

# ارائه یک مدل جدید ممیتیکی مبتنی بر اتوماتای یادگیر ساختار ثابت

مهدی رضاپور میرصالح و محمدرضا میبیدی

موجود قبل می‌گردد. در این مدل، جستجوی محلی معادل یک عملگر ژنتیکی است که باعث پالایش ساختار ژنتیکی کروموزوم‌ها می‌شود [۱] و [۲]. مهم‌ترین عیب این مدل یادگیری، از بین رفتن سیر تکامل عمومی کروموزوم‌هاست که توسط الگوریتم ژنتیکی به دست آمده است. به عبارت دیگر جستجوی محلی با ایجاد تغییر در ساختار ژنتیکی کروموزوم‌ها، روند تکامل عمومی کروموزوم‌ها را دچار خدشه می‌کند [۳] اما در مدل یادگیری بالدوینی ساختار ژنتیکی افراد تغییر نمی‌یابد. با توجه به این که در این مدل نتیجه جستجوی محلی در ساختار ژنتیکی کروموزوم اثری نخواهد گذاشت، بایستی از پارامتر دیگری جهت اعمال اثر جستجوی محلی استفاده شود. در اکثر مقالات چاپ‌شده، نتیجه جستجوی محلی بر روی شایستگی اعمال می‌شود. در این روش یک یا چند موجود که از لحاظ ساختار ژنتیکی شبیه به هم هستند و احتمالاً در همسایگی یکدیگر در فضای جستجوی قرار دارند، پس از اعمال جستجوی محلی، دارای شایستگی یکسانی خواهند شد. در نتیجه فضای شایستگی موجودات که نسل بعد طبق آن انتخاب خواهد شد دارای سطح همواری خواهد بود و بسیاری از مینی‌م‌های محلی را پوشش خواهد داد. هرچند که این ویژگی باعث افزایش هزینه زمانی جهت رسیدن به جواب نهایی مسئله می‌شود اما از قرارگرفتن موجودات در بسیاری از مینی‌م‌های محلی جلوگیری کرده و از همگرایی زودرس اجتناب می‌ورزد [۳].

یکی از روش‌های یادگیری تقویتی، اتوماتای یادگیر است. در یادگیری تقویتی یک عامل یادگیرنده در طی یادگیری با فعل و انفعالات مکرر با محیط، به یک سیاست کنترل بهینه می‌رسد. جهت آشنایی با جزئیات تئوری یادگیری تقویتی و کاربردهای مختلف آن به [۴] و [۵] مراجعه شود. از اتوماتای یادگیر برای مدل‌کردن سیستم‌های یادگیر استفاده می‌شود که به صورت تصادفی اقدامی را از مجموعه اقدام‌های خود انتخاب کرده و در یک محیط ناشناخته اعمال می‌کند. سپس پاسخ محیط دریافت شده و احتمال اقدام‌ها بر طبق الگوریتم یادگیری به روز می‌شوند و روال فوق تکرار می‌گردد [۶] و [۷]. اتوماتاهای یادگیر در کاربردهای گوناگونی مورد استفاده قرار گرفته‌اند که جهت آشنایی با آنها می‌توان به [۸] و [۹] مراجعه نمود.

اولین بار در [۱۰] مدل ترکیبی GALA توسط رضاپور و میبیدی پیشنهاد شد. در این مدل تکاملی، کروموزوم‌ها به وسیله اتوماتای مهاجرت اشیا بازنمایی شده‌اند. هدف از بازنمایی منحصر به فرد در این مدل، استفاده هم‌زمان از اثرات مثبت تکامل و یادگیری محلی است. از این مدل در کاربردهای مختلفی استفاده گردیده که در ادامه مقاله به آنها پرداخته شده است. مهم‌ترین مشکل مدل ترکیبی GALA، همانند سایر روش‌هایی که از مدل یادگیری لامارکی استفاده می‌کنند، از بین رفتن سیر تکامل عمومی است. در این مقاله مدل ممیتیکی جدیدی با نام GALA۲ بر مبنای مدل GALA ارائه شده است. در روش ارائه‌شده جهت افزایش سرعت همگرایی و فرار از همگرایی زودرس، به طور هم‌زمان از دو مدل یادگیری لامارکی و بالدوینی استفاده می‌شود.

ادامه این مقاله به شکل زیر سازماندهی شده است: در بخش‌های ۲ و ۳ به ترتیب اتوماتای یادگیر و اتوماتای مهاجرت اشیا شرح داده شده‌اند.

چکیده: الگوریتم ممیتیکی یکی از انواع الگوریتم‌های تکاملی است که با استفاده از جستجوی عمومی و جستجوی محلی فضای حل مسئله را به صورت بهینه جستجو می‌نماید. تعادل بین جستجوی عمومی و محلی، همواره یکی از مسایل مهم در این دسته از الگوریتم‌ها است. در این مقاله یک مدل جدید ممیتیکی با نام GALA۲ ارائه شده است. این مدل از ترکیب الگوریتم ژنتیک و اتوماتای مهاجرت اشیا که نوع خاصی از اتوماتای یادگیر ساختار ثابت می‌باشد، تشکیل شده است. در مدل ارائه‌شده جستجوی عمومی توسط الگوریتم ژنتیک و یادگیری محلی به وسیله اتوماتای یادگیر انجام می‌شود. در این مدل جهت افزایش سرعت همگرایی و فرار از همگرایی زودرس، به طور هم‌زمان از دو مدل یادگیری لامارکی و بالدوینی استفاده شده است. در این مدل تکاملی، جهت استفاده توأم از اثرات مثبت تکامل و یادگیری محلی، کروموزوم‌ها به وسیله اتوماتای مهاجرت اشیا بازنمایی شده‌اند. جهت نمایش برتری مدل ارائه‌شده نسبت به سایر روش‌های موجود، از این مدل برای حل مسئله تناظر گراف استفاده گردیده است.

کلیدواژه: الگوریتم ممیتیک، مم، جستجوی محلی، جستجوی عمومی، اتوماتای یادگیر، اتوماتای مهاجرت اشیا.

## ۱- مقدمه

الگوریتم‌های تکاملی، الگوریتم‌های قدرتمندی در جستجوی سراسری هستند که دارای توانایی اکتشاف و بهره‌برداری در فضای مسئله می‌باشند. اکتشاف و بهره‌برداری دو هدف اساسی در جستجو هستند. اکتشاف به معنای اطمینان از سراسری بودن جستجو است و از این جهت اهمیت دارد که فضای مسئله باید به نحو قابل اطمینانی برای یافتن سراسری نقطه بهینه، جستجو شود. بهره‌برداری از این لحاظ مورد توجه است که به تلاش پیرامون بهترین جواب برای یافتن پاسخ‌های بهتر می‌پردازد. الگوریتم‌های مناسب و کارای زیادی وجود دارند که به این دو هدف توجه می‌کنند و هر دو را ارضا می‌کنند که به این گونه از الگوریتم‌ها روش‌های ترکیبی گفته می‌شود. ترکیب الگوریتم‌های ژنتیک با جستجوی محلی هم از این دسته از الگوریتم‌ها محسوب می‌شوند. این الگوریتم‌ها که از ترکیب الگوریتم ژنتیکی متعارف با الگوریتم‌های جستجوی محلی ایجاد شده‌اند، به گونه‌ای بسیار کارا تر توانایی جستجو دارند. این الگوریتم‌ها به الگوریتم‌های ممیتیک یا الگوریتم‌های با یادگیری لامارکی و یادگیری بالدوینی هم معروف هستند. مهم‌ترین نقش را در این الگوریتم‌ها، الگوریتم‌های جستجوی محلی بر عهده دارند. در مدل یادگیری لامارکی بر خلاف مدل یادگیری بالدوینی، خصوصیات ژنتیکی یک موجود در طول حیات آن تغییر می‌کند. به عبارت دیگر با اعمال جستجوی محلی و یافتن یک جواب بهتر، ساختار ژنتیکی و شایستگی جواب یافته‌شده جایگزین

این مقاله در تاریخ ۱۶ آبان ماه ۱۳۹۶ دریافت و در تاریخ ۱۱ تیر ماه ۱۳۹۷ بازنگری شد.

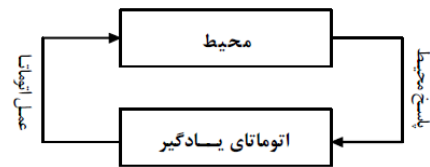
مهدی رضاپور میرصالح (نویسنده مسئول)، گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه پیام نور، تهران، ایران، (email: mrezapoorm@pnu.ac.ir).  
محمدرضا میبیدی، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، (email: mmeybodi@aut.ac.ir).

از اتوماتای یادگیر در حوزه‌های مختلفی نظیر بهینه‌سازی [۱۱] و [۱۲]، شبکه‌های اجتماعی [۱۳] تا [۱۶]، گراف‌ها [۱۷] و [۱۸]، شبکه‌های پتری [۱۹] و [۲۰]، شبکه‌های کامپیوتری [۲۱] و [۲۲]، سیستم‌های توصیه‌گر [۱۴] تا [۲۳] و غیره استفاده شده است.

یکی از انواع اتوماتای یادگیر ساختار ثابت، اتوماتای مهاجرت اشیا [۲۴] است که توسط پنج‌تایی  $\langle \alpha, \Phi, \beta, F, G \rangle$  نشان داده می‌شود.  $\alpha = \{\alpha_1, \dots, \alpha_r\}$  مجموعه  $r$  عضوی اقدام‌های مجاز برای اتوماتای یادگیر،  $\Phi = \{\Phi_1, \dots, \Phi_s\}$  مجموعه  $s$  عضوی وضعیت‌های اتوماتای یادگیر و  $\beta = \{0, 1\}$  مجموعه ورودی‌های محیط می‌باشد که در آن ۱ به معنای پاسخ نامطلوب و ۰ به معنای پاسخ مطلوب از طرف محیط است.  $F: \Phi \times \beta \rightarrow \Phi$  تابع نگاشت وضعیت‌ها است که با توجه به وضعیت فعلی و پاسخ دریافتی از محیط وضعیت بعدی را مشخص می‌کند و نهایتاً  $G: \Phi \rightarrow \alpha$  تابع نگاشت خروجی می‌باشد که خروجی اتوماتا را با توجه به وضعیت فعلی مشخص می‌کند. در این اتوماتا هر اقدام یک دسته را نشان می‌دهد. در اتوماتاهای با ساختار ثابت پاسخ محیط به اتوماتا سبب می‌شود که اتوماتا از یک وضعیت به وضعیت دیگر منتقل شود، در صورتی که در اتوماتای مهاجرت اشیا، اشیا به وضعیت‌ها نسبت داده می‌شوند و پاسخ محیط به اتوماتا سبب گردش اشیا در بین وضعیت‌های اتوماتا می‌گردد و از طریق این گردش دسته‌بندی اشیا انجام می‌گیرد. اگر شیء  $w_i$  در خروجی  $\alpha_k$  اتوماتای مهاجرت اشیا قرار داشته باشد این شیء متعلق به دسته شماره  $k$  است. برای خروجی  $\alpha_k$  مجموعه وضعیت  $\{\Phi_{(k-1)N+1}, \dots, \Phi_{kN}\}$  در نظر گرفته می‌شود که  $N$  عمق حافظه را نشان می‌دهد. به طور کلی می‌توان  $\Phi_{(k-1)N+1}$  را داخلی‌ترین وضعیت و  $\Phi_{kN}$  را خارجی‌ترین وضعیت (وضعیت مرزی) این خروجی دانست. اگر دو شیء  $w_i$  و  $w_j$  به ترتیب در وضعیت‌های  $\Phi_{(k-1)N+p}$  و  $\Phi_{(k-1)N+q}$  (مشروط به این که  $p, q < N, p < q$ ) قرار داشته باشند در این صورت قطعیت تعلق شیء  $w_i$  بیشتر است. بنابراین برای خروجی  $\alpha_k$  وضعیت  $\Phi_{(k-1)N+1}$  وضعیت با بیشترین درجه قطعیت و وضعیت  $\Phi_{kN}$  وضعیت با کمترین درجه قطعیت نامیده می‌شود. اگر شیء  $w_i$  در خروجی  $\alpha_k$  اتوماتای مهاجرت اشیا قرار داشته باشد و پاسخ ۰ از محیط دریافت شود  $w_i$  به سمت داخلی‌ترین وضعیت اقدام متناظر  $\Phi_{(k-1)N+1}$  حرکت خواهد کرد و در صورتی که پاسخ ۱ از محیط دریافت شود  $w_i$  به سمت وضعیت مرزی  $\Phi_{kN}$  سوق داده خواهد شد.

