Journal of Information Technology Management
ISSN: 2008-5893
EISSN: 2423-5059
Vol. 9, No. 3; PP. 531-548
Fall 2017

University of Tehran
Faculty of Management

Proposing Innovative Genetic Algorithms Model to Solve the Problem of the Professors' Educational Planning Considering Students' Opinions

Laleh Asgari [1], Mohammad Reza Keyvanpour [2]

Abstract: Timing of curriculum planning for students and faculty could be done using diverse methods. The present research concerns with curriculum planning for professors considering the students' opinions. In doing so, the courses and the timing are determined based on the professors' common timetable, the professors' intensive courses timing and the class limitations. To achieve this goal, the genetic algorithm methodology was used in two steps. In the first stage, single-point cutting operator was used and in the second stage of the algorithm, a new intelligent operator called cyclic reverse list (RIL) was used provided that gold, silver and bronze time types were used for different courses. The advantages of this algorithm are using a new appropriate function (hot rolled), as well as new criteria and a new operator (RIL). Unlike conventional methods, in this method the appropriateness is considered in proportion with the whole population and we try to remove the impossible solutions. The optimal solution is chosen from among a multitude of provided responses. Therefore, it was found that we can reach the optimal solutions with regard to a better appropriateness.

Key words: ***Fitness function, Genetic Algorithm, Mimetic Algorithm, Rotatory Inverse List (RIL), Timing.***

1. Msc, Social and Economic System Engineer, Specialized Data Mining Laboratory, Alzahra University, Tehran, Iran

2. Associate Prof. of Software Engineer, Faculty of Engineering, Alzahra University, Tehran, Iran

Submitted: 20 / December / 2016
Accepted: 09 / April / 2017
Corresponding Author: Laleh Asgari
Email: laasgari@gmail.com

مدیریت فناوری اطلاعات
Journal of Information Technology Management
دانشکدهٔ مدیریت دانشگاه تهران
دورهٔ ۹، شمارهٔ ۳
پاییز ۱۳۹۶
صص. ۵۴۸- ۵۳۱

ارائهٔ مدل ابتکاری الگوریتم ژنتیک برای حل مسئلهٔ برنامهٔ آموزشی استادان با تأمین نظر دانشجویان

لاله عسگری[۱]، محمدرضا کیوان‌پور[۲]

چکیده: زمان‌بندی در برنامه‌ریزی درسی دانشجویان و استادان با روش‌های متنوعی صورت می‌گیرد. این تحقیق به حل مسئلهٔ برنامهٔ آموزشی استادان با تأمین نظر دانشجویان می‌پردازد. در این مسئله، تخصیص درس و زمان به استادان با در نظر گرفتن ساعت جلسهٔ مشترک استادان و زمان‌بندی ساعات تدریس فشردهٔ آنان و محدودیت کلاس‌ها انجام می‌شود. بدین منظور، روش الگوریتم ژنتیک در دو مرحله به‌کار برده شده است. در مرحلهٔ اول الگوریتم، از عملگرِ برش تک‌نقطه‌ای استفاده شد و در مرحلهٔ دوم الگوریتم، عملگر هوشمند جدیدی به نام فهرست معکوس حلقوی با در نظر گرفتن زمان‌های طلایی، نقره‌ای و برنزی برای درس‌های مختلف به‌کار رفت. مزیت این الگوریتم استفاده از تابع برازندگی جدید و همچنین معیار انتخاب جدید و یک عملگر جدید است. این روش برخلاف روش‌های معمول، برازش کل جمعیت را در نظر می‌گیرد و تلاش می‌کند جواب‌های امکان ناپذیر را حذف کند. در این الگوریتم، جواب نهایی از جواب‌های بهینهٔ متعدد تولید شده انتخاب می‌شود. نتایج نشان داد این روش با برازش بهتری به جواب‌های بهینه می‌رسد.

واژه‌های کلیدی: الگوریتم ژنتیک، تابع برازندگی، زمان‌بندی، فهرست معکوس حلقوی، ممتیک الگوریتم.

۱. کارشناس‌ارشد مهندسی سیستم‌های اقتصادی- اجتماعی، آزمایشگاه تخصصی داده‌کاوی، دانشگاه الزهرا، تهران، ایران
۲. دانشیار مهندسی نرم‌افزار کامپیوتر، دانشکدهٔ فنی مهندسی دانشگاه الزهرا، تهران، ایران

تاریخ دریافت مقاله: ۱۳۹۵/۰۹/۳۰
تاریخ پذیرش نهایی مقاله: ۱۳۹۶/۰۱/۲۰
نویسندهٔ مسئول مقاله: لاله عسگری
E-mail: laasgari@gmail.com

مقدمه

زمان‌بندی درسی در ساده‌ترین جمله، یعنی تخصیص زمان و کلاس به استاد و درس. زمان‌بندی درس‌ها در جدول هفتگی، بر اساس معیارها و امکانات محیط، مشخصات درس، ساعات حضور استادان و درخواست دانشجویان صورت می‌گیرد. زمان‌بندی در حرکت قطار، هواپیما، مترو و درس دانشجویان، مسابقه‌های ورزشی، امتحان دانشجویان و همایش‌ها کاربرد دارد؛ به‌گونه‌ای که از تداخل زمانی دو برنامه با یکدیگر جلوگیری می‌کند و به رضایت کارفرما و ارباب‌رجوع می‌انجامد. از آنجا که در بیشتر دانشگاه‌های ایران، دانشجویان ارشد و دکتری در کنار اشتغالی که دارند، به تحصیل نیز می‌پردازند، برای تأمین رضایت دانشجویان بهتر است برنامه‌ای تنظیم شود که تعداد روزهای تحصیل دانشجوبان را فشرده و در روزهای متوالی تنظیم کند. همچنین برنامهٔ درسی استادان نیز در روزهای متوالی تنظیم شود تا ساعات پیوسته‌ای برای تحقیق آزاد شود. از سوی دیگر، در نظر گرفتن ساعت جلسهٔ مشترک برای استادان در روز مشترک موضوع مهمی است. هدف این تحقیق بهینه‌سازی زمان استادان و دانشجویان است که با ضرورت آزادسازی زمان روزانهٔ استادان و دانشجویان انجام می‌شود.

