

استراتژی‌های جدید حافظ تنوع برای الگوریتم ژنتیک و کاربرد آن برای بهینه‌سازی مقیاس بزرگ

حسن اسمخان^۱، کارشناسی ارشد

۱- دانشکده فنی و مهندسی - دانشگاه بناب - بناب - ایران - h.ismkhan@bonabu.ac.ir

چکیده: جهت افزایش کارایی الگوریتم ژنتیک، رویکردهای فراوانی باهدف حفظ تنوع منتشر شده است. باوجود این، اکثر این رویکردها فقط می‌توانند روی مسائل بهینه‌سازی پیوسته اعمال شوند. این بدان معنا نیست که الگوریتم‌های ژنتیک در حل مسائل بهینه‌سازی گسسته به تنوع جمعیت نیاز ندارند. در حقیقت، تعریف مفهوم تفاوت بین جواب‌های راه‌حل‌های مسائل بهینه‌سازی گسسته، با توجه به تفاوت ظاهری آن‌ها ساده نیست. برای مثال در مسئله فروشنده دوره‌گرد، چگونه باید تشابه بین دو جواب را سنجید. این مقاله استراتژی‌های حافظ تنوعی برای الگوریتم ژنتیک ارائه می‌دهد که بر پایه تشابه بین دو جواب استوارند. این استراتژی‌ها نه تنها می‌توانند روی مسائل بهینه‌سازی پیوسته اعمال شوند، بلکه با پیشنهاد راهکارهای جدید معناگرا برای محاسبه تشابه بین جواب‌های مسائل بهینه‌سازی گسسته، اعمال موفقیت‌آمیز آن روی مسائل بهینه‌سازی گسسته نیز امکان‌پذیر است.

واژه‌های کلیدی: الگوریتم ژنتیک، تنوع، انتخاب، جایگزینی.

Novel Diversity-Preservative Strategies for Genetic Algorithms and Its Application for Large-Scale Optimization

H. Ismkhan¹, MSc

1- Faculty of Engineering, University of Bonab, Bonab, Iran, Email: h.ismkhan@bonabu.ac.ir

Abstract: In order to increase performance of genetic algorithms, many approaches with aim of preserving diversity have been published. However, most of these approaches can be only applied to continuous optimization problems. This does not mean that genetic algorithms do not need population diversity, when they are applied to combinatorial optimization problems. In fact, defining the concept of similarity between solutions of combinatorial optimization problems, due to their apparent differences, is not straightforward. For example, for travelling salesman problem, how to measure similarity between solutions? This paper presents diversity preservative strategies which are based on similarity between solutions. These strategies not only can be applied to continuous optimization problems, but also by proposing novel semantic-oriented approaches to compute similarity between solutions of combinatorial optimization problems, it is possible to apply to combinatorial optimization problems, successfully.

Keywords: Genetic algorithm, diversity, selection, replacement.

تاریخ ارسال مقاله: ۱۳۹۵/۱۱/۰۱

تاریخ اصلاح مقاله: ۱۳۹۶/۰۸/۰۹

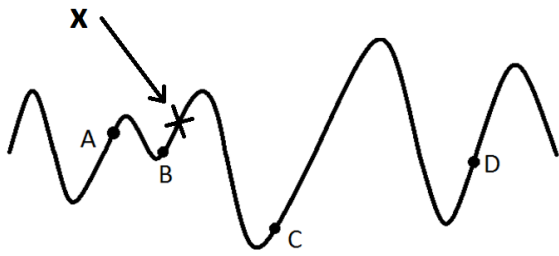
تاریخ پذیرش مقاله: ۱۳۹۶/۱۱/۱۷

نام نویسنده مسئول: حسن اسمخان

نشانی نویسنده مسئول: ایران - دانشگاه بناب - طبقه دوم دانشکده فنی و مهندسی

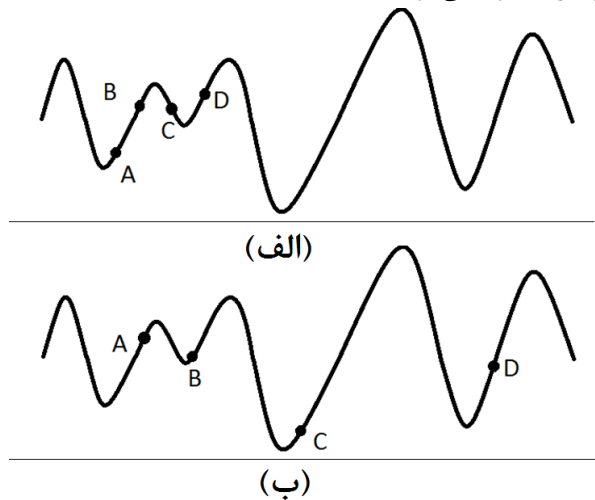
۱- مقدمه

الگوریتم‌های ژنتیک با کاربردهای فراوان در دنیای مهندسی [۳-۱]، با یک مشکل عمده به نام همگرایی زودرس^۱ مواجه هستند. از آنجایی که سطح پایین تنوع در جمعیت الگوریتم ژنتیک مهم‌ترین عامل مشکل همگرایی زودرس است، محققان روش‌های متنوعی را جهت حفظ تنوع ارائه داده‌اند [۴-۸]. تا جایی که حتی در برخی کارهای تحقیقی حفظ تنوع خود به عنوان تابع هدف لحاظ شده است [۹-۱۱]. برخورداری از سطح تنوع کافی موجب توزیع مناسب راه‌حل‌ها در فضای جستجو شده، قدرت اکتشافی^۲ را بیشتر کرده، احتمال همگرایی زودرس روی بهینه محلی را کاهش داده و در نتیجه احتمال رسیدن به بهینه فراگیر^۳ را افزایش می‌دهد. برای نمونه دو نمودار (الف) و (ب) را از شکل ۱ در نظر بگیرید. شاید در نگاه اول جمعیت جواب (الف) بهتر از (ب) به نظر آید چراکه مقدار برازش^۴ جواب‌های (الف) در حالت کلی بهتر از دسته جواب‌های (ب) هستند و جواب D از جمعیت (الف) از تمامی جواب‌های موجود در جمعیت (ب) بهتر^۵ است. اما توجه داشته باشید که سطح تنوع جواب‌های جمعیت (ب) بهتر از (الف) است. جواب C در جمعیت (ب) با اینکه به ظاهر بدتر از دیگر جواب‌هاست، اما در واقع شانس اصلی الگوریتم در رسیدن به بهینه فراگیر محسوب می‌شود.



شکل ۲: تزریق جواب جدید به جمعیت و نیاز به استراتژی جایگزینی

اهمیت روش‌های انتخاب^۶، اینکه برای اعمال عملگرهای ترکیب یا جهش کدام جواب‌ها از جمعیت انتخاب می‌شود، شاید امری واضح باشد، طوری که با وجود روش‌های سنتی [۱۵-۱۲]، حتی ساختارهای جدیدی نیز صرفاً باهدف اعمال استراتژی انتخاب کارآمد [۲۰-۱۶] برای جمعیت جواب‌ها طراحی شده است. اهمیت وجود توازن بین قدرت اکتشافی و قدرت استخراجی یا بهره‌کشی^{۱۰} الگوریتم‌های تکاملی^{۱۱} بر اهمیت استراتژی انتخاب می‌افزاید چراکه اعمال عملگرهای الگوریتم ژنتیک روی جواب‌های والد^{۱۲} مشابه، فرزندان مشابه آن‌ها را تولید کرده و باعث افزایش قدرت بهره‌کشی می‌شود. در مقابل انتخاب والد‌های متفاوت، باعث تولید جواب‌های فرزند متفاوت با جمعیت فعلی شده و در نتیجه قدرت اکتشافی الگوریتم را افزایش می‌دهد. حال توجه به این نکته ضروری است که افزایش بیش از حد قدرت اکتشافی، همگرایی الگوریتم را با تأخیر مواجه کرده و نیز افزایش بیش از حد قدرت بهره‌کشی هم‌گرایی زودرس را موجب می‌شود لذا الگوریتم ژنتیک به ایجاد توازن بین قدرت اکتشافی و بهره‌کشی نیاز مبرم دارد [۲۴-۲۱].



شکل ۱: دو نمونه متفاوت از نحوه توزیع جواب‌ها در فضای جستجو

این مقاله، جهت حفظ سطح تنوع در جمعیت الگوریتم ژنتیک، روش جدیدی را برای استراتژی‌های جایگزینی و انتخاب ارائه می‌دهد. روش ارائه شده بر پایه تشابه بین دو جواب استوار بوده و نه تنها هیچ‌گونه پیچیدگی ساختاری را بر الگوریتم تحمیل نمی‌کند، جمعیت را بانظم خاصی چنان می‌چیند که جواب‌های مشابه کنار یکدیگر و جواب‌های متفاوت در اندیس‌های دورتری نسبت به هم در آرایه مربوط به جمعیت قرار می‌گیرند.

البته استراتژی‌های ارائه شده این مقاله جهت اعمال روی مسائل گسسته با چالش نحوه محاسبه تشابه برای جواب‌های این دسته از مسائل مواجه خواهند شد. برای توضیح بیشتر بررسی مثال‌هایی برای دسته مسائل بهینه‌سازی گسسته می‌تواند این چالش را آشکارتر کند.

مثال ۱. مسئله فروشنده دوره‌گرد^{۱۳}: در این مسئله هدف پیدا کردن تور همیلتون کمینه در یک گراف وزن‌دار می‌باشد که جواب‌های جمعیت معمولاً به صورت دنباله‌ای از اعداد کد می‌شود، طوری که اگر N تعداد نودهای موجود در گراف باشد، هر عدد کد مابین ۱ تا N خواهد بود، البته بسته به اندیس شروع که می‌تواند از ۰ شروع شود هر عدد می‌تواند بین ۰ و N-۱ نیز باشد. برای مثال برای گراف وزن‌دار با پنج نود، دنباله ۳-۴-۲-۵-۱ بیان‌گر یک تور از نود ۱ به ۵، ۵ به ۲، ۲ به ۴، ۴ به ۳ و ۳ به ۱ است.

