

استفاده از مدل محاسباتی مخچه برای محاسبه نرخ جهش در الگوریتم ممتیک برای حل مسأله برنامه‌ریزی دروس دانشگاهی

امیرمسعود رحمانی
دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران
rahmani@sr.iau.ac.ir

نرجس خاتون ناصری
دانشگاه آزاد اسلامی واحد ماهشهر
naseri50542@yahoo.com

امین جولا
دانشگاه آزاد اسلامی واحد ماهشهر
amin.jula@gmail.com

نشان داده‌اند که در بسیاری موارد توانسته‌اند نسبت به دیگر روشها، به موفقیت بیشتری دست یابند [5].

از آنجا که در خانواده الگوریتم ژنتیک، عملگر جهش عامل جلوگیری از گرفتاری فرآیند جستجو در کمینه محلی و گمگشت پاسخهای مناسب است، به دست آوردن احتمال اجرای آن با استفاده از دانش موجود در فرآیند جستجو، می‌تواند تأثیر هدایت شده و مطلوب تری در سرعت یافتن پاسخ بهینه، داشته باشد. برای دستیابی به این هدف، می‌توان از مدل محاسباتی مخچه (CMAC) که یک ماشین آموزش پذیر سریع است، استفاده نمود.

در این نوشتار با تشریح CMAC طراحی شده برای به دست آوردن نرخ جهش در الگوریتم ممتیک پیشنهادی، تأثیر استفاده از آن برای حل مسأله برنامه‌ریزی دروس دانشگاهی بررسی خواهد شد.

در ادامه پس از تعریف مسأله برنامه‌ریزی دروس دانشگاهی و الگوریتمهای ممتیک، به تشریح ساختار و عملکرد CMAC پرداخته، پس از توضیح قالب کروموزوم، عملگرهای تولید مثل و جستجوی محلی پیشنهادی، نحوه استفاده از CMAC در محاسبه هوشمند نرخ جهش مورد بررسی قرار خواهد گرفت.

تعاریف اولیه

مسأله برنامه‌ریزی دروس دانشگاهی

برنامه‌ریزی دروس دانشگاهی یعنی تخصیص یک درس به دو منبع بازه زمانی و اتاق به گونه‌ای که محدودیت‌های مورد نظر در آن نقض نشوند. در یک مسأله برنامه‌ریزی دو دسته محدودیت وجود دارد. دسته اول محدودیت‌هایی که پاسخ ارایه شده می‌بایست لزوماً برآورده شوند که محدودیت‌های سخت نام دارند و دسته دوم که محدودیت‌های نرم نامیده می‌شوند و بهتر است در پاسخ ارایه شده رعایت شوند، اما رعایت آنها الزامی نیست. محدودیت‌های گفته شده برای برنامه‌ریزی دروس دانشگاهی به صورت مفصل در [11] بررسی شده‌اند.

اگر به تعداد q درس، به تعداد p بازه زمانی و به تعداد r اتاق وجود داشته باشد، پاسخ مسأله، ماتریس صحیح $T_{q \times p}$ است که در آن T_{ij}

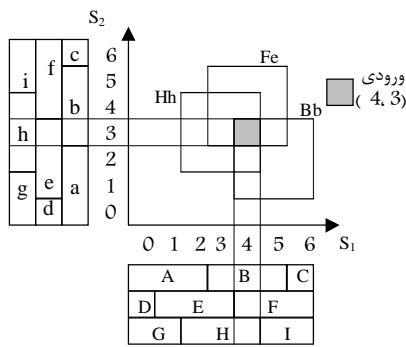
چکیده: الگوریتم ممتیک نمونه‌ای از الگوریتمهای تکاملی است که برای حل یک مسأله بهینه‌سازی، با افزودن جستجوی محلی به یک الگوریتم ژنتیک، منجر به دستیابی به پاسخهای بهتر در زمان کمتر می‌شود. برنامه‌ریزی دروس دانشگاهی نیز از جمله مسایل بهینه‌سازی با فضای جستجوی بسیار بزرگ است که به دلیل تأثیر عوامل متعدد، تحقیقات گسترده‌ای را به سوی خود معطوف داشته است. در این مقاله، با ارایه یک الگوریتم ممتیک ابتکاری، نشان داده شده است که می‌توان برای دستیابی به پاسخ بهینه مسأله برنامه‌ریزی دروس دانشگاهی در زمان کوتاه‌تر، از مدل محاسباتی مخچه CMAC به منظور به دست آوردن احتمال اجرای عملگر جهش، استفاده نمود. نتایج حاصل از مقایسه الگوریتم ممتیک ابتکاری با الگوریتم سنتی تأیید کننده این مطلب است.

