مجله علمی-پژوهشی
رایانش نرم و فن آوری اطلاعات
جلد ۴، شماره ۱، سال ۱۳۹۴

یک مدل جدید برای مسئله جدول زمانی ترم بندی درسی دانشگاه با رهیافت برنامه‌ریزی آرمانی فازی و الگوریتم ژنتیک

شهرام سعیدی
دانشگاه آزاد اسلامی، واحد تبریز،گروه مهندسی کامپیوتر، تبریز، ایران، sh_saeidi@iaut.ac.ir

چکیده – مسئله جدول زمانی، ترمی بندی درسی دانشگاه، مسئله تخصیص زمان، منابع و امکانات موجود در محیط آموزشی دانشگاه می باشد به نحوی که ضمن رعایت محدودیت‌های حاکم بر محیط، از کیفیت و رضایتمندی مطلوبی نیز برای همگان برخوردار باشد. در این مقاله، با درنظر گرفتن شرایط حاکم بر دانشگاه آزاد اسلامی واحد تبریز، یک مدل برنامه ریزی ریاضی خطی چند هدفی جدید برای این مسئله ارایه شده و از روش برنامه ریزی آرمانی فازی برای تجمیع توابع هدف در جهت بیشنه نمودن حداقل میزان رضایت مندی این توابع استفاده گردیده است. با توجه به پیچیدگی غیرچند جمله‌ای –سخت مدل ارایه شده، یک الگوریتم ژنتیک نیز برای حل آن توسعه داده شده است. مقایسه نتایج حاصل از اجرای مدل بر روی داده‌های واقعی برگرفته از محیط، برتری آن از نظر میزان مطلوبیت جواب نهایی را نسبت به روش سنتی دستی برنامه ریزی ترمی دانشگاه نشان می‌دهد.

کلید واژه‌ها– مسئله جدول زمانی، برنامه ریزی خطی، برنامه ریز آرمانی فازی، الگوریتم ژنتیک.

۱. مقدمه

مسئله جدول زمانی، عبارت است از تخصیص بهینه تعداد محدودی از منابع به تعدادی رویداد با رعایت محدودیت‌های زمانی و مکانی حاکم بر محیط. این مسئله، به عنوان یک مسئله پرکاربرد در محیط‌های آموزشی مانند مدارس، آموزشگاه‌ها و دانشگاه‌ها و غیره برای برنامه‌ریزی ترمی درس‌ها و دوره‌ها مورد استفاده قرار می‌گیرد [۱].

محدویت‌های موجود در مسئله، معمولا از کمبود امکانات و منابعی مانند کلاس، استاد، زمان در دسترس، تجهیزات و غیره ناشی شده و به دو نوع محدودیت‌های سخت و نرم تقسیم می‌شوند. محدودیت های سخت به قواعدی اشاره دارد که رعایت آنها الزامی بوده و امکان‌پذیر بودن جواب‌ها را تضمین می‌نمایند. محدودیت‌های نرم، قواعدی هستند که به کیفیت جواب‌های به دست آمده اشاره دارند عدم رعایت آن‌ها، خدشه‌ای به امکان‌پذیری جواب وارد نمی‌سازد.

مسئله جدول زمانی برای یک محیط آموزشی مانند دانشگاه، عبارت خواهد بود از ارایه برنامه درسی یک گروه آموزشی خاص برای یک نیم سال تحصیلی که در آن درس‌های ارایه شده توسط استادان گروه به دانشجویان ورودی‌های مختلف آن رشته، در کلاس‌ها، کارگاه‌ها و آزمایشگاه‌های موجود تخصیص می‌یابد. برخی از محدودیت های سخت این مسئله عبارت خواهند بود از [۲]:

- یک استاد نمی‌تواند در یک زمان در دو کلاس درس حضور داشته باشد.
- در یک زمان نمی‌توان دو درس را در یک کلاس برگزار کرد.
- دروس مربوط به هر ورودی نباید با هم تداخل داشته باشند.

از محدودیت‌های نرم این مسئله می‌توان به موارد زیر اشاره نمود:

- وجود فاصله مناسب بین ساعات درسی.
- پراکنده شدن دروس مربوط به هر ورودی در طول هفته.
- ترجیحات استاد.
- ترجیحات گروه یا دانشکده.

مسئله جدول زمانی از درجه پیچیدگی غیرچندجمله‌ای–سخت برخوردار بوده [۱] و حل آن به روش‌های قطعی، به‌ویژه برای مسائل در ابعاد واقعی بسیار زمان‌گیر بوده و در عمل غیر

قابل استفاده می باشد. به همین دلیل، استفاده از روش‌های هوش مصنوعی و روشهای فراابتکاری برای حل این مسئله اجتناب‌ناپذیر به نظر می رسد [۳].

در این مقاله، یک مدل ریاضی برای مسئله جدول زمانی ترم‌بندی درس‌های یک گروه آموزشی در دانشگاه ارایه شده و روش فراابتکاری مبتنی بر الگوریتم ژنتیک توسعه داده شده است که در آن، ضمن رعایت محدودیت‌های سخت حاکم بر مسئله به عنوان محدویت‌های اصلی مدل، هر کدام از محدودیت‌های نرم مدل به عنوان یک تابع هدف برای مسئله لحاظ گردیده‌اند که با توجه به تعدد این نوع محدودیت‌ها، مدل ارایه شده یک مسئله چند هدفی خواهد بود که برای اجتماع توابع هدف آن، از رهیافت برنامه‌ریزی آرمانی فازی استفاده شده است.

در روش برنامه ریزی آرمانی فازی، یک تابع مطلوبیت فازی برای هر یک از توابع هدف تعریف می گردد که بر اساس آن، به ازای هر جواب به دست آمده، میزان دستیابی (ارضاء) هریک از توابع هدف با یک عدد فازی محاسبه و بیان می‌گردد. در نهایت جوابی انتخاب می شود که رضایت تمامی توابع هدف را با یک حداقل مقدار از پیش تعیین شده ای برآورد نماید. دلیل انتخاب این روش، سهولت پیاده سازی و کارایی بالای آن در تجمیع توابع هدف چندگانه است.

۲. مطالعه ادبیات موضوع

۲،۱. تعریف مسئله

همانگونه که در بخش اول بیان گردید، مسئله جدول زمانی، که در واقع یک مسئله تخصیص چند بعدی می باشد، عبارت-است از تعیین این که در چه زمانی، افراد و منابع خاصی در مکان معینی حضور داشته باشند. بنابراین، عوامل دخیل در یک مسئله جدول زمانی، عبارتند از افراد (دانشجویان و استادان)، منابع (ویدیو پروژکتور، تخته سیاه و غیره)، مکان(کلاس، آزمایشگاه، سایت کامپیوتر و غیره) و زمان (ساعات برگزاری هر درس). برنامه ترمی هر گروه آموزشی در دانشگاه که در هر نیم سال توسط مدیر گروه مربوطه ارایه می شود، نمونه ای از یک مسئله جدول زمانی است.

بدیهی است این تجمیع به گونه ای باید صورت پذیرد که باعث خدشه دار شدن محدودیت های حاکم بر مسئله و عوامل آن نشود. هدف اصلی در تهیه یک برنامه جدول زمانی، نه تنها اجتناب از تداخل و نقض محدودیت ها، بلکه تلاش برای ارتقا کیفیت آن از حیث افزایش میزان رضایتمندی عوامل دخیل در این مسئله نیز هست.

مطالعه کارهای پیشین و پژوهش های انجام شده در این زمینه نشان می‌دهد که محدودیت های حاکم در مسئله برای عوامل موجود در آن، تا حدود زیادی می تواند متاثر از جامعه مورد مطالعه و شرایط فرهنگی و اجتماعی آن باشد. بنابراین ارایه یک مدل جامع برای حل این مسئله که پاسخگوی نیازهای تمامی محیط های آموزشی در تمامی جوامع باشد، تقریبا غیر ممکن است و مدل های ارایه شده توسط پژوهشگران، بر اساس نیازهای حاکم و خواسته‌های همان محیط انجام گردیده است.

