علوم و مهندسی کامپیوتر
نشریه علمی پژوهشی انجمن کامپیوتر ایران
مجلد ۷، شماره ۱ و ۳ (الف)، ۱۳۸۸
صفحات ۵۷-۶۷
مقاله عادی

طراحی قانون محلی خودکاره‌های سلولی به وسیله الگوریتم‌های تکاملی

سمانه‌السادات موسوی حمید بیگی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

چکیده

خودکاره سلولی مدل ریاضی برای سامانه‌های فیزیکی است که از اجزای ساده و همگونی تشکیل شده‌اند. اجزای ساده این مدل از طریق تعاملات محلی، می‌توانند رفتار سراسری پیچیده‌ای از خود نشان دهند. در زمینه خودکاره‌های سلولی، دو نوع مسأله وجود دارد: مسأله پیش‌رو و مسأله پس‌رو. در مسأله پیش‌رو، قانون خودکاره سلولی معلوم است و هدف بررسی رفتار قانون و ویژگی‌های خودکاره سلولی براساس مقدار اولیه سلول‌ها می‌باشد. اما در مسأله پس‌رو، قانون خودکاره سلولی مشخص نیست و تنها توصیفی از برخی ویژگی‌های خودکاره سلولی داده شده است و هدف پیدا کردن یک قانون یا مجموعه‌ای از قانون‌هاست که ویژگی‌های توصیف شده را داشته باشند. مسأله پس‌رو یک مسأله ذاتاً مشکل است و به همین دلیل از الگوریتم‌های تکاملی، که گونه‌ای از الگوریتم‌های مکاشفه‌ای هستند، برای حل این مسأله استفاده شده است. در مسأله پس رو هم فضای قانون‌ها و هم فضای پیکربندی‌های اولیه‌ی خودکاره سلولی دارای نرخ رشد نمایی هستند، بنابراین در الگوریتم‌های تکاملی که برای حل مساله پس‌رو ارائه شده‌اند امکان ارزیابی دقیق قانون‌ها با استفاده از تمام پیکربندی‌های اولیه در زمان چند جمله‌ای وجود ندارد. در این مقاله، به کمک الگوریتم‌های تکاملی رویکرد جدیدی برای حل مسأله پس‌رو ارایه می‌شود. الگوریتم از سه گام اصلی تشکیل شده است. در گام نخست الگوریتم طراحی شده با رویکرد پیشنهادی، فضای پیکربندی‌های اولیه خوشه‌بندی شده و سپس در گام دوم به کمک الگوریتم‌های تکاملی به جستجوی بهترین قانون برای هر خوشه پرداخته می‌شود و در پایان، بهترین قانون با جستجو در فضای ایجاد شده توسط قانون‌های گام دوم پیدا می‌شود. نتایج آزمایش‌ها نشان می‌دهند که الگوریتم پیشنهادی برای مسأله‌های استاندارد ارزیابی باعث افزایش کارایی الگوریتم می‌شود.

کلمات کلیدی: خودکاره سلولی، الگوریتم‌های تکاملی، الگوریتم ژنتیک، خودکاره سلولی تکاملی، خوشه‌بندی.

۱- مقدمه

خودکاره سلولی[۱] در اواخر دهه ۱۹۴۰ توسط فون نیومن[۲] مطرح و پس از او توسط ریاضیدانی به نام اولام[۳] به عنوان مدلی برای بررسی رفتار سامانه‌های پیچیده پیشنهاد شد [۱، ۲]. خودکاره سلولی یک سامانه دینامیکی گسسته با ارتباطات محلی است و با استفاده از همین ارتباطات محلی، می‌تواند رفتار سراسری پیچیده‌ای از خود نشان دهد. توانایی خودکاره سلولی در مدل‌سازی انواع سامانه‌ها باعث استفاده روزافزون از این مدل در کاربردهای گسترده‌ای مانند شبیه‌سازی فرآیندهای فیزیکی (زلزله، آتش‌سوزی، سیل، بهمن، مراحل تکامل حیات)، شبیه‌سازی فرآیندهای اجتماعی (انتشار شایعه، نحوه گسترش شهرها)، شبیه‌سازی پدیده‌های بیولوژیکی (انتشار بیماری‌های واگیردار، نحوه گسترش سرطان)، شبیه‌سازی آزمایشات شیمیایی (انتشار گازها، خوردگی فلزات، رشد بلور) و کاربردهای متنوع دیگری مانند پردازش تصویر و تولید دنباله‌های اعداد تصادفی در رمزنگاری شده است [۳، ۴].

در زمینه مطالعات مربوط به خودکاره‌های سلولی دو نوع مسأله کلی وجود دارد: مسأله پیش‌رو[۴] و مسأله پس‌رو[۵] [۵]. در مسأله پیش‌رو هدف مطالعه رفتار و ویژگی‌های قانون خودکاره سلولی است. از دیدگاه ریاضی مسأله پیش‌رو معمولا شامل پیدا کردن مقادیری است که قابل محاسبه از جدول قانون باشند و رفتار قانون را با شروع از یک پیکربندی اولیه تصادفی مشخص کنند. مطالعه رفتار قوانین و دسته‌بندی آن‌ها، مشابه پژوهش‌های انجام شده توسط ولفرام[۶] می‌باشد [۲، ۵]. در مقابل در مسأله پس‌رو توصیفی از برخی ویژگی‌های قانون داده شده است و به دنبال پیدا کردن یک قانون یا مجموعه‌ای از قانون‌ها هستیم که

ویژگی‌های توصیف شده را داشته باشند. مسأله پس‌رو در کشف قانون حاکم بر پدیده‌ها، کاربرد بسیار دارد. مسأله پس‌رو یک مسأله ذاتا مشکل است و در زمان چند جمله‌ای قابل حل نمی‌باشد. به همین دلیل به منظور حل آن از الگوریتم‌های تکاملی استفاده شده است. در پژوهش‌های انجام شده در رابطه با جستجوی قانون خودکاره سلولی، استفاده از الگوریتم‌های تکاملی به روش‌های مختلفی صورت گرفته است. در این پژوهش‌ها از الگوریتم ژنتیک[۷] [۶، ۷]، برنامه‌نویسی ژنتیک[۸] [۸، ۹]، الگوریتم‌های هم‌تکاملی[۹] [۱۰] و الگوریتم‌های تکاملی چند هدفه[۱۰] [۱۱] برای جستجوی قانون خودکاره‌های سلولی استفاده شده است. در مساله پس‌رو هم فضای قانون و هم فضای پیکربندی‌های اولیه خودکاره سلولی دارای نرخ رشد نمایی هستند بنابراین در الگوریتم های تکاملی ارائه شده برای حل مساله پس‌رو امکان ارزیابی دقیق قانون‌ها با استفاده از تمامی پیکربندی‌های اولیه در زمان چند جمله‌ای وجود ندارد.

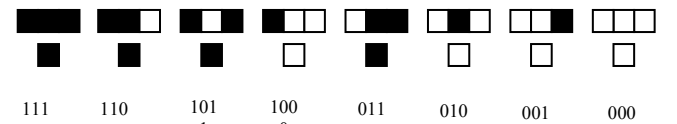
در این مقاله، رویکرد جدیدی برای ارزیابی دقیق‌تر قانون‌ها در مساله پس‌رو توسط الگوریتم‌های تکاملی ارائه می‌شود. الگوریتم طراحی شده با رویکرد پیشنهادی از سه گام اصلی تشکیل شده است. در گام نخست، فضای پیکربندی‌های اولیه خوشه بندی شده و پیکربندی‌های مشابه از هم جدا می‌شوند و ارزیابی با تعدادکمتری پیکربندی اولیه انجام می شود. در گام دوم، الگوریتم تکاملی برای جستجوی قانون بکار می رود. در پایان بهترین قانون از مجموعه قوانین پیدا شده، استخراج و استفاده می شود. برای ارزیابی کارایی الگوریتم پیشنهادی از شبیه‌سازی کامپیوتری روی چندین مساله استاندارد که در این زمینه وجود دارد استفاده شده است. نتایج شبیه‌سازی‌ها نشان می‌دهند که الگوریتم پیشنهادی در تمام اجراها نتایج بهتری نسبت به الگوریتم‌های مشابه تولید می‌کند.

ادامه این مقاله به صورت زیر سازماندهی شده است. در بخش ۲، مقدماتی در زمینه خودکاره‌های سلولی بیان شده است. بخش ۳ شامل مباحث مربوط به خودکاره‌های سلولی تکاملی[۱۱] و روش‌های مختلف استفاده از الگوریتم‌های تکاملی برای جستجو و طراحی قانون خودکاره‌های سلولی است. الگوریتم پیشنهادی در بخش ۴ ارایه خواهد شد. در بخش ۵ نخست مسائل مورد بررسی به عنوان محک ارزیابی ارائه و سپس کارایی الگوریتم ارزیابی می شود و در پایان کارایی الگوریتم پیشنهادی با الگوریتم‌های مشابه خودکاره سلولی تکاملی مقایسه می‌شود. در بخش ۶ نیز نتیجه گیری مقاله آمده است.