برای زمان‌بندی از روش‌هایی همچون برنامه‌ریزی عدد صحیح، برنامه‌ریزی منطقی[1] و هوش مصنوعی، تپه‌نوردی (HC)[2]، شبیه‌سازی تغییر دما (SA)[3]، الگوریتم ژنتیک، هایپر هیورستیک، متاهیورستیک، الگوریتم جست‌وجوی متوازن (HSA)[4]، الگوریتم جست‌وجوی ممنوعه[5]، سیل بزرگ (GDA)، جست‌وجوی همسایگی متغیر (VNS)[6]، تئوری مورچه و الگوریتم ژنتیک استفاده شده است. مزیت استفاده از الگوریتم ژنتیک نسبت به روش‌های دیگر، تولید جواب بهینه حتی در تعداد نسل‌های زیاد است.

پیشینهٔ پژوهش

یکی از روش‌های مشهور روش تپه‌نوردی (HC) است. این روش اجرای ساده و آسانی دارد. تله‌گذاشتن در نقطهٔ بهینه آسان است. روش HC می‌تواند برای تولید راه‌حل‌های با کیفیت خوب با کارهای دیگر مقایسه شود (اسمیت اسمان، ریورز و اسمیت، ۱۹۹۶). الگوریتم جست‌وجوی

---

1. Logic Programming
2. Hill Climbing
3. Simulated Aannealing
4. Harmony Search Algorithm
5. Tabu Search
6. Variable Neighborhood Search (VNS)

ممنوعه نیز از چند صفت حرکت همسایگی و چشم‌پوشی از راه‌حل‌هایی که در تابو لیست هستند، استفاده می‌کند (گلور، ۱۹۸۶؛ اسلونی، احمدی و کابرک، ۲۰۰۱).

روش شبیه‌سازی تغییر دما (SA) راه‌حل‌های بهتری ارائه می‌دهد یا راه‌حل‌های بدتر را با کم کردن احتمال در جست‌وجو پیدا می‌کند (کرک پاتریک، گلت و وچی، ۱۹۸۳). سیل بزرگ[1] نیز به روش SA کار می‌کند، اما از مرزی مانند SA به‌عنوان روابط استفاده می‌نماید که به ندرت کم می‌شود (داک، ۱۹۹۳). راه‌حل جست‌وجوی همسایگی متغیر (VNS)[2] بر پایهٔ استراتژی استفادهٔ بیشتر از ساختار همسایگی و تغییر آنها به‌طور سیستماتیک در طول جست‌وجو عمل می‌کند (عبداله، برک و مک‌کلام، ۲۰۰۵). از روش‌های دیگر، بهینه‌سازی کلونی مورچگان (ACO)[3] است (دوریگو و کارو، ۱۹۹۹). ACO از رفتار مورچه‌ها و روشی که آنها برای غذا کاوش می‌کنند، الهام می‌گیرد. کارهای زیادی با استفاده از روش ACO برای زمان‌بندی امتحانات در ادبیات انجام شده است (الی، ۲۰۰۶). الگوریتم ممتیک نیز ترکیب الگوریتم ژنتیک و الگوریتم جست‌وجوی محلی به‌واسطهٔ ترکیب کردن الگوریتم جست‌وجوی محلی است (براک، نوال و ویر، ۱۹۹۶).

اشی هارا، ساکاگوچی و یاماموری (۲۰۰۴) با استفاده از عملگرهای ژنتیک سنتی در مسئلهٔ زمان‌بندی و با طبقه‌بندی محدودیت‌ها، محدودیت‌ها را به سه گروه ضروری، شرطی مطلوب و شرطی دانشجویان دسته‌بندی کردند و با استفاده از تابع شایستگی خاص و عملگر crossover تک‌نقطه‌ای، به تولید جدول زمانی پرداختند.

هسو و چاو (۲۰۰۹)، به‌کمک روش ابتکاری و در نظرگرفتن محدودیت‌های زمانی دانشجو، بودجهٔ دانشگاه و منابع دانشکده‌ها و همچنین با استفاده از دو تابع شایستگی، نیاز دانشجویان در اخذ درس یا اخذ دوبارهٔ آن بدون ایجاد تأخیر در تحصیل را مشخص کردند و با در نظر گرفتن اندازهٔ کلاس‌ها به تخصیص منابع پرداختند.

در مسئله‌ای دیگر، ونگ، لیو و یو (۲۰۰۹) با استفاده از عملگر جدیدی به نام خودلقاحی[4]، محدودیت‌های ارائهٔ درس در اتاق مشخص و اضافه‌نشدن درس‌های تکراری به زمان‌بندی را لحاظ کردند و از استدلال مبتنی بر مورد[5] روش ذخیره و بازیابی راه‌حل‌های قبلی برای جلوگیری از تداخل درس‌ها و زمان‌ها بهره بردند.

عبدندهر و آلی (۲۰۰۷) با در نظر گرفتن محدودیت‌های زمان سخنرانی‌ها و برنامه‌ریزی زمان امتحانات و با روش برنامه‌نویسی منطقی و هوش مصنوعی به حل مسئله پرداختند.

---

1. Great deluge
2. Variable Neighborhood Search (VNS)
3. Ant Colony Optimization (ACO)
4. Self fertilization
5. Case based Reasoning

بهداد، دهقان و ذاکر تولایی (۱۳۸۵)، به‌کمک روش پیشرو، مجموع ساعت‌های بی‌استفادۀ دانشجویان و استادان را کمینه کردند و با استفاده از ترکیب عملگرهای crossover تک‌نقطه‌ای، دو نقطه‌ای و یکنواخت، به برنامه‌ریزی روی محدودیت‌های سخت پرداختند و با لحاظ کردن وزن به محدودیت‌ها، تابع شایستگی جدید به‌دست آوردند.

همر و موهوب (۲۰۱۴) معتقدند دیدگاه متاهیورستیک پیچیده (ETP) ، چند فاز برای حل مسئلۀ زمان‌بندی امتحان منظور می‌کند. این دیدگاه شامل فاز پیش پردازش، فاز ساختار و فاز پیشرفت است.

پیلا (۲۰۱۴) برای حل مسئله‌ای از الگوریتم درخت پارسه استفاده کرد. این الگوریتم به هر دانشجو هیورستیکی نسبت می‌دهد تا سختی زمان‌بندی عملی/ عمومی را برای این دانشجو نمایش دهد. الگوریتم به‌ترتیب هیورستیک و تخصیص به هر دانشجو اجرا می‌شود. هیورستیک به سه بخش تخصیص، درس و جلسه دسته‌بندی می‌گردد. نتایج این مطالعه نشان داده است هیورستیک تخصیص و ثانویه برای یافتن راه‌حل مسئله است.

