

بررسی مساله زمانبندی درسی دانشگاهی با استفاده از ترکیب الگوریتم ممتیک بهبودیافته و الگوریتم سرد شدن شبیه‌سازی شده

مجید جودکی^۱، سید رسول موسوی^۲، محمد علی منتظری^۳

^۱ دانشجوی کارشناسی ارشد دانشگاه صنعتی اصفهان، m.joudaki@ec.iut.ac.ir

^۲ استادیار دانشکده برق کامپیوتر دانشگاه صنعتی اصفهان، srm@cc.iut.ac.ir

^۳ استادیار دانشکده برق کامپیوتر دانشگاه صنعتی اصفهان، montazer@cc.iut.ac.ir

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 فرامکاشفه-ای (Metaheuristic) بهره برده‌اند. در این مقاله یک مساله زمانبندی واقعی مورد بررسی قرار می‌گیرد و یک روش مبتنی بر الگوریتم ممتیک (Memetic Algorithm) بهبودیافته که از الگوریتم سرد شدن شبیه‌سازی شده (Simulated Annealing) به عنوان رویه جستجوی محلی خود استفاده می‌کند، ارائه می‌شود. منظور از بهبود در الگوریتم ممتیک، استفاده از روش‌های مکاشفه‌ای (Heuristic) در تولید جمعیت اولیه و همچنین تغییر عملگر تقاطع (Crossover) در این الگوریتم می‌باشد. همچنین یک عملگر به نام *improvement* جهت بهبود راه‌حل‌های تولید شده و کاهش تعداد نقض محدودیت‌ها طراحی شده است. بکارگیری روش سرد شدن شبیه‌سازی شده به عنوان رویه جستجوی محلی در الگوریتم ممتیک باعث افزایش توانایی بهره‌برداری (exploiting) این الگوریتم خواهد شد. کارآمدی این روش در مقایسه با برخی روش‌های جدید، با توجه به نتایج به دست 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 ممتیک، جستجوی محلی، سرد شدن شبیه‌سازی شده، مساله زمانبندی درسی دانشگاهی.

به داشتن چند امتحان در یک روز اصلا علاقه‌ای ندارد.

در یک UCTP برگزاری همزمان دو درس با دانشجویان مشترک در یک دوره زمانی یکسان را تداخل می‌گویند. هدف از بررسی یک UCTP در حالت کلی کاهش تعداد تداخل بین دروس با دانشجویان و یا مدرسان مشترک و همچنین رفع همزمانی دروس نیازمند یک اتاق مشترک، می‌باشد [2]. برای موسسه‌های بزرگ مانند دانشگاه‌ها با افزایش تعداد محدودیت‌ها رسیدن به یک جواب قابل قبول 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 رنگ‌آمیزی گراف، انواع مکاشفه‌ها، برنامه‌ریزی خطی، شبکه‌های عصبی، روش‌های فرامکاشفه‌ای، تکنیک‌های ارضای محدودیت و روش‌های مبتنی بر دانش از این جمله‌اند.

۱- مقدمه

تخصیص دروس به اتاق‌های معین در دوره‌های زمانی مشخص را مساله زمانبندی درسی دانشگاهی (UCTP) می‌گویند. منظور از کلاس، مکان تشکیل درس و منظور از دوره زمانی، زمان تشکیل آن می‌باشد. در این مساله هر درس نشان‌دهنده یک رویداد است. اگر رویداد نشان دهنده یک امتحان باشد آن مساله را زمانبندی امتحانی می‌گویند. این دو نوع مساله طبیعتاً 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. در زمانبندی درسی یک دانشجو تمایل به داشتن چندین درس در یک روز دارد ولی

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

تبدیل مساله زمانبندی به مساله شناخته شده رنگ‌آمیزی گراف می‌تواند در رسیدن به جواب مساله کاملاً مفید و موثر باشد [7, 8, 9]. ایده کار به این صورت است که راس‌ها نشان‌دهنده دروس و یال‌های بین رئوس نماینده محدودیت‌ها می‌باشند. مساله رنگ‌آمیزی گراف عبارت است از تخصیص تعداد محدودی رنگ به رئوس گراف به طوری که رئوس مجاور هم‌رنگ نباشند. تعداد رنگ‌ها نشان‌دهنده تعداد دوره‌های زمانی در دسترس می‌باشد.

در کارهای [11, 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 و نرم مساله باشد. کمینه کردن 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 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 و جستجوی ممنوعه به کار 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 کار [15] نیز از الگوریتم Great Deluge در حل مساله استفاده کرده‌اند. آنها بر این باورند که مزیت کار آنها تنظیم کردن تنها دو متغیر 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, 18] مشاهده نمود.

در مورد UCTP، الگوریتم ژنتیک 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 در [20, 21] گزارش شده است. کار انجام شده در [22, 23] از الگوریتم GreatDeluge ولی با نرخ کاهش غیرخطی 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 تحت بررسی دارای مقادیر خاصی هستند. به همین دلیل علی‌رغم جواب‌های خوب این روش، الگوریتم ارائه شده یک الگوریتم کلی نبوده و برای نمونه‌های 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 الگوریتم‌ها صرف‌نظر کرده‌ایم.