۲- خودکاره‌های سلولی

خودکاره‌های سلولی مدل ریاضی برای سامانه‌هایی هستند که از تعداد بسیار زیادی مولفه ساده و یکسان با تعاملات محلی تشکیل شده‌اند. یک خودکاره سلولی با چهار ویژگی مشخص می‌شود: شبکه‌ای که سلول‌ها در آن قرار می‌گیرند، قانون محلی، مجموعه حالت ها و همسایگی سلول. به صورت غیر رسمی یک خودکاره سلولی d بعدی، شامل یک شبکه d بعدی از سلول‌هاست. هر سلول حالتی دارد که از یک مجموعه متناهی حالت‌ها انتخاب می‌شود. سلول‌ها حالت‌شان را به صورت هم‌زمان براساس یک قانون محلی، به روز رسانی می‌کنند. حالت جدید هر سلول بستگی به حالت پیشین یک مجموعه از سلول‌ها دارد. این مجموعه از سلول‌ها، همسایگی سلول نامیده می‌شود. حالت تمام سلول‌های شبکه در یک زمان، یک پیکربندی[۱۲] را برای خودکاره سلولی توصیف می‌کند. پیکربندی نشان دهنده حالت کل شبکه است. قانون و پیکربندی اولیه خودکاره سلولی، تکامل خودکاره سلولی را مشخص می‌کنند. تکامل بیان می‌نماید که چگونه پیکربندی‌ها با شروع از پیکربندی اولیه در گام‌های متوالی براساس قانون تغییر می‌کنند [۱۲].

قانون خودکاره سلولی را می‌توان به صورت یک جدول نمایش داد. در جدول قانون تمام حالات ممکن برای همسایگی به همراه حالت جدید سلول لیست شده است. در شکل ۱ جدول قانون یک خودکاره سلولی که در آن اندازه همسایگی برای هر سلول ۳ است را مشاهده می‌کنید. در این شکل هر مربع نشان‌دهنده یک سلول است. به گونه‌ای که مربع‌های سیاه سلول‌های با حالت ۱ و مربع‌های سفید سلول‌های با حالت ۰ را نشان می‌دهند. در شکل ۱ جدول قانون هم با استفاده از مربع‌های سیاه و سفید و هم به صورت رشته‌های ۰ و ۱ نمایش داده شده است. طبق این قانون، سلولی که در گام فعلی تعداد یک‌ها (صفرها) در همسایگی‌اش بیشتر از تعداد صفرها (یک‌ها) باشد، حالت جدید آن در گام بعد، یک (صفر) خواهد بود.



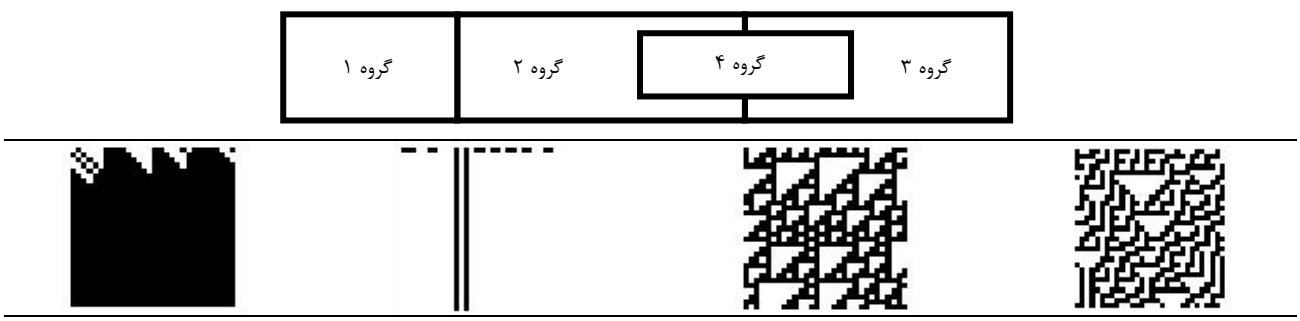
111	110	101	100	011	010	001	000
1	1	1	0	1	0	0	0

شکل ۱- جدول قانون یک خودکاره سلولی

۲-۱- دسته‌بندی خودکاره‌های سلولی

تلاش‌های مختلفی برای دسته‌بندی قوانین حاکم بر خودکاره‌های سلولی انجام گرفته است. یکی از مهم‌ترین دسته‌بندی‌ها، توسط ولفرام انجام شده است. ولفرام برای بررسی رفتار هر قانون، رفتار دینامیکی آن را در طی یک دوره زمانی طولانی مورد مطالعه قرار داد و همان‌طور که در شکل ۲ نشان داده شده است، قوانین خودکاره‌های سلولی را براساس رفتارشان به چهار گروه مجزا تقسیم کرده است [۱۳]:

- گروه یک: رفتار این گروه از قوانین بسیار ساده و محدود است. خودکاره سلولی با شروع از هر پیکربندی اولیه‌ای به یک پیکربندی ثابت، مانند پیکربندی تمام صفر یا تمام یک، می‌رسد.



شکل ۲- نمودار مکان ـ زمان چند نمونه خودکاره سلولی. در این شکل رفتار یک قانون از هر گروه را بر روی یک پیکربندی اولیه مشاهده می‌کنید

- گروه دو: قوانین این گروه مانند یک فیلتر، پیکربندی‌های ساده، جدا و پریودیک می‌سازند. تقریبا تمام پیکربندی‌های اولیه بعد از تعدادی گام به یک پیکربندی ثابت یا یک مجموعه متناوب از پیکربندی‌ها می‌رسند که این مجموعه به پیکربندی اولیه بستگی دارد.
- گروه سه: قوانین این گروه، پیکربندی‌های غیر پریودیک و غیر قابل پیش‌بینی ایجاد می‌کنند و رفتارهای نامنظم و پیچیده‌ای دارند.
- گروه چهار: هر قانونی که در سه گروه بالا نباشد، در این گروه جای دارد. قوانین این گروه رفتار پیچیده‌ای دارند و با اعمال قوانین این دسته روی برخی پیکربندی‌های اولیه، ساختارهای پیچیده‌ای با عمر نسبتا طولانی تولید می‌شود.

۳- خودکاره سلولی تکاملی

خودکاره سلولی تکاملی، به خودکاره سلولی گفته می‌شود که قانون حاکم بر آن با استفاده از الگوریتم‌های تکاملی به دست آمده باشد. الگوریتم‌های تکاملی، الگوریتم‌های بهینه‌سازی فرامکاشفه‌ای[۱۳] هستند که در جمعیتی از راه‌حل‌های ممکن به جستجوی راه‌حل بهینه می‌پردازند. این الگوریتم‌ها به صورت تکراری و به منظور یافتن بهترین راه‌حل از مکانیزم‌هایی استفاده می‌کنند که الهام گرفته از مکانیزم‌های تکامل موجودات زنده در طبیعت هستند.

این مکانیزم‌ها عبارتند از: جهش، ترکیب، انتخاب طبیعی و بقای بهترین‌ها. در تمام الگوریتم‌های تکاملی ابتدا تعدادی راه‌حل تصادفی، موسوم به جمعیتی از افراد، تولید می‌شود. سپس افراد جمعیت براساس تابع از پیش‌تعریف شده‌ای، موسوم به تابع برازش، ارزیابی می‌شوند. پس از آن از بین افراد جمعیت، گروهی به صورت تصادفی به عنوان والدین برای ایجاد یک جمعیت جدید انتخاب می‌شوند. البته برای این انتخاب از استراتژی‌های گوناگونی می‌توان استفاده کرد. پس از مشخص شدن والدین، با اعمال عملگرهای ترکیب و جهش روی جمعیت والدین، یک جمعیت جدید از افراد تولید می‌شود. چرخه ارزیابی، انتخاب و تولید جمعیت جدید تا رسیدن به یک معیار خاتمه از پیش تعیین‌شده ادامه پیدا می‌کند. به هر چرخه یک نسل گفته می‌شود و تکرار چرخه باعث تکامل جمعیت در نسل‌های متوالی می‌شود.

از خودکاره‌های سلولی تکاملی برای کشف قانون حاکم بر پدیده‌ها می‌توان استفاده کرد. برای مطالعه رفتار سامانه‌ای که قانون حاکم بر آن را نمی‌دانیم و نیاز به بازسازی رفتار سامانه داریم، می‌توانیم از مدل‌سازی سامانه به وسیله خودکاره سلولی استفاده کنیم و پس از آن با استفاده از الگوریتم‌های تکاملی قانون حاکم و یا تقریبی از قانون حاکم بر محیط را بیابیم. به این منظور، از خودکاره‌های سلولی تکاملی در مدل‌سازی مناطق شهری [۱۴]. تشخیص ارقام دست نویس [۱۵] و تولید دنباله تصادفی برای الگوریتم‌های رمزنگاری [۴] استفاده شده است. استفاده از الگوریتم‌های تکاملی برای طراحی قانون‌های خودکاره‌های سلولی به روش‌های مختلفی صورت گرفته است، که در ادامه این بخش چند نمونه از آن‌ها بررسی می‌شوند.

۳-۱- جستجوی قانون با استفاده از الگوریتم ژنتیک

نخستین پژوهش‌ها در رابطه با تکامل قانون‌های خودکاره‌های سلولی با استفاده از الگوریتم ژنتیک توسط پاکارد و همکارانش انجام شده است [۱۶]. اما مهم‌ترین پژوهش‌ها در این زمینه توسط میشل و همکارانش انجام شده است. به طور عمده این پژوهش‌ها قابلیت استفاده از خودکاره سلولی تکاملی برای حل مسأله رده‌بندی براساس چگالی[۱۴] را نشان می‌دهند [۶، ۷، ۱۷-۲۰].