رودوا، مولر و موری (۲۰۱۰) برای حل یک مسئلۀ برنامه‌نویسی برپایۀ محدودیت، روش‌های هایپر هیورستیک و روش‌های متاهیورستیک را معرفی کردند، به‌گونه‌ای که از هیچ استراتژی نمی‌توان انتظار داشت از یکدیگر بهتر اجرا شود. ویلیام و گومز بیان کردند که مسائل np-hard انعطاف‌ناپذیرند؛ به این معنا که الگوریتم کارایی وجود ندارد تا برای پیدا کردن راه‌حل بهینه برای انواع مشکلات تضمین شود (مارک و لوئیس، ۲۰۰۶).

وحید و محدحسین (۲۰۱۴) در پژوهشی، از الگوریتم خاصی به نام الگوریتم جست‌وجوی متوازن در پنج مرحله استفاده کردند که عبارت‌اند از: ۱. مقداردهی اولیه پارامترهای HSA و UCTP؛ ۲. مقداردهی اولیۀ حافظۀ هارمونی (HM) با زمان‌بندی امکان‌پذیر تصادفی بر پایۀ پارامترهای اندازۀ حافظۀ تصادفی (HMS)؛ ۳. تعبیه‌کردن راه‌حل هارمونی جدید؛ ۴. به‌روزرسانی حافظۀ هارمونی. ۵. متوقف کردن معیارها.

میتال، دوشی و سوناسرا (۲۰۱۵)، پژوهشی با هدف ایجاد جدول زمان‌بندی خودکار به‌وسیلۀ الگوریتم ژنتیک انجام دادند. آنها سیستمی را پیشنهاد کردند که در آن جدول زمان‌بندی به‌طور مکانیکی رأی انستیتو درست کند. ساختار جدول زمان‌بندی شامل داده‌های ورودی، رابطۀ بین داده‌های ورودی، محدودیت‌های سیستم و کاربرد الگوریتم ژنتیک می‌شود. ابتکار این روش الگوریتم تولید جدول زمانی اتوماتیک است.

شن جونگ، لین و پی کااو (۲۰۰۷)، در مقالۀ سیستم زمان‌بندی تطبیقی با الگوریتم ژنتیک، به مرتب‌کردن برنامه‌های آموزشی استخدامی پرداختند.

کاهار و کندال (۲۰۱۵) مسئلهٔ زمان‌بندی امتحانات را با هدف در نظر گرفتن محدودیت‌های سخت و نرم به‌کار بردند. آنان در این مقاله الگوریتم سیل بزرگ را توسعه‌یافته و تغییریافته معرفی کردند.

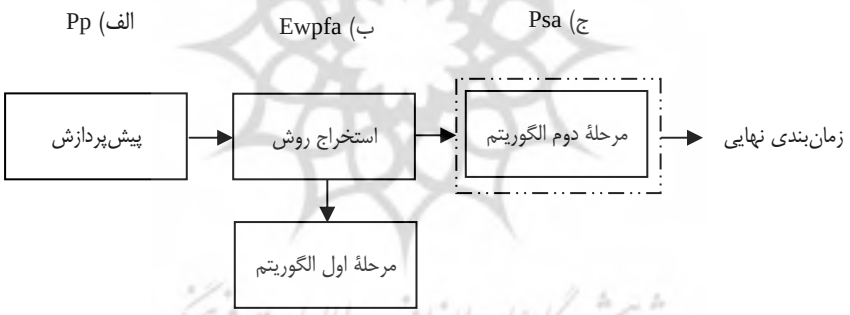
معمولاً یافتن راه‌حل امکان‌پذیر برای نیازهای زمان‌بندی، روش‌های مختلفی دارد و از کشوری به کشور دیگر، متفاوت است. چالش زمان‌بندی خودکار، وجود جواب‌های غیربهینهٔ متعدد یا به بیان دیگر، بالابردن دقت در حل مسئله است. در این تحقیق برای تأمین رضایت دانشجویان شاغل در حین تحصیل در مقاطع ارشد و دکتری دانشگاه‌های ایران، تنظیم برنامه‌ای که تعداد روزهای تحصیل دانشجویان را فشرده کرده و در روزهای متوالی چینش کند تا ساعات پیوسته‌ای برای تحقیق آزاد شود و ساعت جلسهٔ مشترک برای استادان در روز مشترک در نظر گرفته شود، ضروری به نظر می‌رسید. بدین منظور از عملگر جدید فهرست معکوس حلقوی[1]، استفاده شد. این روش، برازش را نسبت به کل جمعیت در نظر می‌گیرد و تلاش می‌کند جواب‌های امکان‌ناپذیر را حذف کند. به‌کمک این روش، در هر تکرار، جواب بهینه تولید شده و با این الگوریتم جدید فشرده‌سازی زمان‌بندی بدون تولید جواب ناکارآمد، امکان‌پذیر می‌شود. در ارتباط با تفاوت این روش با الگوریتم جست‌وجوی متوازن یا سایر روش‌های یادشده، می‌توان گفت که در این روش کلیهٔ جواب‌ها امکان‌پذیرند و با وزن‌دهی از بین آنها جواب بهینه انتخاب می‌شود، اما در الگوریتم جست‌وجوی متوازن، به‌صورت تصادفی جواب‌های امکان‌ناپذیر تولید شده و براساس ملاحظهٔ تصادفی و تنظیم پرتاب، حذف می‌شوند. سایر روش‌های اشاره شده نیز در تعداد تکرارهای بیشتر، با تولید جواب‌های امکان‌ناپذیر متعدد به جواب می‌رسند. از این رو الگوریتم‌هایی که سعی در کم کردن جواب‌های غیر بهینه دارند، برای مقایسه با الگوریتم RIL پیشنهاد می‌شوند. الگوریتم‌هایی مانند HC، SA، ممتیک الگوریتم، VNS جواب‌های غیربهینه تولید می‌کنند، اما نسبت به یکدیگر شدت و ضعف دارند.