الگوریتم‌های بهینه‌سازی که از روش سرد شدن شبیه‌سازی شده بهره می‌برند در [24, 33, 25, 26] پیشنهاد شده است. مثال‌های قابل ذکر دیگری از تکنیک‌های چند هدفه که در جداول زمانبندی امتحانی به کار رفته‌اند را می‌توان در [34, 27] و به صورت مبسوط‌تر در [28] دید. در کارهایی نظیر [29, 30]، دو الگوریتم بر پایه الگوریتم اجتماع مورچگان برای حل UCTP ارائه شده است. [37, 38, 31, 32, 35, 36] نمونه کارهایی هستند که در آنها الگوریتم ممتیک بر UCTP اعمال شده است. در [2] تعداد زیادی از روش‌های فرامکاشفه‌ای برای حل UCTP مورد بررسی قرار گرفته است.

در کنار الگوریتم‌های فرامکاشفه‌ای، الگوریتم‌های ابرمکاشفه‌ای نیز در زمینه مساله زمانبندی درسی دانشگاهی مورد بررسی قرار گرفته‌اند [39, 40, 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 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 مساله زمانبندی درسی دانشگاهی معرفی می‌گردد. روش پیشنهادی برای بررسی مساله زمانبندی درسی دانشگاهی تحت عنوان $MA_SA_ECE_UCTP^2$ در بخش ۳ بررسی می‌شود. در بخش ۴ 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. نتیجه-

گیری از کار انجام شده و ارائه پیشنهاد برای کارهای آینده در بخش ۵ بررسی می‌شوند.

۲- 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. تعداد زیادی از تحقیقات انجام گرفته در این زمینه، محدودیت‌های مذکور را به دو طبقه تقسیم‌بندی کرده‌اند [2]:

محدودیت‌های سخت: محدودیت‌های سخت آن دسته از محدودیت‌ها هستند که ارضاء آنها در جدول زمانبندی اجباری است. **محدودیت‌های نرم:** 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 قابل قبول است که همه‌ی محدودیت‌های سخت آن ارضاء شده باشند. به عنوان مثال یک دانشجو نباید دو درس در یک زمان یکسان داشته باشد. از طرف دیگر، محدودیت‌های نرم، محدودیت‌هایی هستند که با توجه به سیاست‌های دانشگاه و نیاز افرادی که می‌خواهند از آن استفاده کنند، اعمال می‌شوند. به عنوان مثال یک دانشجو سه درس متوالی در یک روز نداشته باشد.

نکته دیگری که در مورد UCTP در همه شکل‌های مختلف آن وجود دارد این است که همه آنها از نظر پیچیدگی محاسباتی به کلاس مسائل NP-hard تعلق دارند و ما در حالت کلی نمی‌توانیم امیدوار به کشف یک الگوریتم با پیچیدگی زمانی از مرتبه چندجمله‌ای برای حل آنها باشیم [2]. حل مسائل زمانبندی واقعی با استفاده از الگوریتم‌های دقیق در عمل غیر ممکن می‌باشد چرا که زمان اجرای این دسته از الگوریتم‌ها با رشد اندازه مساله به صورت نمایی افزایش می‌یابد. بنابراین می‌توان از الگوریتم‌های غیردقیق مانند الگوریتم‌های 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 توسط Ben Paecher و دومین مورد مساله زمانبندی درسی در دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان می‌باشد.

در حالت اول یک UCTP را می‌توان در حالت کلی به صورت

زیر بیان کرد [20]: یک مجموعه از رخدادهای یا کلاس‌ها (E) که باید در ۴۵ برش زمانی (۵ روز و هر روز شامل ۹ ساعت) زمانبندی شوند، یک مجموعه از اتاق‌ها (R) که رخدادهای آن اتفاق می‌افتند، یک مجموعه از دانشجویان (S) که رخدادهای را اخذ می‌کنند و یک مجموعه از خصوصیات (F) که توسط اتاق‌ها ارضاء شده و برای رخدادهای مورد نیاز است. هر دانشجو تعدادی از این رخدادهای را اخذ می‌کند و هر اتاق دارای اندازه‌ای ثابت می‌باشد. یک جدول زمانبندی را **قابل قبول** می‌گویند اگر به هر رخداد یک برش زمانی و یک اتاق اختصاص یابد و محدودیت‌های سخت نیز در جدول ارضاء شوند. این محدودیت‌ها عبارتند از: الف) هیچ دانشجویی بیش از یک درس در یک زمان 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) فقط یک رخداد در هر اتاق، در هر برش زمانی قرار داشته باشد. محدودیت نرم عبارتند از: الف) یک دانشجو در آخرین برش زمانی روز کلاس نداشته باشد. ب) یک دانشجو بیش از دو کلاس پشت سر هم نداشته باشد. ج) یک دانشجو، تنها یک کلاس در یک روز نداشته باشد. هدف این مساله کاهش تعداد نقض‌های محدودیت نرم در یک مساله عملی (قابل قبول) است. تمامی راه‌حل‌های غیرعملی بی‌ارزش هستند.