واژه‌های کلیدی: الگوریتم ممتیک (Memetic Algorithm)، مسأله برنامه‌ریزی دروس دانشگاهی، الگوریتم ژنتیک، نرخ جهش، مدل محاسباتی مخچه (CMAC) (Cerebellar Model Arithmetic Computing).

مقدمه

برنامه‌ریزی دروس دانشگاهی مسأله‌ای است که در آن برای دروس دانشگاهی یک جدول برنامه هفتگی تولید می‌شود. برنامه‌ریزی باید به گونه‌ای باشد که دروس در تعداد معینی از اتاق‌ها و بازه‌های زمانی به گونه‌ای قرار گیرند که در یک اتاق و بازه زمانی مشخص، بیش از یک درس قرار نگیرد. گونه‌های مختلفی از این مسأله در دانشگاه‌های مختلف و مطابق با قوانین، نیازها و محدودیت‌های آنها وجود دارد. با توجه به محدودیت‌ها و پارامترهای مؤثر فراوان و تعداد پاسخ‌های ممکن، یافتن یک پاسخ دقیق و کامل در زمان محدود برای این مسأله بسیار مشکل و گاهی غیرممکن می‌نماید. به همین دلیل این مسأله در زمره مسایل NP-hard است.

از دیگر سو، الگوریتم‌های ممتیک در موارد متعدد و در کنار دیگر روشهای حل مسایل NP-hard برای حل این مسأله به کار رفته [3-6] و



شکل ۱: نمایش مدل CMAC

با شیفیت دادن هر بلوک به اندازه فاصله کوچکی بنام عنصر، فوق مکعبهای جدیدی در لایه‌ها یا سطرهاى مختلف بوجود می‌آیند. مثلاً برای بلوکهای A, B, C مربوط به متغیر ورودی S1 بلوکهای D, E, F در لایه دوم و بلوکهای G, H, I در لایه سوم ایجاد می‌شود. با توجه به اینکه فقط بلوکهای موجود در لایه‌های متناظر تشکیل فوق مکعب می‌دهند، بنابراین فوق مکعبهای Ig, Hh, Fe, Ee به دست می‌آید.

مدل CMAC با استفاده از جدول جستجو و بر اساس سلولهای حافظه انجمنی (فوق مکعبها) خروجی را بدست می‌آورد. به ازای متغیرهای ورودی تعدادی از سلولهای حافظه انجمنی فعال شده و با بازیابی وزنه‌های ذخیره شده در آنها و حاصلجمع آنها می‌توان خروجی را بدست آورد. به طور مثال با اعمال ورودی $(S_1, S_2) = (4, 3)$ ، خروجی مدل مجموع وزنه‌های موجود در فوق مکعبهای Hh, Fe, Bb است. خروجی مدل CMAC مطابق رابطه زیر بدست می‌آید:

$$y(s) = a^T(s) W = \sum_{j=1}^M a_j(s) W_j \quad (1)$$

که در آن S یک حالت ورودی مشخص، M حجم حافظه و برابر تعداد فوق مکعبها است و W بردار وزنها یا عبارت دیگر اطلاعات ذخیره شده در حافظه‌ها است و اگر محل حافظه j توسط یکی از فوق مکعبهای حالت ورودی S پوشانده شود $a_j(s)$ برابر یک در غیر این صورت برابر صفر است.

CMAC با تغییر وزن فوق مکعبهای متناظر با ورودیها خروجی صحیح را یاد می‌گیرد. الگوریتم یادگیری آن به صورت زیر است:

$$W_{new} = W_{old} + \frac{a}{N_e} a(s)(\hat{y}(s) - a^T(s)W_{old}) \quad (2)$$

که در آن α نرخ یادگیری است بطوریکه $0 < \alpha < 1$ ، $\hat{y}(s)$ مقدار خروجی یا پاسخ مورد انتظار، N_e تعداد لایه‌ها، $\hat{y}(s) - a^T(s)W_{old}$ نیز معادل خطا برای نمونه آموزشی است. خطا در $a(s)$ ضرب می‌شود تا فقط وزن محل‌های حافظه که توسط فوق مکعبهای ورودی پوشانده می‌شود، تنظیم شوند [12].

$k = 1 \leq k \leq r$ بدین معنی است که درس i در بازه زمانی j و در اتاق k برگزار می‌شود. سعی می‌شود T به گونه‌ای یافته شود که در آن محدودیت‌های نرم کمترین میزان نقض را داشته باشند و محدودیت‌های سخت کاملاً رعایت شده باشند.

2-2 الگوریتم ممیتیک

الگوریتم ممیتیک گونه‌ای از الگوریتم‌های تکاملی است که در آن جستجوهای ابتکاری محلی با الگوریتم ژنتیک ترکیب می‌شوند تا زمان دستیابی به پاسخ بهینه کاهش یابد [6].

