1086 – 1389 – 10 – 21

# بررسی مساله زمانبندی‌درسی دانشگاهی با استفاده از ترکیب الگوریتم ممتیک بهبودیافته و الگوریتم سردشدن شبیه‌سازی شده

مجید جودکی [1]، محمد علی منتظری*[2] ، سید رسول موسوی [3]

**چکیده - مساله زمانبندی به عنوان یکی از مسائل پیچیده‌ی بهینه‌سازی شناخته می‌شود. یک نمونه از مسائل زمانبندی، برنامه‌ریزی درسی دانشگاهی است که هر نیمسال در دانشکده‌های یک دانشگاه انجام می‌شود. تنوع محدودیت‌ها در این مساله باعث می‌شود برنامه‌ریزی در دانشکده‌های مختلف به گونه‌های متفاوتی انجام شود. کارهای زیادی برای حل این مساله انجام شده است که اکثر آنها از روش‌های فرامکاشفه‌ای[۱] بهره برده‌اند. در این مقاله یک مساله زمانبندی واقعی مورد بررسی قرار می‌گیرد و یک روش مبتنی بر الگوریتم ممتیک[۲]() بهبودیافته که از الگوریتم سردشدن شبیه‌سازی شده[۳] به عنوان رویه جستجوی محلی[٤] خود استفاده می‌کند، ارائه می‌شود. منظور از بهبود در الگوریتم ممتیک، استفاده از روش‌های مکاشفه‌ای[٥] در تولید جمعیت اولیه و همچنین تغییر عملگر تقاطع[٦] در این الگوریتم می‌باشد. همچنین یک عملگر به نام عملگر بهبود[۷] جهت بهبود راه‌حل‌های تولید شده و کاهش تعداد نقض محدودیت‌ها طراحی شده است. بکارگیری روش سردشدن شبیه‌سازی شده به عنوان رویه جستجوی محلی در الگوریتم ممتیک باعث افزایش توانایی بهره‌برداری[۸] این الگوریتم خواهد شد. کارآمدی این روش در مقایسه با برخی روش‌های جدید، با توجه به نتایج به‌دست آمده بر روی داده‌های استاندارد نشان داده شده است. همچنین مقایسه نتایج حاصل از این روش با روش انجام شده به صورت دستی بر روی داده‌های واقعی نشان دهنده برتری این روش می‌باشد.**

***کلید واژه-*** **الگوریتم سرد شدن شبیه سازی شده، الگوریتم ممتیک، جستجوی محلی، مساله زمانبندی درسی دانشگاهی.**

## 1- مقدمه

تخصیص دروس به کلاس‌های معین در دوره‌های زمانی[۹] مشخص را مساله زمانبندی درسی دانشگاهی[۱۰] (UCTP) می‌گویند[1]. منظور از کلاس، مکان تشکیل درس و منظور از دوره زمانی، زمان تشکیل آن می‌باشد. در این مساله هر درس را به عنوان یک رویداد تلقی می‌کنند. اگر رویداد نشان دهنده یک امتحان باشد آن مساله را زمانبندی امتحانی[۱۱] می‌گویند. این دو نوع مساله طبیعتا با یکدیگر متفاوت هستند[1]. زمانبندی درسی هر هفته تکرار می‌شود در صورتی که زمانبندی امتحانی فقط یک بار در طول ترم تکرار خواهد شد. در زمانبندی درسی دو درس نمی‌توانند به صورت همزمان در یک کلاس تشکیل شوند در صورتی که در زمانبندی امتحانی این امر امکان‌پذیر است. به عنوان مثال در یک سالن می‌توان چند امتحان را به صورت همزمان برگزار نمود. در زمانبندی درسی یک دانشجو تمایل به داشتن چندین درس در یک روز دارد ولی در زمانبندی امتحانی علاقه ای به داشتن چند امتحان در یک روز وجود ندارد[2].

در یک UCTP برگزاری همزمان دو درس با دانشجویان و یا اساتید مشترک در یک دوره زمانی یکسان را تداخل[۱۲] می‌گویند. هدف از بررسی یک UCTP در حالت کلی کاهش تعداد تداخل بین دروس با دانشجویان و یا مدرسان مشترک و همچنین رفع همزمانی دروسی است که به یک اتاق مشترک نیاز دارند[2]. برای موسسه‌های بزرگ مانند دانشگاه‌ها با افزایش تعداد محدودیت‌ها[۱۳] رسیدن به یک جواب قابل‌قبول[۱٤] بسیار مشکل‌تر خواهد بود.

حل یک UCTP کار بسیار مشکلی است و علت این امر بزرگ بودن اندازه و متفاوت بودن ساختار و ماهیت انواع این مساله می‌باشد. برای بررسی و حل UCTP روش‌های متفاوتی ارائه شده است. روش‌های مبتنی بر رنگ‌آمیزی گراف[۱٥]، انواع مکاشفه‌ها، برنامه‌ریزی خطی[۱٦]، شبکه‌های عصبی مصنوعی[۱۷]، روش‌های فرامکاشفه‌ای، تکنیک‌های ارضای محدودیت[۱۸] و روش‌های مبتنی بر دانش[۱] از این جمله‌اند.

مقاله در تاریخ 22 بهمن ماه 1380 دریافت شد. (شما باید تاریخ ارسال مقاله برای نشریه را در اینجا بنویسید).

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

محمد علی منتظری و سید رسول موسوی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، (email: montazer , srm@cc.iut.ac.ir)

[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
[11] Examination Timtabling Problem
[12] Conflict
[13] Constraints
[14] Feasible
[15] Graph Coloring
[16] Linear Programming
[17] Artificial Neural Network
[18] Constraint Satisfaction

در [3] یک روش برنامه‌ریزی صحیح[2] برای حل UCTP ارائه شده است. در [4] و [5] از یک مدل برنامه‌ریزی صحیح ترکیبی استفاده شده است و یک روش برنامه‌ریزی خطی صحیح[3] را نیز می‌توان در [6] مشاهده کرد.

تبدیل مساله زمانبندی به مساله شناخته شده رنگ‌آمیزی گراف می‌تواند در رسیدن به جواب مسأله کاملا مفید و موثر باشد [7] تا [9]. مساله رنگ‌آمیزی گراف عبارت است از تخصیص تعداد محدودی رنگ به رئوس یک گراف، به طوری که رئوس مجاور همرنگ نباشند[2]. در تبدیل مساله زمانبندی به مساله رنگ‌آمیزی گراف، ایده کار به این صورت است که راس‌های گراف نشان‌دهنده دروس و یال‌های بین رئوس نماینده محدودیت‌ها می‌باشند. تعداد رنگ‌ها نشان‌دهنده تعداد برش‌های زمانی در دسترس می‌باشد. اختصاص یک رنگ به یک راس درست مانند تخصیص یک برش زمانی به یک درس می‌باشد.

در کارهای [11] و [12] از شبکه‌های عصبی برای حل UCTP استفاده شده است. در این کارها ارضای محدودیت‌های مساله را با تابع انرژی سیستم مدل کرده‌اند و همگرا شدن سیستم به حالت پایدار نشان‌دهنده نقاط بهینه محلی می‌باشد. نویسندگان این مقاله‌ها تابع انرژی شبکه‌ی عصبی را طوری فرموله کرده‌اند که شامل یک مجموع وزن‌دار از نقض محدودیت‌های سخت[4] و نرم[5] مساله باشد. کمینه کردن این مجموع به معنی رسیدن شبکه عصبی به یک حالت قابل‌قبول که علاوه بر ارضای محدودیت‌های سخت، تعداد زیادی از محدودیت‌های نرم نیز در آن ارضاء شده است.

بسیاری از کارهای انجام شده در ارتباط با UCTP در سال‌های اخیر از روش‌های فرامکاشفه‌ای و الگوریتم‌های جستجوی محلی استفاده کرده‌اند. در [13] ترکیب روش‌های تپه‌نوردی[6] و جستجوی ممنوعه[7] به‌کار برده شده است. استفاده از روش‌های مکاشفه‌ای برای تولید نقاط شروع و سپس اعمال الگوریتم جستجوی ممنوعه، کار دیگری است که در [14] مشاهده می‌شود. نویسندگان در [15] از الگوریتم سیل بزرگ[8] در حل مساله 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 [16] تاثیر دو نوع همسایگی بر الگوریتم جستجوی ممنوعه در حل UCTP مورد بررسی قرار گرفته است. کارهای مشابه دیگری را می‌توان در [17] و [18] مشاهده نمود.