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 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 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

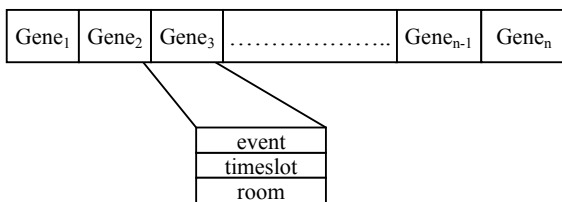
هر کروموزوم شامل مجموعه‌ای از ژن‌ها است که نشان دهنده یک راه‌حل کاندید می‌باشد. ساختار هر کروموزوم در شکل ۲ نمایش داده شده است. در این نمایش هر ژن از یک سه‌تایی $\langle \text{event, timeslot, room} \rangle$ تشکیل شده است. طول کروموزوم-ها (تعداد ژن‌ها در هر کروموزوم) مساوی و برابر با تعداد رویدادها می‌باشد. علت استفاده از این نوع نمایش ساده بودن و سادگی اعمال عملگرهای MA بر روی آن می‌باشد.

```

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

```

شکل ۱: شبه کد الگوریتم MA_SA_ECE_UCTP



شکل ۲: نحوه نمایش کروموزوم‌ها

۳-۲- رویه مقداردهی اولیه مکاشفه‌ای

از آنجا که UCTP بسیار شبیه به مساله رنگ‌آمیزی گراف می‌باشد [2]، می‌توان برای ساخت کروموزوم‌ها از رویه‌های مکاشفه‌ای استفاده شده در مساله رنگ‌آمیزی گراف استفاده نمود [36]. اگر هر رویداد را به عنوان یک راس در نظر بگیریم و چون رویدادها نمی‌توانند در یک برش‌زمانی یکسان قرار گیرند، پس بین هر جفت راس یک یال (محدودیت) اضافه کنیم آنگاه می‌توان تعداد

مشخص شده است. به عنوان مثال درس x باید در روز شنبه ساعت ۸ تا ۹ زمانبندی شود.

زمانبندی دروس در زمان‌هایی که توسط اساتید مربوط به آن دروس از قبل مشخص شده است. به عنوان مثال استاد x فقط روزهای شنبه و دوشنبه تدریس می‌کنند و یا استاد y روزهای یکشنبه صبح تدریس نمی‌کند.

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.

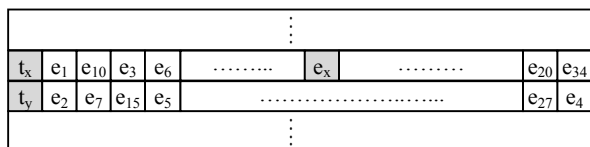
عدم زمانبندی دروس در روز پنج‌شنبه. از جمله مزیت‌های کار انجام شده در این مقاله استفاده از تکنیک‌های مکاشفه‌ای و روش‌های جستجوی فرامکاشفه‌ای در بررسی UCTP می‌باشد. تفاوت عمده این کار با کارهای انجام شده، انعطاف‌پذیری آن برای بررسی داده‌های واقعی و استفاده از آن در محیط واقعی دانشکده می‌باشد. بالا بودن 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.

۳- روش پیشنهادی (MA_SA_ECE_UCTP)

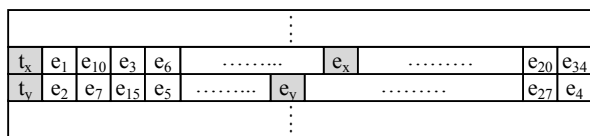
در الگوریتم MA_SA_ECE_UCTP از ترکیب الگوریتم متیک (MA) و روش جستجوی سرد شدن شبیه‌سازی شده (SA) استفاده شده است. نحوه عملکرد MA_SA_ECE_UCTP مطابق شکل ۱ به شرح ذیل می‌باشد: برای هر کروموزوم، ابتدا با استفاده از رویه مقداردهی اولیه مکاشفه‌ای، کروموزوم‌ها مقداردهی اولیه می‌شوند. سپس رویه جستجوی محلی مبتنی بر SA جهت بهبود کروموزوم و تبدیل آن به یک راه‌حل بهینه محلی، فراخوانی می‌شود. بعد از این مرحله رویه improvement بر روی کروموزوم اعمال شده و موجب بهبود هر چه بیشتر آن می‌گردد. در حلقه while بعد از انتخاب دو کروموزوم متناسب با میزان برازندگی آنها از بین جمعیت، و اعمال عملگرهای تقاطع و جهش با احتمال به ترتیب pC و pM یک فرزند تولید می‌شود. سپس رویه‌های SA و improvement به ترتیب فراخوانی شده و 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.