حال مطابق نمودار شکل ۲، اگر یک جواب جدید مانند X در نسل^۶ فعلی توسط یکی از عملگرهای ترکیب^۷ یا جهش^۸ تولید شود، اینکه جایگزین کدام یک از جواب‌های موجود در جمعیت شود قابل بحث خواهد بود. نمودار شکل ۲ غیرکارآمد بودن استراتژی جایگزین کردن بجای بدترین را بهتر نمایان می‌سازد، چراکه در صورت جایگزینی X به جای C، نه تنها تنوع جمعیت کاهش می‌یابد، بلکه الگوریتم ژنتیک نزدیک‌ترین موقعیت را جهت رسیدن به بهینه فراگیر از دست خواهد داد. البته طراحی استراتژی جایگزینی کارآمد باوجود عدم اطلاع از نقطه یا نقاط بهینه فراگیر و چگونگی توزیع جواب‌ها در فضای جستجو مشکل را دوچندان می‌نماید.

۲- مرور کارهای قبل مثال ۲. مسئله دور همیلتون با تعداد برچسب (رنگ) های

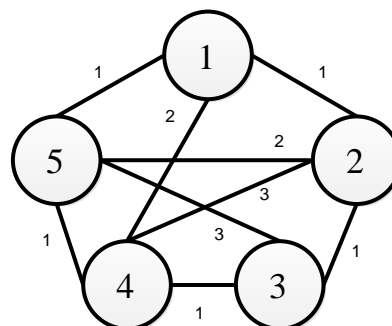
کارهای تحقیقی کمی در خصوص حفظ تنوع جمعیت الگوریتم‌های تکاملی جهت حل مسائل بهینه‌سازی گسسته صورت گرفته است. البته این بدان معنا نیست که این نوع الگوریتم‌ها در حل این نوع مسائل دچار مشکل همگرایی زودرس نبوده و به حفظ تنوع نیاز ندارند، بلکه عمده دلیل آن عدم کارایی روش‌های ارائه شده برای بهینه‌سازی پیوسته، در اعمال روی مسائل گسسته است. علاوه بر این، همان‌طور که در معرفی دو نمونه مسئله گسسته اشاره شد، ظاهر برخی از مسائل گسسته پیچیده بوده و رویکردهایی مانند روش بیتی جهت محاسبه تشابه بین جواب‌های مسائل گسسته مناسب نیستند. همچنین با توجه به ظاهر متفاوت انواع مختلف مسائل گسسته استفاده از یک یا چند روش محدود در محاسبه تشابه برای تمامی مسائل گسسته به هیچ عنوان مناسب نخواهند بود. در مقابل جهت حفظ تنوع در حل مسائل پیوسته، کارهای تحقیقی متعددی انجام شده است، اما به عقیده نویسندگان، تقریباً اکثر آن‌ها یا حداقل مواردی که در این مقاله تحقیقی بررسی شده‌اند را می‌توان به چهار دسته کلی طبقه‌بندی کرد. دسته اول روش‌هایی هستند که در آن یک ساختار جدید و در اکثر مواقع نسبتاً پیچیده برای نگه‌داری جواب‌های جمعیت پیشنهاد شده است [۲۷-۳۴]. دسته دوم شامل روش‌های خود-تطبیق^{۱۷} پارامترهای الگوریتم تکاملی هستند. دسته دیگر را می‌توان به روش‌های شامل ارائه عملگر جدید اختصاص داد [۱۴، ۳۵، ۳۶]. دسته چهارم به روش‌هایی اختصاص دارد که در آن استراتژی‌های جایگزینی برای حفظ تنوع پیشنهاد می‌شود [۳۷-۳۹]. البته برخی روش‌های دیگر نیز مانند کنترل اندازه جمعیت [۳۶، ۴۰] یا باز-مقداردهی [۳۳، ۴۱] جمعیت نیز به منظور حفظ تنوع معمول است. در ادامه این بخش به توضیح اجمالی برخی از این کارهای تحقیقی می‌پردازد.

۲-۱- تعریف ساختار برای جمعیت

الگوریتم ژنتیک سلولی^{۱۸} از جمله روش‌هایی است که با شکل دادن به ساختار جمعیت، تنوع را با هزینه زمانی پایین کنترل می‌کند [۱۷، ۱۸]. دلیل این هزینه پایین نیز در این است که در این ساختار جواب‌های متشابه در مکان‌های نزدیک به هم قرار می‌گیرند. طراحی ساختارهای جدید و استراتژی انتخاب برای این روش در کارهای تحقیقی معمول است [۳۴، ۱۷]. علاوه بر ساختار پیچیده جمعیت، تعداد زیاد پارامترها از جمله مشکلات الگوریتم ژنتیک سلولی هست [۱۸].

علاوه بر ساختار سلولی، روش ساختار فضایی پراکنده^{۱۹} [۳۰] که در آن جمعیت جواب‌ها در فضای اقلیدسی دوبعدی نگه‌داری می‌شود و همچنین الگوریتم ژنتیک چند جمعیتی دو هدفه^{۲۰} [۴۲] که در آن جمعیت به چندین خوشه تقسیم می‌شود، از جمله دیگر ساختارهایی است که جهت کنترل تنوع طراحی شده‌اند.

کمینه^۴: در این مسئله که بانام‌های مسئله فروشنده دوره‌گرد برچسب شده (LTSP^{۱۵}) یا مسئله فروشنده دوره‌گرد رنگارنگ (CTSP^{۱۶}) نیز معروف است، هدف پیدا کردن تور با تعداد مینیمم برچسب در یک گراف با یال‌های برچسب دار می‌باشد. برای توضیح بیشتر گراف شکل ۳ را با مجموعه برچسب‌های {۱، ۲، ۳} در نظر بگیرید. این گراف پنج نود دارد و برای هر یال آن یک برچسب نصب شده است که عضوی از مجموعه برچسب‌هاست. برای نمونه تور ۱-۵-۴-۳-۲-۱ فقط از ۱ برچسب استفاده می‌کند بنابراین هزینه آن ۱ است و به‌عنوان نمونه دیگر هزینه تور ۵-۱-۴-۳-۲-۵، با توجه به اینکه ۳ برچسب دارد، ۳ محاسبه می‌شود.



شکل ۳: یک گراف برچسب دار

با بررسی این دو مثال کاملاً آشکار است که به‌همان اندازه که به‌کارگیری روش بیتی [۲۵] برای محاسبه شباهت بین جواب‌های مسائل بهینه‌سازی گسسته غیرمنطقی است، به‌کارگیری یک یا چند روش محدود [۲۶] برای تمامی مسائل گسسته نیز غیرمنطقی خواهد بود چراکه اکثر مسائل گسسته تفاوت‌های حداقل ظاهری آشکاری با یکدیگر دارند. البته از آنجایی که یکی از اهداف این مقاله حل مسائل بهینه‌سازی گسسته می‌باشد، تعریف مفهوم تشابه برای مسائل گسسته یکی از ایده‌های اصلی این مقاله خواهد بود، طوری که با ارائه رویکردی معناگرا برای تعریف تشابه برای هر کدام از مسائل گسسته بیان شده، استفاده از استراتژی‌های ارائه شده به‌صورت معقول‌تری امکان‌پذیر خواهد بود.

با این توضیحات، بخش بعد به‌مرور کارهای قبلی حافظ تنوع می‌پردازد که البته فقط برای مسائل بهینه‌سازی پیوسته ارائه شده‌اند. بخش سوم استراتژی‌های پیشنهادی را ارائه می‌دهد. بخش چهارم به نحوه تعریف تشابه برای مسائل بهینه‌سازی پیوسته و هر کدام از مسائل بهینه‌سازی گسسته بررسی شده می‌پردازد. بخش پنجم کارایی استراتژی‌های پیشنهادی را در حل مسائل بهینه‌سازی پیوسته با ایجاد رقابت و مقایسه با روش‌های به‌روز اخیر می‌سنجد. بخش ششم به بررسی کارایی اعمال استراتژی جایگزینی به الگوریتم ژنتیک در حل مسائل گسسته با استفاده از مفاهیم تعریف معناگرای تشابه می‌پردازد. بالاخره بخش نهمی یک خلاصه از کل مقاله را ارائه می‌دهد.

۳- استراتژی‌های پیشنهادی حافظ تنوع

این مقاله جهت حفظ تنوع جمعیت، الگوریتمی برای نحوه انتخاب و الگوریتمی برای نحوه جایگزینی ارائه می‌دهد که هر دو بر پایه تشابه و تفاوت بین دو جواب استوار هستند. با وجود آنکه اکثر الگوریتم‌های حافظ تنوع موجود و تقریباً تمامی مراجع حافظ تنوع این مقاله فقط قابلیت اعمال روی مسائل پیوسته را دارند، الگوریتم جایگزینی پیشنهادی به طرز موفقیت‌آمیزی روی نمونه‌های متفاوت از دو مسئله بهینه‌سازی گسسته نیز اعمال می‌شود. علاوه بر این، با استفاده از این الگوریتم‌های پیشنهادی، بدون نیاز به یک ساختار پیچیده، می‌توان به جواب‌های جمعیت نظم داد، طوری که جواب‌های مشابه، در اندیس‌های نزدیک به هم و جواب‌های متفاوت در اندیس‌های دورتری نسبت به هم قرار گیرند، با این معنا که با دور شدن از اندیس یک جواب، تشابه کمتر می‌شود. به عنوان مثال در یک جمعیت با ۲۰۰ جواب، تشابه جواب i ام با جواب $i+1$ ام بیشتر از تشابه جواب i ام با $i+100$ ام خواهد بود. علاوه بر این، روش‌های پیشنهادی، جهت حفظ تنوع نیازی به محاسبه سطح تنوع در مراحل اجرای الگوریتم ژنتیک ندارند. با توجه به اینکه هدف کلی الگوریتم‌های ارائه شده حفظ تنوع جمعیت در یک سطح مناسب است، روش پیشنهادی Diversity Preservative GA یا به اختصار DivP-GA نامیده می‌شود.

در ادامه، این بخش به بیان الگوریتم جایگزینی و الگوریتم انتخاب می‌پردازد. جهت بیان نحوه محاسبه تشابه بین دو جواب، بخش بعدی مقاله به صورت مجزا در نظر گرفته شده است که در آن ضمن بیان نحوه محاسبه تشابه برای مسائل پیوسته، ضمن معرفی دو نوع مسئله بهینه‌سازی گسسته، نحوه محاسبه تشابه برای جواب‌های این نوع مسائل گسسته نیز، با یک رویکرد معناگرا، ارائه خواهد شد.

۳-۱- الگوریتم جایگزینی پیشنهادی

الگوریتم پیشنهادی جایگزینی سعی می‌کند، جواب جدید را با شبیه‌ترین جواب‌های موجود در جمعیت جهت جایگزینی مورد بررسی قرار دهد، اما از آنجایی که این پروسه به خودی خود بسیار کند است دو روش اکتشافی^{۲۴} نیز پیشنهاد می‌شود.

