

بررسی مسأله زمان بندی درسی دانشگاهی با استفاده از ترکیب الگوریتم ممتیک بهبودیافته و الگوریتم سردشدن شبیه سازی شده

مجید جودکی، محمدعلی منتظری و سیدرسول موسوی

کلاس، مکان تشکیل درس و منظور از دوره زمانی، زمان تشکیل آن می باشد. در این مسأله هر درس را به عنوان یک رویداد تلقی می کنند. اگر رویداد نشان دهنده یک امتحان باشد آن مسأله را زمان بندی امتحانی^{۱۱} می گویند. این دو نوع مسأله طبیعتاً با یکدیگر متفاوت هستند [۱]. زمان بندی درسی هر هفته تکرار می شود، در صورتی که زمان بندی امتحانی فقط یک بار در طول ترم تکرار خواهد شد. در زمان بندی درسی دو درس نمی توانند به صورت هم زمان در یک کلاس تشکیل شوند، در صورتی که در زمان بندی امتحانی این امر امکان پذیر است. به عنوان مثال در یک سالن می توان چند امتحان را به صورت هم زمان برگزار نمود. در زمان بندی درسی یک دانشجو تمایل به داشتن چندین درس در یک روز دارد ولی در زمان بندی امتحانی علاقه ای به داشتن چند امتحان در یک روز وجود ندارد [۲].

در یک UCTP برگزاری هم زمان دو درس با دانشجویان و یا اساتید مشترک در یک دوره زمانی یکسان را تداخل^{۱۲} می گویند. هدف از بررسی یک UCTP در حالت کلی کاهش تعداد تداخل بین دروس با دانشجویان و یا مدرسان مشترک و همچنین رفع هم زمانی دروسی است که به یک اتاق مشترک نیاز دارند [۲]. برای مؤسسه های بزرگ مانند دانشگاه ها با افزایش تعداد محدودیت ها^{۱۳} رسیدن به یک جواب قابل قبول^{۱۴} بسیار مشکل تر خواهد بود.

حل یک UCTP کار بسیار مشکلی است و علت این امر بزرگ بودن اندازه و متفاوت بودن ساختار و ماهیت انواع این مسأله می باشد. برای بررسی و حل UCTP روش های متفاوتی ارائه شده است. روش های مبتنی بر رنگ آمیزی گراف^{۱۵}، انواع مکاشفه ها، برنامه ریزی خطی^{۱۶}، شبکه های عصبی مصنوعی^{۱۷}، روش های فرامکاشفه ای، تکنیک های ارضای محدودیت^{۱۸} و روش های مبتنی بر دانش^{۱۹} از این جمله اند.

در [۳] یک روش برنامه ریزی صحیح^{۲۰} برای حل UCTP ارائه شده است. در [۴] و [۵] از یک مدل برنامه ریزی صحیح ترکیبی استفاده شده است و یک روش برنامه ریزی خطی صحیح^{۲۱} را نیز می توان در [۶] مشاهده کرد.

تبدیل مسأله زمان بندی به مسأله شناخته شده رنگ آمیزی گراف

چکیده: مسأله زمان بندی به عنوان یکی از مسایل پیچیده بهینه سازی شناخته می شود. یک نمونه از مسایل زمان بندی، برنامه ریزی درسی دانشگاهی است که هر نیم سال در دانشکده های یک دانشگاه انجام می شود. تنوع محدودیت ها در این مسأله باعث می شود برنامه ریزی در دانشکده های مختلف به گونه های متفاوتی انجام شود. کارهای زیادی برای حل این مسأله انجام شده است که اکثر آنها از روش های فرامکاشفه ای^۱ بهره برده اند. در این مقاله یک مسأله زمان بندی واقعی مورد بررسی قرار می گیرد و یک روش مبتنی بر الگوریتم ممتیک^۲ بهبودیافته که از الگوریتم سردشدن شبیه سازی شده^۳ به عنوان رویه جستجوی محلی^۴ خود استفاده می کند، ارائه می شود. منظور از بهبود در الگوریتم ممتیک، استفاده از روش های مکاشفه ای^۵ در تولید جمعیت اولیه و همچنین تغییر عملگر تقاطع^۶ در این الگوریتم می باشد. همچنین یک عملگر به نام عملگر بهبود^۷ جهت بهبود راه حل های تولید شده و کاهش تعداد نقض محدودیت ها طراحی شده است. به کارگیری روش سردشدن شبیه سازی شده به عنوان رویه جستجوی محلی در الگوریتم ممتیک باعث افزایش توانایی بهره برداری^۸ این الگوریتم خواهد شد. کارآمدی این روش در مقایسه با برخی روش های جدید، با توجه به نتایج به دست آمده بر روی داده های استاندارد نشان داده شده است. همچنین مقایسه نتایج حاصل از این روش با روش انجام شده به صورت دستی بر روی داده های واقعی نشان دهنده برتری این روش می باشد.

کلید واژه: الگوریتم سردشدن شبیه سازی شده، الگوریتم ممتیک، جستجوی محلی، مسأله زمان بندی درسی دانشگاهی.

۱- مقدمه

تخصیص دروس به کلاس های معین در دوره های زمانی^۹ مشخص را مسأله زمان بندی درسی دانشگاهی^{۱۰} (UCTP) می گویند [۱]. منظور از

این مقاله در تاریخ ۲۱ دی ماه ۱۳۸۹ دریافت و در تاریخ ۱۰ بهمن ماه ۱۳۹۰ بازنگری شد.

مجید جودکی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان (email: m.joudaki@cc.iut.ac.ir)

محمدعلی منتظری، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان (email: montazer@cc.iut.ac.ir)

سیدرسول موسوی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان (email: srm@cc.iut.ac.ir)

11. Examination Timtabling Problem
12. Conflict
13. Constraints
14. Feasible
15. Graph Coloring
16. Linear Programming
17. Artificial Neural Network
18. Constraint Satisfaction
19. Knowledge Base Technic
20. Integer Programming
21. Integer Linear Programming

1. Metaheuristic
2. Memetic Algorithm
3. Simulated Annealing
4. Local Search
5. Heuristic
6. Crossover
7. Improvement
8. Exploiting
9. Time Slot
10. University Course Timetabling Problem

از نمونه‌های مفروض به الگوریتم داده شود، نیاز به تنظیم مجدد پارامترها خواهد بود و به همین دلیل از مقایسه روش پیشنهادی با این الگوریتم صرف نظر کرده‌ایم.

برخی از الگوریتم‌های بهینه‌سازی که از روش سردشدن شبیه‌سازی شده بهره می‌برند، در [۳۳] تا [۲۶] پیشنهاد شده است. مثال‌های قابل ذکر دیگری از تکنیک‌های چندهدفه^۹ که در جداول زمان‌بندی امتحانی به کار رفته‌اند را می‌توان در [۲۷] و [۲۸] و به صورت مبسوط‌تر در [۲۹] دید. در کارهایی نظیر [۳۰] و [۳۱]، دو الگوریتم بر پایه الگوریتم اجتماع مورچگان^{۱۰} برای حل UCTP ارائه شده است. مراجع [۳۲] تا [۳۷] نمونه کارهایی هستند که در آنها الگوریتم ممتیک بر UCTP اعمال شده است. در [۲] تعداد زیادی از روش‌های فرامکاشفه‌ای برای حل UCTP مورد بررسی قرار گرفته است.

در کنار الگوریتم‌های فرامکاشفه‌ای، الگوریتم‌های ابرمکاشفه‌ای^{۱۱} نیز در زمینه مسأله زمان‌بندی درسی دانشگاهی مورد بررسی قرار گرفته‌اند [۳۸] تا [۴۰]. الگوریتم‌های ابرمکاشفه‌ای، الگوریتم‌هایی هستند که در فضای الگوریتم‌های مکاشفه‌ای (الگوریتم‌های سطح پایین) جستجو کرده و بهترین ترکیب آنها را برای حل مسأله مورد نظر به دست می‌آورند. از جمله مزیت‌های این الگوریتم‌ها اعمال آنها بر گستره بزرگی از انواع مسائل بهینه‌سازی و کاهش احتمال گیرافتادن الگوریتم در نقاط بهینه محلی می‌باشد.

ترتیب مطالب در ادامه به این صورت می‌باشد: در بخش بعدی الگوریتم ممتیک توضیح داده می‌شود. در بخش ۳ مسأله زمان‌بندی درسی دانشگاهی معرفی می‌گردد. روش پیشنهادی برای بررسی مسأله زمان‌بندی درسی دانشگاهی تحت عنوان MA_SA_ECE_UCTP^{۱۲} در بخش ۴ بررسی می‌شود. در بخش ۵ نتایج حاصل از اعمال الگوریتم پیشنهادی بر روی دسته‌ای از داده‌های استاندارد و مقایسه با برخی از روش‌ها و همچنین اعمال این روش بر روی داده‌های دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان نشان داده شده است. نتیجه‌گیری از کار انجام‌شده و ارائه پیشنهاد برای کارهای آینده در بخش ۶ مورد بررسی قرار می‌گیرد.

۲- الگوریتم ممتیک

تکامل ژنتیکی تنها شکل تکامل که توسط ریچارد داوکینز^{۱۳} [۳۰] اشاره شده است، نمی‌باشد. تکامل فرهنگی شکل دیگری از تکامل است که نسبت به تکامل ژنتیکی دارای حوزه عمل بزرگ‌تری می‌باشد. در مقایسه با انتقال خصوصیت‌ها در تکامل ژنتیکی، جریان اطلاعات در یک فرآیند تکاملی را انتقال فرهنگی می‌گویند. ژن‌ها به‌عنوان اجزای کروموزوم در انتقال صلاحیت‌های شایسته، به‌عنوان یک واحد حیاتی در انتخاب طبیعی به‌کار گرفته می‌شوند. نقطه مقابل ژن، در تکامل فرهنگی مِم^{۱۴} نامیده می‌شود که مخفف کلمه یونانی Mimeme می‌باشد [۳۰]. یک مِم به‌عنوان واحد انتقال فرهنگی همانند تقلید^{۱۵} تعبیر می‌شود.

می‌تواند در رسیدن به جواب مسأله کاملاً مفید و مؤثر باشد [۷] تا [۹]. مسأله رنگ‌آمیزی گراف عبارت است از تخصیص تعداد محدودی رنگ به رئوس یک گراف، به طوری که رئوس مجاور هم‌رنگ نباشند [۲]. در تبدیل مسأله زمان‌بندی به مسأله رنگ‌آمیزی گراف، ایده کار به این صورت است که رأس‌های گراف نشان‌دهنده دروس و یال‌های بین رئوس نماینده محدودیت‌ها می‌باشند. تعداد رنگ‌ها نشان‌دهنده تعداد برش‌های زمانی در دسترس می‌باشد. اختصاص یک رنگ به یک رأس درست مانند تخصیص یک برش زمانی به یک درس می‌باشد.