۳-۱- نحوه نمایش راه‌حل با استفاده از کروموزوم

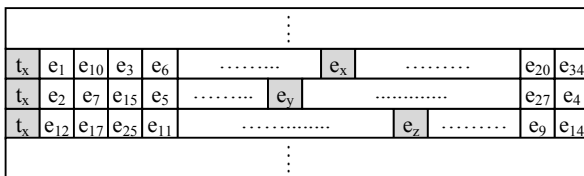
ساختار همسایگی N3: در این ساختار سه رویداد e_x ، e_y و e_z به صورت تصادفی انتخاب شده و برش‌های زمانی آنها (برش‌های زمانی که به این سه رویداد تخصیص یافته‌اند) به صورت چرخه-ای با یکدیگر تعویض می‌شوند. در شکل ۴، ۵ و ۶ نحوه عمل هر یک از ساختارهای همسایگی نشان داده شده است.



شکل ۴: نحوه عملکرد ساختار همسایگی N1. رویداد e_x از برش زمانی t_x به برش زمانی t_y انتقال یافته است.



شکل ۵: نحوه عملکرد ساختار همسایگی N2. 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.



شکل ۶: نحوه عملکرد ساختار همسایگی N3. 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 و رویداد e_z از برش زمانی t_x به برش زمانی t_z انتقال یافته است.

رویه جستجوی محلی مبتنی بر SA می‌باشد و دیگرام عملکرد آن در شکل ۱۵ نمایش داده شده است.

زیادی از مسائل زمانبندی ساده را به مسئله رنگ‌آمیزی گراف (و برعکس) تبدیل نمود [2]. هر برش زمانی را می‌توان به عنوان یک رنگ فرض نمود. شکل ۳ رویه‌های مکاشفه‌ای استفاده شده در مقداردهی اولیه را نشان می‌دهد. این مرحله با یک جدول زمانبندی اولیه تهی و لیستی از رویدادها، L ، که شامل رویدادهای اختصاص نیافته می‌باشد شروع می‌شود (در ابتدا $|L| = n$). با استفاده از رویه‌های مکاشفه‌ای شکل ۳ هر بار یک رویداد و یک محل مناسب انتخاب شده و رویداد به آن محل تخصیص می‌یابد.

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

شکل ۳: رویه‌های مکاشفه‌ای برای ساختن کروموزوم‌ها

۳-۳- رویه جستجوی محلی

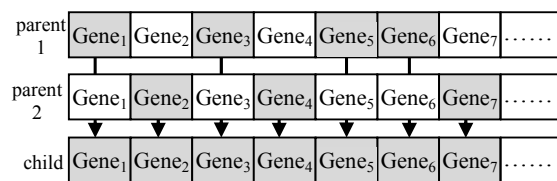
هدف از رویه جستجوی محلی در الگوریتم MA تبدیل یک راه‌حل به یک نقطه بهینه محلی می‌باشد. در این رویه با استفاده از ساختارهای همسایگی که در ادامه خواهند آمد، یک کروموزوم بهبود داده می‌شود. ساختارهای همسایگی به مجموعه حرکت-هایی گفته می‌شود که بر روی یک راه‌حل انجام شده و موجب به‌وجود آمدن راه‌حل‌های همسایه می‌شود. این مجموعه حرکت‌ها تا حد امکان ساده بوده و باعث کاهش فضای جستجوی اطراف کروموزوم خواهد شد. 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 رویداد 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 زمانبندی جدید می‌باشد که در تخصیص رویداد e_x با جدول زمانبندی قبلی متفاوت می‌باشد.

ساختار همسایگی N2: در این ساختار دو رویداد e_x و e_y به صورت تصادفی انتخاب شده و برش‌های زمانی آنها (برش‌های زمانی که به این دو رویداد تخصیص یافته‌اند) با یکدیگر تعویض می‌شود.

۴-۳- 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 رویداد انتخاب شده و تعداد نقض محدودیت ایجاد شده توسط این رویداد در هر دو والد بررسی می‌شود و ژن شامل این رویداد از والدی انتخاب می‌شود که تعداد نقض محدودیت کمتری داشته باشد. سپس این ژن به همراه ژن‌های مرتبط با آن (ژن‌های شامل رویدادهایی که دارای دانشجوی مشترک با این رویداد 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.



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.

۴-۵- عملگر بهبود کروموزوم (improvement)

این عملگر با جابجا کردن دانشجویان از یک گروه درسی به گروه دیگر تعداد نقض محدودیت نرم را کاهش می‌دهد. گروه‌های درسی عبارتند از یک درس که در ساعت‌های متفاوتی ارائه شده‌اند. 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 l و $2l$ می‌باشد. در ابتدا به صورت تصادفی دانشجوی x در یکی از گروه‌های درسی قرار می‌گیرد. در حین جستجو برای رسیدن به جواب، عملگر بهبود تعداد نقض محدودیت انتساب x به هر یک از گروه‌های درسی را بررسی کرده و در صورت نیاز دانشجوی x را بین این گروه‌های درسی جابجا می‌کند.

