

طراحی نظام جامع تصمیم‌گیری و زمان‌بندی برنامه‌ریزی نوین آموزشی
دانشگاه پیام نور با رویکرد الگوریتم زنبور عسل (مطالعه موردی):
دانشگاه پیام نور مرکز لامرد

Comprehensive Decision-Making System Design and the New
Educational Planning Scheduling Using Bees Algorithm
Approach (Case Study: Payam-e Noor University, Lamard)

تاریخ پذیرش مقاله: ۹۶/۰۴/۲۷

تاریخ دریافت مقاله: ۹۵/۱۲/۰۲

Mahmoud Safari

محمود صفری*

Gholam Reza Ismailian

غلامرضا اسماعیلیان**

Abstract: In this study, a single-objective model is presented for academic courses schedule problem. The purpose of this problem is to make appropriate and acceptable timetable for university courses taking into account a set of constraints and preferences of professors, students and universities according to the educational environment in Iran. In schedule problems, constraints are divided into two categories: hard and soft, hard constraints must be met, and ensure the feasibility of a solution, and soft constraints that represent utility and preferences of a problem, they are considered for better quality of a schedule. Given that an unsolvable problem is associated with computational complexity, meta-heuristic algorithms and bees algorithm are used to solve the models. To speed up the application, the initial population is produced in a way that a lot of hard constraints are saturated. In order to test the mathematical model and the app in question, the Lamerd PNU data is used. Solving model using bees method are derived from two important functions of dance, to escape from local optimum and improve the efficiency of the problem, and finally reach an acceptable solution, the results show that the accuracy of the mathematical model is 94 per cent, and c is better than classic status and general condition.

Keywords: New education planning system, Decision-making expert system, Nonlinear mathematical modeling, Bees algorithm, PNU

چکیده: پژوهش حاضر، مدلی برای مسئله زمان‌بندی درس‌های دانشگاهی به صورت تک‌هدفه کرده که هدف تهیه جدول زمانی درس‌های دانشگاهی قابل قبول، با در نظر گرفتن مجموعه‌ای از محدودیت‌ها و ترجیح‌های استادان، دانشجویان و دانشگاه با توجه به فضای آموزشی است. در این مسائل، جدول زمانی محدودیت‌ها، به دودسته سخت و نرم که محدودیت‌های سخت باید برآورده شده و شدنی بودن جواب را تضمین کند و محدودیت‌های نرم که بیان‌کننده مطلوبیت و ترجیح‌های مسئله بوده و برای کیفیت بهتر جدول زمانی در نظر گرفته می‌شوند.

با توجه به اینکه مسئله‌ای حل نشدنی یا پیچیدگی محاسباتی در پیش روی است، برای حل مدل‌ها از الگوریتم‌های فرا ابتکاری، الگوریتم زنبور عسل استفاده شده است. برای افزایش سرعت برنامه جمعیت اولیه را به شکلی تولید کرده که تعداد زیادی از محدودیت‌های سخت اشباع شوند. برای آزمودن مدل ریاضی و برنامه مورد نظر از داده‌های دانشگاه پیام نور لامرد استفاده شده است. حل مدل با روش زنبور عسل از دو تابع مهم رقص برای فرار از بهینه محلی و بهبود کارایی مسئله و در نهایت رسیدن به یک جواب قابل قبول، بهره برداری می‌شود. نتایج نشان می‌دهد در نهایت دقت مدل ریاضی ۹۴ درصد بوده و نسبت به وضعیت کلاسیک و عمومی شرایط بهتری دارد.

واژگان کلیدی: نظام برنامه‌ریزی نوین آموزشی، نظام خبره تصمیم‌گیری، مدل‌سازی ریاضی غیرخطی، الگوریتم زنبور عسل، دانشگاه پیام نور.

*دانشجوی کارشناسی ارشد مهندسی صنایع دانشگاه پیام نور مرکز بین‌المللی عسلویه (نویسنده

مسئول : m_safari555@yahoo.com

**استادیار گروه مهندسی صنایع دانشگاه پیام نور مرکز تهران

مقدمه

جدول زمانی نوع خاصی از مسئله زمان‌بندی است. در سال ۱۹۹۶، آقای زن^۱ تهیه جدول زمانی را به‌عنوان مسئله قرار دادن منابع خاص، با توجه به محدودیت‌ها، در تعداد محدودی بازه زمانی و مکانی، باهدف ارضاء مجموعه‌ای از اهداف تا حد ممکن توصیف کرد. این تعریف عمومی توصیفی از مسائل تهیه جدول زمانی است که به‌طورکلی پذیرفته شده است. در حال حاضر در اغلب دانشگاه‌ها زمان‌بندی درس‌ها به‌صورت دستی و توسط افراد مجرب انجام می‌گیرد. در برنامه‌ریزی دستی، با روشی تکراری درس‌ها زمان‌بندی می‌شوند بدین‌صورت که در هر تکرار، یک درس انتخاب می‌شود و زمان‌بندی از دو درسی شروع می‌شود که تعداد دانشجوی بیشتری هم‌زمان برای آن دو درس ثبت‌نام کرده باشند و بنابراین این دو درس نباید در یک‌زمان ارائه شوند و باید انتخاب استاد و بازه زمانی به‌گونه‌ای صورت گیرد که محدودیت‌های دیگر را نیز نقض نکند.

این روند همچنان ادامه پیدا می‌کند تا تمامی درس‌ها به استاد و زمانی تخصیص داده شوند. از آنجا که زمان‌بندی دستی وقت‌گیر است و لزوماً تمام خواسته‌های استادان و دانشجویان را به بهترین نحو برآورده نمی‌کند، استفاده از روش کامپیوتری که در زمانی بسیار کمتر، به نتیجه مطلوبی دست یابد، مناسب به نظر می‌رسد. اما بیشتر این روش‌ها، به‌نوعی شبیه‌سازی نحوه زمان‌بندی توسط نیروی انسانی است. در این روش‌ها، ابتدا مفیدترین درس‌ها زمان‌بندی می‌شوند. سپس درس‌ها با اولویت کمتر، به‌طور متناوب به جدول زمانی ناتمام اضافه‌شده تا اینکه تمام درس‌ها زمان‌بندی شوند (دید^۲، ۲۰۰۷).

البته، این‌گونه مسائل زمان‌بندی نمی‌تواند به‌صورت کاملاً نرم‌افزاری انجام شود. زیرا معیارهای بهتر بودن نسبی زمان‌بندی‌ها، لزوماً به‌آسانی به زبان نرم‌افزارهای کامپیوتری بیان نمی‌شوند. از طرفی، از آنجا که فضای جواب در این‌گونه مسائل بسیار بزرگ است، دخالت نیروی انسانی می‌تواند مسئله را در راستای جواب‌های مناسب‌تر هدایت کند، درحالی‌که روش‌های کامپیوتری به‌صورت کامل، قادر به یافتن این جواب‌های مناسب نیستند (دید، ۲۰۰۷).