در روش اکتشافی اول می‌توان چنین استدلال کرد که هر جواب جدید تولید شده به والدین شبیه است چراکه عملگر ترکیب، جواب فرزند را بر اساس دو جواب والد می‌سازد. در خصوص عملگر جهش نیز به طور مشابه، جواب جدید بر اساس تک والد (جواب) ورودی عملگر جهش ساخته شده و به این دلیل منطقی است که شبیه آن باشد. البته الگوریتم جایگزینی پیشنهادی علاوه بر والد یا والدین، تعدادی از جواب‌های نزدیک اندیس والد را نیز بررسی می‌کند. برای عملگر ترکیب که به دو والد نیاز دارد، جهت تسریع هر چه بیشتر، شباهت بین جواب جدید با هر کدام از دو والد که بیشتر باشد، فقط جواب‌های نزدیک آن والد بررسی می‌شود.

از جمله دیگر روش‌های طراحی ساختار، ارائه الگوریتم ژنتیک با جمعیت دوتایی (DP-GA)^{۲۱} هست [۲۹، ۳۱]. در این روش علاوه بر جمعیت اصلی، یک جمعیت کمکی نیز وجود دارد که در آن، برای هر نسل، نحوه انتخاب والد برای عملگر ترکیب با دو رویکرد انتخاب می‌شود: در اولین رویکرد انتخاب هر دو والد از یک جمعیت بوده که حاصل آن فرزندان هم‌خون^{۲۲} است و در رویکرد دوم انتخاب هر کدام از والدین از جمعیت متفاوت بوده که حاصل آن فرزندان دورگه^{۲۳} است. البته این روش از ابتکار جالب دیگری نیز بهره می‌برد که در آن، مقدار برازش هر جواب موجود در جمعیت کمکی تابع متوسط فاصله آن از جواب‌های موجود در جمعیت اصلی هست که البته موجب بیشتر شدن هزینه زمانی می‌شود. یک از اهداف مقاله [۲۵]، در واقع تلاشی بر فائق آمدن بر همین مشکل است.

۲-۲- ارائه روش‌های خود-تطبیق

این نوع روش‌ها معمولاً جهت کنترل نرخ استفاده از عملگر جهش استفاده می‌شود [۱۴، ۳۲]. عمده دلیل آن نیز نقش مهم این عملگر در تغییر جواب‌های موجود در جمعیت، افزایش تنوع و در نتیجه افزایش قدرت اکتشافی الگوریتم ژنتیک است [۴۳]. در برخی موارد نرخ استفاده از عملگر جهش تابعی از سطح تنوع جمعیت است [۳۵، ۴۴]. اما در خصوص این رویکردها نحوه محاسبه تنوع بحث برانگیز خواهد بود چراکه محاسبه سطح تنوع، با رویکرد محاسبه میزان تفاوت بین جواب‌های جمعیت، حتی به صورت تقریبی زمان بر است.

۲-۳- ارائه عملگر جدید

تعریف عملگرهای جدید برای محافظت از سطح تنوع نیز در برخی کارهای تحقیقی یافت می‌شود. این عملگرها فقط به تعریف عملگر جهش [۳۶] محدود نیستند، بلکه برخی از عملگرهای ترکیب مانند عملگر شیفت سلولی^{۲۴} [۱۵]، حداقل به باور نویسندگان آن می‌تواند تنوع جمعیت را محافظت کند. در الگوریتم ژنتیک ارائه شده در مقاله [۴۵]، تنوع با عملگری به نام ترکیب بسیار مخرب^{۲۵} تضمین می‌شود.

۲-۴- استراتژی‌های جایگزینی

از جمله استراتژی‌های جایگزینی سنتی حافظ تنوع، روش ازدحامی است [۳۸، ۳۹] که البته امروزه نیز از آن استفاده می‌شود [۴۶، ۴۷]. در روش ازدحامی، جواب جدید با شبیه‌ترین جواب به خود از میان تعدادی جواب تصادفی جمعیت رقابت می‌کند. البته کنترل تنوع هدف تمامی استراتژی‌های جایگزینی نیست [۴۸]. در مقابل، هدف اصلی استراتژی جایگزینی ارائه شده در [۳۷] بانام CD/RW، صرفاً حفظ تنوع جمعیت است، ولی مشابه اکثر استراتژی‌های حافظ تنوع از نظر سرعت بسیار کند عمل می‌کند.

index-1، به صورت تصادفی انتخاب می‌شود اما در انتخاب جواب دوم سعی می‌شود تا متفاوت‌ترین جواب نسبت به جواب اول انتخاب شود. برای این منظور، مطابق بحث نهایی بخش قبل در خصوص نحوه جای‌گیری جواب‌ها مطابق شکل ۵، جواب‌های موجود حول اندیس بافاصله $|P|/2$ از اندیس جواب اول می‌تواند تقریب مناسبی برای پیدا کردن متفاوت‌ترین جواب باشد. شکل ۶ الگوریتم انتخاب پیشنهادی را نشان می‌دهد. در این الگوریتم R دقیقاً همان پارامتر R الگوریتم جایگزینی است.

گام اول: مجموعه S را با مقدار اولیه تهی در نظر بگیر.
 گام دوم: اندیس جواب اول، index-1، را به صورت تصادفی از اعداد مابین 0 تا $|P|-1$ انتخاب کن.
 گام سوم: به تعداد R جواب حول اندیس: $(index-1 + |P|/2) \bmod |P|$
 را به مجموعه S اضافه کن.
 گام چهارم: اندیس متناظر با متفاوت‌ترین جواب موجود در S را نسبت $P[index-1]$ را به‌عنوان اندیس جواب دوم در نظر بگیر.

شکل ۶: الگوریتم انتخاب پیشنهادی

اکتشافی دوم نیز پیشنهاد می‌دهد که به هنگام محاسبه تشابه، جهت تسریع بیشتر، فقط تعداد معین و محدودی از متغیرها به صورت تصادفی بررسی شوند.

با این اوصاف می‌توان الگوریتم جایگزینی را به فرم شکل ۴ بیان کرد که ورودی آن جمعیت P با تعداد $|P|$ جواب، جواب جدید X و دو اندیس index-1 و index-2، که محل جواب‌های والد هستند. طبیعی است که اگر جواب جدید با عملگر جهش تولید شده باشد index-1 و index-2 مساوی خواهند بود. در این الگوریتم R یک پارامتر می‌باشد.

گام اول: مجموعه S را با مقدار اولیه تهی در نظر بگیر.
 گام دوم: اگر شباهت $P[index-1]$ با X بیشتر از شباهت $P[index-2]$ با X است در این صورت S را با R جواب حول index-1 از P مقداردهی کن و در غیر این صورت، S را با R جواب حول index-2 از P مقداردهی کن.
 گام سوم: شبیه‌ترین جواب به X را از S انتخاب کرده و اندیس متناظر آن در P را بانام most-similar معین کن.
 گام چهارم: اگر مقدار برازش X از مقدار برازش $P[most-similar]$ بیشتر بود، X را در P جایگزین آن کن.

شکل ۴: الگوریتم جایگزینی پیشنهادی

این الگوریتم نیز برای تسریع بیشتر از اکتشافی دوم بیان شده برای الگوریتم جایگزینی بهره می‌برد.

۴- رویکردهای معناگرای جدید محاسبه تشابه بین دو جواب مسائل بهینه‌سازی پیوسته و گسسته

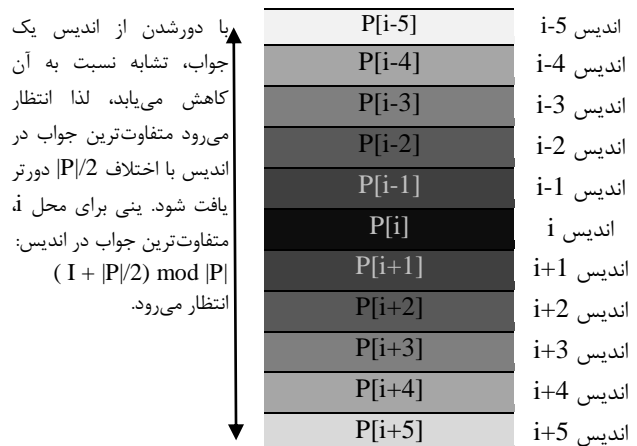
۴-۱- نحوه محاسبه تشابه برای مسائل بهینه‌سازی پیوسته

با وجود آن‌که محاسبه تشابه بین دو جواب مسئله پیوسته می‌تواند به راحتی با ملاک قراردادن یکی از انواع فاصله‌ها مشخص شود اما توجه به نوع عملگرهای مورد استفاده در الگوریتم ژنتیک یا دیگر الگوریتم‌های تکاملی مبتنی بر جمعیت نیز مهم است. اگر عملگرهای معمول مانند ترکیب یا جهش بر اساس مقادیر معمول جواب‌ها عمل می‌کنند، در این صورت استفاده از انواع فاصله مینکوسکی^{۲۷} مانند فاصله منهتن^{۲۸} یا فاصله اقلیدسی^{۲۹} مناسب خواهد بود. اما اگر عملگرها به صورت بیتی اعمال می‌شوند در این صورت استفاده از انواع فاصله‌های مبتنی بر بیت مانند فاصله همینگ^{۳۰} مناسب خواهد بود. توجه داشته باشید که با مشخص کردن معیار تفاوت به وسیله فاصله بین دو جواب، معیار تشابه نیز مشخص خواهد بود. برای نمونه اگر فاصله جواب A با X بیشتر از فاصله جواب A با Y باشد، در این صورت تشابه A با Y بیشتر از تشابه A با X است.

۴-۲- معیار شباهت برای مسئله فروشنده دوره‌گرد (بهینه‌سازی گسسته)

برای محاسبه شباهت بین دو جواب مسئله فروشنده دوره‌گرد، بجای استفاده از روش بیتی [۲۵] یا روش [۲۶]، روش معناداری

درواقع هدف الگوریتم جایگزینی پیشنهادی این است تا جمعیت جواب‌ها را طوری نظم دهد که مانند شکل ۵ جواب‌های مشابه نزدیک هم بوده و با دور شدن از اندیس یک جواب از مقدار تشابه نسبت به آن جواب کاسته شود. البته نتایج آزمایش‌ها نیز وجود چنین نظمی را برای جمعیت الگوریتم ژنتیک استفاده‌کننده از این استراتژی جایگزینی، به شرط استفاده از الگوریتم انتخابی پیشنهادی، تأیید می‌کند.