مدل مفهومی

از آنجا که فشرده‌سازی زمان درس دانشجویان و استادان و نزدیک‌کردن انتخاب استادان و دانشجویان برای آزاد ساختن زمان مطالعه و کار همزمان دانشجویان مهم است، در این تحقیق با استفاده از الگوریتم ژنتیک دو مرحله‌ای و در نظر گرفتن محدودیت‌های عدم تداخل درس‌ها برای هر آرایش ترمی، کمینه‌شدن ساعات بی‌استفادهٔ دانشجویان، عدم تخصیص کلاس به چند درس همزمان و تداخل‌نداشتن دروس دانشجویان هم‌رشته و هم‌دوره، به حل مسئله پرداخته شده

1. Rotaory InverseList (RIL)

است. برای نیل به این منظور، روش الگوریتم ژنتیک در دو مرحله اجرا شده است؛ در مرحلهٔ اول الگوریتم، از عملگر برش تک‌نقطه‌ای استفاده می‌شود و در مرحلهٔ دوم، عملگر هوشمند جدیدی به نام فهرست معکوس حلقوی با در نظر گرفتن زمان‌های طلایی، نقره‌ای و برنزی برای دروس مختلف، به‌کار برده می‌شود. مزیت این الگوریتم استفاده از تابع برازندگی جدید و همچنین معیار انتخاب جدید و عملگر جدید است. چالش موجود در این راهکارها برای حل مسائل یادشده، نرسیدن به جواب بهینه، تولید جواب غیربهینه و تکرارهای زیاد برای رسیدن به جواب بهینه است. این روش، برازش را نسبت به کل جمعیت در نظر می‌گیرد و سعی در حذف جواب‌های امکان ناپذیر دارد؛ در صورتی که در روش‌های معمول، برازش برای هر کروموزوم انجام می‌شود (ایکو و همکاران، ۲۰۰۴؛ بهداد و همکاران، ۲۰۰۷؛ پیلا، ۲۰۱۴، جونگ و همکاران، ۲۰۰۷). این امر باعث می‌شود که در هر تکرار، جواب بهینه تولید شود و به‌کمک این الگوریتم جدید، امکان فشرده‌سازی زمان‌بندی بدون تولید جواب ناکارآمد فراهم آید. از این رو، جواب‌های بهینهٔ متعددی تولید می‌شود که با احتساب وزن و اولویت به جواب‌ها، جواب نهایی انتخاب می‌شود. بلوک دیاگرام روش پیشنهادی به شرح زیر است:



شکل ۱. دیاگرام زمان‌بندی برنامهٔ درسی با تأمین نظر دانشجویان در دو مرحله

روش‌شناسی پژوهش

پیش پردازش

مسئله به‌صورت مجموعهٔ $\{P, S, L, T\}$ تعریف می‌شود که در آن، P مجموعهٔ استادان، S مجموعهٔ دانشجویان، L مجموعهٔ دروس و T مجموعهٔ زمان‌های تدریس است. اولویت با استادان و دانشجویان است؛ دانشجویان باید در پیش ثبت‌نام انتخاب خود را مشخص کنند و استادان نیز برنامهٔ خود را ارائه دهند؛ سپس برنامهٔ درسی با در نظر گرفتن اشتراک انتخاب دانشجویان و انتخاب استادان و همچنین لحاظ‌کردن محدودیت کلاس‌ها تنظیم شود. در این برنامه، یک بازهٔ

زمانی در هفته برای استادان به‌منظور جلسهٔ مشترک، باز نگه داشته می‌شود. بدیهی است برای این بازه، زمان تدریس در نظر گرفته نمی‌شود. بعد از نگه‌داشتن دوره‌های زمانی باز، زمان‌هایی که باید برای تدریس هر استاد در نظر گرفته شود، لحاظ می‌شوند. روند کار زمان‌بندی[1] در شش مرحله تعیین می‌شود: ۱. برنامهٔ استادان؛ ۲. برنامهٔ آموزش؛ ۳. نزدیک کردن برنامهٔ استادان و آموزش؛ ۴. مشخص کردن روز مشترک؛ ۵. مشخص کردن ساعت مشترک برای جلسه یا دفاعیه؛ ۶. فشرده‌کردن برنامهٔ مشترک استادان و دانشجویان و توالی دروس دانشجویان.

استخراج روش

نمایش مسئله

روش نمایش مسئله به شکل ماتریس سه بعدی است. در این مسئله، ماتریس $T \times P \times L$ را در نظر می‌گیریم؛ به‌گونه‌ای که زمان در بعد k و نام استاد و نام درس در یک صفحه شامل بعدهای I و J در ماتریس لحاظ شوند.

رابطهٔ ۱) $\forall x \in T, \forall Y \in P, \forall Z \in L, T \leq 5$

رابطهٔ ۲) $P \leq 8$

رابطهٔ ۳) $L \leq 40$

رابطهٔ ۴) $W = 5D$

با توجه به اینکه ۵ روز از هفته برای تدریس و آموزش اختصاص داده شده است و هر روز شامل ۵ بازهٔ زمانی می‌شود، ۲۵ بازهٔ زمانی در هفته در نظر گرفته شده است.

تابع برازش

در مرحلهٔ اول الگوریتم، اگر انتخاب استادان P باشد، انتخاب آموزش L باشد و کروموزوم N عنصر داشته باشد، رابطهٔ ۵ برقرار می‌شود.

رابطهٔ ۵) $F_S = \frac{P.L}{N-2}$

N تعداد دروس قابل ارائه یا ارائه‌شده است.

در شرایطی که $F_S \geq 1$ باشد، جواب تأیید شده است و در غیر این صورت نسل‌ها تکرار می‌شود تا به جواب تأیید شده برسد. در مرحلهٔ دوم الگوریتم، اگر C درجهٔ تداخل برنامهٔ زمانی، S تعداد کلاس‌ها و L تعداد استادان باشد، رابطهٔ ۶ برقرار است.

---

1. Timetabling

$$F_S = \sum_{I=1}^{S} = \frac{(C_1)}{S.L_I}$$ رابطهٔ ۶)

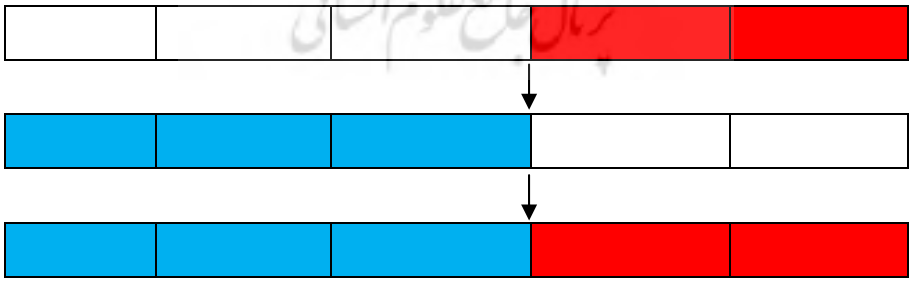
مارک و لوئیس (۲۰۰۶) به مقایسهٔ روش‌های متاهیورستیک در مسئلهٔ زمان‌بندی دروس پرداختند و شش نوع تابع برازش برای الگوریتم ژنتیک تعریف کردند. در این تحقیق برای محاسبهٔ تداخل‌ها از رابطهٔ ششم با تغییر توان دوم مجموع در آن به جای مجموع، استفاده شده است. دلیل این تغییر، افزایش مطلوبیت مسئله برای تابع برازش است. اگر $F_S < ۱$ باشد، جواب تأیید شده است و در غیر این صورت، نسل‌های جدید تولید می‌شود تا به جواب تأیید شده برسد. این تابع برازش برای هر تعداد دانشجو و تمام فصول در نظر گرفته شده است. تابع برازش در قالب رابطهٔ ۷ مشاهده می‌شود (سنجی، رنجان و بیچکار، ۲۰۱۲).