در مورد UCTP، الگوریتم ژنتیک[9] به شکل‌های متفاوت مورد استفاده قرار گرفته است. به عنوان مثال کار انجام شده در [19] ترکیب الگوریتم ژنتیک با یک روش جستجوی محلی و اعمال آن بر UCTP است. روش‌های دیگری برای زمانبندی با استفاده از الگوریتم‌های تکاملی[10] در [20] و [21] گزارش شده است.

کار انجام شده در [22] و [23] از الگوریتم سیل بزرگ ولی با نرخ کاهش غیرخطی[11] استفاده کرده است. در این کار، الگوریتم جواب‌های بسیار خوبی در مقایسه با سایر روش‌ها ارائه نموده است، ولی عیب این کار اینست که پارامترهای زیادی در الگوریتم باید تنظیم شوند و هر کدام از این پارامترها نسبت به نمونه مساله تحت بررسی دارای مقادیر خاصی هستند. به همین دلیل علی‌رغم جواب‌های خوب این روش، الگوریتم ارائه شده یک الگوریتم کلی نبوده و صرفاً برای نمونه مساله‌های خاصی دارای کارایی بالایی می‌باشد. چنانچه یک نمونه مساله جدید خارج 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 صرفنظر کرده‌ایم.

برخی از الگوریتم‌های بهینه‌سازی که از روش سرد شدن شبیه‌سازی شده بهره می‌برند در [24] تا [26] و [33] پیشنهاد شده است. مثال‌های قابل ذکر دیگری از تکنیک‌های چند هدفه[12] که در جداول زمانبندی امتحانی به کار رفته‌اند را می‌توان در [27] و [34] و به صورت مبسوط‌تر در [28] دید. در کارهایی نظیر [29] و [30]، دو الگوریتم بر پایه الگوریتم اجتماع مورچگان[13] برای حل UCTP ارائه شده است. [31] و [32] و [35] تا [38] نمونه کارهایی هستند که در آنها الگوریتم ممتیک برUCTP اعمال شده است. در [2] تعداد زیادی از روش‌های فرامکاشفه‌ای برای حل UCTP مورد بررسی قرار گرفته است.

در کنار الگوریتم‌های فرامکاشفه‌ای، الگوریتم‌های ابرمکاشفه‌ای[14] نیز در زمینه مساله زمانبندی درسی دانشگاهی مورد بررسی قرار گرفته‌اند [39] تا [41]. الگوریتم‌های ابرمکاشفه‌ای، الگوریتم‌هایی هستند که در فضای الگوریتم‌های مکاشفه‌ای(الگوریتم‌های سطح پایین) جستجو کرده و بهترین ترکیب آنها را برای حل مساله مورد نظر به‌دست 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 3 مساله زمانبندی درسی

---

[1] Knowledge Base Technic
[2] Integer Programming
[3] Integer Linear Programming
[4] Hard Constraint
[5] Soft Constraint
[6] Hill Climbing
[7] Tabu Search
[8] Great Deluge
[9] Genetic Algorithm
[10] Evolutionary Algorithms
[11] Nonlinear decreasing Rate
[12] Multi-Object Technic
[13] Ant Colony Algorithm
[14] Hyper Heuristic Algorithms

دانشگاهی معرفی می‌گردد. روش پیشنهادی برای بررسی مساله زمانبندی درسی دانشگاهی تحت عنوان MA_SA_ECE_UCTP[1] در بخش 4 بررسی می‌شود. در بخش 5 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 انجام شده و ارائه پیشنهاد برای کارهای آینده در بخش 6 مورد بررسی قرار می‌گیرد.

.

## 2- الگوریتم ممتیک

تکامل ژنتیکی تنها شکل تکامل که توسط ریچارد داوکینز[۲] [29] اشاره شده است، نمی‌باشد. تکامل فرهنگی شکل دیگری از تکامل است که نسبت به تکامل ژنتیکی دارای حوزهٔ عمل بزرگتری می‌باشد. در مقایسه با انتقال خصوصیت‌ها در تکامل ژنتیکی، جریان اطلاعات در یک فرآیند تکاملی را انتقال فرهنگی می‌گویند. ژن‌ها به عنوان اجزاء کروموزوم در انتقال صلاحیت‌های شایسته، به عنوان یک واحد حیاتی در انتخاب طبیعی به کار گرفته می‌شوند. نقطه مقابل ژن، در تکامل فرهنگی مِم[۳] نامیده می‌شود که مخفف کلمه یونانی Mimeme می‌باشد [29]. یک مم به عنوان واحد انتقال فرهنگی همانند تقلید[۴] تعبیر می‌شود. مثال‌هایی از مم‌ها عبارتند از: آهنگ صدا، ایده‌ها، اصطلاحات قرضی، مد لباس‌ها، درست کردن قهوه یا ساختن ساختمان‌ها. در حین انتقال اطلاعات توسط مم‌ها ممکن است تغییراتی نیز اتفاق بیافتد که اغلب این تغییرات گسترده هستند. به عنوان 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 اینکه تکامل فرهنگی خیلی سریعتر از تکامل ژنتیکی اتفاق می‌افتد و یک فرآیند قوی با منابع کم می‌باشد. دوم اینکه، در تکامل فرهنگی، به ندرت تغییر، محصول کپی خطاها یا تبادل تصادفی واحدهای همساز اطلاعاتی است. در حوزهٔ علم، ترکیب خام ایده‌ها به بهبود نظریهٔها منجر نمی‌شود. یک دانشمند معمولا یک ایده را بعد از ترکیب با ایده‌های خود و اعتبارسنجی خود می‌پذیرد. فرآیند ترکیب ایده‌ها، به نوآوری منجر می‌-گردد. علت این امر گردآوری دانش و مم‌های جدید از سایر حوزه‌های دیگر است. تکامل زیستی با مولفه نوآوری همراه نیست و اجازه ظهور تجربه در کروموزوم‌های والد را نمی‌دهد. می‌توان گفت که تکامل طبیعی یک فرایند بی‌پایان است در حالی که تکامل فرهنگی یک فرآیند مبتنی بر هدف می‌باشد که در آن واحدهای انتقال اطلاعات به صورت هوشمند عمل می‌کنند.

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

- الگوریتم‌های بهینه‌سازی باید کارآمد باشند. به عنوان مثال، آنها باید قادر به پیدا کردن یک راه‌حل قابل قبول در یک زمان کوتاه باشند.
- الگوریتم‌های بهینه‌سازی بر اساس تعریف، مبتنی بر هدف هستند. نمایش اعضای جمعیت به عنوان 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 می-باشد. در نتیجه نیاز به مولفه‌های دیگری برای تغییر راه‌حل‌ها احساس می‌شود. این مساله می‌تواند با اعمال درجه‌ای از جستجوی همسایگی مرتفع گردد.

نمای کلی الگوریتم ممتیک مانند شبه کد نشان داده شده در **Error! Reference source not found.** می‌باشد.

## 3- مساله زمانبندی درسی دانشگاهی

مساله زمانبندی درسی دانشگاهی یک کار بسیار پیچیده می‌باشد. علت این امر محدودیت‌های آموزشی است که در دانشگاه باید مدنظر گرفته شود. محدودیت‌های مذکور به دو دسته تقسیم‌بندی می شوند [2]:

*محدودیت‌های سخت*: آن دسته از محدودیت‌ها هستند که ارضاء آنها در جدول زمانبندی اجباری است.

*محدودیت‌های نرم*: آن دسته از محدودیت‌ها هستند که اگر ارضاء شوند کیفیت جدول زمانبندی را افزایش می‌دهند، ولی ارضاء آنها اجباری نیست.

محدودیت‌های سخت دارای اولویت بالاتری نسبت به محدودیت‌های نرم هستند و یک جدول زمانبندی، زمانی قابل‌قبول است که همه‌ی محدودیت‌های سخت آن ارضاء شده باشند. به عنوان مثال یک دانشجو 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 ارضاء آنها اجباری نمی باشد. به عنوان مثال اینکه یک دانشجو در سه برش زمانی متوالی در یک روز سه درس نداشته باشد[2].