۴-۴- 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 ۱ نمایش داده شده‌اند.

جدول ۱: 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

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 ۲، 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 ، m و l به ترتیب نشان دهنده نمونه‌های small، medium و large 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 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 ۳ ساعت تا ۱۰ ساعت) نبوده است از عبارت inf برای آن استفاده شده است.

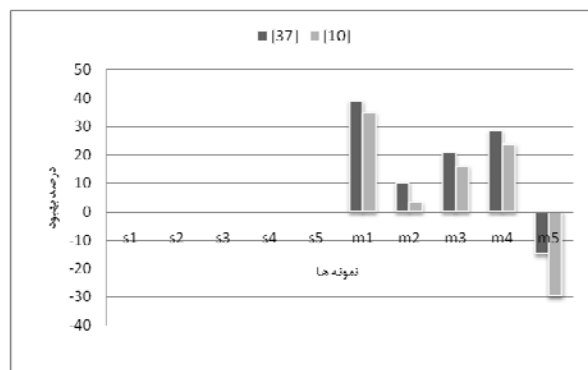
زمان اجرای الگوریتم پیشنهادی برای نمونه‌های small، medium و large به ترتیب ۶۰۰، ۳۰۰ و ۶۰ دقیقه می‌باشد. نتایج به دست آمده از اجرای این روش نشان می‌دهند که MA_SA_ECE_UCTP در همه موارد قادر به پیدا کردن راه‌حل قابل قبول می‌باشد.

جدول ۲: مقایسه MA_SA_UCTP با برخی روش‌های موجود. inf به معنای جواب غیر قابل قبول می‌باشد.

نمونه	MA_SA_UCTP	[5]	[6]	[10]	[13]	[22]	[36]	[37]
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	254	316	236
m2	142	313	161	147	180	258	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 ۸ درصد بهبود روش پیشنهادی نسبت به دو روش [10,37] 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 بجز m5 از دو روش دیگر بهتر بوده 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، به ترتیب s و s' باشد، آنگاه درصد بهبود به صورت زیر محاسبه می‌شود:

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



شکل ۸: مقایسه MA_SA_ECE_UCTP با روش‌های [10,37]. محور افقی نمونه‌های تست و محور عمودی درصد بهبود را نشان می‌دهند.

بیشترین بهبود در نمونه m1 دیده می‌شود. میزان بهبود در نمونه m1 نسبت به روش [10] برابر ۳۴/۸ و نسبت به روش [37] 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 ۸۸-۸۹ 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:

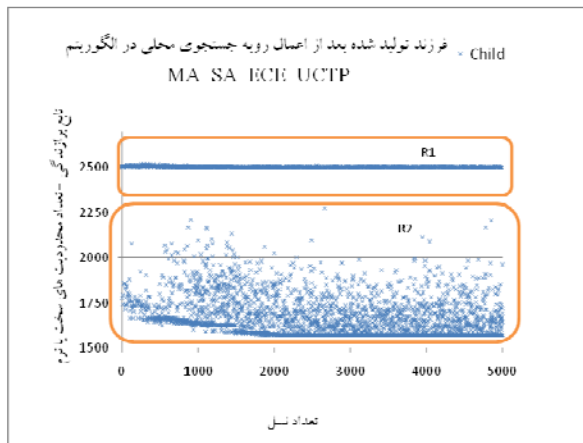
- تعداد کل دانشجویان کارشناسی دانشکده: ۹۷۶ نفر. این تعداد شامل دانشجویان کارشناسی رشته تحصیلی برق (کلیه گرایش‌ها) و دانشجویان رشته تحصیلی کامپیوتر (کلیه گرایش‌ها) می‌باشد.
- تعداد اساتید: ۵۵ نفر که از این تعداد ۴۳ نفر اعضای هیات علمی و ۱۲ نفر شامل اساتید مدعو و دانشجویان دوره دکتری که درس ارائه می‌کنند.
- تعداد کلاس‌ها: ۱۴ اتاق که ۴ کلاس برای دروس ارشد، ۲ کلاس مشترک کارشناسی و ارشد و بقیه برای دوس کارشناسی می‌باشند.
- تعداد دروس نیم‌سال اول ۸۸-۸۹: 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 ۱ می‌باشد.

همانطور که در بخش ۳ اشاره شد این الگوریتم ترکیبی از الگوریتم ممتیک و جستجوی محلی مبتنی بر الگوریتم سرد شدن 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.