### ۳- کارهای مرتبط

GALA یک مدل ترکیبی مبتنی بر الگوریتم ژنتیک و اتوماتای یادگیر است. از این مدل در حوزه‌های مختلفی استفاده شده که مهم‌ترین آنها عبارتند از مسایل حوزه گراف، مسأله زمان‌بندی، مسأله بهینه‌سازی پرس‌وجوهای بانک اطلاعاتی. در همه مقالاتی که از مدل ترکیبی GALA بهره برده‌اند از مدل یادگیری لامارکی جهت اعمال جستجوی محل استفاده شده است. در [۲۵] ضمن استفاده از مدل ترکیبی GALA در حل مسأله تناظر گراف، پیش‌پردازشی ارائه شده که باعث افزایش سرعت همگرایی روش ترکیبی در گراف‌های بدون وزن می‌شود. در این مقاله برتری روش ارائه‌شده به وسیله مدل ترکیبی GALA نسبت به سایر روش‌ها نشان داده شده است. در [۲۶] تا [۲۹] از GALA در بهینه‌سازی در پرس‌وجوهای بانک اطلاعاتی استفاده شده است. در این مقاله‌ها نشان داده شده که با استفاده هم‌زمان از الگوریتم ژنتیک و اتوماتای یادگیر، سرعت همگرایی افزایش یافته و از قرارگرفتن در مینیمم محلی جلوگیری به عمل می‌آید. در [۳۰] با استفاده از مدل ترکیبی، روشی



شکل ۱: تعامل اتوماتای یادگیر با محیط.

شرح کارهای مرتبط با موضوع در بخش ۴ آمده است. بخش ۵ به معرفی مدل ارائه‌شده می‌پردازد. در بخش ۶ به معرفی مسأله تناظر گراف جهت استفاده از مدل پیشنهادی برای حل آن پرداخته شده است. بخش ۷ شامل پیاده‌سازی آزمایشات و ارائه نتایج به دست آمده می‌باشد و نهایتاً در بخش ۷ نتیجه‌گیری به عمل آمده است.

## ۲- تئوری اتوماتای یادگیر

اتوماتای یادگیر [۶] ماشینی است که می‌تواند تعداد محدودی عمل را انجام دهد. هرگاه این ماشین عملی را انتخاب می‌کند، عمل انتخاب‌شده توسط محیط ارزیابی گردیده و نتیجه آن به صورت یک سیگنال بازخوردی مثبت (در صورت مناسب بودن عمل) یا منفی (در صورت نامناسب بودن عمل) به اتوماتا بازگردانده می‌شود. مقدار این سیگنال در انتخاب اعمال بعدی تأثیر می‌گذارد. هدف این فرایند این است که اتوماتا بعد از گذشت مدتی به سمت مناسب‌ترین عمل خود در محیط میل کرده و یا به عبارت دیگر یاد می‌گیرد که کدام عمل، بهترین عمل است. نحوه تعامل اتوماتای یادگیر و محیط در شکل ۱ نشان داده شده است.

مدل ریاضی محیط به صورت سه‌تایی  $\langle \alpha, \beta, \mathcal{C} \rangle$  تعریف می‌شود که  $\alpha = \{\alpha_1, \dots, \alpha_r\}$  مجموعه ورودی‌ها،  $\beta = \{\beta_1, \dots, \beta_m\}$  مجموعه مقادیر سیگنال تقویتی که توسط محیط تولید می‌شود و  $\mathcal{C} = \{c_1, \dots, c_r\}$  مجموعه‌ای از احتمال‌های تنبیه می‌باشد. هر  $c_i$  با  $\alpha_i$  رابطه دارد و در واقع  $c_i$ ‌ها مشخصات و رفتار محیط را تعریف می‌کنند. اگر احتمال‌های تنبیه ثابت باشند، محیط را ایستا و اگر در طول زمان تغییر کنند، محیط را غیر ایستا می‌نامند. بر حسب نحوه تعریف مجموعه  $\beta$ ، محیط به مدل  $P$ ، مدل  $Q$  و مدل  $S$  تقسیم‌بندی می‌شود. محیطی که عناصر سیگنال تقویتی آن فقط دو مقدار ۰ (نشانهگر موفقیت) و ۱ (نشانهگر شکست) را می‌گیرند مدل  $P$  می‌نامند. در محیط از نوع  $Q$ ، سیگنال تقویتی می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله  $[a, b]$  را اختیار کند و در محیط از نوع  $S$ ، سیگنال تقویتی متغیری تصادفی در فاصله  $[a, b]$  است. اگر احتمال‌های جریمه در طول زمان تغییر کنند، محیط را غیر ایستا و در غیر این صورت ایستا می‌نامند.

اتوماتاهای یادگیر به دو خانواده اتوماتاهای یادگیر با ساختار ثابت و اتوماتاهای یادگیر با ساختار متغیر دسته‌بندی می‌شوند [۷]. اتوماتاهای کرینسکی و کرایلو مثال‌هایی از اتوماتاهای با ساختار ثابت هستند. یک اتوماتای با ساختار ثابت را می‌توان با یک پنج‌تایی  $\langle \Phi, \alpha, \beta, F, G \rangle$  نشان داد که در آن  $\alpha = \{\alpha_1, \dots, \alpha_r\}$  مجموعه اقدام‌های اتوماتای یادگیر،  $\Phi = \{\Phi_1, \dots, \Phi_s\}$  مجموعه وضعیت‌های اتوماتا،  $\beta$  مجموعه ورودی اتوماتا (خروجی محیط)،  $F: \Phi \times \beta \rightarrow \Phi$  تابع تغییر وضعیت اتوماتا و  $G: \Phi \rightarrow \alpha$  تابع خروجی اتوماتا می‌باشد. در هر لحظه اتوماتا با توجه به وضعیت فعلی آن، اقدامی را انتخاب و به عنوان ورودی به محیط می‌دهد و محیط پس از اعمال اقدام واردشده توسط اتوماتا، یک پاسخ تصادفی که می‌تواند نشان‌دهنده شکست یا موفقیت اقدام اعمال‌شده باشد را تولید می‌کند و به عنوان ورودی به اتوماتا می‌فرستد. اتوماتا با توجه به پاسخ محیط اقدام مربوط را جریمه می‌کند یا به آن پاداش می‌دهد. تاکنون

می‌شود. در [۳۸] یک الگوریتم جدید مبتنی بر مدل GALA برای حل مسأله خوشه‌بندی سیستم‌های نرم‌افزاری پیشنهاد شده است. همچنین برتری این الگوریتم نسبت به الگوریتم ژنتیک و الگوریتم مبتنی بر اتوماتای مهاجرت اشیا نشان داده شده است. در این مقاله نشان داده شده که روش ترکیبی پایداری بالاتری نسبت به دو الگوریتم دیگر دارد. در [۳۹] الگوریتمی جهت حل مسأله زمان‌بندی مجموع تأخیرات وزن‌دار تک‌ماشین با کمک مدل ترکیبی GALA ارائه شده است. در این مقاله نشان داده شده که روش ترکیبی برتری قابل توجهی نسبت به روش‌های تکراری، حریمانه، کاهشی و ژنتیک دارد. در [۴۰] و [۴۱] با کمک مدل ترکیبی روشی جهت حل مسأله تخصیص داده در پایگاه داده‌های توزیع‌شده ارائه گردیده است. در این مقاله نشان داده شده که روش ارائه‌شده نسبت به سایر روش‌های موجود کارایی بالاتری دارد. همچنین نشان داده که استفاده از اتوماتای یادگیر با اتصالات مشابه اتوماتای ستلین، برتری قابل توجهی نسبت به اتوماتاهای با اتصالات مشابه کریسنکی و کرایلو دارد. در [۴۲] و [۴۳] با کمک مدل ترکیبی GALA روش جدیدی جهت حل مسأله زمان‌بندی گراف وظیفه در سیستم‌های موازی ارائه شده است. در [۴۲] نشان داده شده که روش ارائه‌شده نسبت به روش‌های DCP, MD, DSC, MCP دارای عملکرد بهتری است. همچنین در [۴۳] نشان داده شده که روش ترکیبی در مقایسه با روش‌های FCFS و MET دارای کارایی بالاتری است.

#### ۴- مدل GALA۲

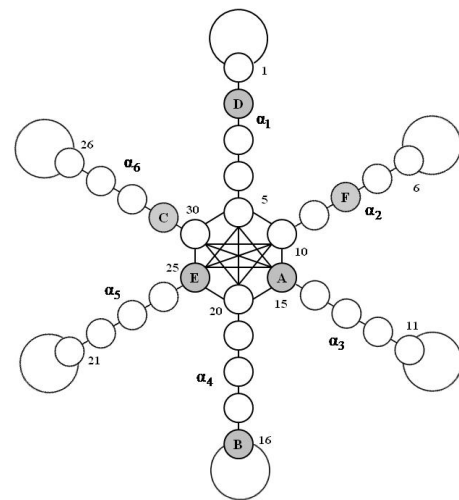
مدل GALA۲، ترکیبی از اتوماتای یادگیر و الگوریتم ژنتیک است که در آن از مدل یادگیری لامارکی و بالدوینی به صورت هم‌زمان استفاده شده است. در این مدل ممتیکی، جستجوی عمومی توسط الگوریتم ژنتیک انجام می‌شود و وظیفه یادگیری محلی بر عهده اتوماتای یادگیر است.

#### ۴-۱ بازنمایی کروموزوم به وسیله اتوماتای مهاجرت

##### اشیا در مدل GALA۲

در مدل GALA۲ کروموزوم‌ها به وسیله اتوماتای یادگیر مهاجرت اشیا بازنمایی می‌شوند. هدف از بازنمایی منحصر به فرد در این مدل، استفاده هم‌زمان از اثرات مثبت تکامل و یادگیری محلی است. در این نوع بازنمایی، هر اتوماتای مهاجرت اشیا دارای  $n$  اقدام است که هر اقدام یک ژن مشخص را نشان می‌دهد. همچنین هر اقدام دارای  $N$  وضعیت است. مقدار هر ژن به عنوان یک شیء از مجموعه  $W = \{w_1, w_2, \dots, w_m\}$  انتخاب می‌شود و در هر لحظه به یکی از وضعیت‌های اقدام مربوط اختصاص داده می‌شود. پس از اعمال جستجوی محلی، اگر شیء تخصیص‌یافته به یک اقدام امیدبخش باشد، اتوماتا پاداش می‌گیرد و وضعیت اقدام مربوط به سمت داخلی‌ترین وضعیت حرکت می‌کند و در غیر این صورت جریمه خواهد شد و وضعیت اقدام مربوط به سمت وضعیت مرزی تغییر خواهد کرد. جریمه یا پاداش اتوماتا باعث تغییر قطعیت انتساب یک مقدار به ژن مربوط خواهد شد. به عبارت دیگر وضعیت هر اقدام در هر لحظه، نشان‌دهنده اثر جستجوی محلی یا همان وضعیت ممتیکی آن اقدام است. در این مدل  $F = [f_1, f_2, \dots, f_n]$  بردار شایستگی ژن‌ها است و در نتیجه شایستگی کروموزوم از رابطه  $f(CR) = \sum_{i=1}^n f_i$  حاصل می‌شود.

شکل ۲ کروموزوم "DFABEC" را به وسیله اتوماتای مهاجرت اشیا با اتصالات مشابه اتوماتای ستلین نشان می‌دهد. این کروموزوم شامل شش



شکل ۲: بازنمایی کروموزوم "DFABEC" به وسیله اتوماتای مهاجرت اشیا با اتصالات مشابه اتوماتای ستلین.

جهت حل مسأله مرتب‌سازی ترتیبی در گراف‌ها ارائه شده است. در این مقاله دو عملگر جدید برای جهش و بازترکیبی تعریف گردیده و نشان داده شده که روش ترکیبی میانگین هزینه تور کمتری نسبت به روش‌های ژنتیکی، حریمانه و ... دارد. در [۳۷] و [۳۲] از روش ترکیبی برای حل مسأله فروشنده دوره‌گرد استفاده شده است. در این مقاله‌ها، روش ترکیبی با روش‌های حریمانه و ژنتیک مقایسه شده و کارایی روش ترکیبی نشان داده شده است. همچنین نشان داده شده که استفاده از اتصالات مشابه اتوماتای کرایلو در بازنمایی کروموزوم‌ها، دارای تأثیر بهتری نسبت به اتصالات مشابه اتوماتاهای ستلین، کریسنکی و اومن است. در [۳۳] یک روش ترکیبی مبتنی بر خوشه‌بندی برای حل مسأله فروشنده دوره‌گرد با مقیاس بزرگ پیشنهاد شده است. این روش ابتدا با استفاده از تکنیک خوشه‌بندی، مسأله اصلی را به چند زیرمسأله با مقیاس کوچک افراز کرده و سپس از مدل ترکیبی GALA برای جستجو در فضای حالت و حل هر زیرمسأله استفاده می‌نماید. در این مقاله نشان داده شده که با استفاده از تکنیک خوشه‌بندی و اجرای روش GALA به طور هم‌زمان بر روی هر خوشه (با یک سیستم چندپردازنده‌ای) می‌توان زمان لازم برای حل هر مسأله را به حداقل مقدار ممکن کاهش داد. در [۳۴] از روش GALA برای حل مسأله پیدا کردن مدار هامیلتونی در گراف استفاده شده است. نشان داده شده که روش ترکیبی در مقایسه با روش‌های حریمانه، الگوریتم ژنتیک و اتوماتای یادگیر دارای کارایی بالاتری است. همچنین یک راهکار اکتشافی برای مقاردهی الگوریتم‌های تصادفی ارائه شده است. در [۳۵] و [۳۷] سه الگوریتم تقریبی برای حل مسئله مینیمم کردن پهنای باند در گراف‌ها به کار گرفته شده که به وسیله تغییر دادن ترتیب سطرها و ستون‌های ماتریس مجاورت، باعث کاهش پهنای باند می‌گردند. در [۳۵] و [۳۶] الگوریتم‌های ارائه‌شده با کمک مدل ترکیبی GALA ایجاد شده‌اند و در [۳۷] با الهام از مدل ترکیبی، الگوریتم جدیدی مبتنی بر مکانیزم جریمه و پاداش ارائه شده است. در [۳۵] و [۳۶] سه الگوریتم تقریبی مورد بررسی و مقایسه قرار گرفته‌اند. الگوریتم اول مبتنی بر اتوماتاهای یادگیر مهاجرت اشیا می‌باشد. دومین الگوریتم، یک الگوریتم ترکیبی می‌باشد که از روش ترکیبی GALA حاصل شده است. الگوریتم سوم نیز از ترکیب اتوماتای یادگیر ساختار متغیر و ژنتیک حاصل شده است. این الگوریتم‌ها بر روی ۱۱۳ نمونه از مسأله‌های واقعی ارزیابی شده‌اند و نتایج آن با تعدادی از الگوریتم‌های مشهور مقایسه گردیده که نتایج بهبودیافته‌ای نسبت به چندین مورد از بهترین الگوریتم‌ها دیده