الگوریتمهای ژنتیک برای جستجوی سراسر فضای جستجو ایجاد می‌شوند، در حالی که جستجوی محلی حوزه همسایگی هر پاسخ یافته شده توسط الگوریتم ژنتیک را برای یافتن پاسخهای بهتر مورد جستجو قرار می‌دهد. انتخاب عملگرهای تولید نسل در بخش ژنتیک یک الگوریتم ممیتیک و نوع و روش جستجوی محلی مورد استفاده در آن، منجر به نتایج اجرای بسیار متفاوت خواهد شد.

3- مدل محاسباتی مخچه

مدل CMAC یا مدل محاسباتی مخچه در سال 1972 توسط J.S. Albus ارایه شد [2و1]. CMAC شبکه ای عصبی است که ساختار و اعمال قسمتی از مغز انسان به نام مخچه را شبیه سازی می‌کند. مدل CMAC مبتنی بر سلولهای حافظه انجمنی و براساس جدول جستجو است.

بلوک دیاگرام مدل CMAC به ازای ورودی یا ورودی های مشخص، خروجی یا خروجی های مناسب تولید می‌کند. متغیر یا متغیرهای ورودی به نواحی گسسته تقسیم می‌شود، که هر ناحیه گسسته آدرس سلولی از حافظه را مشخص می‌کند. با بازیابی اطلاعات ذخیره شده در سلول های حافظه انجمنی انتخاب شده، می توان خروجی را به دست آورد. خروجی، به صورت مجموع اطلاعات ذخیره شده در سلول های حافظه متناظر با ورودی ها است. CMAC با تغییر محتویات سلول های حافظه متناظر با ورودی ها، خروجی صحیح را فرا می‌گیرد. برای هر ورودی، ضربی از مقدار تفاوت خروجی CMAC و خروجی مورد انتظار به عنوان ضرب یادگیری، به محتویات سلولهای حافظه انتخاب شده، افزوده می‌شود.

مدل CMAC با دو متغیر ورودی S1 و S2 در شکل 1 آمده است. متغیرهای ورودی مدل CMAC به نواحی گسسته بنام بلوک تقسیم می‌شوند. به طور مثال بلوکهای A, B, C برای متغیر ورودی S1 و بلوکهای a, b, c برای متغیر S2. مساحتی که توسط نواحی تقسیم شده (بلوکهای دو متغیر ورودی) بدست می‌آید بنام فوق مکعب نامیده می‌شود. به طور مثال فوق مکعبهای Ba, ..., Ba, ..., Bc, ..., Bc, ...، هر فوق مکعب سلولی از حافظه برای ذخیره و بازیابی اطلاعات یا وزنها است.

4- الگوریتم ممتیک پیشنهادی

نمایش کروموزوم

در الگوریتم ممتیک هر کروموزوم یک پاسخ یا راه حل برای مسأله است. در الگوریتم پیشنهادی، یک ژن، یک رکورد با چهار فیلد است که به ترتیب مربوط به شماره استاد، روز هفته، شماره بازه زمانی در روز و شماره اتاق هستند. یک کروموزوم نیز، رکوردی متشکل از آرایه‌ای از ژن‌ها و فیلدی برای نگهداری شایستگی کروموزوم ایجاد شده است. نمونه‌ای از کروموزوم مورد استفاده در شکل 2 آمده است.

۲۳۳۳۳	۱۱۱۱۱	۱۱۱۱۱	۱۱۱۱۱
-------	-------	-------	-------

شکل 2: نمونه‌ای از کروموزوم مورد استفاده

برای ایجاد جمعیت اولیه باید آرایه‌ای از کروموزوم پیشنهادی تعریف شود. در الگوریتم پیشنهادی، کروموزومها کاملاً هوشمندانه تولید می‌شوند. استادان به ترتیب شماره‌ی استادی و بر اساس برنامه‌ای که خود پیشنهاد نموده‌اند در کروموزومها قرار می‌گیرند و برای انعطاف در برنامه‌ریزی همیشه برای تعداد کمتری از بازه‌های زمانی پیشنهادی استاد برنامه‌ریزی می‌شود، اما انتخاب بازه‌های زمانی به صورت تصادفی انجام می‌شود. سپس از میان اتاقهای در اختیار یک اتاق به صورت تصادفی انتخاب و به همه ژنهای مربوط به یک استاد تخصیص داده می‌شود که تا حد امکان کلاسهای هر استاد در یک اتاق برگزار شود. پس از تولید هر کروموزوم، جستجوی محلی بر روی آن اجرا می‌شود تا کروموزوم تولید شده، در حوزه همسایگی خود بهینه باشد.

