Decisions & Operations Research

Volume 6, Issue 1, Spring 2021

Modified Approach to Optimize the University Examination Timetabling Problem

Habibeh Nazif [1,*], Khadijeh Ghaziyani [2]

[1]Department of Mathematics, Payame Noor University, Tehran, Iran.
[2]Department of Mathematics, Ayandegan Institute of Higher Education, Tonekabon, Iran.

Abstract

The timetable is the problem of placing particular resources due to constraints in a limited number of times lots and space, in order to satisfy a set of goals that is used to a variety of problems. Among these problems, one can point out the University Examination Timetabling Problem (UETP), which is the particular importance in educational problems. The university examination timetabling problem defined as the assignment of a certain set of exams to a fixed number of time slots and rooms, so that it meets all the hard constraints, also soft constraints are optimized as much as possible. This research presents a modified approach to optimize the incapacitated UETP. In this approach, a proposed Genetic Algorithm (GA) is modified by local search operators. These operators will make alterations to the timetable. This involves shifting or changing scheduled exams and thus greatly improve the ability of the algorithm to search. The efficiency of the proposed approach is compared with other techniques from literature using the Carter's benchmark. The computational results show that this approach is quite effective and competitive in improving the solutions and is able to produce better solutions in most of the datasets compared with other algorithms.

Keywords: Timetabling, University examination timetabling problem, Genetic algorithm, Local search.

Paper Type: Original

Received: 28/08/2020 Reviewed: 01/11/2020 Revised: 13/01/2021 Accept: 19/01/2021

Citation:

Nazif, H., & Ghaziyani, Kh. (2021). Modified approach to optimize the university examination timetabling problem. *Decisions & operations research*, *6*(1), 132-145

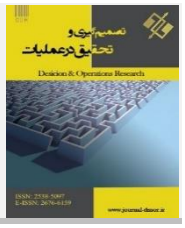
*** Corresponding Author**

Email Address: habibenazif@yahoo.com

DOI: 10.22105/dmor.2021.245798.1209

www.journal-dmor.ir

نشریه علمی
تصمیم‌گیری و تحقیق در عملیات
Decisions and Operational Research
دوره ۶، شماره (۱)، بهار ۱۴۰۰



رویکرد اصلاحی برای بهینه‌سازی مسئله جدول زمانی امتحانات دانشگاهی

حبیبه نظیف[۱،*]، خدیجه قاضیانی[۲]
[۱] گروه ریاضی، دانشگاه پیام نور، تهران، ایران.
[۲] گروه ریاضی، موسسه آموزش عالی آیندگان، تنکابن، ایران.

چکیده

جدول زمانی، مسئله قرار دادن منابع خاص با توجه به محدودیت‌ها در تعداد محدودی بازه‌ی زمانی و مکانی به منظور ارضا مجموعه‌ای از اهداف است که در مسائل متنوعی کاربرد دارد. از جمله این مسائل، می‌توان به مسئله جدول زمانی امتحانات دانشگاهی *(UETP)* اشاره کرد که از اهمیت خاصی در مسائل آموزشی برخوردار است. مسئله جدول زمانی امتحانات دانشگاهی در واقع تخصیص مجموعه‌ای معین از امتحانات به تعداد ثابتی از بازه‌های زمانی و اتاق‌ها می‌باشد، به‌طوری‌که تمام محدودیت‌های سخت را برآورده کند، هم‌چنین محدودیت‌های نرم نیز تا حد ممکن بهینه شوند. این تحقیق به ارائه و بررسی یک رویکرد اصلاحی برای بهینه‌سازی *UETP* بدون ظرفیت می‌پردازد. در این رویکرد یک الگوریتم ژنتیک *(GA)* پیشنهادی به وسیله عملگرهای جستجوی محلی اصلاح می‌شود. این عملگرها تغییراتی که مستلزم انتقال یا تعویض امتحانات زمان‌بندی شده است را در جدول زمانی ایجاد کرده و در نتیجه توانایی جستجوی الگوریتم را تا حد زیادی بهبود می‌بخشند. با استفاده از مجموعه مسائل نمونه کارتر، کارآیی و مؤثر بودن رویکرد پیشنهادی در مقایسه با دیگر روش‌های موجود بررسی می‌شود. نتایج محاسبات نشان می‌دهد که این رویکرد در بهبود جواب‌ها کاملاً مؤثر و رقابتی بوده و قادر است در بیشتر نمونه‌ها، جواب‌های بهتری در مقایسه با الگوریتم‌های دیگر تولید کند.

واژه‌های کلیدی: جدول زمانی، مسئله جدول زمانی امتحانات دانشگاهی، الگوریتم ژنتیک، جستجوی محلی.

نوع مقاله: پژوهشی

دریافت: ۱۳۹۹/۰۶/۰۷ داوری: ۱۳۹۹/۰۸/۱۱ بازنگری: ۱۳۹۹/۱۱/۲۴ پذیرش: ۱۳۹۹/۱۱/۳۰

۱- مقدمه

زمان‌بندی ابزاری در جهت استفاده بهینه از منابع و امکانات در دسترس است و موجب افزایش بازدهی، بهره‌برداری مناسب از ظرفیت‌ها و کاهش زمان مورد نیاز به منظور تکمیل کارها و درنهایت سودآوری می‌شود. در برنامه‌های کلان کشورهای توسعه یافته یکی از بخش‌هایی که در نیل به تحقق برنامه‌ها و اهدافشان نقش به‌سزا و مؤثری ایفا کرده، نظام آموزشی است. با توجه به نقش کلیدی نظام آموزشی دانشگاهی در هر جامعه، یک زمان‌بندی مناسب سبب ارتقای کیفیت آموزشی و رضایت‌مندی استفاده‌کنندگان از این زمان‌بندی می‌شود. جدول زمانی[۱] نوع خاصی از مسئله زمان‌بندی است که در مسائلی با دامنه‌های وسیع و متنوع کاربرد دارد که از آن جمله می‌توان مسائل آموزشی، مسابقات

[۱] Timetabling

* نویسنده مسئول
آدرس رایانامه: habibenazif@yahoo.com
شناسه دیجیتال: DOI: 10.22105/dmor.2021.245798.1209

ورزشی، مسائل حمل‌ونقل، برنامه کاری کارکنان، زمان‌بندی جلسات و زمان‌بندی فرآیندهای تولیدی را نام برد. مسئله زمان‌بندی در دانشگاه‌ها به دو دسته جدول زمانی برای امتحانات و زمان‌بندی دروس تقسیم می‌شود.

با توجه به این‌که در پایان ترم تحصیلی هر دانشگاه، امتحانات پایان‌ترم متناسب با دروس ارائه شده در هر رشته برگزار می‌شود، بنابراین جدول زمانی امتحانات دانشگاهی می‌تواند به‌عنوان یک مسئله تخصیص مطرح شود. کارتر و لاپورته[۱] (۱۹۹۵) مسئله جدول زمانی امتحانات دانشگاهی[۲] *(UETP)* را این‌طور تعریف می‌کنند: تخصیص امتحانات به یک تعداد از بازه‌های زمانی موجود، به‌طوری که هیچ تداخلی وجود نداشته باشد. از آن‌جایی که هر دانشگاه و موسسه علمی شرایط و ضوابط مخصوص به خود را داراست، یافتن یک راه‌حل واحد برای این مسئله بسیار دشوار است.

معمولاً محدودیت‌ها در این نوع مسائل به دو دسته سخت[۳] و نرم[۴] تقسیم می‌شوند. محدودیت‌های سخت، محدودیت‌هایی هستند که حتماً باید برآورده شوند و شدنی بودن جواب را تضمین می‌کنند. محدودیت‌های نرم بیان کننده‌ی مطلوبیت و ترجیحات مسئله هستند که برای کیفیت بهتر جدول زمانی در نظر گرفته می‌شوند و لزومی ندارد که همانند محدودیت‌های سخت به‌طور کامل برآورده شوند. بنابراین، برای به‌دست‌آوردن یک جدول زمانی امکان‌پذیر (شدنی) باید تمام محدودیت‌های سخت برآورده شده، هم‌چنین محدودیت‌های نرم نیز تا حد ممکن بهینه شوند. مجموعه‌ای از محدودیت‌هایی که باید برآورده شوند، معمولاً از یک موسسه به موسسه دیگر متفاوت است (بورک و همکاران[۵] ، ۱۹۹۵a).