در کارهای [۱۰] و [۱۱] از شبکه‌های عصبی برای حل UCTP استفاده شده است. در این کارها ارضای محدودیت‌های مسأله را با تابع انرژی سیستم مدل کرده‌اند و همگراشدن سیستم به حالت پایدار نشان‌دهنده نقاط بهینه محلی می‌باشد. نویسندگان این مقاله‌ها تابع انرژی شبکه عصبی را طوری فرموله کرده‌اند که شامل یک مجموع وزن‌دار از نقض محدودیت‌های سخت^۱ و نرم^۲ مسأله باشد. کمینه‌کردن این مجموع به معنی رسیدن شبکه عصبی به یک حالت قابل قبول که علاوه بر ارضای محدودیت‌های سخت، تعداد زیادی از محدودیت‌های نرم نیز در آن ارضا شده است.

بسیاری از کارهای انجام‌شده در ارتباط با UCTP در سال‌های اخیر از روش‌های فرامکاشفه‌ای و الگوریتم‌های جستجوی محلی استفاده کرده‌اند. در [۱۲] ترکیب روش‌های تپه‌نوردی^۳ و جستجوی ممنوعه^۴ به کار برده شده است. استفاده از روش‌های مکاشفه‌ای برای تولید نقاط شروع و سپس اعمال الگوریتم جستجوی ممنوعه، کار دیگری است که در [۱۳] مشاهده می‌شود. نویسندگان در [۱۴] از الگوریتم سیل بزرگ^۵ در حل مسأله استفاده کرده‌اند. آنها بر این باورند که مزیت کار آنها تنظیم کردن تنها دو پارامتر در الگوریتم می‌باشد. اولین پارامتر زمان اجرا و دومی تخمین میزان کیفیت جواب می‌باشد. در [۱۵] تأثیر دو نوع همسایگی بر الگوریتم جستجوی ممنوعه در حل UCTP مورد بررسی قرار گرفته است. کارهای مشابه دیگری را می‌توان در [۱۶] و [۱۷] مشاهده نمود.

در مورد UCTP، الگوریتم ژنتیک^۶ به شکل‌های متفاوت مورد استفاده قرار گرفته است. به‌عنوان مثال کار انجام‌شده در [۱۸] ترکیب الگوریتم ژنتیک با یک روش جستجوی محلی و اعمال آن بر UCTP است. روش‌های دیگری برای زمان‌بندی با استفاده از الگوریتم‌های تکاملی^۷ در [۱۹] و [۲۰] گزارش شده است.

کار انجام‌شده در [۲۱] و [۲۲] از الگوریتم سیل بزرگ ولی با نرخ کاهش غیر خطی^۸ استفاده کرده است. در این کار، الگوریتم جواب‌های بسیار خوبی در مقایسه با سایر روش‌ها ارائه نموده است، ولی عیب این کار این است که پارامترهای زیادی در الگوریتم باید تنظیم شوند و هر کدام از این پارامترها نسبت به نمونه مسأله تحت بررسی دارای مقادیر خاصی هستند. به همین دلیل علی‌رغم جواب‌های خوب این روش، الگوریتم ارائه‌شده یک الگوریتم کلی نبوده و صرفاً برای نمونه مسأله‌های خاصی دارای کارایی بالایی می‌باشد. چنانچه یک نمونه مسأله جدید خارج

9. Multi-Object Technic
10. Ant Colony Algorithm
11. Hyper Heuristic Algorithms
12. Memetic Algorithm Simulated Annealing Electrical and Computer Engineering_UCTP
13. Richard Dawkins
14. Memes
15. Imitation

1. Hard Constraint
2. Soft Constraint
3. Hill Climbing
4. Tabu Search
5. Great Deluge
6. Genetic Algorithm
7. Evolutionary Algorithms
8. Nonlinear Decreasing Rate

محدودیت‌های سخت دارای اولویت بالاتری نسبت به محدودیت‌های نرم هستند و یک جدول زمان‌بندی، زمانی قابل قبول است که همه محدودیت‌های سخت آن ارضا شده باشند. به‌عنوان مثال یک دانشجوی نباید دو درس در یک برش زمانی یکسان داشته باشد. از طرف دیگر، محدودیت‌های نرم، محدودیت‌هایی هستند که با توجه به سیاست‌های دانشگاه و نیاز افراد، رعایت آنها مطلوب بوده اما ارضای آنها اجباری نمی‌باشد. به‌عنوان مثال این که یک دانشجوی در سه برش زمانی متوالی در یک روز سه درس نداشته باشد [۲].

نکته دیگری که در مورد UCTP در همه شکل‌های مختلف آن وجود دارد این است که همه آنها از نظر پیچیدگی محاسباتی^۱ به کلاس مسایل NP-hard^۲ تعلق دارند و در نتیجه الگوریتمی با پیچیدگی زمانی از مرتبه چندجمله‌ای^۳ برای حل آنها ارائه نشده است [۲]. حل مسایل زمان‌بندی واقعی با استفاده از الگوریتم‌های دقیق^۴ غیر عملی می‌باشد چرا که زمان اجرای این دسته از الگوریتم‌ها با رشد اندازه مسأله به‌صورت نمایی افزایش می‌یابد. بنابراین می‌توان از الگوریتم‌های تقریبی^۵ مانند الگوریتم‌های مکاشفه‌ای و فرامکاشفه‌ای برای حل آنها استفاده نمود.

در این مقاله دو نمونه مسأله زمان‌بندی درسی دانشگاهی را مورد بررسی قرار خواهیم داد. اولین مورد مسأله UCTP مطرح‌شده در ITC^۶ توسط Ben Paecher و دومین مورد مسأله زمان‌بندی درسی در دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان می‌باشد.

در مورد اول یک UCTP را می‌توان در حالت کلی به‌صورت زیر بیان کرد [۱۹]: یک مجموعه از دروس (E) که باید در ۴۵ برش زمانی (۵ روز و هر روز شامل ۹ ساعت) زمان‌بندی شوند، یک مجموعه از اتاق‌ها (R) که دروس در آنها برگزار می‌شوند، یک مجموعه از دانشجویان (S) که دروس را اخذ می‌کنند و یک مجموعه از خصوصیات (F) که مورد نیاز دروس بوده و توسط اتاق‌ها ارضا می‌شوند. هر دانشجوی تعدادی از این دروس را اخذ می‌کند و هر اتاق دارای اندازه‌ای ثابت می‌باشد. یک جدول زمان‌بندی را قابل قبول می‌گویند اگر به هر درس یک برش زمانی و یک اتاق اختصاص یابد و محدودیت‌های سخت نیز در جدول ارضا شوند [۲]. این محدودیت‌ها عبارتند از: الف) هیچ دانشجویی بیش از یک درس در یک برش زمانی نداشته باشد. ب) ظرفیت اتاق به اندازه کافی بزرگ باشد (بزرگ‌تر یا مساوی تعداد دانشجویان درس) و تمامی خصوصیات مورد نیاز درس را ارضا کند. ج) در یک برش زمانی فقط یک درس در یک کلاس برگزار شود. محدودیت‌های نرم عبارتند از: الف) یک دانشجوی در آخرین برش زمانی در روز کلاس نداشته باشد. ب) یک دانشجوی بیش از دو درس در دو برش زمانی پشت سر هم نداشته باشد. ج) یک دانشجوی در یک روز تنها یک کلاس نداشته باشد. هدف این مسأله کاهش تعداد نقض‌های محدودیت نرم در یک راه حل عملی (قابل قبول) است. تمامی راه‌حل‌های غیر عملی فاقد ارزش می‌باشند.

در مورد مسأله زمان‌بندی درسی دانشگاهی در دانشکده برق و کامپیوتر، علاوه بر محدودیت‌های نرم مطرح‌شده در بالا، محدودیت‌های زیر نیز وجود دارند:

- دروس مربوط به دانشجویان سال آخر دوره کارشناسی در روز

1. Complexity of a Computation
2. Polynomially Bounded Algorithm
3. Exact Algorithms
4. Approximation Algorithms
5. International Timetabling Competition
6. Violations

مثال‌هایی از م‌ها عبارتند از: آهنگ صدا، ایده‌ها، اصطلاحات قرضی، مد لباس‌ها، درست کردن قهوه یا ساختن ساختمان‌ها. در حین انتقال اطلاعات توسط م‌ها ممکن است تغییراتی نیز اتفاق بیفتد که اغلب این تغییرات گسترده هستند. به‌عنوان مثال، برای ایجاد نظریه‌های علمی ایده‌ها باید تغییر کنند و یا با ایده‌های دیگر ترکیب شوند.

میان تکامل ژنتیکی و فرهنگی تفاوت‌های عمده‌ای وجود دارد. اول این که تکامل فرهنگی خیلی سریع‌تر از تکامل ژنتیکی اتفاق می‌افتد و یک فرآیند قوی با منابع کم می‌باشد. دوم این که در تکامل فرهنگی، به ندرت تغییر، محصول کپی خطاها یا تبادل تصادفی واحدهای هم‌ساز اطلاعاتی است. در حوزه علم، ترکیب خام ایده‌ها به بهبود نظریه‌ها منجر نمی‌شود. یک دانشمند معمولاً یک ایده را بعد از ترکیب با ایده‌های خود و اعتبارسنجی خود می‌پذیرد. فرآیند ترکیب ایده‌ها، به نوآوری منجر می‌گردد. علت این امر گردآوری دانش و م‌های جدید از سایر حوزه‌های دیگر است. تکامل زیستی با مؤلفه نوآوری همراه نیست و اجازه ظهور تجربه در کروموزوم‌های والد را نمی‌دهد. می‌توان گفت که تکامل طبیعی یک فرآیند بی‌پایان است در حالی که تکامل فرهنگی یک فرآیند مبتنی بر هدف می‌باشد که در آن واحدهای انتقال اطلاعات به‌صورت هوشمند عمل می‌کنند.