[1] Memetic Algorithm Simulated Annealing Electical and Computer Engineering_UCTP
[2] Richard Dawkins
[3] Memes
[4] Imitation

نکته دیگری که در مورد UCTP در همه شکل های مختلف آن وجود دارد این است که همه آنها از نظر پیچیدگی محاسباتی[1] به کلاس مسائل NP-hard تعلق دارند و در نتیجه الگوریتمی با پیچیدگی زمانی از مرتبه چندجمله‌ای[2] برای حل آنها ارائه نشده است[2]. حل مسائل زمانبندی واقعی با استفاده از الگوریتم‌های دقیق[3] غیر عملی می‌باشد چرا که زمان اجرای این دسته از الگوریتم‌ها با رشد اندازه مساله به صورت نمایی افزایش می‌یابد. بنابراین می‌توان از الگوریتم‌های تقریبی[4] مانند الگوریتم-های مکاشفه‌ای و فرامکاشفه‌ای برای حل آنها استفاده نمود.

در این مقاله دو نمونه مساله زمانبندی درسی دانشگاهی را مورد بررسی قرار خواهیم داد. اولین مورد مساله UCTP مطرح شده در ITC[5] توسط Ben Paecher و دومین مورد مساله زمانبندی درسی در دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان می‌باشد.

در مورد اول یک UCTP را می‌توان در حالت کلی به صورت زیر بیان کرد[20]: یک مجموعه از دروس (E) که باید در 45 برش زمانی (5 روز و هر روز شامل 9 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 ارضاء می‌شوند. هر دانشجو تعدادی از این دروس را اخذ می‌کند و هر اتاق دارای اندازه‌ای ثابت می‌باشد. یک جدول زمانبندی را قابل‌قبول می‌گویند اگر به هر درس یک برش زمانی و یک اتاق اختصاص یابد و محدودیت‌های سخت نیز در جدول ارضاء شوند[2]. 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 ارضاء کند. ج) در یک برش زمانی فقط یک درس در یک کلاس برگزار شود. محدودیت‌های نرم عبارتند از : الف) یک دانشجو در آخرین برش زمانی در روز کلاس نداشته باشد. ب) یک دانشجو بیش از دو درس در دو برش زمانی پشت سر هم نداشته باشد. ج) یک دانشجو، در یک روز تنها یک کلاس نداشته باشد. هدف این مساله کاهش تعداد نقض‌های[6] محدودیت نرم در یک راه حل عملی (قابل‌قبول) است. تمامی راه‌حل-های غیرعملی فاقد ارزش می باشند.

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

– دروس مربوط به دانشجویان سال آخر دوره کارشناسی در روز چهارشنبه زمانبندی نشوند.

– یک درس در اتاقی زمانبندی شود که ظرفیت آن خیلی بزرگتر از تعداد دانشجویان آن درس نباشد. به عنوان مثال اگر دو اتاق با ظرفیت 50 و 80 نفر موجود باشند و هر دو برای درس x با تعداد دانشجویان 45 نفر مناسب باشند؛ آنگاه اتاق با ظرفیت 50 نفر انتخاب شود.

– و محدودیت‌های سخت عبارتند از:

– عدم تشکیل دو درس به صورت همزمان(در یک برش زمانی) در یک کلاس.

– عدم وجود تداخل(دروس همزمان) برای دانشجویان.

– عدم وجود تداخل(دروس همزمان) برای اساتید.

– زمانبندی دروس در کلاس‌های مناسب(کلاس‌هایی که دارای امکانات مورد نیاز برای درس هستند).

– زمانبندی دروس در زمان‌هایی که از قبل برای آن درس مشخص شده است. به عنوان مثال درس x باید در روز شنبه ساعت 8 تا 9 زمانبندی شود.

– زمانبندی دروس در زمان‌هایی که اساتید مربوط به آن دروس در دانشکده حضور دارند. به عنوان مثال استاد x فقط روزهای شنبه و دوشنبه تدریس می‌کنند و یا استاد y روزهای یکشنبه صبح تدریس نمی-کند.

– چنانچه درسی در یک کلاس در زمان خاصی زمانبندی شود، آن کلاس در آن زمان در اختیار دانشکده باشد. به عنوان مثال کلاس x روز دوشنبه بعدازظهر در اختیار دانشکده دیگری است.

– عدم زمانبندی دروس در روز پنج‌شنبه.

از جمله مزیت‌های کار انجام شده در این مقاله استفاده از تکنیک‌های مکاشفه‌ای و روش‌های جستجوی فرامکاشفه‌ای در بررسی UCTP می-باشد. تفاوت عمده این کار با کارهای انجام شده، انعطاف‌پذیری آن برای بررسی داده‌های واقعی و استفاده از آن در محیط واقعی دانشکده می‌باشد. بالا بودن دقت برنامه، سرعت، اعمال سیاست‌های آموزشی دانشکده در برنامه‌ریزی، همسو بودن برنامه آموزشی با استفاده بهینه از وقت اساتید و دانشجویان از جمله مزایای این برنامه می‌باشد.

## 4- روش پیشنهادی (MA_SA_ECE_UCTP)

در الگوریتم MA_SA_ECE_UCTP از ترکیب الگوریتم ممتیک(MA) و روش جستجوی سردشدن شبیه‌سازی شده(SA) استفاده شده است. نحوه عملکرد MA_SA_ECE_UCTP مطابق شکل 1 به شرح ذیل می‌باشد: برای هر کروموزوم، ابتدا با استفاده از رویه مقداردهی اولیه مکاشفه‌ای، کروموزوم‌ها مقداردهی اولیه می‌شوند. سپس رویه جستجوی محلی مبتنی بر SA جهت بهبود کروموزوم و تبدیل آن به یک راه‌حل بهینه محلی[7]، فراخوانی می‌شود. بعد از این مرحله عملگر بهبود بر

[1] Complexity of a Computation
[2] Polynomially Bounded Algorithm
[3] Exact Algorithms
[4] Approximation Algorithms
[5] International Timetabling Competition
[6] Violations
[7] Local Optima Solution

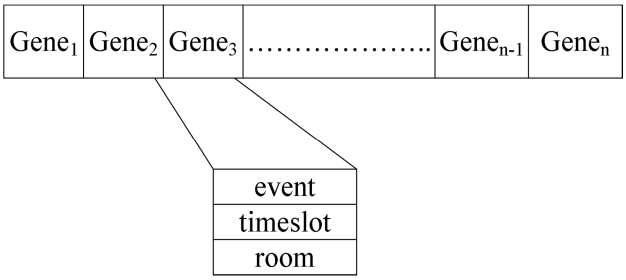
روی کروموزوم اعمال شده و موجب بهبود هر چه بیشتر آن می‌گردد. در حلقه while بعد از انتخاب دو کروموزوم، متناسب با میزان برازندگی[1] آنها از بین جمعیت، و اعمال عملگر تقاطع[2] با احتمال pC و عملگر جهش[3] با احتمال pM یک فرزند[4] تولید می‌شود. سپس رویه SA و عملگر بهبود به ترتیب فراخوانی شده و تعداد نقض محدودیت‌های سخت و نرم موجود در فرزند را بهبود می‌بخشند. در نهایت این فرزند با بدترین عضو جمعیت مقایسه شده و در صورت بهتر بودن جایگزین آن می‌شود. این رویه تا رسیدن به شروط توقف ادامه می‌یابد.

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

شکل1: شبه کد الگوریتم MA_SA_ECE_UCTP

## 1-4 نحوه نمایش راه‌حل با استفاده از کروموزوم

هر کروموزوم شامل مجموعه‌ای از ژنها است که نشان دهنده یک راه-حل کاندید می‌باشد. نمایش استاندارد برای یک جدول زمانبندی به صورت یک ماتریس دو بعدی است که در آن سطرها نشان‌دهنده روزهای هفته و ستون‌ها نمایانگر برش‌های زمانی در روز می‌باشد. 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 می‌شود. این آرایه را یک کروموزوم می‌نامیم. هر عنصر آرایه یک ژن و هر ژن از یک سه‌تایی <event, timeslot, room> تشکیل شده است. تعداد ژن‌ها در هر کروموزوم برابر با تعداد دروس می‌باشد. علت استفاده از این نوع نمایش ساده بودن و سادگی اعمال عملگرهای الگوریتم ممتیک بر روی آن می‌باشد. ساختار هر کروموزوم در شکل 2 نمایش داده شده است.