سنتی است و در این قسمت جمعیت کروموزومها قرار دارد. کروموزوم شماره  $i$  با نماد  $CR_i$  به شکل  $CR_i = [CR_i(1) \ CR_i(2) \ \dots \ CR_i(n)]$  نمایش داده می‌شود که  $CR_i(k)$  مقدار تخصیص یافته به ژن  $k$  ام را نشان می‌دهد و بدیهی است که  $1 \leq k \leq n$  و  $CR_i(k) \in W$ . جمعیت اولیه به صورت تصادفی تولید می‌شود و اشیا به صورت تصادفی در وضعیت مرزی اقدام‌های هر اتوماتا قرار می‌گیرند، سپس با انجام عملگرهای جستجوی عمومی و یادگیری محلی در بین وضعیت‌های دیگر جابه‌جا می‌شوند. در هر نسل، کروموزومها از طریق مکانیزم تورنمنت، انتخاب شده و با احتمال  $r_c$  عمل بازترکیبی بر روی آنها انجام می‌گیرد و با توجه به مکانیزم Crowding فرزندان نسل جدید انتخاب می‌شوند و سپس با احتمال  $r_m$  عمل جهش بر روی فرزندان نسل جدید اعمال می‌گردد.

### مکانیزم انتخاب

در مدل یادگیری لامارکی مکانیزم انتخاب کروموزومهای والد و کروموزومهای نسل بعد بر اساس میزان شایستگی فعلی هر کروموزوم است، همچنین در مدل یادگیری بالدوینی، شایستگی بالقوه به دست آمده پس از انجام جستجوی محلی مبنای انتخاب کروموزومها است. در مدل GALA2 از ترکیب شایستگی فعلی و اثر جستجوی محلی جهت انتخاب کروموزومها استفاده می‌شود. میزان قطعیت انتساب یک شیء به اقدام  $\alpha_k$  را که به وسیله جستجوی محلی تغییر می‌کند به صورت عکس شماره وضعیت آن و با نماد  $\gamma_k$  نمایش داده می‌شود. مقدار  $\gamma_k$  در رابطه  $1/N \leq \gamma_k \leq 1$  صدق می‌کند. بردار  $\Gamma = [\gamma_1, \gamma_2, \dots, \gamma_n]$  معرف قطعیت اقدام‌های یک کروموزوم می‌باشد. همچنین تابع شایستگی ترکیبی کروموزوم  $CR$  به صورت  $f'(CR) = F \cdot (1 + \Gamma)^T = \sum_{i=1}^n f_i(1 + \gamma_i)$  برای مسایل بیشینه‌سازی و  $f'(CR) = F \cdot (1 - \Gamma)^T = \sum_{i=1}^n f_i(1 - \gamma_i)$  برای مسایل کمینه‌سازی تعریف می‌شود که در آن  $f_i$  میزان شایستگی ژن شماره  $i$  است. در مدل ترکیبی GALA2 از  $f'(CR)$  جهت مکانیزم انتخاب والدین استفاده می‌شود.

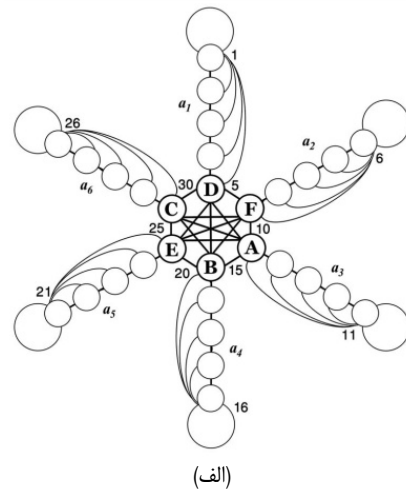
### عملگر جهش

در مدل ارائه شده، عملگر جهش همانند عملگر جهش در الگوریتم ژنتیک سنتی است. ابتدا دو ژن به صورت تصادفی انتخاب می‌گردد و سپس مقادیر آنها با یکدیگر جابه‌جا می‌شوند. با توجه به نوع عملکرد عملگر جهش در تغییر وضعیت ژن‌های انتخابی، دو دسته عملگر جهش را می‌توان تعریف کرد. در دسته اول با اعمال عملگر جهش، وضعیت قبلی (سابقه ممیتیکی) اقدام مربوط حفظ می‌شود ولی در دسته دوم سابقه ممیتیکی اقدام از بین می‌رود. برای دسته اول می‌توان دو نوع عملگر با نام‌های جهش ۱ و جهش ۲ به صورت زیر تعریف نمود:

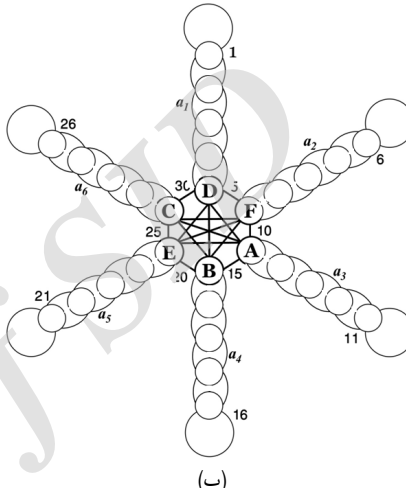
- جهش ۱: عدم تغییر وضعیت قبلی ژن‌ها
  - جهش ۲: انتقال وضعیت قبلی ژن‌ها همراه با مقدار ژن به اقدام جدید
- همچنین برای دسته دوم می‌توان یک عملگر با نام عملگر جهش ۳ به صورت زیر تعریف کرد:
- جهش ۳: فراموش کردن سابقه ممیتیکی و تغییر وضعیت ژن‌ها به وضعیت مرزی

شکل ۴ نحوه تغییر کروموزوم به وسیله عملگر جهش ۱ را نشان می‌دهد. در این شکل مقادیر ژن‌های ۱ و ۲ که به صورت تصادفی انتخاب شده‌اند، بدون تغییر وضعیت اقدام‌ها جابه‌جا شده‌اند.

شکل ۵ نحوه تغییر کروموزوم به وسیله عملگر جهش ۲ را نشان



(الف)



(ب)

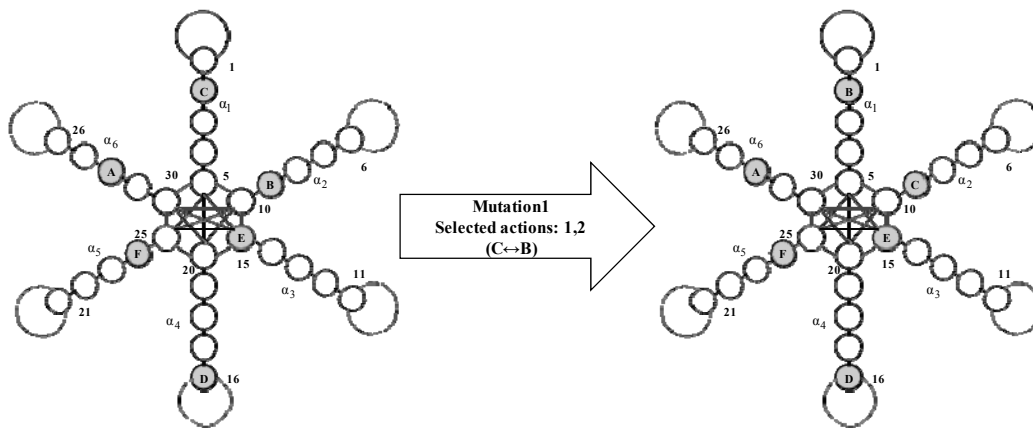
شکل ۳: بازنمایی کروموزوم "DFABEC" به وسیله اتوماتای مهاجرت اشیا، (الف) با اتصالات مشابه اتوماتای کرینسکی و (ب) با اتصالات مشابه اتوماتای کرایلو.

ژن می‌باشد و مقدار هر ژن از مجموعه حروف الفبا انتخاب شده است. این کروموزوم در بازنمایی ترکیبی دارای شش اقدام (ژن) و عمق حافظه پنج است. در این کروموزوم مقدار ژن‌های ۱، ۲ و ۶ به ترتیب برابر با مقادیر D، F و C است که به ترتیب در وضعیت‌های داخلی دوم، سوم و چهارم اقدام مربوط قرار گرفته‌اند. ژن‌های ۳ و ۵ به ترتیب برابر با مقادیر A و E است که هر دو مقدار در وضعیت مرزی اقدام خود قرار گرفته‌اند و در نتیجه دارای حداقل قطعیت ممکن هستند. ژن ۴ دارای مقدار B است و در داخلی‌ترین وضعیت اقدام خود قرار گرفته و در نتیجه دارای حداکثر قطعیت ممکن است. همچنین می‌توان کروموزوم را با سایر اتوماتاها بازنمایی کرد. شکل ۳-الف و ۳-ب کروموزوم "DFABEC" را به ترتیب به وسیله اتوماتای مهاجرت اشیا با اتصالات مشابه اتوماتای کرینسکی و کرایلو نشان می‌دهد.

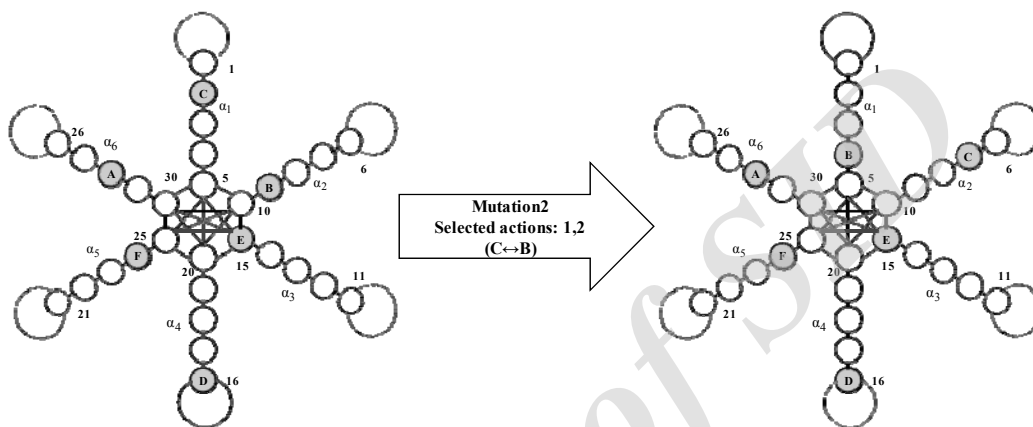
در بازنمایی کروموزوم به وسیله اتوماتای مهاجرت اشیا با اتصالات مشابه اتوماتای کرینسکی (شکل ۳-الف)، وضعیت شیء پس از دریافت پاداش به داخلی‌ترین وضعیت اقدام مربوط تغییر می‌یابد ولی نحوه تغییر وضعیت شیء پس از دریافت جریمه مشابه اتوماتای ستلین است. در بازنمایی کروموزوم به وسیله اتوماتای مهاجرت اشیا با اتصالات مشابه اتوماتای کرایلو (شکل ۳-ب)، وضعیت شیء پس از دریافت پاداش یا جریمه با احتمال ۰/۵ به سمت وضعیت داخلی یا مرزی تغییر می‌یابد.