شکل ۵: نظم مورد انتظار برای جمعیت که با استفاده از الگوریتم‌های جایگزینی و انتخاب قابل حصول است.

۴-۲- الگوریتم انتخاب پیشنهادی

برای عملگر جهش که یک روی یک جواب اعمال می‌شود، انتخاب به صورت اندیس جواب مورد نظر از جمعیت به صورت تصادفی است. برای عملگر ترکیب که به دو جواب نیاز دارد، اندیس جواب اول،

جدول ۱: توابع بهینه‌سازی استفاده شده در آزمایش‌ها

$F_1(\vec{x}) = \sum_{i=1}^D x_i^2$
$F_2(\vec{x}) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i$
$F_3(\vec{x}) = \sum_{i=1}^D (\lfloor x_i + 0.5 \rfloor)^2$
$F_4(\vec{x}) = \sum_{i=1}^D i \cdot x_i^4$
$F_5(\vec{x}) = \sum_{i=1}^D -x_i \cdot \sin(\sqrt{ x_i })$
$F_6(\vec{x}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$
$F_7(\vec{x}) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)) + 20 + e$
$F_8(\vec{x}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$
$F_9(\vec{x}) = \frac{\pi}{D} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 \cdot [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$
$F_{10}(\vec{x}) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 \cdot [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \} + \sum_{i=1}^D u(x_i, 5, 100, 4)$
$F_{11}(\vec{x}) = \sum_{i=1}^D \frac{ \sin(10x_i\pi) }{10x_i\pi}$
$F_{12}(\vec{x}) = \sum_{i=1}^D (x_i \sin(10x_i\pi))$
برای F9 و F10، u و y1 به صورت زیر محاسبه می‌شوند: $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$
$y_i = 1 + \frac{1}{4}(x_i + 1)$

پیشنهاد می‌شود. توجه به این مورد ضروری است که برای دو تور دقیقاً یکسان ۱۲۳۴۵ و ۲۳۴۵۱، روش ارائه شده در [۲۶]، به‌اشتباه، بالاترین میزان تفاوت و کمترین میزان شباهت را قائل است.

برای حل این مشکل روش پیشنهادی در عین سادگی، معنادار می‌باشد، طوری که برای محاسبه میزان تشابه تعداد یال‌های مشترک را به‌عنوان معیاری برای سنجش میزان تشابه در نظر می‌گیرد. بنابراین برای دو تور ۱۲۳۴۵ و ۲۳۴۵۱، میزان تشابه ۵ خواهد بود، چراکه یال‌های ۱ به ۲، ۲ به ۳، ۳ به ۴، ۴ به ۵ و ۵ به ۱ در هر دو تور موجود می‌باشد.

۳-۴- معیار شباهت برای مسئله فروشنده دوره‌گرد رنگارنگ (بهینه‌سازی گسسته)

باوجود تشابه ظاهری بین دو مسئله فروشنده دوره‌گرد و فروشنده دوره‌گرد رنگارنگ، استفاده از معیار شباهت اولی برای مسئله دومی غیرمنطقی خواهد بود، چرا که برای مسئله دوم دو مجموعه از چندین یال متفاوت، در صورت استفاده از مجموعه برجسب یا رنگ‌های مشترک، می‌تواند هزینه یکسانی داشته باشد. لذا روش پیشنهادی بدین ترتیب است که تعداد برجسب (رنگ) های مشترک مورد استفاده در دو تور میزان تشابه آن‌ها خواهد بود. جهت توضیح بیشتر، مثالی با دو تور ۱۲۳۴۵ و ۲۳۴۱۵ روی گراف برجسب‌دار شکل ۳، بررسی می‌شود. تور ۱۲۳۴۵ شامل مجموعه برجسب {۱} و تور ۲۳۴۱۵ شامل مجموعه برجسب {۲، ۱} بوده و اشتراک این دو مجموعه، {۱} می‌باشد که اندازه آن ۱ است، بنابراین تشابه ۱ خواهد بود.

۵- بیان نتایج آزمایش‌های بهینه‌سازی پیوسته

این قسمت نتایج چندین آزمایش را جهت نمایش کارایی DivP-GA در حل مسائل بهینه‌سازی پیوسته گزارش می‌دهد. تمامی این آزمایش‌ها روی کامپیوتر با پردازنده Intel® Core i3-4150 3.50 GHz تحت سیستم عامل Windows 7 انجام شده‌است. در این آزمایش‌ها از دوازده تابع F1 الی F12 لیست شده در جدول ۱ استفاده شده‌است. این توابع، به غیر از F12، از نوع کمینه‌سازی هستند. تابع F12 یک تابع جهت بهینه‌سازی هست. این توابع قابلیت‌گسترش به تعداد دلخواه متغیر را دارند که در آزمایش‌ها از این توابع با تعداد صد متغیر استفاده می‌شود. در تمامی آزمایش‌ها برای DivP-GA از الگوریتم کنترل نسبت جهش ارائه شده در [۴۳] استفاده می‌شود.

در ادامه، قسمت اول این بخش، جهت نمایش کارایی DivP-GA، آن را با الگوریتم‌های به‌روز ارائه شده، در حل مسائل بهینه‌سازی مقایسه می‌کند. قسمت دوم، به ارائه نتایج جهت نمایش وجود نظم مورد انتظار شبیه شکل ۵ می‌پردازد.

۵-۱- مقایسه با الگوریتم‌های به‌روز

جهت نمایش کارایی روش پیشنهادی، DivP-GA با الگوریتم‌های CD/RW [37]، DP-GA [25] و ADM-RCGA [49] مقایسه می‌شود. دلیل انتخاب دو الگوریتم اول به‌عنوان رقیب DivP-GA، هدف مشابه آن‌ها در خصوص حفظ تنوع جمعیت است. دلیل شرکت ADM-RCGA در رقابت این است که نتایج آزمایش‌های انجام شده در [۴۹] نشان می‌دهد این الگوریتم دقت بالایی در حل توابع بهینه‌سازی دارد. برای DivP-GA، CD/RW و DP-GA، اندازه جمعیت، ۸۰۰ جواب تنظیم شد. الگوریتم ADM-RCGA با تعداد ۸۰۰ جواب برای جمعیت، دقت کمی در مقایسه با اندازه ۳۰۰ جواب، همان تعداد پیش‌فرض استفاده شده توسط مبدعان آن، داشت، لذا برای آن اندازه جمعیت ۳۰۰ تنظیم شد. دیگر پارامترهای هر کدام از الگوریتم‌ها مانند نسخه

جدول ۲: رقابت DivP-GA با سایر الگوریتم‌ها

	CD/RW		DP-GA		ADM-RCGA		DivP-GA		نتیجه آزمون آماری در مقابل: DivP-GA		
	متوسط	انحراف معیار	متوسط	انحراف معیار	متوسط	انحراف معیار	متوسط	انحراف معیار	CD/RW	DP-GA	ADM-RCGA
F ₁	4.19E-17	1.51E-16	1.33E-01	7.30E-01	1.12E-10	4.85E-10	3.33E-154	1.79E-153	~	~	~
F ₂	8.28E-25	2.92E-24	1.07E-47	5.85E-47	3.24E-29	1.32E-28	2.69E-22	1.47E-21	~	~	~
F ₃	0.00E+00	0.00E+00	1.33E-01	7.30E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	~	~	~
F ₄	3.81E-45	2.00E-44	2.30E-208	0.00E+00	1.72E-23	8.29E-23	3.27E-37	1.78E-36	~	~	~
F ₅	-3.43E+04	1.53E+03	-3.08E+04	1.07E+03	-4.03E+04	4.11E+02	-4.18E+04	2.81E+01	+	+	+
F ₆	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.12E+01	6.07E+00	0.00E+00	0.00E+00	~	~	+
F ₇	1.01E-14	3.84E-14	0.00E+00	0.00E+00	2.46E-04	3.40E-04	0.00E+00	0.00E+00	~	~	+
F ₈	4.80E-03	8.51E-03	1.12E-01	1.87E-01	6.27E-09	7.91E-09	0.00E+00	0.00E+00	~	+	+
F ₉	2.05E-01	7.23E-02	1.27E+00	5.27E-02	3.97E-08	7.37E-08	5.07E-01	9.70E-02	+	+	-
F ₁₀	4.50E+00	5.89E-01	8.83E+00	3.72E-01	-1.15E+00	3.83E-05	6.27E+00	5.25E-01	-	+	-
F ₁₁	9.56E+00	4.44E-01	6.91E-01	1.17E-01	1.73E-01	1.78E-01	1.02E-01	8.88E-03	+	+	+
F ₁₂	1.17E+02	5.85E+00	1.14E+02	5.01E+00	1.50E+02	3.66E+00	1.79E+02	8.03E-01	+	+	+

جدول ۳: متوسط زمان رسیدن به بهترین جواب در ۳۰ اجرا (ثانیه)

	CD/RW	DP-GA	RCGA-ADM	DivP-GA
F ₁	۷۳/۹۵	۴۵/۶۴	۱۳/۵	۲۰/۹
F ₂	۷۱/۳۶	۴۷/۹۶	۱۶/۵۹	۱۵/۶۲
F ₃	۴۵/۷	۴۲/۶۶	۳۰/۵۸	۱۹/۶۵
F ₄	۸۸/۴۵	۹۵/۹۱	۵۸/۴۶	۶۲/۱۸
F ₅	۶۳/۴۳	۵۵/۱۹	۴۵/۷	۲۵/۳۵
F ₆	۶۵/۸۹	۲۶/۳۹	۳۳/۴۶	۱۰/۳
F ₇	۸۰/۴۴	۴۱/۹	۳۳/۳۶	۱۵/۷
F ₈	۹۰/۶۱	۶۱/۹۵	۵۴/۸۷	۲۸/۴
F ₉	۱۱۷/۶	۹۶/۵	۱۵۰/۶۶	۶۵/۸
F ₁₀	۱۵۷/۷۵	۱۵۶/۴۲	۱۴۸/۵۵	۱۰۱/۱۱
F ₁₁	۰/۲۱	۵۰/۹۶	۳۳/۸۲	۲۲/۱۴
F ₁₂	۵۴/۹۶	۴۷/۸۱	۳۱/۹۳	۱۹/۷۸

اصلی ارائه شده توسط مبدعان آن‌ها است. برای CD/RW مفهومی بنام سهم تنوع^{۳۱} برای یک جواب وجود دارد که جهت محاسبه آن در نسخه اصلی لازم است تا متوسط فاصله اقلیدسی یک جواب از دیگر جواب‌های جمعیت سنجیده شود. جهت تسریع این الگوریتم به جای این محاسبه، فاصله جواب از مرکز ثقل جواب‌های جمعیت ملاک محاسبه قرار داده شد که نه تنها باعث تسریع الگوریتم شد، بلکه دقت آن را نیز نسبت به نسخه اصلی افزایش داد. با توجه به معمول بودن عملگرهای بیتی برای الگوریتم ژنتیک [۲۵، ۳۶]، برای DivP-GA از عملگر یک نقطه^{۳۲} برای ترکیب و از عملگر تغییر تک‌بیتی برای جهش استفاده شد، لذا متعاقباً جهت محاسبه معیار تشابه، فاصله همینگ اعمال شد. جهت تنظیم احتمال اعمال عملگر ترکیب نیز الگوریتم MRCA [۴۳] به کار گرفته شد.