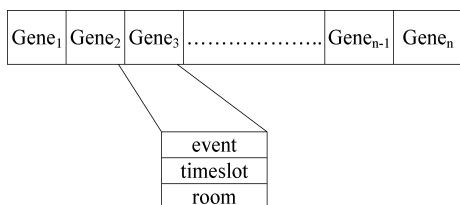
الگوریتم‌هایی که شباهت آنها به تکامل فرهنگی نسبت به تکامل زیستی بیشتر می‌باشد را الگوریتم‌های ممیتیک می‌نامند. علت این نام‌گذاری به راهبرد جستجوی نهفته در این الگوریتم‌ها که به‌طور عمده با الگوریتم‌های تکاملی متفاوت است، بر می‌گردد. برای کارآمدتر شدن جستجو در الگوریتم‌های تکاملی چندین هدف مهم طراحی شده است که توسط الگوریتم ممیتیک برآورده شده‌اند:

- الگوریتم‌های بهینه‌سازی باید کارآمد باشند. به‌عنوان مثال آنها باید قادر به پیدا کردن یک راه حل قابل قبول در یک زمان کوتاه باشند.
- الگوریتم‌های بهینه‌سازی بر اساس تعریف، مبتنی بر هدف هستند. نمایش اعضای جمعیت به‌عنوان راه‌حل‌های کاندید، مشارکت و رقابت آنها برای رسیدن به هدف در حین فرآیند جستجو، مبین این تعریف می‌باشد.
- با توجه به واقعیت درجه توازی بالا، الگوریتم ژنتیک یک رویه جستجوی کورکورانه را با استفاده از منابع عظیم که تنها شامل یک جمعیت کوچک در یک زمان کوتاه می‌باشد، ارائه می‌کند. بنابراین همگرا شدن سریع و عدم بهره‌گیری از خاصیت توزیع اطلاعات از معایب آن می‌باشد. در نتیجه نیاز به مؤلفه‌های دیگری برای تغییر راه‌حل‌ها احساس می‌شود. این مسأله می‌تواند با اعمال درجه‌ای از جستجوی همسایگی مرتفع گردد.
- نمای کلی الگوریتم ممیتیک مانند شبه‌کد نشان داده شده در Error! Reference source not found. می‌باشد.

۳- مسأله زمان‌بندی درسی دانشگاهی

مسأله زمان‌بندی درسی دانشگاهی یک کار بسیار پیچیده است. علت این امر محدودیت‌های آموزشی است که در دانشگاه باید مد نظر قرار گرفته شود. محدودیت‌های مذکور به دو دسته تقسیم‌بندی می‌شوند [۲]:

- محدودیت‌های سخت: آن دسته از محدودیت‌ها هستند که ارضای آنها در جدول زمان‌بندی اجباری است.
- محدودیت‌های نرم: آن دسته از محدودیت‌ها هستند که اگر ارضا شوند، کیفیت جدول زمان‌بندی را افزایش می‌دهند، ولی ارضای آنها اجباری نیست.



شکل ۲: نحوه نمایش کروموزومها.

۴- روش پیشنهادی (MA_SA_ECE_UCTP)

در الگوریتم MA_SA_ECE_UCTP از ترکیب الگوریتم ممتیک (MA) و روش جستجوی سرد شدن شبیه سازی شده (SA) استفاده شده است. نحوه عملکرد MA_SA_ECE_UCTP مطابق شکل ۱ به شرح ذیل می باشد: برای هر کروموزوم، ابتدا با استفاده از رویه مقداردهی اولیه مکاشفه ای، کروموزومها مقداردهی اولیه می شوند. سپس رویه جستجوی محلی مبتنی بر SA جهت بهبود کروموزوم و تبدیل آن به یک راه حل بهینه محلی^۱، فراخوانی می شود. بعد از این مرحله عملگر بهبود بر روی کروموزوم اعمال شده و موجب بهبود هر چه بیشتر آن می گردد. در حلقه while بعد از انتخاب دو کروموزوم، متناسب با میزان برازندگی^۲ آنها از بین جمعیت، و اعمال عملگر تقاطع با احتمال pC و عملگر جهش^۳ با احتمال pM یک فرزند^۴ تولید می شود. سپس رویه SA و عملگر بهبود به ترتیب فراخوانی شده و تعداد نقض محدودیت های سخت و نرم موجود در فرزند را بهبود می بخشند. در نهایت این فرزند با بدترین عضو جمعیت مقایسه شده و در صورت بهتر بودن جایگزین آن می شود. این رویه تا رسیدن به شروط توقف ادامه می یابد.

۴-۱ نحوه نمایش راه حل با استفاده از کروموزوم

هر کروموزوم شامل مجموعه ای از ژن ها است که نشان دهنده یک راه حل کاندید می باشد. نمایش استاندارد برای یک جدول زمان بندی به صورت یک ماتریس دوبعدی است که در آن سطرها نشان دهنده روزهای هفته و ستون ها نمایانگر برش های زمانی در روز می باشد. در این مقاله از یک نمایش ساده ولی مناسب برای الگوریتم ممتیک استفاده شده است. در این نمایش از یک آرایه یک بعدی با طولی برابر با تعداد دروس استفاده می شود. این آرایه را یک کروموزوم می نامیم. هر عنصر آرایه یک ژن و هر ژن از یک سه تایی $\langle \text{event}, \text{timeslot}, \text{room} \rangle$ تشکیل شده است. تعداد ژن ها در هر کروموزوم برابر با تعداد دروس می باشد. علت استفاده از این نوع نمایش ساده بودن و سادگی اعمال عملگرهای الگوریتم ممتیک بر روی آن می باشد. ساختار هر کروموزوم در شکل ۲ نمایش داده شده است.

۴-۲ رویه مقداردهی اولیه مکاشفه ای

از آنجا که UCTP بسیار شبیه به مسأله رنگ آمیزی گراف می باشد [۲]، می توان برای ساخت کروموزومها از رویه های مکاشفه ای استفاده شده در مسأله رنگ آمیزی گراف استفاده نمود [۳۵]. اگر هر درس را به عنوان یک رأس در نظر بگیریم و چون دروسی که دانشجو و یا استاد مشترک دارند، نمی توانند در یک برش زمانی یکسان قرار گیرند، پس بین چنین

1. Local Optima Solution
2. Fitness
3. Mutation
4. Offspring

```

input : a problem instance I
foreach chromosome  $pop_i$  in population do
 $pop_i \leftarrow$  heuristic initial chromosome
 $pop_i \leftarrow$  apply Local Search(SA) on  $pop_i$ 
 $pop_i \leftarrow$  apply improvement routine on  $pop_i$ 
end for
sort population based on fitness
while termination condition not reached do
select two parents from population by proportional selection
 $child \leftarrow$  child solution after crossover with a probability pC
 $child \leftarrow$  child solution after mutation with a probability pM
 $child \leftarrow$  child solution after applying Local Search(SA)
 $child \leftarrow$  child solution after applying improvement
if  $child$  is better than worst member of the population
 $worst\ member \leftarrow child$ 
sort population based on fitness
end while
output : The best solution  $pop_0$  achieved for I

```

شکل ۱: شبه کد الگوریتم MA_SA_ECE_UCTP.

چهارشنبه زمان بندی نشوند.

یک درس در اتاقی زمان بندی شود که ظرفیت آن خیلی بزرگ تر از تعداد دانشجویان آن درس نباشد. به عنوان مثال اگر دو اتاق با ظرفیت ۵۰ و ۸۰ نفر موجود باشند و هر دو برای درس x با تعداد دانشجویان ۴۵ نفر مناسب باشند، آنگاه اتاق با ظرفیت ۵۰ نفر انتخاب شود.

و محدودیت های سخت عبارتند از:

- عدم تشکیل دو درس به صورت هم زمان (در یک برش زمانی) در یک کلاس.
 - عدم وجود تداخل (دروس هم زمان) برای دانشجویان.
 - عدم وجود تداخل (دروس هم زمان) برای اساتید.
 - زمان بندی دروس در کلاس های مناسب (کلاس هایی که دارای امکانات مورد نیاز برای درس هستند).
 - زمان بندی دروس در زمان هایی که از قبل برای آن درس مشخص شده است. به عنوان مثال درس x باید در روز شنبه ساعت ۸ تا ۹ زمان بندی شود.
 - زمان بندی دروس در زمان هایی که اساتید مربوط به آن دروس در دانشکده حضور دارند. به عنوان مثال استاد x فقط روزهای شنبه و دوشنبه تدریس می کنند و یا استاد y روزهای یکشنبه صبح تدریس نمی کند.
 - چنانچه درسی در یک کلاس در زمان خاصی زمان بندی شود، آن کلاس در آن زمان در اختیار دانشکده باشد. به عنوان مثال کلاس x روز دوشنبه بعد از ظهر در اختیار دانشکده دیگری است.
 - عدم زمان بندی دروس در روز پنجشنبه.
- از جمله مزیت های کار انجام شده در این مقاله استفاده از تکنیک های مکاشفه ای و روش های جستجوی فرامکاشفه ای در بررسی UCTP می باشد. تفاوت عمده این کار با کارهای انجام شده، انعطاف پذیری آن برای بررسی داده های واقعی و استفاده از آن در محیط واقعی دانشکده می باشد. بالا بودن دقت برنامه، سرعت، اعمال سیاست های آموزشی دانشکده در برنامه ریزی، همسوی بودن برنامه آموزشی با استفاده بهینه از وقت اساتید و دانشجویان از جمله مزایای این برنامه می باشد.

جدول ۱: رویه‌های مکاشفه‌ای برای ساختن کروموزوم‌ها.

h_1	انتخاب رویداد با تعداد محل مناسب کمتر در جدول زمان بندی نسبت به سایر رویدادها.
h_2	انتخاب رویداد دارای بیشترین تداخل با سایر رویدادها.
h_3	انتخاب رویداد به صورت تصادفی.
h_4	انتخاب محل مناسب برای تعداد کمتری رویداد تخصیص نیافته.
h_5	انتخاب محل با تعداد کمتری رویداد در یک برش زمانی خاص.
h_6	یک محل را به صورت تصادفی انتخاب کن.
h_7	انتخاب رویدادی که دارای تعداد دانشجویان کمتری می‌باشد.

می‌باشد. ساختارهای همسایگی به مجموعه تغییراتی گفته می‌شود که بر روی یک راه حل انجام شده و موجب به وجود آمدن راه حل‌های همسایه می‌شود. این مجموعه حرکت‌ها تا حد امکان ساده بوده و باعث کاوش فضای جستجوی اطراف کروموزوم خواهند شد. دلیل استفاده از ساختارهای همسایگی متفاوت، تنوع در راه حل‌های همسایه به دست آمده برای کاوش بهتر فضای جستجو می‌باشد. به عنوان مثال زمانی که یکی از ساختارهای همسایگی در ارائه راه حل‌های همسایه بهتر نسبت به راه حل جاری، ناتوان باشد، می‌توان از سایر ساختارهای همسایگی بهره برد. ساختارهای همسایگی مورد استفاده در این مقاله عبارتند از:

– ساختار همسایگی N_1 : در این ساختار که ساده‌ترین نوع ساختار همسایگی مورد استفاده در مقاله می‌باشد، یک درس مانند e_x که به برش زمانی t_x تخصیص یافته است، به صورت تصادفی انتخاب شده و به یک برش زمانی تصادفی جدید مانند t_y انتقال داده می‌شود. نتیجه این حرکت رسیدن به یک جدول زمان بندی جدید می‌باشد که در آن برش زمانی تخصیص داده شده به درس e_x با برش زمانی تخصیص داده شده به همان درس در جدول زمان بندی قبلی متفاوت می‌باشد.