در الگوریتم ژنتیک، در ابتدا جمعیتی از قوانین به صورت تصادفی تولید می‌شود. از آن‌جا که تعداد پیکربندی‌های اولیه برای یک خودکاره سلولی k حالته با N سلول k^N است، ارزیابی یک قانون با استفاده از تمام پیکربندی‌های اولیه در زمان چند جمله‌ای امکان‌پذیر نیست. بنابراین هر قانون با استفاده از یک جمعیت تصادفی از پیکربندی‌های اولیه ارزیابی می‌شود. به این صورت که هر قانون روی هر پیکربندی به اندازه تعداد محدودی گام یا رسیدن به یک پیکربندی ثابت اعمال می‌شود. سپس تعیین می‌شود که آیا پیکربندی نهایی درست است یا نه. میزان برازش هر قانون برابر با کسری از پیکربندی‌هاست که به پیکربندی نهایی صحیح رسیده‌اند.

در هر نسل، یک مجموعه جدید از پیکربندی‌های تصادفی ایجاد می‌شود و برازش هر قانون با استفاده از این پیکربندی‌های تصادفی محاسبه می‌شود. قانون‌ها با توجه به برازش‌شان مرتب می‌شوند. تعدادی از بهترین قوانین، بدون هیچ‌گونه تغییری به نسل بعد منتقل می‌شوند و بقیه قوانین از روی بهترین قانون‌ها و با استفاده از ترکیب و جهش ساخته می‌شوند.

۳-۱-۱- موانع موجود

در این بخش به بررسی عواملی که مانع از پیدا کردن قوانین بهتر با استفاده از الگوریتم ژنتیک می‌شوند، می‌پردازیم. در مورد این عوامل بحث مبسوطی در مرجع [۱۹] با محوریت مسأله رده‌بندی براساس چگالی انجام و در مواردی راه کارهایی نیز برای مقابله با این موانع ارایه شده است.

ماهیت تصادفی تابع برازش: در بسیاری از پژوهش‌های انجام شده، برای محاسبه تابع برازش نمونه کوچکی از پیکربندی‌های اولیه به صورت تصادفی انتخاب می‌شود. در حالیکه تعداد تمام حالت‌های ممکن برای پیکربندی‌های اولیه یک خودکاره سلولی دو حالته با N سلول، 2^N است. هیچ توجیهی برای چنین انتخابی به جز زمان و امکانات محدود برای محاسبه وجود ندارد. بنابراین با توجه به وابستگی تابع برازش به پیکربندی‌های اولیه، اگر پیکربندی‌های اولیه، پیکربندی‌های ساده و ضعیفی باشند، مقدار حاصل از تابع برازش، مقدار قابل اطمینانی نخواهد بود. خطای نمونه که ناشی از انتخاب تصادفی پیکربندی‌های اولیه است، معمولا کارایی فرآیند تکامل را کاهش می‌دهد. به منظور گریز از خطای نمونه، در مراجع [۲۱، ۲۲] روشی پیشنهاد شده است که در آن هم قوانین و هم پیکربندی‌ها به صورت هم‌زمان تکامل پیدا می‌کنند.

نحوه نمایش ترتیبی بیت‌ها در قانون: در بیشتر پژوهش‌های انجام شده قوانین به صورت رشته‌های بیتی نمایش داده شده‌اند. این نحوه نمایش تأثیر منفی‌اش را در هنگام ترکیب قوانین نسل قبل و تولید قوانین جدید به جا می‌گذارد. زیرا احتمال تولید قوانینی با ساختارهای خاص از ترکیب قوانین پدر کم است. در این‌گونه موارد، باید به جهش امیدوار بود. البته راه‌حل دیگر این مشکل تغییر نحوه نمایش قانون‌هاست که در بخش جستجوی قانون با استفاده از برنامه‌نویسی ژنتیک به آن اشاره خواهد شد.

کاهش گوناگونی در جمعیت قانون‌ها: با پیشرفت الگوریتم ژنتیک، به سرعت قوانین با برازش بالا به یک سو می‌روند. همگرایی سریع در نسل‌های نخست بسیار مفید است، اما متأسفانه همگرایی سریع باعث می‌شود که در نسل‌های بعد، فضای قوانین در حال جستجو به شدت محدود شود و احتمال گیر افتادن در بیشینه‌های محلی[۱۵] افزایش یابد. در حقیقت کاهش گوناگونی در جمعیت باعث همگرایی زودرس می‌شود.

۳-۲- جستجوی قانون با استفاده از برنامه‌نویسی ژنتیک

در مراجع [۸، ۹] برای طراحی قانون خودکاره‌های سلولی از برنامه‌نویسی ژنتیک استفاده شده است. برنامه‌نویسی ژنتیک نوعی الگوریتم تکاملی است که تلاش می‌کند با استفاده از مکانیزم‌های کلی بیان شده برای الگوریتم‌های تکاملی، به تولید برنامه‌های کامپیوتری بپردازد که بتوانند وظایف تعریف شده را انجام دهند. در برنامه‌نویسی ژنتیک، افراد جمعیت به صورت رشته‌های بیتی نیستند. بلکه در این‌جا هر فرد یک برنامه کامپیوتری است. در برنامه‌نویسی ژنتیک، هر برنامه کامپیوتری به صورت یک درخت نمایش داده می‌شود. هر گره میانی درخت یک تابع یا عملگر ساده است و هر گره برگ یک آرگومان است. نمایش درختی برنامه‌ها یا قوانین باعث سادگی ارزیابی آنها می‌شود.

همان‌طور که در زیر بخش‌های قبل به برخی موانع موجود برای تکامل قوانین خودکاره‌های سلولی اشاره شد، نمایش استاندارد قوانین به صورت رشته‌های بیتی که در بسیاری از پژوهش‌های گذشته برای تکامل قوانین خودکاره‌های سلولی با استفاده از الگوریتم ژنتیک استفاده شده است، ممکن است مانع تکامل شود [۱۹]. بنابراین یک نمایش سطح بالاتر، به عنوان مثال نمایش قانون به صورت زوج‌های شرط- کنش، ممکن است برای فرآیند تکامل مفید واقع شود. به این ترتیب نمایش درختی به کار گرفته شده در برنامه نویسی ژنتیک مناسب‌تر است [۸]. در این روش از توابعی مانند AND, NAND, OR, NOR, NOT, IF, XOR استفاده شده است و قانون خودکاره سلولی ترکیبی از این توابع است. عملوندهای این توابع حالت سلول مرکزی و حالت‌های همسایگان چپ و راست آن هستند.

۳-۳- جستجوی قانون با استفاده از الگوریتم‌های هم‌تکاملی

در الگوریتم‌های تکاملی‌ای که مقدار برازش جمعیت با استفاده از یک جمعیت دیگر سنجیده می‌شود، می‌توان از الگوریتم‌های هم‌تکاملی برای تکامل هم‌زمان دو جمعیت استفاده کرد. به طور مثال در طراحی قانون برای خودکاره‌های سلولی، میزان برازش هر قانون با استفاده از جمعیتی از پیکربندی‌های تصادفی محاسبه می‌شود. همچنین در طراحی قانون برای خودکاره‌های سلولی، خطای حاصل از انتخاب تصادفی پیکربندی‌های اولیه، معمولا کارایی فرآیند تکامل را کاهش می‌دهد. بنابراین برای این مسأله می‌توانیم از الگوریتم‌های هم‌تکاملی بهره ببریم.

در الگوریتم‌های هم‌تکاملی، در ابتدا برای قوانین و پیکربندی‌ها به صورت تصادفی دو جمعیت اولیه تولید می‌شود. سپس در هر نسل، جمعیت قوانین و پیکربندی‌ها براساس یکدیگر ارزیابی می‌شوند و جمعیت قوانین و پیکربندی‌ها، در نسل‌های مختلف به صورت هم‌زمان تکامل می‌یابند. تکامل دو جمعیت معمولا حالت رقابتی دارد.

به عنوان مثال در طراحی قانون برای خودکاره‌های سلولی، بهترین قانونی است که تعداد پیکربندی‌های بیشتری را به حالت نهایی صحیح رسانده باشد و در مقابل بهترین و چالش برانگیزترین پیکربندی، پیکربندی است که تعداد قوانین کم‌تری توانسته باشند آن را به حالت نهایی صحیح برسانند. این توابع برازش در عبارت (۱) آمده است. در این عبارت $f(R_i)$ تابع برازش برای قانون شماره i، $f(IC_j)$ تابع برازش برای پیکربندی اولیه شماره j، n_R تعداد قانون‌ها و n_{IC} تعداد پیکربندی‌های اولیه است. تابع covered در صورتی که قانون i پیکربندی j را به حالت نهایی صحیح برساند، مقدار یک و در غیر این صورت مقدار صفر را برمی‌گرداند [۲۱، ۱۰].

$$f(R_i) = \sum_{j=1}^{n_{IC}} covered(R_i, IC_j) \quad (۱)$$

$$f(IC_j) = \sum_{i=1}^{n_R} \left(1 - covered(R_i, IC_j)\right)$$

یکی از مهم‌ترین مشکلات مورد بحث درالگوریتم‌های هم‌تکاملی این است که در برخی موارد دو جمعیت دارای رفتار دوره‌ای می‌شوند و پیشرفتی حاصل نمی‌شود. در رفتار دوره‌ای حالتی پیش می‌آید که در آن تغییر دو جمعیت در یک فضای محدود حالت نوسانی به خود می‌گیرد. در مرجع [۲۱] برای حل مشکل رفتار دوره‌ای دو جمعیت از وزن‌دهی به افراد جمعیت استفاده شده است. تابع برازشی که ایده وزن‌دهی را شامل می‌شود، در عبارت (۲) نشان داده شده است که $weightIC_j$ نشان دهنده وزن پیکربندی اولیه شماره j و $weightR_i$ نشان دهنده وزن قانون شماره i می‌باشند.