شکل2: نحوه نمایش کروموزوم‌ها

## 2-4 رویه مقداردهی اولیه مکاشفه‌ای

از آنجا که UCTP بسیار شبیه به مساله رنگ‌آمیزی گراف می-باشد[2]، می‌توان برای ساخت کروموزوم‌ها از رویه‌های مکاشفه‌ای استفاده شده در مساله رنگ‌آمیزی گراف استفاده نمود[36]. اگر هر درس را به عنوان یک راس در نظر بگیریم و چون دروسی که دانشجو و یا استاد مشترک دارند، نمی‌توانند در یک برش‌زمانی یکسان قرار گیرند، پس بین چنین جفت راسی یک یال(محدودیت) اضافه می‌کنیم، آنگاه می توان تعداد زیادی از مسائل زمانبندی ساده را به مسأله رنگ‌آمیزی گراف (و برعکس) تبدیل نمود[2]. هر برش‌زمانی را می‌توان به عنوان یک رنگ فرض نمود. جدول 1 مکاشفه‌های استفاده شده در مقداردهی اولیه را نشان می‌دهد. این مرحله با یک جدول زمانبندی 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 1 هر بار یک درس و یک محل مناسب انتخاب شده و درس به آن محل تخصیص می‌یابد.

جدول1: رویه‌های مکاشفه‌ای برای ساختن کروموزوم‌ها

| | |
|---|---|
| $h_1$ | انتخاب رویداد با تعداد محل مناسب کمتر در جدول زمانبندی 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. |

## 3-4 رویه جستجوی محلی

هدف از رویه جستجوی محلی در الگوریتم MA تبدیل یک راه‌حل به یک راه‌حل بهینه محلی می‌باشد. در این رویه با استفاده از ساختارهای همسایگی[5] که در ادامه خواهند آمد، یک کروموزوم بهبود داده می‌شود. منظور از بهبود یک کروموزوم کم کردن تعداد نقض محدودیت‌ها در آن می‌باشد. ساختارهای همسایگی به مجموعه تغییراتی گفته می‌شود که بر

[1] Fitness
[2] Crossover
[3] Mutation
[4] Offspring

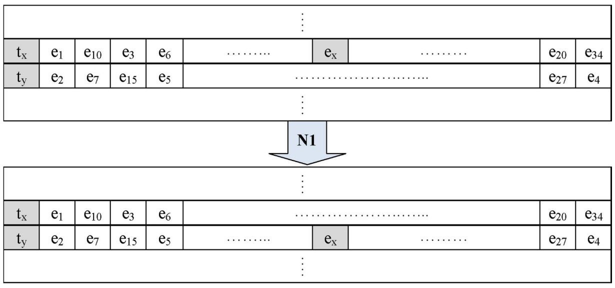
[5] Neighbor Structures

روی یک راه‌حل انجام شده و موجب به‌وجود آمدن راه‌حل‌های 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 راه‌حل‌های همسایه بدست آمده برای کاوش بهتر فضای جستجو می‌باشد. به عنوان مثال زمانی که یکی از ساختارهای همسایگی در ارائه راه‌حل‌های همسایه بهتر نسبت به راه‌حل جاری، ناتوان باشد، می‌توان از سایر ساختارهای همسایگی بهره برد. ساختارهای همسایگی مورد استفاده در این مقاله عبارتند از:

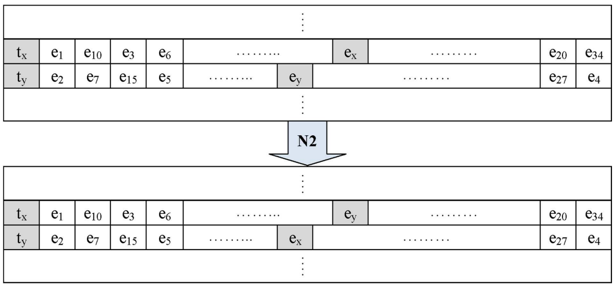
ساختار همسایگی N1: در این ساختار که ساده‌ترین نوع ساختار همسایگی مورد استفاده در مقاله می‌باشد، یک درس مانند $e_x$ که به برش زمانی $t_x$ تخصیص یافته است، به صورت تصادفی انتخاب شده و به یک برش‌زمانی تصادفی جدید مانند $t_y$ انتقال داده می‌شود. نتیجه این حرکت رسیدن به یک جدول زمانبندی جدید می‌باشد که در آن برش زمانی تخصیص داده شده به درس $e_x$ با برش‌زمانی تخصیص داده شده به همان درس در جدول زمانبندی قبلی متفاوت می‌باشد.

ساختار همسایگی N2: 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 یافته‌اند) با یکدیگر تعویض می‌شود.

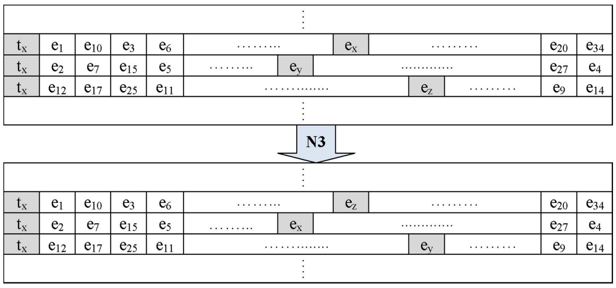
ساختار همسایگی N3: در این ساختار سه درس $e_x$ ، $e_y$ و $e_z$ به صورت تصادفی انتخاب شده و برش‌های زمانی آنها(برش‌های زمانی که به این سه درس اختصاص یافته‌اند) به صورت چرخه‌ای با یکدیگر تعویض می‌شوند. در شکل های 3، 4 و 5 نحوه عمل هر یک از ساختارهای همسایگی نشان داده شده است.




شکل3: نحوه عملکرد ساختارهمسایگی N1. رویداد $e_x$ از برش‌زمانی $t_x$ به برش‌زمانی $t_y$ انتقال یافته است.



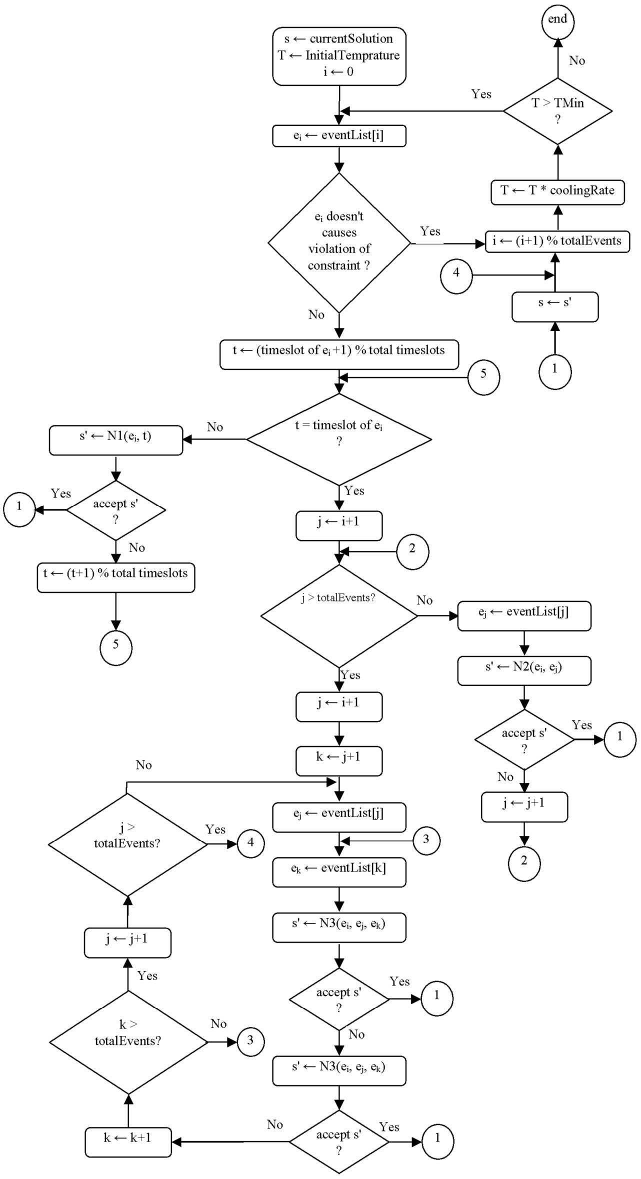

شکل 4: نحوه عملکرد ساختارهمسایگی N2. رویداد $e_x$ از برش‌زمانی $t_x$ به برش‌زمانی $t_y$ و رویداد $e_y$ از برش‌زمانی $t_y$ به برش‌زمانی $t_x$ انتقال یافته است.




شکل 5: نحوه عملکرد ساختارهمسایگی N3. رویداد $e_x$ از برش‌زمانی $t_x$ به برش‌زمانی $t_y$ و رویداد $e_y$ از برش‌زمانی $t_y$ به برش‌زمانی $t_z$ و و رویداد $e_z$ از برش‌زمانی $t_z$ به برش‌زمانی $t_x$ انتقال یافته است.