1. Wren

2. Head

به همین دلیل، بیشتر دستگاه‌های حل، امکان تغییر و دست‌کاری در جواب را برای کاربر فراهم می‌سازند. همه این عوامل ضرورت استفاده از روش‌های مدل‌سازی و حل کامپیوتری را که در زمان بسیار کمتری جواب خوبی فراهم می‌آورند، دوچندان می‌کند. با توجه به این‌که در دنیای واقعی، مشخصات و ابعاد مسئله فوق‌گوناگونی و محدودیت‌های زیادی دارد، در نظر گرفتن تمام این خصوصیات به فضای حل بسیار بزرگی منجر می‌شود که این مسئله را به مسئله‌ای حل‌نشده با پیچیدگی محاسباتی^۱ (زمان معقول) تبدیل کرده (گناوه^۲، ۲۰۱۲) و در برخی منابع نیز این مسئله به‌عنوان یک مسئله با پیچیدگی زمانی چندجمله‌ای^۳ معرفی شده است (شبان^۴، ۲۰۰۷).

مسئله زمان‌بندی درس‌های دانشگاهی با توجه به ساختار و مقررات آموزش عالی از کشوری به کشور دیگر متفاوت است. حتی با توجه به قوانین داخلی هر دانشگاه، محدودیت‌ها و تابع هدف می‌تواند تغییر کند. در این پژوهش، هدف استفاده از جدول زمانی درس‌های دانشگاهی مناسب و قابل قبول و با در نظر گرفتن مجموعه‌ای از محدودیت‌ها و ترجیح‌های استادان، دانشجویان و دانشگاه با توجه به شرایط بومی در ایران بوده که مدل ریاضی را برای انتخاب درس‌ها ارائه می‌کند که دقت بیشتری در تصمیم‌گیری دانشجویان برای انتخاب واحد درسی را دارد.

از الگوریتم فرا ابتکاری جفت‌گیری زنبور عسل برای حل مسئله زمان‌بندی امتحانات و زمان‌بندی درس‌ها دانشگاهی استفاده شده است. الگوریتم جفت‌گیری زنبور عسل^۵ الگوریتمی الهام گرفته از طبیعت است که شبیه‌سازی فرایند واقعی جفت‌گیری زنبور عسل است. در این مقاله مدل ریاضی برای مسائل ارائه نشده است. محدودیت‌های نرم مسئله تنها ترجیحات دانشجویان را در نظر گرفته است. یکی از تفاوت‌های اصلی این الگوریتم با دیگر الگوریتم‌های مبتنی بر جمعیت مانند ژنتیک^۶ این است که به‌جای انتخاب دو والد در هر مرحله، تنها نیاز به انتخاب یک والد است و والد دوم همان ملکه است که در مرحله حضور دارد (سا بر^۷، ۲۰۰۹).

¹ Non-deterministic Polynomial-time Hard (NP-Hard)

² Gunawan

³ Non-deterministic Polynomial-time Complete (NP-Complete)

⁴ Shaban

⁵ Honey-Bee Mating Optimization- HBMO

⁶ Genetic

⁷ Sabar

در پژوهشی مدل جدول زمانی درس‌ها دانشگاهی ارائه شده توسط والدی توسعه و با الگوریتم جستجوی ممنوعه حل شده و همان مدل را با الگوریتم جستجوی ممنوعه با چهار استراتژی همسایگی مختلف حل کردند. استراتژی اول: انتقال یک درس به یک کلاس خالی در یک دوره زمانی، استراتژی دوم: جابجایی دو درس با یکدیگر، استراتژی سوم: استفاده از دو استراتژی اول و دوم به صورت هم‌زمان و موازی در هر تکرار و انتخاب بهترین جواب و استراتژی چهارم: الگوریتم با استراتژی اول شروع و در صورت دست نیافتن به یک جواب بهتر پس از تعداد تکرار مشخص، از استراتژی دوم یک‌بار استفاده می‌کند. در انتها نتایج حاصل از حل با تمام استراتژی‌ها مقایسه و ارزیابی شده است. ارزیابی نتایج نشان‌دهنده برتری استراتژی چهارم نسبت به سه استراتژی دیگر است (آلادان^۱، ۲۰۰۹). مسئله‌های زمان‌بندی درس‌های دانشگاهی و زمان‌بندی امتحانات دانشگاهی ارائه شده در یک دانشگاه، با استفاده از الگوریتم پیشنهادی ماتیک ممنوعه با ساختار همسایگی چندگانه مسائل حل و محدودیت‌های نرم مسائل مبتنی بر ترجیحات دانشجویان طراحی شده است و هدف مسائل کاهش دادن تعداد تخطی از محدودیت‌های نرم است. در الگوریتم پیشنهادی از عملگر جهش، تقاطع و از ۹ ساختار همسایگی و لیست ممنوعه برای ساختار همسایگی‌ها استفاده شده است. نتایج نشان می‌دهد، مطلوبیت جواب‌های حاصل از الگوریتم پیشنهادی نسبت به الگوریتم ماتیک تنها بهتر است (عبدالله^۲، ۲۰۱۲).

قرونی و همکاران به بررسی و توسعه یک الگوریتم تکاملی پیشنهادی برای حل مسئله جدول زمانی درس‌های دانشگاهی پرداختند. هدف اصلی در این مقاله بررسی تأثیر روش‌های حل تکاملی بر روی مسائل جدول درس‌ها دانشگاهی است. الگوریتم تکاملی پیشنهادی از توسعه الگوریتم ماتیک حاصل شده است. الگوریتم پیشنهادی برای حل چندین مسئله در سایزهای کوچک، متوسط و بزرگ مورد استفاده قرار گرفته است. بررسی و مقایسه نتایج حاصل از الگوریتم تکاملی پیشنهادی با دیگر الگوریتم‌های تکاملی مانند ماتیک و ژنتیک گروهی، نشان‌دهنده برتری الگوریتم تکاملی پیشنهادی برای مسئله‌های سایز کوچک و متوسط نسبت به دیگر الگوریتم‌ها است.

1. Aladag

2. Abdullah

جدول زمانی درس‌های دانشگاهی^۱ عبارت از تخصیص تعداد معینی از منابع مانند استادان و درس‌ها، به تعداد محدودی از دوره‌های زمانی و کلاس در یک دوره مشخص با توجه به مجموعه‌ای از محدودیت‌ها، جهت رسیدن به یکسری از اهداف مشخص است. مفاهیم استفاده شده در جدول زمانی درس‌های دانشگاهی به شرح ذیل است:

۱. رویداد: فعالیتی که باید زمان‌بندی شود، مانند جلسات درسی.
۲. برش زمانی: یک تعداد بازه زمانی که هر یک از رویدادها در آن بازه‌های زمانی، زمان‌بندی می‌شوند.
۳. منبع: منابعی که توسط رویدادها مورد نیاز است، مانند کلاس‌ها.
۴. قید: یک محدودیت برای زمان‌بندی رویدادها، مانند ظرفیت کلاس درسی.
۵. افراد: کسی یا کسانی که همراه با رویدادها وجود دارند، مانند استادان و دانشجویان.

۶. برخورد: اگر دو رویداد با یکدیگر برخورد داشته باشند، مانند زمان‌بندی بیش از یک درس در یک کلاس در یک زمان یکسان.

البته معمولاً در این نوع مسائل محدودیت‌ها به دودسته سخت و نرم تقسیم می‌شوند. محدودیت‌های سخت، محدودیت‌هایی هستند که حتماً باید برآورده شوند و شدنی بودن جواب را تضمین می‌کنند و محدودیت‌های نرم بیان‌کننده مطلوبیت و ترجیحات مسئله هستند که برای کیفیت بهتر جدول زمانی در نظر گرفته می‌شوند و حتماً لزومی ندارد که همانند محدودیت‌های سخت به‌طور کامل برآورده شوند. برای به دست آوردن یک جدول زمانی باکیفیت، باید مسئله دارای فضای جواب و کمترین تعداد تجاوز را در محدودیت‌های نرم داشته باشیم (برک^۲، ۲۰۰۲).

