

الگوریتم کلونی مورچه سلولی

سحر کیانفر^۱، محمدرضا میبیدی^۲

^۱ دانشجوی کارشناسی ارشد، آزمایشگاه محاسبات نرم، دانشگاه صنعتی امیرکبیر، تهران
kianfar@aut.ac.ir

^۲ گروه هوش مصنوعی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران
mmeybodi@aut.ac.ir

چکیده

بسیاری از مسائل دنیای واقعی پویا هستند. برای حل یک مسئله بهینه‌سازی پویا نیاز به الگوریتمی داریم که علی‌رغم پیدا کردن بهینه در محیط، بتواند بهینه‌های در حال تغییر را دنبال کند. تاکنون الگوریتم‌های تکاملی مختلفی برای بهینه‌سازی در محیط‌های پویا پیشنهاد شده است. در یک محیط پویا پس از روی دادن تغییر در محیط، الگوریتم نیاز به تنوع کافی جهت جستجوی دوباره محیط دارد، درعین حال استفاده از اطلاعات جستجوهای پیشین روند جستجو را سریع‌تر می‌کند. مشکل اصلی الگوریتم‌های تکاملی معمول در حل مسائل بهینه‌سازی پویا، همگرایی زودرس و کاهش تنوع جمعیتی در طول زمان است. بنابراین، در مواجهه با مسائل بهینه‌سازی پویا نیاز به رویکردهایی است که تنوع را در طول زمان حفظ کنند.

در این مقاله یک الگوریتم کلونی مورچه سلولی برای حل مسائل بهینه‌سازی پیوسته و پویا ارائه می‌شود. ایده‌ی اصلی الگوریتم برای حفظ تنوع، تقسیم‌بندی فضای جستجو با استفاده از اتوماتای سلولی است. جهت ارزیابی الگوریتم پیشنهادی، از تابع محک قله‌های متحرک استفاده می‌شود.

کلمات کلیدی

الگوریتم کلونی مورچه، اتوماتای سلولی، بهینه‌سازی، محیط پویا، تابع قله‌های متحرک

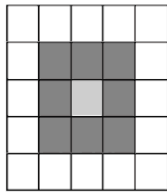
پویا است. با این وجود کارهای کمی برای الگوریتم کلونی مورچه در محیط‌های پویا انجام شده است [۲-۴].

۱- مقدمه

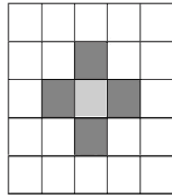
الگوریتم DHCIAC اولین الگوریتم کلونی مورچه‌ی به کار رفته در بهینه‌سازی مسائل پیوسته و پویاست [۲]. این الگوریتم از دو کانال ارتباطی بین مورچه‌ها بهره می‌گیرد. الگوریتم دیگر CANDO نام دارد [۳] و برای حفظ تنوع جمعیت یک شارژ الکتریکی دافع به هر مورچه نسبت می‌دهد. این شارژ از نزدیک شدن مورچه‌ها به یکدیگر تا حدفاصل معینی جلوگیری می‌کند بنابراین از همگرایی همه‌ی مورچه‌ها به یک نقطه ممانعت می‌شود و تنوع در جمعیت حفظ می‌شود.

در این مقاله یک الگوریتم کلونی مورچه مبتنی بر اتوماتای سلولی به نام الگوریتم کلونی مورچه سلولی (CellularACO) ^۲ ارائه می‌شود. در این الگوریتم، یک اتوماتای سلولی فضای جستجو را به تعدادی پارتیشن که هر کدام یک سلول از اتوماتای سلولی است تقسیم‌بندی

در اواخر ۱۹۸۰ زمینه‌ی تحقیقاتی جدیدی در هوش مصنوعی توزیع شده پدیدار شد که هوش جمعی نامیده می‌شود. در این زمینه از حرکت اجتماعی حشرات برای ایجاد الگوریتم‌های جدید الهام گرفته شد. درحقیقت توانایی تولید ساختارهای پیچیده و پیدا کردن راه حل برای مسائل غیربديهی (مرتب‌سازی، جستجوی بهینه، ...) با استفاده از عامل‌هایی که دید کلی از محیطشان ندارند، بدون هیچ کنترل مرکزی یا استراتژی کلی، محققان را جذب کرد. مفاهیم زیادی مثل خودسازماندهی و غیرمنتظره بودن بعدها مطرح شد. علم کامپیوتر از دو مفهوم خودسازماندهی و غیرمنتظره بودن برای تعریف هوش جمعی استفاده کرد. یکی از مزایای اصلی الگوریتم‌های الهام گرفته از طبیعت، مانند الگوریتم کلونی مورچه (ACO) ^۱ انعطاف‌پذیری در محیط‌های