### ۴-۲ جستجوی عمومی در مدل GALA2

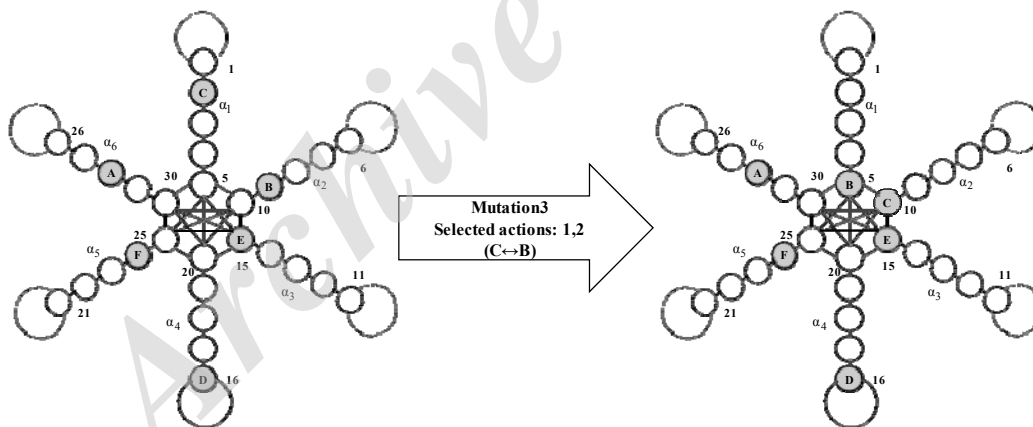
نحوه عملکرد بخش ژنتیکی مدل ارائه شده همانند الگوریتم ژنتیک



شکل ۴: عملگر جهش ۱.



شکل ۵: عملگر جهش ۲.



شکل ۶: عملگر جهش ۳.

می‌نماید. شکل ۷ فرم کلی شبه‌کد عملگر جهش را نشان می‌دهد. لازم به ذکر است که شبه‌کد هر سه نوع عملگر جهش مشابه یکدیگر است و تنها در نحوه تغییر وضعیت اقدام مربوط با هم تفاوت دارند.

**عملگر بازترکیبی**

در مدل ارائه‌شده عملگر بازترکیبی بدین صورت عمل می‌کند: ابتدا دو کروموزوم والد انتخاب می‌شوند و سپس دو اقدام (ژن)  $r_1$  و  $r_2$  به صورت تصادفی در هر دو کروموزوم انتخاب شده و مقدار متناسب به هر اقدام از کروموزوم اول با مقدار متناسب به همان اقدام در کروموزوم دیگر، در صورتی که شایستگی بهتر داشته باشد، تعویض می‌شوند. همانند عملگر جهش، در این عملگر نیز با توجه به نوع عملکرد عملگر بازترکیبی در تغییر وضعیت ژن‌های انتخابی، دو دسته عملگر بازترکیبی می‌توان تعریف

**Procedure Mutation1,2,3 (CR)**

Select two random numbers  $r_1$  and  $r_2$  in  $[1, n]$  where  $r_1 < r_2$   
 Swap (CR.Object (CR.Action( $r_1$ ))), CR.Object (CR.Action( $r_2$ )));  
 UpdateState (CR.Action ( $r_1$ ), CR.Action ( $r_2$ ));  
 End Mutation1,2,3

شکل ۷: شبه‌کد عملگر جهش.

می‌دهد. در این شکل وضعیت قبلی ژن‌ها به اقدام جدید منتقل شده است. شکل ۶ نحوه تغییر کروموزوم به وسیله عملگر جهش ۳ را نشان می‌دهد. در این شکل وضعیت ژن‌ها به وضعیت مرزی اقدام جدید تغییر یافته است.

عملگرهای جهش ۱ و جهش ۲ وضعیت قبلی ژن‌ها را حفظ می‌کنند در حالی که عملگر جهش ۳ سابقه ممتیکی قبلی ژن‌ها را فراموش

بالدوینی) به سمت وضعیت مرزی تغییر می‌کند و در نتیجه قطعیت آن کمتر خواهد شد (شکل ۱۲).

اگر شیء در وضعیت مرزی یک اقدام قرار داشته باشد، جریمه شدن آن باعث تغییر اقدام آن و در نتیجه باعث ایجاد کروموزوم جدیدی می‌شود. تغییر اقدام شیء با استفاده از دانش مسأله و به صورتی که بیشترین کارایی حاصل شود انجام می‌گردد و در این مورد، مقدار ژن عوض می‌شود (یادگیری لامارکی). چگونگی انجام این مورد در شکل ۱۳ نشان داده شده است. چون شیء منتسب به اقدام ۳ (E) در وضعیت مرزی خود بوده است و با توجه به این که از بین سایر اقدامها، تعویض مقدار E و A باعث کسب بیشترین بهره شده است، ابتدا وضعیت اقدام ۶ (A) به وضعیت مرزی تغییر می‌یابد و سپس مقادیر دو اقدام مذکور تعویض می‌شوند. شکل ۱۴ شبه‌کد عملگر جریمه و شکل ۱۵ شبه‌کد مدل GALA۲ را نشان می‌دهد.

### ۵- مسأله تناظر گراف

یک گراف به وسیله عبارت  $G = (E, V)$  نمایش داده می‌شود که  $V$  مجموعه رأسها و  $E \subset V \times V$  مجموعه یال‌ها می‌باشد. دو گراف  $G = (E, V)$  و  $H = (E, V)$  را متناظر گویند اگر و فقط اگر ماتریس‌های مجاورت آنها  $M(G)$  و  $M(H)$  تنها در جابه‌جایی تعدادی از سطرها و ستون‌هایشان با هم متفاوت باشند. وابستگی ماتریس‌های  $M(G)$  و  $M(H)$  تحت نگاشت  $\sigma$  مطابق با (۱) است

$$M(H) = P.M(G).P^T \rightarrow [P.M(G).P^T]_{i,j} = [M(H)]_{\sigma(i),\sigma(j)} \quad (1)$$

به گونه‌ای که  $P$  ماتریس جایگشت نگاشت  $\sigma$  است. اگر اختلاف بین دو گراف را با (۲) نشان دهیم (که در این رابطه  $\| \cdot \|$  نرم ماتریس بوده و مقدار آن از رابطه  $\|M\| = \sum_i \sum_j |m_{ij}|$  به دست می‌آید)، مسأله تناظر گراف به یک مسأله بهینه‌سازی جهت یافتن جایگشت  $\sigma$  به منظور مینیمم کردن  $J(\sigma)$  تبدیل خواهد شد

$$J(\sigma) = \|M(H) - P.M(G).P^T\| \quad (2)$$

خطای تطابق رأس  $k$  در ماتریس  $G$  و رأس  $\sigma(k)$  در ماتریس  $H$  به صورت زیر تعریف می‌شود

$$J_k(\sigma) = \sum_{m=1}^n | [M(H)]_{k,m} - [M(G)]_{\sigma(k),\sigma(m)} | + \sum_{m=1}^n | [M(H)]_{m,k} - [M(G)]_{\sigma(m),\sigma(k)} | \quad (3)$$

که  $n$  تعداد رأس‌های  $G$  گراف است. اگر دو گراف بدون جهت باشند، (۲۱) را می‌توان به صورت زیر نوشت

$$J_k(\sigma) = 2 \times \sum_{m=1}^n | [M(H)]_{k,m} - [M(G)]_{\sigma(k),\sigma(m)} | \quad (4)$$

در نتیجه خطای تناظر گراف را می‌توان به صورت زیر نوشت

$$J(\sigma) = \sum_{k=1}^n J_k(\sigma) \quad (5)$$

ما از رابطه  $f_g = C_{\max} - J(\sigma)$  برای نمایش شایستگی ژنتیکی استفاده نموده‌ایم که در این رابطه  $C_{\max}$  حداکثر مقدار  $J(\sigma)$  است. در این مسأله بازنمایی کروموزوم هر کروموزوم بدین صورت انجام می‌شود که ابتدا یک اتوماتای مهاجرت اشیا با تعداد اقدام‌های (ژن) برابر با تعداد

```

Procedure Crossover 1,2,3 (CR1, CR2)
Select two random numbers r1 and r2 in [1,n] where r1 < r2
For i = r1 to r2 do
  If (fi(CR1) < fi(CR2)) then
    j = Action of CR2 where
      CR2.Object (CR2.Action (j)) = CR1.Object (CR1.Action (i));
      Swap (CR2.Object (CR2.Action (i)), CR2.Object (CR2.Action (j)));
      UpdateState (CR1.Action (i), CR2.Action (j));
  Else
    j = Action of CR1 where
      CR1.Object (CR1.Action (j)) = CR2.Object (CR2.Action (i));
      Swap (CR1.Object (CR1.Action (i)), CR1.Object (CR1.Action (j)));
      UpdateState (CR1.Action (i), CR2.Action (j));
  End If
End For
End Crossover 1,2,3

```

شکل ۸: شبه‌کد عملگر بازترکیبی.

کرد. در دسته اول با اعمال عملگر بازترکیبی، وضعیت قبلی (سابقه ممیتیکی) اقدام مربوط حفظ می‌شود ولی در دسته دوم سابقه ممیتیکی اقدام از بین می‌رود.

برای دسته اول می‌توان دو نوع عملگر را با نام‌های بازترکیبی ۱ و بازترکیبی ۲ به صورت زیر تعریف نمود:

- بازترکیبی ۱: عدم تغییر وضعیت قبلی ژن‌ها
  - بازترکیبی ۲: انتقال وضعیت قبلی ژن‌ها همراه با مقدار ژن به اقدام جدید
- همچنین برای دسته دوم می‌توان یک عملگر بازترکیبی ۳ به صورت زیر تعریف کرد:
- بازترکیبی ۳: فراموش کردن سابقه ممیتیکی و تغییر وضعیت ژن‌ها به وضعیت مرزی

شکل ۸ شبه‌کد عملگر بازترکیبی را نشان می‌دهد.

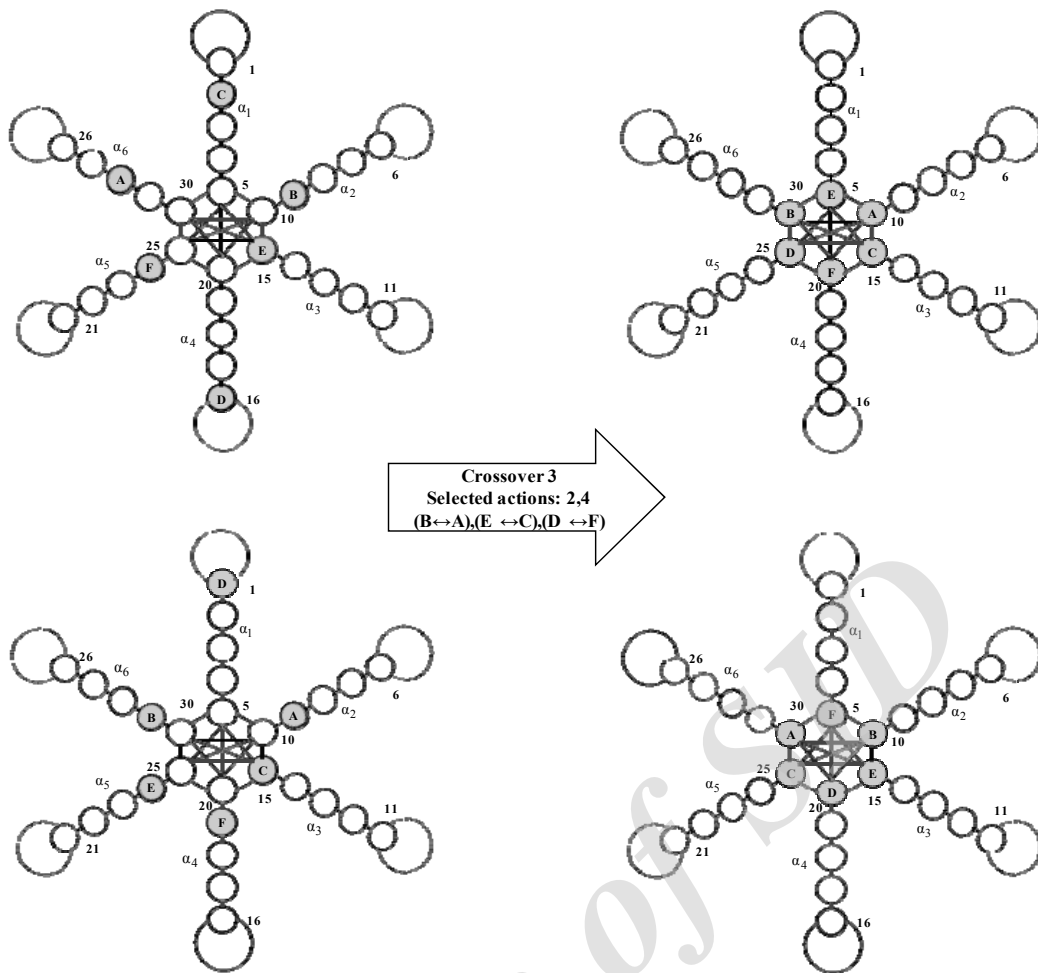
شکل ۹ نحوه عملکرد عملگر بازترکیبی ۳ بدون حفظ سابقه ممیتیکی را نشان می‌دهد. در این مثال ابتدا دو کروموزوم والد (کروموزوم‌های سمت چپ) و سپس دو اقدام (ژن) ۲ و ۴ به صورت تصادفی انتخاب شده‌اند. سپس مقدار منتسب به اقدام شماره ۲ در والد اول (B) با مقدار منتسب به اقدام ۲ در والد دوم (A) تعویض می‌شود. همچنین مقادیر منتسب به اقدام‌های ۳ و ۴ در والد اول (E و D) به ترتیب با مقدار منتسب به اقدام‌های ۳ و ۴ در والد دوم (C و F) تعویض می‌شوند. با توجه به عدم حفظ سابقه ممیتیکی در این دسته از عملگر بازترکیبی، وضعیت اقدام‌هایی که مقدار منتسب به آنها تغییر یافته است، به وضعیت مرزی انتقال می‌یابد.