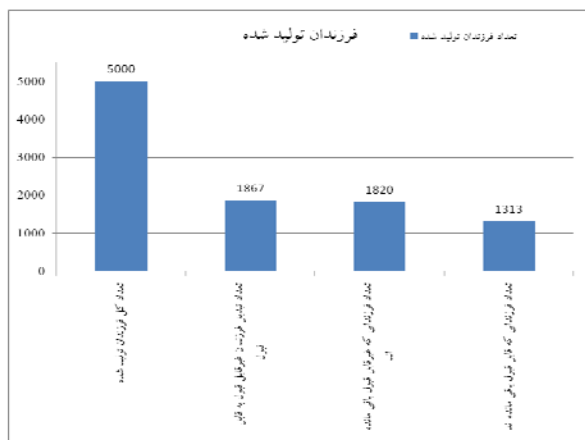
رفتار کلی الگوریتم بر اساس تابع برازندگی بهترین عضو جمعیت آن به صورت نمودار شکل ۹ می‌باشد. در این نمودار محور عمودی نشان دهنده تعداد نقض محدودیت‌ها (مقدار تابع برازندگی) و محور افقی تعداد نسل را نشان می‌دهد. همانطور که

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 ۱۱ مجدداً مقدار تابع برازندگی فرزند تولید شده در طول نسل‌ها بعد از اعمال رویه جستجوی محلی نمایش داده شده است.



شکل ۱۱: تابع برازندگی فرزند تولید شده در الگوریتم MA_SA_ECE_UCTP بعد از اعمال رویه جستجوی محلی. ناحیه R1 فرزندان غیرقابل قبول و ناحیه R2 فرزندان قابل قبول

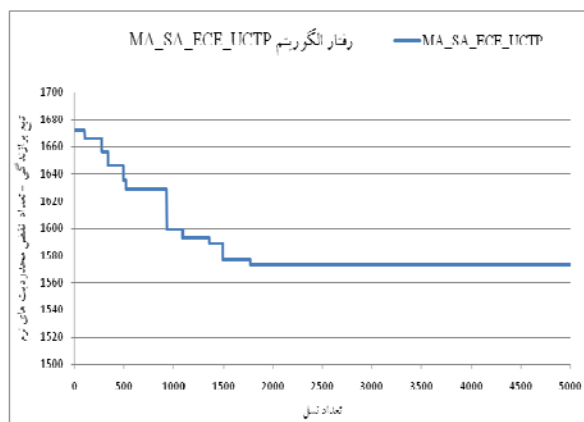
با توجه به شکل ۱۱ مشاهده می‌شود که تعداد زیادی از فرزندان تولید شده که در ناحیه R1 بودند (شکل ۱۰) بعد از اعمال رویه جستجوی محلی به ناحیه R2 منتقل شده‌اند. در نمودار شکل ۱۲ تاثیر این رویه بر تابع ارزیاب فرزندان تولید شده به صورت آماری نشان داده شده است.



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

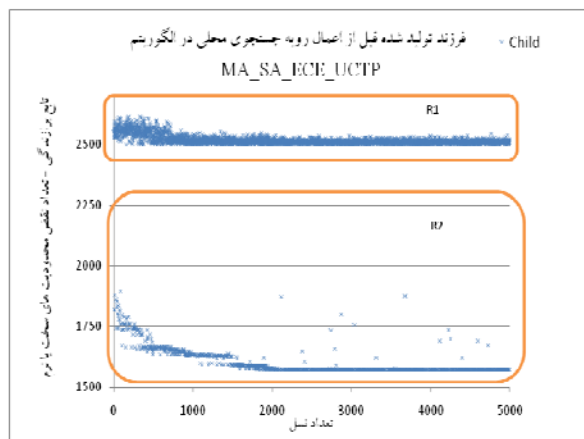
در نمودار شکل ۱۲ ستون اول از سمت چپ تعداد کل فرزندان

در نمودار دیده می‌شود وجود رویه مقداردهی اولیه مکاشفه‌ای باعث ایجاد راه‌حل‌های اولیه قابل قبول شده است (در نسل اول مقدار تابع برازندگی نشان‌دهنده عدم نقض محدودیت سخت می‌باشد). در نسل‌های تقریباً کوچکتر از ۲۰۰۰ مقدار تابع برازندگی سیر نزولی داشته و همواره راه‌حل به دست آمده بهبود داده شده است. در نسل‌های بالاتر از ۲۰۰۰ هیچ‌گونه بهبودی در راه‌حل جواب دیده نمی‌شود.



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

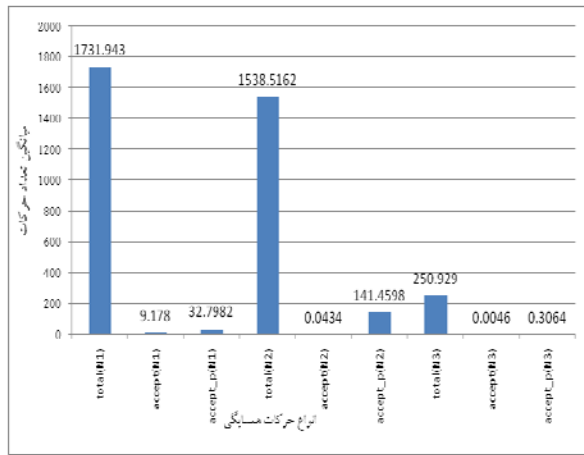
در نمودار شکل‌های ۱۰ و ۱۱ رفتار فرزند تولید شده در این الگوریتم نشان داده شده است.