(ب)



(الف)

شکل ۱) همسایگی در اتوماتای سلولی (الف) ون نیمون (ب) مور

۳- الگوریتم کلونی مورچه پیوسته

برای تطبیق الگوریتم کلونی مورچه برای محیط پیوسته، توزیع احتمال گسسته اختصاص یافته به مولفه‌ها را که بیانگر مقدار فرومون است، با یک توزیع احتمال پیوسته مدل می‌کنیم. بدین منظور ما از توزیع گاوسی نرمال استفاده می‌کنیم. و با نمونه برداری از این توزیع راه‌حل جدید را می‌سازیم. توزیع دارای دو پارامتر میانگین و واریانس است. دو نوع فرومون یکی برای مقادیر میانگین و دیگری برای واریانس تعریف می‌شود. هر سلول C_i یک بردار میانگین μ^i و یک بردار واریانس σ^i دارد. مقادیر این بردارها مستقیماً مقدار فرومون را نشان می‌دهد. برای یک محیط d بعدی داریم:

$$\mu^i = \{\mu_1^i, \mu_2^i, \dots, \mu_d^i\}, \quad (1)$$

$$\sigma^i = \{\sigma_1^i, \sigma_2^i, \dots, \sigma_d^i\}. \quad (2)$$

در شروع الگوریتم مقدار میانگین هر سلول برابر با یک نقطه تصادفی درون سلول و مقدار واریانس با توجه به مشخصات فضای جستجو با استفاده محاسبه می‌شود.

برای هر مورچه، با نمونه‌برداری از توزیع نرمال به دست آمده، با استفاده از یک تولید کننده تصادفی نرمال مانند روش جعبه مولر^۷ با میانگین μ_j^i و واریانس σ_j^i مقدار z_j امین متغیر مورچه i به دست می‌آید، فرآیند ساخت تا انتخاب d امین متغیر تکرار می‌شود. در انتها مقدار تابع هدف محاسبه می‌شود. در انتهای فرآیند ساخت راه‌حل به تعداد مورچه‌ها (M) راه حل جدید ساخته می‌شود. مرحله به روز رسانی فرومون شامل تبخیر و تقویت فرومون است. یعنی برای هر سلول در تکرار l ام مقداری از فرومون روی سلول با استفاده از یک ضریب ثابت فراموش می‌شود (روابط (۳) و (۴)). در این روابط ρ نرخ تبخیر فرومون است.

$$\vec{\mu}^i(t) = (1 - \rho) \vec{\mu}^i(t-1), \quad (3)$$

$$\vec{\sigma}^i(t) = (1 - \rho) \vec{\sigma}^i(t-1). \quad (4)$$

بهترین مورچه Ant_{best} برای تقویت مقادیر فرومون انتخاب می‌شود. مقادیر فرومون با استفاده از مورچه‌ی انتخابی مطابق روابط (۵) و (۶) تقویت می‌شود. میانگین و واریانس سلول به سمت بهترین مکان سلول حرکت می‌کنند.

می‌کند. این ایده برای اولین بار توسط هاشمی و میبیدی [۴، ۵] ارائه شد. اتوماتای سلولی بطور ضمنی کلونی مورچه‌ها را به کلونی‌های کوچکتری تقسیم می‌کند. هریک از این کلونی‌ها وظیفه جستجو در سلول مقیم و سلول‌های همسایه را دارد. در ابتدای الگوریتم مورچه‌ها بطور تصادفی در فضای جستجو توزیع می‌شوند. مورچه‌ها در محیط حرکت می‌کنند تا به سلولی که احتمال وجود قله در آن بیشتر است، برسند. در این مقاله یک الگوریتم کلونی مورچه برای حرکت مورچه‌ها در محیط پیوسته پیشنهاد شده است. زمانی که تعداد مورچه‌های سلول زیاد می‌شود، برخی از مورچه‌های سلول به تصادف انتخاب شده و به سلول‌های دیگری در اتوماتای سلولی انتقال می‌یابند. هنگامی که تغییری در محیط رخ می‌دهد، بهترین مورچه سلول یک جستجوی محلی انجام می‌دهد.