### ۴-۳ جستجوی محلی در مدل GALA۲

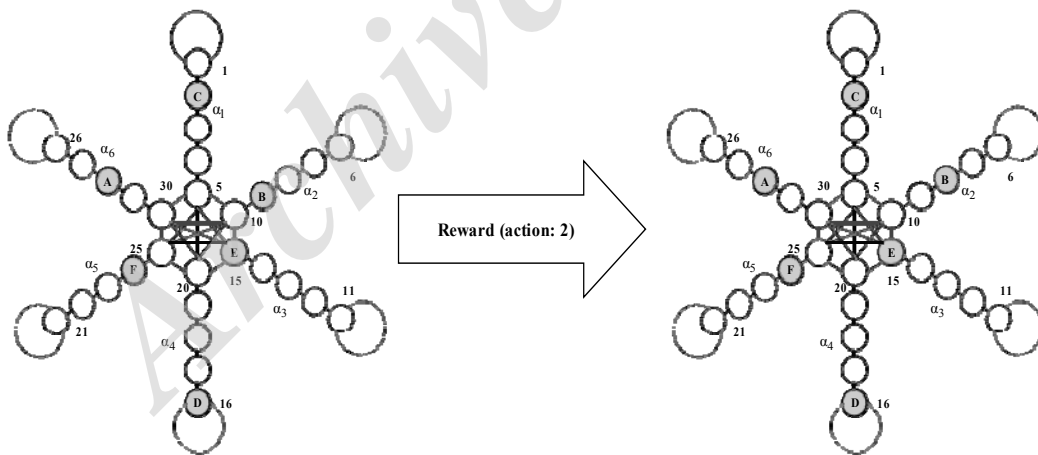
در مدل GALA۲ جستجوی محلی بر عهده اتوماتای یادگیر است. پس از انجام جستجوی محلی بر روی یک کروموزوم، مقدار شیء منتسب به اقدام‌های اتوماتا قبل و بعد از انجام جستجوی محلی با یکدیگر مقایسه می‌شوند. اگر شیء منتسب به یک اقدام (ژن) قبل و بعد از انجام جستجوی محلی یکسان باشند، بدون تغییر مقدار ژن (یادگیری بالدوینی)، آن اقدام پاداش و در غیر این صورت جریمه می‌شود. در اثر پاداش دادن یا جریمه کردن یک اقدام، وضعیت شیء مرتبط با آن اقدام، تغییر می‌کند. اگر اقدام مربوط پاداش بگیرد، وضعیت شیء به سمت وضعیت داخلی‌تر تغییر می‌یابد و در نتیجه قطعیت آن بیشتر خواهد شد. چنانچه شیء در داخلی‌ترین وضعیت خود باشد در همان وضعیت باقی خواهد ماند.

شکل‌های ۱۰ و ۱۱ به ترتیب شبه‌کد و نحوه عملکرد عملگر پاداش را نشان می‌دهند.

اگر اقدام مربوط جریمه شود و شیء منتسب به این اقدام در وضعیت مرزی اقدام خود نباشد، وضعیت شیء بدون تغییر مقدار ژن (یادگیری



شکل ۹: عملگر بازترکیبی بدون حفظ سابقه ممیتیکی.



شکل ۱۱: عملکرد عملگر پاداش بر روی اقدام ۲.

برای انجام جستجوی عمومی در مدل GALA<sub>2</sub> است که در بخش ۴-۲ شرح داده شد.

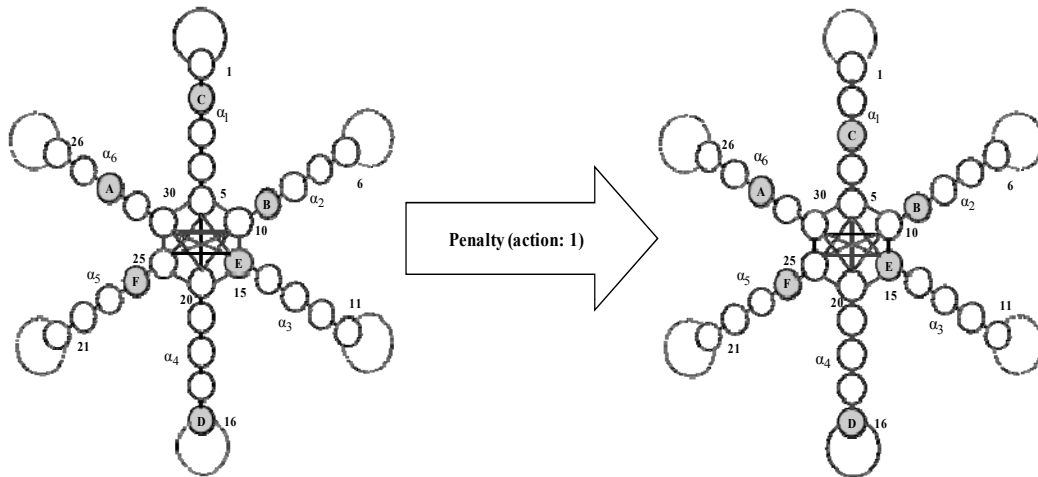
استفاده از دانش مسأله در جستجوی محلی، باعث افزایش کارایی یک الگوریتم ممیتیکی می‌شود. در مسأله تناظر گراف، رأس‌های متناظر بایستی دارای وزن برابر و تعداد یال‌های ورودی و خروجی یکسانی باشند، لذا می‌توان رأس‌های گراف را به دسته‌های هم‌وزن افراز نمود. در این مقاله با توجه به این دانش مسأله جستجوی محلی خاصی ارائه شده است. در جستجوی محلی ارائه‌شده، در هر کروموزوم، ژنی که دارای بیشترین خطای تناظر باشد، انتخاب شده و آن با یکی از رأس‌های هم‌دسته که به صورت تصادفی انتخاب می‌شود، تعویض می‌گردد (شکل ۱۶).

```

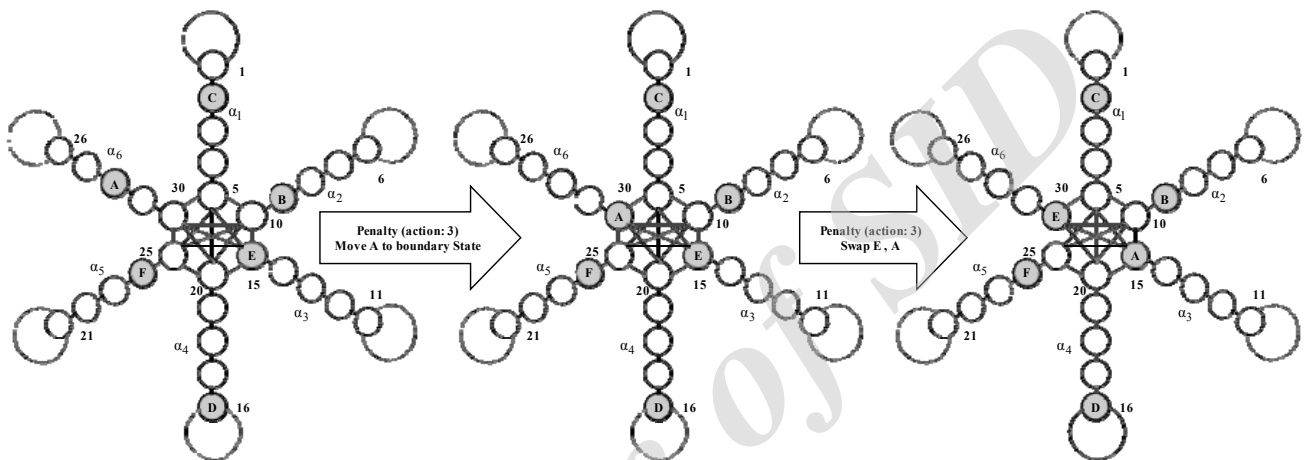
Procedure Reward (CR,u)
If (CR.State (u)-1) mod N <> 0 then
  Dec (CR.State (u));
End If
End Reward
    
```

شکل ۱۰: شبه‌کد عملگر پاداش.

رأس‌های گراف  $G$  ایجاد شده و هر یک از رأس‌های گراف  $H$  به عنوان شیء به یکی از اقدام‌های اتوماتای ایجادشده منتسب می‌گردد. نحوه انجام جستجوی عمومی در این مسأله همانند روش شرح داده شده



شکل ۱۲: عملکرد عملگر جریمه بر روی اقدام ۱.



شکل ۱۳: عملکرد عملگر جریمه بر روی اقدام ۳.

**Function GALA2**

```

t ← 1;
Create the initial Population CR1 ... CRn;
EvaluateHybridFitness ();
While (while termination criteria is not satisfied) do
    NewCR1 = NewCR2 = Select best chromosome CR based on
    EvaluateHybridFitness;
    For i = 2 to n do
        Select CR1, CR2 based on EvaluateHybridFitness;
        If (Random < rc) then
            Crossover (CR1, CR2);
        End If
        If (Random < rm) then
            Mutation (CR1); Mutation (CR2);
        End If
        NewCRi+1 = CR1;
        NewCRi+2 = CR2;
        i = i + 2;
    End For
    For i = 1 to n do
        TempCRi = NewCRi;
        LocalSearch (TempCRi);
        For j = 1 to n do
            If (TempCRi (j) = NewCRi (j)) then
                Reward (NewCRi, u);
            Else
                Penalize (NewCRi, u);
            End If
            CRi = NewCRi;
        End For
    End For
    EvaluateHybridFitness ();
t ← t + 1;
End While
End GALA2
    
```

شکل ۱۵: شبکه‌کد مدل GALA2.

**Procedure Penalize (CR, u)**

```

If (CR.State (U)) mod N <> 0 then
    Inc (CR.State (U));
Else
    bestError = Infinite;
    For U = 1 to n do
        Create chromosome CR' from CR by swapping u and U
        If fU(CR') < bestError Then
            bestError = fU(CR');
            bestNode = U;
        End If
    End For
    CR.State (bestNode) = CR.State (bestNode)*N;
    CR.State (u) = CR.State (u)*N;
    Swap (CR.State (u), CR.State (bestNode));
End If
End Penalize
    
```

شکل ۱۴: شبکه‌کد عملگر جریمه.

**Procedure LocalSearch (CR)**

```

g1 = Worst Gene (CR);
g2 = Select a Random Gene From Same Cluster Members (g1);
Swap (CR (g1), CR (g2));
End LocalSearch;
    
```

شکل ۱۶: شبکه‌کد جستجوی محلی در مسأله تناظر گراف.

**۶- نتایج آزمایش‌ها**

این آزمایش از پایگاه داده‌ای که متشکل از ۱۰۰۰۰ زوج گراف متناظر می‌باشد، استفاده شده و شرح جزئیات این پایگاه داده استاندارد در [۴۴] آمده است. گراف‌های مورد آزمایش در این مقاله شامل ۳ گروه گراف‌های

جهت نمایش کارایی و بررسی تأثیر پارامترهای مختلف مدل GALA2 آزمایش‌های متفاوتی در مسأله تناظر گراف انجام شده و همچنین عملکرد این مدل با سایر روش‌های مشهور حل این مسأله مقایسه گردیده است. در



جدول ۱: مقادیر پارامترهای مختلف مدل GALA۲.

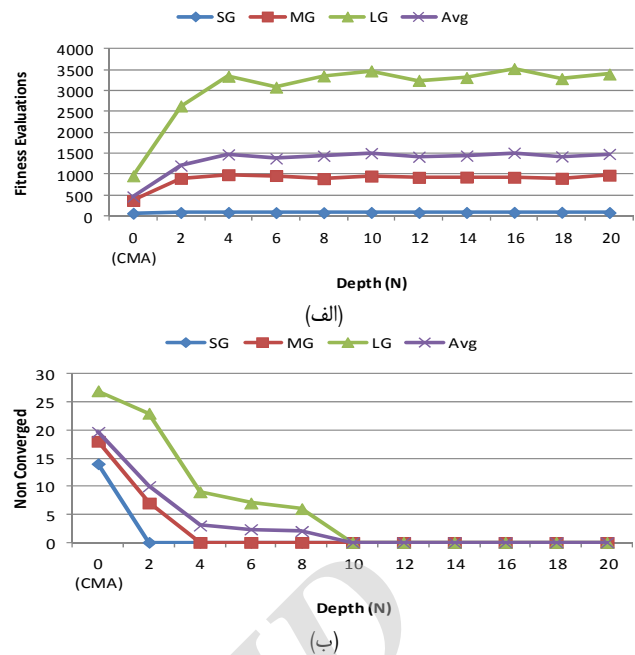
مقادیر پارامترهای مختلف مدل GALA۲	
تعداد افراد جمعیت	۱۰۰
تشکیل جمعیت اولیه	به صورت تصادفی
اندازه کروموزوم	اندازه گراف
نرخ جهش	۰٫۰۵
نرخ بازگزینی	۰٫۰۵
مکانیزم انتخاب	$\mu + \lambda$
شرط خاتمه	رسیدن به مینیمم خطای تناظر یا ۱۰۰۰۰ دور تکرار
الگوریتم یادگیری	Tsetline

جز مدت زمان اجرای الگوریتم در عمق ۲ و ۸ نسبت به مدل GALA۲ برتری دارد، اما تعداد مواردی که در CMA همگرا نشده بسیار بیشتر از مدل ارائه شده است.

