی است. هرقدر این گسست بیشتر باشد، نارضایتی دانشجویان بیشتر می‌شود. براساس گروه‌بندی دروس، ۷۲ گروه درسی در نیم‌سال مورد نظر در دانشکده مورد بررسی وجود داشت. جدول ۲ میزان گسست در جداول زمانی تولید شده توسط الگوریتم ابرابتکاری طراحی شده و وضعیت موجود را به دقیقه نشان می‌دهد. هر عدد بیان‌گر مجموع زمان انتظار بین دروس برای دانشجویان یک گروه در یک هفته است.

همان‌طور که جدول ۲ نشان می‌دهد، جداول زمانی تولید شده توسط ابرابتکاری طراحی شده، قادر است به طور متوسط بیش از یک ساعت در هفته زمان انتظار دانشجویان هر گروه را کاهش دهد. شاخص مهم دیگر برای ارزیابی جدول زمانی تولید شده، میزان بهره‌برداری از فضای کلاس است. منظور از بهره‌برداری از فضای کلاس، درصد صندلی‌های استفاده شده‌ی کلاس در درس تخصیص داده شده به آن است. ابرابتکاری طراحی شده با بهره‌گیری از ابتکاری‌های انتخاب اسلات، ظرفیت کلاس‌های در دسترس را در نظر می‌گیرد و سعی می‌کند کلاسی را انتخاب کند که بهترین انطباق را با تعداد دانشجوی درس داشته باشد. بر این اساس به لحاظ نظری باید میزان بهره‌برداری از ظرفیت کلاس‌ها با استفاده از ابرابتکاری، بهتر از وضعیت عدم استفاده از آن باشد. به لحاظ تجربی نیز محاسبات مربوطه انجام شد و متوسط میزان بهره‌برداری از کلاس‌ها در برنامه‌های پیشنهاد شده توسط ابرابتکاری ۶۴٫۲۳ درصد است که نسبت به وضعیت موجود (۵۳ درصد)، حدود ۱۱ درصد بهبود را نشان می‌دهد. جدول ۳ درصد بهره‌برداری جداول زمانی حاصل از ابرابتکاری و وضع موجود را نشان می‌دهد. چنان که مشاهده می‌شود، بهترین برنامه تولید شده، برنامه شماره هفت است که میزان بهره‌برداری از ظرفیت کلاس‌ها براساس این برنامه ۶۵٫۰۴ درصد است.

۴. اعتبارسنجی الگوریتم ابرابتکاری

الگوریتم ابرابتکاری طراحی شده، متناسب با ماهیت خاص و منحصر بفرد اطلاعات ورودی شامل دروس، محدودیت‌ها و گروه‌های درسی است و امکان مقایسه با الگوریتم‌های دیگر یا ورود اطلاعات مسائل استاندارد را ندارد. به منظور ارزیابی اعتبار ابرابتکاری طراحی شده، با الهام از پژوهش کندال و حسین (۲۰۰۵) و عبدالرحمن و همکاران (۲۰۱۴)، خروجی ابرابتکاری مذکور با داده‌های واقعی مقایسه شد. این ارزیابی نشان داد که جداول زمانی تولید شده توسط ابرابتکاری طراحی شده، بیش از یک‌ساعت از گسستگی زمانی برنامه درسی هفتگی دانشجویان می‌کاهد و با تخصیص کلاس متناسب با تعداد دانشجویان، حدود یازده درصد بهره‌برداری از کلاس‌ها را افزایش می‌دهد. همچنین صحت برنامه‌های تولید شده توسط کارشناسان آموزش مورد تأیید قرار گرفت.

جدول ۲. میزان گسستگی زمانی جداول زمانی تولید شده و موجود.