محدودیت‌های نرم از طریق تابع جریمه ارزیابی می‌شوند و تابع هدف این مسائل از مجموع وزن دهی شده توابع جریمه محدودیت‌های نرم تشکیل می‌شود.

محدودیت‌های سخت عمومی به کار گرفته شده در این مسائل به صورت زیر هستند:

۱. یک منبع (درس، استاد، دانشجو) نمی‌تواند در آن واحد در چند جا (کلاس، دوره زمانی) استفاده شود.

¹. University Course Timetabling

². Burke

۲. در هر دوره زمانی باید منابع در دسترس برای مواردی که زمان‌بندی شده‌اند کافی باشد.

اما محدودیت‌های نرم با توجه به نوع و ترجیح‌ها، برای هر مسئله متفاوت است. در این پژوهش ترجیح‌های استادان، دانشجویان و دانشگاه ملاک است. در حال حاضر، در اغلب دانشگاه‌ها زمان‌بندی درس‌ها به صورت دستی و توسط افراد مجرب انجام می‌گیرد. در برنامه‌ریزی دستی، با روشی تکراری درس‌ها زمان‌بندی می‌شوند بدین صورت که در هر تکرار، یک درس انتخاب می‌شود و زمان‌بندی از دو درسی شروع می‌شود که تعداد دانشجوی بیشتری هم‌زمان برای آن دو درس ثبت‌نام کرده باشند و بنابراین این دو درس نباید در یک زمان ارائه شوند و باید انتخاب استاد و بازه زمانی به گونه‌ای صورت گیرد که محدودیت‌های دیگر را نیز نقض نکند. این روند همچنان ادامه پیدا می‌کند تا تمامی درس‌ها به استاد و زمانی تخصیص داده شوند.

از آنجا که زمان‌بندی دستی وقت‌گیر است و لزوماً تمام خواسته‌های استادان و دانشجویان را به بهترین نحو برآورده نمی‌کند، استفاده از روش کامپیوتری که در زمانی بسیار کمتر، جواب خوبی را فراهم می‌آورد، مناسب به نظر می‌رسد. تحقیق و جستجو درباره زمان‌بندی ماشینی درس‌ها به دهه ۶۰ میلادی برمی‌گردد. بیشتر این روش‌ها، به نوعی شبیه‌سازی نحوه زمان‌بندی توسط نیروی انسانی است. در این روش‌ها، ابتدا مفیدترین درس‌ها زمان‌بندی می‌شوند. سپس درس‌ها با اولویت کمتر، به طور متناوب به جدول زمانی ناتمام اضافه شده تا اینکه تمام درس‌ها زمان‌بندی شوند. پس از آن، محققان روش‌های عمومی‌تر مانند روش‌های برنامه‌ریزی عدد صحیح، جریان شبکه، رنگ‌آمیزی گراف و ... را بررسی کردند. در دو دهه آخر، تلاش‌ها بر روش‌های فرا

ابتکاری متمرکز شده است. البته این گونه مسائل زمان‌بندی نمی‌تواند به صورت کاملاً نرم‌افزاری انجام شود. زیرا معیارهای بهتر بودن نسبی زمان‌بندی‌ها، لزوماً به آسانی به زبان نرم‌افزارهای کامپیوتری بیان نمی‌شوند. از طرفی، از آنجا که فضای جواب در این گونه مسائل بسیار بزرگ است، دخالت نیروی انسانی می‌تواند مسئله را در راستای جواب‌های مناسب‌تر هدایت کند، در حالی که روش‌های کامپیوتری به صورت کامل، قادر به یافتن این جواب‌های مناسب نیستند. به همین دلیل، بیشتر نظام‌های حل، امکان تغییر و دست‌کاری در جواب را برای کاربر فراهم می‌سازند.

زمان‌بندی دستی وقت‌گیر بوده و لزوماً تمام خواسته‌ها و ترجیحات استادان، دانشجویان و دانشگاه را پوشش نمی‌دهد و همچنین با افزایش روزافزون رشته‌ها، گرایش‌ها، تنوع درس‌ها، افزایش تعداد دانشجویان و محدود بودن منابع در یک دانشگاه، کار زمان‌بندی درس‌ها را دشوار می‌کند. همه این عوامل، ضرورت استفاده از روش‌های مدل‌سازی ریاضی مبتنی بر مدل‌های فرا ابتکاری و حل کامپیوتری که در زمان بسیار کمتری جواب خوب و مناسبی را فراهم می‌آورند، دوچندان می‌کند. با توجه به اینکه در دنیای واقعی، مشخصات و ابعاد مسئله فوق‌گوناگونی و محدودیت‌های زیادی دارد، مسئله زمان‌بندی درس‌های دانشگاهی با توجه به ساختار و قوانین آموزش عالی کشور و قوانین داخلی هر دانشگاه، محدودیت‌ها و تابع هدف را تعریف می‌کند که در نظر گرفتن تمام این خصوصیات به فضای حل بسیار بزرگی منجر شده است که این مسئله را به یک مسئله پیچیده ریاضی تبدیل می‌کند.

بررسی نتایج تحقیقات گذشته نشان می‌دهد که مسئله بر مبنای ترجیحات استادان و دانشگاه و به صورت تخصیص استاد به درس و درس به دوره‌های زمانی به صورت هم‌زمان در نظر گرفته و مدل‌سازی می‌شود و با فرض اینکه دانشجویان در هنگام ثبت‌نام حق انتخاب دارند، ترجیحات آنها در نظر گرفته نشده است.

لذا این پژوهش برای حل مدل از الگوریتم فرا ابتکاری زنبور عسل به کمک دو تابع مهم رقص برای فرار از بهینگی محلی و بهبود کارایی مسئله و درنهایت رسیدن به یک جواب قابل قبول با در نظر گرفتن ترجیح‌های دانشگاه، استاد، دانشجو از نمونه‌های داده شده از طرف دانشگاه پیام نور مرکز لامرد، بهره‌برداری و استفاده کرده است.

روش‌شناسی پژوهش

بیشتر مسائل بهینه‌سازی ترکیبی، در سطح مسائل مدل‌سازی پیچیده قرار می‌گیرند و برای حل آنها از روش‌های تقریبی که در یک‌زمان حل قابل قبول، جواب‌هایی نزدیک به بهینه را می‌دهند، استفاده می‌شوند که از الگوریتم‌های فرا ابتکاری، برای این پژوهش استفاده شده است. الگوریتم‌های فرا ابتکاری در واقع روش‌های ابتکاری هوشمندانه‌تری هستند که قواعد آن‌ها تنها برای یک مسئله خاص، بلکه برای طیف وسیعی از مسائل طراحی شده‌اند. در این پژوهش با توجه به موضوع و ساختار مسئله

و پژوهش‌های گذشته، الگوریتم فرا ابتکاری زنبورعسل برای حل مدل استفاده شده است.