4-2 تابع شایستگی

تابع شایستگی در الگوریتم حل مسأله برنامه‌ریزی دروس دانشگاهی، وظیفه یافتن تعداد نقضهای محدودیتها را برعهده دارد. لذا هرچه مقدار شایستگی راه‌حلی کمتر باشد، راه‌حل مناسبتری خواهد بود. رابطه 3 تابع شایستگی مورد استفاده را نشان می‌دهد.

$$f(T) = \sum p_j g_j(T) \quad (3)$$

که در آن p_j میزان جریمه نقض محدودیت j و $g_j(T)$ تعداد نقض‌ها در جدول برنامه‌ریزی T و برای هر محدودیت j است. برای سادگی، در این مقاله مقدار p_j برای محدودیت‌های استفاده شده برابر یک انتخاب شده است.

در مسأله برنامه‌ریزی دروس دانشگاهی، به دلیل ساختار کروموزومها و تعدد فیلدهای تشکیل دهنده ژنها، تابع شایستگی مورد استفاده سهم عمده‌ای از زمان اجرای الگوریتم را به خود اختصاص می‌دهد؛ لذا تلاش در جهت کاهش زمان اجرای این تابع منجر به کاهش زمان اجرای کل الگوریتم خواهد شد. از آنجا که در روش پیشنهادی، به دلیل کروموزوم سازی هوشمندانه، به هیچ عنوان محدودیت عدم برنامه‌ریزی دروس در

ساعات غیر پیشنهادی استادان، نقض نمی‌شود، لذا جستجوی راه‌حلهای برای یافتن تعداد نقضهای این محدودیت لزومی ندارد و به همین دلیل، سرعت عملیات محاسبه شایستگی افزایش چشمگیری خواهد داشت.

4-3 عملگر تقاطع

عملگر تقاطع، یکی از مهمترین عملگرهای تولید نسل در الگوریتم ژنتیک است. این عملگر به نحوی بخشی از یک کروموزوم را به بخشی از یک کروموزوم دیگر الحاق می‌کند که کروموزوم جدید که فرزند نامیده می‌شود، با والدین خود هم اندازه و هم ساختار باشد.

برای اجرای عملگر تقاطع ابتدا برای انتخاب دو کروموزوم والد بر اساس آنچه که در [5] به طور کامل شرح داده شده است، کروموزومی به صورت تصادفی به عنوان والد اول انتخاب می‌شود، سپس از بین سایر کروموزومها آنکه دارای حداکثر میزان اختلاف با والد انتخابی اول باشد به عنوان والد دوم انتخاب می‌شود. پس از آن براساس شیوه عملگر عملگر تقاطع دونقطه، دو نقطه متفاوت از هرکدام از دو کروموزوم والد به طور تصادفی انتخاب شده و تقاطع روی آنها انجام می‌شود. از آنجا که استادان به ترتیب در کروموزومها قرار می‌گیرند، تقاطع هیچ تأثیری در محل قرارگیری آنها ندارد و نظم موجود در کروموزومها بر هم نمی‌خورد.

عملگر جایگزینی و یا انتقال کروموزومها به نسل بعد از یک سیاست نخبه‌گزینی جدید استفاده می‌کند. دو کروموزوم برتر از بین دو کروموزوم والد و دو کروموزوم فرزند، انتخاب شده و کروموزوم بهتر مستقیماً به نسل بعد انتقال می‌یابد و کروموزوم دیگر جایگزین بدترین کروموزوم موجود در نسل فعلی می‌شود.

البته جایگزینی فقط زمانی انجام می‌شود که کروموزوم تولید شده بهتر از کروموزومی باشد که برای جایگزینی انتخاب شده است. این روش جایگزینی باعث می‌شود که همیشه، نسل جدید دارای کروموزومهای بهتری نسبت به نسل پیش از خود باشد.

4-4 عملگر جهش

عملگر جهش را می‌توان به شیوه‌های متفاوت بر روی کروموزومها اجرا نمود؛ اما آنچه در الگوریتم پیشنهادی استفاده شده است یک عملگر جهش ابتکاری است. از آنجا که مشکل اصلی در کروموزومها تداخل در برنامه یک اتاق در یک بازه زمانی و یک روز مشخص است و با توجه به این که روزها و بازه‌های زمانی تخصیص داده شده کاملاً هوشمندانه است و نمی‌توان آنها را کورکورانه تغییر داد، لذا آنچه می‌تواند تغییر کند تنها اتاق محل تشکیل کلاس است. پس یک استاد و یک اتاق را به صورت تصادفی انتخاب نموده و تمام محل‌های برگزاری دروس آن استاد به اتاق انتخاب شده نسبت داده می‌شود. نرخ جهش به صورت هوشمند و بر اساس یک الگوریتم ابتکاری با استفاده از CMAC بدست می‌آید. دلیل استفاده از CMAC سرعت بالا و همگرایی تضمین شده آن است [2و12].