گروه درسی	وضع موجود	جداول زمانی تولید شده بوسیله ابرابتکاری پیشنهادی									
		۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۱	۱۲۵	۰	۰	۲۵۰	۰	۰	۰	۱۲۵	۰	۱۲۵	۲۵۰
۲	۳۱۳	۲۱۹	۲۱۹	۰	۲۱۹	۲۱۹	۲۱۹	۰	۰	۲۱۹	۰
۳	۳۷۵	۵۶۳	۵۶۳	۰	۵۶۳	۵۶۳	۵۶۳	۴۳۸	۲۱۹	۱۵۶	۰
۴	۴۳۸	۳۱	۳۱	۰	۳۱	۳۱	۳۱	۶۲	۰	۴۳۸	۰
۵	۳۱۳	۲۵۰	۲۵۰	۲۵۰	۲۵۰	۲۵۰	۲۵۰	۵۶۳	۳۴۴	۰	۲۵۰
۶	۱۸۸	۴۳۸	۰	۰	۰	۰	۰	۹۴	۱۲۵	۰	۰
۷	-	۵۳۱	۴۰۶	۱۸۸	۴۰۶	۴۰۶	۴۰۶	۲۵۰	۵۶۳	۲۵۰	۱۸۸
۸	۱۲۵	۳۴۴	۳۴۴	۱۲۵	۳۴۴	۳۴۴	۳۴۴	۰	۰	۴۰۶	۱۲۵
۹	۴۰۶	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۱۰	۱۲۵	۳۴۴	۳۴۴	۱۲۵	۳۴۴	۳۴۴	۳۴۴	۰	۰	۴۰۶	۱۲۵
۱۱	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۰	۰	۴۰۶	۱۲۵
۱۲	۳۱	۰	۲۱۹	۴۶۹	۳۴۴	۰	۱۵۶	۱۲۵	۰	۰	۰
۱۳	۴۰۶	۶۲۵	۴۶۹	۳۷۵	۳۱۳	۴۰۶	۳۱۳	۶۲۵	۳۱۳	۳۱	۲۸۱
۱۴	۱۸۸	۹۴	۲۱۹	۱۵۶	۲۱۹	۲۱۹	۲۱۹	۹۴	۰	۰	۱۵۶
۱۵	۰	۶۲	۹۴	۱۲۵	۱۵۶	۴۳۸	۴۳۸	۱۲۵	۶۳	۶۲	۱۲۵
۱۶	۳۱	۶۲	۱۵۶	۵۶۳	۹۴	۹۴	۹۴	۵۶۳	۶۲	۱۲۵	۵۶۳
۱۷	۶۳	۱۸۸	۱۸۸	۰	۱۸۸	۱۸۸	۱۸۸	۴۶۹	۱۸۸	۵۶۳	۰
۱۸	۲۵۰	۶۲	۹۴	۶۲	۱۵۶	۱۲۵	۱۲۵	۱۵۶	۱۲۵	۲۸۱	۰