$$f(R_i) = \sum_{j=1}^{n_{IC}} \left(weightIC_j \times covered(R_i, IC_j)\right)$$
$$weightIC_j = \frac{1}{\sum_{k=1}^{n_R} covered(R_k, IC_j)} \quad (۲)$$

$$f(IC_j) = \sum_{i=1}^{n_R} \left(weightR_i \times \left(1 - covered(R_i, IC_j)\right)\right)$$
$$weightR_i = \frac{1}{\sum_{k=1}^{n_{IC}} \left(1 - covered(R_i, IC_k)\right)}$$

در تابع برازش معرفی شده در عبارت (۲) در محاسبه میزان برازش برای هر قانون، وزن تمام پیکربندی‌ها یکسان نیست. بلکه وزن هر پیکربندی یک رابطه معکوس با تعداد قانون‌هایی دارد که آن پیکربندی را به حالت نهایی صحیح رسانده‌اند. اگر تعداد قانون‌هایی که آن پیکربندی را به حالت نهایی صحیح رسانده‌اند، زیاد باشد، آن پیکربندی به عنوان یک پیکربندی ضعیف یا معمولی وزن کمتری در محاسبه میزان برازش قانون پیدا می‌کند و اگر تعداد قانون‌هایی که آن پیکربندی را به حالت نهایی صحیح رسانده‌اند، کم باشد، آن پیکربندی به عنوان یک پیکربندی قوی وزن بیشتری در محاسبه میزان برازش قانون پیدا می‌کند. در مورد محاسبه میزان برازش پیکربندی‌ها نیز به همین ترتیب عمل می‌شود. به این ترتیب دیگر رفتار دوره‌ای مشاهده نمی‌شود.

۳-۴- جستجوی قانون با استفاده از الگوریتم‌های تکاملی چند هدفه

در برخی از پژوهش‌هایی که در زمینه طراحی قانون برای خودکاره‌های سلولی انجام شده است، سعی شده است الگوریتم جستجو به فضایی که پیش‌بینی می‌شود شامل قوانین مورد نظر است، هدایت شود. به منظور پیش‌بینی فضای شامل قوانین برای هر مسأله، نخست با استفاده از روش‌های دیگر از قبیل جستجوی قانون با استفاده از الگوریتم ژنتیک، قوانینی برای مسأله مورد نظر پیدا می‌کنند و سپس با مطالعه جدول قوانین کشف شده، ویژگی‌های مشترک قوانین را استخراج می‌کنند و در پایان، از این اطلاعات برای هدایت الگوریتم جستجو استفاده می‌کنند.

به این منظور، پارامترهایی مانند چیرگی همسایگی[۱۶]، حساسیت[۱۷]، انتشار فعالیت[۱۸] و غیره پیشنهاد شده است که این پارامترها مستقیما از روی جدول قانون محاسبه می‌شوند [۲۳] و سپس در الگوریتم ژنتیک، جستجو به سمت قوانینی هدایت می‌شود که در محدوده مقداری تعیین شده برای این پارامترها باشند.

به این ترتیب روش مکاشفه‌ای بر مبنای این پارامترها از دو جنبه با الگوریتم ژنتیک ترکیب شده است: ۱) در محاسبه تابع برازش، ۲) در عملگرهای ژنتیک. تابع برازش در روش مکاشفه‌ای بر مبنای پارامترهای یاد شده، در عبارت (۳) آمده است. تابع برازش برابر است با مجموع وزن‌دار F_{IC} که نتیجه اعمال قانون روی پیکربندی‌های اولیه است و F_p که از روش مکاشفه‌ای بر مبنای پارامترها به دست می‌آید. مقدار تابع F_{IC} برابر است با کسری از پیکربندی‌های اولیه که توسط قانون به پیکربندی نهایی صحیح رسیده‌اند. مقدار تابع F_p که تابع پارامتر نامیده می‌شود، با توجه به محدوده تعیین شده برای هر پارامتر محاسبه می‌شود. پارامتر ρ نیز میزان اثرگذاری تابع پارامتر را مشخص می‌کند و وزن مکاشفه نامیده می‌شود. تأثیر وزن مکاشفه در مرجع [۲۴] بررسی شده است.

$$FitnessFunction = F_{IC} + \rho \times F_p \qquad (۳)$$

استفاده از روش مکاشفه‌ای بر مبنای پارامترها در عملگرهای ژنتیک به این صورت است که دو قانون از نقاط مختلف با هم ترکیب می‌شوند و سپس دو قانونی که بهترین از لحاظ برآورده کردن تابع پارامترها باشند، انتخاب می‌شوند. به همین صورت در مورد جهش، یک قانون چند بار در معرض جهش قرار می‌گیرد و قانونی که بهترین از لحاظ برآورده کردن تابع پارامترها باشد، انتخاب می‌شود [۲۵]. در مراجع [۱۱، ۲۶-۲۸]، به جای استفاده از تابع برازش عبارت (۳) برای ارزیابی قانون، از الگوریتم‌های تکاملی چند هدفه استفاده شده است. به این ترتیب F_p و F_{IC} به عنوان دو هدف مستقل که الگوریتم جستجو باید به دنبالشان باشد، معرفی می‌شوند. با در نظر گرفتن تعداد متفاوتی از پارامترها به عنوان اهداف جداگانه، الگوریتم تکاملی چند هدفه را می‌توان با ۲، ۳ و ۴ هدف جداگانه اجرا کرد. در مرجع [۱۱] از الگوریتم تکاملی چند هدفه NSGA[۱۹] استفاده شده است. برای توضیحات بیشتر در مورد این الگوریتم تکاملی چند هدفه می‌توانید به مرجع [۲۹] مراجعه کنید.

۳-۵- استفاده از روش‌های کاهش بعد

در مرجع [۳۰] با استفاده از روش‌های کاهش بعد سعی شده است نمونه‌های انتخابی برای پیکربندی‌های اولیه طوری انتخاب شوند که حداکثر اختلاف ممکن را با هم داشته باشند و در ضمن معرف کل فضای پیکربندی‌های ممکن نیز باشند. به این ترتیب محاسبه برازش یک قانون براساس یک مجموعه از پیکربندی‌های تصادفی که دارای ویژگی‌های کل جامعه است، منطقی‌تر به نظر می‌رسد. روش استفاده شده شامل تحلیل مولفه‌های اصلی و استفاده از خوشه‌بندی براساس مولفه‌های اصلی پیکربندی‌های اولیه است.

۴- الگوریتم پیشنهادی

بررسی پژوهش‌های گذشته این موضوع را روشن کرده است که تعداد زیاد پیکربندی‌های اولیه ممکن، مانع ارزیابی دقیق قوانین است. در برخی پژوهش‌های انجام شده، با استفاده از توزیع داده‌ها بین چندین پردازنده و بهره‌گیری از روش‌های پردازش موازی، قوانین روی تعداد زیادی از پیکربندی‌های اولیه، در حدود چند میلیون پیکربندی اولیه، ارزیابی شده‌اند و نتایج دقیق‌تری به دست آمده است. اما همین تعداد هم به مراتب بسیار کمتر از تعداد کل پیکربندی‌ها است. در الگوریتم پیشنهاد شده در این بخش، ابتدا پیکربندی‌های اولیه بر حسب اینکه شامل کدام همسایگی‌ها هستند، خوشه‌بندی می‌شوند. به این ترتیب پیکربندی‌های مشابه، یعنی پیکربندی‌هایی که همسایگی‌های یکسان بیشتری با هم دارند، در یک خوشه قرار می‌گیرند. سپس الگوریتم ژنتیک را به تعداد خوشه‌ها اجرا می‌کنیم و در هر اجرا، الگوریتم ژنتیک به جستجوی بهترین قانون برای یکی از خوشه‌ها می‌پردازد. سپس این بهترین قانون‌ها در یک الگوریتم ژنتیک جدید تکامل می‌یابند تا به بهترین قانون برای کل فضای پیکربندی‌های اولیه دست یابیم. در ادامه این بخش الگوریتم پیشنهادی شرح داده می شود.

شبه کد الگوریتم پیشنهادی در شکل ۳ آمده است. در این الگوریتم برای محاسبه شباهت بین پیکربندی‌ها از روشی که در بخش بعد توضیح داده خواهد شد، استفاده می‌شود. الگوریتم پیشنهادی که در آن از استراتژی تقسیم و حل کمک گرفته شده است، به این ترتیب است. در گام نخست:

- نخست I پیکربندی اولیه به صورت تصادفی تولید می‌شود.
- متناظر با هر پیکربندی تولید شده در گام قبل، مطابق با شبه کد شکل ۵ یک داده جدید تولید می‌شود.
- پیکربندی‌ها را براساس شباهت بین داده‌های جدید متناظر با هر پیکربندی، در m خوشه، خوشه‌بندی می‌کنیم.

به این ترتیب در پایان گام نخست m خوشه از پیکربندی‌های اولیه، خوشه شماره ۱ تا خوشه شماره m، داریم. در گام دوم برای هر خوشه شماره k (k=1,2,…,m) مطابق با الگوریتم ژنتیک ساده گام‌های زیر را تکرار می‌کنیم.