همانطور که در شکل 6 نشان داده شده است، نحوه عملکرد رویه جستجوی محلی به شرح زیر می‌باشد. راه‌حل 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 سردشدن 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 برش‌زمانی و اتاق خاصی که در s برای این درس مشخص شده بررسی می‌گردد. در صورت عدم وجود اینچنین نقضی، درس بعدی از لیست دروس انتخاب شده و این رویه تکرار می‌شود. چنانچه درس مذکور باعث ایجاد نقض محدودیت شده باشد، باید برش‌زمانی منتسب به این درس را تغییر دهیم. برای این کار ابتدا از ساختار همسایگی N1 استفاده می‌کنیم. در صورت کاهش تعداد نقض محدودیت‌ها، تغییر مذکور را پذیرفته و در غیر اینصورت از ساختار همسایگی N2 استفاده می‌شود. چنانچه این ساختار نیز منجر به بهبود تعداد نقض محدودیت‌ها نشود، انتخاب بعدی استفاده از ساختار همسایگی N3 می‌باشد.

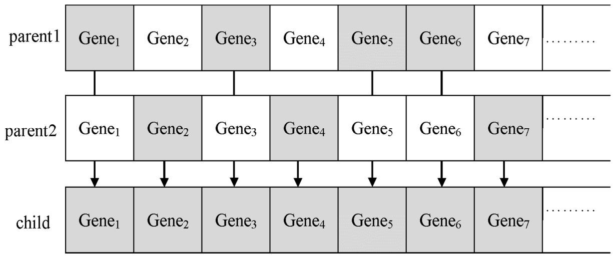
این مراحل برای همه دروس چندین بار تکرار شده و در هر مرحله دما کاهش می‌یابد تا به دمای مینیمم که از قبل توسط کاربر مشخص شده برسد.




شکل 6: دیاگرام نحوه عملکرد رویه جستجوی محلی مبتنی بر سرد شدن شبیه سازی شده

## 4-4 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 می‌شود: ابتدا یک درس انتخاب شده و تعداد نقض محدودیت ایجاد شده 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 7 نشان داده شده است. به عنوان مثال در والد 1 ژن‌های 1، 3، 5 و 6 ژن‌های مرتبط با یکدیگر می‌باشند و ژن 1 از والد 1 انتخاب شده که همراه با ژن‌های مرتبط خود به فرزند انتقال یافته‌اند.




شکل 7: عملگر تقاطع، انتخاب ژن 1 از والد اول و انتقال به همراه ژن‌های مرتبط به فرزند. همین امر برای ژن 2 از والد دوم اتفاق افتاده است.

## 5-4 عملگر بهبود کروموزوم(improvement)

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 جابجا 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 $y_1$ و $y_2$ می‌باشد. در ابتدا به صورت تصادفی دانشجوی $x$ در یکی از گروه‌های درسی قرار می‌گیرد. به عنوان مثال گروه درسی $y_1$ 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 $v_1$ می‌نامد. سپس تعداد نقض محدودیت نرم را زمانی که گروه $y_2$ برای دانشجوی $x$ ثبت شود، محاسبه کرده و آن را $v_2$ نامیده، و اگر $v_2<v_1$ باشد دانشجوی $x$ از لیست دانشجویان گروه درسی $y_1$ حذف شده و به لیست گروه درسی $y_2$ اضافه می‌شود. در غیر اینصورت عملی اتفاق نخواهد افتاد.

## 5- نتایج تجربی

روش MA_SA_ECE_UCTP با زبان برنامه‌نویسی جاوا کد شده و بر روی یک کامپیوتر شخصی با 512 mb حافظه و پردازنده Intel 2.0 GHz ارزیابی شده است. پارامترهای استفاده شده برای این روش در جدول 2 نمایش داده شده‌اند. MA_SA_ECE_UCTP بر روی داده‌های ارائه شده در [8] اعمال شده است. هر چند این داده‌ها همه‌ی محدودیت‌ها و جنبه‌های یک مساله واقعی را نشان نمی‌دهند[9] ولی معیار مناسبی برای مقایسه روش پیشنهادی ما با سایر روش‌های جدید که روی این داده‌ها تست شده‌اند، می‌باشد.

در جدول 3، MA_SA_ECE_UCTP با برخی روش‌های جدید انجام شده‌ی دیگر مقایسه شده است. ستون اول از سمت چپ نمونه‌های مساله را نشان می‌دهد. در این ستون حرف s[1]، m[2] و l[3] به ترتیب نشان

[1] Small
[2] Medium
[3] Large

دهنده نمونه‌های کوچک، متوسط و بزرگ می‌باشند. ستون بعدی نتایج MA_SA_ECE_UCTP را روی نمونه‌های متفاوت نشان می‌دهد. عددهای موجود در هر ستون نشان دهنده 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 راه‌حل قابل-قبول در یک زمان معقول(در روش‌های مختلف زمان متفاوت در نظر گرفته شده است. از 3 ساعت تا 10 ساعت) نبوده است از عبارت Inf [1] برای آن استفاده شده است.

جدول 2: پارامترهای استفاده شده MA_SA_UCTP

| پارامتر | مقدار |
|---|---|
| تعداد نسل | 100 |
| اندازه جمعیت | 40 |
| احتمال تقاطع | 0/8 |
| احتمال جهش | 0/07 |
| دمای اولیه | 2 |
| دمای نهایی | 0/005 |
| نرخ کاهش دما | 0/999 |

زمان اجرای الگوریتم پیشنهادی برای نمونه‌های کوچک، متوسط و بزرگ به ترتیب 60، 300 و 600 دقیقه می‌باشد. نتایج به دست آمده از اجرای این روش نشان می‌دهند که MA_SA_ECE_UCTP در همه موارد قادر به پیدا کردن راه‌حل قابل‌قبول می‌باشد.

جدول 3: مقایسه MA_SA_UCTP با برخی روش‌های موجود. inf به معنای جواب غیر قابل‌قبول می‌باشد.

| نمونه‌ها | MA_SA_ECE_UCTP | [5] | [6] | [10] | [13] | [22] | [37] | [38] |
|---|---|---|---|---|---|---|---|---|
| *s1* | **0** | **0** | **0** | **0** | **0** | 2 | 5 | **0** |
| *s2* | **0** | **0** | **0** | **0** | **0** | 4 | 3 | **0** |
| *s3* | **0** | **0** | **0** | **0** | **0** | 2 | 2 | **0** |
| *s4* | **0** | **0** | **0** | **0** | **0** | **0** | 3 | **0** |
| *s5* | **0** | **0** | **0** | **0** | **0** | 4 | 1 | **0** |
| *m1* | **144** | 317 | 242 | 221 | 227 | 245 | 316 | 236 |
| *m2* | **142** | 313 | 161 | 147 | 180 | 285 | 243 | 158 |
| *m3* | **207** | 357 | 265 | 246 | 235 | 251 | 255 | 261 |
| *m4* | **126** | 247 | 181 | 165 | 142 | 321 | 235 | 176 |
| *m5* | 169 | 292 | 151 | **130** | 200 | 276 | 215 | 147 |
| *l* | 942 | Inf | Inf | 529 | Inf | 1027 | Inf | **296** |

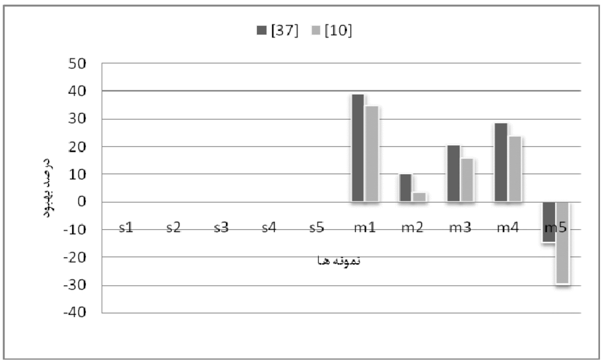
در نمودار شکل 8 درصد بهبود روش پیشنهادی نسبت به دو روش [10] و [38] نشان داده شده است. علت انتخاب این دو روش بهتر بودن آنها نسبت به سایر روش‌ها می‌باشد. همانطور که در نمودار دیده می‌شود، روش ما در همه نمونه‌ها بجزء m5 از دو روش دیگر بهتر بوده

[1] Infeasible

است(درصد مثبت). در نمونه m5 درصد منفی شده است و به معنای بدتر بودن روش ما نسبت به دو روش دیگر است. اگر راه‌حل به دست آمده از MA_SA_ECE_UCTP و روش قابل مقایسه، در یک نمونه مساله، به ترتیب s و s′ باشد و $f(s)$ و $f(s')$ مقدار برازندگی راه‌حل‌ها در دو روش مذکور باشد، آنگاه درصد بهبود به صورت زیر محاسبه می‌شود:

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

بیشترین بهبود در نمونه m1 دیده می‌شود. میزان بهبود در نمونه m1 نسبت به روش [10] برابر 34/8 و نسبت به روش [38] برابر 38/9 می‌باشد.