$$W = \sum_{i=1}^{40} l_i + \sum_{j=1}^{5} C_j + \sum_{K=1}^{5} C_k + \sum_{i=1}^{200} S_i + \sum_{m=1}^{8} sem_m$$ رابطهٔ ۷)

$$W/(total - classes \times 5)$$ رابطهٔ ۸)

در رابطهٔ بالا سه شاخص وزن وجود دارد که به تعداد استادان (L) وابسته است. C تعداد کلاس‌های درس با آزمایشگاه و بدون آزمایشگاه؛ S تعداد دانشجویان و Sem تعداد فصل‌هاست. کل تابع برازش برای آزمون نتیجهٔ بهینه برای یک هفته طراحی شده است. با مقایسهٔ رابطهٔ ۶ و ۷ مشخص می‌شود که تداخل زمانی در آن دیده نمی‌شود.

به‌دلیل وجود عملیات متفاوت نزدیکی انتخاب استادان و دانشجویان و در نظرگرفتن زمان جلسه و فشرده‌سازی زمان تدریس استادان در این مسئله، الگوریتم در دو مرحله انجام شد. در مرحلهٔ اول الگوریتم، نظر استادان و دانشجویان با استفاده از عملگر Crossover تک‌نقطه‌ای مطابق شکل ۲ (ونگ و همکاران، ۲۰۰۹) به هم نزدیک شدند.



شکل ۲. نمایش Crossover تک‌نقطه‌ای

در مرحلهٔ دوم الگوریتم، از عملگر Crossover معکوس حلقوی (الهام‌گرفته شده از ساختمان داده‌های کامپیوتر)، برای تنظیم برنامهٔ فشردهٔ استادان با لحاظ‌کردن زمان جلسه در روز مشترک و زمان‌بندی برنامهٔ استادان در روز غیرمشترک استفاده شده است. این عملگر نسبت به عملگرهای سنتی سرعت زیادی دارد. همچنین از تولید جواب‌های ناممکن جلوگیری می‌کند.

جمعیت اولیه

جمعیت اولیه، اولین گام در الگوریتم ژنتیک است که اولین جمعیت را برای تولید نسل‌های بعدی الگوریتم ایجاد می‌کند. جمعیت اولیه در مرحلهٔ اول الگوریتم، درس‌های انتخاب‌شدهٔ استادان و درس‌های انتخابی توسط آموزش است. جمعیت اولیه در مرحلهٔ دوم الگوریتم، در دو وجه در نظر گرفته می‌شود. در وجه اول، جمعیت اولیه در روز مشترک جلسهٔ استادان شامل پنج دورهٔ زمانی ۸ تا ۱۰، ۱۰ تا ۱۲، ۱۳ تا ۱۵، ۱۵ تا ۱۷ و ۱۷ تا ۱۹ است. در وجه دوم، جمعیت اولیه در روز غیرمشترک جلسهٔ استادان شامل چهار دورهٔ زمانی می‌شود؛ زیرا یکی از دوره‌های زمانی که برای ساعت جلسه مشخص شده است، از فهرست دوره‌هایی که در یک روز هفته می‌توان در برنامهٔ آموزشی لحاظ کرد، حذف شده است.

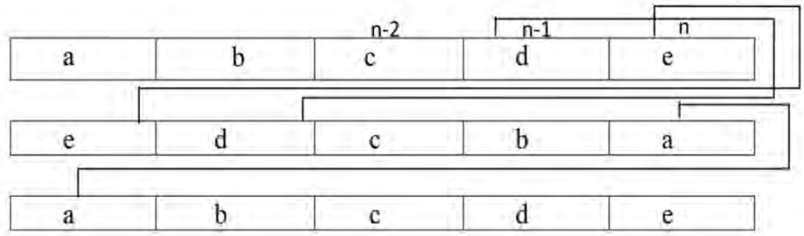
انتخاب محل تکثیر

انتخاب محل تکثیر در این الگوریتم در دو مرحله کاربرد دارد. در مرحلهٔ اول الگوریتم، مقدار g به‌طور تصادفی انتخاب می‌شود تا احتمال چرخ رولت باشد.

رابطهٔ ۹) $$p = \frac{g}{n}$$

احتمال ترکیب (p_c) را ۰/۵ در نظر می‌گیریم. اگر $p < p_c$ بود، سمت چپ رشتهٔ اصلی قبل از برش g با سمت راست رشتهٔ غیراصلی بعد از برش g، به هم چسبانده می‌شود و تولید یک رشته می‌کند. اگر $p > p_c$ بود، سمت راست رشتهٔ اصلی قبل از برش g با سمت چپ رشتهٔ غیراصلی بعد از برش g، به هم چسبانده می‌شود و تولید یک رشته می‌کند. در این حالت عملگر Crossover تک‌نقطه‌ای اتفاق افتاده است. در مرحلهٔ دوم الگوریتم اگر جمعیت اولیه دارای n عضو باشد، اندیس سلول کروموزم اصلی در مرحلهٔ رفت الگوریتم بعد از n بار تکرار $[n-1, n)$ و $[n-2, n-1]$ و $[0, 1)$ می است که در هر بار تکرار به انتهای کروموزوم فرعی اضافه می‌شود. در مرحلهٔ برگشت الگوریتم نیز اندیس سلول کروموزم فرعی بعد از n بار تکرار $[0, 1)$ و $[1, 2]$ و $[2, 3]$ و ... و $[n-1, n]$ است که در هر بار تکرار به انتهای کروموزوم اصلی

اضافه می‌شود. در نهایت، کروموزوم مطابق تابع شایستگی مطرح شده و شکل ۳، بازهٔ زمانی مناسبی تولید می‌کند.