۲،۲. کارهای پیشین

با توجه به اهمیت و جایگاه مسئله جدول زمانی، توجه بسیاری از پژوهش‌گران همواره به آن معطوف بوده و مطالعه در این زمینه کماکان ادامه دارد. نخستین مطالعات در این زمینه، در [۴] و [۵] صورت پذیرفته است و در طول چهار دهه گذشته، مقالات زیادی در مجلات و کنفرانس‌ها ارایه شده است. قسمت عمده این پژوهش‌ها در سال‌های اولیه آن، توسط تکنیک‌های تحقیق در عملیات مانند تکنیک جریان شبکه، رنگ آمیزی گراف، برنامه ریزی عدد صحیح انجام گردیده است. با توجه به پیچیدگی زمانی غیرچند جمله‌ای-سخت مسئله، محققین عمدتا به دنبال استفاده از روش‌های ابتکاری مانند جستجوی ممنوع-(تابو)، شبیه سازی تبرید، شبکه های عصبی، الگوریتم ژنتیک و غیره برای حل مدل پیشنهادی خود بوده‌اند. به عنوان پیشینه تحقیق می‌توان به موارد زیر اشاره نمود:

میسلیز و شارف[۶]، مسئله جدول زمانی را برای کارمندان یک موسسه مدل‌سازی نموده و روشی برای حل آن ارایه نمودند. بورکه و همکارانش در [۷]، گوو و همکارانش در [۸] مسئله جدول زمانی را برای تنظیم برنامه امتحانات درسی دانشگاهی مورد بررسی قرار داده و روش‌هایی برای حل آن توسعه دادند.

تامسون و دوسلند [۹]، گونه های متفاوت روش شبیه سازی تبرید را در حل مسئله جدول زمانی مورد بررسی قرار دادند. ملیسیو و همکارانش [۱۰] از روش شبیه سازی تبرید برای حل مدل پیشنهادی خود در مسئله جدول زمانی استفاده نمودند.

شارف و دی‌گاسپرو [۱۱] از روش‌های جستجوی محلی برای حل مسئله جدول زمانی مدارس استفاده کردند. ویلمن در پایان‌نامه دکتری خویش [۱۲]، مسئله تدوین جدول زمانی را برای یک مدرسه مورد بررسی قرار داد. سوو و سانگ[۱۳] از

روش الگوریتم ژنتیک برای حل مسئله جدول زمانی درس‌های هفتگی دانشگاهی استفاده نموده و الگوریتم پیشنهادی خود را مورد آنالیز قرار دادند.

سوچا و همکارانش [۳]، از روش بهینه سازی کولونی مورچگان برای حل مسئله جدول زمانی ترم بندی دانشگاهی استفاده نمودند. همچنین بورکه و همکارانش [۱۴] مسئله جدول زمانی را برای ترم بندی درس‌های دانشگاهی مورد بررسی قرار دادند.

شوشانا در پایان نامه کارشناسی ارشد خود [۱۵]، به تعریف، مدل سازی و حل مسئله جدول زمانی برای دانشکده مهندسی و علوم کاربردی دانشگاه تورنتو پرداخت. او از روش‌های تحقیق در عملیات و برنامه ریزی خطی در ارایه مدل پیشنهادی خود استفاده نمود.

چادهوری و دب [۱۶]، مسئله جدول زمانی را برای ترم بندی درس‌های دانشگاهی مورد توجه قرار داده و روشی ابتکاری بر اساس ژنتیک فازی برای آن توسعه دادند. آنها خروجی به دست آمده توسط روش خود را با خروجی تولید شده به‌طور دستی توسط کارمندان در کالج سنت خاویر هند مورد مقایسه قرار داده و برتری روش خود را نتیجه گرفتند.

مونترو و همکارانش [۲]، از روش بهینه‌سازی توده ذرات (PSO) برای حل مسئله جدول زمانی ترم بندی درس‌های دانشگاهی استفاده کردند. اولیویرا و ریس [۱۷]، در پژوهش خود، مسئله جدول زمانی را به هشت زیر مسئله تقسیم نموده و بر آن اساس، یک زبان جدید به نام UniLang برای بیان مسایل جدول زمانی ارایه نمودند. این زبان کاربر را قادر می‌سازد تا داده ها، محدودیت ها، معیارهای اندازه گیری کیفیت را به طور شفاف و دقیق بیان نماید.

همچنین فاهریون و دولانسکی [۱۸] و کانگ و وایت [۱۹]از روش برنامه نویسی منطقی ، اولیویرا و ریس [۲۰]، ریس و همکارانش [۲۱]، استاماتوپولوس و همکارانش [۲۲]، از روش برنامه نویسی منطقی محدودیت، گودس و همکارانش [۲۳]، از سیستم خبره برای حل و پیاده سازی مسئله جدول زمانی استفاده نمودند.

جات و یانگ [۲۴]، یک الگوریتم ژنتیک هدایت شونده برای حل مسئله جدول زمانی دانشگاه ارایه نمودند که با روش جستجوی محلی ترکیب شده بود. روش جستجوی هدایت شونده برای برای تولید فرزندان و روش جستجوی محلی برای ارتقا کیفیت جواب های تولید شده مورد استفاده قرار می گرفت.

ناجی عظیمی [۲۵]، مسئله جدول زمانی امتحانات دانشگاه را مورد بررسی قرار داده و چهار روش فرا ابتکاری جستجوی ممنوع، شبیه سازی تبرید، الگوریتم ژنتیک و کولونی مورچگان را برای حل آن مورد مقایسه قرار داده و نتیجه گرفت که سیستم کولونی مورچگان و جستجوی ممنوع، بهتر از روش‌های دیگر عمل می‌کنند.

ژانگ و لائو [۲۶]، روشی برای حل مسئله جدول زمانی درسی دانشگاه با رهیافت برنامه ریزی ارضای محدودیت ارایه نمودند. آنها روش خود را بر روی یک مطالعه موردی آزمایش نموده و برای حل آن، از نرم افزار ILOG استفاده کردند.

موکوس و پاپیکین [۲۷]، با استفاده از الگوریتم شبیه سازی تبرید، روشی برای حل مسئله جدول زمانی درسی دبیرستان ارایه نمودند که پارامترهای آن، از روش بیزی به‌دست می‌آمد. آن ها همچنین روش خود را به صورت یک نرم افزار مبتنی بر وب، برای استفاده عملی دبیرستان‌ها توسعه دادند.

با مطالعه پژوهش‌های پیشین و روش‌های مورد استفاده در آن مقالات، می‌توان این روش‌ها را به‌صورت زیر دسته بندی نمود:

- *روش‌های برنامه‌ریزی ریاضی و تحقیق در عملیات:* این روش‌ها به مدل‌سازی ریاضی مسئله و استفاده از تکنیک‌های شناخته شده حل آن می‌پردازند. با توجه به پیچیدگی زمانی مدل‌های بدست آمده، روش‌های حل کلاسیک (مثلا روش سیمپلکس، شاخه و کرانه و غیره) چندان کارآمد نیستند.
- *استفاده از روش‌های فراابتکاری عام:* این روش‌ها، به عنوان یک ابزار کارآمد برای حل مدل‌های ریاضی پیچیده به‌طور گسترده مورد توجه بوده و استفاده گردیده‌اند.
- *استفاده از روش‌های ابتکاری خاص:* برخی محققین، برای حل مدل پیشنهادی خود، روشی ابتکاری ابداع نموده و در حالت خاص، آن را مورد استفاده قرار داده‌اند.
- *روش‌های هوش مصنوعی و برنامه نویسی منطقی:* این روش کمتر از سایر روش‌ها مورد توجه واقع گردیده است. در زبان‌های برنامه نویسی منطقی، مانند پرولوگ، که بر مبنای منطق صوری استوار هستند، برنامه به‌صورت مجموعه ای از قواعد و جملات منطقی نوشته می‌شود. معمولا از روش قیاس معکوس برای اثبات فرضیه ها با استناد به مشاهدات استفاده می‌شود.