شکل 8: مقایسه MA_SA_ECE_UCTP با روش‌های [10] و [38]. محور افقی نمونه‌های تست و محور عمودی درصد بهبود را نشان می‌دهند.

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 89-88 می‌باشد. این داده‌ها بر اساس اطلاعات آماری ثبت‌نام 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 : 976 نفر. این تعداد شامل دانشجویان کارشناسی رشته تحصیلی برق(کلیه گرایش‌ها) و دانشجویان رشته تحصیلی کامپیوتر(کلیه گرایش‌ها) می‌باشد.

تعداد اساتید : 55 نفر که از این تعداد 43 نفر اعضای هیات علمی و 12 نفر شامل اساتید مدعو و دانشجویان دوره‌ی دکتری که درس ارائه می‌کنند.

تعداد کلاس‌ها : 14 اتاق که 4 کلاس برای دروس ارشد، 2 کلاس مشترک کارشناسی و ارشد و بقیه برای دوس کارشناسی می‌باشند.

تعداد دروس نیم‌سال اول 89-88 : این تعداد شامل 77 درس می-باشد که 56 درس تئوری و بقیه دروس آزمایشگاه هستند. لازم به ذکر است که برخی از دروس در چند گروه تشکیل می‌گردند(2 یا 3 گروه). به عبارتی دیگر تعداد گروه‌های نظری 84 گروه و تعداد گروه‌های عملی 77 گروه می‌باشند.

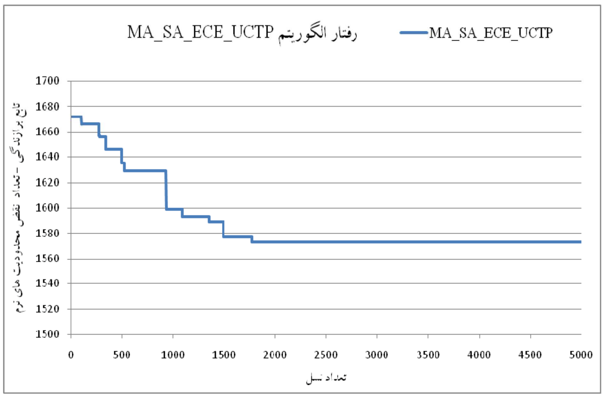
پارامترهای استفاده شده در این روش همان پارامترهای ذکر شده در

جدول 2 می‌باشد.

همانطور که در بخش 4 اشاره شد این الگوریتم ترکیبی از الگوریتم ممتیک و جستجوی محلی مبتنی بر الگوریتم سردشدن شبیه‌سازی شده می‌باشد. در این الگوریتم به جای رویه مقدار دهی اولیه کروموزوم‌ها به صورت تصادفی، از رویه مقداردهی اولیه مکاشفه‌ای استفاده شده است. همچنین لازم به یادآوری است که عملگر تقاطع استفاده شده در این الگوریتم عملگر تقاطع حریصانه می‌باشد. در این الگوریتم از رویه بهبود برای بالا بردن هر چه بیشتر کیفیت راه‌حل‌های 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 2500 در نظر گرفته شده است، به مقدار تابع برازندگی آن دسته از راه‌حل‌های دارای نقض محدودیت سخت اضافه می‌کنیم.

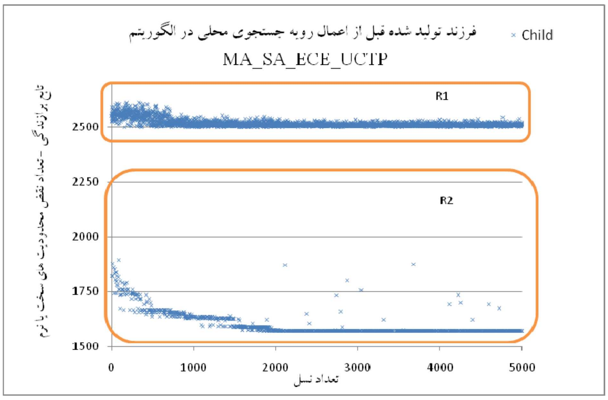
رفتار کلی الگوریتم بر اساس تابع برازندگی بهترین عضو جمعیت آن به صورت نمودار شکل 9 می‌باشد. در این نمودار محور عمودی نشان دهنده تعداد نقض محدودیت‌ها(مقدار تابع برازندگی) و محور افقی تعداد نسل را نشان می‌دهد. همانطور که در نمودار دیده می‌شود وجود رویه مقداردهی اولیه مکاشفه‌ای باعث ایجاد راه‌حل‌های اولیه قابل‌قبول شده است( مقدار تابع برازندگی آنها کمتر از عدد 2500 می‌باشد.). در نسل‌های تقریبا کوچکتر از 2000 مقدار تابع برازندگی سیر نزولی داشته و همواره راه‌حل به‌دست آمده بهبود داده شده است. در نسل‌های بالاتر از 2000 هیچ‌گونه بهبودی در راه‌حل جواب دیده نمی‌شود. این امر نشان دهنده‌ی همگرایی در جمعیت و عدم توانایی الگوریتم در بهبود بیشتر راه‌حل می‌باشد.




شکل 9: رفتار الگوریتم MA_SA_ECE_UCTP بر اساس تابع برازندگی بهترین عضو جمعیت در طول 5000 نسل

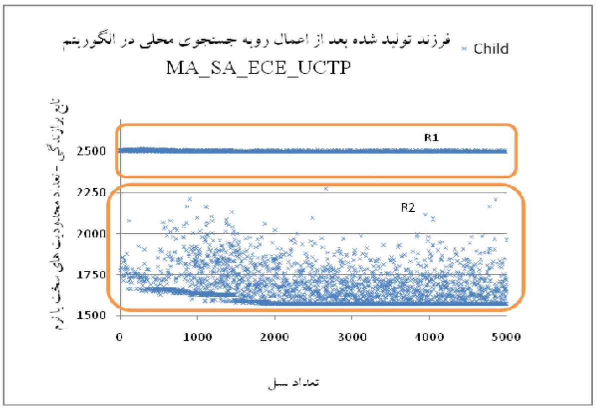
در نمودار شکل‌های 10 و 11 رفتار فرزند تولید شده در این الگوریتم نشان داده شده است. همین مساله در شکل 10 و 11 باعث به وجود آمدن دو ناحیه R1 و R2 شده است. ناحیه R1 نشان‌دهنده راه‌حل‌هایی است که دارای نقض محدودیت سخت می‌باشند و ناحیه R2 راه‌حل‌هایی را نشان می‌دهدکه قابل قبول بوده و دارای نقض محدودیت نرم می‌باشند.

نمودار شکل 10 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 R1 (دارای نقض محدودیت سخت) می‌باشند. در شکل 11 مجددا مقدار تابع برازندگی فرزند تولید شده در طول نسل‌ها بعد از اعمال رویه جستجوی محلی نمایش داده شده است.




شکل 10: تابع برازندگی فرزند تولید شده در الگوریتم MA_SA_ECE_UCTP قبل از اعمال رویه جستجوی محلی. ناحیه R1 فرزندان غیرقابل‌قبول(دارای نقض محدودیت سخت) و ناحیه R2 فرزندان قابل‌قبول(عدم نقض محدودیت سخت)

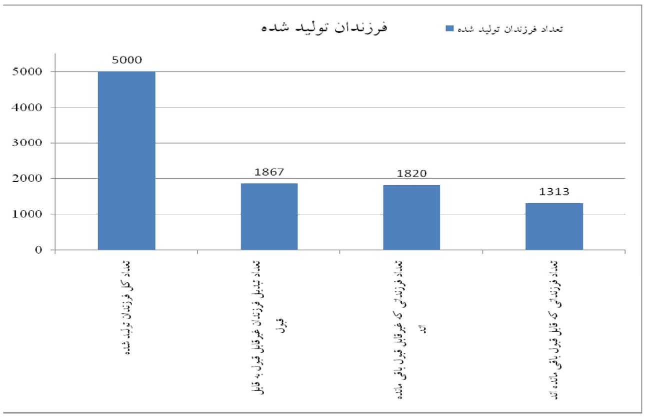
با توجه به شکل 11 مشاهده می‌شود که تعداد زیادی از فرزندان تولید شده که در ناحیه R1 بودند(شکل 10) بعد از اعمال رویه جستجوی محلی به ناحیه R2 منتقل شده‌اند.