۱- الگوریتم زنبورعسل و استراتژی جستجو

در علوم کامپیوتر و پژوهش عملیاتی، الگوریتم زنبورعسل، الگوریتم جستجوی مبتنی بر جمعیت است که در سال ۲۰۰۵ میلادی توسعه یافت (فام^۱، ۲۰۰۵). این الگوریتم تقلید رفتار جستجوگری مواد غذایی زنبورعسل است. در نسخه اولیه این الگوریتم نوعی جستجوی محلی همراه با جستجوی جهانی انجام می‌دهد و می‌تواند برای هر دو بهینه‌سازی ترکیبی و بهینه‌سازی مستمر مورد استفاده قرار گیرد. تنها شرط استفاده از الگوریتم زنبورعسل این است که برخی اندازه‌گیری‌های فاصله مشترک هندسی داده‌های موجود در یک مدل بین راه حل‌ها تعریف شده است. در پژوهش فام و همکارانش وضعیت اثربخشی و توانایی‌های خاص الگوریتم زنبورعسل در تعدادی از مطالعات که به‌طور مستمر انجام شده‌اند، ثابت شده است (فام، ۲۰۱۳).

یک مجموعه از زنبورهای عسل می‌توانند در طول فواصل طولانی (بیش از ۱۴ کیلومتر) و در جهت‌های مختلف به‌طور هم‌زمان به برداشت شهد یا گرده از منابع غذایی متعدد پراکنده شوند. این زنبورهای دیده‌بان به‌طور تصادفی در منطقه اطراف کندو حرکت می‌کنند و سودآوری منابع غذایی وارد شده را ارزیابی می‌کنند (ترشرو^۲، ۲۰۰۵). وقتی آنها به کندو بازمی‌گردند، دیده‌بان‌ها مواد غذایی برداشت شده را ذخیره می‌کنند. آن دسته از زنبورهایی که منبع غذایی بسیار سودآوری پیدا کردند به یک منطقه در کندو به نام "پیست رقص" رفته و آیینی به نام "رقص حرکتی" را اجرا می‌کنند (ون فرش^۳، ۱۹۶۷). در حین این رقص، زنبور دیده‌بان در مورد محلی که کشف کرده با تماشاچیان بیکار صحبت می‌کند که به بهره‌برداری از گل‌ها بپیوندند.

از آنجا که طول رقص متناسب با امتیاز دیده‌بان از منبع غذایی است، کاوشگرهای بیشتری برای برداشت تکه‌های گل با بهترین امتیاز استخدام می‌شوند. بعد از رقص، دیده‌بان برای جمع‌آوری بیشتر غذا به محلی که کشف کرده است، می‌رود. تا زمانی که این محل‌ها سودآور تلقی شوند، موقع برگشت این منابع غذایی غنی توسط دیده

1. Pham

2. Tereshko

3. Von Frisch

بان‌ها تبلیغ می‌شوند. کاوشگرهای استخدام شده نیز ممکن است این رقص را انجام دهند تا میزان استخدام برای پیدا کردن گل‌های پرارزش افزایش یابد.

مراحل الگوریتم به روال زیر است:

- تولید پاسخ‌های اولیه و ارزیابی آنها: پاسخ‌های اولیه در واقع همان منابع غذایی یافته شده توسط زنبورهای پیش‌آهنگ است که با توجه به ارزش آن زنبورها رقصی را انجام می‌دهند.
- انتخاب منابع (پاسخ‌های) بهتر و ارسال زنبورهای کارگر به آن منابع: از بین پاسخ‌های اولیه تعدادی پاسخ با توجه به ارزش آن انتخاب می‌شود و از بین پاسخ‌های انتخاب شده تعدادی پاسخ نخبه نیز انتخاب می‌شود.
- تولید پاسخ‌های همسایه: زمانی که زنبورها برای اعلام موقعیت منبع غذایی عمل رقص را انجام می‌دهند برداشت‌های متفاوتی از مکان منبع غذایی می‌شود چون این رقصی نسبی است و مختصات دقیق مشخص نمی‌شود پس باعث می‌شود یکسری جواب‌های همسایه حول جواب اصلی به دست آید. از این ایده برای تولید پاسخ‌های همسایه استفاده می‌شود.
- مقایسه همه زنبورهای یک منبع غذایی و انتخاب بهترین مورد.
- زنبورهای منابع غیر منتخب با منابع (پاسخ) تصادفی جایگزین می‌شوند منابع انتخاب نشده، رها و زنبورهای آن امر به پیدا کردن منابع جدید می‌شوند.
- ذخیره موقعیت بهترین پاسخ
- بازگشت به مرحله دو در صورت برآورده نشدن شرایط توقف

۲- پارامترهای الگوریتم زنبورعسل

1. n_scout_Bee تعداد زنبورهای پیش‌آهنگ در واقع همان تعداد جمعیت اولیه
2. $n_selected_site < n_scout_Bee$ تعداد بخش‌های انتخاب شده
3. $n_selected_Elite_site < n_selected_site$ تعداد بخش‌های نخبه
4. $n_selected_site_Bee$ تعداد زنبورهای اختصاص داده شده به بخش انتخاب شده

تعداد زنبورهای اختصاص داده شده به 5. n_selected_Elite_site_Bee بخش‌های نخبه

۳- تشریح و بیان مدل ریاضی

مدل ریاضی مسئله در فرمول‌بندی (۱) نشان داده شده است:

$$\min f(x) = w_{s_1} f_{s_1}(x) + w_{s_2} f_{s_2}(x) \quad (1)$$

تابع هدف (۱) بر اساس محدودیت‌های نرم مسئله بیان شده است. اجزای $f_{s1}(x)$ و $f_{s2}(x)$ تابع هدف، مطابق با محدودیت‌های S_1, S_2 است و ضرایب WS_1, WS_2 نیز نشان‌دهنده وزن و اهمیت هر یک از محدودیت‌های نرم (جریمه تخطی برنامه از هر یک از محدودیت‌های نرم متناظر) است.

۴- تنظیم پارامترهای الگوریتم زنبور عسل

در فرایند دستیابی به جواب‌های مناسب و مطلوب، عوامل مختلفی تأثیرگذار است. از جمله این عوامل، منطق به‌کار رفته در توسعه الگوریتم و مقادیر تخصیص داده شده به پارامترهای الگوریتم است. در نتیجه چنانچه ترکیب پارامترهای مؤثر در اجرای الگوریتم به درستی انتخاب نشوند، جواب‌های به‌دست آمده از کیفیت لازم برخوردار نخواهند بود. چگونگی تعیین مقادیر پارامترهای فرا ابتکاری، از موضوعات اصلی تصمیم‌گیری هنگام استفاده از الگوریتم‌های فرا ابتکاری است.

تغییر مقادیر پارامترها و اجزای هر الگوریتم می‌تواند تأثیر زیادی بر عملکرد و کارایی الگوریتم‌ها داشته باشد. زمان رسیدن به جواب بهینه و کیفیت جواب‌های به‌دست‌آمده توسط الگوریتم‌های فرا ابتکاری تا حد زیادی به چگونگی طراحی الگوریتم‌ها و به‌خصوص نحوه کدگذاری، انتخاب، عملگرهای ابتکاری و مقادیر پارامترهای الگوریتم وابسته است. تاکنون یک متدولوژی منسجم برای تحلیل عملگرها و پارامترهای موجود در طراحی الگوریتم فرا ابتکاری پیدا نشده است که قابل قبول عموم محققان باشد. در نتیجه، پژوهشگران برآنند که برای کاهش زمان یافتن یک جواب قابل قبول توسط الگوریتم‌های فرا ابتکاری و یا افزایش کیفیت جواب‌های به‌دست‌آمده توسط آنها، به تنظیم پارامترها و عوامل کنترلی این الگوریتم‌ها اقدام کنند.