۱۹	۲۸۱	۲۱۹	۶۲	۶۲	۲۵۰	۶۲	۶۲	۳۱۳	۶۲	۳۱۳	۶۲
۲۰	۰	۳۷۵	۴۰۶	۳۱۳	۳۷۵	۳۷۵	۳۷۵	۷۵۰	۵۰۰	۵۶۳	۳۱۳
۲۱	۹۴	۱۸۸	۱۲۵	۶۲	۱۲۵	۶۲	۶۲	۱۲۵	۲۵۰	۱۲۵	۶۲
۲۲	-	۶۲	۶۲	۶۲	۶۲	۱۲۵	۱۲۵	۶۲	۶۲	۶۲	۶۲
۲۳	۶۲	۰	۶۲	۰	۰	۰	۰	۰	۰	۶۲	۰
۲۴	۰	۹۴	۰	۱۲۵	۶۲	۶۲	۶۲	۰	۶۲	۲۵۰	۰
۲۵	۰	۶۲	۰	۰	۰	۶۲	۶۲	۶۲	۰	۶۲	۶۲
۲۶	۶۲	۶۲	۰	۰	۰	۹۴	۹۴	۰	۹۴	۰	۱۸۸
۲۷	۱۸۸	۰	۶۲	۰	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۰	۲۵۰	۱۲۵
۲۸	۶۲	۰	۱۲۵	۰	۱۲۵	۱۲۵	۱۲۵	۰	۱۲۵	۰	۰
۲۹	۶۲	۱۲۵	۱۲۵	۶۲	۱۲۵	۱۲۵	۱۲۵	۳۷۵	۱۲۵	۰	۶۲
۳۰	۰	۰	۲۵۰	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۰	۱۲۵
۳۱	۰	۶۲	۰	۰	۰	۰	۰	۰	۰	۰	۰
۳۲	۳۱۳	۶۲	۱۸۸	۱۲۵	۶۲	۱۲۵	۱۲۵	۶۲	۶۲	۱۵۶	۱۵۶
۳۳	۶۲	۶۲	۱۲۵	۱۲۵	۱۵۶	۱۸۸	۱۸۸	۳۱۳	۱۲۵	۱۲۵	۱۲۵
۳۴	۴۳۸	۱۲۵	۱۸۸	۶۲	۱۲۵	۶۲	۶۲	۶۲	۱۲۵	۲۸۱	۱۸۸
۳۵	۱۲۵	۱۸۸	۱۲۵	۱۸۸	۱۲۵	۱۲۵	۱۲۵	۲۱۹	۱۲۵	۱۲۵	۱۲۵
۳۶	-	۹۴	۱۵۶	۵۶۳	۴۰۶	۹۴	۹۴	۲۱۹	۲۱۹	۱۵۶	۱۵۶
۳۷	۲۵۰	۱۲۵	۲۸۱	۶۲	۱۸۸	۶۲	۶۲	۶۲	۱۲۵	۱۲۵	۱۸۸
۳۸	۳۱۳	۶۲	۱۸۸	۶۲	۶۲	۶۲	۶۲	۳۱	۶۲	۲۸۱	۲۸۱
۳۹	۴۰۶	۱۵۶	۶۲	۳۴۴	۱۵۶	۶۲	۶۲	۱۲۵	۲۸۱	۲۸۱	۳۴۴
۴۰	۶۲۵	۰	۶۳	۰	۶۲	۶۲	۶۲	۶۲	۱۸۸	۱۲۵	۱۸۸