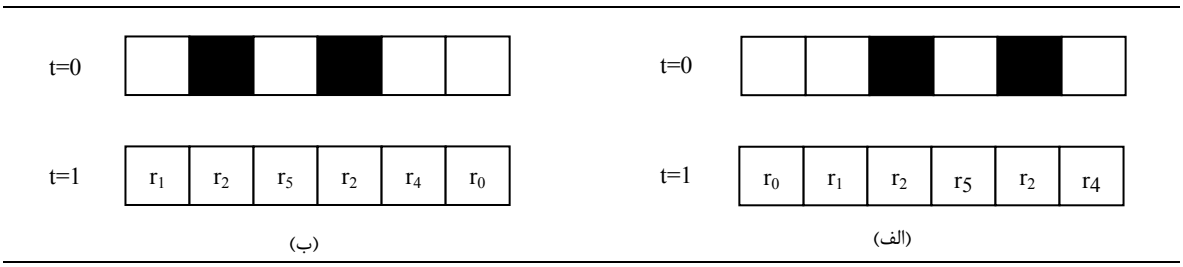
۱. R قانون به صورت تصادفی تولید می‌کنیم.
۲. مقدار برازش هر قانون را با استفاده از اعمال قانون روی پیکربندی‌های خوشه شماره k، محاسبه می‌کنیم.
۳. بخشی از بهترین قانون‌ها را بدون تغییر به نسل بعد منتقل می‌کنیم.
۴. بقیه قانون‌ها را با استفاده از بهترین قانون‌ها و عملگرهای ترکیب و جهش تولید می‌کنیم.
۵. در g نسل متوالی گام‌های ۲ تا ۴ را تکرار می‌کنیم.

Proposed Approach

Input: *I*: number of initial configuration
m: number of clusters
R: number of rules
g: number of generations in Genetic Algorithm
Output: bestRule

```
Generate I random configuration
For k = 1 to I
  Generate New Data from each Configuration
EndFor
Cluster configurations in m clusters according to similarity between new data
For k = 1 to m
  //Run GA for each cluster
  Generate R random rules
  For l = 1 to g
    Calculate fitness of each rule accoriding to initial configurations in k th cluster
    Sort rules according to their fitness
    Select 20% of best rules for next generation with out change
    Generate 80% of rules with cross over and mutation on best rules
  EndFor
  Rules[k] = best rule of last generation
EndFor
Use Rules[1 to m] as rules of first generation in the following GA
For l = 1 to g
  Calculate fitness of each rule
  Sort rules according to their fitness
  Select 20% of best rules for next generation with out change
  Generate 80% of rules with cross over and mutation on best rules
EndFor
Select best rule of last generation as bestRule
```

شکل ۳- شبه کد الگوریتم پیشنهادی



شکل ۴- اعمال قانون روی پیکربندی‌های مشابه

پس از گذشت g نسل الگوریتم ژنتیک پایان می‌پذیرد و بهترین قانون نسل پایانی را به عنوان بهترین قانون برای خوشه انتخاب می‌کنیم. پس از آنکه الگوریتم ژنتیک را برای تمام m خوشه اجرا کردیم، به m بهترین قانون می‌رسیم. در گام سوم یکبار دیگر الگوریتم ژنتیک را اجرا می‌کنیم. با این تفاوت که جمعیت پیکربندی‌های اولیه را از تمام خوشه‌ها انتخاب می‌کنیم و جمعیت قانون‌ها را نیز m بهترین قانون به دست آمده، قرار می‌دهیم. بنابراین:

۱. m بهترین قانون به دست آمده را به عنوان قوانین نسل اول در نظر می‌گیریم.
۲. مقدار برازش هر یک از m قانون را با استفاده از اعمال قانون روی پیکربندی‌های اولیه، محاسبه می‌کنیم.
۳. بخشی از بهترین قانون‌ها را بدون تغییر به نسل بعد منتقل می‌کنیم.
۴. بقیه قانون‌ها را با استفاده از بهترین قانون‌ها و عملگرهای ترکیب و جهش تولید می‌کنیم.
۵. در g نسل متوالی گام های ۲ تا ۴ را تکرار می‌کنیم.

پس از گذشت g نسل الگوریتم ژنتیک پایان می‌پذیرد و بهترین قانون نسل پایانی را به عنوان بهترین قانون خودکاره انتخاب می‌کنیم.

۴-۱- خوشه‌بندی پیکربندی‌ها با ایجاد داده جدید متناظر با هر پیکربندی

در خوشه‌بندی، مجموعه داده‌ها به خوشه‌هایی تقسیم می‌شود که داده‌های موجود در هر خوشه، بیشترین شباهت را با هم دارند و به داده‌های دیگر خوشه‌ها نیز شبیه نیستند. بنابراین در اولین گام برای خوشه‌بندی، باید نحوه محاسبه شباهت بین داده‌ها را مشخص کرد. در داده‌های عددی یکی از روش‌های محاسبه شباهت بین داده‌ها، محاسبه فاصله اقلیدسی بین داده‌ها است. به این ترتیب هر چه فاصله بین داده‌ها کمتر باشد، به این معنی است که داده‌ها شباهت بیشتری با هم دارند. اما استفاده از فاصله اقلیدسی برای محاسبه شباهت بین پیکربندی‌های اولیه دو حالته، یعنی در فضایی که فقط شامل ۰ و ۱ است، مناسب نیست. در این بخش یک شیوه جدید برای محاسبه شباهت پیکربندی‌ها ارایه می‌گردد که در آن پیکربندی‌ها را با استفاده از داده‌های جدیدی که از روی پیکربندی‌ها ایجاد می‌شوند، خوشه‌بندی می‌نماییم. در شکل ۴ دو پیکربندی اولیه برای خودکاره‌های سلولی یک بعدی دو حالته با ۶ سلول نشان داده شده است. مربع‌های سیاه سلول‌های با حالت ۱ و مربع‌های سفید سلول‌های با حالت ۰ را نشان می‌دهند. شعاع همسایگی برای هر سلول را ۱ در نظر گرفته‌شده است. در حالت عمومی، برای خودکاره‌های سلولی یک بعدی k حالته با شعاع همسایگی r، اندازه همسایگی 2r+1 است و قانون باید برای تمام k^{2r+1} حالت ممکن برای همسایگی سلول، حالت بعدی سلول را مشخص کند. بنابراین برای خودکاره سلولی شکل ۴، اندازه همسایگی ۳ خواهد بود و قانون باید برای تمام ۸ حالت ممکن برای همسایگی سلول، حالت بعدی سلول را مشخص کند. قانون را با استفاده از عبارت $r_7r_6r_5r_4r_3r_2r_1r_0$ نشان می‌دهیم که در آن هر یک از r_i ها می‌تواند ۰ یا ۱ باشد. در این صورت فقط حالت‌های خروجی قانون را نشان داده‌ایم و برای اینکه نشان دهیم حالت‌ها، متناظر با کدام همسایگی هستند، از زیرنویس استفاده کرده‌ایم. به عنوان مثال r_7 حالت بعدی سلولی را نشان می‌هد که در حال حاضر دارای همسایگی ۱۱۱ در مبنای ۲ یا همان ۷ در مبنای ۱۰ است.

نمایش پیکربندی‌های شکل ۴-الف و ۴-ب به صورت رشته‌ای از ۰ و ۱ ها، به ترتیب به صورت $c_{15}c_{14}c_{13}c_{12}c_{11}c_{10} = 001010$ و $c_{25}c_{24}c_{23}c_{22}c_{21}c_{20} = 010100$ است و فاصله اقلیدسی این دو رشته از هم $\sqrt{\sum(c_{1i}-c_{2i})^2} = 2$ خواهد بود. هر چند که دو پیکربندی اولیه شکل ۴ دارای فاصله اقلیدسی ۲ از یکدیگر هستند، اما قانونی که قرار است سرانجام آن‌ها را به یک پیکربندی ثابت برساند (مسائل مورد بررسی به عنوان محک ارزیابی که در بخش ۳.۴ شرح داده می شوند، دارای چنین خصوصیتی هستند)، رفتار یکسانی روی این دو پیکربندی خواهد داشت. با دقت در این زمینه متوجه می‌شویم که رفتار یکسان قانون روی این دو پیکربندی به این دلیل است که همسایگی سلول‌ها در این دو پیکربندی مشابه یکدیگر است. بنابراین برای محاسبه شباهت دو پیکربندی از محاسبه فاصله اقلیدسی پیکربندی‌ها استفاده نمی‌کنیم. بلکه متناظر با هر پیکربندی، داده جدیدی ایجاد می‌کنیم که نشان می‌دهد همسایگی سلول‌ها چیست و برای محاسبه شباهت پیکربندی‌ها از آن استفاده می‌کنیم. متناظر با پیکربندی‌های شکل ۴-الف و ۴-ب داده جدید به ترتیب (1,1,2,0,1,1,0,0) و (1,1,2,0,1,1,0,0) است که فاصله اقلیدسی برابر ۰ دارند. در ادامه توضیح می‌دهیم که داده جدید متناظر با هر پیکربندی، چگونه ایجاد می‌شود.

در الگوریتم پیشنهادی، برای هر پیکربندی دو حالته با شعاع همسایگی r، اندازه داده جدید برابر با 2^{2r+1}، یعنی همان اندازه تمام حالات ممکن برای همسایگی، خواهد بود. بنابراین برای هر یک از پیکربندی‌های شکل ۴، چندتایی $(n_0,n_1,n_2,n_3,n_4,n_5,n_6,n_7)$ داده جدید خواهد بود. در داده جدید n_i تعداد سلول‌هایی را نشان می‌دهد که همسایگی‌شان برابر با i است. به عنوان مثال n_5 نشان می‌دهد که در پیکربندی، چند سلول با همسایگی ۱۰۱ در مبنای ۲ یا همان ۵ در مبنای ۱۰ وجود دارد. شبه کدی که در شکل ۵ آمده است، شبه کد تولید داده جدید از روی یک پیکربندی است.