ادامه مقاله بصورت ذیل سازماندهی شده است. در بخش دوم اتوماتای سلولی معرفی می‌گردد. در بخش سوم یک الگوریتم کلونی مورچه برای حرکت مورچه‌ها در محیط پیوسته ارائه می‌شود. در بخش چهارم الگوریتم جستجوی محلی بکار رفته معرفی می‌گردد. در بخش پنجم الگوریتم پیشنهادی یعنی الگوریتم کلونی مورچه سلولی ارائه می‌شود. در بخش ششم طی انجام آزمایشاتی کارایی الگوریتم مورد بررسی قرار می‌گیرد. در پایان نتیجه گیری ارائه می‌شود.

۲- اتوماتای سلولی

اتوماتای سلولی (CA)^۲ در اواخر دهه ۱۹۴۰ توسط ون نیومن^۴ مطرح و پس از او توسط ریاضیدانی بنام اولام^۵ به عنوان مدلی برای بررسی رفتار سیستم‌های پیچیده پیشنهاد شد. اتوماتای سلولی در حقیقت سیستم‌های دینامیکی گسسته‌ای هستند که رفتارشان کاملاً بر ارتباط محلی استوار است [۶].

در اتوماتای سلولی، فضا به صورت یک شبکه تعریف می‌گردد که به هر خانه آن یک سلول گفته می‌شود. زمان به صورت گسسته پیش می‌رود و قوانین آن به صورت سراسری است که از طریق آن در هر مرحله هر سلول، وضعیت جدید خود را با در نظر گرفتن همسایه‌های مجاور خود بدست می‌آورد. اتوماتای سلولی را می‌توان به عنوان سیستم‌های محاسباتی نیز در نظر گرفت که اطلاعات کد شده در خودشان را پردازش می‌کنند. قوانین اتوماتای سلولی، نحوه تأثیر پذیرفتن سلول از سلول‌های همسایه خود را مشخص می‌کند. یک سلول را همسایه سلول دیگر گوئیم هرگاه بتواند آن را در یک مرحله و براساس قانون حاکم تحت تأثیر قرار دهد. همسایگی ون نیمون و مور^۸ با شعاع همسایگی ۱ در یک فضای دو بعدی در شکل ۱ نشان داده شده است.

اگر تعداد ابعاد محیط d باشد و هر بعد به N_p بخش مساوی تقسیم شود، نیاز به یک اتوماتای سلولی d بعدی $CA = (Z^d, \varphi, N, F)$ با مجموعه سلول‌های $C = \{cell_i | 1 \leq i \leq (N_p)^d\}$ داریم. برای هر سلول یک همسایگی تعریف می‌شود که دربرگیرنده سلول‌هایی است که با آن سلول همسایه هستند و تمامی اعمال تکاملی بر روی آن‌ها به صورت محلی انجام می‌پذیرد.

هر سلول در اتوماتای سلولی شامل اطلاعات حافظه‌ی سلول، مورچه‌های فعال در سلول و اطلاعات فرومون است که توسط مورچه‌ها در هر سلول برجای گذاشته می‌شود. جهت استفاده از نتایج جستجوهای پیشین برای هر سلول حافظه در نظر گرفته می‌شود. در این حافظه مکان و شایستگی بهترین موجود در آن سلول از زمان آخرین تغییر تاکنون حفظ می‌شود. این حافظه برای سلول i ام، C_i^{Best} نامیده می‌شود.

مورچه‌ها برحسب مکانشان به حالت سلول‌های متناظرشان اضافه می‌گردند و وظیفه جستجو در آن سلول و همسایگی آن را بر عهده دارند. هر مورچه می‌تواند فعال یا غیرفعال باشد. مورچه‌های غیرفعال در فرآیند جستجو شرکت نمی‌کنند و فقط در حالت سلول موجودند. در ابتدا تمام مورچه‌ها فعال هستند.