بررسی این روش‌ها نشان می‌دهد که هیچکدام از محققین، از

روش برنامه ریزی آرمانی فازی برای حل مسئله جدول زمانی استفاده ننموده اند. همچنین، در پیاده سازی الگوریتم ژنتیک، جهت تسریع همگرایی الگوریتم و کاهش جواب‌های غیر موجه، از روش‌های ابتکاری در ساخت کروموزوم‌ها، تولید جمعیت اولیه و نحوه انجام عملگر جهش استفاده گردیده است که شرح آن در بخش‌های ۴,۲، ۴,۳ و ۴,۵ آورده شده است.

۳. مدل ریاضی پیشنهادی

در این بخش به ارایه یک مدل ریاضی برای بیان مسئله جدول زمانی ترم بندی دانشگاه پرداخته می‌شود. قبل از ارایه مدل، لازم است فرض‌ها و شرایط(محدودیت‌ها) حاکم بر محیط مورد بررسی قرار گیرد. این محدودیت‌ها، همانگونه که قبلا نیز اشاره گردید، به دو نوع محدودیت‌های سخت و محدودیت های نرم تقسیم می‌شوند. نوع اول، محدودیت‌هایی هستند که لازم الاجرا بوده و عدول از آنها مجاز نیست. نوع دوم، محدودیت‌هایی هستند که تا حد ممکن بایستی برآورده شوند و میزان ارضاء آنها، مقدار مطلوبیت جواب نهایی (کیفیت جواب) را نشان می‌دهد.

۳,۱ فرض‌های مدل

در ارایه مدل ریاضی برای مسئله جدول زمانی ترم بندی دانشگاه فرض می‌شود:

- تعداد کلاس‌ها و استادان در طول یک نیم‌سال تحصیلی ثابت می‌ماند.
- یک استاد می‌تواند مجموعه‌ای از درس‌ها را تدریس نماید.
- یک درس می‌تواند توسط بیش از یک استاد در یک نیم سال تحصیلی ارایه شود.
- درس های مورد علاقه برای تدریس، روز و ساعات حضور در دانشگاه توسط هر استاد به مدیر گروه مربوطه اعلام می‌شود.
- زمان های تلف شده(هدر رفته) استاد، بسیار بیشتر از زمان‌های تلف شده دانشجویان ارزش دارد.

۳,۲ محدودیت‌های سخت

تمامی درس‌های مربوط به یک ترم(ورودی) برای یک رشته خاص باید ارایه شود.

درس‌های ارایه شده در طول هفته برای یک ورودی رشته خاص، نباید تداخل داشته باشد.

درس‌های ارایه شده در یک روز و ساعت و کلاس خاص، نباید تداخل داشته باشند.

درس‌های ارایه شده برای یک استاد، نباید تداخل داشته باشد.

حداقل(موظفی) و حداکثر(سقف) ساعت تدریس یک استاد در طول هفته رعایت شود.

حداقل تعداد روزهای حضور استاد در دانشگاه در طول هفته رعایت شود.

هر درس، در مکان(کلاس، آزمایشگاه، سایت) مناسب خود بایستی ارایه گردد.

۳,۳ محدودیت‌های نرم

- گپ (فاصله زمانی خالی) بین کلاسی دانشجویان حداقل شود [۲۸].
- گپ بین ساعات تدریس هر استاد حداقل شود.
- تعداد روزهای حضور دانشجو در دانشگاه حداقل شود.
- تراکم برنامه درسی ارایه شده در طول هفته (بار کاری روی کلاس‌ها) متعادل شود.

۳,۴ پارامترهای مدل

I: تعداد کل درسها برای تمامی گرایش‌های یک گروه آموزشی.

P: تعداد کل استادان گروه آموزشی (اعم از هیات علمی یا مدعو) در یک نیم سال تحصیلی.

J: تعداد گرایش‌های (رشته‌ها) موجود در یک گروه آموزشی.

T_j: تعداد کل نیم‌سال‌های تحصیلی برای رشته j.

C: تعداد مکان های آموزشی اعم از کلاس، آزمایشگاه و سایت.

D: تعداد روزهای کاری هفته که می‌توان کلاس درس برگزار کرد.

S: تعداد تایم (اسلات) موجود در هر روز هفته برای ارایه درس.

U_i: تعداد واحد درسی درس i ام.

$MinTeach_p$: حداقل (موظفی) تدریس استاد p .

$MaxTeach_p$: حداکثر (سقف) تدریس استاد p .

$MaxTPD$: حداکثر ساعات مجاز تدریس استاد در یک روز.

$MinPPD$: حداقل تعداد روزهای حضور استاد در دانشگاه.

Sum_{jt}: تعداد درس‌های ترم t رشته j .

P_{ip}: پارامتر باینری نشان دهنده این که آیا استاد p درس i را تدریس می کند؟

E_{pds}: پارامتر باینری نشان دهنده این که آیا استاد p ، در روز d ، تایم s را وقت حضور خود اعلام نموده است؟

α_{ic}: پارامتر باینری نشان دهنده این که آیا درس i می‌تواند در مکان c برگزار شود؟

β_{ijt}: پارامتر باینری نشان دهنده این که آیا درس i متعلق به ترم t رشته j است؟

۳,۵. متغیر تصمیم

تنها متغیر تصمیم مدل، یک متغیر باینری به شرح زیر است:

X_{picds}: آیا درس i، توسط استاد p، در روز d، تایم s، در مکان c ارایه بشود؟

۳,۶. توابع هدف

مدل ارایه شده برای مسئله در این پژوهش، یک مدل چند هدفی می‌باشد. توابع هدف تعریف شده در مدل پیشنهادی، در واقع همان محدودیت های نرم تعریف شده در بخش ۳,۳ هستند که در ذیل به بیان ریاضی هر یک از آن‌ها پرداخته می‌شود.

۳,۶,۱. حداقل نمودن فواصل زمانی خالی (گپ) بین کلاس‌های دانشجویان هر ورودی

درس‌هایی که در یک روز برای دانشجویان یک ورودی از یک رشته ارایه می‌شوند، حتی الامکان باید پشت سر هم بوده و فاصله زمانی بین دو درس حداقل شود. از طرفی چون دانشجویان یک ورودی خاص، به احتمال زیاد بیش از یک روز هفته در دانشگاه کلاس خواهند داشت، در تابع هدف اول، متوسط فواصل خالی بین کلاس‌ها در روزهای هفته محاسبه شده و حداقل شدن آن، یکی از آرمان‌های مورد نظر است. تعداد فاصله خالی موجود در هر روز و متوسط تعداد فواصل خالی در طول هفته با روابط زیر قابل محاسبه است:

ابتدا یک متغیر باینری کمکی به صورت زیر تعریف می‌کنیم:

Y_{jtds}: آیا برای ورودی (ترم) t رشته j در روز d و تایم s، درسی ارایه شده است؟

با توجه به پارامترها و متغیرهای اصلی تعریف شده قبلی، خواهیم داشت:

$$Y_{jtds} = \sum_{c}\sum_{i}\sum_{p} X_{picds}.P_{ip}.E_{pds}.\beta_{ijt}.\alpha_{ic} \qquad \forall\ d, s, j, t \quad (۱)$$

و با تعریف متغیرهای دیگری به‌صورت زیر به‌عنوان بزرگترین و کوچکترین تایم کلاسی موجود در یک روز برای هر ورودی از هر رشته به صورت زیر:

$$MaxSIS_{jtd} = \text{Max}(s) \mid Y_{jtds}=1 \qquad \forall\ d, j, t \quad (۲)$$

$$MinSIS_{jtd} = \text{Min}(s) \mid Y_{jtds}=1 \qquad \forall\ d, j, t \quad (۳)$$

Z_{jtd}: تعداد تایم های خالی روز d برای ورودی (ترم) t رشته j به کمک رابطه زیر قابل محاسبه است:

$$Z_{jtd} = \sum_{MinSIS_{jtd} < s < MaxSIS_{jtd}} (1 - Y_{jtds}) \qquad \forall\ d, j, t \quad (۴)$$

نهایتا خواهیم داشت:

$$f_1 = \overline{Z} = \frac{\sum_{j=1}^{J} \dfrac{\sum_{t=1}^{T_j}\sum_{d=1}^{D} Z_{jtd}}{T_j.D}}{J} \quad (۵)$$

رابطه (۵)، متوسط تعداد تایم های خالی کلیه ورودی‌ها از کلیه رشته‌ها در طول هفته را محاسبه می کند که به‌عنوان تابع هدف اول درنظر گرفته می‌شود که باید حداقل گردد.