– ساختار همسایگی N_2 : در این ساختار دو درس e_x و e_y به صورت تصادفی انتخاب شده و برش‌های زمانی آنها (برش‌های زمانی که به این دو درس تخصیص پیدا کرده‌اند) با یکدیگر تعویض می‌شود.

– ساختار همسایگی N_3 : در این ساختار سه درس e_x ، e_y و e_z به صورت تصادفی انتخاب شده و برش‌های زمانی آنها (برش‌های زمانی که به این سه درس اختصاص یافته‌اند) به صورت چرخه‌ای با یکدیگر تعویض می‌شوند. در شکل‌های ۳ تا ۵ نحوه عمل هر یک از ساختارهای همسایگی نشان داده شده است.

همان‌طور که در شکل ۶ نشان داده شده است، نحوه عملکرد رویه جستجوی محلی به شرح زیر می‌باشد. راه حل جاری، s ، که یک آرایه از دروس تخصیص یافته شده به برش‌های زمانی و اتاق‌ها می‌باشد، به رویه تحویل داده می‌شود. دمای اولیه در الگوریتم سرد شدن شبیه‌سازی شده، T ، نامیده شده و توسط کاربر تنظیم می‌گردد. اولین درس از لیست دروس، انتخاب شده و نقض محدودیت‌های ناشی از انتساب این درس به برش زمانی و اتاق خاصی که در s برای این درس مشخص شده بررسی می‌گردد. در صورت عدم وجود این چنین نقضی، درس بعدی از لیست دروس انتخاب شده و این رویه تکرار می‌شود. چنانچه درس مذکور باعث ایجاد نقض محدودیت شده باشد، باید برش زمانی متناسب به این درس را تغییر دهیم. برای این کار ابتدا از ساختار همسایگی N_1 استفاده می‌کنیم. در صورت کاهش تعداد نقض محدودیت‌ها، تغییر مذکور را پذیرفته و در غیر این صورت از ساختار همسایگی N_2 استفاده می‌شود. چنانچه این

t_x	e_1	e_{10}	e_3	e_6	e_x	e_{20}	e_{34}
t_y	e_2	e_7	e_{15}	e_5	e_{27}	e_4



t_x	e_1	e_{10}	e_3	e_6	e_{20}	e_{34}
t_y	e_2	e_7	e_{15}	e_5	e_x	e_{27}	e_4

شکل ۳: نحوه عملکرد ساختار همسایگی N_1 . رویداد e_x از برش زمانی t_x به برش زمانی t_y انتقال یافته است.

t_x	e_1	e_{10}	e_3	e_6	e_x	e_{20}	e_{34}
t_y	e_2	e_7	e_{15}	e_5	e_y	e_{27}	e_4



t_x	e_1	e_{10}	e_3	e_6	e_y	e_{20}	e_{34}
t_y	e_2	e_7	e_{15}	e_5	e_x	e_{27}	e_4

شکل ۴: نحوه عملکرد ساختار همسایگی N_2 . رویداد e_x از برش زمانی t_x به برش زمانی t_y و رویداد e_y از برش زمانی t_y به برش زمانی t_x انتقال یافته است.

t_x	e_1	e_{10}	e_3	e_6	e_x	e_{20}	e_{34}
t_x	e_2	e_7	e_{15}	e_5	e_y	e_{27}	e_4
t_x	e_{12}	e_{17}	e_{25}	e_{11}	e_z	e_9	e_{14}



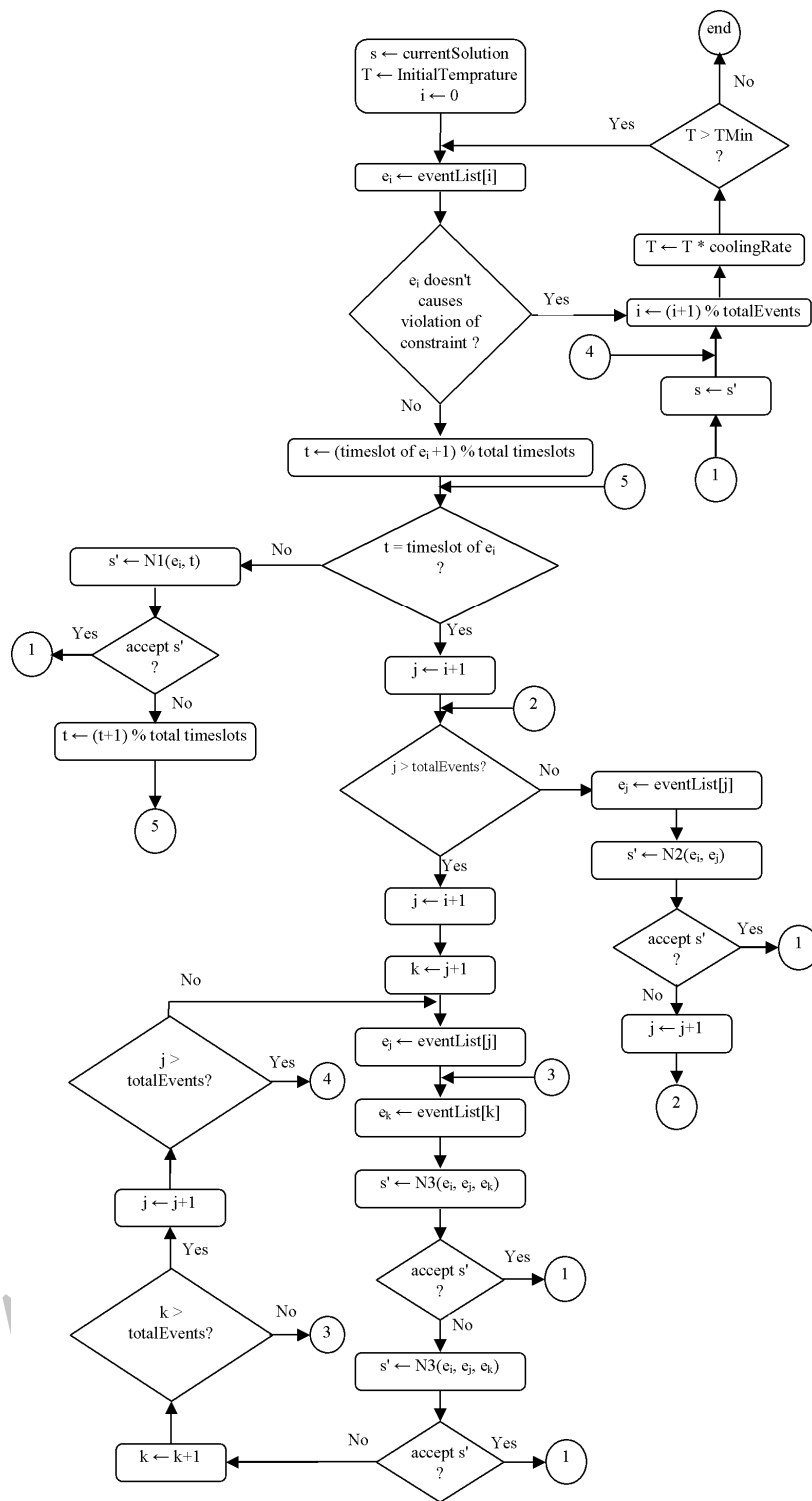
t_x	e_1	e_{10}	e_3	e_6	e_z	e_{20}	e_{34}
t_x	e_2	e_7	e_{15}	e_5	e_x	e_{27}	e_4
t_x	e_{12}	e_{17}	e_{25}	e_{11}	e_y	e_9	e_{14}

شکل ۵: نحوه عملکرد ساختار همسایگی N_3 . رویداد e_x از برش زمانی t_x به برش زمانی t_y و رویداد e_y از برش زمانی t_y به برش زمانی t_z و رویداد e_z از برش زمانی t_z به برش زمانی t_x انتقال یافته است.

جفت رأسی یک یال (محدودیت) اضافه می‌کنیم، آنگاه می‌توان تعداد زیادی از مسایل زمان بندی ساده را به مسأله رنگ آمیزی گراف (و برعکس) تبدیل نمود [۲]. هر برش زمانی را می‌توان به عنوان یک رنگ فرض نمود. جدول ۱ مکاشفه‌های استفاده شده در مقداردهی اولیه را نشان می‌دهد. این مرحله با یک جدول زمان بندی اولیه تهی و لیستی از دروس، L ، که شامل دروس اختصاص نیافته می‌باشد، شروع می‌شود (در ابتدا $|L| = n$ و n تعداد دروس می‌باشد). با استفاده از مکاشفه‌های جدول ۱ هر بار یک درس و یک محل مناسب انتخاب شده و درس به آن محل تخصیص می‌یابد.

۳-۴ رویه جستجوی محلی

هدف از رویه جستجوی محلی در الگوریتم MA تبدیل یک راه حل به یک راه حل بهینه محلی می‌باشد. در این رویه با استفاده از ساختارهای همسایگی^۱ که در ادامه خواهند آمد، یک کروموزوم بهبود داده می‌شود. منظور از بهبود یک کروموزوم کم کردن تعداد نقض محدودیت‌ها در آن



شکل ۶: دیاگرام نحوه عملکرد رویه جستجوی محلی مبتنی بر سردشدن شبیه‌سازی شده.

می‌شود: ابتدا یک درس انتخاب شده و تعداد نقض محدودیت ایجادشده توسط این درس در هر دو والد بررسی می‌شود و ژن شامل این درس از والدی انتخاب می‌شود که تعداد نقض محدودیت کمتری داشته باشد. سپس این ژن به همراه ژن‌های مرتبط با آن (ژن‌های شامل درس‌هایی که دارای دانشجوی مشترک با این درس هستند) به فرزند انتقال می‌یابند و این رویه تا انتقال همه ژن‌ها ادامه می‌یابد. در شرایط برابر، یک والد برای انتخاب ژن به صورت تصادفی انتخاب می‌شود. نحوه عملکرد این رویه در شکل ۷ نشان داده شده است. به عنوان مثال در والد ۱ ژن‌های ۱، ۳، ۵ و ۶ ژن‌های مرتبط با یکدیگر می‌باشند و ژن ۱ از والد ۱ انتخاب شده که همراه با ژن‌های مرتبط خود به فرزند انتقال یافته‌اند.