از جمله محدودیت‌های سخت می‌توان به موارد زیر اشاره کرد:

- *هیچ دانشجویی نباید دارای بیش از یک امتحان در یک بازه‌ی زمانی مشخص باشد.*
- *هر امتحان تنها باید یک مرتبه برنامه‌ریزی شود .*
- *ظرفیت اتاق نباید کمتر از تعداد دانشجویان شرکت کننده در امتحانی باشد که به این اتاق تخصیص داده شده است.*
- *در پایان فرایند زمان‌بندی، نباید امتحان زمان‌بندی نشده‌ای باقی بماند.*
- *دانشجویان با امتحانات متوالی در یک روز، باید به یک اتاق تخصیص داده شوند.*

هم‌چنین، نمونه‌هایی از محدودیت‌های نرم عبارتند از:

- *بازه‌ی زمانی امتحانات حتی‌الامکان کوتاه باشد.*
- *تعداد دانشجویانی که امتحانات متوالی دارند، حداقل باشد.*
- *امتحانات حتی‌الامکان با فاصله مناسبی در برنامه امتحانی هر دانشجو پخش شده باشد.*

مسئله جدول زمانی امتحانات دانشگاهی را می‌توان به دو دسته، مسائل با ظرفیت[۶] و بدون ظرفیت[۷] طبقه‌بندی کرد. در *UETP* با ظرفیت، ظرفیت اتاق به عنوان یک محدودیت سخت مورد توجه قرار می‌گیرد، درحالی که در *UETP* بدون ظرفیت، ظرفیت اتاق درنظر گرفته نمی‌شود. تمرکز ما در این تحقیق، *UETP* بدون ظرفیت است.

مسئله جدول زمانی امتحانات دانشگاهی یک مسئله بهینه‌سازی ترکیبی کلاسیک است که اساس کار دانشگاه‌های سراسر دنیا قرار گرفته است. این مسئله یک چارچوب مشترک را برای اختصاص امتحانات در بازه‌های زمانی سازگار ارائه می‌دهد. اهمیت و ضرورت تحقیق در *UETP* هر سال با افزایش ثبت‌نام کنندگان در دانشگاه‌ها پررنگ‌تر شده و گروه آموزشی دانشگاه‌ها نیاز به حل مسئله زمان‌بندی در ارائه دروس یا امتحانات به‌وسیله‌ی اختصاص این رویدادها به بازه‌های زمانی خاص، اتاق‌ها و مدرس‌ها دارند. درحال حاضر در اغلب

---

[۱] Carter and Laporte
[۲] University Examination Timetabling Problem
[۳] Hard Constraint
[۴] Soft Constraint
[۵] Burke et al.
[۶] Capacitated
[۷] Un-Capacitated

دانشگاه‌ها زمان‌بندی امتحانات به‌صورت دستی انجام می‌شود. از آن‌جایی که زمان‌بندی دستی وقت‌گیر است و لزوماً تمام خواسته‌های دانشجویان را به بهترین نحو برآورده نمی‌کند، استفاده از روش کامپیوتری که در زمان بسیار کمتر، جواب خوبی را فراهم می‌آورد، مناسب به‌نظر می‌رسد.

تحقیق و جستجو درباره زمان‌بندی ماشینی به دهه‌ی ۶۰ میلادی برمی‌گردد. پس‌از آن محققان به بررسی روش‌های عمومی‌تر پرداختند، مانند روش‌های برنامه‌ریزی عدد صحیح، جریان شبکه، رنگ آمیزی گراف و .... در دو دهه‌ی اخیر، تلاش‌ها بر روی روش‌های ابتکاری و فراابتکاری متمرکز شده است.

با توجه به کارایی روش‌های فراابتکاری در حل بهینه مسائل جدول زمانی، این تحقیق به معرفی یک رویکرد اصلاحی برای بهینه‌سازی مسئله جدول زمانی امتحانات دانشگاهی بدون ظرفیت می‌پردازد. این رویکرد از اصلاح الگوریتم ژنتیک[۱] *(GA)* پیشنهادی به کمک عملگرهای جستجوی محلی به‌وجود آمده و می‌تواند توانایی *GA* را تا حد زیادی در به‌دست‌آوردن جواب‌های با کیفیت بهبود بخشد. این عملگرها به انتقال یک امتحان زمان‌بندی شده در یک بازه‌ی زمانی جدید یا تعویض دو امتحان زمان‌بندی شده بین دو بازه‌ی زمانی مختلف اشاره می‌کنند.

۲- پیشینه تحقیق

روش‌های اولیه برای حل *UETP* از سال ۱۹۶۰ با به‌کارگیری مفهوم مرتب‌سازی ابتکاری به‌طور وسیع مورد استفاده قرار گرفتند، در این روش‌ها ابتدا امتحانات با استفاده از روش‌های ابتکاری مرتب شده و سپس به‌صورت متوالی در دوره‌های زمانی شدنی طوری زمان بندی می‌شوند که هیچ امتحانی با امتحانات دیگر تداخل نداشته باشد. کارتر و همکاران[۲] (۱۹۹۶) چند استراتژی توالی به شرح زیر ارائه دادند:

- *مرتب‌سازی تصادفی[۳] (RO): امتحانات به‌طور تصادفی برای زمان‌بندی انتخاب می‌شوند.*
- *بالاترین درجه[۴] (LD): امتحانات براساس تعداد تعارضی که با امتحانات دیگر دارند، زمان‌بندی می‌شوند، به طوری که ابتدا امتحانات با تعداد تعارض بالاتر زمان‌بندی می‌شوند.*
- *بیشترین ثبت‌نام‌ها[۵] (LE): امتحانات براساس تعداد دانشجویان ثبت‌نام شده زمان‌بندی می‌شوند. امتحانات با بالاترین تعداد دانشجویان ثبت‌نام شده، در ابتدا زمان‌بندی می‌شوند.*
- *بیشترین درجه‌ی موزون[۶] (LWD): این روش همانند روش LD است، اما ابتدا امتحانات برحسب تعداد دانشجویان ثبت‌نامی وزن‌دهی شده، سپس امتحانات با بیشترین تعداد از تعارض دانشجویان، در ابتدا زمان‌بندی می‌شوند.*
- *درجه‌ی اشباع[۷] (SD): امتحانات بعدی براساس تعداد بازه‌های زمانی موجود برای امتحانات در زمان جاری، زمان‌بندی می‌شوند. امتحانات با کمترین تعداد از بازه‌های زمانی موجود، در ابتدا زمان‌بندی می‌شوند.*

بورک و همکاران[۸] (۱۹۹۵b) الگوریتم ممتیک را برای حل مسئله جدول زمانی امتحانات دانشگاهی بررسی نمودند. آن‌ها هر جواب در جمعیت را به‌عنوان یک تعداد از مم‌ها نشان داده و برای ساخت جمعیت اولیه از یک چرخ رولت موزون استفاده کردند که بازه‌های زمانی را برای زمان‌بندی هر امتحان انتخاب می‌کرد. در این تحقیق، یک مجموعه از مسائل نمونه جدید برای حل *UETP* معرفی شدند.

---

۱ Genetic Algorithm
۲ Carter et al.
۳ Random Ordering
۴ Largest Degree
۵ Largest Enrolment
۶ Largest Weighted Degree
۷ Saturation Degree
۸ Burke et al.

چوو فانگ[۱] (۱۹۹۹) با اختصاص‌دادن ۱۰ امتحان به ۱۲ بازه‌ی زمانی (سه روز) و ۵۰ دانشجو، کارایی یک الگوریتم ژنتیک و یک جستجوی ممنوع را با هم مقایسه کردند. نتایج نشان دادند که جستجوی ممنوع جواب‌های بهتری در مقایسه با الگوریتم ژنتیک تولید می‌کند.

کیسی و تامسون[۲] (۲۰۰۲) یک رویکرد دومرحله‌ای را در روش جستجوی انطباقی تصادفی حریصانه برای حل *UETP* معرفی کردند. در مرحله نخست، جواب اولیه با استفاده از روش حریصانه که به‌وسیله مرتب‌کردن امتحانات براساس یکی از ابتکاری‌های سطح پایین *(LD, LWD, LE, SD)* به‌وجود آمده، تولید می‌شود. در مرحله دوم از روش تبرید شبیه‌سازی شده برای کمینه کردن هزینه محدودیت های نرم استفاده می‌شود. پس از آزمایش الگوریتم پیشنهادی بر روی مجموعه مسائل نمونه، نتایج نشان دادند که روش ابتکاری *SD* قادر است بهترین جدول زمانی را در مقایسه با دیگر روش‌های ابتکاری سطح پایین تولید کند.