شکل 11: تابع برازندگی فرزند تولید شده در الگوریتم MA_SA_ECE_UCTP بعد از اعمال رویه جستجوی محلی. ناحیه R1 فرزندان غیرقابل‌قبول و ناحیه R2 فرزندان قابل‌قبول

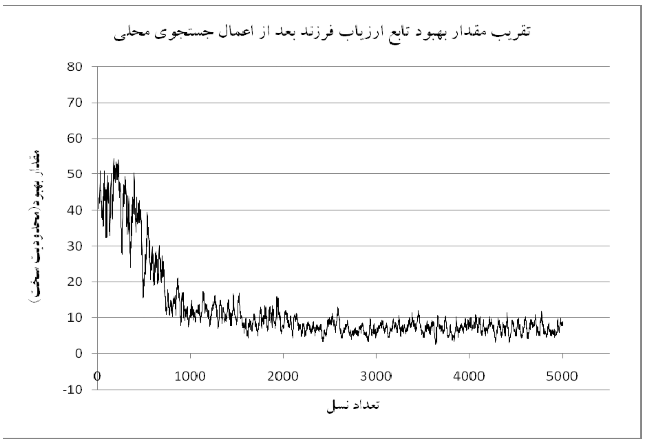
در نمودار شکل 12 تاثیر این رویه بر تابع ارزیاب فرزندان تولید شده به صورت آماری نشان داده شده است.

در نمودار شکل 12 ستون اول از سمت چپ تعداد کل فرزندان تولید شده را در طی نسل‌ها نشان می‌دهد و ستون دوم از سمت چپ تعداد فرزندانی را نشان می‌دهد که بعد از اعمال رویه جستجوی محلی از حالت غیرقابل‌قبول به حالت قابل‌قبول 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 قابل‌قبول 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 قابل‌قبول بوده‌اند.




شکل 12: تعداد فرزندان در حالت‌های متفاوت بعد از اعمال رویه جستجوی محلی در الگوریتم MA_SA_ECE_UCTP

در نمودار شکل 13 تقریب مقدار بهبود تابع ارزیاب فرزند بعد از اعمال جستجوی محلی در طول همه‌ی 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 10 رسیده است. علت این امر این است که با افزایش تعداد نسل، تعداد فرزندانی که نقض محدودیت سخت در آنها اتفاق می‌افتد کاهش می‌یابد و در نتیجه تاثیر رویه جستجوی محلی بر آنها چشمگیر نخواهد بود.




شکل 13: تقریب مقدار بهبود تابع ارزیاب فرزند بعد از اعمال جستجوی محلی در الگوریتم MA_SA_ECE_UCTP

با توجه به شکل 13 تاثیر 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 تقریبا از نسل‌های 2000 به بعد یکنواخت شده و تا پایان اجرا تغییر زیادی نداشته است.

در نمودار شکل 14 میانگین تعداد دفعات استفاده از ساختارهای همسایگی در هر بار اعمال جستجوی محلی بر فرزند تولید شده در هر نسل، نشان داده شده است.

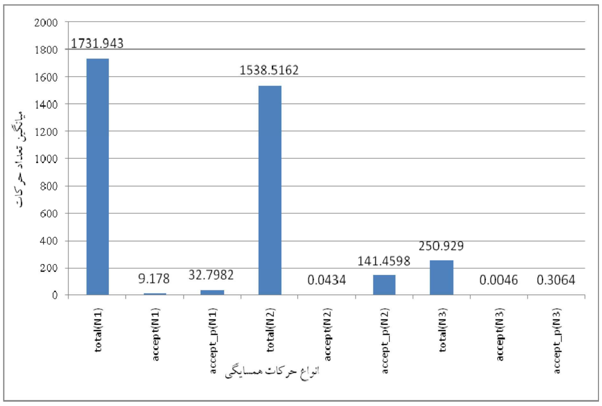
در این نمودار محور عمودی نمایانگر میانگین تعداد دفعات استفاده از ساختارهای همسایگی می‌باشد و عناصر محور افقی به شرح ذیل می‌باشند:

total(N1) : میانگین تعداد کل حرکات انجام شده که منجر به ساختار همسایگی N1 شده‌اند.

accept(N1) : میانگین تعداد استفاده از ساختار همسایگی نوع N1 که منجر به راه‌حل‌های جدید بهتر از راه‌حل جاری شده‌اند. این راه‌حل‌های جدید همگی پذیرفته می‌شوند.

accept_p(N1) : میانگین تعداد استفاده از ساختار همسایگی نوع N1 که منجر به راه‌حل‌های بدتر نسبت به راه‌حل جاری شده‌اند ولی با توجه به شرط پذیرش در الگوریتم سردشدن شبیه‌سازی شده با احتمال p، پذیرفته شده‌اند.

برای ساختارهای همسایگی N2 و N3 نیز ستون‌هایی به ترتیب بالا رسم شده است. در جدول 4 اعداد موجود در نمودار بالا به صورت درصد نشان داده شده است.




شکل 14: میانگین تعداد دفعات استفاده شده از ساختارهای همسایگی N1، N2 و N3 در اعمال رویه جستجوی محلی بر فرزند تولید شده در هر نسل.

همانطور که ملاحظه می‌شود بیشترین تعداد استفاده از ساختار همسایگی N1 و کمترین تعداد مربوط به ساختار همسایگی N3 می‌باشد. همچنین تاثیر ساختار همسایگی N3 با توجه به جدول 4 در پذیرش راه‌حل‌های بهتر بسیار کم بوده و اعمال یک ساختار همسایگی جدید به جای آن می‌تواند پیشنهاد شود.

عددهای موجود در این جدول مربوط به میانگین حرکات در جستجوی محلی برای یک فرزند می‌باشد. همانطور که در جدول دیده می‌شود 49/1835 درصد از کل حرکات منجر به ساختار همسایگی N1 شده‌اند که از این تعداد 0/5299 درصد ساختارهایی بوده‌اند که راه‌حل‌های جدید بهتر تولید کرده‌اند و از میان راه‌حل‌های تولید شده بدتر 1/8937 درصد پذیرفته شده‌اند. برای ساختارهای همسایگی N2 و N3 نیز ترتیب عددهای موجود در جدول به همین صورت می‌باشد.

جدول 4: میانگین تعداد دفعات استفاده شده از ساختارهای N1، N2 و N3 در اعمال رویه جستجوی محلی بر فرزند تولید شده در یک نسل که به صورت درصد نشان داده شده است.

| | total | accept | accept_p |
|---|---|---|---|
| **Total(moves)** | 3521/3882 | 9/226 | 174/5644 |
| **N1** | 49/1835% | 0/5299% | 1/8937% |
| **N2** | 43/6906% | 0/0028% | 9/1946% |
| **N3** | 7/1259% | 0/0018% | 0/1221% |

## 6- 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 مقدار دهی اولیه مکاشفه‌ای در ارضاء محدودیت‌های سخت و قابل‌قبول شدن راه‌حل-های اولیه بسیار موثر بود. همانگونه که مشاهده گردید الگوریتم ارائه شده در مقایسه با تعدادی از کارهای جدید انجام شده در زمینه مساله زمانبندی دزسی دانشگاهی در اکثر نمونه‌های تست بسیار بهتر می‌باشد. یکی دیگر از مزایای الگوریتم ارائه شده کاربردی بودن آن می‌باشد به نحوی که MA_SA_ECE_UCTP بر داده‌های حقیقی دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان نیز اعمال شده است. نتایج به دست آمده نشان می‌دهند که این کار با کار انجام شده به صورت 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 استثناء را برطرف نماییم، می‌توان از این الگوریتم استفاده نمود. تاثیر روال جستجوی محلی در تبدیل راه‌حل‌های کاندید به راه‌حل‌های قابل‌قبول 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 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 دراین مقاله) و یا تغییر عملگرهای استاندارد آن بهره برد. اعمال رویه‌های مکاشفه‌ای در ارضاء محدودیت‌های نرم نیز از جمله کارهایی است که می‌توان در حل این مساله 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 11 مشاهده شد، الگوریتم از نسل تقریبا 2000 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 بکار برد. به عنوان مثال ساختار همسایگی N3 با توجه به جدول 4 در بهبود جمعیت تولید شده خیلی موثر نبوده و می‌تواند با یک ساختار همسایگی مطلوبتر جایگزین شود. علاوه بر استفاده از انواع ساختارهای همسایگی متفاوت، نحوه انتخاب نقاط همسایه به طور موثر 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 موثر نیز از جمله پیشنهادهایی می‌باشند که می‌توان در کارهای آینده از آنها استفاده نمود.