برای هر سلول پارامتر فرومون تعریف می‌شود. این مقدار با حرکت مورچه‌ها در محیط بروزرسانی می‌شود و مورچه‌های سلول آن را در فرآیند ساخت راه‌حل به کار می‌گیرند. مقدار فرومون می‌تواند میانگین یا واریانس توزیع نرمالی که برای ساخت راه‌حل توسط مورچه‌ها به کار می‌رود، یا هر دو مورد باشد. در ابتدا فرومون همه‌ی سلول‌ها برابر با یک مقدار اولیه است.

۵-۱- روند اجرای الگوریتم

بعد از مقداردهی اولیه که شامل توزیع مورچه‌ها در اتوماتای سلولی است، الگوریتم در یک حلقه قوانین بروزرسانی اتوماتای سلولی را تکرار می‌کند. اتوماتای سلولی شامل ۶ قانون است که آن‌ها را بصورت موازی بر روی تک‌تک سلول‌ها اجرا می‌کند. این قوانین عبارتند از: (۱) نظارت بر تغییرات محیط، (۲) جابجایی مورچه‌های غیرفعال (۳) به روزرسانی حافظه سلول‌ها، (۴) بهنگام سازی مکان مورچه‌ها (با استفاده از الگوریتم کلونی مورچه) (۵) به‌روزرسانی فرومون سلول‌ها، (۶) بررسی تراکم مورچه‌های درون سلول. در ادامه جزئیات قوانین آورده شده است.

نظارت بر تغییرات محیط: برای شناسایی تغییر در محیط شایستگی بهترین مکان سلول C_i^{Best} محاسبه می‌شود، اگر تغییر کرده باشد، تغییر در محیط رخ داده است. بمحض شناسایی تغییر در محیط، شایستگی مورچه‌های داخل سلول مجدداً محاسبه شده و حافظه سلول فراموش می‌شود.

جابجایی مورچه‌های غیرفعال: همگام جابجایی مورچه با استفاده از الگوریتم کلونی مورچه ممکن است مکان جدید مورچه در سلول

$$\vec{\mu}^i(t) = \rho \cdot \vec{x}_{best}, \quad (5)$$

$$\vec{\sigma}^i(t) = \rho * \left| \vec{x}_{best} - \vec{\mu}^i \right|, \quad (6)$$

\vec{x}_{best} مکان بهترین راه‌حل ساخته شده در سلول است.

۴- جستجوی محلی HJ

HJ یک روش جستجوی الگو^۸ است که در سال ۱۹۶۱ توسط هوک و جیوز^۹ معرفی شد [۸]. علی‌رغم گذشت زمان طولانی از ارائه الگوریتم، هنوز هم در بسیاری از مسائل جستجوی محلی قطعی به عنوان انتخاب اول بکار می‌رود. HJ راه‌حل به دست آمده از جستجوی سراسری را به عنوان نقطه پایه انتخاب می‌کند و در دو فاز اکتشاف و حرکت الگو آن را تقویت می‌کند.

الگوریتم با یک حرکت اکتشافی آغاز می‌شود که در آن همه‌ی متغیرهای نقطه پایه به نوبت با یک گام از پیش تعریف شده (s_0) حرکت می‌کند. سپس حرکت الگو همه‌ی تغییرات موفق فاز اکتشاف را همزمان تکرار می‌کند. نتیجه حرکت ارزیابی می‌شود و در صورت موفق بودن حرکت نقطه جدید به عنوان نقطه پایه به کار می‌رود. این پروسه تا زمانی که بهبودی در هیچ یک از ابعاد حاصل نشود، تکرار می‌شود. اندازه گام‌های بکار رفته در طول زمان کاهش می‌یابد. شبه کد جستجوی محلی در شکل ۲ نشان داده شده است.

Hooke-Jeeves Pattern Search Algorithm

1. Obtain initial base point x^t . Determine set of step lengths
2. Move the base point along every one of the d dimensional axes at a time and evaluate the result. Adopt each new point if improvement on the previous point. This takes at least d , at most $2d$ evaluations. If any of the moves was successful, go to 3. If none was successful, go to 4.
3. Repeat the successful moves in a combined pattern move. If the new point has a better fitness, assume it as the new base point. Return to 2 whichever the outcome.
4. Adjust step length to next smaller step. If there is a smaller step, continue from 2. If not, terminate.