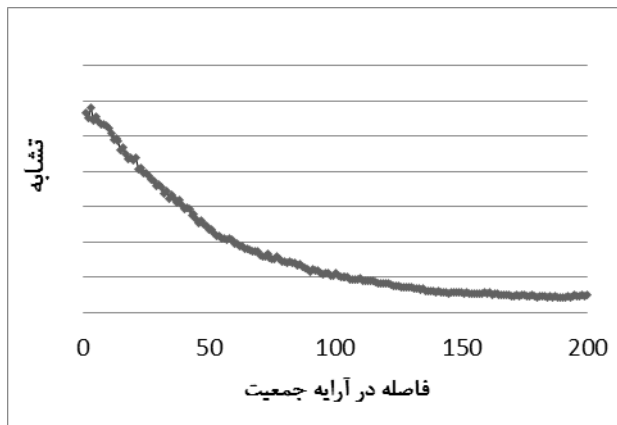
جدول ۲ نتایج این آزمایش را که در آن هرکدام از الگوریتم‌های یادشده به تعداد ۳۰ اجرا، هرکدام از توابع جدول ۱ را با تعداد ۱۰۰ متغیر حل کرده‌اند. متوسط و انحراف معیار هرکدام از چهار الگوریتم در این جدول نمایش داده شده‌اند. سه ستون آخر نتیجه آزمون Wilcoxon-Signed-Rank می‌باشد. علامت + نشان می‌دهد که DivP-GA به صورت معناداری از الگوریتم رقیب معین شده بهتر است. علامت - نشان‌دهنده برتری معنادار الگوریتم رقیب مقابل DivP-GA بوده و علامت ~ نشان‌دهنده عدم وجود تفاوت معنادار بین DivP-GA و رقیب آن است. این نتایج نشان می‌دهد، DivP-GA در پنج مورد بهتر و در یک مورد بدتر از CD/RW بوده و در بقیه موارد برتری آن قابل ملاحظه نیست. DivP-GA در شش مورد بهتر از DP-GA بوده و در بقیه موارد برتری آن قابل ملاحظه نیست. در مقابل ADM-RCGA، DivP-GA در شش مورد بهتر از آن، در یک مورد بدتر از آن بوده و در بقیه موارد تفاوت قابل ملاحظه‌ای بین آن‌ها نیست.

جدول ۳ متوسط زمان رسیدن به بهترین جواب هرکدام از الگوریتم‌ها را در حل مسائل $F_i (1 \leq i \leq 12)$ ، برحسب ثانیه، نشان می‌دهد. الگوریتم DivP-GA در نه مسئله سریع‌تر از دیگر الگوریتم‌ها عمل می‌کند. جهت بررسی بیشتر سرعت الگوریتم‌ها، آزمایش دیگری ترتیب داده شد که در آن، رقابت‌کننده‌ها مسئله F₈ را با اندازه‌های ۲۰، ۴۰، ۶۰، ۸۰ و ۱۰۰ متغیر به تعداد ۳۰ اجرا حل کردند.

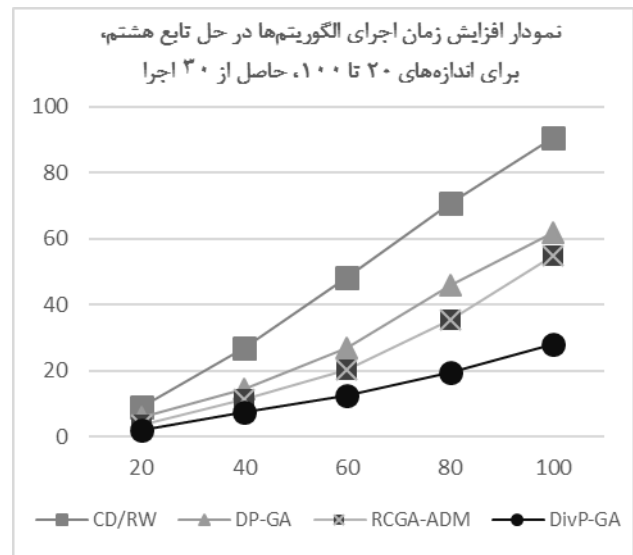
حاصل متوسط زمان اجرای این الگوریتم‌ها در شکل ۷ خلاصه شده است. با افزایش اندازه مسئله، نمودار DivP-GA به صورت واضحی زیر نمودار زمان اجرای دیگر الگوریتم‌ها قرار می‌گیرد، که نشان از سرعت بالا و پیچیدگی زمانی کم آن نسبت به بقیه الگوریتم‌های رقابت‌کننده است.

۵-۲- بررسی وجود نظم مورد انتظار ساختار جمعیت DivP-GA

برای نشان دادن اینکه DivP-GA ساختار جمعیت را مشابه شکل (۵) منظم می‌کند، آزمایشی ترتیب داده شد که در آن متوسط تشابه یک جواب با دیگر جواب‌ها در ۳۰ اجرا، به ترتیب اندیس حساب می‌شود. طوری که در خاتمه هر اجرا، تشابه جواب اندیس pos یعنی $Population[pos]$ با جواب دیگر اندیس‌ها، یعنی $Population[pos+i] \bmod |P|$ از ۰ تا ۱۹۹ افزایش می‌یابد) محاسبه شد. برای تمامی مسئله‌ها نموداری شبیه به نمودار شکل ۸ حاصل شد. همان‌طور که در این نمودار قابل مشاهده است شبیه‌ترین جواب‌ها نسبت به یک جواب را می‌توان در اندیس‌های نزدیک آن جواب پیدا کرد.



شکل ۸: بررسی تشابه یک جواب با دیگر جواب‌های جمعیت به ترتیب افزایش اندیس



شکل ۷: نمودارهای زمان اجرای الگوریتم‌ها

جدول ۴: رقابت DivP-GA با الگوریتم ژنتیک [50]

نام	متوسط هر الگوریتم در ۵۰ اجرا		درصد بهبود	نتیجه آزمون آماری
	الگوریتم ژنتیک [50]	DivP-GA		
eil51	۴۳۲/۸	۴۲۸/۶۸	۰/۸	+
eil76	۵۴۴/۸۲	۵۴۲/۹	۰/۴	+
kroA100	۲۱۷۷۸/۷۲	۲۱۷۳۲/۲	۰/۲	+
lin105	۱۴۶۲۷/۲۲	۱۴۵۴۷/۵۲	۰/۵	+
kroA200	۳۰۰۲۲/۵۶	۲۹۹۵۲/۸	۰/۲	~
a280	۲۶۲۸/۸۸	۲۶۲۲/۸	۰/۳	~
lin318	۴۳۷۵۸/۸۶	۴۳۶۲۹/۹	۰/۳	~
pcb442	۵۳۷۸۰/۴	۵۳۱۹۶/۸۴	۱/۱	+
att532	۲۹۶۷۹/۱۸	۲۹۰۸۳/۹	۲	+
rat575	۷۰۵۶/۶۲	۶۹۸۸/۱۶	۱	+
rat783	۹۲۶۲/۱۴	۹۱۵۰/۵۲	۱/۲	+
pr1002	۲۷۸۳۲۸/۱	۲۷۵۱۵۳/۲	۱/۲	+

۶-۲- بررسی همگرایی DivP-GA

شاید بعد از بررسی جدول ۴ این سؤال پیش آید که چرا اختلاف بین DivP-GA و الگوریتم ژنتیک [۵۰] قابل توجه نیست و اینکه آیا DivP-GA نیز مانند الگوریتم رقیب قابلیت کنترل سطح تنوع جمعیت را از دست داده و در دام مشکل همگرایی زودرس متوقف می‌شود. بررسی عملکرد دو الگوریتم با تعداد نسل‌های بزرگ می‌تواند جواب دقیقی به این سؤال ارائه دهد. لذا آزمایشی جهت بررسی این مورد ترتیب داده می‌شود که دو الگوریتم دقیقاً با شرایط آزمایش پیش، اما با تعداد نسل ۱۰۰۰،۰۰۰ در ۵۰ اجرا به رقابت می‌پردازند. شکل ۹ نتیجه این آزمایش را به نحوه گویایی نمایش می‌دهد. این شکل هزینه متوسط بهترین تور حاصل از هر الگوریتم را در هر نسل نمایش می‌دهد. البته با توجه به مقارنه تصادفی تورها در نسل اول، نمودار از نسل ۴۰۰۰۰۰ به بعد رسم شده است.

۶- نتایج آزمایش‌ها برای مسائل بهینه‌سازی گسسته

این قسمت، نتایج چندین آزمایش را جهت نمایش کارایی DivP-GA در حل مسائل بهینه‌سازی گسسته شامل مسئله فروشنده دوره‌گرد و فروشنده دوره‌گرد رنگارنگ گزارش می‌دهد. تمامی الگوریتم‌های مربوط به آزمایش‌های این بخش با زبان ++C، و در محیط Microsoft VC++ 2015 روی سیستم عامل Windows-7 ۶۴ بیتی کامپایل و فایل اجرایی تولید می‌شود. میزبان این آزمایش‌ها نیز کامپیوتر اداری با پردازنده Intel® Core™ i3-4150 CPU @ 3.50 GHz و حافظه RAM با اندازه 4 G است.