یابد تا تغییرهای بیشتر در کروموزومها، منجر به تولید پاسخهای بهتر شود. با توجه به این موارد، برای آموزش پویای CMAC قدرمطلق حاصل تفاضل ضریب موفقیت فعلی و قبلی از عدد ثابت 0/2 تفریق می‌شود. سپس حاصل به دست آمده برای افزایش نرخ جهش، با مقدار فوق مکعبی از CMAC که به وسیله سه پارامتر پیش تر گفته شده انتخاب شده است جمع شده و یا برای کاهش نرخ جهش، از آن کم می‌شود. سپس از مقدار فعلی فوق مکعب انتخابی و مقدار به دست آمده جدید میانگین گیری شده و حاصل در فوق مکعب انتخاب شده قرار می‌گیرد. مقدار ثابت 0/2 نیز بر اساس سعی و خطا و آزمایش مقادیر مختلف بدست آمده است. شکل 3، CMAC مورد استفاده و پارامترهای ورودی آن را نشان می‌دهد.

6-4 جستجوی محلی

یک الگوریتم جستجوی محلی از یک حالت اولیه S_0 آغاز شده و وارد حلقه‌ای می‌شود که فضای جستجو را که همان حوزه همسایگی $N(S_0)$ یک کروموزوم است، پیمایش می‌نماید. جستجو در هر بار تکرار حلقه، حرکت از یک حالت S_i به یک حالت همسایه S_{i+1} آن است [12].

جستجو در هر مرحله، توسط تابع شایستگی، کیفیت هر حالت را تخمین می‌زند و کار را ادامه می‌دهد تا بهترین کروموزوم را در همسایگی کروموزوم فعلی بیابد. سه جستجوی محلی اصلی به نامهای تپه‌نوردی، شبیه‌سازی گداختگی و جستجوی حرام وجود دارند [1]. جستجوی محلی مورد استفاده در الگوریتم پیشنهادی، یک جستجوی محلی ابتکاری است که برای مسأله برنامه‌ریزی دروس دانشگاهی، در مقایسه با روشهای سنتی دارای کارایی بهتری است. این روش ابتکاری نسبت به روشهای سنتی نیازمند زمان اجرای بسیار کمتری است. به همین دلیل، بخش ژنتیکی الگوریتم ممتیک می‌تواند در یک زمان مشخص نسل‌های بیشتری را تولید و منجر به دستیابی به پاسخ بهتر در زمان بسیار کمتر شود.

این جستجو، ابتدا کروموزوم را برای یافتن کلاسهایی که برای یک استاد در یک روز و یک بازه زمانی برنامه‌ریزی شده‌اند، جستجو نموده و برای هر یک، برنامه ارایه شده توسط استاد را بررسی و بازه زمانی دیگری را که دچار این مشکل نباشد انتخاب و جایگزین آن می‌نماید. در همین حال اگر اتاقی را بیابد که در یک روز و یک بازه زمانی مشخص بیش از یک کلاس به آن اختصاص داده شده باشد، برنامه اتاقها را بررسی و اتاقی را که در آن روز و بازه زمانی، در آن برنامه‌ریزی نشده باشد انتخاب و جایگزین اتاق مشخص شده در کروموزوم می‌کند. سپس شایستگی کروموزوم جدید محاسبه شده، اگر شایستگی بهبود یافته باشد، کروموزوم حاصل از جستجوی محلی جایگزین کروموزوم ابتدایی می‌شود، در غیر این صورت همان کروموزوم ابتدایی به عنوان بهترین راه‌حل در همسایگی خود شناخته شده و بدون تغییر باقی خواهد ماند. از آنجا که این جستجوی محلی با استفاده از دانش موجود در برنامه و

5-4 محاسبه نرخ جهش با استفاده از CMAC

آموزش ابتدایی CMAC

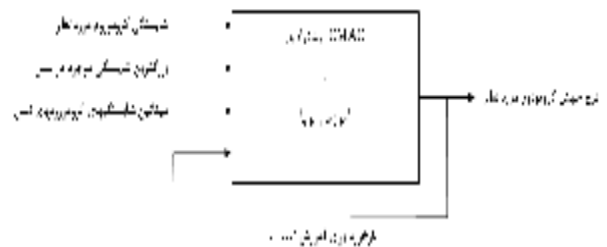
در ابتدا CMAC با سه پارامتر میانگین شایستگیهای کروموزومهای نسل (f_{avg}) ، شایستگی کروموزوم مورد نظر برای اعمال عملگر جهش (f) و بزرگترین مقدار شایستگی موجود در نسل (f_{max}) ، به صورت ایستا و با استفاده از رابطه 4 مقاداردهی اولیه می‌شود [11].

$$\begin{cases} P_m = 0.05 (|f_{max} - f / (f_{max} - f_{avg})|, f \geq f_{avg} \\ P_m = 0.3, \text{ otherwise} \end{cases} \quad (4)$$

هر کدام از پارامترها به گونه ای مورد تقسیم بندی قرار می‌گیرد که شامل شش بلوک شود. همچنین CMAC مورد استفاده به صورت دولا به پیاده سازی می‌شود، یعنی در هر مراجعه به CMAC دو فوق مکعب از آن فعال خواهند شد.