۳,۶,۲. حداقل نمودن تعداد فواصل خالی (گپ) بین کلاس‌های یک استاد در یک روز

درس‌هایی که یک استاد در یک روز خاص تدریس می‌نماید، بهتر است به‌صورت متوالی و پشت سرهم بوده و حتی‌الامکان هیچ فاصله زمانی خالی بین آنها نباشد. از آنجایی که یک استاد ممکن است در بیش از یک روز هفته در دانشگاه حضور داشته باشد، و همچنین با توجه به فرض در نظر گرفته شده برای اهمیت بیشتر زمان استاد نسبت به دانشجو، به جای حداقل کردن متوسط تعداد این فضاهای زمانی هدر رفته استاد در طول هفته، بیشترین تعداد فاصله خالی بین دو کلاس یک استاد در روزهای مختلف هفته محاسبه شده و حداقل کردن متوسط این مقادیر به ازای تمامی استادان، به‌عنوان تابع هدف دوم مسئله تعریف شده است.

مشابه پارامترهای کمکی تعریف شده در بخش قبل، دو پارامتر کمکی جدید برای مشخص نمودن بزرگترین و کوچکترین شماره تایم کلاسی یک استاد در یک روز معین، به‌شرح زیر

تعریف می‌شود:

$$MaxSIP_{pd} = \mathrm{Max}(s) \mid \sum_{c}\sum_{i} X_{picds} = 1 \quad \forall\, p, d \qquad (۶)$$

$$MinSIP_{pd} = \mathrm{Min}(s) \mid \sum_{c}\sum_{i} X_{picds} = 1 \quad \forall\, p, d \qquad (۷)$$

و با استفاده از روابط فوق، تعداد تایم خالی یک استاد در روز خاص با رابطه (۸) محاسبه می‌شود:

$$Gap_{pd} = \sum_{MinSIP_{pd}<s<MaxSIP_{pd}} (1 - X_{picds}).E_{pds} \quad \forall\, p, d \qquad (۸)$$

نکته مهم آن که یک تایم به شرطی برای یک استاد خالی(تلف شده) محسوب می‌شود که استاد آن تایم را به‌عنوان وقت حضور خود به مدیر گروه اعلام نموده باشد، ولی هیچ کلاس درسی برای وی در آن زمان منظور نشده باشد. به‌همین دلیل جمله E_{pds} در رابطه فوق منظور شده است.

بزرگترین مقدار برای تعداد فاصله خالی برای هر استاد، از رابطه (۹) و متوسط این مقادیر به ازای کلیه استادان از رابطه (۱۰) قابل محاسبه است:

$$\mathrm{MaxGap_p} = \mathrm{Max}(PGap_{pd}) \quad \forall\, p \qquad (۹)$$

$$f_2 = \overline{MaxGap} = \frac{\sum_{p=1}^{P} MaxGap_p}{P} \qquad (۱۰)$$

۳,۶,۳. حداقل نمودن متوسط تعداد روزهای حضور دانشجویان در دانشگاه

یکی دیگر از آرمان‌های مدنظر در برنامه ریزی ترمی دانشگاه، حداقل کردن تعداد روزهای حضور دانشجویان در دانشگاه در طول هفته می باشد. از آن جایی‌که دانشجویان ورودی‌های مختلف از گرایش‌های مختلف، ممکن تعداد متفاوتی از روزهای هفته را در دانشگاه حضور داشته باشند، متوسط تعداد این روزهای حضور برای کلیه دانشجویان محاسبه گردیده و حداقل‌سازی آن به عنوان تابع هدف سوم در نظر گرفته می‌شود.

تعداد روزهایی که دانشجویان ورودی t رشته j در طول هفته در دانشگاه کلاس دارند، از رابطه زیر به‌دست می‌آید:

$$\mathrm{NumP_{jt}} = \sum_{d=1}^{D} \left(\prod_{s=1}^{S} Y_{jtds} \mid \mathrm{Y_{jtds}} \neq 0\right) \quad \forall\, j, t \qquad (۱۱)$$

در نتیجه متوسط تعداد روزهای حضور کلیه دانشجویان در طول هفته (تابع هدف سوم) برابر خواهد بود با:

$$f_3 = \overline{NumP} = \frac{\sum_{j=1}^{J}\sum_{t=1}^{T} NumP_{jt}}{J.T} \qquad (۱۲)$$

۳,۶,۴. تعادل تراکم کلاس‌های ارایه شده در طول هفته

آخرین هدف در نظر گرفته شده در مدل پیشنهادی، یکنواختی تعداد درس‌های ارایه شده توسط گروه آموزشی در طول هفته است که به معنای تعادل بار کاری استفاده از کلاس‌های درس و تایم های زمانی می‌باشد. منظور از بار کاری یک روز خاص، تعداد کل تایم‌های کلاسی است که در طول آن روز برگزار می‌شود. برای نیل به این هدف، بار کاری یک روز با رابطه (۱۳) و متوسط بار کاری روزهای هفته با رابطه (۱۴) محاسبه گردیده و نهایتا مقدار قدرمطلق انحراف از میانگین بار کاری هر روز هفته با رابطه (۱۵) محاسبه می‌شود که می‌بایست کمینه گردد.

$$\mathrm{WL_d} = \sum_{c=1}^{C}\sum_{s=1}^{S} X_{picds}.\alpha_{ic} \quad \forall\, \mathrm{p, i, d} \qquad (۱۳)$$

$$\overline{WL} = \frac{\sum_{d=1}^{D} WL_d}{D} \qquad (۱۴)$$

$$f_4 = \sum_{d=1}^{D} \left| WL_d - \overline{WL} \right| \qquad (۱۵)$$

۳,۷.   محدودیت های مدل

محدویت‌های موجود در مدل، همان محدویت‌های سخت هستند که در بخش ۳,۲ تعریف گردیدند. در این بخش به تعریف ریاضی این محدویدت‌ها پرداخته می‌شود.