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

نمودار شکل ۱۰ مقدار تابع برازندگی فرزند تولید شده در طول نسل‌ها را نشان می‌دهد. این مقدار تابع برازندگی، قبل از اعمال

بار اعمال جستجوی محلی بر فرزند تولید شده در هر نسل که منجر به هر یک از ساختارهای همسایگی N1، N2 و N3 شده است، نشان داده شده است.



شکل ۱۴: میانگین تعداد دفعات استفاده شده از ساختارهای همسایگی N1، N2 و N3 در اعمال رویه جستجوی محلی بر فرزند تولید شده در هر نسل.

در این نمودار محور عمودی نمایانگر میانگین تعداد حرکات می باشد و عناصر محور افقی به شرح ذیل می باشند:

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

accept(N1): میانگین تعداد حرکات از نوع ساختار همسایگی N1 که منجر به راه حل های جدید بهتر از راه حل جاری شده اند. این راه حل های جدید همگی پذیرفته می شوند.

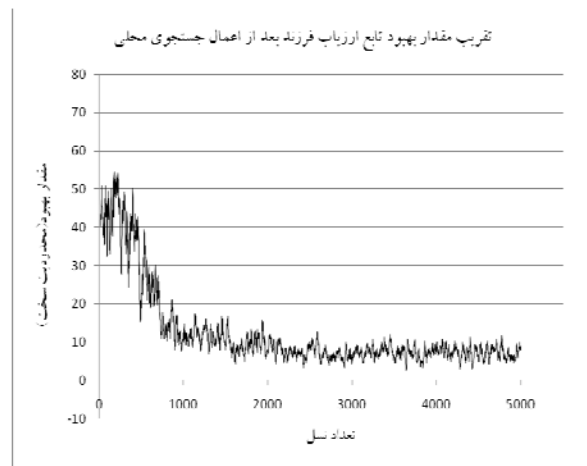
accept_p(N1): میانگین تعداد حرکات از نوع ساختار همسایگی N1 که منجر به راه حل های بدتر نسبت به راه حل جاری شده اند ولی با توجه به شرط پذیرش با احتمال p در الگوریتم سرد شدن شبیه سازی شده، پذیرفته شده اند.

برای ساختارهای همسایگی N2 و N3 نیز ستون هایی به ترتیب بالا رسم شده است. در جدول ۳ اعداد موجود در نمودار بالا به صورت درصد نشان داده شده است.

جدول ۳: 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%

تولید شده را در طی نسل ها نشان می دهد و ستون دوم از سمت چپ تعداد فرزندان را نشان می دهد که بعد از اعمال رویه جستجوی محلی از حالت غیر قابل قبول به حالت قابل قبول تبدیل شده اند. با توجه به اعداد نشان داده شده در این نمودار ۳۷/۳۴ درصد فرزندان دچار این تغییر شده اند. ۳۶/۴ درصد از فرزندان تولید شده دارای حالت غیر قابل قبول بوده اند و بعد از اعمال رویه جستجوی محلی نیز در این حالت باقی مانده اند. ۲۶/۲۶ درصد نیز از فرزندان بعد از تولید، دارای حالت قابل قبول بوده اند. در نمودار شکل ۱۳ تقریب مقدار بهبود تابع ارزیاب فرزند بعد از اعمال جستجوی محلی در طول همه ی نسل ها مشخص شده است. این بهبود فقط شامل بهبود نقض محدودیت سخت می باشد. در این نمودار محور عمودی تعداد بهبود نقض محدودیت سخت را نشان می دهد و محور افقی نمایانگر تعداد نسل می باشد. همانطور که مشاهده می شود 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 می افتد کاهش می یابد و در نتیجه تاثیر رویه جستجوی محلی بر آنها چشمگیر نخواهد بود.



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

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

در نمودار شکل ۱۴ میانگین تعداد حرکات انجام شده در هر

برای رهایی از نقاط بهینه محلی یکی از راهکارهایی است که می‌توان برای رفع این مشکل بکار برد. علاوه بر انواع همسایگی، 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.

۶- 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. Gislén, 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*