شکل ۲) شبه کد الگوریتم جستجوی محلی HJ [۸]

۵- الگوریتم کلونی مورچه سلولی

در الگوریتم کلونی مورچه سلولی، یک اتوماتای سلولی فضای جستجو را به تعدادی پارتیشن افراز می‌کند، که هریک از این پارتیشن‌ها یک سلول در اتوماتای سلولی است. اتوماتای سلولی بطور ضمنی کلونی مورچه‌ها را به تعدادی کلونی کوچکتر تقسیم می‌کند که هر کلونی ساکن یک سلول است. هر کلونی مقیم سلول مسئول پیدا کردن بهینه در آن سلول و سلول‌های همسایه‌ی آن سلول است.

رخ می‌دهد، تعداد قله‌ها، شکل قله‌ها و مکان قله‌ها تغییر می‌کند. میزان شدت تغییرات و فرکانس تغییرات قابل تنظیم است. از معیار کارایی خطای آفلاین^{۱۱} استفاده می‌شود. خطای آفلاین برابر با میانگین بهترین مقدار شایستگی محاسبه شده در هر گام از الگوریتم است. یعنی داریم:

$$offline_error = \frac{1}{Max_FE} \sum_{t=1}^{Max_FE} (f(Ant_{best}(t))) \quad (7)$$

در صورتی که در تمام موارد ارزیابی شایستگی، مقدار کارایی آفلاین برابر صفر باشد، مقدار خطای آفلاین صفر خواهد بود. پارامترهای تابع محک قله‌های متحرک در جدول ۱ ذکر شده است.

در الگوریتم کلونی مورچه نرخ تبخیر فرومون برابر با ۰,۳ و تعداد مورچه‌ها برابر ۱۰۰ در نظر گرفته می‌شود. برای اتوماتای سلولی تعداد تقسیم بندی و ظرفیت سلول برابر ۱۰ و همسایگی نوع مور با شعاع ۲ در نظر می‌گیریم. برای جستجو محلی اندازه گام ولیه ۵، شتاب کاهش برابر ۰,۲، تعداد دفعات کاهش هر بعد برابر ۴ است. نتایج بدست آمده با نتایج الگوریتم‌های [۴] Cellular Pso و [۱۱] mQSO مقایسه شده است. الگوریتم Cellular PSO از جهت شباهت ساختاری به الگوریتم پیشنهادی انتخاب شده است. الگوریتم mQSO به عنوان یک الگوریتم استاندارد برای مقایسه در محیط‌های پویا به کار می‌رود. نتایج بدست آمده در مقایسه با الگوریتم‌های دیگر برای فرکانس تغییر ۲۵۰۰ و ۵۰۰۰ به ترتیب در جدول ۲ و ۳ درج شده است. همانطور که نتایج نشان می‌دهد الگوریتم دارای خطای کمتری نسبت به الگوریتم‌های دیگر است. در محیط پویا نیاز به حفظ تنوع امری ضروری است و تقسیم‌بندی فضای جستجو این تنوع را در طول اجرای الگوریتم حفظ می‌کند. مزیت دیگر الگوریتم کلونی مورچه سلولی استفاده از یک جستجوی محلی قوی است.

جدول ۱) پارامترهای تابع محک قله‌های متحرک

پارامتر	مقدار
تعداد قله‌ها	۱۰-۲۰۰
تعداد ابعاد	۵
محدوده ارتفاع قله‌ها	[۳۰ ۷۰]
ارتفاع استاندارد قله‌ها	۵۰
محدوده عرض قله‌ها	[۱ ۱۲]
محدوده فضای جستجو	[۰ ۱۰۰]
فرکانس تغییر	۱۰۰۰-۵۰۰۰
شکل قله‌ها	مخروطی
گام‌های تغییر	ثابت

دیگری واقع شود. اگر سلول جدید یکی از سلول‌های همسایه باشد، مورچه از حالت سلول فعلی حذف و به حالت سلول جدید اضافه می‌شود. اما اگر سلول در مکان دورتری باشد، با توجه به محدودیت محلی بودن قوانین اتوماتای سلولی، انتقال یک مرتبه و ناگهانی مورچه از یک سلول به سلول دیگر امکان‌پذیر نیست. برای رساندن مورچه به مقصد آن را بین سلول‌های میانی جابجا می‌کنیم. مورچه هنگام جابجایی غیرفعال می‌شود و در تکامل شرکت نمی‌کند.