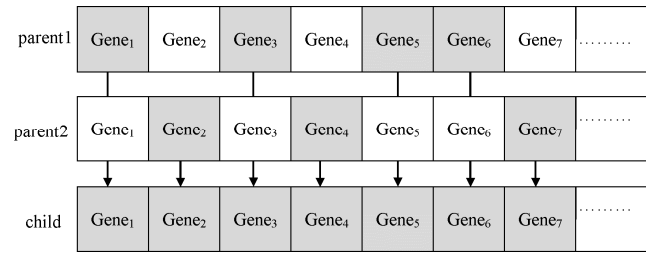
ساختار نیز منجر به بهبود تعداد نقض محدودیت‌ها نشود، انتخاب بعدی استفاده از ساختار همسایگی N^3 می‌باشد. این مراحل برای همه دروس چندین بار تکرار شده و در هر مرحله دما کاهش می‌یابد تا به دمای مینیمم که از قبل توسط کاربر مشخص شده برسد.

۴-۴ عملگر تقاطع حریصانه

تفاوت مهم این عملگر با عملگر تقاطع الگوریتم ممتیک استاندارد در نحوه انتخاب ژن‌ها از والدین و انتقال آنها به فرزند می‌باشد. در این عملگر به جای انتخاب ژن‌ها به صورت تصادفی از رویه زیر استفاده

جدول ۲: پارامترهای استفاده شده MA_SA_UCTP.

پارامتر	مقدار
تعداد نسل	۱۰۰
اندازه جمعیت	۴۰
احتمال تقاطع	۰٫۸
احتمال جهش	۰٫۰۷
دمای اولیه	۲
دمای نهایی	۰٫۰۰۵
نرخ کاهش دما	۰٫۹۹۹



شکل ۷: عملگر تقاطع، انتخاب ژن ۱ از والد اول و انتقال به همراه ژن‌های مرتبط به فرزند. همین امر برای ژن ۲ از والد دوم اتفاق افتاده است.

جدول ۳: مقایسه MA_SA_UCTP با برخی روش‌های موجود. INF به معنای جواب غیر قابل قبول می‌باشد.

نمونه‌ها	MA_SA_UCTP	[۵]	[۶]	[۳۹]	[۱۲]	[۲۱]	[۳۶]	[۳۷]
s1	*	*	*	*	*	۲	۵	*
s2	*	*	*	*	*	۴	۳	*
s3	*	*	*	*	*	۲	۲	*
s4	*	*	*	*	*	*	۳	*
s5	*	*	*	*	*	۴	۱	*
m1	۱۴۴	۳۱۷	۲۴۲	۲۲۱	۲۲۷	۲۴۵	۳۱۶	۲۳۶
m2	۱۴۲	۳۱۳	۱۶۱	۱۴۷	۱۸۰	۲۸۵	۲۴۳	۱۵۸
m3	۲۰۷	۳۵۷	۲۶۵	۲۴۶	۲۳۵	۲۵۱	۲۵۵	۲۶۱
m4	۱۲۶	۲۴۷	۱۸۱	۱۶۵	۱۴۲	۲۲۱	۲۳۵	۱۷۶
m5	۱۶۹	۲۹۲	۱۵۱	۱۳۰	۲۰۰	۲۷۶	۲۱۵	۱۴۷
l	۹۴۲	Inf	Inf	۵۲۹	Inf	۱۰۲۷	Inf	۲۹۶

معیار مناسبی برای مقایسه روش پیشنهادی ما با سایر روش‌های جدید که روی این داده‌ها تست شده‌اند، می‌باشد.

در جدول ۳، MA_SA_ECE_UCTP با برخی روش‌های جدید انجام شده دیگر مقایسه شده است. ستون اول از سمت چپ نمونه‌های مسأله را نشان می‌دهد. در این ستون حرف s^۱، m^۲ و l^۳ به ترتیب نشان‌دهنده نمونه‌های کوچک، متوسط و بزرگ می‌باشند. ستون بعدی نتایج MA_SA_ECE_UCTP را روی نمونه‌های متفاوت نشان می‌دهد. عددهای موجود در هر ستون نشان‌دهنده تعداد نقض محدودیت‌های نرم می‌باشد. شماره مرجع روش‌هایی که با روش پیشنهادی مقایسه شده‌اند در بالای هر ستون آمده است. در هر سطر کمترین (بهترین) مقادیر به صورت پررنگ نمایش داده شده‌اند. چنانچه روشی قادر به پیدا کردن راه حل قابل قبول در یک زمان معقول (در روش‌های مختلف زمان متفاوت در نظر گرفته شده است، از ۳ ساعت تا ۱۰ ساعت) نبوده است، از عبارت Inf^۴ برای آن استفاده شده است.

زمان اجرای الگوریتم پیشنهادی برای نمونه‌های کوچک، متوسط و بزرگ به ترتیب ۶۰، ۳۰۰ و ۶۰۰ دقیقه می‌باشد. نتایج به دست آمده از اجرای این روش نشان می‌دهند که MA_SA_ECE_UCTP در همه موارد قادر به پیدا کردن راه حل قابل قبول می‌باشد.

در نمودار شکل ۸ درصد بهبود روش پیشنهادی نسبت به دو روش [۳۷] و [۳۹] نشان داده شده است. علت انتخاب این دو روش بهتر بودن آنها نسبت به سایر روش‌ها می‌باشد. همان‌طور که در نمودار دیده می‌شود،

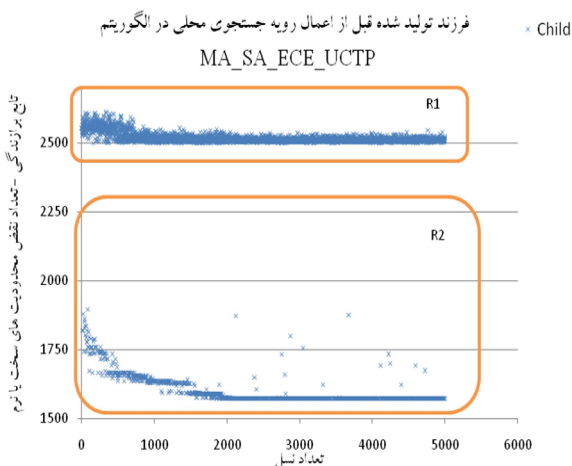
۴-۵ عملگر بهبود کروموزوم

برای توضیح عملگر بهبود کروموزوم، ابتدا باید یک گروه درسی را تعریف کنیم. چنانچه یک درس در ساعت‌های متفاوتی ارائه شود، آنگاه هر ارائه را یک گروه درسی می‌نامند. این عملگر با جابه‌جا کردن دانشجویان از یک گروه درسی به گروه دیگر تعداد نقض محدودیت نرم را کاهش می‌دهد. اگر انتقال یک دانشجو از یک گروه به گروه دیگر باعث کاهش تعداد نقض محدودیت نرم شود، این انتقال انجام می‌شود. به عنوان مثال فرض کنید که دانشجوی x درس y را اخذ کرده است. درس y دارای دو گروه درسی y_۱ و y_۲ می‌باشد. در ابتدا به صورت تصادفی دانشجوی x در یکی از گروه‌های درسی قرار می‌گیرد. به عنوان مثال گروه درسی y_۱ برای دانشجوی x ثبت می‌شود. در حین جستجو برای رسیدن به جواب بهینه، عملگر بهبود تعداد نقض محدودیت نرم را برای این انتساب بررسی می‌نماید و این تعداد را v_۱ می‌نامد. سپس تعداد نقض محدودیت نرم را زمانی که گروه y_۲ برای دانشجوی x ثبت شود، محاسبه کرده و آن را v_۲ نامیده و اگر v_۲ < v_۱ باشد، دانشجوی x از لیست دانشجویان گروه درسی y_۱ حذف شده و به لیست گروه درسی y_۲ اضافه می‌شود. در غیر این صورت عملی اتفاق نخواهد افتاد.

۵- نتایج تجربی

روش MA_SA_ECE_UCTP با زبان برنامه‌نویسی جاوا کد شده و بر روی یک کامپیوتر شخصی با ۵۱۲ mb حافظه و پردازنده Intel ۲٫۰ GHz ارزیابی شده است. پارامترهای استفاده شده برای این روش در جدول ۲ نمایش داده شده‌اند. MA_SA_ECE_UCTP بر روی داده‌های ارائه شده در [۸] اعمال شده است. هر چند این داده‌ها همه محدودیت‌ها و جنبه‌های یک مسأله واقعی را نشان نمی‌دهند [۹] ولی

1. Small
2. Medium
3. Large
4. Infeasible



شکل ۱۰: تابع برازندگی فرزند تولید شده در الگوریتم MA_SA_ECE_UCTP قبل از اعمال رویه جستجوی محلی. ناحیه R1 فرزندان غیر قابل قبول (دارای نقض محدودیت سخت) و ناحیه R2 فرزندان قابل قبول (عدم نقض محدودیت سخت).

تعداد دروس نیم‌سال اول ۸۹-۸۸: این تعداد شامل ۷۷ درس می‌باشد که ۵۶ درس تئوری و بقیه دروس آزمایشگاه هستند. لازم به ذکر است که برخی از دروس در چند گروه تشکیل می‌گردند (۲ یا ۳ گروه). به عبارتی دیگر تعداد گروه‌های نظری ۸۴ گروه و تعداد گروه‌های عملی ۷۷ گروه می‌باشند.

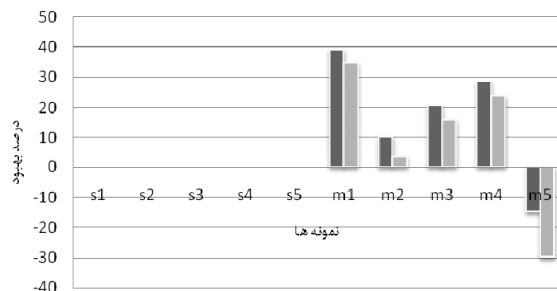
پارامترهای استفاده شده در این روش همان پارامترهای ذکر شده در جدول ۲ می‌باشد.

همان‌طور که در بخش ۴ اشاره شد، این الگوریتم ترکیبی از الگوریتم ممتیک و جستجوی محلی مبتنی بر الگوریتم سردشدن شبیه‌سازی شده می‌باشد. در این الگوریتم به جای رویه مقداره‌دهی اولیه کروموزوم‌ها به صورت تصادفی، از رویه مقداره‌دهی اولیه مکاشفه‌ای استفاده شده است. همچنین لازم به یادآوری است که عملگر تقاطع استفاده شده در این الگوریتم عملگر تقاطع حریمانه می‌باشد. در این الگوریتم از رویه بهبود برای بالابردن هرچه بیشتر کیفیت راه حل‌های تولید شده استفاده شده است.