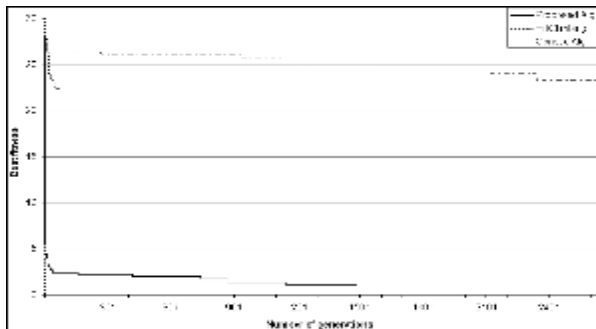
به دست آوردن نرخ جهش و آموزش پویای CMAC

تا تولید سی نسل از مقادیر CMAC استفاده نموده و از تغییر در آنها اجتناب می‌شود تا تأثیر آن در روند اجرا مشخص شود. سپس آموزش CMAC می‌بایست در طول اجرا، به صورت پویا دنبال شود. بدین منظور هر ده تولید نسل جدید را یک ((مرحله)) نامیده، پس از پایان هر مرحله، نسبت تعداد تولید نسل تاکنون، بر میانگین شایستگیهای کروموزومهای نسل را به دست آورده، حاصل تقسیم آن بر حاصل همین نسبت در مرحله قبل محاسبه می‌شود. آنچه به دست می‌آید، ((ضریب موفقیت)) نامیده می‌شود.

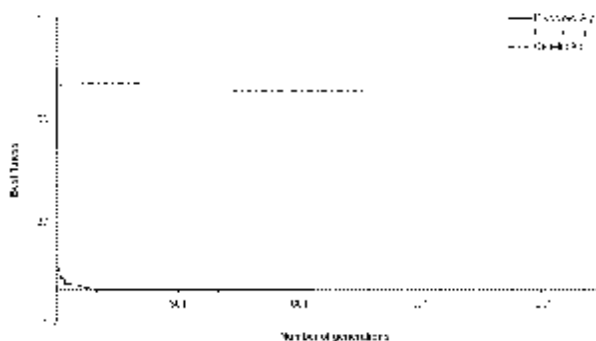


شکل 3: CMAC طراحی شده برای الگوریتم پیشنهادی

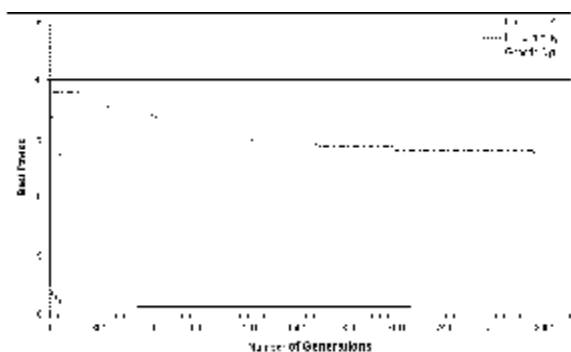
اگر ضریب موفقیت به دست آمده در این مرحله از اجرا از ضریب موفقیت مرحله قبل بزرگتر باشد، بدین معنی است که روند اجرا مناسب است و کروموزومهای فعلی در فرآیند تکامل به سمت تولید پاسخ بهینه در حرکت هستند. پس می‌بایست احتمال اجرای عملگر جهش که باعث دگرگونی در نسل می‌شود، کاهش یابد تا همین روند حفظ شود. اما اگر ضریب موفقیت به دست آمده در این مرحله از ضریب موفقیت مرحله قبل کوچکتر و یا با آن برابر باشد، این نتیجه به دست می‌آید که کروموزومهای فعلی به احتمال فراوان منجر به تولید نسلهای بهتر نخواهند شد، از این رو احتمال انجام عملگر جهش می‌بایست افزایش



شکل 4: مقایسه نتایج اجرای الگوریتم ممتیک پیشنهادی و الگوریتم ممتیک استفاده کننده از جستجوی محلی تپه نوردی و الگوریتم ژنتیک در مسأله ای با 20 استاد