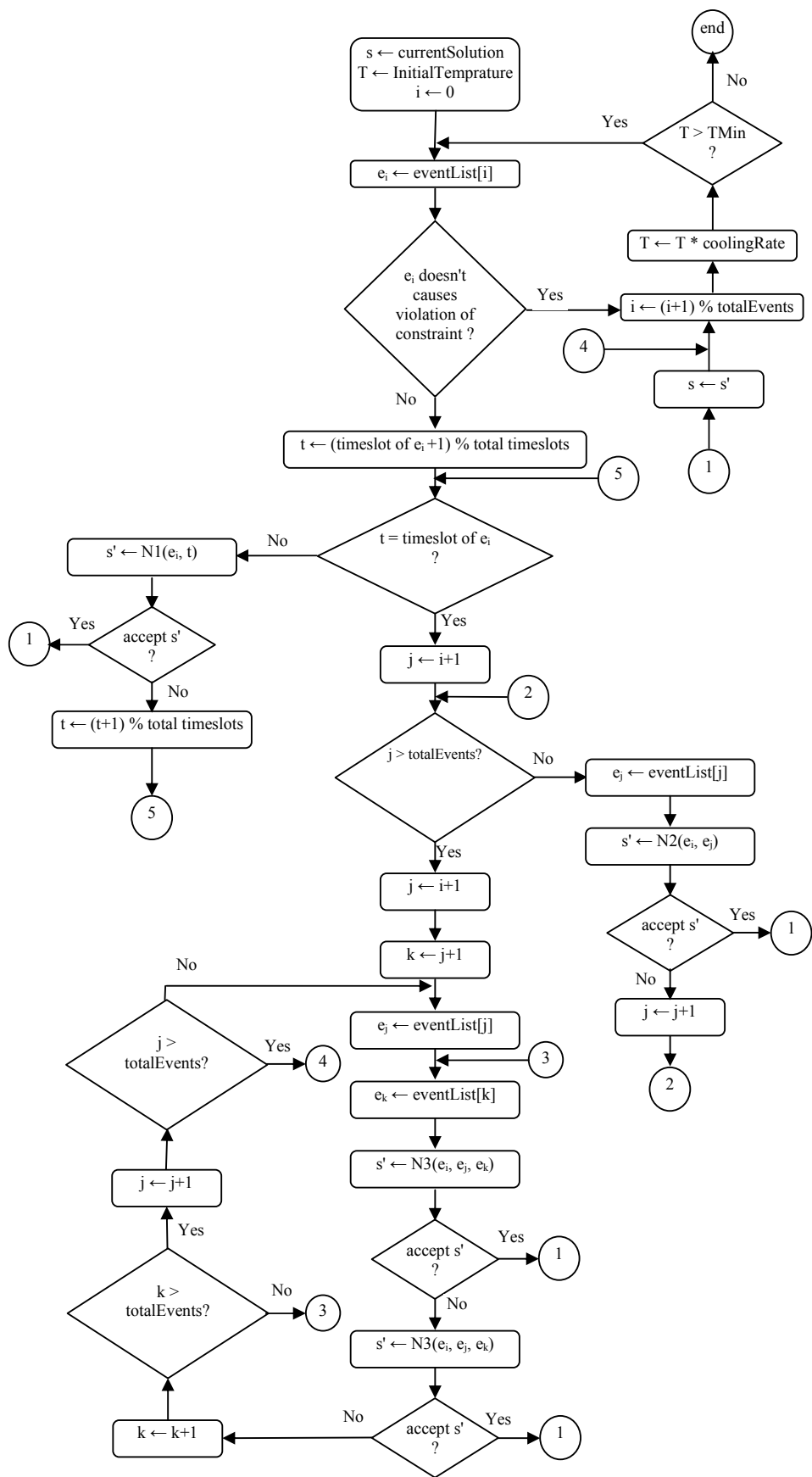
عددهای موجود در این جدول مربوط به میانگین حرکات در جستجوی محلی برای یک فرزند می‌باشد. همانطور که در جدول دیده می‌شود ۴۹/۱۸۳۵ درصد از کل حرکات منجر به ساختار همسایگی N1 شده‌اند که از این تعداد ۰/۵۲۹۹ درصد ساختارهایی بوده‌اند که راه‌حل‌های جدید بهتر تولید کرده‌اند و از میان راه‌حل‌های تولید شده بدتر ۱/۸۹۳۷ درصد پذیرفته شده‌اند. برای ساختارهای همسایگی N2 و N3 نیز ترتیب عددهای موجود در جدول به همین صورت می‌باشد.

۵- 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 قابل قبول شدن راه‌حل‌های 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 قابل قبول و رسیدن به جواب-های 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 improvement در این 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 گیر کردن الگوریتم در این نقاط می‌باشد. بهره‌گیری از ساختارهای همسایگی مختلف

problems". *Computer Journal*, Vol. 10, No. 1, pp. 85-86, 1967.

- 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.Elmoahmed, 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] L. Paquete, and C. Fonseca. "A study of examination timetabling with multiobjective evolutionary algorithms," *In:4th metaheuristics international conference (MIC 2001)*, Porto, pp. 149-154, 2001.
 - [30] 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.
 - [31] 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.
 - [32] 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.
 - [33] 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.
 - [34] <http://iridia.ulb.ac.be/supp/IridiaSupp2002-001/index.html>
 - [35] 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.
 - [36] 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.
 - [37] S. Abdulla, and A. R. Hamdan. "A hybrid approach for University Course Time Tabling" *IJCSNS*, Vol.8, No.8, 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



شکل ۱۵: دیاگرام نحوه عملکرد رویه جستجوی محلی مبتنی بر سرد شدن شبیه سازی شده

¹ University Course Timetabling Problem

² Memetic Algorithm_Simulated Annealing_Electical and Computer Engineering_UCTP