Generate New Data from a Configuration

Input: *Config*: an initial configuration with *N* cells
r, radius of neighborhood
Output: $(n_0 \dots n_{2^{2r+1}-1})$: a tuple of numbers

For $k = 0$ to $2^{2r+1} - 1$
  $n_k = 0$
EndFor
For $k = 1$ to N
  index = compute neighborhood of *k* th cell in *Config* as a decimal number
  $n_{index} = n_{index} + 1$
EndFor

شکل ۵- شبه کد محاسبه داده جدید متناظر با هر پیکربندی

با این روش پیکربندی‌ها به داده‌هایی تبدیل می‌شوند که مقایسه آن‌ها ساده‌تر است. به عنوان مثال داده‌های جدیدی که از روی خودکاره‌های سلولی‌ای با چهار سلول و شعاع همسایگی ۱ ایجاد می‌شوند، مطابق با جدول شماره ۱ خواهند بودند و از آن‌ها برای خوشه‌بندی پیکربندی‌ها استفاده می‌کنیم. به این ترتیب داده‌های جدید به دست آمده از روی پیکربندی‌هایی که حاصل از شیفت دادن می‌باشند، دقیقا مشابه هم هستند و واضح است که اعمال قانون روی چنین پیکربندی‌هایی نتیجه مشابهی را به دنبال خواهد داشت. در حالتی که پیکربندی‌ها شیفت داده شده یکدیگر نباشند، هر چه داده‌های جدید متناظر با پیکربندی‌ها، مشابهت بیشتری با هم داشته باشند، نتیجه اعمال قانون روی پیکربندی‌های متناظرشان نیز شبیه‌تر خواهد بود.

جدول ۱- پیکربندی‌های ۴ سلولی و داده جدید متناظرشان براساس همسایگی با شعاع یک

ردیف	پیکربندی خودکاره سلولی	داده جدید متناظر با پیکربندی
۱	۰۰۰۰	(۴,۰,۰,۰,۰,۰,۰,۰)
۲	۱۰۰۰	(۱,۱,۱,۰,۱,۰,۰,۰)
۳	۰۱۰۰	(۱,۱,۱,۰,۱,۰,۰,۰)
۴	۰۰۱۰	(۱,۱,۱,۰,۱,۰,۰,۰)
۵	۰۰۰۱	(۱,۱,۱,۰,۱,۰,۰,۰)
۶	۱۱۰۰	(۰,۱,۰,۱,۱,۰,۱,۰)
۷	۱۰۱۰	(۰,۰,۲,۰,۰,۲,۰,۰)
۸	۱۰۰۱	(۰,۱,۰,۱,۱,۰,۱,۰)
۹	۰۱۱۰	(۰,۱,۰,۱,۱,۰,۱,۰)
۱۰	۰۱۰۱	(۰,۰,۲,۰,۰,۲,۰,۰)
۱۱	۰۰۱۱	(۰,۱,۰,۱,۱,۰,۱,۰)
۱۲	۱۱۱۰	(۰,۰,۰,۱,۰,۱,۱,۱)
۱۳	۱۱۰۱	(۰,۰,۰,۱,۰,۱,۱,۱)
۱۴	۱۰۱۱	(۰,۰,۰,۱,۰,۱,۱,۱)
۱۵	۰۱۱۱	(۰,۰,۰,۱,۰,۱,۱,۱)
۱۶	۱۱۱۱	(۰,۰,۰,۰,۰,۰,۰,۴)

در جدول ۱ رشته‌های ستون **پیکربندی خودکاره سلولی**، بیانگر حالت سلول‌ها در یک خودکاره سلولی یک بعدی دو حالته با ۴ سلول هستند و چندتایی‌های ستون **داده جدید متناظر با پیکربندی**، همسایگی‌های سلول‌های خودکاره سلولی را نشان می‌دهند. داده‌های این ستون مطابق با الگوریتم شکل ۵ به دست آمده‌اند. به عنوان مثال اگر حالت سلول‌ها در خودکاره سلولی با رشته ۱۰۰۱ نمایش داده شده باشد، آن‌گاه همسایگی سلول‌ها به ترتیب از راست، ۰۱۱ یا همان ۳، ۰۰۱ یا همان ۱، ۱۰۰ یا همان ۴ و ۱۱۰ یا همان ۶ خواهد بود و داده جدید به صورت (۰,۱,۰,۱,۱,۰,۱,۰) خواهد بود. اولین عدد، به ترتیب از چپ، در چندتایی‌ای که نمایش دهنده داده جدید است، تعداد سلول‌هایی را نشان می‌دهد که همسایگی‌شان برابر با ۰ است. دومین عدد در رشته عددی نمایش دهنده داده جدید، تعداد سلول‌هایی را نشان می‌دهد که همسایگی‌شان برابر با ۱ است. سومین عدد در رشته عددی نمایش دهنده داده جدید، تعداد سلول‌هایی را نشان می‌دهد که همسایگی‌شان برابر با ۲ است و همین طور n امین عدد در رشته عددی نمایش دهنده داده جدید، تعداد سلول‌هایی را نشان می‌دهد که همسایگی‌شان برابر با n-1 است. بنابراین (۰,۱,۰,۱,۱,۰,۱,۰) نشان می‌دهد که پیکربندی مورد نظر دارای ۱ سلول با همسایگی ۱، ۱ سلول با همسایگی ۳، ۱ سلول با همسایگی ۴ و ۱ سلول با همسایگی ۶ است. با محاسبه فاصله اقلیدسی بین چندتایی‌ها در ستون **داده جدید متناظر با پیکربندی** در جدول ۱، متوجه می‌شویم که پیکربندی‌های ردیف‌های ۲، ۳، ۴ و ۵ مشابه یکدیگر هستند. همچنین پیکربندی‌های ردیف‌های ۶، ۸، ۹ و ۱۱، پیکربندی‌های ردیف‌های ۷ و ۱۰ و پیکربندی‌های ردیف‌های ۱۲، ۱۳، ۱۴ و ۱۵ مشابه هم هستند. علاوه بر این هیچ پیکربندی مشابه با پیکربندی‌های ردیف‌های ۱ و ۱۶ وجود ندارد. با این حال نزدیک‌ترین پیکربندی‌ها به پیکربندی ردیف ۱، پیکربندی‌های ردیف‌های ۲، ۳، ۴ و ۵ هستند و نزدیک‌ترین پیکربندی‌ها به پیکربندی ردیف ۱۶، پیکربندی‌های ردیف‌های ۱۲، ۱۳، ۱۴ و ۱۵ می‌باشند.

۵- ارزیابی کارایی الگوریتم پیشنهادی

برای ارزیابی کارایی الگوریتم پیشنهادی از شبیه سازی کامپیوتری روی چند مساله استاندارد استفاده شده است. در ادامه این بخش نخست مسائل ارزیابی که برای این منظور بکار رفته اند ارائه و سپس کارایی الگوریتم پیشنهادی مورد ارزیابی قرار گرفته و کارایی آن با کارایی الگوریتم های مشابه مقایسه می شود.

۵-۱- مسائل مورد بررسی به عنوان محک ارزیابی

در این بخش دو مساله استاندار که معمولا برای ارزیابی کارایی خودکاره های سلولی تکاملی بکار می روند شرح داده می شوند. این دو مساله عبارتند از مساله رده بندی براساس چگالی و مساله همگام سازی.

۵-۱-۱- مسأله رده‌بندی براساس چگالی

مسأله رده‌بندی براساس چگالی یکی از مسائل استانداردی است که برای ارزیابی کارایی خودکاره های سلولی تکاملی بکار می رود این مساله این‌گونه تعریف می شود. که اگر در پیکربندی اولیه یک خودکاره سلولی دو حالته، تعداد یک‌ها (صفرها) بیشتر از تعداد صفرها (یک‌ها) باشد، پس از تعدادی گام با اعمال قانون مورد نظر باید به پیکربندی نهایی تمام یک (صفر) برسیم. در شکل ۶ اعمال یک قانون مناسب برای مسأله رده‌بندی براساس چگالی، روی دو پیکربندی اولیه تصادفی نمایش داده شده است.

۵-۱-۲- مسأله همگام‌سازی

در مسأله همگام‌سازی، هدف پیداکردن قانونی است که بتواند با شروع از هر پیکربندی اولیه‌ای پس از تعداد محدودی گام، خودکاره سلولی را به حالتی برساند که از آن پس خودکاره سلولی بین دو پیکربندی تمام صفر و تمام یک نوسان کند. در شکل ۷ اعمال یک قانون مناسب برای مسأله همگام‌سازی، روی یک پیکربندی اولیه تصادفی نمایش داده شده است [۳۱].

۵-۲- نتایج ارزیابی کارایی الگوریتم پیشنهادی

در این بخش کارایی الگوریتم پیشنهادی روی دو مسأله رده‌بندی براساس چگالی و مسأله همگام‌سازی که پیش‌تر شرح داده شد مورد ارزیابی قرار می‌گیرد. برای مقایسه نتایج حاصل از الگوریتم پیشنهادی، الگوریتم پیشنهادی چندین بار اجرا شده است و نتایج به دست آمده برای مقدار برازش بهترین قانون در نسل پایانی، با استفاده از نمودار جعبه‌ای نمایش داده شده است. پارامترهای مورد استفاده برای پیاده‌سازی شبه کد شکل ۳ در جدول شماره ۲ آورده شده‌اند. برای بررسی نتایج

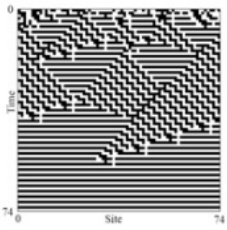
از ۲۰ اجرای مختلف استفاده شده است. نمودار شکل ۸، برازش بهترین قانون‌های نسل پایانی را برای مسأله رده‌بندی براساس چگالی نشان می‌دهد. در این نمودار برازش بهترین قانون‌های به دست آمده با استفاده از الگوریتم ژنتیک ساده (روش معرفی شده در بخش ۱.۳ که در نمودارها به اختصار روش پایه نامیده شده است) و برازش بهترین قانون‌های به دست آمده با استفاده از الگوریتم پیشنهادی با هم مقایسه شده‌اند. این مقایسه در سمت راست شکل ۸، با استفاده از مقادیر و در سمت چپ شکل ۸ با استفاده از نمودار جعبه‌ای نشان داده شده است.