بیشتر پژوهشگران، مقادیر پارامترهای الگوریتم‌های فرا ابتکاری را با روش آزمون و خطا و یا بر اساس پژوهش‌های انجام‌شده روی مسائل مشابه تنظیم می‌کنند. در این پژوهش، بر اساس آزمایش‌های مقدماتی، یک طرح غربال‌گری روی متغیرهای مسئله صورت گرفته است و پارامترهایی را که در الگوریتم، تأثیر زیادی بر بهتر شدن کیفیت جواب نهایی می‌گذارند، شناسایی و مقادیر مناسب پارامترها تنظیم شده‌اند. با انجام طرح غربال‌گری و به‌طور تجربی پارامترهایی انتخاب شده است که برای به دست آوردن یک جواب باکیفیت خوب و در یک‌زمان قابل قبول هستند. مقادیر پارامترهای الگوریتم با انجام طرح غربال‌گری در جدول (۱) نشان داده شده است.

جدول (۱) مقادیر پارامترهای الگوریتم زنبور عسل با انجام طرح غربال‌گری

| مقدار پارامتر | پارامترهای الگوریتم |
|---------------|---|
| ۱۰ | تعداد زنبورهای عسل |
| ۵ | تعداد پاسخ‌های انتخاب‌شده |
| ۲ | تعداد پاسخ‌های نخبه |
| ۵ | تعداد زنبورهای اختصاص داده شده به پاسخ‌های انتخاب شده |
| ۱۰ | تعداد زنبورهای اختصاص داده شده به پاسخ‌های نخبه |

در ساختار اصلی الگوریتم زنبور عسل این پارامترها وجود دارد اما این پارامترها هیچ وابستگی به میزان هزینه پاسخ‌ها ندارند و به همه پاسخ‌ها به یک دید نگاه می‌کنند. در صورتی که نیاز به پاسخ‌های بهتر را بیشتر بررسی کرده و زنبور بیشتری به آن اختصاص می‌دهد. برای این منظور، مجموعه‌ای از نوآوری در الگوریتم ایجاد شده که پارامترهای آن در جدول (۲) نمایش داده شده است.

جدول (۲) مقادیر پارامترهای الگوریتم زنبورعسل بهبود یافته

| مقدار پارامتر | پارامترهای الگوریتم |
|--------------------------------------|---------------------------------|
| ۵ | تعداد زنبورهای عسل |
| ۲ | تعداد زنبورهای اختصاص داده شده |
| ۱ | میزان برازش هر پاسخ |
| $\frac{1}{F} = \text{هزینه هر پاسخ}$ | معیار انتخاب یا عدم انتخاب پاسخ |
| $\frac{F}{\text{mean}(F)} = d$ | |

اجرای الگوریتم

در این قسمت، الگوریتم استفاده شده برای حل مسئله زمان‌بندی درس‌های دانشگاه پیام نور مرکز لامرد را توضیح داده شده که الگوریتم با استفاده از نرم‌افزار ریاضی متلب^۱ در رایانه‌ای با وضعیت پردازشی^۲ معلوم کدنویسی و اجرا شده است. در این مسئله برای تخصیص درس‌ها به روز و بازه زمانی از یک جدول استفاده می‌کنیم. تعداد ستون‌های این جدول از دستور زیر به دست می‌آید.

$$n\text{Period}=(n_rooz-1)*(n_zaman-1);$$

درواقع تعداد روزها را در تعداد بازه زمانی ضرب کرده‌ایم. تعداد سطرها از دستور زیر به دست می‌آید.

$$n\text{Room}=n_kelas-1;$$

^۱. Matlab (R2012b)

^۲- مسئله روی کامپیوتری با پردازشگر Intel Cori7 1600 M با حافظه ۴GB و با استفاده از سیستم عامل Windows 10 Enterprise اجرا شده است.

سطرهای جدول را کلاس‌ها در نظر گرفته‌ایم. تعداد زنبورهای پیش‌آهنگ به صورت زیر است.

```
nScoutBee=5;
```

تعداد زنبورهای تخصیص داده شده به صورت زیر است.

```
nBee0=round(0.3*nScoutBee);
```

از آنجایی که هر زنبور در الگوریتم زنبورعسل به عنوان یک متغیر مسئله مورد استفاده قرار می‌گیرد ما این متغیر را به صورت زیر تعریف می‌کنیم.

```
empty_bee.jadval=[];
empty_bee.Position=[];
empty_bee.Cost=[];
empty_bee.ghaidCost=[];
empty_bee.mahCast=[];
```

این متغیر (زنبور) دارای پنج مؤلفه است:

- ۱- جدول تخصیص درس‌ها به روز و بازه زمانی
- ۲- تبدیل جدول به یک آرایه برای انجام محاسبات راحت‌تر و سریع‌تر
- ۳- هزینه هر جدول (پاسخ)
- ۴- هزینه محدودیت‌های مدل برای جدول
- ۵- هزینه تابع هدف مدل برای جدول

۱- **تابع هزینه:** برای محاسبه هزینه هر پاسخ از تابع هزینه استفاده می‌کنیم. در این تابع، هزینه تخلف از محدودیت‌ها و هزینه تابع هدف را محاسبه می‌کنیم.

```

function [z1,z,cost2]=timetabling(x,nF)
w=10000;
%% ostad dar yek zaman 2 dar ndashte bashad
cost=0;
for i=2:nF:numel(x)
    for j=i+nF:nF:numel(x)
        if x(i)==x(j) && x(i+2)==x(j+2) && x(i+3)==x(j+3)
            cost=cost+1;
        end
    end
end
%% mahdodiyat yek ostad va dars dar yek roz
cost1=0;
for i=5:nF:numel(x)
    for j=i+nF:nF:numel(x)
        if x(i)==x(j) && x(i-1)==x(j-1) && x(i-2)==x(j-2)
            cost1=cost1+1;
        end
    end
end
%% mahdodiyat zamanhaei ke ostad nemitavanad biayad
cost2=0;
for i=6:nF:numel(x)
    t=num2str(x(i));
    d=mod(x(i-4),5);
    if d==0
        p=5;
    else
        p=d;
    end
    if isempty(find(num2str(p)==t(:)))
        cost2=cost2+1;
    end
end
z=mean([cost cost1]);
z1=cost2+(w*z);
end

```

```

%% mahdodiyat zamanhaei ke ostad nemitavanad biayad
cost2=0;
for i=6:nF:numel(x)
    t=num2str(x(i));
    d=mod(x(i-4),5);
    if d==0
        p=5;
    else
        p=d;
    end
    if isempty(find(num2str(p)==t(:)))
        cost2=cost2+1;
    end
end
z=mean([cost cost1]);
z1=cost2+(w*z);
end

```

درواقع برای حل مدل و محاسبه هزینه از روش تابع جریمه استفاده شده است. متغیر W ضریب جریمه است و مقدار آن را بزرگ در نظر می‌گیریم تا الگوریتم از محدودیت‌های مدل تخطی نکند.