## ۶-۲ آزمایش ۲: بررسی کارایی مدل GALA۲ بر روی گرافهای وزن دار و بدون وزن

هدف از انجام این آزمایش بررسی تأثیر وزن یالهای گراف در مدل GALA۲ می باشد. در ماتریس همسایگی گرافهای بدون وزن، وزن همه رأسها صفر و وزن یالها یک یا صفر است. وزن ۱ در هر یال به معنای وجود یال و وزن صفر به معنای عدم وجود یال است. بنابراین هنگام انجام جستجوی محلی، همه رأسها در یک دسته افزاز خواهند شد. در حالی که در گرافهای وزن دار تعداد دستههای افزاز شده بیشتر خواهد بود و لذا امکان انتخاب دقیق تر یک رأس جایگزین در جستجوی محلی بیشتر است. عمق حافظه ژن در همه موارد برابر ۱۰ در نظر گرفته شده و همه گرافها دارای چگالی ۰٫۵ می باشند. متوسط مدت زمان اجرای الگوریتم (RT)، متوسط تعداد فراخوانیهای تابع شایستگی (FE) در موارد همگرا شده و تعداد موارد همگرا نشده (NC) سه پارامتری هستند که در این آزمایش مورد توجه قرار گرفته اند. جدول ۳ نتایج اجرای مدل ارائه شده بر روی گرافهای با وزنهای مختلف را نشان می دهد.

این نتایج، حاصل اجرای الگوریتم GALA۲ برای گرافهای با وزنهای ۱، ۲۰، ۴۰، ۶۰، ۸۰ و ۱۰۰ می باشد. گراف با وزن ۱ معادل گراف بدون وزن است. همان گونه که مشاهده می شود، هر چند مدت زمان اجرای الگوریتم و تعداد فراخوانیهای تابع شایستگی در همه موارد به جز مدت زمان اجرای الگوریتم در گرافهای با وزن ۱۰۰ نسبت به مدل GALA۲ برتری دارد، اما تعداد مواردی که در CMA همگرا نشده است بسیار بیشتر از مدل ارائه شده می باشد. شکل ۱۸-الف متوسط تعداد فراخوانی تابع شایستگی را نشان می دهد. همان گونه که مشاهده می شود، متوسط تعداد فراخوانی تابع شایستگی در همه موارد الگوریتم GALA۲ از CMA بیشتر است اما نکته قابل توجه عدم تغییر متوسط تعداد فراخوانی تابع شایستگی در گرافهای وزن دار است. همچنین همان گونه که در شکل ۱۸-ب نشان داده شده است، در مدل ارائه شده در گرافهای با وزن بزرگتر از ۴۰ همه موارد همگرا شده اند. لذا می توان نتیجه گرفت که مدل GALA۲ در گرافهای وزن دار پایداری بهتری دارد. با توجه به این که در گرافهای بدون وزن، نیمی از یالها در دسته یالهای با وزن ۱ (در صورت وجود یال) و بقیه در دسته یالهای با وزن صفر (در صورت عدم وجود یال) قرار می گیرند، احتمال موفقیت جستجوی محلی در نظر گرفته شده بسیار اندک خواهد بود. لذا به این دلیل مدل ارائه شده در گرافهای بدون وزن در مقایسه با گرافهای وزن دار دارای



شکل ۱۷: مقایسه متوسط (الف) تعداد فراخوانیهای تابع شایستگی و (ب) تعداد موارد همگرا نشده برای گرافهای کوچک، متوسط و بزرگ با توجه به عمق حافظه.

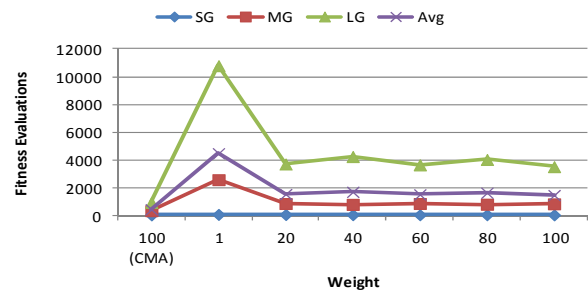
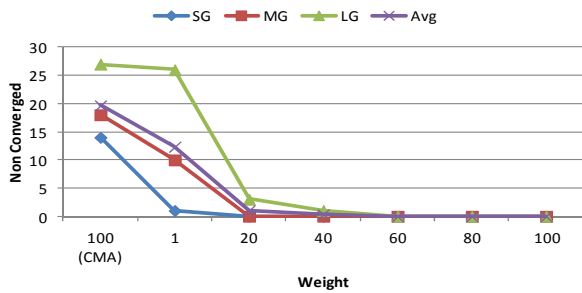
تصادفی با اندازه کوچک ( $n < 50$ )، متوسط ( $50 \leq n < 100$ ) و بزرگ ( $100 \leq n < 200$ ) می باشند. مدل ارائه شده با الگوریتمهای ژنتیک [۴۵]، Ullmann [۴۶]، VF و VF۲ [۴۷] مقایسه شده و کد برنامه نویسی این الگوریتمها در <http://amalfi.dis.unina.it/graph> موجود می باشد. همه نتایج این بخش متوسط ۳۰ تکرار آزمایش است. در این آزمایشها از جدول ۱ جهت مقداردهی پارامترهای مدل GALA۲ استفاده شده است.

## ۶-۱ آزمایش ۱: بررسی عمق حافظه ژن در مدل GALA۲

هدف از انجام این آزمایش بررسی تأثیر عمق حافظه ژن در مدل GALA۲ می باشد. مدل ارائه شده برای بررسی گرافهای تصادفی کوچک (SG)، متوسط (MG) و بزرگ (LG) مورد استفاده قرار گرفته است. وزن رأسها و یالهای هر سه گروه از مجموعه  $[0, 100]$  انتخاب شده اند و همه گرافها دارای چگالی ۰٫۵ می باشند. متوسط مدت زمان اجرای الگوریتم (RT)، متوسط تعداد فراخوانیهای تابع شایستگی (FE) در موارد همگرا شده و تعداد موارد همگرا نشده (NC) سه پارامتری هستند که در این آزمایش مورد توجه قرار گرفته اند.

شکل ۱۷-الف متوسط تعداد فراخوانی تابع شایستگی را نشان می دهد. همان گونه که مشاهده می شود متوسط تعداد فراخوانی تابع شایستگی در همه موارد الگوریتم GALA۲ از CMA بیشتر است. اما نکته قابل توجه عدم تغییر متوسط تعداد فراخوانی تابع شایستگی در عمق بزرگتر از ۴ در هر سه گروه از گرافها است. همچنین همان گونه که در شکل ۱۷-ب نشان داده شده است در مدل ارائه شده، در عمق بزرگتر از ۱۰ همه موارد همگرا شده اند، لذا می توان نتیجه گرفت که مدل GALA۲ در عمق ۱۰ و بیشتر دارای پایداری مناسبی است.

جدول ۲ نتایج اجرای مدل ارائه شده را در عمقهای مختلف حافظه نشان می دهد. این نتایج، حاصل اجرای الگوریتم برای عمقهای ۰ تا ۲۰ می باشد. عمق ۰ در مدل ارائه شده معادل الگوریتم ممتیک متعارف (CMA) است. همان گونه که مشاهده می شود، هر چند مدت زمان اجرای الگوریتم و تعداد فراخوانیهای تابع شایستگی در CMA در همه موارد به



شکل ۱۸: مقایسه متوسط (الف) تعداد فراخوانی‌های تابع شایستگی و (ب) تعداد موارد همگرا نشده برای گراف‌های کوچک، متوسط و بزرگ با توجه به وزن‌های مختلف.

جدول ۲: مقایسه متوسط مدت زمان اجرای الگوریتم (RT)، متوسط تعداد فراخوانی‌های تابع شایستگی (FE) در موارد همگرا شده و تعداد موارد همگرا نشده (NC) در الگوریتم ممیتیک متعارف (CMA) و مدل GALAY در گراف‌های تصادفی کوچک (SG)، متوسط (MG) و بزرگ (LG) با توجه به پارامتر عمق حافظه.

Alg.	Depth (N)	SG			MG			LG			Average		
		FE	RT	NC	FE	RT	NC	FE	RT	NC	FE	RT	NC
CMA	۰	۶۸	۱,۱۷	۱۴	۳۷۳	۱۰,۰۹	۱۸	۹۶۳	۳۶,۰۷	۲۷	۴۶۸	۱۵,۷۷	۲۰
	۲	۸۵	۱,۰۷	۰	۹۰۲	۲۰,۲۰	۷	۲۶۳۴	۸۶,۵۴	۲۳	۱۲۰۷	۳۵,۹۴	۱۰
	۴	۸۶	۱,۰۸	۰	۹۹۰	۲۳,۰۱	۰	۳۳۴۶	۱۱۷,۱۶	۹	۱۴۷۴	۴۷,۰۸	۳
	۶	۸۴	۱,۱۰	۰	۹۶۹	۲۱,۴۲	۰	۳۰۸۵	۱۰۱,۶۲	۷	۱۳۷۹	۴۱,۳۸	۲
	۸	۸۱	۱,۰۷	۰	۸۹۶	۰,۲۰۴	۰	۳۳۵۱	۱۰۹,۸۷	۶	۱۴۴۳	۴۳,۶۶	۲
GALAY	۱۰	۸۷	۱,۱۳	۰	۹۵۷	۲۱,۱۷	۰	۳۴۷۳	۱۱۴,۰۹	۰	۱۵۰۶	۴۵,۴۷	۰
	۱۲	۸۷	۱,۱۴	۰	۹۲۶	۲۰,۴۲	۰	۳۲۴۶	۱۰۶,۵۸	۰	۱۴۲۰	۴۲,۷۱	۰
	۱۴	۸۶	۱,۱۴	۰	۹۳۳	۲۰,۵۴	۰	۳۳۱۴	۱۰۸,۱۴	۰	۱۴۴۴	۴۳,۲۷	۰
	۱۶	۸۹	۱,۱۵	۰	۹۲۷	۲۰,۴۹	۰	۳۵۲۹	۱۱۵,۵۰	۰	۱۵۱۵	۴۵,۷۱	۰
	۱۸	۹۰	۱,۱۸	۰	۹۰۳	۱۹,۸۶	۰	۳۲۸۹	۱۰۷,۷۵	۰	۱۴۲۸	۴۲,۹۳	۰
	۲۰	۸۴	۱,۱۰	۰	۹۷۷	۲۱,۴۸	۰	۳۳۹۷	۱۱۱,۱۱	۰	۱۴۸۶	۴۴,۵۶	۰

جدول ۳: مقایسه متوسط مدت زمان اجرای الگوریتم (RT)، متوسط تعداد فراخوانی‌های تابع شایستگی (FE) در موارد همگرا شده و تعداد موارد همگرا نشده (NC) در الگوریتم ممیتیک متعارف (CMA) و مدل GALAY در گراف‌های تصادفی کوچک (SG)، متوسط (MG) و بزرگ (LG) با توجه به پارامتر وزن گراف.

Alg.	Weight	SG			MG			LG			Average		
		FE	RT	NC	FE	RT	NC	FE	RT	NC	FE	RT	NC
CMA	۱۰۰	۶۸	۱,۱۷	۱۴	۳۷۳	۱۰,۰۹	۱۸	۹۶۳	۳۶,۰۷	۲۷	۴۶۸	۱۵,۷۷	۲۰
	۱	۱۰۲	۱,۲۹	۱	۲۶۰۱	۵۹,۱۴	۱۰	۱۰۷۹۱	۳۵۵,۸۴	۲۶	۴۴۹۸	۱۳۸,۷۶	۱۲
	۲۰	۹۹	۱,۲۷	۰	۸۸۳	۱۹,۵۴	۰	۳۷۲۳	۱۲۱,۳۳	۳	۱۵۶۹	۴۷,۳۸	۱
	۴۰	۸۲	۱,۰۴	۰	۸۲۳	۱۸,۲۲	۰	۴۲۶۲	۱۳۹,۷۳	۱	۱۷۲۲	۵۲,۹۹	۰
	۶۰	۷۶	۰,۹۷	۰	۸۹۲	۱۹,۶۸	۰	۳۶۶۵	۱۱۹,۶۹	۰	۱۵۴۴	۴۶,۷۸	۰
	۸۰	۸۱	۱,۰۳	۰	۸۳۵	۱۸,۴۴	۰	۴۰۵۲	۱۳۰,۵۱	۰	۱۶۵۶	۴۹,۹۹	۰
	۱۰۰	۷۴	۰,۹۵	۰	۸۶۵	۱۹,۰۹	۰	۳۵۳۵	۱۱۴,۹۴	۰	۱۴۹۱	۴۴,۹۹	۰

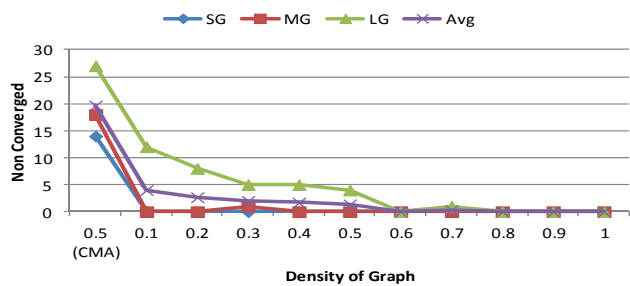
کارایی مناسبی نمی‌باشد.