به روزرسانی حافظه سلول‌ها: بهترین مکان یافت شده در سلول از زمان آخرین تغییر محیط با استفاده از مورچه‌های موجود در سلول بهنگام می‌شود. در صورتی که شایستگی مورچه‌ای بیش‌تر از بهترین مکان سلول باشد، جایگزین آن می‌شود.

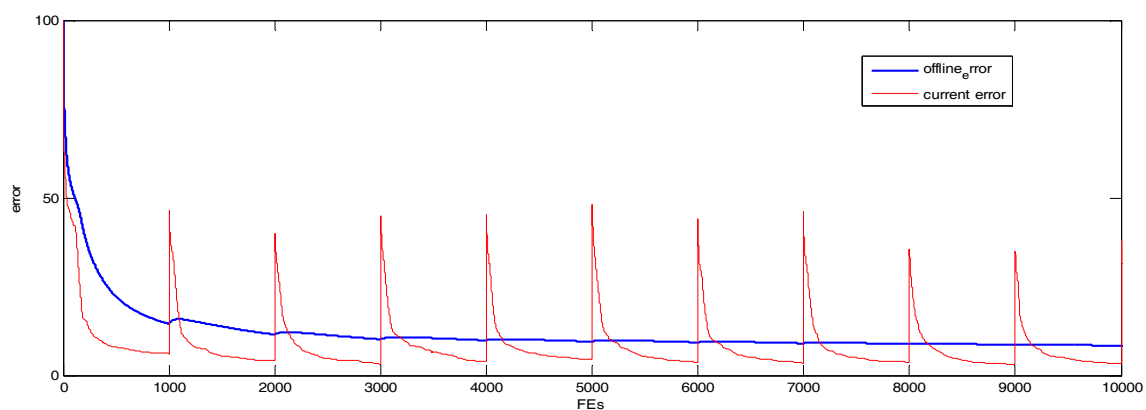
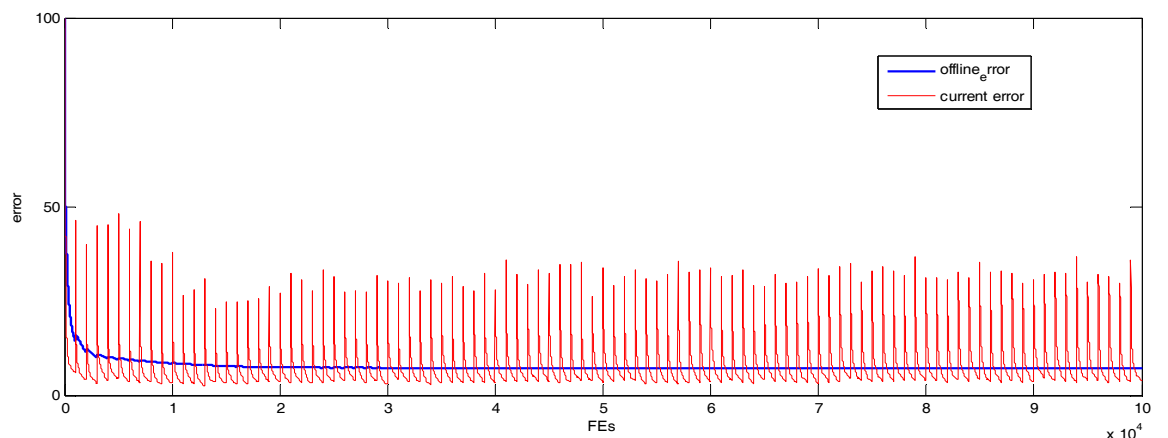
بهنگام سازی مکان مورچه‌ها: مورچه‌ها از الگوریتم کلونی مورچه پیشنهادی در بخش ۳ برای جابجایی در سلول‌ها و پیدا کردن بهینه استفاده می‌کنند. در هر تکرار از اتوماتا اگر تغییری در محیط رخ داده باشد، مورچه‌های سلول در یک تکرار بعد از تغییر با استفاده از جستجوی محلی مکان خود را بهنگام می‌کند. مورچه‌ها برای جستجوی محلی الگوریتم HJ را به کار می‌گیرند. پس از هر تغییر در محیط امکان تغییر مکان بهینه هست، بنابراین برای دنبال کردن بهینه‌ها از یک جستجوی محلی پس از هر تغییر استفاده می‌شود.