آسمونی و همکاران[۳] (۲۰۰٤) با انجام اصلاحاتی در الگوریتم ارائه شده توسط کارتر و همکاران (۱۹۹٦)، ترکیبی از دو روش مرتب سازی ابتکاری را بدون در نظر گرفتن سه روش دیگر *(LD, LE, SD)*، به‌صورت هم‌زمان برای تمام امتحانات به‌کار بردند. آن‌ها برای ارزیابی امتحانات مرتب‌شده از توابع فازی استفاده کردند.

عبدالله و بورک[٤] (۲۰۰٦) یک رویکرد مبتنی بر نمایش یک گراف بهبود یافته از مجاورت جواب را برای مسائل جدول زمانی امتحانات دانشگاهی با ظرفیت و بدون ظرفیت توسعه دادند. نتایج نشان دادند که روش پیشنهادی آن‌ها در برخی موارد از بهترین جواب‌های موجود پیشی گرفته است.

آسمونی[۵] (۲۰۰۸) روش‌های فازی را برای حل مسائل زمان‌بندی دروس دانشگاهی و زمان‌بندی امتحانات ارائه داد. او در مرحله‌ی ساخت برای تولید جدول زمانی، روش‌های فازی را با روش‌های مرتب‌سازی ابتکاری چندگانه ترکیب کرد. ابوناصر و همکاران[٦] (۲۰۱۰) یک الگوریتم بهینه‌سازی کلونی مورچگان ترکیبی با جستجوی محلی کامل را برای حل *UETP* بسط و گسترش دادند. الگوریتم پیشنهادی آن‌ها نتایج قابل قبولی را با استفاده از مجموعه مسائل نمونه کارتر به‌دست آورد.

پیلای و بانژاف[۷] (۲۰۱۰) با استفاده از الگوریتم ژنتیک، رویکردهای دومرحله‌ای را برای حل *UETP* به‌کار گرفتند. آن‌ها روش ابتکاری سطح پایین جدیدی به نام بالاترین هزینه[۸] (*HC*) معرفی کرده و متوجه شدند که ترکیب روش‌های ابتکاری *SD* با *HC* و استفاده از *LWD* زمانی‌که مقایسه پارتو از امتحان انجام می‌شود، بهترین نتیجه را در مرحله اول ارائه می‌دهد. در دومین مرحله، الگوریتم ژنتیک با هدف کمینه‌کردن هزینه محدودیت‌های نرم، جواب‌های یافته‌شده در مرحله اول را با به‌کارگیری عملگر جهش تصحیح می‌کند. پیلای[۹] (۲۰۱۲) بر روی اثر نمایش‌های متفاوت بر کیفیت زمان‌بندی مطالعه کرد و الگوریتم تکاملی را با نمایش‌های متفاوت اجرا کرد.

عبدالرحمن و همکاران[۱۰] (۲۰۱٤) برای حل *UETP* ، روی یک استراتژی تجزیه انطباقی تحقیق کردند که امتحانات را در دو مجموعه سخت و راحت گروه‌بندی می‌کرد. الگوریتم پیشنهادی روی مسائل نمونه کارتر اجرا شده و نتایج حاصل قابل مقایسه با رویکردهای دیگر می‌باشد. اسحاق و همکاران[۱۱] (۲۰۱٦) یک الگوریتم ژنتیک ترکیبی را برای حل *UETP* بدون ظرفیت پیشنهاد داده و در آن از سه روش بهینه‌سازی استفاده کردند. آن‌ها آزمایش‌محاسباتی متفاوتی برای طراحی الگوریتم پیشنهادی انجام داده و روش خود را روی مجموعه مسائل نمونه کارتر ارزیابی نمودند.

---

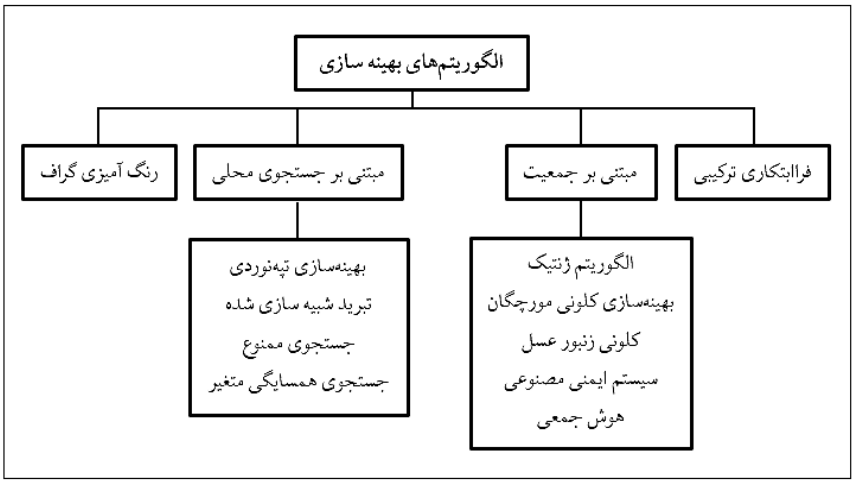
۱ Chu and Fang
۲ Casey and Thompson
۳ Asmuni et al.
٤ Abdullah and Burke
۵ Asmuni
٦ Abounacer et al.
۷ Pillay and Banzhaf
۸ Highest Cost
۹ Pillay
۱۰ Abdul-Rahman et al.
۱۱ Ishak et al.

لیته و همکاران[۱] (۲۰۱۸) یک الگوریتم ممتیک سلولی برای حل مسئله جدول زمانی پیشنهاد دادند. الگوریتم های تکاملی سلولی، فراابتکاری مبتنی بر جمعیت هستند. آن‌ها با الگوریتم های غیرسلولی از این‌نظر تفاوت دارند که جمعیت در یک ساختار سلولی سازمان یافته است و امکان واقعی سازی جمعیت را فراهم می‌نماید که همین امر به بهبود تنوع جمعیت کمک می‌کند. آن‌ها الگوریتم تکاملی سلولی پیشنهادی را با رویکرد جستجوی محلی با پذیرش یک آستانه ترکیب کردند. الگوریتم پیاده‌سازی شده از ترکیب ژنتیکی امکان پذیر و اپراتورهای جستجوی محلی استفاده کرده، بنابراین اکتشاف را به فضای جواب شدنی محدود می‌کند. هم‌چنین در این تحقیق، اثر پذیرش آستانه مورد استفاده در الگوریتم ترکیبی برای مسئله جدول زمانی بررسی شد. نتایج نشان دادند که برای تنظیم مجدد سخت‌ترین امتحانات در دوره های بهتر، میزان کاهش آستانه کمی مورد نیاز است. هم‌چنین، این امکان را می‌دهد که مجموعه‌ای آسان از امتحانات در دوره‌های خوب نیز قرار گیرد.

اسی و همکاران[۲] (۲۰۱۸) یک الگوریتم ژنتیک را برای رنگ آمیزی گراف متناظر با مسئله جدول زمانی تجزیه و تحلیل کردند. در رویکرد آن‌ها هر جواب نشدنی به یک جواب شدنی ترمیم می‌شود. روش آن‌ها در جاوا پیاده سازی شد و نتایج حاصل از آزمایشات بر اساس محدودیت‌ها و نیازهای مشخص شده، بهبود جواب اولیه را نشان می‌دهد.