## 7- قدردانی

نویسندگان این مقاله بدینوسیله از سرکار خانم منصف کارشناس محترم آموزش دانشکده برق، کامپیوتر و فناوری اطلاعات دانشگاه صنعتی اصفهان بخاطر همکاری بی دریغ در راستای انجام پروژه مربوطه قدردانی می نمایند.

## 8- مراجع

[1] E. Burke, J. Kingston, K. Jackson, and R. Weare. “Automated University Timetabling: The State of the Art” *The Computer Journal*, Vol. 40, No. 9, pp. 565-571, 1997.

[2] R Lewis. “A survey of metaheuristic-based techniques for university timetabling problems”, *OR Spectrum*, Vol. 30, No. 1, pp. 167-190, 2008.

[3] D. Costa. “A tabu search algorithm for computing an operational timetable” *Eur J Oper Res*, Vol. 79, pp. 98–110, 1994.

[4] L. Gislen, C. Peterson, and B. Soderberg. “Teachers and classes with neural networks” *Int J Neural Syst*, Vol. 1, pp. 167–176, 1989.

[5] D. Corne, P. Ross, and H. Fang. “Evolving timetables” In: Lance C. Chambers (ed) *The practical handbook of genetic algorithms*, Vol. 1, CRC, Florida, pp. 219–276, 1995.

[6] M. Carrasco, and M. Pato. “A multiobjective genetic algorithm for the class/teacher timetabling problem,” *In: Burke E, Erben W (eds) Practice and theory of automated timetabling (PATAT) III*, Vol .2079, Springer, Berlin, pp. 3–17, 2001.

[7] D. Abramson, H. Krishnamoorthy, and H. Dang. “Simulated annealing cooling schedules for the school timetabling problem” *Asia Pacific J Operational Res*, Vol. 16, pp. 1–22, 1996.

[8] E. Burke, Y. Bykov, and M. Petrovic. “A multicriteria approach to examination timetabling,” *In: Burke E, Erben E (eds) practice and theory of automated timetabling (PATAT) III*, Vol. 2070, Springer, Berlin, pp. 118–131, 2001.

[9] A. Alkan and E. Ozcan.” Memetic algorithms for timetabling evolutionary computation,” *Proc of the 2003 IEEE Congress on Evol Comput*., Vol. 3, pp. 1796–1802, 2003.

[10] S. Abdullah, E. K. Burke, and B. Mccollum. “A hybrid evolutionary approach to the university course timetabling problem,” *In Proceedings of the IEEE Congress on Evolutionary Computation*, Singapore, September 2007.

[11] P. Moscato. “On evolution, search, optimization, genetic algorithms and martial art: towards memetic algorithms,” caltech concurrent computation program, Technical Report, 1989.

[12] P. Moscato. “Memetic algorithms: A short introduction,” *In: New Ideas in Optimisation, Mcgraw-Hill’S Advanced Topics In Computer Science Series*, pp. 219–234.

[13] O. Rossi-Doria, B. Paechter. “A memetic algorithm for university course timetabling” in: *Combinatorial Optimisation 2004 Book of Abstracts*, Lancaster, UK, Lancaster University, 2004.

[14] J. Holland, *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: University of Michigan Press, 1975.

[15] E. Burke, Y. Bykov, J. Newall, and S. Petrovi. “A Time-Predefined approach to Course imetabling” *Journal of Operations Research*, Vol. 13, No. 2 ,pp. 139-151, 2003.

[16] M. Nandhini and D. S. Kanmani. “A Survey of Simulated Annealing Methodology for University Course Timetabling” *International Journal of Recent Trends in Engineering*, Vol. 1, No. 2, pp. 255-257 May 2009.

[17] 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.

[18] A. Schaerf. “Tabu search techniques for large high-school timetabling problems,” *In: Proceedings of the thirteenth national*

*conference on artificial intelligence*. AAAI Press/MIT Press, Portland (OR), pp. 363–368, 1996.
[19] 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.
[20] C. H. Aladag and G. Hocaoglu. “A tabu search algorithm to solve course timetabling problem” *Hacettepe Journal of Mathematics and Statistics*, Vol. 36, NO.1, pp. 53–64, 2007.
[21] R. Alvarez, E. Crespo, and J. M. Tamarit. “Design and implementation of a course scheduling system using tabu search” *European Journal of Operational Research*, Vol. 137, pp. 512–523, 2002.
[22] S. Abdullah, and H. Turabieh. “Generating University Course Timetabling Using Genetic Algorithms and Local Search,” *International Conference on Convergence and Hybrid Information Technolog*, 2008.
[23] D. Landa-Silva, and J. H. Obit, “Great Deluge with Nonlinear Decay Rate for Solving Course Timetabling Problems,” *Proceedings of the 2008 IEEE Conference on Intelligent Systems (IS 2008)*, IEEE Press, pp. 8.11-8.18, 2008.
[24] Dario Landa-Silva, Joe Henry Obit. “Evolutionary Non-Linear Great Deluge for University Course Timetabling,” *Proceedings of the 2009 International Conference on Hybrid Artificial Intelligence Systems (HAIS 2009)*, Lecture Notes in Artificial Intelligence, Vol. 5572, Springer, pp. 269-276, 2009.
[25] D. Abramson, “Constructing school timetables using simulated annealing: sequential and parallel algorithms” *Manag Sci*, Vol. 37, No. 1, pp. 98–113, 1991.
[26] F. Melicio, and J. Caldeira. “Timetabling implementation aspects by simulated annealing,” *In: Jifa Gu (ed), IEEE systems science and systems engineering*, Beijing, Aceite, pp. 553–557, 1998.
[27] S.Elmohamed, G. Fox, and P. Coddington. “A comparison of annealing techniques for academic course scheduling,” *In: Burke E, CarterM(eds) Practice and theory of automated timetabling (PATAT) II*, Vol. 1408, Springer, Berlin, pp. 146–166, 1998.
[28] S.Petrovic, and Y. Bykov. “A multiobjective optimisation approach for exam timetabling based on rajectories,” *In: Burke E, De Causmaecker P (eds) The paractice and theory of automated timetabling (PATAT) IV*, Vol. 2740, Springer, Berlin, pp. 181–194, 2003.
[29] Dawkins, R., *The Selfish Gene*, Oxford University Press, 1976.
[30] L. Paquete, and C. Fonseca. “A study of examination timetabling with multiobjective evolutionary algorithms,” *In:4$^{th}$ metaheuristics international conference (MIC 2001)*, Porto, pp. 149–154, 2001.
[31] K.Socha, and M. Samples. “Ant algorithms for the university course timetabling problem with egard to the state-of-the-art,” *In: Evolutionary computation in combinatorial optimization (EvoCOP 2003)*, Vol. 2611, Springer, Berlin, pp. 334–345, 2003.
[32] K. Socha, J. Knowles, and M. Samples. “A max-min ant system for the university course timetabling problem,” *Proceedings of the 3rd International Workshop on Ant Algorithms (ANTS 2002)*, Springer Lecture Notes in Computer Science, Vol. 2463, pp. 1-13, 2002.
[33] B. Paechter, A. Cumming, M. G. Norman, and H. Luchian. “Extensions to a memetic timetabling system,” *Proceedings of the 1st International Conference on Practice and Theory of Automated Timetabling*, LNCS 1153, pp. 251–265, 1996.
[34] 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, pp. 329–351, 2002.
[35] http://iridia.ulb.ac.be/supp/IridiaSupp2002-001/index.html
[36] B. McCollum. “University Timetabling: Bridging the Gap between Research and Practice,” *Proc of the 6th Int Conf on the Practice and Theory of Automated Timetabling*, pp. 15–35, 2006.
[37] M. A. Al-Betar, A. T. Khader, and T. A. Gani. “A Harmony Search Algorithm for University Course Timetabling,” *In:E. Burke,M. Gendreau. (eds.). The Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling*, Montréal, Canada, 2008.
[38] S. Abdulla, and A. R. Hamdan. “A hybrid approach for University Course Time Tabling” *IJCSNS*, Vol.8, No.8, 2008.
[39] 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 Journal*, Vol. 10, No. 1, pp. 85-86, 1967.