شکل 5: مقایسه نتایج اجرای الگوریتم ممتیک پیشنهادی و الگوریتم ممتیک استفاده کننده از جستجوی محلی تپه نوردی و الگوریتم ژنتیک در مسأله ای با 40 استاد



شکل 6: مقایسه نتایج اجرای الگوریتم ممتیک پیشنهادی و الگوریتم ممتیک استفاده کننده از جستجوی محلی تپه نوردی و الگوریتم ژنتیک در مسأله ای با 15 استاد

بررسی و مقایسه نتایج به دست آمده و نمودارهای رسم شده در شکل‌های 7 و 8 نشان می‌دهد که استفاده از CMAC به دلیل بکارگیری دانش موجود در فرآیند حل مسأله، منجر به بهبود پاسخهای میانی و پایانی شده، در زمانهای کوتاهتر پاسخهای بهتری را موجب می‌شود.

کاملاً هوشمندانه رفتار می‌کند، تا آنجا که ممکن است کروموزوم ابتدایی را بهبود می‌بخشد.

5- پیاده سازی و مقایسه نتایج

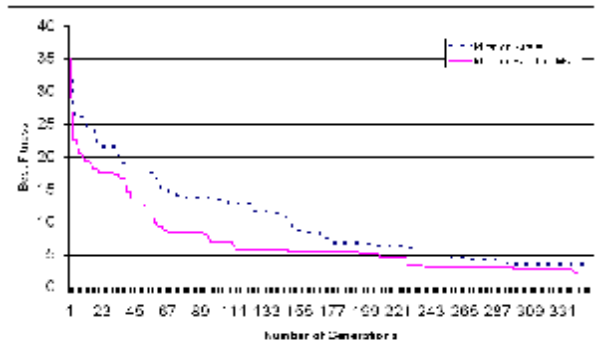
مسأله برنامه‌ریزی دروس دانشگاهی و الگوریتم‌های ارائه شده برای حل آنها بر روی یک کامپیوتر پنتیوم 4 با پردازنده 2/8 گیگاهرتز با 518 مگابایت حافظه اصلی و در 210 ثانیه اجرا شدند. مسأله برنامه‌ریزی دارای محدودیتهای زیر است: (1) یک استاد نمی‌تواند در یک روز و بازه زمانی بیش از یک درس ارائه نماید. (2) در روز و بازه‌های زمانی که استاد برنامه اعلام نکرده است، نمی‌توان برنامه‌ریزی نمود. (3) در یک روز و یک بازه زمانی در یک اتاق بیش از یک کلاس را نمی‌توان برنامه‌ریزی نمود.

برنامه اعلام شده توسط استادان نیز باید دارای شرایط زیر باشد: (1) هر استاد باید حداقل برای سه روز برنامه اعلام نماید. (2) برنامه اعلامی استاد باید میزان ساعات موظفی وی را پوشش دهد (در اینجا 15 ساعت موظف). (3) هر استاد باید حداقل دو بازه زمانی علاوه بر میزان موظفی خود اعلام نماید تا انعطاف در برنامه‌ریزی وجود داشته باشد.

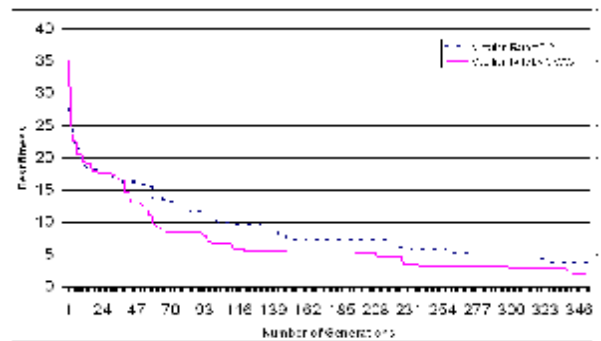
در پیاده سازی انجام شده، برای هر روز چهار بازه زمانی سه ساعته در نظر گرفته شده است اما می‌توان آن را برای مسایل مختلف تغییر داد. در اجرای آزمایشی، الگوریتم پیشنهادی برای سه مسأله مختلف که دارای 15، 20 و 40 استاد و هر کدام به ترتیب دارای 6، 8 و 17 اتاق بودند، اجرا شد. احتمال اجرای عملگر تقاطع 0/9 و تعداد کروموزومهای هر نسل را به طور ثابت 100 در نظر گرفته و ساعت حضور استادان نیز به صورت تصادفی ایجاد شد. در طول اجرای برنامه برای هر کدام از سه مسأله، مقدار شایستگی (تعداد نقض محدودیت) بهترین کروموزوم نگهداری شد. بار دیگر برای حل مسایل مذکور از جستجوی محلی تپه‌نوردی به عنوان نمونه‌ای از جستجوهای محلی سنتی، به جای جستجوی محلی پیشنهادی استفاده شد. نتایج حاصل از این سه اجرا در شکل‌های 4 و 5 و 6 آمده است. بررسی این نمودارها نشان می‌دهد که جستجوی محلی پیشنهادی نسبت به جستجوی تپه نوردی که یکی از انواع جستجوهای محلی سنتی است از سرعت بسیار بالاتری برخوردار است و در زمان یکسان منجر به دستیابی به پاسخهای بسیار بهتر می‌شود.