۶-۱- نتایج برای مسئله فروشنده دوره‌گرد

در این آزمایش، رقیب DivP-GA، الگوریتم ارائه شده در [۵۰] است که جهت بررسی صحیح کارایی دو نوع الگوریتم ژنتیک از الگوریتم جستجوی محلی استفاده نمی‌شود. برای هر دو الگوریتم در شرایطی یکسان، اندازه جمعیت ۱۰، تعداد نسل‌ها ۲۰۰۰۰ و احتمال اعمال عملگر ترکیب ۰/۹ تنظیم می‌شود. قابل ذکر است که در هر نسل، در صورتی که عملگر ترکیب اعمال نشود، عملگر جهش اعمال می‌شود، با این توضیح که عملگر حریصانه بهبودیافته (IGX) به‌عنوان عملگر ترکیب و عملگر پل-مضاعف [۵۰] به‌عنوان عملگر جهش اعمال می‌شود. نمونه مسئله‌ها نیز از کتابخانه استاندارد مسئله فروشنده دوره‌گرد (TSPLIB) [۵۱] با اندازه‌های مختلف تا ۱۰۰۰ نود هستند. جدول ۴ نتایج این آزمایش را که حاصل از ۵۰ اجرا هر کدام از الگوریتم‌ها در حل هر کدام از نمونه‌هاست، نشان می‌دهد. ستون آخر این جدول نتیجه آزمون Wilcoxon-Signed-Rank می‌باشد. علامت + نشان می‌دهد که DivP-GA به‌صورت معناداری از الگوریتم رقیب معین شده بهتر است. برای ۹ نمونه داده استاندارد DivP-GA به‌صورت معناداری بهتر از الگوریتم رقیب عمل می‌کند، در حالی که در ۳ نمونه داده تفاوت معناداری بین آن‌ها نیست.

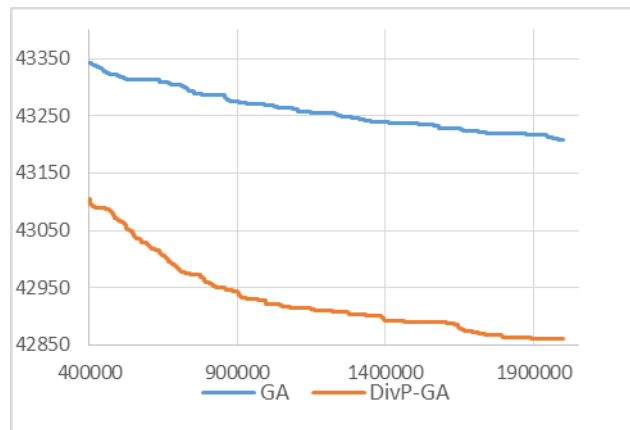
تفاوت که DivP-GA استراتژی جایگزینی ارائه شده با استفاده از رویکرد جدید محاسبه تشابه برای مسئله فروشنده دوره گرد رنگارنگ را استفاده می‌کند. در مجموع از ۴۸ مجموعه داده شامل ۱۶ مجموعه داده با تعداد نودهای ۲۸۰ (Inst-280-*)، ۱۶ مجموعه داده با تعداد نودهای ۵۳۲ (Inst-532-*) و ۱۶ مجموعه داده با تعداد نودهای ۱۰۰۰ (Inst-1000-*)، متفاوت قابل دسترس در <http://josilber.scripts.mit.edu> استفاده می‌شود. هر کدام از ۱۶ مجموعه داده هر دسته، در چگالی بال‌ها (p) و نوع آن‌ها، که می‌تواند LR، LC، RR باشد، تفاوت دارند که برای اطلاعات بیشتر می‌توان به [۵۳] مراجعه کرد. نتایج این آزمایش که حاصل ۵۰ اجرای الگوریتم در حل هر کدام از نمونه‌هاست در جدول ۶، جدول ۷ و جدول ۸ خلاصه می‌شود. ستون آخر این جداول نتیجه آزمون Wilcoxon-Signed-Rank می‌باشد. علامت + نشان‌دهنده برتری معنادار DivP-GA، علامت - به معنای برتری معنی‌دار الگوریتم رقیب و علامت ~ به معنای عدم وجود تفاوت معنادار بین دو الگوریتم می‌باشد.

جدول ۶: رقابت DivP-GA با H-2Ph-EA برای دسته مسائل با اندازه ۲۸۰ و تعداد رنگ ۷۸۴

نام	نوع	m	ρ	متوسط هر الگوریتم در ۵۰ اجرا		درصد بهبود	نتیجه آزمون آماری
				H-2Ph-EA	DivP-GA		
I-280-1	LC	۷۸۴	۰/۲	۸۱/۸۸	۷۹/۸۴	۲/۶	+
I-280-2	LR	۷۸۴	۰/۲	۴۱/۳	۳۹/۷۲	۴	+
I-280-3	RC	۷۸۴	۰/۲	۷۸/۶	۷۶/۶	۳/۳	+
I-280-4	RR	۷۸۴	۰/۲	۳۷/۵۲	۳۶/۹۴	۱/۶	~
I-280-5	LC	۷۸۴	۰/۵	۵۳/۷۴	۵۱/۹۴	۳/۵	+
I-280-6	LR	۷۸۴	۰/۵	۲۱/۴۸	۲۰/۲	۷/۳	+
I-280-7	RC	۷۸۴	۰/۵	۵۱/۶	۴۸/۸۴	۴/۵	+
I-280-8	RR	۷۸۴	۰/۵	۲۰/۵۴	۱۸/۴۴	۱۱/۴	+
I-280-9	LC	۷۸۴	۰/۸	۴۲/۷۸	۴۱/۶۶	۲/۷	+
I-280-10	LR	۷۸۴	۰/۸	۱۴/۷۸	۱۳/۴۲	۱۰/۱	+
I-280-11	RC	۷۸۴	۰/۸	۴۰/۶	۳۹/۵۲	۲/۷	+
I-280-12	RR	۷۸۴	۰/۸	۱۳/۹۲	۱۲/۷۸	۸/۹	+
I-280-13	LC	۷۸۴	۱/۰	۲۶/۱۴	۲۵/۷۴	۱/۱	~
I-280-14	LR	۷۸۴	۱/۰	۱۱/۳۴	۱۱/۱۴	۱/۸	~
I-280-15	RC	۷۸۴	۱/۰	۳۵/۷۲	۳۵/۹	۰/۵	~
I-280-16	RR	۷۸۴	۱/۰	۱۱/۷۴	۱۱/۲	۶/۵	+

جدول ۶ نتایج مربوط به حل نمونه‌های با اندازه ۲۸۰ نود و تعداد ۷۸۴ رنگ را نشان می‌دهد. برای این دسته، به غیر از ۴ نمونه که تفاوت معناداری بین دو الگوریتم رقیب نیست، در ۱۲ مورد، DivPGA به صورت معناداری از الگوریتم رقیب پیشی می‌گیرد.

جدول ۷ نتایج مربوط به حل نمونه با اندازه‌های ۵۳۲ نود و تعداد ۲۸۳۰ رنگ را نشان می‌دهد. در نمونه ۶ این دسته تفاوت معناداری بین دو الگوریتم رقیب نیست. در ۱۴ مورد از ۱۶ مورد این دسته، DivPGA به صورت معناداری از الگوریتم رقیب پیشی می‌گیرد. برای نمونه با چگالی ۰،۲ و نوع LC از این دسته DivP-GA مغلوب رقیب



شکل ۹: مقایسه DivP-GA و GA [50] با تعداد نسل ۱۰۰۰۰۰۰

با وجود آنکه در نمودار شکل ۹، واضح است که تعداد پله‌های نزولی متوسط هزینه برای DivP-GA بیشتر است، اما جهت بررسی بیشتر، آزمایش‌های مشابهی با تعداد نسل بیشتر ۲۰۰۰۰۰۰ و ۳۰۰۰۰۰۰ نیز انجام شد که نتیجه تمامی آن‌ها در جدول ۵ خلاصه می‌شود. با افزایش نسل از ۲۰۰۰۰۰۰ به ۳۰۰۰۰۰۰ هزینه رقیب DivP-GA از متوسط ۴۳۱۴۰/۸۲ به ۴۳۱۴۱/۶۸ افزایش می‌یابد که نشان از همگرایی این الگوریتم دارد. توجه داشته باشید که این احتمال وجود دارد که اگر دوباره ۵۰ اجرای دیگر با تعداد نسل ۳۰۰۰۰۰۰ برای این الگوریتم انجام شود، ممکن است متوسط هزینه حاصل از ۵۰ اجرا با نسل ۲۰۰۰۰۰۰ (یعنی ۴۳۱۴۰/۸۲) کمی بیشتر یا کمتر شود اما چیزی که مسلم است در ۵۰ اجرا با تعداد نسل ۳۰۰۰۰۰۰ الگوریتم [۵۰] نتوانست هزینه را بهبود دهد.

جدول ۵: رقابت DivP-GA با الگوریتم ژنتیک [50]

تعداد نسل	متوسط هر الگوریتم در ۵۰ اجرا		نتیجه آزمون آماری
	الگوریتم ژنتیک [50]	DivP-GA	
۱۰۰۰۰۰۰	۴۳۲۰۷/۸۸	۴۲۸۶۰/۹۴	+
۲۰۰۰۰۰۰	۴۳۱۴۰/۸۲	۴۲۷۹۰/۱۴	+
۳۰۰۰۰۰۰	۴۳۱۴۱/۶۸	۴۲۷۵۱/۱	+

۳-۶- نتایج برای مسئله فروشنده دوره گرد رنگارنگ

جهت بررسی کارایی استراتژی جایگزینی ارائه شده با استفاده از رویکرد جدید محاسبه تشابه برای مسئله فروشنده دوره گرد رنگارنگ، الگوریتم سه سطحی ارائه شده در [۵۲] با نام H-3Ph-EA لحاظ می‌شود. البته با توجه به آن که سطح آخر این الگوریتم صرفاً اعمال یک الگوریتم جستجوی محلی است، جهت حفظ شرایط عادلانه و بررسی صحیح قابلیت الگوریتم ژنتیک، برای هر دو سطح سوم در نظر گرفته نمی‌شود و در این قسمت با نام H-2Ph-EA به آن اشاره می‌شود. قابل ذکر است که DivP-GA نیز مشابه H-2Ph-EA بوده و هر دو از عملگرهای پیش‌فرض ارائه شده در مرجع [۵۲] استفاده می‌کنند، با این

نتایج آزمایش‌ها نشان داد که در حل مسائل بهینه‌سازی پیوسته، دقت استراتژی‌های پیشنهادی بهتر از دیگر روش‌های به‌روز بوده، طوری که در حل دوازده تابع بهینه‌سازی با تعداد ۱۰۰ متغیر، حداقل در پنج مورد به‌طور قابل‌ملاحظه‌ای از رقبای دیگر پیشی گرفت. نتایج آزمایش‌ها در حل مسائل بهینه‌سازی گسسته، کارایی استراتژی پیشنهادی جایگزینی بر مبنای روش جدید محاسبه معناگرایی تشابه را بیشتر آشکار کرد طوری که برای نمونه‌های مسئله فروشنده دوره‌گرد، استفاده از آن کارایی الگوریتم ژنتیک را در حل اکثر نمونه مسائل کتابخانه استاندارد، با اندازه‌های مختلف از ۵۰ تا ۱۰۰۰، بهبود بخشید. این مورد در حل ۴۸ نمونه مسئله از مسئله فروشنده دوره‌گرد رنگارنگ، با اندازه‌های مختلف از ۲۸۰ نود با ۷۸۰ رنگ تا ۱۰۰۰ نود با ۱۰۰۰۰ رنگ، بیشتر نمایان شد طوری که در برخی مسائل بهبود بیشتر از ۱۰ درصد بود.