نتایج نشان می‌دهند که کارایی الگوریتم پیشنهادی در جستجوی قانون‌هایی با برازش بالاتر، بیشتر از روش معرفی شده در بخش ۱.۳ است. بعلاوه واریانس کارایی الگوریتم پیشنهادی کمتر از الگوریتم ژنتیک ساده است و این نشان می‌دهد که کارایی الگوریتم پیشنهادی در اجراهای مختلف از نوسانات کمی برخوردار است. مشابه مسأله رده‌بندی براساس چگالی، در نمودار شکل ۹ برازش بهترین قانون‌های نسل پایانی برای مسأله همگام‌سازی با استفاده از روش معرفی شده در بخش ۱.۳ و الگوریتم پیشنهادی با هم مقایسه شده‌اند.

(الف) پیکربندی اولیه تصادفی که در آن تعداد یک‌ها بیشتر از تعداد صفرها است

(ب) پیکربندی اولیه تصادفی که در آن تعداد صفرها بیشتر از تعداد یک‌ها است

شکل ۶- نمودار مکان ـ زمان برای دو پیکربندی اولیه تصادفی پس از اعمال یک قانون مناسب برای مسأله رده‌بندی براساس چگالی

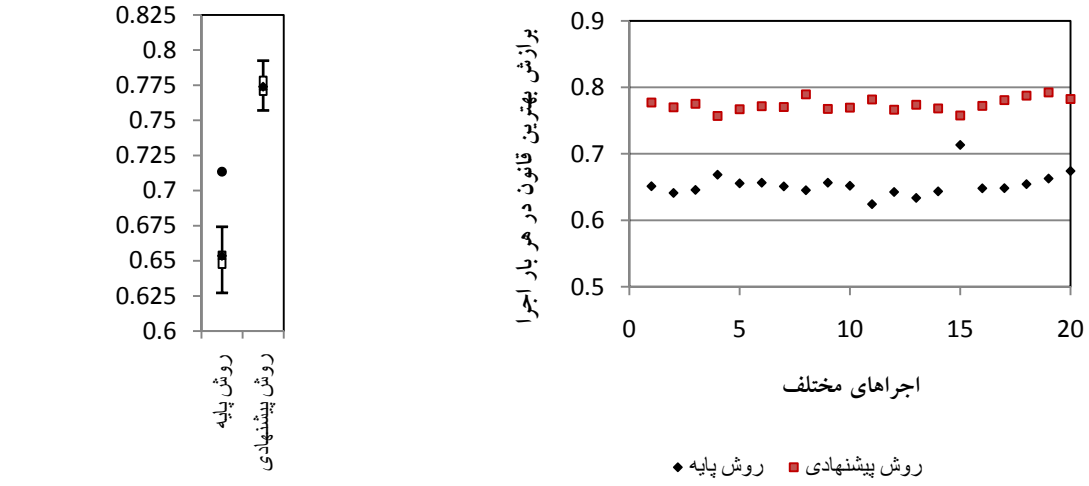


شکل ۷- نمودار مکان ـ زمان برای یک پیکربندی اولیه تصادفی پس از اعمال یک قانون مناسب برای مسأله همگام‌سازی [۳۱]

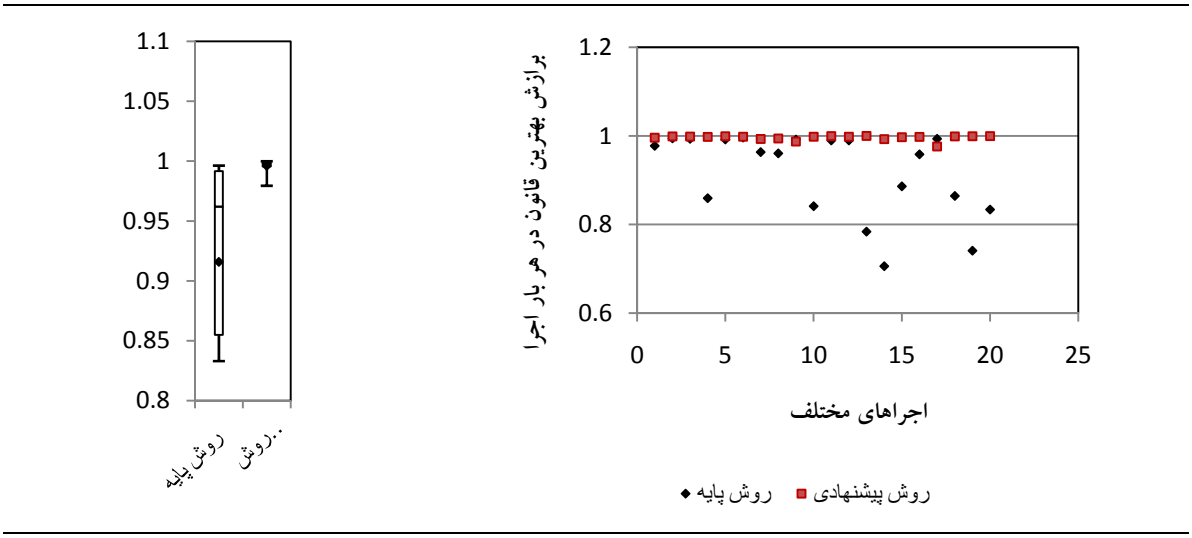
جدول ۲- مقادیر پارامترها در پیاده‌سازی انجام شده

پارامتر	مقدار
تعداد خوشه‌ها (m)	۱۰۰
اندازه جمعیت قوانین (R)	۱۰۰
تعداد پیکربندی‌های اولیه (I)	۵۰۰۰
تعداد سلول‌های خودکاره سلولی (N)	۱۴۹
تعداد نسل‌ها در هر بار اجرای الگوریتم ژنتیک (g)	در گام دوم الگوریتم تعداد نسل‌ها حداکثر ۵۰ است. در مواردی که خوشه‌ها شامل پیکربندی‌های ساده باشند و سریع به قانونی با برازش بالا برسیم، تعداد نسل‌ها کمتر می‌شود. در گام سوم الگوریتم g=۵۰
تعداد گام‌های اعمال یک قانون روی یک پیکربندی (M)	۲۹۸=۱۴۹×۲
شعاع قانون (r)	۳
اندازه مجموعه حالت (k)	۲
فاصله دو نسل	۰.۸
نرخ جهش	۰.۰۲

با توجه به شکل ۸ و مقایسه برازش بهترین قانون‌ها در اجراهای مختلف، مشاهده می‌شود که در الگوریتم پیشنهادی بیشتر اجراها قادرند قانون‌هایی با برازش نزدیک به ۰.۸ را پیدا کنند، در حالیکه در جستجوی قانون با استفاده از الگوریتم ژنتیک به طور متوسط برازش قانون‌های پیدا شده، در حدود ۰.۶۵ است. در شکل ۹ نیز که نتایج مربوط به مسأله همگام‌سازی را مشاهده می‌کنید، الگوریتم پیشنهادی نتایج همگن‌تری را نسبت به الگوریتم ژنتیک ارایه می‌دهد.



شکل ۸- برازش بهترین قانون‌های نسل پایانی روی ۱۰۰۰۰ پیکربندی اولیه با توزیع دو جمله‌ای برای مسأله رده‌بندی براساس چگالی



شکل ۹- برازش بهترین قانون‌های نسل پایانی روی ۱۰۰۰۰ پیکربندی اولیه با توزیع دو جمله‌ای برای مسأله همگام‌سازی

بکارگیری الگوریتم پیشنهادی برای مسأله همگام‌سازی در بیشتر اجراها منجر به کشف قانون‌هایی با برازش نزدیک به ۱.۰ شده است. استفاده از دو گام برای جستجوی قانون همان‌گونه که در مرجع [۲۸] آمده است، همانند فیلتری برای جداسازی قانون‌های بهتر از سایر قوانین عمل می‌کند و باعث تکامل قانون‌های بهتر در گام دوم می‌شود. از این‌رو استفاده از دو گام برای جستجوی قانون روش مناسبی است. در این مقاله این دو گام با گام دیگری که چگونگی تقسیم فضای پیکربندی‌های اولیه با استفاده از خوشه‌بندی را شرح می‌دهد، تکمیل شده‌اند.

۶- نتیجه‌گیری

در این مقاله روشی ارایه شد که با استفاده از خوشه‌بندی فضای پیکربندی‌های اولیه به جستجوی بهترین قانون برای مسأله رده‌بندی براساس چگالی و همگام‌سازی می‌پرداخت. به این منظور پس از معرفی کلیات الگوریتم پیشنهادی که بر پایه روش تقسیم و حل بنا شده بود، یک روش جدید برای محاسبه شباهت پیکربندی‌های اولیه ارایه شد و به بررسی نحوه خوشه‌بندی فضا پرداختیم. نتایج حاصل بیانگر این بودند که در تمام اجراها الگوریتم پیشنهادی نتایج بهتری را به دست می‌داد. بر این اساس بهترین قانون به دست آمده با این روش برای مسأله رده‌بندی براساس چگالی دارای برازشی در حدود ۰.۸ است. هرچند که بهترین قانون به دست آمده تاکنون برای مسأله رده‌بندی براساس چگالی با استفاده از الگوریتم‌های تکاملی چند هدفه به دست آمده است و دارای برازشی در حدود ۰.۸۹ [۲۸] است، اما نتایج به دست آمده در این مقاله بهترین نتایجی هستند که در آن‌ها الگوریتم تکاملی مورد استفاده الگوریتم ژنتیک بوده است.