بروزرسانی فرومون سلول‌ها: بعد از ساخت راه‌حل در هر سلول، فرومون آن سلول با استفاده از مکان و شایستگی بهترین مورچه سلول بهنگام می‌شود. همانطور که در بخش ۳ اشاره شد. فرومون با استفاده از بهترین مورچه سلول بهنگام می‌شود.

بررسی تراکم مورچه‌های سلول: برای کنترل تراکم مورچه‌ها در الگوریتم پیشنهادی در هر بار بروزرسانی، هر سلول تعداد مورچه‌های فعال خود را با ظرفیت سلول مقایسه می‌کند، در صورتی که تعداد مورچه‌ها بیش‌تر باشد، از بین بدترین مورچه‌های فعال به تعداد ظرفیت اضافه سلول مورچه به تصادف انتخاب شده و برای حفظ تنوع کلونی آن‌ها را به سلول‌های که بصورت تصادفی انتخاب شده اند منتقل می‌شود. در صورتی که سلول جدید در مکانی غیر از همسایگی سلول فعلی باشد، مورچه تا رسیدن به سلول مقصد غیرفعال می‌شود. با حرکت تصادفی مورچه‌ها، قدرت اکتشاف الگوریتم جستجو افزایش می‌یابد.

۶- آزمایشات

برای ارزیابی کارایی محیط‌های پویا توابع محک متفاوتی ارائه شده است. تابع محک قله‌های متحرک^{۱۱} که رایج ترین و کاملترین آن‌هاست مورد استقبال فراوانی قرار گرفته است [۹, ۱۰]. این تابع شامل یک محیط چند بعدی با چندین قله است. هنگامی که تغییری در محیط



شکل ۳) خطای جاری برای فرکانس تغییر ۱۰۰۰. الف) برای ۱۰۰۰۰۰ ارزیابی، ب) برای ۱۰۰۰۰۰ ارزیابی تابع

جدول ۳) خطای آفلاین به همراه میانگین خطای استاندارد
الگوریتم‌های مختلف به ازای تعداد قله‌ای مختلف برای فرکانس تغییر
۵۰۰۰

نام الگوریتم			تعداد قله‌ها
Cellular ACO	Cellular PSO	mQSO	
0.33 ± 0.03	3.66 ± 0.22	0.53 ± 0.01	۱
1.36 ± 0.02	1.79 ± 0.12	1.05 ± 0.06	۵
1.32 ± 0.02	1.84 ± 0.08	1.31 ± 0.03	۱۰
1.87 ± 0.01	2.63 ± 0.11	1.69 ± 0.05	۲۰
2.16 ± 0.02	2.91 ± 0.10	1.78 ± 0.02	۳۰
2.24 ± 0.01	3.16 ± 0.11	1.86 ± 0.02	۴۰
2.28 ± 0.02	3.23 ± 0.11	1.95 ± 0.02	۵۰
2.24 ± 0.01	3.43 ± 0.10	1.95 ± 0.01	۱۰۰
2.05 ± 0.00	3.38 ± 0.09	1.90 ± 0.01	۲۰۰

جدول ۲) خطای آفلاین به همراه میانگین خطای استاندارد
الگوریتم‌های مختلف به ازای تعداد قله‌ای مختلف برای فرکانس تغییر
۲۵۰۰

نام الگوریتم			تعداد قله‌ها
Cellular ACO	Cellular PSO	mQSO	
0.38 ± 0.01	4.91 ± 0.28	7.79 ± 0.72	۱
3.53 ± 0.36	2.95 ± 0.20	3.53 ± 0.18	۵
4.51 ± 0.37	2.97 ± 0.15	3.20 ± 0.14	۱۰
4.14 ± 0.12	3.51 ± 0.14	3.83 ± 0.12	۲۰
4.05 ± 0.12	3.87 ± 0.12	4.03 ± 0.12	۳۰
3.67 ± 0.06	3.89 ± 0.10	3.90 ± 0.11	۴۰
3.49 ± 0.05	4.16 ± 0.15	3.95 ± 0.10	۵۰
3.10 ± 0.02	4.18 ± 0.11	3.81 ± 0.10	۱۰۰
2.58 ± 0.01	4.04 ± 0.09	3.66 ± 0.07	۲۰۰

salesman problem," IEEE transactions on evolutionary computation, vol. 1, pp. 53-66, 1997.