ادامه‌ی جدول ۲.

گروه درسی	وضع موجود	جداول زمانی تولید شده بوسیله ابرابتکاری پیشنهادی									
		۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۴۰	۶۲۵	۰	۶۳	۰	۶۲	۶۲	۶۲	۶۲	۱۸۸	۱۲۵	۱۸۸
۴۱	۵۹۴	۳۷۵	۴۳۸	۲۱۹	۱۲۵	۳۴۴	۳۴۴	۲۸۱	۱۵۶	۶۲	۲۱۹
۴۲	۶۲	۲۵۰	۱۲۵	۲۱۹	۴۰۶	۶۲	۶۲	۴۳۸	۱۵۶	۶۲	۲۱۹
۴۳	۸۱۳	۲۱۹	۶۲	۱۲۵	۰	۰	۰	۶۸۸	۶۲	۱۲۵	۱۲۵
۴۴	۵۰۰	۱۸۸	۱۸۸	۱۸۸	۱۸۸	۱۸۸	۱۸۸	۰	۱۲۵	۱۲۵	۱۸۸
۴۵	۴۳۸	۱۸۸	۱۸۸	۰	۱۸۸	۱۸۸	۱۸۸	۶۲	۴۰۶	۹۴	۰
۴۶	۶۲	۰	۰	۱۵۶	۰	۰	۰	۲۵۰	۲۱۹	۳۴۴	۱۵۶
۴۷	۱۵۶	۶۲	۶۲	۰	۶۲	۶۲	۶۲	۳۱۳	۱۲۵	۵۰۰	۰
۴۸	۰	۳۱۳	۳۱۳	۹۴	۳۱۳	۳۱۳	۳۱۳	۰	۳۱۳	۰	۹۴
۴۹	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۵۰	۱۸۸	۱۲۵	۱۲۵	۰	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۸۸	۶۲	۰
۵۱	۰	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۲۵	۱۸۸	۱۸۸	۱۸۸	۱۲۵
۵۲	۰	۰	۶۲	۳۱	۶۲	۰	۰	۰	۱۲۵	۰	۰
۵۳	۲۸۱	۰	۰	۲۱۹	۰	۰	۰	۱۲۵	۱۲۵	۳۴۴	۲۱۹
۵۴	۵۳۱	۱۲۵	۲۱۹	۰	۱۲۵	۳۴۴	۳۴۴	۳۱	۳۱	۰	۰
۵۵	۹۳۸	۲۵۰	۲۵۰	۰	۲۵۰	۲۵۰	۲۵۰	۲۵۰	۶۲	۶۲	۰
۵۶	۱۸۸	۵۰۰	۶۲۵	۳۱	۶۲۵	۶۲۵	۶۲۵	۲۵۰	۳۱۳	۰	۳۱
۵۷	۲۱۹	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰
۵۸	۱۸۸	۲۵۰	۲۵۰	۵۹۴	۲۵۰	۲۵۰	۲۵۰	۵۶۳	۲۸۱	۱۸۸	۵۹۴
۵۹	۵۰۰	۵۳۱	۴۰۶	۹۴	۴۰۶	۴۰۶	۴۰۶	۹۴	۱۲۵	۳۴۴	۹۴
۶۰	۵۳۱	۳۷۵	۹۴	۰	۰	۰	۰	۳۱	۹۶۹	۳۷۵	۹۴
۶۱	۶۳	۱۸۸	۱۸۸	۱۸۸	۱۸۸	۱۸۸	۱۸۸	۶۲	۱۲۵	۶۲	۱۸۸
۶۲	۲۱۹	۲۵۰	۲۵۰	۶۲	۲۵۰	۲۵۰	۲۵۰	۶۲	۳۱۳	۰	۶۲
۶۳	۰	۰	۱۲۵	۰	۶۲	۰	۰	۱۲۵	۶۲	۰	۱۲۵
۶۴	۲۵۰	۶۲	۶۲	۱۲۵	۶۲	۹۴	۹۴	۲۱۹	۹۴	۱۲۵	۱۲۵
۶۵	۰	۰	۰	۰	۰	۰	۰	۶۲	۰	۶۲	۰
۶۶	۱۸۸	۶۲	۶۲	۶۲	۶۲	۶۲	۶۲	۱۲۵	۱۲۵	۶۲	۶۲
۶۷	۰	۰	۰	۱۸۸	۰	۰	۰	۰	۰	۰	۱۸۸
۶۸	۰	۰	۰	۶۲	۱۸۸	۶۲	۶۲	۰	۱۲۵	۰	۰
۶۹	۰	۶۲	۶۲	۶۲	۶۲	۶۲	۶۲	۶۲	۹۴	۹۴	۶۲
۷۰	۱۲۵	۰	۰	۱۲۵	۰	۰	۰	۱۲۵	۰	۰	۱۲۵
۷۱	۲۱۹	۹۴	۲۱۹	۶۲	۱۸۸	۱۸۸	۱۸۸	۰	۲۱۹	۰	۶۲
۷۲	۱۲۵	۹۴	۹۴	۱۸۸	۹۴	۹۴	۹۴	۵۳۱	۶۲	۴۰۶	۵۶۳
میانگین	**۲۰۶**	**۱۴۹**	**۱۵۲**	**۱۱۶**	**۱۵۱**	**۱۴۲**	**۱۴۳**	**۱۶۴**	**۱۳۸**	**۱۴۶**	**۱۲۱**

جدول ۳. درصد بهره‌برداری از فضاهای آموزشی در وضع موجود و جدول زمانی تولید شده توسط الگوریتم ابرابتکاری پیشنهادی.