۳,۷,۱. محدودیت اول

تمامی درس‌های مربوط به یک ترم(ورودی) برای یک رشته خاص باید ارایه شود.

$$Sum_{jt} = \sum_{d=1}^{D}\sum_{s=1}^{S}\sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{i=1}^{I} X_{picds}.\beta_{ijt}.\alpha_{ic} \quad \forall\, j,t \qquad (۱۶)$$

۳,۷,۲. محدودیت دوم

درس‌های ارایه شده در طول هفته برای یک ورودی رشته خاص، نباید تداخل داشته باشد.

$$\sum_{c=1}^{C}\sum_{p=1}^{P}\sum_{i=1}^{I} X_{picds}.\beta_{ijt}.\alpha_{ic} = 1 \qquad \forall\, j,t,d,s \qquad (۱۷)$$

۳,۷,۳. محدودیت سوم

درس‌های ارایه شده در یک روز و تایم و کلاس خاص، نباید تداخل داشته باشند.

$$\sum_{p=1}^{P}\sum_{i=1}^{I} X_{picds}.\alpha_{ic} = 1 \qquad \forall\, c,d,s \qquad (۱۸)$$

۳,۷,۴. محدودیت چهارم

درس‌های ارایه شده برای یک استاد، نباید تداخل داشته باشد.

$$\sum_{c=1}^{C}\sum_{i=1}^{I} X_{picds}.\alpha_{ic} = 1 \qquad \forall\, p,d,s \qquad (۱۹)$$

۳,۷,۵. محدودیت پنجم

حداقل(موظفی) و حداکثر(سقف) واحدهای تدریس یک استاد در طول هفته رعایت شود.

$$\sum_{d=1}^{D}\sum_{s=1}^{S}\sum_{c=1}^{C}\sum_{i=1}^{I} X_{picds}.U_i.\alpha_{ic} \geq \mathrm{MinTeach}_p \qquad \forall p \qquad (۲۰)$$

$$\sum_{d=1}^{D}\sum_{s=1}^{S}\sum_{c=1}^{C}\sum_{i=1}^{I} X_{picds}.U_i.\alpha_{ic} \leq \mathrm{MaxTeach}_p \qquad \forall p \qquad (۲۱)$$

۳,۷,۶. محدودیت ششم

حداقل تعداد روزهای حضور استاد در دانشگاه در طول هفته رعایت شود.

$$\sum_{d=1}^{D}\Big(\prod_{S=1}^{S}\prod_{c=1}^{C}\prod_{i=1}^{I} X_{picds} \mid X_{picds} \neq 0\Big) \geq MinP_{pd} \qquad \forall\, p \qquad (۲۲)$$

۳,۷,۷. محدودیت هفتم

هر درس، در مکان(کلاس، آزمایشگاه، سایت) مناسب خود بایستی ارایه گردد. این محدودیت، با درنظر گرفتن پارامتر باینری α_{ij} در محدودیت‌های پیشین، لحاظ شده است.

۳,۸. رهیافت برنامه ریزی آرمانی فازی

مدل پیشنهادی ارایه گردیده در این پژوهش، از نوع چند هدفی است و این اهداف، لزوما هم‌جهت نمی باشند. به عبارتی، بهینه سازی یکی از این اهداف، به معنای بهینه سازی سایر اهداف نخواهد بود. برای تجمیع این اهداف (آرمان‌ها)، از رهیافت برنامه ریزی آرمانی فازی استفاده شده است. در این روش، برای هر یک از توابع هدف، یک تابع عضویت فازی به‌شرح زیر درنظر می‌گیریم. با توجه به آن‌که کلیه توابع هدف تعریف شده در این تحقیق از نوع کمینه سازی هستند، مطلوبیت مقادیر بزرگ‌تر از مقدار بهینه هر تابع هدف، با افزایش مقدار تابع هدف کاهش یافته و نهایتا به صفر خواهد رسید. در نتیجه برای هر تابع هدف f_k (k=1,2,3,4) ، یک تابع عضویت فازی به‌صورت زیر می‌توان تعریف نمود:

$$\psi_{fk}(x) = \begin{cases} 1 & f_k(x) \leq f_k^* \\ 1 - \dfrac{f_k(x) - f_k^*}{d_k} & f_k^* \leq f_k(x) \leq f_k^* + d_k \\ 0 & f_k(x) \geq f_k^* + d_k \end{cases} \qquad (۲۳)$$

که در آن:

f_k^* : مقدار بهینه تابع هدف k ام و d_k انحراف قابل قبول از مقدار بهینه تابع هدف k ام است.

با استفاده از رابطه (۲۴)، توابع هدف مدل تجمیع گردیده و به‌صورت یک هدف واحد نوشته خواهد شد. این رابطه، به‌منظور بیشینه کردن حداقل مقدار مطلوبیت توابع هدف نوشته شده است.

$$\mathrm{Max}\ \lambda = \mathrm{Min}\ \{\psi_{f1}(x), \psi_{f2}(x), \psi_{f3}(x), \psi_{f4}(x)\} \qquad (۲۴)$$

در الگوریتم ژنتیک پیشنهادی، مقدار λ به عنوان میزان برازندگی هر کروموزوم مورد استفاده قرار خواهد گرفت. نهایتا مدل برنامه ریزی ریاضی پیشنهادی به صورت زیر خواهد بود:

$$\begin{aligned} &\mathrm{Max}\ \lambda = \min\{\psi_{f1}(x), \psi_{f2}(x), \psi_{f3}(x), \psi_{f4}(x)\} \\ &\text{s.t.} \\ &\qquad \lambda \leq \psi_{fk}(x) \qquad k=1, 2, 3, 4 \end{aligned}$$

به اضافه محدودیت‌های شماره (۱۶) الی (۲۲)

۴. الگوریتم ژنتیک

با توجه به حاصل‌ضرب متغیرهای باینری و درنتیجه غیرخطی بودن مدل ریاضی پیشنهادی و همچنین ماهیت غیر چند جمله‌ای-سخت مسئله جدول زمانی، استفاده از یک روش

فراابتکاری برای حل مسایل واقعی در ابعاد متوسط یا بزرگ، اجتناب ناپذیر خواهد بود. در این بخش، به ارایه یک الگوریتم ژنتیک برای حل مدل پیشنهادی مسئله جدول زمانی ترم بندی دانشگاه پرداخته می‌شود.

۴,۱. طراحی کروموزوم

الگوریتم ژنتیک، با اجتماعی از کروموزوم‌ها سروکار دارد که هر یک از آن‌ها، نمایش دهنده یک جواب برای مسئله است. هرچند که ممکن است این جواب غیر موجه بوده و یا درصورت موجه بودن، بهینه نباشد. با توجه به آن که تنها متغیر تصمیم مدل پیشنهادی، متغیر باینری X_{picds} می‌باشد، هر کروموزوم طوری طراحی می‌شود که دربرگیرنده اطلاعات مورد نیاز این متغیر (کدام استاد؟ #P، در چه روزی؟ #D، در چه تایمی؟ #T، در چه کلاسی؟ #C، چه درسی؟ #I) باشد. برای این منظور، ساختار کلی یک کروموزوم (به طول 2C+2) درنظر گرفته شده است که در شکل (۱) مشاهده می‌شود:

c_1	c_2	c_3		c_c	c_1	c_2		c_c		
P#	P#	P#	…	P#	I#	I#	…	I#	D#	T#

ژن C   ژن C

شکل ۱: ساختار یک کروموزوم

در این ساختار از چپ به راست، بخش اول به تعداد C ژن اول، دربرگیرنده کد استادان هستند که نشان می دهند کدام استاد در کدام کلاس حضور دارد. بخش دوم حاوی C ژن بعدی، حاوی کد درس‌ها هستند که نشان می دهند کدام درس در کدام کلاس ارایه می‌شود و بخش سوم، شامل دو ژن آخر، به ترتیب نشان دهنده شماره روز هفته و شماره تایم آن روز می‌باشند.

۴,۲. پیاده سازی الگوریتم ژنتیک

قالب کلی الگوریتم ژنتیک را می توان به صورت زیر بیان نمود [۲۹]:

۱. انتخاب جمعیت اولیه ای از کروموزوم‌ها

۲. تا زمانی که شرایط پایان محقق نشده‌اند:

تکرار

a. در صورت تحقق شرط آمیزش، انجام عملگر آمیزش با انتخاب تصادفی والدین و اعمال این عملگر

b. در صورت تحقق شرط جهش، انتخاب کروموزوم‌هایی از جمعیت، تعیین محل جهش و اعمال این عملگر

c. ارزیابی برازندگی فرزندان

پایان تکرار تا تولید فرزندان به اندازه کافی

انتخاب جمعیت جدید

۳. تکرار از مرحله ۲

در این الگوریتم، پس از تولید نسل اولیه با تعداد P کروموزوم به روش ابتکاری شرح داده شده در بخش ۴,۳ و محاسبه میزان برازندگی هریک از آن‌ها با استفاده از رابطه (۲۴)، حلقه تکرار اصلی برای تولید نسل‌های آتی آغاز شده و به تعداد N=100 مرتبه ادامه می‌یابد.