جدول ۸: رقابت DivP-GA با H-2Ph-EA برای دسته مسائل با اندازه

۱۰۰۰ و تعداد رنگ ۱۰۰۰۰

نام	نوع	m	ρ	متوسط هر الگوریتم در ۵۰ اجرا		درصد بهبود	نتیجه آزمون آماری
				H-2Ph-EA	DivP-GA		
I-1000-1	LC	۱۰۰۰۰	۰/۲	۳۵۵/۴۶	۱۵۴ ۳۳۸	۵	+
I-1000-2	LR	۱۰۰۰۰	۰/۲	۲۱۸/۲	۱۷۲ ۲۱۲	۲/۵	+
I-1000-3	RC	۱۰۰۰۰	۰/۲	۳۲۹/۳	۱۷۲ ۳۱۴	۴/۶	+
I-1000-4	RR	۱۰۰۰۰	۰/۲	۲۱۵/۵۴	۲۰۹/۵	۲/۹	+
I-1000-5	LC	۱۰۰۰۰	۰/۵	۱۹۴/۵	۱۵۶ ۱۸۸	۳/۲	+
I-1000-6	LR	۱۰۰۰۰	۰/۵	۱۲۲/۴	۱۲۱/۷۴	۱/۴	~
I-1000-7	RC	۱۰۰۰۰	۰/۵	۱۸۲/۶۴	۱۷۹/۴	۱/۸	+
I-1000-8	RR	۱۰۰۰۰	۰/۵	۱۲۰/۷۸	۱۱۸	۲/۴	+
I-1000-9	LC	۱۰۰۰۰	۰/۸	۱۳۵/۸۸	۱۳۴/۵	۱	~
I-1000-10	LR	۱۰۰۰۰	۰/۸	۸۷/۳۲	۸۴/۱۴	۳/۷	+
I-1000-11	RC	۱۰۰۰۰	۰/۸	۱۳۲/۱۴	۱۸۲ ۱۲۷	۳/۴	+
I-1000-12	RR	۱۰۰۰۰	۰/۸	۸۵/۴۶	۸۳/۶	۲/۹	+
I-1000-13	LC	۱۰۰۰۰	۱/۰	۱۱۲	۱۱۰/۱۲	۱/۷	~
I-1000-14	LR	۱۰۰۰۰	۱/۰	۷۱/۱۲	۶۸/۸۴	۳/۳	+
I-1000-15	RC	۱۰۰۰۰	۱/۰	۱۱۱/۸۴	۱۱۰/۴	۱/۳	~
I-1000-16	RR	۱۰۰۰۰	۱/۰	۷۱/۶۲	۶۹/۴۶	۳/۱	+

سیاسگزاری

از تمامی کسانی که ما را در انجام تحقیق یاری نموده‌اند نهایت تشکر را داریم.

مراجع

[۱] مجید محمدپور، حمید پروین، «الگوریتم ژنتیک آشوب‌گونه

می‌شود. البته این بدان معنا نیست که نتیجه مشابه را بتوان برای دیگر نمونه‌های از نوع LC بسط داد. همان‌طور که از جداول این زیر بخش مشخص است، DivP-GA در حل اکثر نمونه‌ها با نوع LC که البته اندازه یا چگالی متفاوت دارند به‌صورت معناگذاری از الگوریتم رقیب پیشی می‌گیرد.

جدول ۷: رقابت DivP-GA با H-2Ph-EA برای دسته مسائل با اندازه

نام	نوع	m	ρ	متوسط هر الگوریتم در ۵۰ اجرا		درصد بهبود	نتیجه آزمون آماری
				H-2Ph-EA	DivP-GA		
I-532-1	LC	۲۸۳۰	۰/۲	۲۰۱/۴۲	۲۰۵/۱	-	-
I-532-2	LR	۲۸۳۰	۰/۲	۹۴/۴۲	۹۰/۸۸	۳/۹	+
I-532-3	RC	۲۸۳۰	۰/۲	۱۹۸/۷	۱۹۴/۲۲	۲/۳	+
I-532-4	RR	۲۸۳۰	۰/۲	۹۹/۴	۹۶/۳۸	۳/۲	+
I-532-5	LC	۲۸۳۰	۰/۵	۱۴۲/۳	۱۳۹/۵۲	۲/۷	+
I-532-6	LR	۲۸۳۰	۰/۵	۵۵/۲۸	۵۴	۲/۴	~
I-532-7	RC	۲۸۳۰	۰/۵	۱۴۶/۷۸	۱۴۱/۲۸	۳/۹	+
I-532-8	RR	۲۸۳۰	۰/۵	۵۶/۴	۵۲/۹	۵/۹	+
I-532-9	LC	۲۸۳۰	۰/۸	۱۱۷/۹	۱۱۵/۷۶	۱/۸	+
I-532-10	LR	۲۸۳۰	۰/۸	۴۱/۴	۴۰/۸۶	۱/۳	~
I-532-11	RC	۲۸۳۰	۰/۸	۱۲۱/۴۶	۱۱۶/۴۶	۴/۳	+
I-532-12	RR	۲۸۳۰	۰/۸	۳۹/۶	۳۷/۳۸	۴/۵	+
I-532-13	LC	۲۸۳۰	۱/۰	۱۱۷/۵۸	۱۱۲/۸۶	۴/۲	+
I-532-14	LR	۲۸۳۰	۱/۰	۳۳/۲۲	۳۲/۱۴	۴/۹	+
I-532-15	RC	۲۸۳۰	۱/۰	۱۱۷/۶	۱۱۲/۲۴	۴/۸	+
I-532-16	RR	۲۸۳۰	۱/۰	۳۴/۲	۳۲/۱۶	۵/۸	+

۵۳۲ و تعداد رنگ ۲۸۳۰

جدول ۸ نتایج مربوط به حل نمونه‌های با اندازه ۱۰۰۰ نود و تعداد ۱۰۰۰۰ رنگ را نشان می‌دهد. برای این دسته، مشابه دسته اول، به غیر از ۴ نمونه که تفاوت معناگذاری بین دو الگوریتم رقیب نیست، در ۱۲ مورد، DivPGA به‌صورت معناگذاری از الگوریتم رقیب پیشی می‌گیرد.

۷- نتیجه

در این مقاله جهت افزایش کارایی الگوریتم ژنتیک، راه‌کارهایی را جهت حفظ تنوع جمعیت به‌کار گرفته شد. جهت حفظ تنوع، الگوریتم‌هایی برای نحوه جایگزینی و نحوه انتخاب بر اساس تشابه بین دو جواب طراحی شد. باوجود راحتی به‌کارگیری هر کدام از این استراتژی‌ها روی مسائل بهینه‌سازی پیوسته، اعمال آن روی مسائل بهینه‌سازی گسسته میسر نیست. چرا که روش مناسبی جهت محاسبه تشابه بین دو جواب بهینه‌سازی گسسته وجود ندارد و از طرفی محاسبه تشابه با روش مبتنی یا اعمال یک روش مشابه برای محاسبه تشابه در تمامی انواع مسائل بهینه‌سازی گسسته منطقی نمی‌نماید. لذا این مقاله جهت محاسبه تشابه بین دو جواب مسئله بهینه‌سازی گسسته رویکردهای معناگرایی جدیدی ارائه داد طوری که برای هر کدام از دو مسئله بهینه‌سازی گسسته روش متفاوتی با توجه به ذات مسئله پیشنهاد شد.