شکل ۳. نمایش جابه‌جایی الگوریتم فهرست معکوس حلقوی

یافته‌های پژوهش

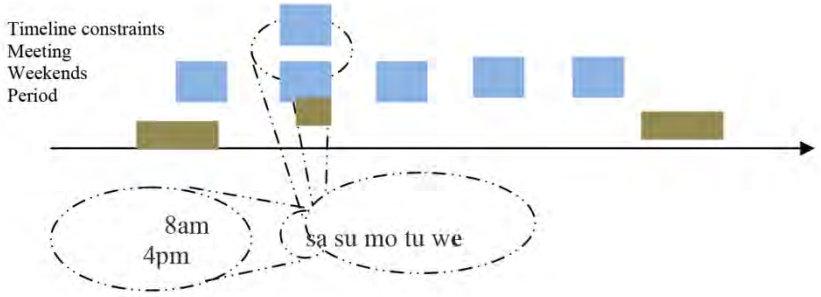
مرحلهٔ اول الگوریتم

این عملگر در مرحلهٔ اول الگوریتم به‌کار می‌رود، به‌گونه‌ای که همزمان با برش کروموزوم استادان، کروموزوم آموزش نیز برش خورده و با یکدیگر ترکیب می‌شوند و نسل جدید ایجاد می‌کنند. تولید نسل جدید تا جایی که شرایط تابع شایستگی مرحلهٔ اول الگوریتم (مطابق شکل ۴) احراز شود، ادامه دارد.

```
FOR each individual of population
 Select a number between 1 and 8 as Lij , cij which reperesnts the
 ith generation of population and the jth chromosome
in this population.
create a class for courses of students view
create a class for courses of professor view
flag the constraints
select two gens random from two chromosome(c,l)
exchange the pieces from Selection Professor chromosome and Student chromosome
Selection as exchange left student chromosome Lwith right professor chromosome c.
exchange the pieces right student chromosome Lwith left professor chromosome c.
check offspring using rules mentioned in section 3.3.3
next
next
```

شکل ٤. pseudo code مرحلهٔ اول الگوریتم crossover تک‌نقطه‌ای

بعداز اجرای مرحلهٔ اول، برنامه وارد اجرای مرحلهٔ دوم الگوریتم می‌شود. برای تبیین زمان‌های آزاد و زمان جلسات شکل ۵ نمایش داده شده است.



شکل ۵. نمایی از محدودیت خط زمان و زمان‌های آزاد

مرحلهٔ دوم الگوریتم

در مرحلهٔ رفت الگوریتم فهرست معکوس حلقوی، مطابق شکل ۶ آخرین سلول رشتهٔ اصلی در اولین سلول رشتهٔ فرعی ریخته شده، رشتهٔ اصلی در متغیری ذخیره می‌شود، رشتهٔ اصلی یک سلول به عقب برگردانده شده و این حالت در رشتهٔ دیگری نگهداری می‌شود تا زمانی که رشتهٔ اصلی خالی شود. در مرحلهٔ برگشت الگوریتم مطابق شکل ۶، سلول‌های کروموزوم فرعی را یک سلول به جلو برده و سلول‌های کروموزوم اصلی را یک سلول به عقب برمی‌گردانیم، حالت‌های به‌دست آمده ذخیره می‌شود. رشتهٔ اصلی یک سلول به جلو برده و به حالت‌ها اضافه می‌شود.

شکل ٦. نمودار وضعیت فهرست معکوس حلقوی

با این روش برنامه‌ای که می‌تواند بین تمام برنامه‌ها بهترین باشد، تنظیم می‌شود؛ به‌گونه‌ای که مجموع فاصلهٔ زمانی بین دروس حداکثر ۲ است. از مقایسه این روش با الگـوریتم ژنتیـک بـا عملگر دو نقطه‌ای مشاهده می‌شود که روش پیشنهاد شـده نسـبت بـه روش اسـتاندارد تضـمین بیشتری برای فشردگی ایجاد می‌کند.

```
Algorithm 2: constraints design
input file of all parameters
if time-slot of professors is common
Save professors common time as meeting
Decrease time-slot of meeting chromosome of professors program
 for i=1 to n
{If main chromosome <> nul
{Shift main chromosome one to right
Save chromosome in string
Save cell n-i in secondary chromosome
Save chromosome in string
} for i=n downto 1
If (main chromosom= null){
Shift n_th cell secondary chromosome to the n-i_th cell of main chromosom
Save cell n-i
Save chromosome in string
}
If meeting<> time slot 1
Assign time- slot [pervious meeting] to course[i]
course[i]=hard
Print golden time
Assign time- slot[ next meeting] to course[i]   course[i]= medium
Print silver time
Assign time-slot[next next meeting] to course[i] course[i]= soft
Print bronz time
```

شکل ۷. pseudo code مرحلهٔ دوم الگوریتم RIL crossover

در مرحلهٔ دوم الگوریتم، از عملگر هوشمند جدیدی به نام فهرست معکوس حلقوی با در نظر گرفتن زمان‌های طلایی، نقره‌ای و برنـزی بـرای دروس مختلـف اسـتفاده شـده اسـت. در ایـن الگوریتم جوابی بهینه است که علاوه بر در نظر گرفتن زمان جلسهٔ مشترک، درس‌های سخت از نظر اکثر استادان و دانشجویان را زودترین زمان قبل از جلسه (زمان طلایـی)، زمـان بعـد را بـه درس متوسط (زمان نقره‌ای) و زمان قبل از جلسه (زمان برنزی) را به درس آسان اختصاص دهد. با در نظر گرفتن درجهٔ اهمیت درس‌ها، در جدول زیر درجهٔ اهمیت ۱ زمان طلایی، درجهٔ اهمیت

۲ زمان نقره‌ای و درجهٔ اهمیت ۳ زمان برنزی است. در زمان‌بندی برنامه، ابتدا درس‌های زمان طلایی قرار می‌گیرد و پس از آن به‌ترتیب زمان نقره‌ای و زمان برنزی جای داده می‌شود.