در مرحله a، انتخاب تصادفی والدین از جمعیت کنونی با استفاده از روش چرخ رولت انجام می‌گیرد. به تعداد نصف جمعیت، جفت کروموزوم به‌طور تصادفی انتخاب شده و به احتمال α عملگر تقاطع بر روی آنها انجام می‌گیرد تا کروموزوم های جدید (فرزندان) به‌دست آیند. مقدار پارامتر α برابر ۰/۸ تعیین شده است. بر روی فرزندان حاصل، به احتمال β عملگر جهش ژنی نیز صورت می‌گیرد. مقدار پارامتر β نیز برابر ۰/۲ در نظر گرفته شده است. میزان برازندگی و احتمال انتخاب کروموزوم g به‌عنوان والد، به روش چرخ رولت، به‌صورت زیر محاسبه می‌شود:

$$\text{Fitness}(g) = \lambda \qquad (۲۴)$$

$$\text{احتمال انتخاب کروموزوم } g = \frac{Fitness(g)}{\sum_{g=1}^{PopSize} Fitness(g)} \qquad (۲۵)$$

چرخ رولت، ابتدا یک عدد تصادفی یکنواخت پیوسته در بازه [۰-۱] تولید می کند. سپس کروموزومی را که مقدار احتمال تجمعی انتخاب آن، بزرگتر یا مساوی این عدد تصادفی باشد، به عنوان والد انتخاب می‌کند.

۴,۳. تولید جمعیت اولیه

جمعیت اولیه در الگوریتم ژنتیک، معمولا به‌صورت تصادفی تولید می‌شود. اما با توجه به پیچیدگی کروموزوم و ساختار مسئله و محدودیت‌های حاکم بر آن، اگر مقداردهی به ژن کروموزوم ها در جمعیت اولیه به‌صورت کاملا تصادفی انجام گیرد، بسیاری از جواب‌های تولید شده در این نسل، غیر موجه خواهند بود، چرا که ممکن است یک استاد در یک روز و ساعت خاص، همزمان به چندین کلاس منسوب شده یا درسی به استاد

غیر مربوطه واگذار شود. این وضعیت باعث تضعیف کارایی الگوریتم و کاهش سرعت همگرایی آن خواهد شد. بنابراین در تولید جمعیت اولیه و همچنین اِعمال عملگرهای متفاوت در طول اجرای الگوریتم، از روش‌های ابتکاری برای مقداردهی یا تغییر مقدار ژن‌ها استفاده شده است تا تعداد کروموزوم‌های غیر موجه به حداقل برسد. نکته مهم بعدی، تعیین تعداد کروموزوم‌ها در هر نسل، یا به عبارتی اندازه جمعیت می‌باشد. با توجه به آن‌که در جمعیت نهایی حاصل از اجرای الگوریتم ژنتیک، انتظار می‌رود که برای تمامی تایم ها در تمامی روزهای هفته، تمامی کلاس‌ها دارای کد درس و کد استاد موجهی باشند، بنابر این، جمعیت در نسل اولیه می‌بایست به تعداد D.S کروموزوم موجه داشته باشد. اگر فرض کنیم که $\frac{1}{K}$ کروموزم‌ها در هر نسل موجه هستند، پس اندازه جمعیت در نسل‌های آتی برابر خواهد بود با:

$$PopSize = K.D.S \qquad (۲۶)$$

روش ابتکاری تولید کروموزم‌ها در جمعیت اولیه چنین است:

با شروع از اولین روز هفته و اولین تایم روز:

ژن‌های بخش اول هر کروموزوم، به ترتیب با کد استادان مقدار دهی می‌شوند. کد استادی که انتخاب شود، از لیست استادان حذف می‌شود تا مجددا انتخاب نگردد.

ژن های بخش دوم با شروع از اولین خانه، بایستی با کد درس‌ها مقدار دهی شوند. در هر خانه (کلاس)، کد درسی نوشته می‌شود که استاد منسوب شده به آن کلاس (کد استاد در خانه(کلاس) متناظر در بخش اول کروموزوم وجود دارد)، آن درس را تدریس می‌نماید.

تکرار تا آخرین تایم در آخرین روز هفته.

۴،۴. عملگر تقاطع

عملگر تقاطع، بر روی دو کروموزوم انتخابی از نسل فعلی به روش چرخ رولت با احتمال α اعمال می‌شود. با توجه به ساختار سه بخشی هر کروموزوم، این عملگر (سه نقطه‌ای) بر روی هر سه بخش از والدین به‌طور مستقل انجام می‌گیرد تا فرزندان حاصل شوند.

۴،۵. عملگر جهش

عملگر جهش، با احتمالβ مقدار یکی از ژن‌های انتخاب شده تصادفی یک کروموزوم را تغییر می‌دهد. در الگوریتم ژنتیک کلاسیک، این تغییر مقدار به صورت کاملا تصادفی انجام می‌گیرد و در صورتی که منجر به تولید جواب بهتری گردیده باشد، کروموزوم جدید حاصل از جهش نگهداری شده و درغیر این-صورت حذف می‌شود.

در الگوریتم ژنتیک پیشنهادی، برای کاهش تعداد جواب‌های غیر موجه، عملگر جهش به روش ابتکاری زیر صورت می‌پذیرد:

اگر ژن انتخاب شده برای جهش، متعلق به بخش اول کروموزوم باشد، در این صورت می‌خواهیم کد استاد منسوب به یک کلاس را تغییر دهیم. کد استاد جدید به نحوی انتخاب می-شود که اولا در کروموزوم کنونی، استفاده نگردیده است (جلوگیری از انتساب همزمان استاد به دو کلاس متفاوت)، ثانیا، استاد انتخاب شده درس ارایه شده در آن کلاس را تدریس نماید. درصورت عدم امکان انتخاب استاد جدید، عمگلر جهش نادیده گرفته می‌شود.

اگر ژن انتخاب شده برای جهش، متعلق به بخش دوم کروموزوم باشد، به معنای آن است که می‌خواهیم کد درس منسوب شده به یکی از کلاس‌ها را تغییر دهیم. از آن جایی که استاد آن کلاس تغییر نمی‌کند، کد درس جدید به گونه‌ای انتخاب می‌شود که جزو درس‌های تدریسی استاد حاضر در آن کلاس باشد.

۵. پیاده سازی و نتایج محاسباتی

از آن جایی که مدل ریاضی ارایه شده در بخش سوم، از درجه پیچیدگی غیر چندجمله ای-سخت می‌باشد، از روش فراابتکاری الگوریتم ژنتیک به شرحی که در بخش چهارم گذشت، برای حل این مدل استفاده شده است. این الگوریتم، به زبان برنامه نویسی C در یک کامپیوتر شخصی تحت سیستم عامل Windows XP پیاده سازی گردیده و بر روی دو مثال جامع منتخب از محیط واقعی پیاده سازی گردیده است که در ادامه به‌شرح آن ها پرداخته شده است.

با توجه به توضیحات بخش (۴،۳)، با درنظر گفتن مقادیر $D=6$ و $S=6$ و همچنین با فرض آن که فقط ده درصد جواب‌های تولید شده موجه باشند، اندازه جمعیت در نسل‌ها برابر ۳۶۰ کروموزوم و مقدار احتمالات عملگرهای تقاطع و جهش به ترتیب برابر $\alpha=0.8$ و $\beta=0.2$ در نظر گرفته شده‌اند.