- india, 2009.
- [17] H. Ishibuchi, Y. Sakane, N. Tsukamoto and Y. Nojima, "Implementation of cellular genetic algorithms with two neighborhood structures for single-objective and multi-objective optimization," *Soft computing*, vol. 15, no. 9, pp. 1749-1767, 2011.
- [18] B. Dorronsoro and P. Bouvry, "Cellular genetic algorithms without additional parameters," *The journal of supercomputing*, vol. 63, no. 3, pp. 816-835, 2013.
- [19] G. Dick and P. A. Whigham, "Spatially-structured evolutionary algorithms and sharing: do they mix," in *Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time (Natural Computing Series)*, New York, Springer-Verlag, 2005.
- [20] K. M. Bryden, D. A. Ashlock, S. Corns and S. J. Willson, "Graph-based evolutionary algorithms," *IEEE transactions on evolutionary computation*, vol. 10, no. 5, pp. 550-567, 2006.
- [21] M. Crepinsek, S. H. Liu and M. Mernik, "Exploration and exploitation in evolutionary algorithms: a survey," *Acm computing surveys*, vol. 45, no. 3, pp. A1-A33, 2013.
- [22] E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE transactions on evolutionary computation*, vol. 9, no. 2, pp. 126-142, 2005.
- [23] H. B. Amor and A. Rettinger, "Intelligent exploration for genetic algorithms using self-organizing maps in evolutionary computation," in *Gecco*, Washington, DC, USA, 2005.
- [24] M. Črepinšek and M. Mernik, "Analysis of exploration and exploitation in evolutionary algorithms by ancestry trees," *International journal of innovative computing and applications*, vol. 3, no. 1, pp. 11-19, 2011.
- [25] T. Park and K. R. Ryu, "A dual-population genetic algorithm," *IEEE transactions on evolutionary computation*, vol. 14, no. 6, pp. 865-884, 2010.
- [26] S. Ronald, "More Distance Functions for Order-Based Encoding," in *IEEE international conference on evolutionary computation*, 1998.
- [27] G. S. Hornby, "Alps: the age-layered population structure for reducing the problem of premature convergence," in *The 8th annual conference on genetic and evolutionary computation*, New York, 2006.
- [28] Y. Li and X. Zeng, "Multi-population co-genetic algorithm with double chain-like agents structure for parallel global numerical optimization," *Applied intelligence*, vol. 32, no. 3, pp. 292-310, 2010.
- [29] T. Park and K. R. Ryu, "A dual population genetic algorithm with evolving diversity," in *IEEE congress on evolutionary computation*, 2007.
- [30] G. Dick, "The spatially-dispersed genetic algorithm," in *GECCO'03 Proceedings of the 2003 international conference on Genetic and evolutionary computation: Part II*, 2003.
- [31] T. Park, R. Choe and K. R. Ryu, "Dual-population genetic algorithm for nonstationary optimization," in *The 10th annual conference on genetic and evolutionary computation*, 2008.
- [32] Y. Li and X. Zeng, "Feature selection method with multi-population agent genetic algorithm," in *Advances in Neuro-Information Processing*, Springer-Verlag, 2009, pp. 493-500.
- مبتنی بر حافظه و خوشه‌بندی برای حل مسائل بهینه‌سازی پویا»،
مجله مهندسی برق دانشگاه تبریز، ۴۶، ۳، ۳۱۸-۲۹۹، ۱۳۹۵.
- [۲] فرناز درخشان، فاطمه خضولو، «طراحی و پیاده‌سازی یک روش تلفیقی هوشمند برای کنترل ترافیک شهری در تقاطع‌ها»،
مجله مهندسی برق دانشگاه تبریز، ۴۷، ۳، ۱۰۲۴-۱۰۱۳، ۱۳۹۶.
- [۳] علی حسامی نقشبندی، هیوا شمس، «بخش بار بهینه مقید به پایداری سیگنال کوچک با استفاده از الگوریتم ژنتیک»،
مجله مهندسی برق دانشگاه تبریز، ۴۷، ۳، ۹۵۰-۹۳۹، ۱۳۹۶.
- [4] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *Ieee transactions on evolutionary computation*, vol. 6, no. 4, pp. 257-271, 1999.
- [5] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: nsga-ii," *IEEE transactions on evolutionary computation*, vol. 3, no. 2, pp. 182-197, 2002.
- [6] Zaharie, Daniela, "Control of population diversity and adaptation in differential evolution algorithms," in *9th international conference on soft computing*, Brno, czech republic, 2003.
- [7] O. Lehman, K. O. Stanley and R. Miikkulainen, "Effective diversity maintenance in deceptive domains," in *Genetic and evolutionary computation conference*, New York, 2013.
- [8] M. Hutter, "Fitness uniform selection to preserve genetic diversity," Technical Report IDSIA-01-01, Manno-Lugano, Switzerland, 2001.
- [9] M. Laumanns, L. Thiele, K. Deb and E. Zitzler, "Combining convergence and diversity in evolutionary multi-objective optimization," *Evolutionary computation*, vol. 10, no. 3, pp. 263-282, 2002.
- [10] A. Toffolo and E. Benini, "Genetic diversity as an objective in multi-objective evolutionary algorithms," *Evolutionary computation*, vol. 11, no. 2, pp. 151-167, 2003.
- [11] C. C. D. Ronco and E. Benini, "A simplex crossover based evolutionary algorithm including the genetic diversity as objective," *Applied soft computing*, vol. 13, no. 4, pp. 2104-2123, 2013.
- [12] H. Xie and M. Zhang, "Impacts of sampling strategies in tournament selection for genetic programming," *Soft computing*, vol. 16, no. 4, pp. 615-633, 2012.
- [13] R. K. Ursem, "Diversity-guided evolutionary algorithms," in *PPSN VII Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, Granada, Spain, 2002.
- [14] K. Q. Zhu, "A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows," in *IEEE international conference on tools with artificial intelligence*, 2003.
- [15] X. Zhao and J. Hao, "Exploration/exploitation tradeoff with cell-shift and heuristic crossover for evolutionary algorithms," *Journal of system science & complexity*, vol. 20, no. 1, pp. 66-74, 2007.
- [16] R. Patel, M. M. Raghuvanshi and A. N. Jaiswal, "Modifying genetic algorithm with species and sexual selection by using k-means algorithm," in *IEEE international advance computing conference*, Patiala,

- technology, vol. 11, no. 3, pp. 323-327, 2004.
- [45] L. J. Eshelman, "The CHC adaptive search algorithm. How to have safe search when engaging in non-traditional genetic recombination," in *Foundation of Genetic Algorithms I*, San Mateo, California, Morgan Kaufmann, 1991, pp. 265-283.
- [46] S. Kundu, S. Biswas, S. Das and P. N. Suganthan, "Crowding-based local differential evolution with speciation-based memory archive for dynamic multimodal optimization," in *the 15th annual conference on Genetic and evolutionary computation*, Amsterdam, The Netherlands, 2013.
- [47] M. M. Kordmahalleh, A. Homaifar and D. B. Kc, "Hierarchical multi-label gene function prediction using adaptive mutation in crowding niching," in *IEEE 13th international conference on bioinformatics and bioengineering*, Chania, 2013.
- [48] H. Satoh, M. Yamamura and S. Kobayashi, "Minimal generation gap model for GAs considering both exploration and exploitation," in *the 4th International Conference on Fuzzy Logic, Neural Nets and Soft Computing*, Iizuka, Japan, 1996.
- [49] P. H. Tang and M. H. Tseng, "Adaptive directed mutation for real-coded genetic algorithms," *Applied soft computing*, vol. 13, no. 1, pp. 600-614, 2013.
- [50] H. Ismkhan and K. Zamanifar, "Developing Programming Tools to Handle Traveling Salesman Problem by Three Object Oriented Languages," *Applied Computational Intelligence and Soft Computing*, 2014.
- [51] G. Reinelt, "Tspplib—a traveling salesman problem library," *Orsa journal on computing*, vol. 3, pp. 376-384, 1991.
- [52] H. Ismkhan, "Effective three-phase evolutionary algorithm to handle the large-scale colorful traveling salesman problem," *Expert Systems With Applications*, vol. 67, p. 148-162, 2017.
- [53] Y. Chen, N. Cornick, A. O. Hall, R. Shajpal, J. Silberholz, I. Yahav and B. L. Golden, "Comparison of Heuristics for Solving the Gmlst Problem," in *Telecommunications Modeling, Policy, and Technology*, Springer, 2008, pp. 191-217.
- [33] S. Tsutsui, A. Ghosh, D. Corne and Y. Fujimoto, "A real coded genetic algorithm with an explorer and an exploiter populations," in *The 7th international conference on genetic algorithms*, San Francisco, CA, 1997.
- [34] E. Alba, H. Alfonso and B. Dorronsoro, "Advanced models of cellular genetic algorithms evaluated on sat," in *the 7th Annual Conference on Genetic and Evolutionary Computation*, Washington DC, USA, 2005.
- [35] J. Song and D. H. Kim, "Subcarrier and bit allocation scheme for the ma problem based on the ant colony optimization to minimize power consumption in ofdma systems," *International journal of innovative computing, information and control*, vol. 7, no. 8, pp. 4755-4764, 2011.
- [36] L. Lin and M. Gen, "Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation," *Soft computing*, vol. 13, no. 2, pp. 157-168, 2009.
- [37] M. Lozano, F. Herrera and J. R. Cano, "Replacement strategies to preserve useful diversity in steady-state genetic algorithms," *Information sciences*, vol. 178, no. 23, pp. 4421-4433, 2008.
- [38] O. J. Mengshoel and D. E. Goldberg, "The crowding approach to niching in genetic algorithms," *Evolutionary computation*, vol. 16, no. 3, pp. 315-354, 2008.
- [39] S. W. Mahfoud, "Niching methods for genetic algorithms," Doctoral Dissertation, University of Illinois at Urbana-Champaign, Champaign, 1995.
- [40] V. K. Koumousis and C. P. Katsaras, "A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance," *Ieee transactions on evolutionary computation*, vol. 10, no. 1, pp. 19-28, 2006.
- [41] A. Zhou, Y. Jin, Q. Zhang and B. Sendhoff, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *the 4th International Conference on Evolutionary Multi-criterion Optimization*, Matsushima, Japan, 2007.
- [42] J. Yao, N. Kharm and P. Grogono, "Bi-objective multipopulation genetic algorithm for multimodal function optimization," *IEEE transactions on evolutionary computation*, vol. 14, no. 1, pp. 80-102, 2010.
- [43] H. Ismkhan, "A Novel Intelligent Algorithm to Control Mutation Rate Using the Concept of Local Trap," *New Generation Computing*, vol. 34, no. 1, pp. 177-192, 2016.
- [44] L. M. Yi, C. Z. Xing and S. G. Yun, "An adaptive genetic algorithm with diversity-guided mutation and its global convergence property," *Journal of central south university*

^ Mutation

^ Selection

^ Exploitation

^ Evolutionary algorithms

^ Parent

^ Travelling Salesman Problem (TSP)

^ Minimum Labeling Hamiltonian Cycle Problem (MLHCP)

^ Labeled Traveling Salesman Problem (LTSP)

^ Colorful Traveling Salesman Problem (CTSP)

^ Self-adaptive

^ Cellular GA (CGA)

^ Spatially-dispersed GA (sdGA)

زیرنویس‌ها

^ Premature convergence

^ Exploration

^ Global optimum

^ Fitness

۵ در این مقاله منظور از کلمه "بهتر" اشاره به مقدار برآزش بیشتر می‌باشد. برای نمونه

وقتی X از Y بهتر است یعنی مقدار برآزش X از Y بیشتر می‌باشد. در ادامه نیز مفهوم

کلمات "بدتر"، "بهترین" و "بدترین" به صورت مشابه تعریف می‌شود.

^ Generation

^ Crossover

- ^{۲۰} Bi-objective multi-population GA (BMPGA)
- ^{۲۱} Dual-Population GA (DP-GA)
- ^{۲۲} Inbred offspring
- ^{۲۳} Crossbred offspring
- ^{۲۴} cell-shift crossover
- ^{۲۵} highly disruptive crossover operator (HUX)
- ^{۲۶} Heuristic
- ^{۲۷} Minkowski distance
- ^{۲۸} Manhattan distance
- ^{۲۹} Euclidean distance
- ^{۳۰} Hamming distance
- ^{۳۱} Contribution of Diversity
- ^{۳۲} One-point crossover
- ^{۳۳} Improved Greedy Crossover
- ^{۳۴} Double-Bridge