جدول زمانی	وضع موجود	جداول زمانی تولید شده به وسیله‌ی ابرابتکاری پیشنهادی									
		۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
بهره‌برداری (٪)	۵۳	۶۴٫۰۱	۶۴٫۱۲	۶۴٫۲۱	۶۳٫۸۹	۶۳٫۸۳	۶۳٫۸۱	۶۵٫۰۴	۶۴٫۴۵	۶۴٫۶۴	۶۴٫۲۷

۵. نتیجه‌گیری

مسئله ایجاد جدول زمانی برای دروس دانشگاهی یکی از مسائل مهم و پیچیده است که راه‌حل ساده برای آن وجود ندارد. یکی از رویکردهای نوین برای حل این مسئله، توسعه‌ی الگوریتم‌های ابرابتکاری است. هدف این مقاله، در نظرگرفتن دروس دوقسمتی و تخصیص اولیه‌ی کلاس‌ها به بخش‌های آموزشی در مدل‌سازی ریاضی و حل این مسئله با یک الگوریتم ابرابتکاری جدید است تا بتوان نسبت به وضعیت برنامه‌ریزی دستی بهبودهایی ایجاد کرد. این هدف از طریق طراحی یک رویکرد ابرابتکاری مبتنی بر الگوریتم رقابت استعماری به عنوان ابتکاری سطح بالا و تعدادی ابتکاری مبتنی بر نظریه گراف همراه با چند استراتژی و کدنویسی آن در نرم‌افزار متلب محقق شده است.

این پژوهش از لحاظ جنبه کاربردی به دلیل نزدیکی مدل‌سازی به دنیای واقعی دانشگاه‌های کشور می‌تواند برای برنامه‌ریزان دانشگاهی مفید واقع شود و ایده‌هایی برای اجرا و پیاده‌سازی الگوریتم ابرابتکاری در عمل به آنها ارائه کند.

یافته‌های این پژوهش نشان داد که الگوریتم ابرابتکاری طراحی شده قادر است تعداد برنامه‌های بیشتر و متنوع‌تری را نسبت به برنامه‌ی فعلی تولید کند. همچنین به دلیل تولید ۱۰ برنامه به جای یک برنامه به کارشناسان این اجازه را می‌دهد که بهترین برنامه را برای بخش مربوطه انتخاب کنند. جداول زمانی تولید شده توسط ابرابتکاری طراحی شده، قادر است به طور متوسط بیش از ۱ ساعت در هفته زمان انتظار دانشجویان هر گروه را کاهش دهد. متوسط میزان بهره‌برداری از کلاس‌ها را حدود ۱۱ درصد بهبود دهد و در نهایت این که برنامه تولید شده توسط کارشناسان بخش دانشکده مربوطه دست‌کم یک ماه کاری به طول می‌انجامد حال آن‌که برنامه تولید شده توسط الگوریتم ابرابتکاری مربوطه طی حداکثر ۳/۵ روز با یک رایانه‌ی معمولی قادر است ۵۰ جدول زمانی درسی متنوع تولید کند.

پانوشت‌ها

1. University Course Timetabling Problem
2. Heuristic & Metaheuristic Algorithms
3. Hyper-heuristic
4. Design of Experiments

منابع (References)