۲- تولید پاسخ‌های اولیه (جمعیت اولیه): در این الگوریتم پاسخ‌های اولیه را به شکلی تولید می‌کنیم که تعدادی از محدودیت‌های سخت مدل را ارضا کند و موجب محاسبات کمتری در تابع هزینه شود در نتیجه کارایی و سرعت برنامه بالا می‌رود. نحوه تولید پاسخ‌ها به شکلی است که محدودیت‌های ظرفیت کلاس، تداخل درس‌ها در یک‌زمان در یک کلاس و بحث عملی بودن درس‌ها را ارضا می‌کند. این فن باعث می‌شود که فضای حالت مسئله، کوچک‌تر و جستجو نیز بهتر شود. دستورات استفاده شده به صورت زیر است.

```

for i=1:nScoutBee
zaman_kelas=cell(nRoom,nPeriod);
temp1=[];
for j=2:n_dars
tekrar_har_dars=tab_dars{j,4};
zarfiyat_har_dars=tab_dars{j,5};
amali_bodan_dars=tab_dars{j,6};
if zarfiyat_har_dars==15 && amali_bodan_dars==1
for k=1:tekrar_har_dars
ran_kelas=randi([17 24],1);
ran_zaman=randi(30,1);
while ~isempty(zaman_kelas{ran_kelas,ran_zaman})
ran_kelas=randi([17 24],1);
ran_zaman=randi(30,1);
end
zaman_kelas{ran_kelas,ran_zaman}=[tab_dars{j,1} tab_dars{j,3} j];
temp1=[temp1 ran_kelas ran_zaman tab_dars{j,1}
tab_dars{j,3} ceil(ran_zaman/(n_zaman-1))
tab_mahdodiyat_ostad{ceil(ran_zaman/(n_zaman-
1))+1,tab_dars{j,3}+1} j];
end

```



```

elseif amali_bodan_dars~=1 && zarfiyat_har_dars==50
for k=1:tekrar_har_dars
ran_kelas=randi([1 11],1);
ran_zaman=randi(30,1);
while ~isempty(zaman_kelas{ran_kelas,ran_zaman})
ran_kelas=randi([1 11],1);
ran_zaman=randi(30,1);
end
zaman_kelas{ran_kelas,ran_zaman}=[tab_dars{j,1}
tab_dars{j,3} j];
temp1=[temp1 ran_kelas ran_zaman tab_dars{j,1}
tab_dars{j,3} ceil(ran_zaman/(n_zaman-1))
tab_mahdodiyat_ostad{ceil(ran_zaman/(n_zaman-
1))+1,tab_dars{j,3}+1} j];
end
elseif amali_bodan_dars~=1 && zarfiyat_har_dars==30
for k=1:tekrar_har_dars
ran_kelas=randi([12 14],1);
ran_zaman=randi(30,1);
while ~isempty(zaman_kelas{ran_kelas,ran_zaman})
ran_kelas=randi([12 14],1);
ran_zaman=randi(30,1);
end
zaman_kelas{ran_kelas,ran_zaman}=[tab_dars{j,1} tab_dars{j,3}
j];
temp1=[temp1 ran_kelas ran_zaman tab_dars{j,1}
tab_dars{j,3} ceil(ran_zaman/(n_zaman-1))
tab_mahdodiyat_ostad{ceil(ran_zaman/(n_zaman-
1))+1,tab_dars{j,3}+1} j];
end

```

```

elseif amali_bodan_dars~=1 && zarfiyat_har_dars==80
for k=1:tekrar_har_dars
ran_kelas=randi([15 16],1);
ran_zaman=randi(30,1);
while ~isempty(zaman_kelas{ran_kelas,ran_zaman})
ran_kelas=randi([15 16],1);
ran_zaman=randi(30,1);
end
zaman_kelas{ran_kelas,ran_zaman}=[tab_dars{j,1}
tab_dars{j,3} j];
temp1=[temp1 ran_kelas ran_zaman
tab_dars{j,1} tab_dars{j,3}
ceil(ran_zaman/(n_zaman-1))
tab_mahdodiyat_ostad{ceil(ran_zaman/(n_zama
n-1))+1,tab_dars{j,3}+1} j];
end
end
end
bee(i).jadval=zaman_kelas;
bee(i).Position=temp1;

[bee(i).Cost,bee(i).ghaidCost,bee(i).mahCast]=CostFunction(bee(i).P
osition,nF);
end

```

۳- بدنه اصلی الگوریتم: در بدنه اصلی الگوریتم زنبور عسل برای بهبود جستجو و فرار از بهینه محلی از یک نوآوری استفاده شده است. در این فن از یک تابع برازش استفاده شده است که این تابع برازش در فرمول (۲) تعریف شده است.

$$F_i = \frac{1}{\text{cost}_i + 1} \quad (2)$$

این تابع نشان می‌دهد که کدام پاسخها بهتر هستند. حال برای مقایسه همه پاسخها با میانگین آنها با استفاده از فرمول (۳) میانگین پاسخها را محاسبه می‌کنیم.

$$\bar{F} = \frac{1}{n} \sum_{i=1}^n F_i \quad (3)$$

حال برای به دست آوردن فاصله هر پاسخ از میانگین از فرمول (۴) استفاده می‌کنیم.

$$d_i = \frac{F_i}{\bar{F}} \quad (4)$$

درواقع با استفاده از این فاکتور می‌توان تشخیص داد که هر پاسخ چه میزان از میانگین پاسخ‌ها بیشتر یا کمتر است. هرچه این فاکتور بیشتر باشد پاسخ بهتر است. با استفاده از این فاکتور می‌توان به پاسخ‌هایی که مناسب نیستند، فرصتی دوباره داد و به پاسخ‌هایی که بهتر هستند زنبورها را بیشتری اختصاص داد. برای این منظور از یک تابع احتمال جستجوی تصادفی مانند فرمول (۵) استفاده می‌کنیم.

$$p_i = \begin{cases} 0.6 & d_i < 0.9 \\ 0.2 & 0.9 \leq d_i < 0.95 \\ 0.05 & 0.95 \leq d_i < 1.15 \\ 0 & d_i \geq 1.15 \end{cases} \quad (5)$$

درواقع این تابع احتمال بیان می‌کند که هر پاسخی از میانگین پاسخ‌ها کمتر باشد با احتمال بیشتری کنار گذاشته می‌شود.

مقدار زنبورهای اختصاص داده شده به پاسخ‌ها بر اساس فرمول (۶) هست.

$$n_{bee_i} = [d_i \times n_0] \quad (6)$$

که n_0 یک مقدار اولیه برای تعداد زنبورها هست. پس از انتخاب پاسخ‌های مناسب نیاز است که پاسخ‌ها همسایه این پاسخ‌ها بررسی شود؛ یعنی یک جستجوی محلی را انجام می‌دهیم برای این منظور از تابع رقص استفاده می‌کنیم. در این الگوریتم از دو تابع رقص استفاده شده است. تابع اول برای فرار از بهینه محلی و ارضا کردن محدودیت‌های سخت و تابع دوم برای برآورد کردن محدودیت‌های نرم استفاده شده است.