برای بررسی تأثیر استفاده از CMAC در محاسبه نرخ جهش، مسأله ای با 80 استاد و 27 اتاق نیز با استفاده از الگوریتم پیشنهادی حل شده و نتایج حاصل از آن، با نتایج حاصل از اجرای الگوریتم با نرخهای جهش ثابت 0/1 و 0/2 مقایسه شده است.

- controller (CMAC)", Journal of Dynamic systems, measurements and control, 1975
- [3] E. K. Burke, G. Kendall, and E. Soubeiga, "A tabu search hyperheuristic for timetabling and rostering," Journal of Heuristics, vol. 9, no. 6, 2003.
- [4] E. Burke and J. Newall, "A multistage evolutionary algorithm for the timetable problem," IEEE Trans. Evol. Comput. vol. 3, no. 1, pp. 63–74, Apr. 1999
- [5] Knowles J.D., Corne D.W. M-PAES,"A memetic algorithm for multiobjective optimization", Proceedings of the congress on evolutionary computation, (CEC 2000), IEEE press, pp. 325–332, 2000.
- [6] Natalio Krasnogor, Jim Smith, "A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues, IEEE Transactions on Evolutionary Computation, Vol. 9, No. 5, October 2005.
- [7] N. L. Law, K.Y. Szeto,"Adaptive Genetic Algorithm with Mutation and Crossover Matrices", IJCAI, pp. 2330-2333, 2006.
- [8] Rahmani A. M., Teshnehlab M., "One-Day_Ahead Temperature Prediction using a new neural network", International Journal of Engineering Science (IUST), Vol. 16(3), 4, pp. 1-8, 2005.
- [9] O. Rossi-Doria, C. Blum, J. Knowles, M. Samples, K. Socha, B. Paechter. "A local search for the timetabling problem", Proceedings of the 4th international conference on the Practice and Theory of Automated Timetabling, pp. 124–127. Gent, Belgium, 2002.
- [10] Andrea Schaerf, Luca Di Gaspero, "Local Search Techniques for Educational Timetabling Problems", 2002.
- [11] Srinivas, M., Patnaik, L. M., "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms", IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, No.4, pp. 656-667, IEEE
- [12] Talbi E.G. Taxonomy of hybrid metaheuristics. Journal of heuristics, vol.1 8, 541–564, 2002.
- [13] D. De Werra, "An introduction to timetabling", European Journal of Operations Research, 19:151-162. , 1985.



شکل 7: مقایسه استفاده از نرخ جهش ثابت 0/1 و نرخ جهش حاصل از CMAC



شکل 8: مقایسه استفاده از نرخ جهش ثابت 0/2 و نرخ جهش حاصل از CMAC

6- نتیجه گیری

مسأله زمانبندی دروس دانشگاهی به دلیل وجود محدودیتهای فراوان از مشهورترین مسایل NP-hard و در زمره مسایل بهینه سازی مشکل قرار می گیرد. بررسیها نشان می دهد که در زمینه مسایل بهینه سازی مشکل، طراحی الگوریتمهای فوق ابتکاری می تواند منجر به دستیابی به نتایج بهتر شود. نتایج حاصل از اجراهای مختلف الگوریتم پیشنهادی این مقاله در حل مسایل متفاوت برنامه ریزی دروس دانشگاهی و مقایسه نمودارهای حاصل از آنها نشان می دهد که طراحی الگوریتم فوق ابتکاری به منظور حل مسأله برنامه ریزی دروس دانشگاهی نیز منجر به یافتن پاسخهای بهتر در زمانهای کوتاهتر برای این مسأله مشکل بهینه سازی می شود. همچنین استفاده از مدل محاسباتی مخچه، CMAC، به منظور استفاده از دانش درون فرآیند در حال اجرا برای محاسبه نرخ جهش، می تواند موجب تسریع در یافتن پاسخ بهینه یک مسأله بهینه سازی شود.

مراجع

- [1] Albus.J.S., "Theoretical and practical aspects of a cerebellar model", PhD Dissertation, University of Maryland, 1972
- [2] Albus.J.S., "A new approach to manipulator control: the cerebellar model articulation