1. Daskalaki, S., Birbas, T. and Housos, E. "An integer programming formulation for a case study in university timetabling", *European Journal of Operational Research*, **153**(1), pp. 117-135 (2004).
2. Gozali, A.A., Kurniawan, B., Weng, W. and et al. "Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy", *IEEJ Transactions on Electrical and Electronic Engineering*, **15**(3), pp. 389-400 (2020).
3. Gary, M.R. and Johnson, D.S. "Computers and intractability: A guide to the theory of NP-Completeness", New York: W.H. Freeman and Company (1979).
4. Gülcü, A. and Akkan, C. "Robust university course timetabling problem subject to single and multiple disruptions", *European Journal of Operational Research*, **283**(2), pp. 630-46 (2020).
5. Hosseini Dehmiry, A. and Samadi, M.A. "A multiobjective programming model for university course timetabling problem", *Journal of Advanced Mathematical Modeling*, **12**(1), pp. 90-107 (In Persian) (1400/2021).
6. Alirezaee, M., Khalili, M. and Mansourzadeh, S. "University timetabling problem with mathematical two stage modeling", *Commercial Strategies*, **4**(1), pp. 87-96 (In Persian) (1399/2020).
7. Beigi, S. and Basiri, E. "Multi-Objective modeling of university course schedules considering a two-week periodicity: A case study", *Journal of Operational Research In Its Applications (Applied Mathematics)-Lahijan Azad University*, **18**(3), pp. 15-30 (10 Sep 2021).
8. Khoshnoodifar, M. and Fathi Vajargah, K. "Internationalization distance education curricula in iran higher education", *Journal of Technology of Education (JTE)*, **6**(2), pp. 87-104 (In Persian) (1391/2012).
9. Chen, M.C., Goh, S.L., Sabar, N.R. and et al. "A survey of university course timetabling problem: Perspectives, trends and opportunities", IEEE Access, **9**, pp. 106515-29 (27 Jul 2021).
10. Ryser-Welch, P. and Miller, J.F. "A review of hyper-heuristic frameworks", *In Proceedings of the Evo20 Workshop*, AISB (2014).
11. Monadjemi, S.A., Masoudian, S., Estaki, A. and et al. "Designing automated timetable for university courses using genetic algorithm", *Journal of Technology of Education (JTE)*, **4**(2), pp. 113-127 (In Persian) (1389/2010).
12. Ong, Y.S. and Keane, A.J. "Meta-lamarckian learning in memetic algorithms", *IEEE Transactions on Evolutionary Computation*, **8**(2), pp. 99-110 (2004).
13. Cowling, P., Kendall, G. and Soubeiga, E. "A hyper-heuristic approach to scheduling a sales summit", *In International Conference on the Practice and Theory of Automated Timetabling*, Springer, Berlin, Heidelberg, pp. 176-190 (2000).
14. Montazeri, M., Bahrololoum, A., Nezamabadi-pour, H. and et al. "Cooperating of local searches based hyper-heuristic approach for solving traveling salesman problem", *In Proceedings of the International Conference on Evolutionary Computation Theory and Applications*, pp. 24-26 (2011).

15. Macias-Escobar, T., Cruz-Reyes, L. and Dorronsoro, B., *A Study on the Use of Hyper-Heuristics Based on Meta-Heuristics for Dynamic Optimization*, In Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications, Springer, Cham pp. 295-314 (2021). DOI:10.1007/978-3-030-68776-2-18.

16. Pillay, N. and Qu, R., *Hyper-Heuristics: Theory and Applications*, Switzerland: Springer (2018).

17. Qu, R. and Burke, E.K. "Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems", *Journal of the Operational Research Society*, **60**(9), pp. 1273-1285 (2009).

18. Kalender, M., Kheiri, A., Özcan, E. and et al. "A greedy gradient-simulated annealing hyper-heuristic for a curriculum-based course timetabling problem", *In 2012 12th UK Workshop on Computational Intelligence (UKCI)*, IEEE, pp. 1-8 (2012).

19. Soria-Alcaraz, J.A., Ochoa, G., Swan, J. and et al. "Effective learning hyper-heuristics for the course timetabling problem", *European Journal of Operational Research*, **238**(1), pp. 77-86 (2014).

20. Soria-Alcaraz, J.A., Özcan, E., Swan, J. and et al. "Iterated local search using an add and delete hyper-heuristic for university course timetabling", *Applied Soft Computing*, **1**(40), pp. 581-93 (2016).

21. Soria-Alcaraz, J.A., Ochoa, G., Sotelo-Figueroa, M.A. and et al., *Iterated VND Versus Hyper-Heuristics: Effective and General Approaches to Course Timetabling*, In Nature-Inspired Design of Hybrid Intelligent Systems Springer, Cham, pp. 687-700 (2017).

22. Atashpaz-Gargari, E. and Lucas, C. "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition", Paper Presented at the 2007 IEEE Congress on Evolutionary Computation, IEEE (2007).

23. Rezaeipanah, A., Matoori, S.S. and Ahmadi, G. "A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search", *Applied Intelligence*, **51**(1), pp. 467-92 (2021 Jan).

24. Zhu, K., Li, L. and Li, M. "A survey of computational intelligence in educational timetabling", *International Journal of Machine Learning and Computing*, **11**(1), pp. 40-47 (1 Jan 2021).