۵،۱. مثال اول

مثال اول برنامه ریزی ترمی دانشجویان مهندسی تکنولوژی سخت افزار گروه مهندسی کامپیوتر و فنآوری اطلاعات دانشگاه

آزاد اسلامی تبریز در نیم‌سال اول ۹۲-۹۱ می‌باشد که داده‌های ورودی مورد نیاز از مدیر گروه مربوطه اخذ گردیده است. با توجه به حجم عظیم داده‌های مربوط به درس‌های دانشجویان ورودی‌های مختلف، فضاهای آموزشی و همچنین اطلاعات مربوط به استادان تمام و قت و مدعو (اعم از لیست درس‌های تدریسی و روزهای پیشنهادی آنان) از درج آنها در مقاله خودداری شده است.

الگوریتم ژنتیک پیشنهادی، با دریافت این داده‌ها و اجرای مدل بر روی آنها عمل نموده و خروجی خود را تولید کرده است. نهایتا کیفیت برنامه ترمی (جواب) حاصل از الگوریتم پیشنهادی و برنامه ارایه شده توسط مدیر گروه به‌صورت دستی، مورد مقایسه قرار گرفته‌اند.

با توجه به توابع هدف چندگانه و نحوه تجمیع آنها که در بخش (۳،۸) توضیح داده است، در این قسمت ابتدا به تعیین مقادیر f^*_k و d_k پرداخته می‌شود. حداقل تعداد تایم‌های (گپ) خالی قابل قبول در یک روز برای دانشجویان و استادان، برابر صفر و حداکثر تعداد آن به‌ترتیب برابر ۲ و۳ درنظر گرفته شده‌اند. حداقل تعداد روزهای حضور دانشجو در هفته برابر ۲ روز و حداکثر آن برابر ۶ روز (کل ایام کاری هفته) فرض شده است. نهایتا حداقل و حداکثر مجموع قدرمطلق انحراف از بار کاری کلاس‌ها در طول هفته نیز به ترتیب برابر صفر و بیست درنظر گرفته شده است. این مقادیر در جدول (۱) آورده شده‌اند.

جدول ۱: مقادیر آستانه ای فازی توابع هدف

k	f^*_k	d_k
۱	۰	۳
۲	۰	۲
۳	۲	۴
۴	۰	۲۰

الگوریتم ژنتیک پیشنهادی با دریافت داده ورودی اجرا گردیده، مقادیر توابع هدف و مقادیر توابع عضویت فازی حاصل از آن به همراه ارزیابی نتایج برنامه ریزی درسی ارایه شده به‌صورت دستی توسط مدیر گروه به ترتیب در جدول‌های (۲) و (۳) آورده شده است. همان‌گونه که مشاهده می‌شود، کیفیت جواب نهایی (مقدار λ) در روش پیشنهادی بیشتر از روش دستی می‌باشد. مقدار ۰/۶ برای این جواب نشان می‌دهد که کلیه توابع هدف حداقل به میزان ۶۰ درصد مورد حصول واقع شده اند، در صورتی که این مقدار برای روش دستی، برابر ۴۵ درصد است.

نکته حایز اهمیت بعدی زمان مورد نیاز برای دست‌یابی به جواب نهایی است که در روش پیشنهادی، با احتساب زمان ورود داده ها به سیستم توسط اپراتور و اجرای برنامه، در کمتر از ۶۰ دقیقه قابل حصول است، درحالی که انجام این عمل به روش دستی معمولا چندین ساعت (یا روز) به‌طول می‌انجامد.

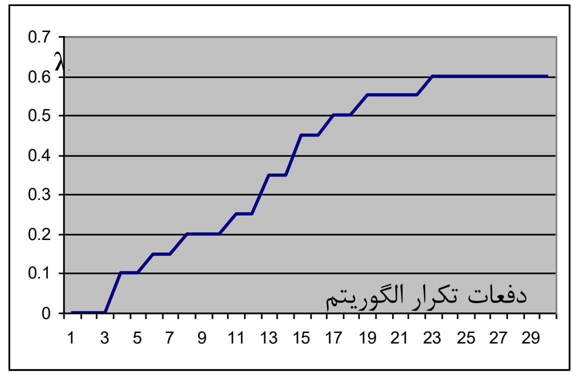
جدول ۲: مقادیر توابع هدف به دست آمده از الگوریتم ژنتیک و روش دستی در مثال اول

روش مورد استفاده	f_1	f_2	f_3	f_4
الگوریتم ژنتیک پیشنهادی	۰	۰	۳	۸
روش سنتی دستی	۰	۰	۴	۱۱

جدول ۳: مقایسه کیفیت جواب به دست آمده از الگوریتم ژنتیک و روش دستی در مثال اول

روش مورد استفاده	$\psi_{f1}(x)$	$\psi_{f2}(x)$	$\psi_{f3}(x)$	$\psi_{f4}(x)$	λ
الگوریتم ژنتیک پیشنهادی	۱	۱	۰/۷۵	۰/۶	۰/۶
روش سنتی دستی	۱	۱	۰/۵	۰/۴۵	۰/۴۵

همگرایی الگوریتم ژنتیک در دست‌یابی به جواب نهایی در شکل (۲) نمایش داده شده است که در آن، محور عمودی نشان دهنده مقدار λ و محور افقی، معرف دفعات تکرار الگوریتم است. همان‌گونه که مشاهده می‌گردد، روش پیشنهادی در تکرار ۲۳ ام به جواب نهایی دست یافته است.



شکل ۲: همگرایی روش پیشنهادی در مثال اول

۵،۲. مثال دوم

دومین مثال، برنامه ریزی ترمی دانشجویان کارشناسی نرم افزار کامپیوتر گروه مهندسی کامپیوتر و فن‌آوری اطلاعات دانشگاه آزاد اسلامی تبریز در نیم‌سال اول ۹۱-۹۰ می باشد مقادیر پارامترهای f^*_k و d_k مطابق جدول (۱) درنظر گرفته شده‌اند.

الگوریتم ژنتیک پیشنهادی با دریافت داده‌های ورودی اجرا گردیده و نتایج حاصل از آن به همراه ارزیابی نتایج برنامه ریزی درسی ارایه شده به‌صورت دستی توسط مدیر گروه در جدول های (۴) و (۵) آورده شده است. همان‌گونه که مشاهده می شود، کیفیت جواب نهایی (مقدار λ) در روش پیشنهادی بیشتر از روش دستی می باشد. مقدار ۰/۷ برای این جواب نشان می‌دهد که کلیه توابع هدف حداقل به میزان ۷۰ درصد مورد حصول واقع شده‌اند، در صورتی که این مقدار در روش دستی، برابر ۶۰ درصد است.

جدول ۴: مقادیر توابع هدف به دست آمده از الگوریتم ژنتیکو روش دستی در مثال دوم

روش مورد استفاده	f_1	f_2	f_3	f_4
الگوریتم ژنتیک پیشنهادی	۰	۰	۲	۶
روش سنتی دستی	۰	۰	۳	۸

جدول ۵: مقایسه کیفیت جواب به دست آمده از الگوریتم ژنتیک و روش دستی در مثال دوم

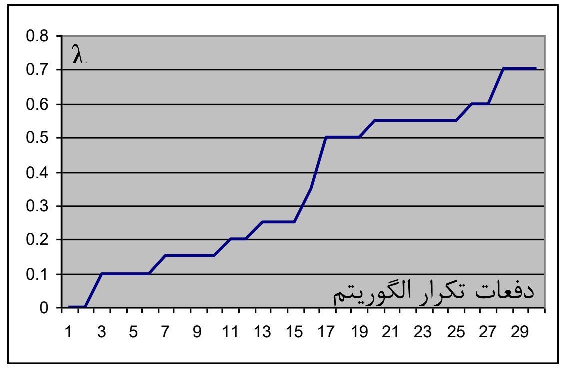
روش مورد استفاده	$\psi_{f1}(x)$	$\psi_{f2}(x)$	$\psi_{f3}(x)$	$\psi_{f4}(x)$	λ
الگوریتم ژنتیک پیشنهادی	۱	۱	۱	۰/۷	۰/۷
روش سنتی دستی	۱	۱	۰/۷۵	۰/۶	۰/۶

با توجه به نتایج به دست آمده در مثال دوم، میزان نیل به اهداف در جواب حاصل از الگوریتم ژنتیک پیشنهادی نسبت به روش سنتی دستی، هر چند در توابع هدف اول و دوم برابر است، اما در مقدار سایر توابع هدف و نهایتا در مقدار تابع هدف نهایی (λ)، از مطلوبیت بیشتری برخوردار است.