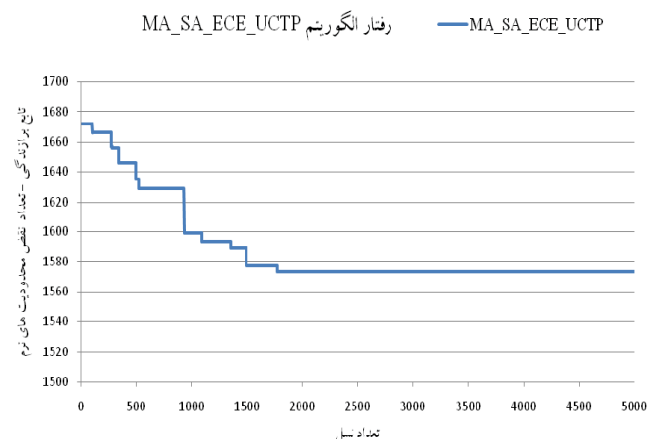
برای تمایز بین فرزندان دارای نقض محدودیت سخت و فرزندان دارای نقض محدودیت نرم، یک عدد ثابت که در اینجا ۲۵۰۰ در نظر گرفته شده است، به مقدار تابع برازندگی آن دسته از راه حل‌های دارای نقض محدودیت سخت اضافه می‌کنیم.

رفتار کلی الگوریتم بر اساس تابع برازندگی بهترین عضو جمعیت آن به صورت نمودار شکل ۹ می‌باشد. در این نمودار محور عمودی نشان‌دهنده تعداد نقض محدودیت‌ها (مقدار تابع برازندگی) و محور افقی تعداد نسل را نشان می‌دهد. همان‌طور که در نمودار دیده می‌شود، وجود رویه مقداره‌دهی اولیه مکاشفه‌ای باعث ایجاد راه حل‌های اولیه قابل قبول شده است (مقدار تابع برازندگی آنها کمتر از عدد ۲۵۰۰ می‌باشد). در نسل‌های تقریباً کوچک‌تر از ۲۰۰۰ مقدار تابع برازندگی سیر نزولی داشته و همواره راه حل به دست آمده بهبود داده شده است. در نسل‌های بالاتر از ۲۰۰۰ هیچ‌گونه بهبودی در راه حل جواب دیده نمی‌شود. این امر نشان‌دهنده همگرایی در جمعیت و عدم توانایی الگوریتم در بهبود بیشتر راه حل می‌باشد.

در نمودار شکل‌های ۱۰ و ۱۱ رفتار فرزند تولید شده در این الگوریتم نشان داده شده است. همین مسأله در شکل ۱۰ و ۱۱ باعث به وجود آمدن دو ناحیه R1 و R2 شده است. ناحیه R1 نشان‌دهنده راه حل‌هایی است که دارای نقض محدودیت سخت می‌باشند و ناحیه R2 راه حل‌هایی را نشان می‌دهد که قابل قبول بوده و دارای نقض محدودیت نرم می‌باشند.



شکل ۸: مقایسه MA_SA_ECE_UCTP با روش‌های [۳۷] و [۳۹]. محور افقی نمونه‌های تست، محور عمودی درصد بهبود، میله‌های پررنگ روش [۳۷] و میله‌های کم‌رنگ روش [۳۹] را نشان می‌دهند.



شکل ۹: رفتار الگوریتم MA_SA_ECE_UCTP بر اساس تابع برازندگی بهترین عضو جمعیت در طول ۵۰۰۰ نسل.

روش معرفی شده در همه نمونه‌ها به جز m5 از دو روش دیگر بهتر بوده است (درصد مثبت). در نمونه m5 درصد منفی شده است و به معنای بدتر بودن روش معرفی شده نسبت به دو روش دیگر است. اگر راه حل به دست آمده از MA_SA_ECE_UCTP و روش قابل مقایسه در یک نمونه مسأله، به ترتیب s و s' باشد و f مقدار برازندگی راه حل‌ها در دو روش مذکور باشد، آنگاه درصد بهبود به صورت زیر محاسبه می‌شود

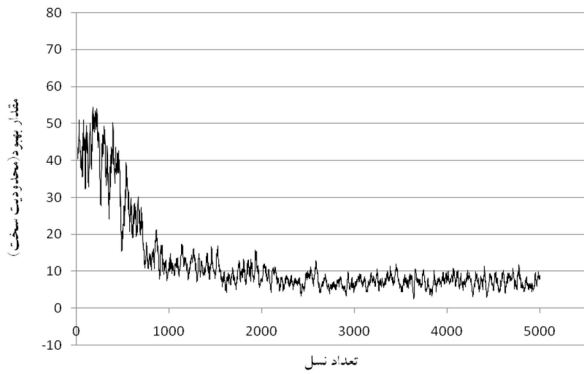
$$ImpPercent = - \frac{(f(s) - f(s')) \times 100}{f(s')}$$

بیشترین بهبود در نمونه m1 دیده می‌شود. میزان بهبود در نمونه m1 نسبت به روش [۳۹] برابر ۳۴/۸ و نسبت به روش [۳۷] برابر ۳۸/۹ می‌باشد.

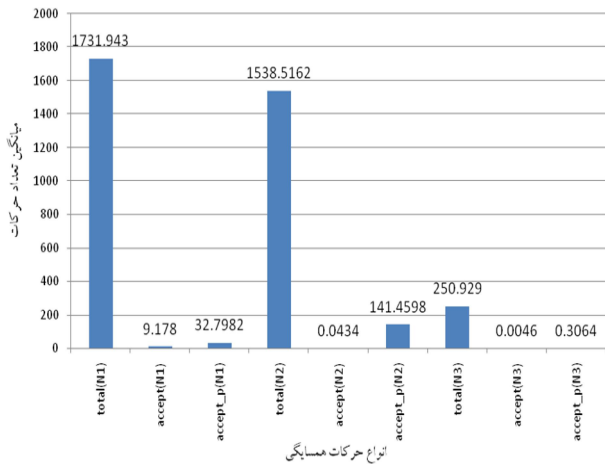
در قسمت دوم این بخش به ارزیابی روش MA_SA_ECE_UCTP در اعمال بر داده‌های واقعی دانشکده برق و کامپیوتر می‌پردازیم. داده‌های جمع‌آوری شده برای این مقاله مربوط به نیم‌سال اول ۸۹-۸۸ می‌باشد. این داده‌ها بر اساس اطلاعات آماری ثبت نام مقدماتی در نیم‌سال مذکور تهیه شده‌اند که بعد از اعمال مراحل پیش‌پردازش و آماده کردن آنها، در بانک اطلاعاتی برای اجرای برنامه ذخیره شده‌اند. این اطلاعات عبارتند از: تعداد کل دانشجویان کارشناسی دانشکده: ۹۷۶ نفر. این تعداد شامل دانشجویان کارشناسی رشته تحصیلی برق (کلیه گرایش‌ها) و دانشجویان رشته تحصیلی کامپیوتر (کلیه گرایش‌ها) می‌باشد.

تعداد اساتید: ۵۵ نفر که از این تعداد ۴۳ نفر اعضای هیأت علمی و ۱۲ نفر شامل اساتید مدعو و دانشجویان دوره دکتری که درس ارائه می‌کنند. تعداد کلاس‌ها: ۱۴ اتاق که ۴ کلاس برای دروس ارشد، ۲ کلاس مشترک کارشناسی و ارشد و بقیه برای دروس کارشناسی می‌باشند.

تقریب مقدار بهبود تابع ارزیاب فرزند بعد از اعمال جستجوی محلی



شکل ۱۳: تقریب مقدار بهبود تابع ارزیاب فرزند بعد از اعمال جستجوی محلی در الگوریتم MA_SA_ECE_UCTP.



شکل ۱۴: میانگین تعداد دفعات استفاده شده از ساختارهای همسایگی N_1 ، N_2 و N_3 در اعمال رویه جستجوی محلی بر فرزند تولید شده در هر نسل.

در نمودار شکل ۱۳ تقریب مقدار بهبود تابع ارزیاب فرزند بعد از اعمال جستجوی محلی در طول همه نسل‌ها مشخص شده است. این بهبود فقط شامل بهبود نقض محدودیت سخت می‌باشد. در این نمودار محور عمودی تعداد بهبود نقض محدودیت سخت را نشان می‌دهد و محور افقی نمایانگر تعداد نسل می‌باشد. همان‌طور که مشاهده می‌شود این مقدار بهبود با افزایش نسل کاهش یافته و به زیر عدد ۱۰ رسیده است. علت این امر این است که با افزایش تعداد نسل، تعداد فرزندان که نقض محدودیت سخت در آنها اتفاق می‌افتد، کاهش می‌یابد و در نتیجه تأثیر رویه جستجوی محلی بر آنها چشم‌گیر نخواهد بود.

با توجه به شکل ۱۳ تأثیر جستجوی محلی در بهبود نقض محدودیت سخت در نسل‌های ابتدایی چشم‌گیر می‌باشد. با همگرا شدن جمعیت و یکسان شدن اعضای آن مشاهده می‌شود که مقدار این بهبود کمتر می‌شود. این مقدار بهبود تقریباً از نسل‌های ۲۰۰۰ به بعد یکنواخت شده و تا پایان اجرا تغییر زیادی نداشته است.

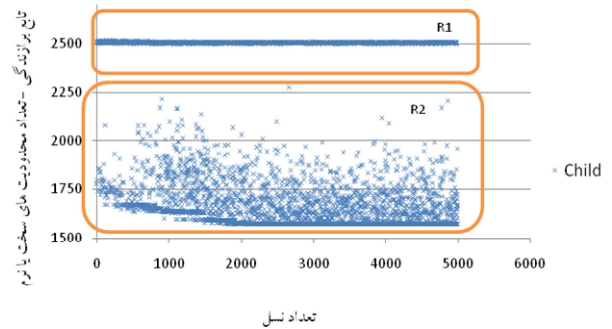
در نمودار شکل ۱۴ میانگین تعداد دفعات استفاده از ساختارهای همسایگی در هر بار اعمال جستجوی محلی بر فرزند تولید شده در هر نسل نشان داده شده است.

در این نمودار محور عمودی نمایانگر میانگین تعداد دفعات استفاده از ساختارهای همسایگی می‌باشد و عناصر محور افقی به شرح ذیل می‌باشند: $total(N_1)$: میانگین تعداد کل حرکات انجام شده که منجر به ساختار همسایگی N_1 شده‌اند.

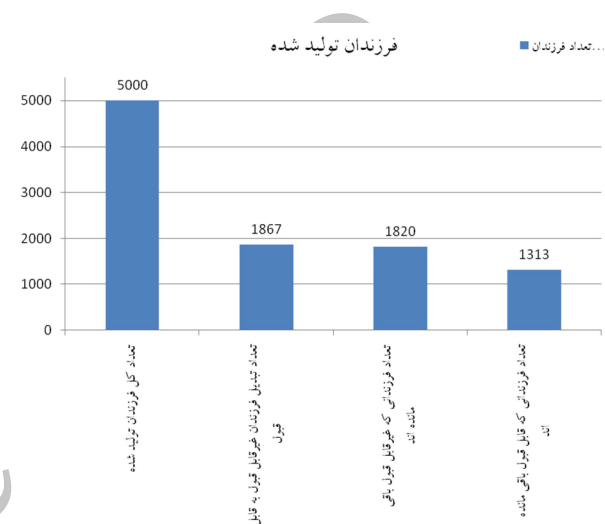
$accept(N_1)$: میانگین تعداد استفاده از ساختار همسایگی نوع

فرزند تولید شده بعد از اعمال رویه جستجوی محلی در الگوریتم

MA_SA_ECE_UCTP



شکل ۱۱: تابع برازندگی فرزند تولید شده در الگوریتم MA_SA_ECE_UCTP بعد از اعمال رویه جستجوی محلی. ناحیه R_1 فرزندان غیر قابل قبول و ناحیه R_2 فرزندان قابل قبول.