لیته و همکاران[۳] (۲۰۱۹) نوع جدیدی از الگوریتم تبرید شبیه سازی شده را برای حل مسئله جدول زمانی امتحانات دانشگاهی ارائه دادند. در این روش، امتحانی که برای برنامه ریزی انتخاب شده فقط درصورتی منتقل و حرکت مربوط به آن ارزیابی می‌شود اگر آن امتحان در سطوح دمایی که قبل از آن انجام می‌شود، حرکت قابل قبولی داشته باشد. رویکرد پیشنهادی با استفاده از دومین مجموعه مسائل نمونه بین المللی جدول زمانی مورد آزمایش قرار گرفت. ماکلاسون و همکاران[۴] (۲۰۱۹) یک الگوریتم ترکیبی از رنگ آمیزی گراف برای تولید جواب شدنی اولیه و الگوریتم سیل بزرگ با رویکرد متفاوت، یعنی ابرابتکاری به‌جای فراابتکاری را برای حل یک مسئله زمان‌بندی دانشگاه در دنیای واقعی پیشنهاد کردند. نتیجه این مطالعه نشان داد که به‌طور کلی الگوریتم پیشنهادی برای حل مسئله جدول زمانی دانشگاه‌های واقعی بسیار موثر است. تنها در حدود ۳۰۰ ثانیه، الگوریتم پیشنهادی می‌تواند یک جدول زمانی تولید کند که بسیار بهتر از جدول زمانی است که به‌صورت دستی تهیه می‌شود. این مطالعه هم‌چنین نشان داد که در چارچوب ابرابتکاری، الگوریتم سیل بزرگ از الگوریتم‌های تبرید شبیه‌سازی شده و تپه نوردی بهتر عمل می‌کند. علاوه بر الگوریتم، سهم اصلی این مطالعه یک مجموعه داده جدید عمومی برای بررسی مسئله جدول زمانی است.

شکل ۱ چندین الگوریتم بهینه‌سازی را برای حل مسائل جدول زمانی امتحانات دانشگاهی نشان می‌هد که توسط محققین در دهه‌های گذشته روی مسائل نمونه کارتر اعمال شده است.



شکل ۱- الگوریتم‌های بهینه‌سازی.

Figure 1- Optimization algorithms.

[۱] Leite et al.

[۲] Assi et al.

[۳] Leite et al.

[۴] Muklason et al.

۳- رویکرد اصلاحی

در این بخش جزئیات رویکرد اصلاحی، که به منظور بهبود جواب ها یک الگوریتم ژنتیک پیشنهادی را به‌وسیله عملگرهای جستجوی محلی اصلاح می‌کند، به تفصیل بیان و شرح داده می‌شود.

ایده‌ی استفاده از الگوریتم ژنتیک، نخستین بار توسط هالند[۱] در دهه‌ی ۶۰ میلادی مطرح شد. امروزه الگوریتم ژنتیک به‌عنوان یکی از الگوریتم‌های تکاملی در زمینه‌های مختلف مورد استفاده قرار می‌گیرد و به روشی رایج برای بهینه‌سازی تبدیل شده است. الگوریتم ژنتیک با یک جمعیت اولیه تصادفی از کروموزوم‌ها آغاز می‌شود و در نسل‌های بعدی تکرار می‌گردد. در طول هر نسل، برازندگی هر فرد (کروموزوم) توسط تابع برازندگی اندازه‌گیری شده و مناسب‌ترین افراد برای حضور در نسل بعد انتخاب می‌شوند. جمعیت جدید با اعمال عملگرهای ژنتیکی شامل انتخاب، ترکیب، جهش و دیگر عملگرهای احتمالی به‌وجود می‌آید. سپس این جمعیت جدید که معمولاً برازندگی بیشتری دارد، جایگزین جمعیت پیشین می‌شود و این فرآیند آن‌قدر تکرار می‌گردد تا شرایط خاتمه برای رسیدن به بهترین جواب محقق شود.

۳-۱- ساختار الگوریتم ژنتیک

عملگرهای ژنتیک در *GA* پیشنهادی برای بهینه‌سازی *UETP* بدون ظرفیت به‌صورت زیر درنظر گرفته می‌شوند:

مهم‌ترین بخش یک *GA* برای حل مسئله بهینه‌سازی، انتخاب نمایش مناسب ژن برای مسئله است. **شکل ۲** نمونه‌ای از نمایش عدد صحیح را برای *UETP* تشریح می‌کند که در آن بازه‌های زمانی $t_i \ (i = 1,2,...,10)$ به امتحانات $e_j \ (j = 1,2,...,20)$ اختصاص داده شده‌اند.

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
e_3	e_1	e_2	e_{10}	e_9	e_8	e_{12}	e_{19}	e_{16}	e_{13}
e_5	e_4	e_6		e_{17}	e_{11}	e_{15}		e_{20}	e_{14}
e_7	e_{18}								

شکل ۲- نمایش ژنی GA برای UETP بدون ظرفیت.

Figure 2- GA gene representation for uncapacitated UETP.

برای جمعیت آغازین از ترکیب روش‌های ابتکاری سطح پایین *HC* و *SD* استفاده شده و جدول زمانی تولید شده است.

از آن‌جایی‌که برآورده شدن تمام محدودیت‌های نرم امکان‌پذیر نیست، لذا می‌توان کیفیت یک جدول زمانی را با کمینه‌کردن این نقض‌ها بهبود داد. در این تحقیق، تابع هزینه مجاورت[۲] ارائه شده توسط کارتر و همکاران (۱۹۹۶) برای ارزیابی کیفیت جدول زمان‌بندی درنظر گرفته شده است. این تابع با هدف پخش زمان‌بندی امتحانات هر دانشجو توسعه داده شده است. قابل ذکر است که، در پیاده سازی تابع ارزیابی این محدودیت سخت که هیچ دانشجویی نمی‌تواند بیش از یک امتحان در یک زمان مشترک داشته باشد نیز برآورده می شود. علاوه براین، جواب باید به روشی توسعه داده شود که پخش شدن امتحانات هر دانشجو را افزایش دهد، به‌طوری که دانشجویان مابین امتحاناتشان به اندازه کافی وقت داشته باشند. تابع هزینه مجاورت به‌صورت زیر تعریف می‌شود:

$$C = \frac{\sum_{i=1}^{M-1} \sum_{j=i+1}^{M} proximity\left(t_{e_i}, t_{e_j}\right) . com_{ij}}{N}, \qquad (۱)$$

به‌طوری که M تعداد کل امتحانات، N تعداد کل دانشجویان و com_{ij} تعداد دانشجویان مشترک بین امتحانات e_i و e_j را نشان می‌دهد هم‌چنین

[۱] Holland

[۲] Proximity Cost Function

$$proximity\ (t_{e_i}, t_{e_j}) = \begin{cases} 2^{(5-\Delta t)} & if \quad 1 \le |\Delta t| \le 5 \\ o & otherwise \end{cases} . \qquad (۲)$$

معادله (۲) مقدار مجاورت بین دو بازه‌ی زمانی را محاسبه می‌کند، به‌طوری‌که امتحانات e_i و e_j به‌ترتیب در بازه‌های زمانی t_{e_i} و t_{e_j} زمان‌بندی شده‌اند و $i \neq j$, $\Delta t = \left| t_{e_i} - t_{e_j} \right|$. هدف این تحقیق، کمینه کردن مقدار **معادله (۱)** است.

برای انتخاب والدین از میان جمعیت، مکانیزم انتخاب چرخ رولت درنظرگرفته شده است.

هم‌چنین، از عملگر تقاطع یک نقطه‌ای برای تولید فرزندان ا ستفاده شده ا ست. هرچند ممکن ا ست این عملگر جواب‌های ن شدنی تولید کند که در این‌صورت باید فرزندان را برای به‌دست‌آوردن جواب‌های شدنی اصلاح کرد.

به‌طور مثال، **شکل ۳** که در آن هر والد شامل ۱۰ امتحان زمان‌بندی شده در ۵ بازه‌ی زمانی ا ست را ملاحظه کنید. ابتدا یک نقطه‌ی تقاطع (با نماد "|") به‌صورت تصادفی انتخاب می‌شود (در این‌جا نقطه‌ی تقاطع بین ژن ۳ و ٤ انتخاب شده است)، سپس قسمت اول والدین ۱ و ۲ در فرزندان جدید ۱ و ۲ کپی شده و بخش باقی‌مانده از فرزندان ۱ و ۲ به‌ترتیب با قسمت دوم والدین ۲ و ۱ پر می شوند. **شکل ٤** فرزندان تولید شده را نشان می‌دهد.

	t_1	t_2	t_3	t_4	t_5
والد ۱	e_8	e_3	e_1	e_9	e_2
	e_{10}	e_5	e_6	e_7	
		e_4			

	t_1	t_2	t_3	t_4	t_5
والد ۲	e_5	e_3	e_1	e_8	e_6
	e_7	e_9	e_4	e_2	
	e_{10}				

شکل ۳- والدین انتخاب‌شده.