مراجع

[1] N. Ganguly, B. K. Sikdar, A. Deutsch, G. Canright, and P. Chaudhuri, *A Survey on Cellular Automata,* Technical Report, Centre for High Performance Computing, Dresden University of Technology, 2003.

[2] P. Sarkar, "A Brief History of Cellular Automata," *ACM Computing Surveys*, vol. 32, no. 1, pp. 80-107, 2000.

[3] L. N. Castro, "Fundamentals of Natural Computing: An Overview," *Physics of Life Reviews*, vol. 4, no. 1, pp.1-36, 2007.

[4] F. Seredynski, P. Bouvry, and A. Y. Zomaya, "Cellular Automata Computations and Secret Key Cryptography," *Parallel Computing*, vol. 30, no. 5-6, pp. 753-766, 2004.

[5] H. Gutowitz (Ed.), *Cellular Automata Theory and Experiment*, MIT Press, 1991.

[6] R. Das, M. Mitchell, and J. P. Crutchfield, "A Genetic Algorithm Discovers Particle-Based Computation in Cellular Automata," *Proc. the 3th Conference on Parallel Problem Solving from Nature*, pp. 344-353, 1994.

[7] M. Mitchell, J. P. Crutchfield, and R. Das, "Evolving Cellular Automata with Genetic Algorithms: A Review of Recent Work," *Proc. the First International Conference on Evolutionary Computation and Its Applications*, 1996.

[8] D. Andre, F. H. Bennett, and J. R. Koza, "Discovery by Genetic Programming of a Cellular Automata Rule that is Better than any Known Rule for the Majority Classification Problem," *Proc. the First Annual Conference of Genetic Programming*, pp. 3-11, 1996.

[9] D. Andre, F. H. Bennett, and J. R. Koza, "Evolution of Intricate Long Distance Communication Signals in Cellular Automata Using Genetic Programming," *Proc. the 5th International Workshop on the Synthesis and Simulation of Living Systems*, 1997.

[10] H. Juille, and J. B. Pollack, "Coevolving the Ideal Trainer: Application to the Discovery of Cellular Automata Rules," *Proc. the Third Annual Genetic Programming Conference*, pp. 519-527, 1998.

[11] G. M. B. Oliveira, J. C. Bortot, and P. P. B. Oliveira, "Multi Objective Evolutionary Search for One Dimensional

Cellular Automata in the Density Classification Task," *Proc. the 8th International Workshop on the Synthesis and Simulation of Living Systems*, pp.202-206, 2003.

[12] H. Beigy, and M. R. Meybodi, "A Mathematical Framework for Cellular Learning Automata," *Advances in Complex Systems*, vol. 7, no. 3-4, pp. 295-319, 2004.

[13] S. Wolfram, *New Kind of Science*, Wolfram Media Inc., 2002.

[14] N. Kato, T. Okuno, R. Suzuki, and H. Kanoh, "Modeling Virtual Cities Based on Interaction Between Cells," *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, pp. 143-148, 2000.

[15] C. C. Oliveira, and P. P. B. Oliveira, "An Approach to Searching for Two-Dimensional Cellular Automata for Recognition of Handwritten Digits," *Proc. the 7th Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence*, vol. 5317, pp. 462-471, 2008.

[16] N. H. Packard, "Adaptation toward the edge of chaos," *Dynamic Patterns in Complex Systems*, pp. 293-301, 1988.

[17] J. P. Crutchfield, M. Mitchell, and R. Das, "The Evolutionary Design of Collective Computation in Cellular Automata," *Evolutionary Dynamics-Exploring the Interplay of Selection, Neutrality, Accident, and Function, Santa Fe Institute Series in the Sciences of Complexity*, Oxford University Press, pp. 361-411, 2001.

[18] M. Mitchell, P. T. Hraber, and J. P. Crutchfield, "Revisiting the Edge of Chaos: Evolving Cellular Automata to Perform Computations," *Complex Systems*, vol. 7, pp. 89-130, 1993.

[19] M. Mitchell, J. P. Crutchfield, and P. T. Hraber, "Evolving Cellular Automata to Perform Computations: Mechanisms and Impediments," *Physica D*, vol. 75, no. 1-3, pp. 361-391, 1994.

[20] M. Mitchell, "Computation in Cellular Automata: A Selected Review," *Non-Standard Computation*, pp. 95-140, 1998.

[21] H. Juille, and J. B. Pollack, "Coevolutionary Learning: A Case Study," *Proc. the Fifteenth International Conference on Machine Learning*, pp. 251-259, 1998.

[22] J. Paredis, "Coevolving Cellular Automata: Be Aware of the Red Queen," *Proc. the 7th International Conference on Genetic Algorithms*, pp. 393-400, 1997.

[23] G. M. B. Oliveira, P. P. B. Oliveira, and N. Omar, "Definition and Application of a Five-parameter Characterization of One-dimensional Cellular Automata Rule Space," *Artificial Life*, vol. 7, no. 3, pp. 277-301, 2001.

[24] G. M. B. Oliveira, J. C. Bortot, and P. P. B. Oliveira, "Multi Objective Evolutionary Search for One Dimensional Cellular Automata in the Density Classification Task," *Proc.*

the 8th International Workshop on the Synthesis and Simulation of Living Systems, pp. 202-206, 2003.

[25] G. M. B. Oliveira, P. P. B. Oliveira, and N. Omar, "Improving Genetic Search for One-Dimensional Cellular Automata, Using Heuristics Related to Their Dynamic Behavior Forecast," *Proc. IEEE Conference on Evolutionary Computation*, pp. 348-355, 2001.

[26] G. M. B. Oliveira, J. C. Bortot, and P. P. B. Oliveira, "Further Results on Multi Objective Evolutionary Search for One-Dimensional, Density Classifier, Cellular Automata, and Strategy Analysis of the Rules," *Advances in Technological Applications of Logical and Intelligent Systems*, vol. 186, p. 133-159, 2009.

[27] P. P. B. Oliveira, J. C. Bortot, and G. M. B. Oliveira, "The Best Currently Known Class of Dynamically Equivalent Cellular Automata Rules for Density Classification," *Neurocomputing*, vol. 70, no. 1-3, pp. 35-43, 2006.

[28] D. Wolz, and P. P. B. Oliveira, "Very Effective Evolutionary Techniques for Searching Cellular Automata Rule Spaces," *Journal of Cellular Automata*, vol. 3, no. 4, pp. 289-312, 2008.

[29] N. Srinivas, and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Journal of Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, 1994.

[30] M. Najafi, and H. Beigy, *"Using PCA to Improve Evolutionary Cellular Automata Algorithms," Proc. the 10th Annual Conference on Genetic and Evolutionary Computation*, pp. 1129-1130, 2008.

[31] R. Das, J. P. Crutchfield, M. Mitchell, and J. E. Hanson, "Evolving Globally Synchronized Cellular Automata," *Proc. the 6th International Conference on Genetic Algorithms*, pp. 336-343, 1995.

سمانه‌السادات موسوی فارغ‌التحصیل دوره کارشناسی‌ارشد مهندسی کامپیوتر از دانشگاه صنعتی شریف است و دوره کارشناسی‌ارشد را در سال ۸۸ به پایان رسانده است. او همچنین در خلال سال‌های ۸۲ تا ۸۶ مشغول به تحصیل در دوره کارشناسی مهندسی کامپیوتر در دانشگاه صنعتی شریف بوده است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

moussavi@ce.sharif.edu

حمید بیگی تحصیلات خود را در مقاطع کارشناسی و کارشناسی‌ارشد مهندسی کامپیوتر به ترتیب در سال‌های ۱۳۷۱ و ۱۳۷۴ از دانشگاه شیراز و در مقطع دکتری مهندسی کامپیوتر در سال ۱۳۸۳ از دانشگاه صنعتی امیرکبیر به پایان رسانده است و هم اکنون دانشیار

دانشکده مهندسی دانشگاه صنعتی شریف می‌باشد. زمینه‌های پژوهشی ایشان عبارتند از الگوریتم‌های موازی، الگوریتم‌های یادگیری، هوش محاسباتی و کاربردهای آن در شبکه‌های کامپیوتری می‌باشد.
آدرس پست‌الکترونیکی ایشان عبارت است از:

beigy@sharif.edu

اطلاعات بررسی مقاله:

تاریخ ارسال: ۸۹/۸/۱۳
تاریخ اصلاح: ۹۰/۲/۶
تاریخ قبول شدن: ۹۰/۲/۱۴
نویسنده مرتبط: دکتر حمید بیگی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران.

[1] Cellular Automaton (CA)
[2] Von Neumann
[3] Ulam
[4] Forward Problem
[5] Inverse Problem
[6] Wolfram
[7] Genetic Algorithm
[8] Genetic Programming
[9] Coevolutionary Algorithms
[10] Multiobjective Algorithms
[11] Evolutionary Cellular Automata (EvCA)
[12] Configuration
[13] Metaheuristic Optimization Algorithms
[14] Density Classification Problem
[15] Local Maxima
[16] Neighborhood Dominance (ND)
[17] Sensitivity
[18] Activity Propagation (AP)
[19] Nondominated Sorting Genetic Algorithm (NSGA)