شکل ۱۲: تعداد فرزندان در حالت‌های متفاوت بعد از اعمال رویه جستجوی محلی در الگوریتم MA_SA_ECE_UCTP.

نمودار شکل ۱۰ مقدار تابع برازندگی فرزند تولید شده در طول نسل‌ها را نشان می‌دهد. این مقدار تابع برازندگی، قبل از اعمال رویه جستجوی محلی بر فرزندان می‌باشد. با توجه به نمودار مشاهده می‌شود که تعداد زیادی از فرزندان تولید شده قبل از اعمال رویه جستجوی محلی در ناحیه R_1 (دارای نقض محدودیت سخت) می‌باشند. در شکل ۱۱ مجدداً مقدار تابع برازندگی فرزند تولید شده در طول نسل‌ها بعد از اعمال رویه جستجوی محلی نمایش داده شده است.

با توجه به شکل ۱۱ مشاهده می‌شود که تعداد زیادی از فرزندان تولید شده که در ناحیه R_1 بودند (شکل ۱۰) بعد از اعمال رویه جستجوی محلی به ناحیه R_2 منتقل شده‌اند.

در نمودار شکل ۱۲ تأثیر این رویه بر تابع ارزیاب فرزندان تولید شده به صورت آماری نشان داده شده است.

در نمودار شکل ۱۲ ستون اول از سمت چپ تعداد کل فرزندان تولید شده را در طی نسل‌ها نشان می‌دهد و ستون دوم از سمت چپ تعداد فرزندان را نشان می‌دهد که بعد از اعمال رویه جستجوی محلی از حالت غیر قابل قبول به حالت قابل قبول تبدیل شده‌اند. با توجه به اعداد نشان داده شده در این نمودار، $37/34$ درصد فرزندان دچار این تغییر شده‌اند. $36/4$ درصد از فرزندان تولید شده دارای حالت غیر قابل قبول بوده‌اند و بعد از اعمال رویه جستجوی محلی نیز در این حالت باقی مانده‌اند. $26/26$ درصد نیز از فرزندان بعد از تولید، دارای حالت قابل قبول بوده‌اند.

است که در این بین الگوریتم SA نتایج بهتری تولید کرده است و این نتایج در بخش‌های قبلی ارائه شد. در کارهای آینده می‌توان از ایجاد عملگرهای جدید برای MA (مانند عملگر بهبود در این مقاله) و یا تغییر عملگرهای استاندارد آن بهره برد. اعمال رویه‌های مکاشفه‌ای در ارضای محدودیت‌های نرم نیز از جمله کارهایی است که می‌توان در حل این مسأله مورد بررسی قرار گیرد. در اکثر الگوریتم‌های فرامکاشفه‌ای مشکل عمده‌ای که وجود دارد نقاط بهینه محلی و گیرافتادن الگوریتم در این نقاط می‌باشد. همان‌طور که در شکل ۱۱ مشاهده شد، الگوریتم از نسل تقریباً ۲۰۰۰ به بعد دچار همگرایی شده و بهبودی در آن مشاهده نمی‌شود. نیاز به یک مکانیزم برای ایجاد تشویش در جمعیت جهت رهایی از نقاط بهینه محلی احساس می‌شود. بهره‌گیری از ساختارهای همسایگی مختلف برای رهایی از نقاط بهینه محلی یکی از راهکارهایی است که می‌توان برای رفع این مشکل به کار برد. به‌عنوان مثال ساختار همسایگی N۳ با توجه به جدول ۴ در بهبود جمعیت تولیدشده خیلی مؤثر نبوده و می‌تواند با یک ساختار همسایگی مطلوب‌تر جایگزین شود. علاوه بر استفاده از انواع ساختارهای همسایگی متفاوت، نحوه انتخاب نقاط همسایه به‌طور مؤثر نیز در رهایی الگوریتم از نقاط بهینه محلی مفید خواهد بود. استفاده از رویه‌های جستجوی محلی متفاوت دیگر، ترکیب این روال‌های جستجو برای ایجاد یک رویه جستجوی مؤثر نیز از جمله پیشنهادهایی می‌باشند که می‌توان در کارهای آینده از آنها استفاده نمود.

۷- قدردانی

نویسندگان این مقاله بدین وسیله از سرکار خانم منصف کارشناس محترم آموزش دانشکده برق، کامپیوتر و فناوری اطلاعات دانشگاه صنعتی اصفهان به‌خاطر همکاری بی‌دریغ در راستای انجام پروژه مربوطه قدردانی می‌نمایند.

مراجع

- [1] E. Burke, J. Kingston, K. Jackson, and R. Weare, "Automated university timetabling: the state of the art," *The Computer J.*, vol. 40, no. 9, pp. 565-571, 1997.
- [2] R. Lewis, "A survey of metaheuristic-based techniques for university timetabling problems," *Operations Research*, vol. 30, no. 1, pp. 167-190, Jan. 2008.
- [3] D. Costa, "A tabu search algorithm for computing an operational timetable," *Eur J. Oper. Res.*, vol. 79, no. 1, pp. 98-110, Jul. 1994.
- [4] L. Gislén, C. Peterson, and B. Soderberg, "Teachers and classes with neural networks," *Int J. Neural Syst.*, vol. 1, no. 2, pp. 167-176, Nov. 1989.
- [5] D. Come, P. Ross, and H. Fang, "Evolving timetables," in *Lance C. Chambers (ed.) The Practical Handbook of Genetic Algorithms*, CRC, Florida, vol. 1, pp. 219-276, 1995.
- [6] M. Carrasco and M. Pato, "A multiobjective genetic algorithm for the class/teacher timetabling problem," in E. Burke and W. Erben (eds.) *Practice and Theory of Automated Timetabling (PATA T) III*, Springer, Berlin, vol. 2079, pp. 3-17, 2001.
- [7] D. Abramson, M. Krishnamoorthy, and H. Dang, "Simulated annealing cooling schedules for the school timetabling problem," *Asia Pacific Journal of Operational Research*, vol. 16, pp. 1-22, 1996.
- [8] E. Burke, Y. Bykov, and M. Petrovic, "A multicriteria approach to examination timetabling," in E. Burke and W. Erben (ed.) *Practice and Theory of Automated Timetabling (PATAT) III*, Springer, Berlin, vol. 2070, pp. 118-131, Jan. 2001.
- [9] A. Alkan and E. Ozcan, "Memetic algorithms for timetabling evolutionary computation," in *Proc. of the 2003 IEEE Congress on Evol. Comput.*, vol. 3, pp. 1796-1802, Dec. 2003.
- [10] P. Moscato, On Evolution, Search, Optimization, Genetic Algorithms, and Martial Art: Towards Memetic Algorithms, Caltech Concurrent Computation Program, Technical Report, 1989.
- [11] P. Moscato and C. Cotta, "A gentle introduction to memetic algorithms," *Handbook of Metaheuristics*, pp. 105-144, 2003.

جدول ۴: میانگین تعداد دفعات استفاده‌شده از ساختارهای N۱، N۲ و N۳ در اعمال رویه جستجوی محلی بر فرزند تولیدشده در یک نسل که به‌صورت درصد نشان داده شده است.

	total	accept	accept_p
Total (moves)	۳۵۲۱,۳۸۸۲	۹,۲۲۶	۱۷۴,۵۶۴۴
N۱	۴۹,۱۸۳۵٪	۰,۵۲۹۹٪	۱,۸۹۳۷٪
N۲	۴۳,۶۹۰۶٪	۰,۰۰۲۸٪	۹,۱۹۴۶٪
N۳	۷,۱۲۵۹٪	۰,۰۰۱۸٪	۰,۱۲۲۱٪

N۱ که منجر به راه حل‌های جدید بهتر از راه حل جاری شده‌اند. این راه حل‌های جدید همگی پذیرفته می‌شوند. accept_p(N۱): میانگین تعداد استفاده از ساختار همسایگی نوع N۱ که منجر به راه حل‌های بدتر نسبت به راه حل جاری شده‌اند ولی با توجه به شرط پذیرش در الگوریتم سردشدن شبیه‌سازی شده با احتمال p، پذیرفته شده‌اند. برای ساختارهای همسایگی N۲ و N۳ نیز ستون‌هایی به‌ترتیب بالا رسم شده است. در جدول ۴ اعداد موجود در نمودار بالا به‌صورت درصد نشان داده شده است.

همان‌طور که ملاحظه می‌شود بیشترین تعداد استفاده از ساختار همسایگی N۱ و کمترین تعداد مربوط به ساختار همسایگی N۳ می‌باشد. همچنین تأثیر ساختار همسایگی N۳ با توجه به جدول ۴ در پذیرش راه حل‌های بهتر بسیار کم بوده و اعمال یک ساختار همسایگی جدید به‌جای آن می‌تواند پیشنهاد شود.

عدددهای موجود در این جدول مربوط به میانگین حرکات در جستجوی محلی برای یک فرزند می‌باشد. همان‌طور که در جدول دیده می‌شود ۴۹,۱۸۳۵ درصد از کل حرکات منجر به ساختار همسایگی N۱ شده‌اند که از این تعداد ۰,۵۲۹۹ درصد ساختارهایی بوده‌اند که راه حل‌های جدید بهتر تولید کرده‌اند و از میان راه حل‌های تولیدشده بدتر ۱,۸۹۳۷ درصد پذیرفته شده‌اند. برای ساختارهای همسایگی N۲ و N۳ نیز ترتیب عدددهای موجود در جدول به همین صورت می‌باشد.