Figure 3- Selected parents.

در نتیجه

	t_1	t_2	t_3	t_4	t_5
فرزند ۱	e_8	e_3	e_1	e_8	e_6
	e_{10}	e_5	e_6	e_2	
		e_4			

	t_1	t_2	t_3	t_4	t_5
فرزند ۲	e_5	e_3	e_1	e_9	e_2
	e_7	e_9	e_4	e_7	
	e_{10}				

شکل ٤- فرزندان تولید شده.

Figure 4- Generated offspring.

با توجه به **شکل ۳** ملاحظه می‌کنید که فرزندان تولید شده جواب‌های نشدنی می‌باشند، زیرا در فرزند ۱ امتحانات e_7 و e_9 و در فرزند ۲ امتحانات e_6 و e_8 زمان‌بندی نشده‌اند، درحالی که تمام امتحانات باید در جدول زمان‌بندی شوند. هم‌چنین، هر امتحان فقط باید یک‌بار زمان‌بندی شود، در صورتی که در فرزندان تولید شده امتحانات تکراری مشاهده می شود. بنابراین باید در بخش دوم فرزندان، امتحانات تکراری از جدول زمان‌بندی حذف شده و امتحانات باقی‌مانده‌ی بخش دوم، مجدد در بازه‌های زمانی قبلی براساس شدنی‌بودن زمان‌بندی شوند. سپس، امتحانات زمان‌بندی نشده طوری در جدول زمان‌بندی وارد شوند که جواب‌های شدنی حاصل شود. **شکل ۵** یک نمونه از اصلاح فرزندان که منجر به جواب‌های شدنی است را نشان می‌دهد.

	t_1	t_2	t_3	t_4	t_5
فرزند ۱	e_8	e_3	e_1	e_2	e_9
	e_{10}	e_5	e_6	e_7	
		e_4			

	t_1	t_2	t_3	t_4	t_5
فرزند ۲	e_5	e_3	e_1	e_2	e_8
	e_7	e_9	e_4	e_6	
	e_{10}				

شکل ۵- فرزندان اصلاح شده.
Figure 5- Modified offspring.

پس‌از اعمال عملگر تقاطع به‌منظور اجتناب از به‌دام افتادن در بهینه محلی و در نتیجه همگرایی زودرس، هم‌چنین ایجاد تنوع و گوناگونی در جمعیت از عملگر جهش استفاده می‌شود. در این تحقیق، از جهش جایگذاری استفاده شده است که در آن ابتدا یک ژن به‌طور تصادفی در فرزند انتخاب و حذف شده، سپس ژن در موقعیتی که به‌صورت تصادفی انتخاب شده، جایگذاری می‌شود.

به‌علاوه، با استفاده از استراتژی جایگزینی نخبه‌گرا جمعیت جدید جایگزین جمعیت پیشین می‌گردد. روش نخبه‌گرایی برای جلوگیری از حذف افراد با برازندگی بالا استفاده می‌شود، لذا این رویکرد باعث افزایش کارایی الگوریتم ژنتیک می‌گردد.

۲-۳- جستجوی محلی

الگوریتم ژنتیک پیشنهادی به کمک عملگرهای جستجوی محلی (انتقال امتحان[۱] (*ShE*) و تعویض امتحان[۲] (*ChE*)) که برای انتخاب جمعیت جدید معرفی شده‌اند، اصلاح می شود، به‌طوری که کیفیت جواب‌های به‌دست‌آمده بهبود می‌یابد. به این‌ترتیب، بعداز اعمال عملگر جهش، یکی از عملگرهای جستجوی محلی به صورت تصادفی انتخاب شده و اجرا می شود و یک جواب جدید را به‌دست می‌آورد. این عملگرها تغییراتی را در جدول زمانی ایجاد خواهند کرد، این تغییرات مستلزم انتقال یا تعویض امتحانات زمان‌بندی شده است. در ShE بر انتقال یک امتحان زمان‌بندی شده به یک بازه‌ی زمانی جدید تمرکز می شود، درحالی که *ChE* به تعویض دو امتحان زمان‌بندی شده، بین دو بازه‌ی زمانی متفاوت اشاره دارد. در ادامه به تشریح این عملگرها می‌پردازیم:

۱-۲-۳- انتقال امتحان (ShE)

درطی این رویکرد، الگوریتم امتحان از پیش زمان‌بندی شده‌ی e_k در بازه‌ی زمانی t_q را به یک موقعیت جدید، یعنی بازه‌ی زمانی t_p منتقل می‌کند، اگر این انتقال به‌طور هم‌زمان، کاهش در هزینه‌ی محدودیت‌های نرم و بهبود کیفیت کلی جدول زمانی را در پی داشته باشد. برای این منظور ماتریس تعارض جدید بازه‌ی زمانی/ امتحان تولید می‌شود، به‌طوری که عنصر $\sigma_{k,i}$، تعداد دانشجویان مشترک بین امتحان e_k با دیگر امتحانات زمان‌بندی‌شده در بازه‌ی زمانی t_i را نشان می‌دهد. لازم به ذکر است که، جدول زمانی شدنی است اگر $\sigma_{k,i} = 0 \ (k = 1,2,...,E)$، E تعداد کل امتحاناتی است که باید زمان‌بندی شوند. به‌عبارت‌دیگر، مقدار $\sigma_{k,i}$ از معادله زیر به‌دست می‌آید:

$$\sigma_{k,i} = \sum_{s=1}^{n_i} C_{k,e_{i,s}}, \qquad (۳)$$

به‌طوری که $e_{i,s}$، امتحان s ($s = 1,2,...,n_i$) زمان‌بندی شده در بازه‌ی زمانی t_i و n_i تعداد کل امتحانات در بازه‌ی زمانی t_i را نشان می‌دهد. به‌علاوه، $C_{k,e_{i,s}}$ تعداد دانشجویان مشترک بین امتحان e_k و امتحان $e_{i,s}$ زمان‌بندی شده در بازه‌ی زمانی t_i را نمایش می‌دهد.

معادلات جدیدی برای محاسبه هزینه‌ی انتقال امتحان e_k به بازه‌ی زمانی جدید t_p در جدول زمانی، معرفی می‌شوند. فرض کنید A هزینه‌ی جریمه کل از جدول زمانی باشد بدون این‌که امتحان e_k در جدول زمانی، زمان‌بندی شده باشد. **معادله (٤)**، هزینه‌ی جریمه کل Q از جدول زمانی اصلی را که در آن امتحان e_k در بازه‌ی زمانی t_q زمان‌بندی شده، به‌صورت زیر محاسبه می‌کند:

---

[۱] Shift Exam
[۲] Change Exam

$$Q = A + \sum_{m=1}^{5} 2^{5-m}(\sigma_{k,q+m} + \sigma_{k,q-m}). \qquad (۴)$$

$\sigma_{k,q+m}$ و $\sigma_{k,q-m}$ تعداد دانشجویان مشترک بین امتحان e_k با دیگر امتحانات زمان‌بندی‌شده به ترتیب در بازه‌های زمانی t_{q+m} و t_{q-m} را نشـــان می‌دهند که $q+m \le T,\ q-m \ge 1$ (T تعداد بازه‌های زمانی). امتحان e_k در بازه‌ی زمانی جدید t_p وارد می شود اگر و فقط اگر $\sigma_{k,p} = ۰$. در این‌صورت، هزینه‌ی جریمه کل Q_1 به‌وسیله‌ی **معادله (۵)** بیان می‌شود:

$$Q_1 = A + \sum_{m=1}^{5} 2^{5-m}(\sigma_{k,p+m} + \sigma_{k,p-m}). \qquad (۵)$$

$\sigma_{k,p+m}$ و $\sigma_{k,p-m}$ تعداد دانشجویان مشترک بین امتحان e_k با دیگر امتحانات زمان‌بندی‌شده به ترتیب در بازه‌های زمانی t_{p+m} و t_{p-m} را نشان می‌دهند که $p+m \le T,\ p-m \ge 1$. با تفاضل **معادله (٤)** از **معادله (۵)** داریم:

$$Q_1 - Q = \sum_{m=1}^{5} 2^{5-m}(\sigma_{k,p+m} + \sigma_{k,p-m} - \sigma_{k,q+m} - \sigma_{k,q-m}), \qquad (۶)$$

به‌طوری که $q+m \le T,\ q-m \ge 1,\ p+m \le T,\ p-m \ge 1$.