### ۳-۶ آزمایش ۳: بررسی تأثیر چگالی گراف بر کارایی مدل GALAY

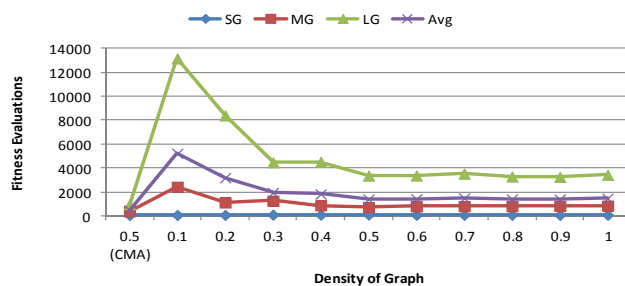
چگالی‌های مختلف را نشان می‌دهد. این نتایج، حاصل اجرای الگوریتم برای چگالی‌های ۰/۱، ۰/۲، ۰/۳، ۰/۴، ۰/۵، ۰/۶، ۰/۷، ۰/۸، ۰/۹ و ۱ می‌باشد. همان گونه که مشاهده می‌شود، هرچند مدت زمان اجرای الگوریتم و تعداد فراخوانی‌های تابع شایستگی در CMA در همه موارد نسبت به مدل GALAY برتری دارد، اما تعداد مواردی که در CMA همگرا نشده بسیار بیشتر از مدل ارائه شده است.

شکل ۱۹- الف متوسط تعداد فراخوانی تابع شایستگی را نشان می‌دهد. همان گونه که مشاهده می‌شود، متوسط تعداد فراخوانی تابع شایستگی مدل ارائه شده در همه موارد از CMA بیشتر است. اما نکته قابل توجه عدم تغییر متوسط تعداد فراخوانی تابع شایستگی در گراف‌های با چگالی بالاتر ۰/۵ است. همچنین همان گونه که در شکل ۱۹- ب نشان داده شده است، در مدل ارائه شده، در گراف‌های با چگالی بزرگ‌تر از ۰/۶ همه موارد همگرا شده‌اند. لذا می‌توان نتیجه گرفت که مدل GALAY در گراف‌های

هدف از انجام این آزمایش بررسی تأثیر میزان چگالی گراف بر کارایی مدل GALAY می‌باشد. چگالی گراف از  $D = 2|E|/(|V|(|V|-1))$  به دست می‌آید و به عبارت دیگر، چگالی گراف احتمال وجود یک یال بین دو رأس را نشان می‌دهد. در این آزمایش عمق حافظه ژن در همه موارد برابر ۱۰ در نظر گرفته شده و وزن یال‌ها از مجموعه  $[0, 100]$  انتخاب گردیده است. متوسط مدت زمان اجرای الگوریتم (RT)، متوسط تعداد فراخوانی‌های تابع شایستگی (FE) در موارد همگرا شده و تعداد موارد همگرا نشده (NC) سه پارامتری هستند که در این آزمایش مورد توجه قرار گرفته‌اند. جدول ۴ نتایج اجرای مدل ارائه شده بر روی گراف‌های با



(ب)



(الف)

شکل ۱۹: مقایسه متوسط (الف) تعداد فراخوانی‌های تابع شایستگی و (ب) تعداد موارد همگرانشده برای گراف‌های کوچک، متوسط و بزرگ با توجه به پارامتر چگالی گراف.

جدول ۴: مقایسه تأثیر چگالی گراف بر کارایی مدل GALA۲.

Alg.	Density	SG			MG			LG			Average			
		FE	RT	NC	FE	RT	NC	FE	RT	NC	FE	RT	NC	
CMA	۰.۵	۶۸	۱,۱۷	۱۴	۳۷۳	۱۰,۰۹	۱۸	۹۶۳	۳۶,۰۶	۲۷	۴۶۸	۱۵,۷۷	۲۰	
	۰.۱	۶۵	۰,۸۹	۰	۲۴۲۸	۵۶,۰۵	۰	۱۳۱۶۶	۴۳۸,۰۷	۱۲	۵۲۲۰	۱۶۵,۰۰	۴	
	۰.۲	۷۳	۱,۰۱	۰	۱۱۱۴	۲۵,۳۲	۰	۸۳۹۶	۲۷۶,۶۲	۸	۳۱۹۴	۱۰۰,۹۹	۳	
	۰.۳	۹۱	۱,۲۱	۰	۱۲۶۳	۲۸,۳۵	۱	۴۴۷۰	۱۴۶,۴۲	۵	۱۹۴۱	۵۸,۶۶	۲	
	۰.۴	۸۹	۳,۸۱	۰	۸۶۸	۴۷,۶۵	۰	۴۵۱۶	۲۳۳,۲۵	۵	۱۸۲۴	۹۴,۹۰	۲	
	GALA۲	۰.۵	۹۶	۱,۲۷	۰	۷۲۵	۱۶,۳۵	۰	۳۳۵۷	۱۱۰,۵۲	۴	۱۳۹۳	۴۲,۷۱	۱
		۰.۶	۷۷	۱,۰۴	۰	۷۹۰	۱۷,۶۶	۰	۳۳۶۹	۱۱۱,۳۱	۰	۱۴۱۲	۴۳,۳۴	۰
		۰.۷	۷۸	۱,۰۲	۰	۸۰۹	۱۸,۱۸	۰	۳۵۳۰	۱۱۵,۸۶	۱	۱۴۷۲	۴۵,۰۲	۰
		۰.۸	۸۷	۱,۰۹	۰	۸۰۷	۱۷,۸۱	۰	۳۲۸۴	۱۰۷,۱۱	۰	۱۳۹۳	۴۲,۰۰	۰
		۰.۹	۷۳	۰,۹۲	۰	۸۲۳	۱۸,۱۷	۰	۳۲۵۹	۱۰۵,۹۲	۰	۱۳۸۵	۴۱,۶۷	۰
۱		۷۶	۰,۹۵	۰	۸۴۰	۱۹,۹۳	۰	۳۴۲۴	۱۱۴,۰۱	۰	۱۴۴۷	۴۴,۹۶	۰	

جدول ۵: بررسی تأثیر نحوه انتقال سابقه ممتیکی بر کارایی مدل GALA۲.

Alg.	Memory History Type	SG			MG			LG			Average		
		FE	RT	NC	FE	RT	NC	FE	RT	NC	FE	RT	NC
CMA	-	۶۸	۱,۱۷	۱۴	۳۷۳	۱۰,۰۹	۱۸	۹۶۳	۳۶,۰۶	۲۷	۴۶۸	۱۵,۷۷	۲۰
	SH	۹۰	۱,۱۵	۰	۹۵۱	۲۱,۰۰	۰	۳۳۷۵	۱۰۹,۱۶	۱	۱۴۷۲	۴۳,۷۷	۰
GALA۲	TH	۸۶	۱,۱۴	۰	۹۲۷	۲۰,۲۵	۰	۳۷۵۲	۱۲۰,۹۵	۴	۱۵۸۸	۴۷,۴۵	۱
	FH	۹۲	۱,۱۳	۰	۹۲۸	۲۰,۱۳	۰	۴۰۴۳	۱۲۸,۶۵	۰	۱۶۸۸	۴۹,۹۷	۰

مدت زمان اجرای الگوریتم و تعداد فراخوانی‌های تابع شایستگی در CMA در همه موارد نسبت به مدل GALA۲ برتری دارد، اما تعداد مواردی که در CMA همگرا نشده بسیار بیشتر از مدل ارائه شده است. نتایج این آزمایش تأیید دیگری بر نظریه بالدوین است.

#### ۵-۶ آزمایش ۵: مقایسه مدل GALA۲ با سایر

##### الگوریتم‌های تناظر گراف

هدف از انجام این آزمایش بررسی عملکرد الگوریتم GALA۲ برای حل مسأله تناظر گراف در مقایسه با سایر روش‌های مشهور حل این مسأله می‌باشد. شکل ۲۰ متوسط تعداد فراخوانی تابع شایستگی در اجراهای همگرا شده در الگوریتم‌های GA، VF، VF۲، Ullmann، GALA و GALA۲ در ۳۰ اجرای مختلف را نشان می‌دهد. در این آزمایش از گراف‌های با تعداد رأس ۱۰ تا ۲۰۰ رأس استفاده شده است. در همه موارد الگوریتم ارائه شده دارای عملکرد بهتری نسبت به سایر روش‌ها است اما در گراف‌های کوچک و متوسط، به دلیل محدودتر بودن فضای جستجو، الگوریتم GALA۲ اختلاف بیشتری با سایر روش‌ها دارد.

با چگالی بالاتر از ۰/۵ پایداری بهتری دارد.

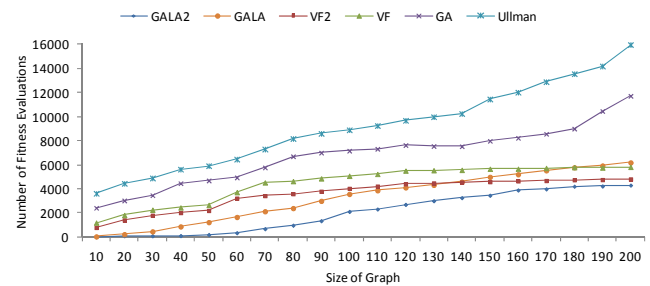
#### ۶-۴ آزمایش ۴: بررسی تأثیر نحوه انتقال سابقه

##### ممتیکی بر کارایی مدل GALA۲

هدف از انجام این آزمایش بررسی تأثیر نحوه انتقال سابقه ممتیکی بر کارایی مدل GALA۲ می‌باشد. عمق حافظه ژن در همه موارد برابر ۱۰ در نظر گرفته شده و همه گراف‌ها دارای چگالی ۰/۵ بوده و وزن یال‌ها از مجموعه [۰, ۱۰۰] انتخاب شده است. متوسط مدت زمان اجرای الگوریتم (RT)، متوسط تعداد فراخوانی‌های تابع شایستگی (FE) در موارد همگرا شده و تعداد موارد همگرانشده (NC) سه پارامتری هستند که در این آزمایش مورد توجه قرار گرفته‌اند.

جدول ۵ نتایج اجرای مدل ارائه شده در سه حالت انتقال سابقه ممتیکی هنگام انجام عملگرهای جهش و ترکیبی را نشان می‌دهد. این سه حالت عبارت از حفظ سابقه قبلی هر ژن (SH)، انتقال سابقه قبلی یک ژن به همراه مقدار ژن (TH) و فراموش کردن سابقه ممتیکی ژن (FH) است. همان گونه که مشاهده می‌شود در حالت فراموش کردن سابقه ممتیکی ژن، همه موارد همگرا شده‌اند و باعث پایداری بیشتر خواهد شد. هرچند