۶- نتیجه‌گیری و کارهای آینده

در این مقاله MA و ترکیب آن با رویه جستجوی محلی مبتنی بر SA برای حل UCTP مورد بررسی قرار گرفت. استفاده از رویه مقداردهی اولیه مکاشفه‌ای در ارضای محدودیت‌های سخت و قابل قبول شدن راه حل‌های اولیه بسیار مؤثر بود. همان‌گونه که مشاهده گردید الگوریتم ارائه‌شده در مقایسه با تعدادی از کارهای جدید انجام‌شده در زمینه مسأله زمان‌بندی درسی دانشگاهی در اکثر نمونه‌های تست بسیار بهتر می‌باشد. یکی دیگر از مزایای الگوریتم ارائه‌شده، کاربردی بودن آن است به‌نحوی که MA_SA_ECE_UCTP بر داده‌های حقیقی دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان نیز اعمال شده است. نتایج به‌دست آمده نشان می‌دهند که این کار با کار انجام‌شده به‌صورت دستی توسط کارشناس خبره قابل مقایسه و در اکثر موارد از نظر مدت زمان لازم برای تولید جدول زمان‌بندی بهتر می‌باشد. چنانچه بتوانیم UCTP در این دانشکده را به درستی فرموله کرده و حالات استثنا را برطرف نماییم، می‌توان از این الگوریتم استفاده نمود. تأثیر روال جستجوی محلی در تبدیل راه حل‌های کاندید به راه حل‌های قابل قبول و رسیدن به جواب‌های بهینه از جمله نتایج این کار می‌باشد. در این کار ترکیب روال‌های جستجوی محلی متفاوتی با الگوریتم MA به کار برده شده

- Evolutionary Computation in Combinatorial Optimization (EvoCOP 2003)*, vol. 2611, pp. 334-345, Springer, Berlin, Aug. 2003.
- [33] K. Socha, J. Knowles, and M. Samples, "A max - min ant system for the university course timetabling problem," in *Proc. of the 3rd Int. Workshop on Ant Algorithms, ANTS 2002, Springer Lecture Notes in Computer Science*, vol. 2463, pp. 1-13, Sep. 2002.
- [34] <http://iridia.ulb.ac.be/supp/IridiaSupp2002-001/index.html>
- [35] B. McCollum, "University timetabling: bridging the gap between research and practice," in *Proc. of the 6th Int Conf. on the Practice and Theory of Automated Timetabling*. Lecture Notes in Computer Science, vol. 3867, pp. 3-23, 2006.
- [36] M. A. Al-Betar, A. T. Khader, and T. A. Gani, "A harmony search algorithm for university course timetabling," in E. Burke and M. Gendreau. (eds.), *The Proc. of the 7th Int. Conf. on the Practice and Theory of Automated Timetabling (PATAT 2008)*, Montreal, Canada, 18-22 Aug. 2008.
- [37] S. Abdulla and A. R. Hamdan, "A hybrid approach for university course time tabling," *IJCSNS*, vol. 8, no. 8, pp. 127-131, Aug. 2008.
- [38] D. J. A. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and it's application to timetabling problems," *Computer J.*, vol. 10, no. 1, pp. 85-86, 1967.
- [39] S. Abdullaha, E. K. Burke, and B. Mccollum, "A hybrid evolutionary approach to the university course timetabling problem," in *Proc. of the IEEE Congress on Evolutionary Computation*, vol. 1, pp. 1764-1768, Sep. 2007.
- [40] E. K. Burke, B. McCollum, A. Meisels, S. Petrovic, and R. Qu, "A graph-based hyper-heuristic for educational timetabling problems," *European Journal of Operational Research*, vol. 176, pp. 177-192, Jan. 2007.
- مجید جودگی** در سال ۱۳۸۳ مدرک کارشناسی مهندسی کامپیوتر (نرم افزار) خود را از دانشگاه شهید چمران اهواز و در سال ۱۳۸۸ مدرک کارشناسی ارشد مهندسی کامپیوتر (هوش مصنوعی) خود را از دانشگاه صنعتی اصفهان دریافت نمود. از سال ۱۳۸۴ الی ۱۳۸۶ نامبرده به عنوان کارشناس و مدرس دروس عملی در دانشگاه آزاد به کار مشغول بود و پس از اتمام دوره کارشناسی ارشد خود از سال ۱۳۸۸ در دانشکده مهندسی دانشگاه آزاد مشغول به فعالیت گردید و اینک نیز عضو هیأت علمی این دانشکده می باشد. زمینه های علمی مورد علاقه نامبرده شامل موضوعاتی مانند یادگیری ماشین، محاسبات نرم، الگوریتم های تکاملی و داده کاوی می باشد.
- محمد علی منتظری** تحصیلات خود را در دوره کارشناسی (رشته مهندسی الکترونیک) و کارشناسی ارشد (رشته مهندسی سخت افزار کامپیوتر) در دانشگاه هیوستن آمریکا در سال ۱۳۵۷ و دکترای خود را در رشته هوش مصنوعی و رباتیک در دانشگاه یومیسست انگلستان در سال ۱۳۷۵ به پایان رسانید. وی از سال ۱۳۷۵ به عنوان عضو هیأت علمی دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان به تدریس دروس سخت افزار و نرم افزار کامپیوتر در مقطع کارشناسی و کارشناسی ارشد مشغول بوده و تحقیقات خود را نیز در این زمینه ادامه داده است. دکتر منتظری از سال ۱۳۷۵ تا سال ۱۳۸۱ مسئولیت ریاست دانشکده برق و کامپیوتر را به عهده داشته، که در این مدت فعالیت های رباتیک دانشگاه صنعتی اصفهان را پایه گذاری نمود. وی همچنین از سال ۱۳۸۲ به مدت ۲ سال با دانشگاه دیواری کالیفرنیا ای آمریکا به عنوان عضو مدعو همکاری داشته است.
- سید رسول موسوی** با استعانت از خداوند متعال تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد در رشته مهندسی کامپیوتر به ترتیب از دانشگاه صنعتی شریف و دانشگاه صنعتی امیرکبیر و در مقطع دکتری در رشته کامپیوتر در سال ۱۳۸۵ از دانشگاه امپریال کالج به پایان رسانده و هم اکنون استادیار گروه مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی اصفهان می باشد. نامبرده همچنین در طی بیست سال اخیر در داخل و خارج کشور در حوزه مهندسی نرم افزار مشغول به فعالیت بوده است. زمینه های تحقیقاتی و صنعتی مورد علاقه نامبرده شامل نظریه محاسبات و الگوریتم ها به ویژه در حوزه مسائل ان پی- سخت، توانایی های محاسباتی بالقوه در طبیعت، زیست اطلاعات و زیست شناسی محاسباتی و مهندسی نرم افزار بخصوص در حوزه های کلان صنعت می باشد.
- [12] O. Rossi-Doria and B. Paechter, *A Memetic Algorithm for University Course Timetabling*, in *Combinatorial Optimisation 2004 Book of Abstracts*, Lancaster, UK, Lancaster University, 2004.
- [13] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [14] E. Burke, Y. Bykov, J. Newall, and S. Petrovi, "A time-predefined approach to course imetabling," *J. of Operations Research*, vol. 13, no. 2, pp. 139-151, 2003.
- [15] M. Nandhini and D. S. Kanmani, "A survey of simulated annealing methodology for university course timetabling," *Int. J. of Recent Trends in Engineering*, vol. 1, no. 2, pp. 255-257, May 2009.
- [16] M. Chiarandini, K. Socha, M. Birattari, and O. Rossi - Doria, *An Effective Hybrid Approach for the University Course Timetabling Problem*, FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep AIDA-2003-05, 2003.
- [17] A. Schaerf, "Tabu search techniques for large high - school timetabling problems," in *Proc. of the 13th National Conf. on Artificial Intelligence*, vol. 1, no. 13, pp. 363-368, Aug. 1996.
- [18] C. H. Aladag, G. Hocaoglu, and M. A. Basaran, *The Effect of Neighborhood Structures on Tabu Search Algorithm in Solving Course Timetabling Problem*, Expert Systems with Applications, doi: 10.1016/j.eswa.2009.04.051, 2009.
- [19] C. H. Aladag and G. Hocaoglu, "A tabu search algorithm to solve course timetabling problem," *Hacettepe J. of Mathematics and Statistics*, vol. 36, no. 1, pp. 53-64, Feb. 2007.
- [20] R. Alvarez, E. Crespo, and J. M. Tamarit, "Design and implementation of a course scheduling system using tabu search," *European J. of Operational Research*, vol. 137, no. 3, pp. 512-523, Mar. 2002.
- [21] S. Abdullaha and H. Turabieh, "Generating university course timetabling using genetic algorithms and local search," in *Proc. Int. Conf. on Convergence and Hybrid Information Technolog*, vol. 1, pp. 254-260, Nov. 2008.
- [22] D. Landa - Silva and J. H. Obit, "Great deluge with nonlinear decay rate for solving course timetabling problems," in *Proc. of the 2008 IEEE Conf. on Intelligent Systems (IS 2008)*, vol. 1, pp. 8-11, Nov. 2008.
- [23] D. Landa-Silva and J. Henry Obit, "Evolutionary non - linear great deluge for university course timetabling," in *Proc. of the 2009 Int. Conf. on Hybrid Artificial Intelligence Systems (HAIS 2009)*, Lecture Notes in Artificial Intelligence, Springer, vol. 5572, pp. 269-276, Jun. 2009.
- [24] D. Abramson, "Constructing school timetables using simulated annealing: sequential and parallel algorithms," *Manag Sci*, vol. 37, no. 1, pp. 98-113, Jan. 1991.
- [25] F. Melicio and J. Caldeira, "Timetabling implementation aspects by simulated annealing," in *Proc. IEEE Systems Science and Systems Engineering*, vol. 1, pp. 553-557, 1998.
- [26] B. Paechter, A. Cumming, M. G. Norman, and H. Luchian, "Extensions to a memetic timetabling system," in *Proc. of the 1st Int. Conf. on Practice and Theory of Automated Timetabling*, LNCS 1153, vol. 1153, pp. 251-265, Jul. 1996.
- [27] S. Elmohamed, G. Fox, and P. Coddington, "A comparison of annealing techniques for academic course scheduling," in E. Burke and M. Carter (eds.) *Practice and Theory of Automated Timetabling (PATAT) II*, vol. 1408, pp. 146-166, Springer: Berlin, 1998.
- [28] O. Rossi - Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Stutzle, "A comparison of the performance of different metaheuristics on the timetabling problem," *Lecture Notes in Computer Science 2740*, vol. 2740, pp. 329-351, Aug. 2002.
- [29] S. Petrovic and Y. Bykov, "A multiobjective optimisation approach for exam timetabling based on rajectories," in E. Burke and P. De Causmaecker (eds.) *The Practice and Theory of Automated Timetabling (PATAT) IV*, vol. 2740, pp. 181-194, Springer, Berlin, Sep. 2003.
- [30] R. Dawkins, *The Selfish Gene*, Oxford University Press, 1976.
- [31] L. Paquete and C. Fonseca, "A study of examination timetabling with multiobjective evolutionary algorithms," in *Proc. of Fourth Metaheuristics Int. Conf., MIC 2001*, Porto, pp. 149-154, Jul. 2001.
- [32] K. Socha and M. Samples, "Ant algorithms for the university course timetabling problem with egard to the state-of-the-art," in