۴- تابع رقص اول: برای برآورد کردن محدودیت‌های سخت و فرار از بهینه محلی از این تابع استفاده می‌کنیم. در این تابع یک موجودیت که در ماتریس مکان-زمان قرار دارد را به یک جایگاه خالی به طوری که هیچ محدودیت سختی را نقض نکند، منتقل می‌کنیم. با این کار پاسخ‌های همسایه‌ای که از محدودیت‌های سخت تجاوز کردند اصلاح می‌شوند. این اصلاح شدن ممکن است هزینه‌ای داشته باشد و آن هزینه افزایش مقدار هزینه محدودیت‌های نرم باشد که در نهایت یک تعادل بین ارضا کردن کامل محدودیت‌های سخت و ارضا بعضی محدودیت‌های نرم تا جای ممکن می‌شود. تابع رقص اول در ابتدا یک موجودیت به طور تصادف انتخاب می‌کند و آن را به یک مکان مناسب انتقال می‌دهد و این کار آن قدر انجام می‌شود تا هزینه محدودیت‌های سخت به صفر رسد. دستورات این تابع به صورت زیر هست.

```
function [z,c]=UniformBeeDance1(x,y,nF,n_zaman,mahdodiyat)
temp2=[];
for i=1:nF:numel(y)
temp2=[temp2;y(1,i:(i+nF)-1)];
end
[ntemp,~]=size(temp2);
ran=randi(ntemp,1);
```

```
kelas=temp2(ran,1);
zaman=temp2(ran,2);
dars=temp2(ran,3);
ostad=temp2(ran,4);
index=temp2(ran,7);
if kelas>=1 && kelas<=11
    ran_kelas=randi([1 11],1);
    ran_zaman=randi(30,1);
    while ~isempty(x{ran_kelas,ran_zaman})
        ran_kelas=randi([1 11],1);
        ran_zaman=randi(30,1);
    end
elseif kelas>=12 && kelas<=14
    ran_kelas=randi([12 14],1);
    ran_zaman=randi(30,1);
    while ~isempty(x{ran_kelas,ran_zaman})
        ran_kelas=randi([12 14],1);
        ran_zaman=randi(30,1);
    end
elseif kelas>=15 && kelas<=16
    ran_kelas=randi([15 16],1);
    ran_zaman=randi(30,1);
    while ~isempty(x{ran_kelas,ran_zaman})
        ran_kelas=randi([15 16],1);
        ran_zaman=randi(30,1);
    end
end
```

شعبه‌شناسی و مطالعات فرهنگی
پرتال جامع علوم انسانی

```

elseif kelas>=17 && kelas<=24
    ran_kelas=randi([17 24],1);
    ran_zaman=randi(30,1);
    while ~isempty(x{ran_kelas,ran_zaman})
        ran_kelas=randi([17 24],1);
        ran_zaman=randi(30,1);
    end
z=x;
z{kelas,zaman}=[];
z{ran_kelas,ran_zaman}=[dars ostad index];
temp2(ran,1)=ran_kelas;
temp2(ran,2)=ran_zaman;
temp2(ran,5)=ceil(ran_zaman/(n_zaman-1));
temp2(ran,6)=cell2mat(mahdodiyat(temp2(ran,5)+1,ostad+1));
c=y;
j=1;
for i=1:nF:numel(c)
    c(1,i:(i+nF)-1)=temp2(j,:);
    j=j+1;
end

```

۵- **تابع رقص دوم:** این تابع برای رسیدن به مقصود اصلی این برنامه یعنی پیدا کردن یک نقطه بهینه با توجه به ارضا کردن کل محدودیت‌های سخت هست. در این مقاله، هدف اصلی برآورد کردن محدودیت‌های استادان و استفاده کردن از تمام زمان حضور آنها در دانشگاه هست. پس این تابع هر موجودیت را به مکانی مناسب در ماتریس مکان-زمان انتقال می‌دهد که استاد آن درس در آن زمان در دانشگاه حضور دارد به طوری که در نهایت هیچ‌یک از محدودیت‌های سخت را نقض نکرده باشند.

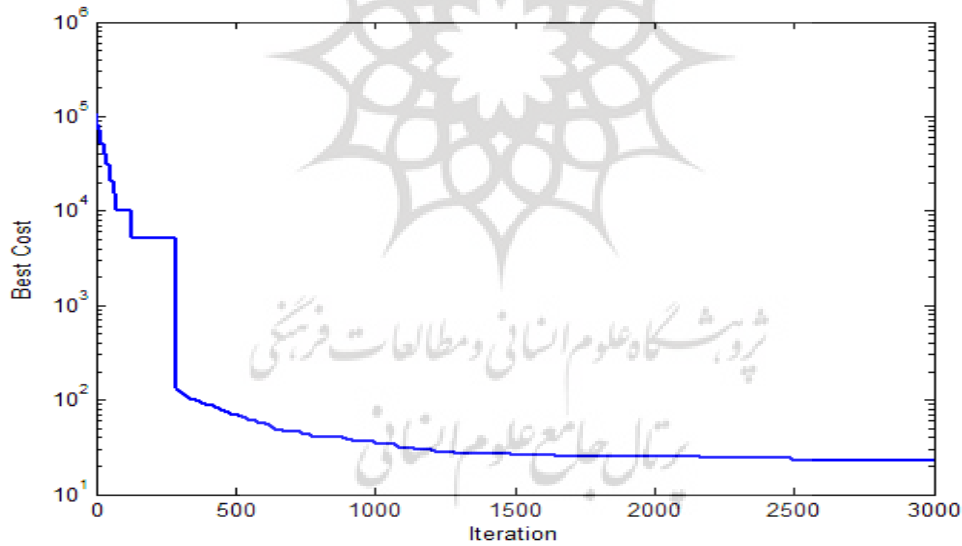
۶- **نتایج الگوریتم:** پس از اجرا کردن برنامه با ۳۰۰۰ بار تکرار حلقه اصلی برنامه و با جمعیت اولیه پنج زنبور عسل نتایج زیر به دست آمده است که در شکل (۵) و شکل (۶) نشان داده شده است. همان‌طور که در شکل (۵) مشاهده می‌شود ۲۳ موجودیت در جای مناسب خود قرار نگرفته‌اند. در واقع از ۳۷۰ جایگاه صحیح تنها ۲۳ جایگاه اشتباه است این یعنی الگوریتم به دقت حدود ۹۴ درصد رسیده است.

```

Command Window
Iteration 2977: Best Cost = 23 ghaid = 0 cost = 23 time= 7359.8648
Iteration 2978: Best Cost = 23 ghaid = 0 cost = 23 time= 7362.3852
Iteration 2979: Best Cost = 23 ghaid = 0 cost = 23 time= 7364.8857
Iteration 2980: Best Cost = 23 ghaid = 0 cost = 23 time= 7367.4833
Iteration 2981: Best Cost = 23 ghaid = 0 cost = 23 time= 7369.9768
Iteration 2982: Best Cost = 23 ghaid = 0 cost = 23 time= 7372.8704
Iteration 2983: Best Cost = 23 ghaid = 0 cost = 23 time= 7375.3416
Iteration 2984: Best Cost = 23 ghaid = 0 cost = 23 time= 7377.9704
Iteration 2985: Best Cost = 23 ghaid = 0 cost = 23 time= 7380.6636
Iteration 2986: Best Cost = 23 ghaid = 0 cost = 23 time= 7383.2738
Iteration 2987: Best Cost = 23 ghaid = 0 cost = 23 time= 7385.7144
Iteration 2988: Best Cost = 23 ghaid = 0 cost = 23 time= 7388.2324
Iteration 2989: Best Cost = 23 ghaid = 0 cost = 23 time= 7390.8224
Iteration 2990: Best Cost = 23 ghaid = 0 cost = 23 time= 7393.345
Iteration 2991: Best Cost = 23 ghaid = 0 cost = 23 time= 7395.7879
Iteration 2992: Best Cost = 23 ghaid = 0 cost = 23 time= 7398.3665
Iteration 2993: Best Cost = 23 ghaid = 0 cost = 23 time= 7401.0072
Iteration 2994: Best Cost = 23 ghaid = 0 cost = 23 time= 7403.4676
Iteration 2995: Best Cost = 23 ghaid = 0 cost = 23 time= 7405.9902
Iteration 2996: Best Cost = 23 ghaid = 0 cost = 23 time= 7408.3664
Iteration 2997: Best Cost = 23 ghaid = 0 cost = 23 time= 7411.1137
Iteration 2998: Best Cost = 23 ghaid = 0 cost = 23 time= 7413.6315
Iteration 2999: Best Cost = 23 ghaid = 0 cost = 23 time= 7416.2679
Iteration 3000: Best Cost = 23 ghaid = 0 cost = 23 time= 7418.9761

```

شکل (۵) خروجی برنامه متلب برای میزان هزینه



شکل (۶) نمودار روند کاهش هزینه

نتیجه گیری

در این پژوهش با ارائه تعاریفی جامع در زمینه جدول زمانی درس‌های دانشگاهی و بررسی پژوهش‌های موجود در این زمینه و با توجه به فضای آموزشی عمومی در ایران، یک مدل برای زمان‌بندی درس‌های دانشگاهی ارائه شد.