جدول ۱. نمونهٔ خروجی برنامه نرم افزاری الگوریتم RIL

ساعت / روز	۸-۱۰	۱۰-۱۲	۱۳-۱۵	۱۵-۱۷	۱۷-۱۹
شنبه	نام درس: معماری کامپیوتر نام استاد: استاد ۱ درجهٔ اهمیت: ۱	نام درس: شبکهٔ کامپیوتری پیشرفته نام استاد: استاد ۲ درجه اهمیت: ۱	نام درس: هوش مصنوعی نام استاد: استاد ۳ درجهٔ اهمیت: ۲	نام درس: نرم‌افزار نام استاد: استاد ۴ درجهٔ اهمیت: ۳	نام درس: مباحث ویژه در الگوریتم‌ها نام استاد: استاد ۵ درجهٔ اهمیت: ۳

آزمون و ارزیابی

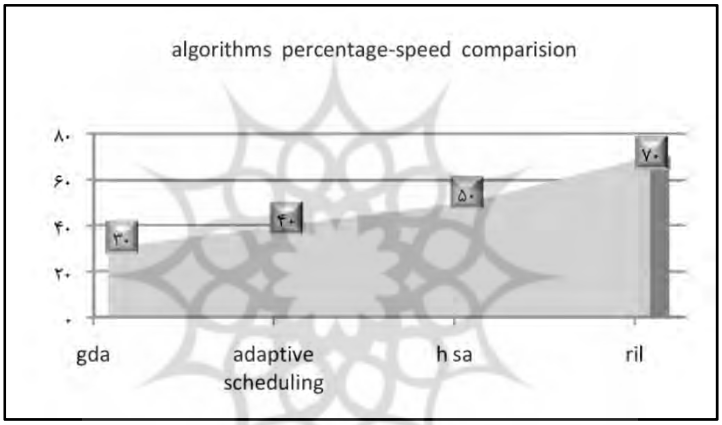
الگوریتم فهرست معکوس حلقوی برای دانشکده‌های دانشگاه امام حسین تهران به‌منظور زمان‌بندی درس استادان و دانشجویان مقطع کارشناسی ارشد با داده‌های واقعی ۲۰۰ دانشجو، ۸ استاد در ۱۰ رشته و ۴۰ گرایش به زبان C# به اجرا درآمد. معیار ارزیابی الگوریتم، تعداد نسل‌ها و سرعت الگوریتم است. محاسبات مرحلهٔ اول الگوریتم Crossover تک‌نقطه‌ای در ۴۹ نسل به جواب رسید. در مرحلهٔ دوم الگوریتم، از عملگر فهرست معکوس حلقوی استفاده شد که نتایج آن نشان می‌دهد عملگر فهرست معکوس حلقوی در مقایسه با روش استاندارد عملگر Crossover دو نقطه‌ای و HSA تعداد نسل‌های کمتری داشته و سریع‌تر به جواب می‌رسد.

جدول ۲. مقایسهٔ تکرار الگوریتم‌های ژنتیک دو نقطه‌ای،HSA و Ril

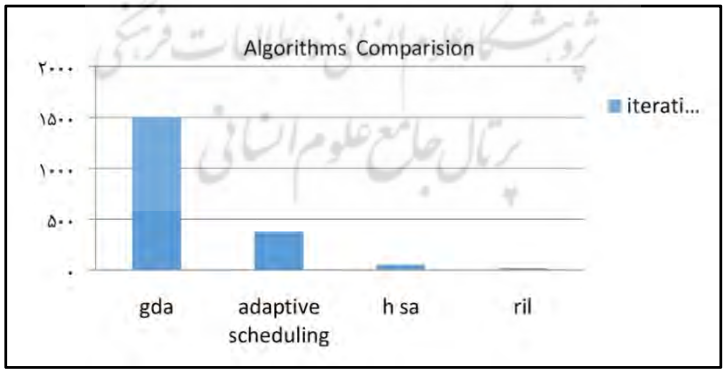
نمونه‌های مسئله	الگوریتم ژنتیک دونقطه‌ای	HSA	RIL
نمونهٔ ۱	۱۵	۱۰	۲۵
نمونهٔ ۲	۱۶	۱۰	۲۵
نمونهٔ ۳	۲۵۷	۵۰	۲۵
نمونهٔ ۴	۸۵۲	۵۰	۲۵
نمونهٔ ۵	۱۳۸۶	۵۰	۲۵
نمونهٔ ۶	۲۷۲۹	۵۰	۲۵
نمونهٔ ۷	۲۹۵۹	۵۰	۲۵
نمونهٔ ۸	۳۷۰۵	۵۰	۲۵
نمونهٔ ۹	۴۹۴۸	۵۰	۲۵
نمونهٔ ۱۰	۷۰۷۴	۵۰	۲۵
نمونهٔ ۱۱	۸۰۲۴	۵۰	۲۵

با توجه به جدول ۲، فهرست معکوس حلقوی و پس از آن HSA تعداد نسل کمتری داشتند، اما الگوریتم ژنتیک دونقطه‌ای تعداد نسل‌های خیلی بیشتری داشت.در شکل ۸ روش‌های الگوریتم ژنتیک تطبیقی (AS)[1]، سیل بزرگ (GDA)، فهرست معکوس حلقوی (RIL) و الگوریتم جست‌وجوی متوازن (HSA) از نظر سرعت الگوریتم با هم مقایسه شده‌اند. در شکل ۹ نیز روش‌های یاد شده از لحاظ تعداد نسل‌ها با یکدیگر مقایسه شده‌اند. همان‌طور که مشخص است، فهرست معکوس حلقوی با تعداد تکرار کمتر به جواب می‌رسد.

با توجه به شکل ۸ مشاهده می‌شود که سرعت الگوریتم RIL بیشتر از HSA است و الگوریتم ژنتیک دو نقطه‌ای کمترین سرعت را دارد.



شکل ۸. مقایسهٔ سرعت الگوریتم



شکل ۹. مقایسهٔ تکرار الگوریتم

1. Adaptive Scheduling

در شکل ۹ نیز مشخص می‌شود که لیست معکوس حلقوی دارای کمترین تکرار است و جست‌وجوی متوازن، الگوریتم ژنتیک تطبیقی و سیل بزرگ به‌ترتیب در رتبه‌های دوم تا چهارم قرار دارند.

نتیجه‌گیری و پیشنهادها

در این تحقیق با کم‌کردن فاصلهٔ انتخاب استادان و دانشجویان و فشرده‌کردن زمان‌بندی، برنامهٔ درسی استادان و دانشجویان، اولویت‌بندی زمان‌ها، در نظر گرفتن جلسهٔ مشترک استادان و برنامهٔ جدید آموزشی برای دانشگاه‌های کشورمان تدوین می‌شود.