- [11] M. Rezapoor Mirsaleh and M. R. Meybodi, "A learning automata-based memetic algorithm," *Genetic Programming and Evolvable Machines*, vol. 16, no. 4, pp. 399-453, Dec. 2015.
- [12] M. Rezapoor Mirsaleh and M. R. Meybodi, "Balancing exploration and exploitation in memetic algorithms: a learning automata approach," *Computational Intelligence*, vol. 34, no. 1, pp. 282-309, Feb. 2018.
- [13] A. Rezvanian and M. R. Meybodi, "A new learning automata-based sampling algorithm for social networks," *International J. of Communication Systems*, vol. 3, no. 5, pp. 1-21, Mar. 2015.
- [14] M. Ghavipour and M. R. Meybodi, "Irregular cellular learning automata-based algorithm for sampling social networks," *Engineering Applications of Artificial Intelligence*, vol. 59, pp. 244-259, Mar. 2017.
- [15] M. Rezapoor Mirsaleh and M. R. Meybodi, "A michigan memetic algorithm for solving the community detection problem in complex network," *Neurocomputing*, vol. 214, pp. 535-545, 19 Nov. 2016.
- [16] B. Moradabadi and M. R. Meybodi, "Link prediction based on temporal similarity metrics using continuous action set learning automata," *Physica A: Statistical Mechanics and its Applications*, vol. 160, no. 2, pp. 361-373, Feb. 2016.
- [17] M. Rezapoor Mirsaleh and M. R. Meybodi, "A new memetic algorithm based on cellular learning automata for solving the vertex coloring problem," *Memetic Computing*, vol. 8, no. 3, pp. 211-222, Sept. 2016.
- [18] M. Rezapoor Mirsaleh and M. R. Meybodi, "A michigan memetic algorithm for solving the vertex coloring problem," *J. of Computational Science*, vol. 24, no. 2, pp. 389-401, Oct. 2017.
- [19] S. M. Vahidipour, M. R. Meybodi, and M. Esnaashari, "Finding the shortest path in stochastic graphs using learning automata and adaptive stochastic petri nets," *International J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 3, pp. 427-455, Jan. 2017.
- [20] S. M. Vahidipour, M. R. Meybodi, and M. Esnaashari, "Cellular adaptive petri net based on learning automata and its application to the vertex coloring problem," *Discrete Event Dynamic Systems*, vol. 27, no. 4, pp. 609-640, May. 2017.
- [21] A. M. Saghiri and M. R. Meybodi, "A self-adaptive algorithm for topology matching in unstructured peer-to-peer networks," *J. of Network and Systems Management*, vol. 24, no. 2, pp. 393-426, Apr. 2016.
- [22] A. M. Saghiri and M. R. Meybodi, "An approach for designing cognitive engines in cognitive peer-to-peer networks," *J. of Network and Computer Applications*, vol. 70, pp. 17-40, Jul. 2016.
- [23] M. Ghavipour and M. R. Meybodi, "An adaptive fuzzy recommender system based on learning automata," *Electronic Commerce Research and Applications*, vol. 20, pp. 105-115, Nov./Dec. 2016.
- [24] B. J. Oommen and D. C. Y. Ma, "Deterministic learning automata solutions to the equipartitioning problem," *IEEE Trans. on Computers*, vol. 37, no. 1, pp. 2-13, Jan. 1988.
- [25] M. Rezapoor Mirsaleh and M. R. Meybodi, "Improving GA+LA algorithm for solving graph isomorphic problem," in *Proc. of the 11th Annual CSI Computer Conf. of Iran*, pp. 474-483, Tehran, Iran, Jan. 2006.
- [26] K. Asghari, A. Safari Mamaghani, and M. R. Meybodi, "An evolutionary approach for query optimization problem in database," in *Proc. of Int. Joint Conf. on Computers, Information and System Sciences, and Engineering, CISSE'07*, University of Bridgeport, England, 9 pp., Dec. 2007.
- [27] A. Safari Mamaghani, K. Asgari, M. R. Meybodi, and F. Mahmoodi, "A new method based on genetic algorithm for minimizing join operations cost in data base," in *Proc. of 13th Annual CSI Computer Conf. of Iran*, 6 pp., Kish Island, Iran, Mar. 2008.
- [28] A. Safari Mamaghani, K. Asghari, F. Mahmoodi, and M. R. Meybodi, "A novel hybrid algorithm for joint ordering problem in database queries," in *Proc. of 6th WSEAS Int. Conf. on Computational Intelligence, Man-Machine Systems and Cybernetics*, pp. 104-109, Tenerife, Spain, 14-16 Dec. 2007.
- [29] K. Asghari, A. Safari Mamaghani, F. Mahmoodi, and M. R. Meybodi, "A relational databases query optimization using hybrid evolutionary algorithm," *J. of Computer and Robotics*, vol. 1, no. 1, pp. 28-39, Jan. 2008.
- [30] B. Zaree and M. R. Meybodi, "A hybrid method for sorting problem," in *Proc. of the 3rd Int. Conf. on Information and Knowledge Technology, IKT'07*, 8 pp., Mashhad, Iran, 26-28 Nov. 2007.
- [31] B. Zaree and M. R. Meybodi, "An evolutionary method for solving symmetric TSP," in *Proc. of the 3rd Int. Conf. on Information and Knowledge Technology, IKT'07*, 8 pp., Mashhad, Iran, 26-28 Nov. 2007.



شکل ۲۰: مقایسه الگوریتم GALA2 و سایر الگوریتم‌های تناظر گراف.

## ۷- نتیجه گیری

در این مقاله یک مدل جدید مبتنی بر اتوماتای یادگیر ارائه شد. در این روش برای استفاده هم‌زمان از اثرات مثبت جستجوی عمومی و جستجوی محلی، از اتوماتای مهاجرت اشیا برای بازنمایی کروموزوم‌ها استفاده گردید. در این نوع بازنمایی علاوه بر نگهداری مقادیر ژنتیکی، اثر جستجوی محلی در حافظه هر ژن ذخیره می‌شود. به عبارت دیگر شایستگی کروموزوم‌ها در مدل GALA2 از ترکیب شایستگی ژنتیکی کروموزوم که محصول جستجوی عمومی است و سابقه جستجوی محلی که در وضعیت‌های اتوماتای مهاجرت اشیا ذخیره می‌شود به دست می‌آید. جهت نشان دادن کارایی مدل ارائه‌شده، از این مدل در حل مسأله تناظر گراف استفاده شد و با سایر روش‌های معروف حل این مسأله مورد مقایسه قرار گرفت. نتایج به دست آمده از آزمایش‌ها، برتری روش ارائه‌شده نسبت به سایر روش‌ها را نشان می‌دهد. همچنین پارامترهای مختلف این مدل در آزمایش‌های گوناگونی بررسی گردید. توسعه این مدل جهت استفاده از آن در محیط پویا و ارائه یک چارچوب ریاضی برای تحلیل مدل پیشنهادی از جمله کارهایی است که در آینده می‌تواند انجام شود.

## مراجع

- [1] C. Xianshun, O. Yew-Soon, L. Meng-Hiot, and T. Kay Chen, "A multi-facet survey on memetic computation," *IEEE Trans. on Evolutionary Computation*, vol. 15, no. 5, pp. 591-607, Oct. 2011.
- [2] Q. H. Nguyen, Y. S. Ong, and N. Krasnogor, "A study on the design issues of memetic algorithm," in *Proc. CEC IEEE Congress on Evolutionary Computation*, pp. 2390-2397, Singapore, Singapore, 25-28 Sept. 2007.
- [3] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 5, pp. 474-488, Oct. 2005.
- [4] J. Hong and V. Prabhu, "Distributed reinforcement learning control for batch sequencing and sizing in just-in-time manufacturing systems," *Applied Intelligence*, vol. 20, no. 1, pp. 71-87, Jan. 2004.
- [5] M. A. Wiering and H. van Hasselt, "Ensemble algorithms in reinforcement learning," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 4, pp. 930-936, Aug. 2008.
- [6] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: an Introduction*, Prentice-Hall, Inc., 1989.
- [7] M. A. L. Thathachar and P. S. Sastry, "Varieties of learning automata: an overview," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 6, pp. 711-722, Dec. 2002.
- [8] K. Najim and A. S. Poznyak, *Learning Automata: Theory and Applications*, Pergamon Press, Inc., 1994.
- [9] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata: Techniques for Online Stochastic Optimization*, Kluwer Academic Publishers, 2004.
- [10] M. Rezapoor Mirsaleh and M. R. Meybodi, "A hybrid algorithm for solving graph isomorphism problem," in *Proc. of the 2nd Int. Conf. on Information and Knowledge Technology, IKT'05*, 13 pp., Tehran, Iran, 24-26 May 2005.

- [43] A. Bansal and R. Kaur, "Task graph scheduling on multiprocessor system using genetic algorithm," *International J. of Engineering Research & Technology*, vol. 1, no. 5, pp. 1-5, Jan. 2012.
- [44] P. Foggia, C. Sansone, and M. Vento, "A database of graphs for isomorphism and sub-graph isomorphism benchmarking," in *Proc. of the 3rd IAPR TC-15 Int. Workshop on Graph-Based Representations*, pp. 176-187, May 2001.
- [45] W. Yuan-Kai, F. Kuo-Chin, and H. Jorng-Tzong, "Genetic-based search for error-correcting graph isomorphism," *IEEE Trans. on, Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 4, pp. 588-597, Aug. 1997.
- [46] J. R. Ullmann, "An algorithm for subgraph isomorphism," *J. ACM*, vol. 23, no. 1, pp. 31-42, Jan. 1976.
- [47] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub) graph isomorphism algorithm for matching large graphs," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1367-1372, Oct. 2004.
- [32] B. Zaree and M. R. Meybodi, "A hybrid method for solving traveling salesman problem," in *Proc. of the 6th IEEE/ACIS Int. Conf. on Computer and Information Science, ICIS'07*, pp. 394-399, Melbourne, Australia, 11-13 Jul. 2007.
- [33] B. Zaree, K. Asghari, and M. R. Meybodi, "A hybrid method based on clustering for solving large traveling salesman problem," in *Proc. of 13th Annual CSI Computer Conf. of Iran*, 6 pp., Kish Island, Iran, 9-11 Mar. 2008.
- [34] K. Asghari and M. R. Meybodi, "Searching for hamiltonian cycles in graphs using evolutionary methods," in *Proc. of the 2nd Joint Congress on Fuzzy and Intelligent Systems*, pp. 132-143, Tehran, Iran, 28-30 Oct. 2008.
- [35] A. Safari Mamaghani and M. R. Meybodi, "Hybrid algorithms (learning automata + genetic algorithm) for solving graph bandwidth minimization problem," in *Proc. of the 2nd Joint Congress on Fuzzy and Intelligent Systems*, 1 pp., Tehran, Iran, 28-30 Oct. 2008.
- [36] A. S. Mamaghani and M. R. Meybodi, "A learning automaton based approach to solve the graph bandwidth minimization problem," in *Proc. 5th Int. Conf. on Application of Information and Communication Technologies, AICT'11*, 5 pp., Baku, Azerbaijan 12-14 Oct. 2011.
- [37] A. Isazadeh, H. Izadkhan, and A. Mokarram, "A learning based evolutionary approach for minimization of matrix bandwidth problem," *Appl. Math*, vol. 6, no. 1, pp. 51-57, Apr. 2012.
- [38] A. Safari Mamaghani and M. R. Meybodi, "Hybrid evolutionary algorithms for solving software clustering problem," in *Proc. of the 2nd Joint Congress on Fuzzy and Intelligent Systems*, pp. 464-473, Tehran, Iran, Oct. 2008.
- [39] K. Asghari and M. R. Meybodi, "Solving single machine total weighted tardiness scheduling problem using learning automata and genetic algorithm," in *Proc. of the 3rd Iran Data Mining Conf., IDMC'09*, Tehran, Iran, 16 pp., Dec. 2009.
- [40] A. S. Mamaghani, M. Mahi, and M. R. Meybodi, "A learning automaton based approach for data fragments allocation in distributed database systems," in *Proc. IEEE 10th Int. Conf. on Computer and Information Technology, CIT'10*, pp. 8-12, Jun. 2010.
- [41] A. S. Mamaghani, M. Mahi, M. R. Meybodi, and M. H. Moghaddam, "A novel evolutionary algorithm for solving static data allocation problem in distributed database systems," in *Proc. 2nd Int. Conf. on Network Applications Protocols and Services, NETAPPS'10*, pp. 14-19, Kedah, Malaysia, 22-23 Sept. 2010.
- [42] V. Majid Nezhad, H. Motee Gader, and E. Efimov, "A novel hybrid algorithm for task graph scheduling," *IJCSI International J. of Computer Science Issues*, vol. 8, no. 1, pp. 32-38, Nov. 2011.

**مهدی رضاپور میرصالح** در سال ۱۳۷۹ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه خوارزمی و در سال ۱۳۸۲ مدرک کارشناسی ارشد خود را در این رشته از دانشگاه صنعتی امیرکبیر (پلی تکنیک) تهران دریافت نمود. از سال ۱۳۸۷ نامبرده به عنوان عضو هیأت علمی در دانشگاه پیام نور مشغول به کار بوده و در سال ۱۳۹۵ موفق به اخذ درجه دکترا در رشته مهندسی کامپیوتر از دانشگاه صنعتی امیرکبیر (پلی تکنیک) تهران گردیده است. زمینه‌های علمی مورد علاقه نامبرده متنوع بوده و شامل موضوعاتی مانند محاسبات نرم و کاربردهای آن، الگوریتم‌های تکاملی، شبکه‌های اجتماعی و بانکداری الکترونیکی می‌باشد.

**محمد رضا میبیدی** تحصیلات خود را در مقاطع کارشناسی و کارشناسی ارشد اقتصاد به ترتیب در سال‌های ۱۳۵۲ و ۱۳۵۶ از دانشگاه شهید بهشتی و در مقاطع کارشناسی ارشد و دکتری علوم کامپیوتر به ترتیب در سال‌های ۱۳۵۹ و ۱۳۶۲ از دانشگاه اوکلاهامای آمریکا به پایان رسانده و هم‌اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر می‌باشد. نامبرده قبل از پیوستنش به دانشگاه صنعتی امیرکبیر در سال‌های ۱۳۶۲ تا ۱۳۶۴ استادیار دانشگاه میشیگان غربی و در سال‌های ۱۳۶۴ تا ۱۳۷۰ دانشیار دانشگاه اوهایو در ایالات متحده آمریکا بوده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: الگوریتم‌های موازی، پردازش موازی، محاسبات نرم و کاربردهای آن، شبکه‌های کامپیوتری و مهندسی نرم‌افزار.