شکل (۳) نحوه همگرایی روش پیشنهادی را در رسیدن به مقدار نهایی تابع هدف در این مثال نشان می‌دهد. الگوریتم ژنتیک در تکرار ۲۸ ام به جواب نهایی دست یافته است.

۶. نتیجه‌گیری

در این مقاله، یک مدل ریاضی برای بهینه سازی و حل مسئله جدول زمانی ترم بندی درسی دانشگاه، با رهیافت برنامه ریزی آرمانی فازی ارایه گردید که با توجه به پیچیدگی زمانی آن، یک روش فراابتکاری مبتنی بر الگوریتم ژنتیک نیز به‌منظور حل مدل پیشنهادی توسعه داده شد که در پیاده سازی آن نیز از روش‌های ابتکاری استفاده گردیده است. الگوریتم ژنتیک بر روی دو مثال واقعی اقتباس شده از دو گروه آموزشی متفاوت دانشگاه آزاد اسلامی تبریز اجرا گردیده و نتایج حاصله در هر دو مثال، برتری روش پیشنهادی در دستیابی به جواب مسئله را نسبت به روش سنتی دستی به روشنی مشخص نمود.



شکل ۳: همگرایی روش پیشنهادی در مثال دوم

سپاسگزاری

این مقاله از طرح تحقیقاتی که با بودجه پژوهشی و حمایت مالی دانشگاه آزاد اسلامی واحد تبریز به انجام رسیده است، استخراج شده است که بدین‌وسیله مورد تقدیر و تشکر قرار می‌گیرد.

مراجع

[1] G. Schmidt, and T. Strohlein, "Timetable Construction - An Annotated Bibliography", *The Computer Journal*, 23(4), 307–316, 1980.

[2] E. Montero, M. Riff, L. Altamirano, "A PSO algorithm to solve a Real Course+Exam Timetabling Problem", *International conference on swarm intelligence*, 24(1), 2011.

[3] K. Socha,, M. Sampels, M. Manfrin, "Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art", *In Proceedings of Evolutionary Computation in Combinatorial Optimization*, volume 2611 of LNCS, pages 334–345. Springer-Verlag, 2003.

[4] J. Appleby, D. Blake, and E. Newman, "Techniques for Producing School Timetables on a Computer and Their Application to other Scheduling Problems" *The Computer Journal*, Vol. 3, pp.237-245, 1960.

[5] C. Gotlieb, "The Construction of Class-Teacher Time-Tables", *Proc.IFIP Cong, Munchen*, pp.73-77,1963.

[6] A. Meisels and A. Schaerf, "Modelling and Solving Employee Timetabling Problems", *Annals of Mathematics and Artificial Intelligence*, 39(1-2): 41–59, 2003.

[7] E.K. Burke, D. Elliman, P.H. Ford, R.F. Weare, "Examination Timetabling in British Universities: A Survey", *In Selected papers from the First International Conference on Practice and Theory of Automated Timetabling*, 76–90. Springer-Verlag, 1996.

[8] R. Qu, E.K. Burke, B. Mccollum, L.T. Merlot, S.Y. Lee, "A Survey of Search Methodologies and Automated System Development for Examination Timetabling", *Journal of Scheduling*, 12(1), 55–89, 2009.

[9] J. Thompson, and K.A Dowsland, "Variants of simulated annealing for the examination timetabling problem", *Annals of Operations Research*, 63, 105-128, 1996.

[10] F. Melicio, P. Calderia, A. Rosa, "Solving timetabling problem with simulated annealing", *Filipe, J. (ed.), Kluwer Academic Press*, 171-178, 2000.

[11] A. Schaerf, and L. DiGaspero, "Local Search Techniques for Educational Timetabling Problems", *In Proceedings of the 6th International Symposium on Operations Research*, 13–23, 2001.

[12] R. Willemen, "School Timetable Construction: Algorithms and Complexity", *PhD thesis, Technische Universiteit Eindhoven*, 2002.

[13] E. Yu, and K. Sung, "A Genetic algorithm for university weekly courses timetabling problem", *Intl. Trans. In OP. Res.*, 9, 703-717, 2002.

[14] E. Burke, D. deWerra, J. Kingston, Applications to Timetabling. In J. L. Gross and J. Yellen, editors, Handbook of Graph Theory, pages 445–474. CRC Press London, 2004.

[15] H.G Shoshana, "Defining, Modeling and Solving a Real University Course Problem", Master of science thesis, graduate department of mechanical and industrial engineering, university of Toronto, 2007.

[16] A. Chaudhuri and K. Deb, "Fuzzy genetic heuristic for university course Timetable problem", *Int. J. Advanced Soft Comput. Appl.* 2(1), 2010.

[17] L.P. Reis and E. Oliveira, "A Constraint Logic Programming Approach to Examination Scheduling", *AICS'99, Artificial Intelligence and Cognitive Science Conference*, Cork, Ireland, September 2000.

[18] R. Fahrion and G. Dollanski, "Construction of University Faculty Timetables Using Logic Programming Techniques", *Discrete Applied Mathematics*, Vol. 35, pp. 221-236, 1992.

[19] L. Kang and G. White, "A Logic Approach to the Resolution of Constraints in Timetabling", *European Journal of Operational Research*, Vol.61, pp.306, 317, 1992.

[20] L.P. Reis and E. Oliveira, "A Language for Specifying Complete Timetabling Problems", in Edmund Burke and Wilhelm Erben editors, Practice and Theory of Automated Timetabling III, Third International Conference, Vol. 2079, pp. 322-341, Berlin, 2001

[21] L.P. Reis, ,P. Teixeira, and E. Oliveira, "Examination Timetabling using Constraint Logic Programming", *ECP'99, 5th European Conference on Planning, Durham, U.K.*, 1999.

[22] O. Stamatopoulos, E. Viglas, and S. Karaboyas, "Nearly Optimum Timetable Construction Through CLP and Intelligent Search", *International Journal on Artificial Intelligence Tools*, Vol. 7, No. 4, pp. 415-442, 1998.

[23] E. Gudes, T. Kuflik and A. Meisels, "On Resource Allocation by an Expert System", *Engineering Applications of Artificial Intelligence*, Vol. 3, pp. 101-109, 1990.

[24] S.N. Jat, S. Yang, "A Guided search Genetic Algorithm for University Course Timetableing Problem", *Multidisciplinary International Conference on Scheduling: Theory and Applications*, MISTA, 2009.

[25] Z. NajiAzimi, "Comparision of Methaheuristic for Examination Timetabling problem", *J. Appl. Math. & Computing*, 16(1-2), 337–354, 2004.

[26] L. Zhang, S. Lau, "Constructing university timetable using constraint satisfaction programming approach", *International Conference on Computational Intelligence for Modelling, Control and Automation*, 2005.

[27] J. Mockus, L. Pupeikiene, "On Multi-Start Algorithms for Optimization of High School Timetables", *INFORMATICA*, 23(3), 405–425, 2012.

[28] D. Datta, K. Deb,C.M. Fonseca, "Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm", KanGAL Report, 2006.

[29] R.R. Colin, E.R.Jonathan, Genetic Algorithms-Principles and Perspectives, Kulwar Academic Publishers, UK, 2004.