اگر $Q_1 - Q < ۰$ باشد، به‌عبارتی هزینه‌ی جریمه کل هنگامی‌که امتحان e_kدر بازه‌ی زمانی t_p زمان‌بندی شده کمتر از هزینه‌ی جریمه کل در بازه‌ی زمانی t_q باشد، آن‌گاه وقتی امتحان e_k در بازه‌ی زمانی t_p وارد می‌شود، منجر به کاهش هزینه‌ی جریمه کل در تمام جدول زمانی می‌شود. این الگوریتم براساس اولین حرکت بهبودی اجرا شده و زمانی‌که یک جواب بهبود یافته پیدا شود، متوقف خواهد شد. اگر تمام امتحانات بررسی شدند و هیچ حرکت رو به بهبودی یافت نشد، این روش پایان خواهد یافت و فرض می‌شود که جدول زمانی بر اساس این روش بهینه است.

۲-۲-۳- تعویض امتحان (ChE)

در طی اجرای این روش، موقعیت‌های دو امتحان از دو بازه‌ی زمانی متفاوت در جدول زمانی تعویض می‌شـــود اگر منجر به کاهش هزینه محدودیت نرم شـود. ابتدا فرض کنید امتحانات e_i و e_j به ترتیب در بازه‌های زمانی t_p و t_q زمان‌بندی شـده باشـند. فرض کنید A^* هزینه‌ی جریمه کل از جدول زمانی با شد بدون این‌که امتحانات e_i و e_jدر جدول زمانی، زمان‌بندی شده با شند. **معادله (۷)**، هزینه‌ی جریمه کل R از جدول زمانی اصـــلی، که در آن امتحانات e_i و e_j به ترتیب در بازه‌های زمانی t_p و t_q زمان‌بندی شده‌اند را به‌صورت زیر محاسبه می‌کند:

$$R = A^* + \sum_{m=1}^{5} 2^{5-m}(\sigma_{i,p+m} + \sigma_{i,p-m} + \sigma_{j,q+m} + \sigma_{j,q-m}). \qquad (۷)$$

$\sigma_{i,p+m}$ و $\sigma_{i,p-m}$ تعداد دانشجویان مشترک بین امتحان e_i با دیگر امتحانات زمان‌بندی شده به ترتیب در بازه‌های زمانی t_{p+m} و t_{p-m} را نمایش داده، هم‌چنین $\sigma_{j,q+m}$ و $\sigma_{j,q-m}$ تعداد دانشجویان مشترک بین امتحان e_j با دیگر امتحانات زمان‌بندی‌شده به ترتیب در بازه‌های زمانی t_{q+m} و t_{q-m} را نشـان می‌دهند که $p+m \le T,\ p-m \ge 1,\ q+m \le T,\ q-m \ge 1$. اکنون فرض کنید این دو امتحان بتوانند تعویض شوند، یعنی $\sigma_{i,q} = \sigma_{j,p} = ۰$. در این صورت، هزینه‌ی جریمه کل R_1 بعداز این‌که امتحانات e_i و e_j به‌ترتیب در بازه‌های زمانی t_p و t_q زمان‌بندی شدند، به‌وسیله‌ی **معادله (۸)** بیان می‌شود:

$$R_1 = A^* + \sum_{m=1}^{5} 2^{5-m}(\sigma_{i,q+m} + \sigma_{i,q-m} + \sigma_{j,p+m} + \sigma_{j,p-m}). \qquad (۸)$$

با تفاضل **معادله (۷)** از **معادله (۸)** خواهیم داشت:

$$R_1 - R = \sum_{m=1}^{5} 2^{5-m}(\sigma_{i,q+m} + \sigma_{i,q-m} + \sigma_{j,p+m} + \sigma_{j,p-m} - \sigma_{i,p+m} - \sigma_{i,p-m} - \sigma_{j,q+m} - \sigma_{j,q-m}), \qquad (۹)$$

به‌طوری که $q+m \le T,\ q-m \ge 1,\ p+m \le T,\ p-m \ge 1$.

اگر $R_1 - R < 0$، لذا امتحانات e_i و e_j تعویض می‌شوند. الگوریتم بر اساس اولین حرکت بهبود یافته اجرا می‌شود و اگر بعد از این‌که تمام امتحانات بررسی شدند، هیچ حرکت بهبود یافته‌ای پیدا نشود، پس جدول زمانی بر اساس این روش بهینه است. به‌طور کلی، چارچوب رویکرد اصلاحی در **شکل ٦** شرح داده شده است.

جمعیت آغازین را با استفاده از ترکیب روش‌های ابتکاری سطح پایین *HC* و *SD* تولید کنید؛
مقدار برازندگی هر فرد در جمعیت را ارزیابی کنید؛
تا زمانی‌که شرط توقف برقرار نشده، تکرار کنید؛
با استفاده از انتخاب چرخ رولت والدین را از میان جمعیت انتخاب کنید؛
عملگر تقاطع یک نقطه‌ای را برای هر زوج به‌کار برده و مجموعه‌ای از فرزندان را به‌دست آورید؛
عملگر جهش جایگذاری را برای فرزندان به‌کار برده و مجموعه‌ای از فرزندان جدید را تعیین نمایید؛
مقدار برازندگی هر فرزند را ارزیابی کنید؛
یکی از عملگرهای جستجوی محلی را به تصادف انتخاب کرده و بر روی فرزندان به‌کار برید؛
از استراتژی جایگزینی نخبه‌گرا برای تولید جمعیت بعدی استفاده کنید؛
بهترین جواب شدنی را برگردانید.

شکل ٦- چارچوب رویکرد اصلاحی.
Figure 6- Framework of the modified approach.

۴- طراحی آزمایش‌های محاسباتی

برای بررسی کارایی عملکرد رویکرد اصلاحی پیشنهادی در مقایسه با الگوریتم‌های دیگر از مجموعه مسائل نمونه کارتر استفاده شده است (کارتر و همکاران، ۱۹۹٦). این مجموعه داده استاندارد شامل ۱۲ مسئله زمان‌بندی واقعی امتحانات است که از سه دبیرستان کانادایی، پنج دانشگاه کانادایی، یک دانشگاه بریتانیایی و یک دانشگاه از خاورمیانه جمع‌آوری شده است. **جدول ۱** مجموعه مسائل نمونه کارتر را نمایش می‌دهد که اطلاعات گوناگون و مهمی از آن به‌دست می‌آید.

الگوریتم پیشنهادی با استفاده از نرم‌افزار *MATLAB* روی کامپیوتر دوهسته‌ای ۲/٦٦ *GHz* با ٤ *GB* حافظه اجرا شده است. برای اندازه گیری و مقایسه کیفیت جواب‌ها از تابع هزینه مجاورت **معادله (۱)** استفاده می‌شود. لازم به‌ذکر است که در رویکرد پیشنهادی، اندازه جمعیت به‌طور ثابت ۵۰۰ کروموزوم درنظر گرفته شده است. به‌علاوه، به لحاظ آماری برای هر مجموعه داده ۳۰ اجرا درنظر گرفته می‌شود، اگر بعد از ۵۰ نسل متوالی هزینه جریمه بهبود نیابد، اجرای الگوریتم خاتمه خواهد یافت.

جدول ۱- مجموعه مسائل نمونه کارتر.
Table 1- The Carter's benchmark dataset.

مجموعه داده	بازه‌های زمانی	امتحانات	دانشجویان	ثبت‌نام‌ها	چگالی
sta-83	13	139	611	5751	0.14
yor-83	21	181	941	6034	0.29
ear -83	24	190	1125	8109	0.27
lse-91	18	381	2726	10918	0.06
ute-92	10	184	2750	11793	0.08
hec-92	18	81	2823	10632	0.42
tre-92	23	261	4360	14901	0.18
kfu-93	20	461	5349	25113	0.06
rye-93	23	486	11483	45051	0.08
car-91	35	682	16925	56877	0.13
car-92	32	543	18419	55522	0.14
uta-92	35	622	21266	58976	0.13

۵- نتایج محاسباتی

در این بخش ابتدا نتایج محاسباتی الگوریتم ژنتیک پیشنهادی برای بهینه‌سازی *UETP* را بدون این‌که عملگرهای جستجوی محلی در آن اعمال شـوند، ارائه می‌دهیم. در **جدول ۲**، بهترین، میانگین و بدترین هزینه جریمه به‌دسـت آمده از *GA* پیشـنهادی روی ۱۲ مجموعه داده برای ۳۰ اجرا گزارش شده است.

جدول ۲ نشان می‌دهد که *GA* پیشنهادی جواب‌هایی تقریباً با اختلاف کم بین بهترین و بدترین مقدار به استثنای مجموعه داده *yor-83* تولید کرده است. این امر نشان‌دهنده‌ی توانمند بودن الگوریتم پیشنهادی در یافتن جواب‌ها ست، هرچند برای بهبود جواب‌های نهایی، از عملگرهای جستجوی محلی استفاده شده و رویکرد اصلاحی اجرا می‌شود.

جدول ۲- نتایج محاسباتی GA بدون اعمال جستجوی محلی.
Table 2- GA computational results without performing local search.

مجموعه داده	بهترین	میانگین	بدترین
sta-83	160.15	161.75	162.20
yor-83	38.11	44.84	46.17
ear -83	41.01	43.58	44.21
lse-91	13.21	13.76	14.06
ute-92	29.33	31.64	32.97
hec-92	11.93	12.14	13.76
tre-92	9.11	10.04	11.16
kfu-93	16.20	17.33	18.22
rye-93	11.19	12.48	13.07
car-91	6.20	6.58	7.01
car-92	4.87	5.43	5.71
uta-92	4.15	4.33	4.69

جدول ۳ نتایج محاسباتی رویکرد اصلاحی (*GA* پیشنهادی همراه با اعمال عملگرهای جستجوی محلی) را روی مجموعه مسائل نمونه کارتر نمایش می‌دهد. همان‌طور که از **جدول ۳** مشاهده می شود، *GA* پیشنهادی همراه با اعمال عملگرهای جستجوی محلی به‌عبارت دیگر، رویکرد اصـلاحی، جواب‌های با کیفیت‌تری نسـبت به وقتی که الگوریتم ژنتیک بدون اعمال عملگرهای جسـتجوی محلی اجرا می‌شود، ارائه داده است.

جدول ۳- نتایج محاسباتی رویکرد اصلاحی.
Table 3- Computational results of the modified approach.

مجموعه داده	بهترین	میانگین	زمان اجرا (ثانیه)
sta-83	157.90	158.87	224
yor-83	37.94	38.72	262
ear -83	37.01	37.99	301
lse-91	10.93	11.66	1231
ute-92	25.82	26.74	287
hec-92	10.23	11.43	201
tre-92	8.33	8.88	563
kfu-93	14.44	15.05	1584
rye-93	9.15	10.02	1602
car-91	4.74	5.33	2085
car-92	4.10	4.54	1866
uta-92	3.31	3.50	1994

اکنون، کارایی رویکرد اصلاحی پیشنهادی با نتایج ارائه شده از روش‌های دیگر در تحقیقات مشابه، مقایسه می‌شود. نتایج محاسباتی در **جدول ٤** گزارش شده است. برای هر الگوریتم مقادیر ورودی بهترین مقدار هزینه جریمه را نشان می‌دهد. در این تحقیق، به دلیل این‌که برای مطالعات مختلف در اجرای الگوریتم‌ها از محاسبات متنوع استفاده شده، مقایسه زمان اجرا مورد توجه قرار نگرفته است. روش‌هایی که با رویکرد اصلاحی پیشنهادی مقایسه می‌شوند، عبارتند از:

- *رویکرد دومرحله‌ای چند آغازی (عبدالله و بورک، ۲۰۰۶).*
- *الگوریتم ژنتیک ترکیبی (اسحاق و همکاران، ۲۰۱۶).*
- *رویکرد ساخت مبتنی بر تجزیه انطباقی و ترتیب (عبدالرحمن و همکاران، ۲۰۱۴).*
- *مرتب سازی ابتکاری چندگانه فازی (آسمونی و همکاران، ۲۰۰۴).*
- *الگوریتم ژنتیک (پیلای و بانژاف، ۲۰۱۰).*
- *بیش از حد ابتکاری مبتنی بر گراف (بورک و همکاران[۱]، ۲۰۰۷).*
- *برنامه‌نویسی ژنتیک (بدرالدینو فاطیما[۲]، ۲۰۱۰).*
- *بهینه‌سازی جفت‌گیری زنبور عسل (سابر و همکاران[۳]، ۲۰۰۹).*
- *روش ساخت ابتکاری ترتیبی (کارمیا و همکاران[۴]، ۲۰۰۰).*

در ضمن، نماد "-" به این معنی است که جوابی گزارش نشده است.

جدول ۴- مقایسه نتایج رویکرد اصلاحی با روش‌های دیگر.
Table 4- Comparison of the results of the modified approach with other methods.

مجموعه داده	رویکرد اصلاحی	عبداله و بورک	اسحاق و همکاران	عبدالرحمن وهمکاران	آسمونی و همکاران	پیلای و بانژاف	بورک و همکاران	بدرالدین و فاطیما	سابر و همکاران	کارمیا و همکاران
sta-83	157.9	159.0	158.9	162.3	160.4	157.8	158.2	157.3	157.0	158.2
yor-83	37.9	36.2	38.3	45.2	40.7	39.3	41.4	45.6	36.2	36.2
ear -83	37.0	36.0	37.0	40.6	37.0	35.9	37.9	41.2	34.7	29.3
lse-91	10.9	11.9	11.7	12.0	12.1	10.9	11.3	13.0	10.0	9.6
ute-92	25.8	26.0	26.7	27.9	27.8	27.2	28.0	-	25.9	24.4
hec-92	10.2	10.8	10.8	12.4	11.8	11.5	12.3	13.0	10.7	9.2
tre-92	8.3	8.5	8.6	9.7	8.7	8.4	8.8	9.3	7.9	9.4
kfu-93	14.4	15.2	14.9	16.2	15.8	14.4	15.2	15.9	13.0	13.8
rye-93	9.2	-	9.8	10.3	10.4	9.3	-	-	11.0	6.8
car-91	4.7	4.8	5.4	5.3	5.3	4.9	5.4	7.0	4.8	6.6
car-92	4.1	4.1	4.5	4.9	4.6	4.2	4.5	5.8	3.9	6.0
uta-92	3.3	3.6	3.4	3.5	3.6	3.4	3.9	3.8	3.1	3.5

با توجه به **جدول ٤**، مشاهده می‌شود که رویکرد اصلاحی مجموعه داده *car-91* را بهبود بخشیده، هم‌چنین، در بیشتر نمونه‌ها جواب‌های بهتری در مقایسه با الگوریتم‌های دیگر تولید کرده است. هرچند روش پیشنهادی بهترین بازده را برای هر مجموعه از مسائل نمونه ندارد، ولی کارایی این روش به شکلی واضح با دیگر روش‌ها قابل مقایسه و رقابت است.

به‌طور کلی، رویکرد ا صلاحی پی شنهادی در مقای سه با الگوریتم‌های ا سحاق و همکاران (۲۰۱٦)، عبدالرحمن و همکاران (۲۰۱٤)، آسمونی و همکاران (۲۰۰٤) و بورک و همکاران (۲۰۰۷) برای تمام ۱۲ مجموعه داده بهتر عمل کرده به‌علاوه، جواب‌های بهتری در ۱۱ نمونه (به استثنای sta-83) نسبت به روش بدرالدین و فاطیما (۲۰۱۰) ارائه کرده است. هم‌چنین در ارائه جواب‌های با کیفیت، در اکثر نمونه‌ها نسبت به عبداله و بورک (۲۰۰٦) و پیلای و بانژاف (۲۰۱۰) پیشی گرفته و نتایج مطلوبی را در مقایسه با الگوریتم‌های سابر و همکاران (۲۰۰۹) و کارمیا و همکاران (۲۰۰۰) داشته است.

[۱] Burke et al.
[۲] Bader-El-Den and Fatima
[۳] Sabar et al.
[۴] Caramia et al.

۶- نتیجه‌گیری و پیشنهادات

هدف کلی این تحقیق بررسی توانایی یک رویکرد اصلاحی پیشنهادی در حل *UETP* بدون ظرفیت است که در دو مرحله انجام می شود:

- *مرحله ساخت (ساخت جدول زمانی شدنی).*
- *مرحله بهبود (بهبود کیفیت جدول زمانی طراحی شده در مرحله اول به‌وسیله تقلیل مقدار هزینه‌ی جریمه مسئله).*

بزرگ‌ترین چالش در مرحله ساخت رسیدن به یک جواب شدنی با تعداد مورد نیاز از بازه‌های زمانی است، به‌طوری‌که در جدول زمانی شدنی، از قرار گرفتن دانشجویان به‌طور هم‌زمان برای بیش از یک امتحان اجتناب می‌شود. در این تحقیق، برای تولید و طراحی جدول زمانی شدنی از ترکیب روش‌های ابتکاری سطح پایین *HC* و *SD* استفاده شده است. در مرحله بهبود سعی می شود الگوریتم ژنتیک پیشنهادی با عملگرهای جستجوی محلی به منظور پخش امتحانات در میان بازه‌های زمانی امکان‌پذیر ارائه شده توسط مؤسسه، به‌کار گرفته شود، طوری‌که وقفه کافی در بین امتحانات برای مطالعه و مرور وجود داشته باشد. در طی اعمال این عملگرها، انتقال یک امتحان زمان‌بندی شده در یک بازه‌ی زمانی جدید یا تعویض دو امتحان زمان‌بندی شده بین دو بازه‌ی زمانی مختلف درنظر گرفته می‌شود، به شرطی که سبب کاهش در مقدار تابع هزینه‌ی جریمه گردد.

برای ارزیابی رویکرد پیشنهادی، کارایی این الگوریتم در مقایسه با روش‌های ارائه شده توسط محققان بر روی مجموعه مسائل نمونه کارتر بررسی شده است. نتایج نشان می‌دهند که الگوریتم پیشنهادی کاملاً رقابتی بوده و قادر است در بیشتر نمونه‌ها، جواب‌های بهتری در مقایسه با الگوریتم‌های دیگر تولید کند. قابل ذکر است که، رویکرد پیشنهادی مجموعه داده *car-91* را بهبود بخشیده، هم‌چنین، در بقیه نمونه‌ها نتایج مطلوبی را به همراه داشته و توانسته در بهبود جواب‌ها مؤثر عمل کند.

رویکرد اصلاحی پیشنهادی می‌تواند برای حل انواع دیگر از مسائل جدول زمانی مانند مسئله زمان‌بندی دروس دانشگاهی مورد استفاده قرار گیرد.

منابع

Abdullah, S., & Burke, E. K. (2006, June). A multi-start very large neighbourhood search approach with local search methods for examination timetabling. *ICAPS* (pp. 334-337). https://www.aaai.org/Papers/ICAPS/2006/ICAPS06-034.pdf.

Abdul-Rahman, S., Burke, E. K., Bargiela, A., McCollum, B., & özcan, E. (2014). A constructive approach to examination timetabling based on adaptive decomposition and ordering. *Annals of operations research*, *218*(1), 3–21.

Abounacer, R., Boukachour, J., Dkhissi, B., & Alaoui, A. E. H. (2010). A hybrid ant colony algorithm for the exam timetabling problem. *Revue African research journal in computer science (ARIMA)*, 15–42. (**In French**). hal-01286691

Asmuni, H. (2008). *Fuzzy methodologies for automated university timetabling solution construction and evaluation* (Doctoral dissertation, University of Nottingham). Retrieved from https://ethos.bl.uk/OrderDetails.do?did=1&uin=uk.bl.ethos.514699

Asmuni, H., Burke, E. K., Garibaldi, J. M., & McCollum, B. (2004, August). Fuzzy multiple heuristic orderings for examination timetabling. *International conference on the practice and theory of automated timetabling (PATAT)* (pp. 334-353). Berlin, Heidelberg: Springer. https://link. springer.com/chapter/10.1007/11593577_19

Assi, M., Halawi, B., & Haraty, R. A. (2018). Genetic algorithm analysis using the graph coloring method for solving the university timetable problem. *Procedia computer science*, *126*, 899-906. https://doi.org/10.1016/j.procs.2018.08.024

Bader-El-Den, M., & Fatima, S. (2010, April). Genetic programming for auction based scheduling. *European conference on genetic programming (EuroGP)* (pp. 256-267). Berlin, Heidelberg: Springer. https://link.springer.com/ chapter/10.1007/978-3-642-12148-7_22

Burke, E. K., McCollum, B., Meisels, A., Petrovic, S., & Qu, R. (2007). A graph-based hyper-heuristic for educational timetabling problems. *European journal of operational research*, *176*(1), 177-192.

Burke, E., Elliman, D., Ford, P., & Weare, R. (1995a, August). Examination timetabling in British universities: A survey. *International conference on the practice and theory of automated timetabling (PATAT)* (pp. 76-90). Berlin, Heidelberg: Springer. https://link.springer.com/chapter/10.1007/3-540-61794-9_52

Burke, E. K., Newall, J. P., & Weare, R. F. (1995b, August). A memetic algorithm for university exam timetabling. *International conference on the practice and theory of automated timetabling (PATAT)* (pp. 241-250). Berlin, Heidelberg: Springer. https://link.springer.com/chapter/10.1007/3-540-61794-9_63

Caramia, M., Dell'Olmo, P., & Italiano, G. F. (2000, September). New algorithms for examination timetabling. *International workshop on algorithm engineering (WAE)* (pp. 230-241). Berlin, Heidelberg: Springer. https://link. springer.com/chapter/10.1007/3-540-44691-5_20

Carter, M. W., & Laporte, G. (1995, August). Recent developments in practical examination timetabling. *International conference on the practice and theory of automated timetabling (PATAT)* (pp. 1-21). Berlin, Heidelberg: Springer. https://link.springer.com/chapter/10.1007/3-540-61794-9_49

Carter, M. W., Laporte, G., & Lee, S. Y. (1996). Examination timetabling: algorithmic strategies and applications. *Journal of the operational research society*, *47*(3), 373-383.

Casey, S., & Thompson, J. (2002, August). GRASPing the examination scheduling problem. *International conference on the practice and theory of automated timetabling (PATAT)* (pp. 232-244). Berlin, Heidelberg: Springer. https://link.springer.com/chapter/10.1007/978-3-540-45157-0_15

Chu, S. C., & Fang, H. L. (1999, December). Genetic algorithms vs. tabu search in timetable scheduling. *1999 third international conference on knowledge-based intelligent information engineering systems. Proceedings (Cat. No. 99TH8410)* (pp. 492-495). Adelaide, SA, Australia, Australia: IEEE. https://ieeexplore.ieee.org/abstract/document/820230

Ishak, S., Lee, L. S., & Ibragimov, G. (2016). Hybrid genetic algorithm for university examination timetabling problem. *Malaysian journal of mathematical sciences*, *10*(2), 145-178.

Leite, N., Fernandes, C. M., Melicio, F., & Rosa, A. C. (2018). A cellular memetic algorithm for the examination timetabling problem. *Computers and operations research*, *94*, 118-138. https://doi.org/10.1016/j.cor.2018.02.009

Leite, N., Melício, F., & Rosa, A. C. (2019). A fast simulated annealing algorithm for the examination timetabling problem. *Expert systems with applications*, *122*, 137-151. https://doi.org/10.1016/j.eswa.2018.12.048

Muklason, A., Syahrani, G. B., & Marom, A. (2019). Great deluge based hyper-heuristics for solving real-world university examination timetabling problem: new data set and approach. *Procedia computer science*, *161*, 647-655. https://doi.org/10.1016/j.procs.2019.11.168

Pillay, N. (2012). Evolving hyper-heuristics for the uncapacitated examination timetabling problem. *Journal of the operational research society*, *63*(1), 47-58.

Pillay, N., & Banzhaf, W. (2010). An informed genetic algorithm for the examination timetabling problem. *Applied soft computing*, *10*(2), 457-467.

Sabar, N. R., Ayob, M., & Kendall, G. (2009, August). Solving examination timetabling problems using honey-bee mating optimization (ETP-HBMO). *Multidisciplinary international conference on scheduling: theory and applications (MISTA)* (pp. 10-12). Ireland, Dublin. http://www.mistaconference.org/2009/papers/399-408-123-P.pdf

Licensee Decisions & Operations Research. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).