آوارز-والدز و همکاران (۲۰۰۲)، امین‌توسی و همکاران (۲۰۰۵)، ایسمیولوا و همکاران (۲۰۰۷)، کاسمیکرو همکاران (۲۰۰۹)، آلاداگ و همکاران (۲۰۰۹)، چاو چین و همکاران (۲۰۱۱)، گناون و همکاران (۲۰۱۲)، میرندا و همکاران (۲۰۱۲)، مهرگان و همکاران (۲۰۱۲)، سابر و همکاران (۲۰۱۲)، عبدالله و همکاران (۲۰۱۲)، قرونی و همکاران (۲۰۱۳) برای حل مسئله زمان‌بندی امتحانات و زمان‌بندی درس‌های دانشگاهی از الگوریتم‌های فرا ابتکاری جفت‌گیری زنبور عسل، الگوریتم مم‌تیک ممنوعه با ساختار همسایگی چندگانه، الگوریتم حل تکاملی ماتیک و ژنتیک گروهی، الگوریتم جستجوی ممنوعه، الگوریتم شبیه‌سازی شده تبریدی، الگوریتم جستجوی محلی، مدل‌سازی ریاضی عدد صحیح، روش متدولوژی سیستم نرم، کلاس‌بندی فازی بهینه و مدل‌سازی چندهدفه ترکیبی استفاده کرده‌اند و نتایج آنها نشان می‌دهد که مسئله بر مبنای ترجیحات استادان و دانشگاه و به صورت تخصیص استاد به درس و درس به دوره‌های زمانی به صورت هم‌زمان در نظر گرفته و مدل‌سازی می‌شود و با فرض اینکه دانشجویان در هنگام ثبت‌نام حق انتخاب دارند، ترجیحات آنها در مدل‌های فوق، در نظر گرفته نشده است.

برای حل مدل از الگوریتم فرا ابتکاری زنبور عسل در ابتدا به شکل کلاسیک و در نهایت به شکل نو و جدیدی استفاده شده است. در بخش حل مدل با روش زنبور عسل از دو تابع مهم رقص برای فرار از بهینگی محلی و بهبود کارایی مسئله و در نهایت رسیدن به یک جواب قابل قبول استفاده شده است. برای ارزیابی الگوریتم، با استفاده از ترجیح‌های دانشگاه، استاد، دانشجو که از نمونه‌های داده شده از طرف دانشگاه پیام نور مرکز لامرد استفاده شده و نتایج نشان می‌دهد که در نهایت دقت مدل ریاضی کلاسیک ۸۹ درصد و دقت مدل ریاضی جدید ۹۴ درصد بوده و در مقایسه با وضعیت مدل ریاضی کلاسیک و عمومی شرایط بهتری دارد.

پیشنهاد‌های پژوهش

با در نظر گرفتن این موضوع که جدول زمانی درس‌ها هر دانشگاه، با توجه به قوانین داخلی آن دانشگاه و ساختار و مقررات آموزش عالی هر کشور متفاوت است، می‌توان مسائل فراوانی با ویژگی‌های جدید تعریف کرد. برای توسعه مدل‌های ارائه شده در این پژوهش، می‌توان با نقض مفروضات یا اضافه کردن مفروضات جدید و همچنین اضافه یا حذف کردن محدودیت‌های سخت و نرم، مانند اینکه روابط پیش‌نیازی، به دوره‌های زمانی تخصیص داده شوند و در نتیجه مدلی جدید متناسب با شرایط دانشگاه مورد مطالعه، طراحی کرد.

برای نمونه می‌توان با اضافه کردن محدودیت‌های زیر به‌عنوان محدودیت‌های نرم یا سخت، مدل‌های ارائه شده در این مدل را توسعه داد:

- ✓ درس‌ها با توجه به روابط پیش‌نیازی، به دوره‌های زمانی تخصیص داده شوند.
- ✓ نباید ساعات درسی استادان از حد تعیین شده تجاوز کند.
- ✓ میزان ساعت تدریس استادان به‌طور یکنواخت در طول هفته تخصیص داده شود (فشار کاری استادان).
- تحقیقات مربوط به رویکرد حل مسئله شامل توسعه و یا ترکیب الگوریتم‌های فرا ابتکاری دیگر است.
- ✓ برای حل مدل، می‌توان از ترکیب الگوریتم‌ها استفاده کرد مثلاً برای ساخت جمعیت اولیه از الگوریتم ممتیک استفاده کنیم.
- ✓ می‌توان مدل را به مسئله‌ای چندهدفه تبدیل و از الگوریتم‌های چندهدفه دیگر مانند الگوریتم شبیه‌سازی تبریدی چندهدفه (MOSA) استفاده کرد.

منابع

- Abdullah. S. & Turabieh. H. (2012). On the use of multi neighborhood structures within a Tabu-based memetic approach to university timetabling problems. *Information Sciences, Malaysia*, 146-168.
- Aladag, C. H.; Hocaoglu, G. & Basaran, M. A. (2009). The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem. *Expert Systems with Applications*, 12349–12356.
- Aladag, C. H. & Hocaoglu, G. (2007). A tabu search algorithm to solve a course timetabling problem. *Hacettepe Journal of Mathematics and Statistics*, 53- 64.
- Burke, E. K. & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 266–280.
- Gunawan, A.; Ng, K. M. & Poh, K. L. (2012). A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem. *Computers & Operations Research, Singapore*, 3074–3088.
- Head, C. & Shaban, S. (2007). A heuristic approach to simultaneous course/student timetabling. *Computers & Operations Research*, 919–933.
- Pham, D. T.; & Koç, E. (2010). Design of a two-dimensional recursive filter using the bee's algorithm. *International Journal of Automation and Computing*, 7 (3), 399-402.
- Qaurooni. D. & Akbarzadeh-T. M-R. (2013). Course timetabling using evolutionary operators. *Applied Soft Computing*, 2504–2514.
- Sabar, N. R.; Ayob, M.; Kendall, G. & Qu, R. (2012). A honey-bee mating optimization algorithm for educational timetabling problems. *European Journal of Operational Research*, 533-543.
- Tereshko, V. & Loengarov, A. (2005). Collective decision making in honey-bee foraging dynamics. *Computing and Information Systems*, 9 (3), 1.
- Wren, A. (1996). Scheduling, timetabling and rostering – A special relationship? In: *Burke and Ross*, 46–75.