- [۸] R. Hooke and T. A. Jeeves, "Direct Search Solution of Numerical and Statistical Problems" J. ACM, vol. 8, pp. 212--229, 1961.
- [۹] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, p. 1882 Vol. 3, 1999.
- [۱۰] J. Branke, "evolutionary algorithms for dynamic optimization problems-a survey," Insitute AIFB 1999.
- [۱۱] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," Evolutionary Computation, IEEE Transactions on, vol. 10, pp. 459-472, 2006.

شکل ۳ خطای جاری را برای فرکانس تغییر ۱۰۰۰، در محیط ۵۰ قله‌ای نشان می‌دهد. محور افقی تعداد ارزیابی‌های تابع شایستگی و محور عمودی بیانگر خطا است. خط قرمز کمترین خطا در هر ارزیابی و خط آبی میانگین خطا یا همان خطای آفلاین است. نمودار الف خطا را تا برقراری شرط توقف و نمودار ب ۱۰ تغییر اول را نشان می‌دهد. مشاهده می‌شود که بعد از روی دادن تغییر در محیط الگوریتم کلونی مورچه سلولی به خوبی آن را دنبال می‌کند. علت این امر اعمال جستجوی محلی بعد از شناسایی تغییر در محیط است.

۷- نتیجه

در این مقاله یک الگوریتم کلونی مورچه مبتنی بر اتوماتای سلولی به نام الگوریتم کلونی مورچه سلولی برای حل مسائل بهینه‌سازی پویا با متغیرهای پیوسته ارائه شد. الگوریتم پیشنهادی برای حفظ تنوع و استفاده از نتایج پیشین جستجو، تقسیم‌بندی فضای جستجو با استفاده از اتوماتای سلولی، را به کار می‌گیرد. جستجو در هر سلول و همسایگانش به صورت موازی و جداگانه، توسط الگوریتم کلونی مورچه‌ی پیوسته انجام می‌شود. استفاده از اتوماتای سلولی موجب حفظ تنوع جمعیت مورچه‌ها می‌شود.

پس از شرح کامل روند پیشنهادی، آزمایشاتی در محیط پویای قله‌های متحرک جهت ارزیابی و مقایسه کارایی مدل پیشنهادی با سایر روش‌های تکاملی انجام شد. در این آزمایشات نشان داده شد که روش پیشنهادی از کارایی نسبتاً خوبی نسبت به سایر روش‌ها برخوردار است.

مراجع

- [۱] M. Dorigo, V. Maniezzo, and A. Colormi, "The Ant System: Optimization by a colony of cooperating agents" IEEE Transaction on Systems, man, and cybernetics-part b, vol. 26, pp. 29-41, 1996.
- [۲] J. Drezo and P. Siarry, "An ant colony algorithm aimed at dynamic continuous optimization," Applied Mathematics and Computation, vol. 181, pp. 457-467, 2006.
- [۳] W. Tfaili and P. Siarry, "A new charged ant colony algorithm for continuous dynamic optimization," Applied Mathematics and Computation, vol. 197, pp. 604-613, 2008.
- [۴] A. Hashemi, M. Meybodi, Z. Cai, Z. Li, Z. Kang, and Y. Liu, "Cellular PSO: A PSO for Dynamic Environments Advances in Computation and Intelligence." vol. 5821: Springer Berlin / Heidelberg, pp. 422-433, 2009.
- [۵] A. B. Hashemi and M. R. Meybodi, "A multi-role cellular PSO for dynamic environments," in Computer Conference, 2009. CSICC 2009. 14th International CSI, pp. 412-417, 2009.
- [۶] A. W. B. John Von Neumann, Theory of Self-Reproducing Automata: University of Illinois Press 1966.
- [۷] M. Dorigo and L. M. Gambardella, "Ant Colony System: A cooperative learning approach to the traveling

تقدیر و تشکر

قسمتی از این کار تحقیقاتی با حمایت مالی مرکز تحقیقات مخابرات ایران انجام گرفته است.

زیر نویس‌ها

- ¹ Ant Colony Optimization
² Cellular Ant Colony Optimization
³ Cellular Automata
⁴ Von Neumann
⁵ Ulam
⁶ moor
⁷ Box-Muller
⁸ Pattern Search
⁹ Hook-Jeevs
¹⁰ Moving peak benchmark
¹¹ Offline error