الگوریتم ارائه شده نسبت به سایر الگوریتم‌ها از نظر شاخص‌های زمان، سرعت و تعداد نسل و نیز، عدم وجود جواب امکان‌ناپذیر بهتر است. در این الگوریتم جواب‌های بهینهٔ متعددی تولید می‌شود که با احتساب وزن و اولویت به جواب‌ها، جواب نهایی انتخاب خواهد شد. با این روش، سرعت پردازش بیشتر شده، دقت حل مسئله بهبود یافته و از پیمایش فضای حالت ناممکن جلوگیری به عمل آمده است. در مجموع مشخص شد که این روش با برازش بهتری به جواب‌های بهینه می‌رسد.

از مزیت‌های دیگر روش فوق این است که زمان‌های طلایی، نقره‌ای و برنزی نیز برای پوشش درس‌های سخت، متوسط و آسان در روز جلسهٔ استادان در نظر گرفته شده است. قوت دیگر الگوریتم مد نظر در این است که از جواب‌های امکان‌ناپذیر جلوگیری می‌کند و کارایی آن نسبت به روش سنتی بیشتر است.

پیشنهادهایی برای تحقیقات آتی

- با توجه به کارایی الگوریتم فهرست معکوس حلقوی، علاوه بر استفاده در زمان‌بندی درسی، از آن برای زمان‌بندی حرکت قطارها، هواپیماها استفاده شود.
- همچنین در توسعهٔ تحقیق حاضر، می‌توان علاوه بر عنوان‌های درسی، انتخاب روزهای تدریس را نیز برعهدهٔ دانشجویان گذاشت و برنامه را بر اساس رشتهٔ تحصیلی آنها مرتب کرد.
- با این روش می‌توان برای زمان‌بندی امتحانات نیز تحقیق کرد.

فهرست منابع

محمد، ب.، دهقانی، ت.، ذاکرتولایی، م. (۱۳۸۵). رویکردی نوین در زمان‌بندی دروس دانشگاه با استفاده از الگوریتم ژنتیک، *دوازدهمین کنفرانس سالانهٔ انجمن کامپیوتر ایران*، تهران، دانشگاه شهید بهشتی.

Abdennadher, S. & Aly, M. (2007). Constraint-Based University Timetabling for the German University in Cairo, *21st Workshop on (Constraint) Logic Programming*, Würzburg, Germany.

Abdullah, S., Burke, E.K. & Collum, B. (2005). An investigation of variable neighbourhood search for university course timetabling. In: *Proceedings of MISTA 2005. The 2nd Multidisciplinary Conference on Scheduling: Theory and Applications*, 18-21 July, pp. 413-427.

Abdullah, S. & Ahmadi, S. & Dror, M. & Burke, E.K. & McCollum, B. (2003). *A Tabu based Large Neighbourhood Search Methodology for the Capacitated Examination Timetabling Problem.* DOI: 10.1057/palgrave.jors.2602258.

Behdad, M. & Dehghan, T. & Zaker Tavalayee, M. (2007). A New Approach in University Timetabling by Using Genetic Algoritm. *12th Annual Conference of Iranian Computer Society*, Tehran, Shahid Beheshti University.
(*in Persian*)

Sutar, S.R. & Bichkar, R.S. (2012). University Timetabling based on Hard Constraints using Genetic Algorithm, *International Journal of Computer Applications*, 42(15),1-7.

Burke, E.K., Newall, J.P. & Weare, R.F. (1996). A Memetic Algorithm for University Exam Timetabling, PATAT I, Edinburgh, UK, Lecture Notes in Computer Science 1153. *Springer-Verlag. (Editors: E.K. Burke and P. Ross)*, pp. 3-21.

Dorigo, M.I. & Caro, G.D. (1999). *Ant Colony Optimization: A new Meta-Heuristic*. Available in: http://staff.washington.edu/paymana/swarm/dorigo99-cec.pdf.

Doshi, H. & Mittal, D. & Negpure, R. & Sunasra, M. (2015). Automatic timetable generation using genetic algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(2), 245-248.

Dueck, G. (1993). New Optimization Heuristics: The Great Deluge Algorithm and the Record-to-Record Travel. *Journal of Computational Physics,* 104(1), 86-92.

Eley, M. (2006). Ant algorithms for the exam timetabling problem Michael. *International Conference on the Practice and Theory of Automated Timetabling*, PP. 167-180.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations research*, 13(5), 533-549.

Hmer, A. & Mouhoub, M. (2014). A Multi-Phase Hybrid Metaheuristics Approach for the Exam Timetabling, *10th International Conference of the Practice and Theory of Automated Timetabling*, PP. 26-29.

Hsu, C. & Chao, H. (2009). A Student-Oriented Class-Course Timetabling Model with the Capabilities of Making Good Use of Student Time, Saving College Budgets and Sharing Departmental Resources Effectively. *Proceedings of the 2009 WRI Global Congress on Intelligent Systems*, PP. 379-384.

Ikuo, O. & Kunihito, Y. & Yoshiyuki, S. (2004). Timetabling For Satisfying Professors' Requirements and Student's Desires using Genetic Algoritm, *Memoirs of the Faculty of Engineering, Miyazaki University*, PP. 313-318.

Juang, Y. & Kao, H. & Lin, S. (2007). An adaptive scheduling system with genetic algorithms for arranging employee training programs, *Elsevier Expert Systems with Applications*, 33(3), 642-651.

Kahar, M. & Kendall, G. (2015). A great deluge algorithm for a real-world examination timetabling problem. *Journal of The Operational Research Society*, 66(1), 116-133.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983). Optimization by Simulated Annealing Science. *Journal of Computational Physics*, 220(4598), 671-680.

Liu, J. & Yu, X. & Wang, Z. (2009). Self-Fertilization Based Genetic Algoritm for University TimeTabling Problem. *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, PP. 1001-1004.

Marc, R. & Lewis, R. (2006). *Metaheuristics for University Course Timetabling*. A thesis submitted in partial fulfilment of the requirements of Napier University for the degree of Doctor of Philosophy.

Müller, T. & Rudová, H. & Murray, K. (2010). Complex university course timetabling. *Journal of Scheduling*, 14(2), 187-207.

Pilla, N. (2014). A study of the practical and tutorial scheduling problem. *10 th International Conference of the Practice and Theory of Automated Timetabling, PTAT*, PP. 26-29.

Rayward Smith, V.J., Osman, I.H., Reeves, G.D., Smith, G.D. (1996). *Modern Huristics Search Methods*. John Wiley& Sons Ltd.

Wahid, J. & Mohd Hussin, N. (2014). Harmony search algorithm for curriculum- based course timetabling problem. *International Journal of Soft Computing and Software Engineering*, 3(3), 365-371.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی