

Journal of Soft Computing and Information Technology (JSCIT)

Babol Noshirvani University of Technology, Babol, Iran

Journal Homepage: [jscit.nit.ac.ir](http://jscit.nit.ac.ir)

Volume 9, Number 3, Fall 2020, pp. 14-26

Received: 07/28/2019, Revised: 01/06/2020, Accepted: 02/13/2020



## Improvement of Genetic Algorithm Using a Fuzzy Control Combined with Coevolutionary Algorithm

Mohsen Ghalehnoie

Department of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood, Iran.  
ghalehnoie@shahroodut.ac.ir

Corresponding author address: Mohsen Ghalehnoie, Faculty of Electrical and Robotic Engineering, Shahrood University of Technology, Shahrood, Iran, Post Code: 36199 – 95161.

**Abstract-** In order to achieve the best performance in genetic algorithm, proper determination of parameters is necessary. This paper addresses the intelligent determination of the crossover probability between two selected parents in each generation. Unlike most existing techniques that utilize the diversity characteristics of each generation to determine the crossover probability throughout the current generation, this paper defines some novel phenotype and genotype features, and develops a zero-order Takagi-Sugeno fuzzy controller to derive the proper crossover probability for each selected parent pair. As such, each pair has a unique probability parameter that results in the flexibility of the standard genetic algorithm depending on the region being searched and avoids premature convergence. In addition, in the proposed method, the consequent part of the fuzzy rules is not fixed but is generated through a coevolutionary process and simultaneously with the decision variables of the optimization problem. This enhances the efficiency of the proposed method. The simulation results on a set of optimization benchmarks show the performance of this method. Its effectiveness is also investigated by applying it to the complicated problem of terrain avoidance/terrain following fly.

**Keywords-** Coevolutionary algorithm, Crossover probability, Genetic algorithm, Genotype characteristic, Phenotype characteristic, Takagi-Sugeno fuzzy controller, Terrain avoidance/terrain following fly.



## بهبود الگوریتم ژنتیک با استفاده از ترکیب منطق فازی و الگوریتم هم تکاملی

محسن قلعه نوئی

دانشکده مهندسی برق و رباتیک، دانشگاه صنعتی شاهرود، شاهرود، ایران.

ghalehnoie@shahroodut.ac.ir

\* نشانی نویسنده مسئول: محسن قلعه نوئی، شاهرود، میدان هفتم تیر، دانشگاه صنعتی شاهرود، پردیس مهندسی و فناوری‌های نوین، دانشکده مهندسی برق و رباتیک، کد پستی: ۳۶۱۹۹-۹۵۱۶۱

چکیده- به منظور رسیدن به بهترین عملکرد در الگوریتم ژنتیک، تعیین مناسب پارامترها ضروری است. این مقاله بر تعیین هوشمند پارامتر احتمال باز ترکیب بین هر دو والد انتخابی در هر نسل دلالت دارد. بر خلاف اغلب روش‌های موجود که در آنها ویژگی‌های پراکندگی هر نسل به منظور تعیین احتمال باز ترکیب در کل نسل جاری استفاده می‌شود، این مقاله ویژگی‌های فنوتیپ و ژنوتیپ جدیدی را برای هر زوج والد انتخابی در نسل جاری تعریف نموده و به کمک یک کنترل کننده فازی تاکاگی-سوگنوی مرتبه صفر، پارامتر احتمال باز ترکیب تعیین می‌گردد. بدین ترتیب، هر زوج والد انتخابی در هر نسل دارای پارامتر باز ترکیب منحصر بفردی است که سبب انعطاف الگوریتم ژنتیک بسته به ناحیه مورد جستجو شده و از همگرایی زودرس الگوریتم ژنتیک جلوگیری می‌نماید. علاوه بر این، در روش پیشنهادی، جمله مربوط به بخش آنگاه در هر یک از قواعد فازی ثابت نبوده بلکه از طریق یک فرآیند تکاملی و همزمان با دیگر متغیرهای تصمیم تولید می‌شود. این عمل سبب افزایش بیشتر کارایی این روش می‌گردد. نتایج شبیه سازی بر روی مجموعه‌ای از مسائل بهینه سازی استاندارد، عملکرد مطلوب این روش را نشان می‌دهد. همچنین، کارایی این روش با اعمال بر روی مسأله پیچیده پرواز تعقیب عوارض زمین بررسی می‌گردد.

واژه‌های کلیدی: احتمال باز ترکیب، الگوریتم ژنتیک، الگوریتم هم تکاملی، پرواز تعقیب عوارض زمین، کنترل فازی تاکاگی-سوگنوی، ویژگی ژنوتیپ، ویژگی فنوتیپ.

### ۱- مقدمه

همچنین بسته به دانش کاربر از مسأله مورد نظر، اجرای چنین راهکاری یا غیر ممکن است و یا حداقل زمان بر و پیچیده خواهد بود. لذا محققان سعی دارند که با استفاده از تکنیک‌های مختلف و به صورت خودکار این پارامترها را تنظیم نمایند.

مرور پیشینه تحقیقات در این حوزه نشان می‌دهد که دو دسته فعالیت عمده برای غلبه بر مشکلات موجود در الگوریتم ژنتیک پیشنهاد شده است [۱۳-۵]. گروه اول بر روی اصلاح نوع عملگرها بویژه دو عملگر باز ترکیب و جهش تمرکز دارند. بطور مثال در [۱۴] نشان داده شده است که عملگر باز ترکیب چند نقطه‌ای<sup>۲</sup> به مراتب عملکرد بهتری از باز ترکیب تک نقطه‌ای<sup>۱</sup> دارد. همچنین در [۱۵]، یک الگوریتم تطبیقی برای انتخاب مناسب عملگرهای باز ترکیب و

الگوریتم ژنتیک در حل مسائلی که راه حل مشخصی برای آنها قابل تصور نبوده و یافتن جواب‌های بهینه در آنها سخت است، کاربردهای فراوانی دارد [۴-۱]. استفاده از الگوریتم ژنتیک مشکلاتی نظیر سرعت جستجو، دقت پایین به دلیل همگرایی زودرس و گرفتار شدن در بهینه محلی را به دنبال دارد. دلیل عمده این مشکلات، انتخاب نامناسب عملگرهای باز ترکیب<sup>۱</sup> و جهش<sup>۲</sup> و تنظیم نادرست پارامترهای کنترلی آنهاست. در این راستا، محققان عقیده دارند که تنظیم پارامترها و انتخاب عملگرهای مناسب، تحت کنترل یک کاربر، بهترین راه حل برای غلبه بر مشکلات مذکور می‌باشد. با توجه به پیچیدگی و گستردگی الگوهای مربوط به مسائل بهینه سازی،

جهش در حین اجرای الگوریتم ژنتیک ارائه شده است که عملکرد بهتری را در مقایسه با سایر روش‌های دیگر نشان می‌دهد. در [۱۶] نیز روشی مشابه، مبتنی بر کنترل فازی، برای تعیین نوع عملگر بازترکیب در هر نسل از الگوریتم ژنتیک ارائه شده است تا از همگرایی زودرس ممانعت بعمل آید. در این روش، پاره‌ای از ویژگی‌های فنوتیپ<sup>۵</sup> و ژنوتیپ<sup>۶</sup> مربوط به پراکندگی جمعیت در نسل جاری به عنوان ورودی در سیستم فازی دریافت شده و بر اساس یک سری قواعد فازی از پیش تعریف شده، نوع عملگر بازترکیب به عنوان خروجی تعیین می‌شود. سپس بر اساس این عملگر، نسل آتی از روی نسل جاری تولید می‌گردد. این مقاله نشان می‌دهد که انتخاب معیارهای فنوتیپ (یعنی ویژگی‌های قابل مشاهده در عملکرد یک کروموزوم یا عملکرد مجموعه‌ای از کروموزوم‌ها مبتنی بر میزان برازندگی آنها در یک نسل) و یا معیارهای ژنوتیپ (یا همان ویژگی‌های ساختاری بر اساس محتوای عددی ژن‌های یک کروموزوم و یا تفاوت این محتوا بین کروموزوم‌های مختلف در یک نسل) به تنهایی نمی‌تواند معیار مناسبی برای میزان پراکندگی نسل و تعیین نوع عملگر باشد.

جدا از نوع عملگرها، انتخاب پارامترهای الگوریتم ژنتیک بویژه پارامترهای مربوط به احتمال بازترکیب ( $p_c$ ) و احتمال جهش ( $p_m$ ) به صورت بحرانی بر رفتار و کارایی الگوریتم ژنتیک موثر است [۱۷]. لذا دسته دوم فعالیت‌ها، از تنظیم هوشمندانه و تطبیقی این پارامترها برای غلبه بر مشکلات الگوریتم ژنتیک بهره می‌گیرند. بطور کلی این تطبیق می‌تواند برای هر کروموزوم و یا برای کل کروموزوم‌های یک نسل انجام پذیرد. اصطلاحاً، این دو سطح تطبیق را به ترتیب سطح کروموزوم و سطح الگوریتم می‌نامند [۱۶]. بطور مثال در [۱۸]، میزان برازندگی نسل جاری به یک کنترل‌کننده فازی ارسال شده و سپس میزان پارامترهای الگوریتم ژنتیک از طریق قواعد فازی مشخصی تعیین می‌شود. با معلوم شدن پارامترها، عملگرهای بازترکیب و جهش بر روی نسل جاری اعمال شده و نسل آتی تولید می‌گردد. از آنجا که پارامترهای تنظیم شده برای اعمال عملگرهای ژنتیک بر روی همه کروموزوم‌های نسل جاری یکسان فرض می‌شود، این روش نوعی از تطبیق در سطح الگوریتم است. بطور مشابه، در [۱۹] نیز، با در نظر گرفتن درصد بهبود میانگین برازندگی در دو نسل متوالی، پارامترهای  $p_m$  و  $p_c$  به صورت فازی تنظیم شده و کاربرد آن در برنامه‌ریزی حمل و نقل نشان داده می‌شود. معیارهای دیگری همانند درصد تغییرات بهترین برازندگی در دو نسل آخر، درصد انحراف میانگین برازندگی‌ها از بهترین برازندگی در آخرین نسل و تعداد تکرارهای بدون بهبود، از جمله معیارهایی است که به عنوان ورودی جهت تنظیم و تطبیق فازی پارامترهای

$p_c$  و  $p_m$  در سطح الگوریتم کاربرد دارند [۲۰]. علاوه بر این، در [۲۱]، معیار واریانس برازندگی کروموزوم‌ها در هر نسل به عنوان معیار پراکندگی جهت تنظیم پارامترهای جهش و بازترکیب توسط یک کنترل‌کننده فازی ثابت مورد استفاده قرار گرفته است. همچنین در مرجع [۲۲] با استفاده از خوشه‌بندی کروموزوم‌های نسل جاری و بر اساس بزرگ یا کوچک بودن اندازه خوشه‌ای که شامل بدترین و بهترین کروموزوم است، میزان پارامتر  $p_c$  و  $p_m$  از طریق یک سیستم فازی ثابت تعیین می‌گردد. در این روش، فاصله اقلیدسی کروموزوم‌ها (به عنوان یک ویژگی ژنوتیپ) برای خوشه‌بندی هر نسل استفاده شده است. قابل ذکر است که در مواردی نیز ظاهراً پارامترهای الگوریتم ژنتیک توسط روابط تطبیقی ابتکاری تنظیم می‌گردند ولی با دقت بیشتر در آنها مشاهده می‌شود که این روابط نیز در حقیقت نگاشت ورودی/خروجی یک سیستم فازی ثابت هستند. از آن جمله می‌توان به [۲۳] اشاره داشت که در آن برازندگی کروموزوم‌هایی که در فرآیند بازترکیب شرکت دارند با میانگین برازندگی کروموزوم‌های نسل جاری و همچنین کمترین یا بیشترین برازندگی در همان نسل قیاس شده و پارامتر  $p_c$  تطبیق می‌یابد. برای آشنایی بیشتر با سایر روش‌های تطبیق پارامترها و تنظیم نوع عملگرها، خواننده می‌تواند به مقالات مروری [۲۴] و [۲۵] رجوع نماید.

به‌رحال در اغلب فعالیت‌های موجود نکاتی قابل تأملی وجود دارد. (الف) بیشتر فعالیت‌ها در سطح الگوریتم می‌باشد. اگرچه این روش‌ها از نظر بار محاسباتی مقرون بصره هستند ولی هنگامی که ویژگی‌های مسأله در نواحی مختلف جستجو تفاوت داشته باشد، مناسب نخواهد بود. بدین ترتیب، مناسب‌تر است که عملیات تنظیم پارامترها در سطح کروموزوم انجام شوند. (ب) همانطور که در [۱۶] بیان می‌شود، برای درک بهتر از میزان پراکندگی جمعیت در یک نسل، باید ویژگی‌های فنوتیپ و ژنوتیپ را بطور همزمان در نظر گرفت. لذا یافتن چنین ویژگی‌هایی در سطح کروموزوم‌ها ضروری است. (ج) اگرچه منطق فازی و کنترل فازی ابزار قدرتمندی برای شبیه‌سازی عملکرد یک فرد خبره در تعیین نوع عملگرها و تنظیم پارامترهای مربوطه می‌باشد ولی باید توجه داشت که رسیدن به قواعد فازی مناسب، نیازمند تجربه فراوان بوده و از مسأله‌ای به مسأله دیگر متفاوت است. لذا، دستیابی به مکانیزمی که چنین پایگاه قواعدی را به صورت خودکار و بر اساس معیارهای تعریف شده تشکیل دهد، مفید است.

بر همین اساس، در این مقاله سعی شده تا راه‌حل نوینی در سطح کروموزوم ارائه گردد. در اینجا، فرض بر ثابت بودن نوع عملگرهای ژنتیک بوده و تنها پارامترهای مربوطه، به صورت فازی تنظیم می‌-

داشت که در آن ژن‌های هر فرد در جمعیت بجای بیت‌های باینری به صورت کوانتوم-بیت‌ها هستند و بدین ترتیب هر فرد در جامعه، خود تداعی‌گر چندین عضو باینری سنتی خواهد بود [۳۱-۲۸].

ادامه این مقاله بدین صورت سازماندهی می‌شود. در بخش ۲، الگوریتم ژنتیک استاندارد و پارامترهای مورد استفاده در این مقاله شرح داده می‌شود. کنترل‌کننده‌های فازی توسعه داده شده برای تنظیم پارامتر احتمال بازترکیب در سطح کروموزوم، در بخش ۳ تبیین می‌شود. نتایج پیاده‌سازی این کنترل‌کننده‌ها بر روی چندین مسأله بهینه‌سازی استاندارد و قیاس آنها با الگوریتم ژنتیک استاندارد، در بخش ۴ آورده شده است. مسأله پرواز تعقیب عوارض زمین و نتایج اعمال روش پیشنهادی بر روی آن در همین بخش ارائه می‌گردد. در نهایت، نتیجه‌گیری در بخش ۵ صورت می‌گیرد.

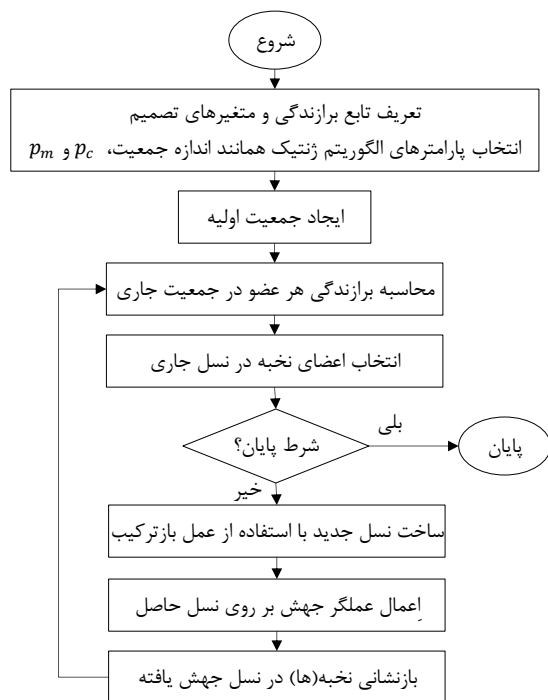
## ۲- الگوریتم ژنتیک استاندارد

شکل ۱، ساختار استاندارد الگوریتم ژنتیک به منظور حل مسائل بهینه‌سازی را نشان داده می‌شود [۳۲، ۳۳]. در این ساختار، هر نسل، شامل تعدادی کروموزوم است که محتوای عددی ژن‌های هر کروموزوم، معادل محتوای عددی یکی از متغیرهای تصمیم در مسأله بهینه‌سازی می‌باشد. لذا، با معلوم بودن هر کروموزوم، میزان برازندگی آن بر اساس تابع هدف در مسأله بهینه‌سازی تعیین می‌گردد. در صورتی که این برازندگی‌ها شرط پایان را برآورده نسازند، نسل آتی با استفاده از عملگرهای بازترکیب، جهش و نخبه‌گزینی تولید می‌شود. برای این منظور، پس از انتخاب نخبه‌ها، دو والد به

گردند. از آنجا که در جمعیت‌های بزرگ، عملگر بازترکیب بر روی همگرایی زودرس اثر بیشتری دارد در حالی که در جمعیت‌هایی با تعداد افراد کم، عملگر جهش موثرتر است [۲۶] و نظر به این واقعیت که معمولاً تعداد افراد هر نسل نسبتاً زیاد انتخاب می‌شود، این مقاله تنها بر روی تنظیم فازی احتمال بازترکیب بین دو والد در نسل جاری تمرکز دارد. استراتژی مشابهی توسط نویسنده در [۲۷] آورده شده است جایی که معیار برازندگی هر والد (به عنوان یک ویژگی فنوتیپ) تنها ورودی مورد نیاز در یک کنترل‌کننده فازی خود تنظیم به منظور تعیین میزان احتمال بازترکیب  $p_c$  است. در مقابل، در مقاله حال حاضر، ورودی‌ها در سیستم فازی عبارتند از میزان شباهت برازندگی دو والد به یکدیگر (ویژگی فنوتیپ) و میزان نزدیکی اقلیدسی آنها (ویژگی ژنوتیپ). در اینجا نیز، مشابه با ایده موجود در [۲۷]، پایگاه قواعد فازی توسط یک الگوریتم هم‌تکاملی شکل می‌یابد. در نهایت روش پیشنهادی بر روی چند مسأله بهینه‌سازی استاندارد اعمال شده و عملکرد آن در مقایسه با حالتی که پایگاه قواعد بصورت غیرتکاملی و تجربی تنظیم شده است و همچنین با الگوریتم ژنتیک استاندارد مقایسه می‌شود. علاوه بر این، این الگوریتم بر روی مسأله پرواز تعقیب عوارض زمین برای هدایت موشک‌های کروموزوم اعمال می‌گردد.

بطور خلاصه، ویژگی‌ها و نوآوری‌های این مقاله را می‌توان در سه بخش بیان نمود. (۱) تنظیم پارامتر احتمال بازترکیب بین هر دو والد انتخابی بطور مستقل از سایر والدین. (۲) تعریف ویژگی‌های فنوتیپ و ژنوتیپ جدید که نشان‌دهنده میزان شباهت دو والد انتخابی است؛ بر خلاف روش‌های موجود که چنین ویژگی‌هایی را برای کل یک نسل ارائه می‌دهند و بدین ترتیب پارامتر احتمال بازترکیب بین هر دو والد انتخابی در آن نسل یکسان می‌باشد. (۳) خود تنظیم بودن پایگاه قواعد فازی بگونه‌ای که بخش «آنگاه» از طریق یک الگوریتم هم‌تکاملی همزمان با مسأله بهینه‌سازی اصلی شکل می‌یابد.

لازم به ذکر است که گرچه روش پیشنهادی به دلیل بار محاسباتی ناشی از پیچیدگی الگوریتم طولانی می‌باشد ولی در حل مسائل برون خط که زمان نسبت به دقت پاسخ از اهمیت پایین‌تری برخوردار است، کارا خواهد بود. همچنین به لطف پیشرفت رایانه‌های کوانتومی می‌توان امید داشت که بتوان از این روش در مسائل برخط نیز استفاده نمود. از طرف دیگر، با رشد استفاده از مفاهیم کوانتوم در الگوریتم‌های تکاملی می‌توان سرعت اجرا و همگرایی این روش را چندین برابر نمود تا قابل استفاده در رایانه‌های سنتی باشد. امروزه این رویکرد به سرعت در حال رشد می‌باشد. بطور مثال می‌توان به نسخه‌های کوانتومی الگوریتم ژنتیک و الگوریتم مورچگان اشاره



شکل ۱: فلوچارت الگوریتم استاندارد بر اساس مراجع [۳۲، ۳۳].

بیشترین و متوسط برزندگی‌ها در نسل جاری و یا نسل پیشین می‌باشد. نسل جاری توسط اندیس  $t$  و نسل پیشین با اندیس  $t - 1$  مشخص است. علاوه بر این،  $HD(C_{f_{max,t}}, C_{f_{min,t}})$  بیان‌گر فاصله همینگ<sup>۶</sup> بین دو کروموزوم با بیشترین و کمترین برزندگی در نسل جاری است و در مواردی کاربرد دارد که متغیرهای تصمیم به صورت باینری کدگذاری<sup>۸</sup> شده باشند. بدین ترتیب، می‌توان روابط  $T_1$  الی  $T_3$  را ویژگی فنوتیپ و  $T_4$  را یک معیار ژنوتیپ دانست. اگرچه این ویژگی‌ها از نسلی به نسل دیگر متفاوت است ولی برای کلیه کروموزوم‌های موجود در یک نسل یکسان می‌باشد و لذا پارامترهای تعیین شده بر اساس این معیارها، برای هر دو زوج انتخابی در یک نسل، یکسان و ثابت خواهد بود.

در این مقاله، ویژگی فنوتیپ،

$$T_{pheno.} = \frac{|f_{Parent_{1,t}} - f_{Parent_{2,t}}|}{f_{max,t} - f_{min,t}} \quad (۳)$$

که در آن  $f_{Parent_i}$  میزان برزندگی والدین انتخاب شده در نسل جاری است و همچنین ویژگی ژنوتیپ،

$$T_{Geno.} = \frac{\|Parent_1 - Parent_2\|}{\max_{1 \leq i < j \leq n} \|Chrom_i - Chrom_j\|} \quad (۴)$$

که در آن  $\|Parent_1 - Parent_2\|$  فاصله اقلیدسی بین دو والد انتخابی و  $\max_{1 \leq i < j \leq n} \|Chrom_i - Chrom_j\|$  بیشترین فاصله اقلیدسی بین دو کروموزوم در نسل جاری است. معیارهای (۳) و (۴) به ترتیب شباهت و نزدیکی دو والد را تعریف می‌کنند. از آنجا که این معیارها برای هر دو کروموزوم در هر نسل، اختصاصی و منحصر بفرد است، لذا پارامتر  $p_c$ ، نه تنها در نسل‌های مختلف، بلکه در میان هر دو زوج انتخابی در یک نسل نیز متفاوت خواهد بود. این تفاوت برای مسائلی که در محدوده‌های مختلف جستجو، ویژگی‌های مسأله متفاوت است، حائز اهمیت می‌باشد.

### ۳-۱- الگوریتم ژنتیک فازی شده بر اساس معیار شباهت فنوتیپ

در جوامع بشری، همواره احتمال برقراری پیوند بین دو نفر، بسته به سطح خانوادگی آن‌ها متفاوت است. از طرف دیگر، شکل ۲ را در نظر بگیرید. در این شکل، اعضای  $x_2$  و  $x_3$  از نظر برزندگی با یکدیگر متفاوت بوده ولی اعضای  $x_1$  و  $x_2$  برزندگی‌های مشابهی دارند. پیوند به کمک عملگر (۱) بین دو عضو  $x_2$  و  $x_3$ ، فرزندی تولید خواهد کرد که نسبت به  $x_2$  از نقطه بهینه  $x_{opt}$  دورتر خواهد بود. اما در صورتی که زوج  $x_1$  و  $x_2$  با یکدیگر ترکیب شوند، فرزندان به نقطه  $x_{opt}$  نزدیک‌تر خواهند شد. ذکر این نکته ضروری است که برای پیوند  $x_2$  و  $x_3$  نیز باید تا حدودی شانس در نظر گرفت (به عبارت

صورت تصادفی انتخاب شده و یک عدد تصادفی تولید می‌گردد. اگر این عدد کوچک‌تر از پارامتر احتمال بازترکیب ( $p_c$ ) باشد، با اعمال عملگر بازترکیب، دو فرزند برای نسل آتی تولید می‌گردد، در غیر اینصورت این دو والد به صورت مستقیم به نسل بعد انتقال می‌یابند. فرآیند بازترکیب تا تکمیل نسل جدید ادامه می‌یابد. سپس، عملگر جهش بر این نسل اعمال شده و در نهایت، کروموزوم‌های (های) نخبه جایگذاری می‌شود. این فرآیند تا حصول شرایط پایان ادامه می‌یابد. در این مقاله، برای پیوند دو والد، از عملگر بازترکیب خطی استفاده می‌شود بگونه‌ای که،

$$\begin{cases} Child_1 = \alpha \cdot Parent_1 + (1 - \alpha) \cdot Parent_2 \\ Child_2 = (1 - \alpha) \cdot Parent_1 + \alpha \cdot Parent_2 \end{cases} \quad (۱)$$

و در آن  $0 < \alpha < 1$  یک عدد صحیح دلخواه ولی ثابت است. علاوه بر این، شرایط پایان زیر مورد استفاده قرار می‌گیرد:

(الف) حداکثر تولید دویست نسل.

(ب) در دو نسل متوالی، تغییرات در بهترین برزندگی کمتر از  $10^{-6}$  باشد.

باید توجه داشت که در الگوریتم استاندارد، پارامترها همواره و برای همه نسل‌ها و همه کروموزوم‌ها ثابت هستند. همانطور که در بخش مقدمه اشاره شد، ثابت بودن پارامترهایی همچون  $p_m$  و  $p_c$  مناسب نبوده و در اغلب روش‌های نوین، این پارامترها برای هر نسل بصورت جداگانه تعیین می‌شوند. در ادامه، رویکردی مشابه ولی در سطح کروموزوم‌ها پیش گرفته شده است.

### ۳- الگوریتم پیشنهادی

در این بخش، روش‌های نوینی به منظور تعیین پارامتر احتمال بازترکیب  $p_c$  بین دو والد انتخابی در هر نسل از الگوریتم ژنتیک و بر اساس کنترل فازی معرفی می‌گردد. همانطور که در مقدمه بیان شد، محققان غالباً ویژگی‌های فنوتیپ و ژنوتیپ مربوط به پراکندگی یک نسل را برای تعیین پارامتر  $p_c$  و به منظور اعمال عملگر بازترکیب بر روی کروموزوم‌های یک نسل استفاده می‌کنند. تعدادی از این ویژگی‌های عبارتند از [۱۶، ۲۰].

$$\begin{cases} T_1 = \frac{f_{max,t} - f_{min,t}}{f_{max,t}} \\ T_2 = \frac{f_{min,t-1} - 1}{f_{min,t}} \\ T_3 = \frac{(f_{avg,t} - f_{min,t})}{f_{min,t}} \\ T_4 = \frac{HD(C_{f_{max,t}}, C_{f_{min,t}})}{\text{Length of the Chromosome}} \end{cases} \quad (۲)$$

که در آن  $f_{max}$ ،  $f_{min}$  و  $f_{avg}$  به ترتیب نشان‌دهنده کمترین،

است. بدین ترتیب، الگوریتم روش پیشنهادی (که ما آن را FGA می‌نامیم) کاملاً مشابه الگوریتم استاندارد (SGA) در شکل ۱ خواهد بود با این تفاوت که در مرحله اعمال عملگر بازترکیب، به ازای هر دو والد انتخابی، ابتدا معیار (۳) محاسبه شده و سپس بر اساس سیستم فازی (۵) و (۶)، پارامتر  $p_c$  به صورت اختصاصی برای پیوند این دو والد تعیین می‌شود.

به هر حال نکته قابل تأمل، انتخاب مناسب پارامترهای  $c_1$ ،  $c_2$  و  $c_3$  است که به تجربه فرد و شناخت وی از مسأله بهینه‌سازی بستگی دارد. بطور مثال در بخش ۴ ما این مقادیر را به ترتیب 0.5، 0.9 و 0.1 در نظر می‌گیریم. این اعداد به این معنی است که بیشترین احتمال بازترکیب زمانی رخ می‌دهد که دو والد بیشترین مشابهت فنوتیپیک را دارا باشند؛ و کمترین احتمال هنگامی رخ می‌دهد که کمترین شباهت وجود دارد.

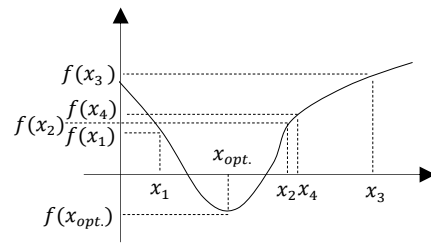
از طرف دیگر این سوال در ذهن تداعی می‌شود که آیا یکسان و ثابت بودن پارامترهای  $c_1$ ،  $c_2$  و  $c_3$  در همه نسل‌ها مناسب است؟ مسلماً با متغیر در نظر گرفتن این سه پارامتر در هر نسل، کنترل-کننده فازی منعطف‌تر شده و بسته به ناحیه‌ای که جستجو در حال انجام است تغییر می‌کند و انتظار می‌رود که عملکرد بهتری از خویش نشان دهد. به همین منظور، در ادامه، راهکاری مبتنی بر الگوریتم هم‌تکاملی پیشنهاد می‌گردد که در آن علاوه بر جمعیت کروموزوم‌های مربوط به مسأله اصلی، جمعیت دیگری برای سه پارامتر  $c_1$ ،  $c_2$  و  $c_3$  شکل می‌یابد و تا رسیدن به شرط پایان، این دو جمعیت هم‌زمان با یکدیگر رشد می‌کنند. ما این روش را CoFGA می‌نامیم که در ادامه جزئیات آن بطور مبسوط و جامع شرح داده می‌شود.

### ۳-۲- روش CoFGA

ایده اصلی، معرفی جمعیت ثانویه‌ای است با تعداد اعضای برابر با جمعیت اصلی که هر کروموزوم در این جمعیت ثانویه نشان‌دهنده یک پایگاه قواعد به سبک (۵) است و این جمعیت به موازات و به صورت هم‌تکاملی با جمعیت مسأله اصلی رشد می‌نماید. در ساختار کروموزوم‌های مربوط به جمعیت ثانویه (که ما آن را جمعیت پایگاه قواعد یا Rule base population می‌نامیم)، هر ژن بیان‌گر یکی از مقادیر  $c_1$  الی  $c_3$  در پایگاه قواعد است (شکل ۴ را ببینید). بطور مثال، اگر کروموزومی در جمعیت پایگاه قواعد به صورت

$c_1$	$c_2$	$c_3$
-------	-------	-------

شکل ۴: ساختار کروموزوم در جمعیت پایگاه قواعد مربوط به روش CoFGA. هر ژن در این کروموزوم بیان‌گر میزان احتمال فازی در یکی از قواعد فازی مربوط به کنترل‌کننده است.



شکل ۲: یک تابع هدف نمونه به همراه چند نقطه تصادفی در آن. پاره‌ای از این نقاط ممکن است خروجی نسبتاً مشابهی داشته باشند ولی از نظر فیزیکی از یکدیگر فاصله دارند. عدم پیوند بین این نقاط در الگوریتم ژنتیک، به معنی عدم جستجوی فضای فیزیکی مابین آنها خواهد بود.

دیگر احتمال بازترکیب دو عنصر که از نظر برازندگی مشابه نیستند نبایستی صفر باشد، چرا که وجود چنین پیوندی سبب می‌گردد تا فضاهای دورتر نیز به منظور یافتن نقاط بهینه احتمالی جستجو شده و پراکندگی کروموزوم‌ها در نسل آتی حفظ گردد. همچنین اگر عضو  $x_3$  در سمت چپ نقطه بهینه باشد، ترکیب آن با  $x_2$  شانس دستیابی به فرزندی در نزدیکی نقطه بهینه را خواهد داشت.

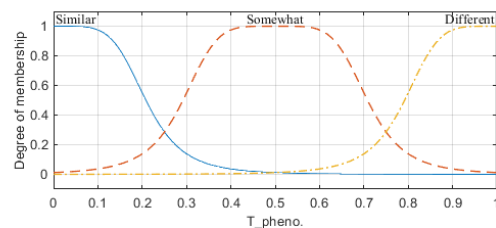
لذا، در این بخش از مقاله، معیار شباهت دو کروموزوم ( $T_{pheno}$ )، مطابق آنچه در شکل ۳ مشاهده می‌شود، به سه زیرمجموعه فازی «Similar»، «Somewhat» و «Different» افزای می‌گردد و کنترل‌کننده فازی تاکاگی-سوگنو مرتبه صفر با پایگاه قواعد زیر برای محاسبه پارامتر  $p_c$  تعریف می‌شود:

$$\begin{cases} IF T_{pheno} \text{ is Similar THEN } p_c = c_1 \\ IF T_{pheno} \text{ is Somewhat THEN } p_c = c_2 \\ IF T_{pheno} \text{ is Different THEN } p_c = c_3 \end{cases} \quad (5)$$

که در آن  $c_1$ ،  $c_2$  و  $c_3$  اعداد ثابت حقیقی و متعلق به بازه  $[0, 1]$  هستند. بدین ترتیب، با مشخص شدن  $T_{pheno}$  به ازای هر دو والد انتخابی، احتمال بازترکیب این دو به صورت زیر محاسبه می‌شود،

$$p_c = \frac{c_1\mu_1 + c_2\mu_2 + c_3\mu_3}{\mu_1 + \mu_2 + \mu_3} \quad (6)$$

بطوری که  $\mu_1$ ،  $\mu_2$  و  $\mu_3$  به ترتیب میزان عضویت ورودی  $T_{pheno}$  در سه مجموعه فازی «Similar»، «Somewhat» و «Different»



شکل ۳: افزای معیار شباهت دو کروموزوم به سه مجموعه فازی. در اینجا، ویژگی فنوتیپ تعریف شده، به سه مجموعه فازی افزای شده است.

تعریف می‌گردد، بایستی برای هر کروموزوم در جمعیت پایگاه قواعد نیز یک تابع برازندگی تعیین نمود که ما در این مقاله آن را به صورت زیر در نظر می‌گیریم،

$$Fit(Rulebase_i) = \frac{f_{avg}^{children} - f_{avg}^{parents}}{|f_{avg}^{parents}|} \quad (A)$$

که در آن  $Rulebase_i$ ، کروموزوم  $i$ ام از جمعیت پایگاه قواعد است. همچنین،  $f_{avg}^{parents}$  میانگین برازندگی والدین انتخابی از جمعیت اصلی، پیش از اعمال عملگر بازترکیب فازی بر روی آنهاست. علاوه بر این،  $f_{avg}^{children}$  میانگین برازندگی فرزندان حاصل از اعمال عملگر بازترکیب فازی (طبق فرمول (5) و (6) و بر اساس ژن‌های موجود در کروموزوم  $Rulebase_i$ ) می‌باشد. در حقیقت، معیار (A) نشان-دهنده میزان بهبود برازندگی ناشی از اعمال عملگر بازترکیب با احتمال  $p_c$  است که این پارامتر احتمال بر اساس کروموزوم جاری از جمعیت پایگاه قواعد و توسط کنترل‌کننده فازی (5) تعیین می‌شود.

لذا، در هر مرحله پس از ارزیابی جمعیت اصلی، تعدادی از اعضای این جمعیت به عنوان والدین نمونه انتخاب می‌شوند. سپس به ازای هر کروموزوم موجود در جمعیت پایگاه قواعد، پارامتر  $p_c$  برای هر جفت والد محاسبه و عملگر بازترکیب به آنها اعمال می‌شود. در پایان و براساس معیار (A) برازندگی کروموزوم مورد نظر از جمعیت پایگاه قواعد بدست می‌آید. حال که برازندگی همه اعضای جمعیت پایگاه قواعد در کنار برازندگی همه اعضای جمعیت اصلی مشخص شد، نه تنها بهترین کروموزوم در جمعیت پایگاه قواعد برای تعیین کنترل-کننده فازی مشخص است، بلکه می‌توان با اعمال عملگر بازترکیب و جهش هر دو جمعیت اصلی و جمعیت پایگاه قواعد را برای گام بعدی رشد داد. فلوجارت روش CoFGA در شکل 5 آورده شده است که به خوبی روند کار را نشان می‌دهد.

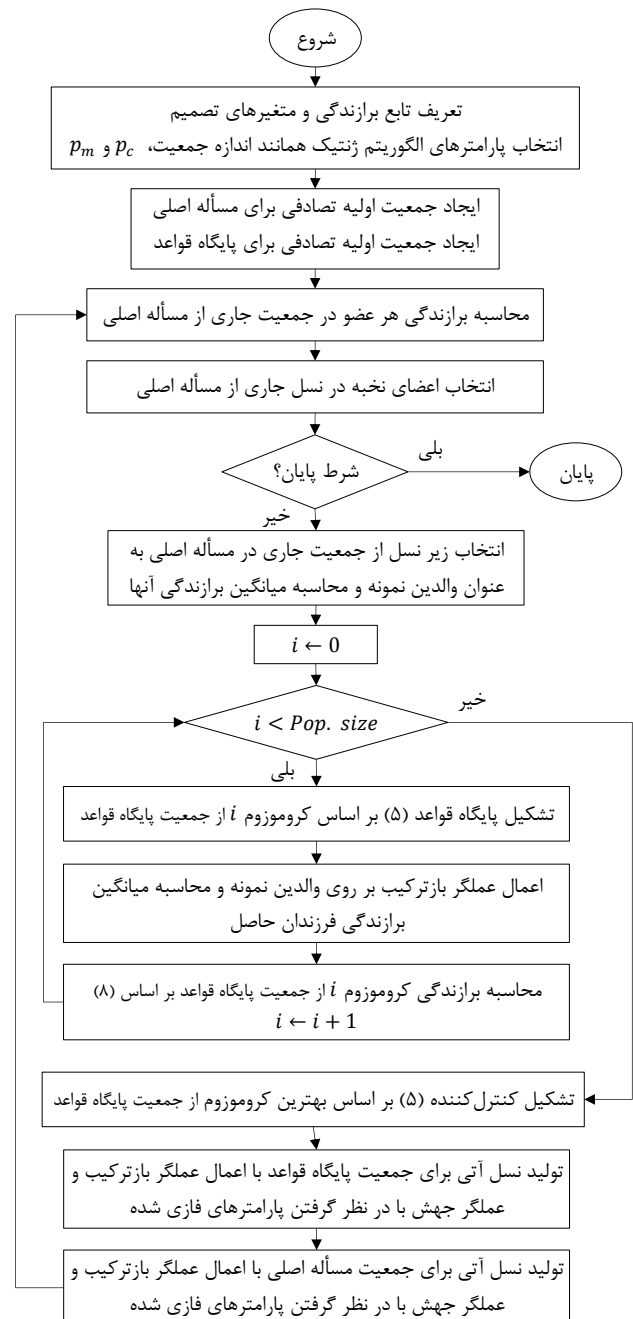
### 3-3- افزودن ویژگی ژنوتیپ و معرفی روش‌های ExtFGA و ExtCoFGA

همانطور که پیش از این نیز بیان شد، در نظر گرفتن ویژگی فنوتیپ به تنهایی کفایت نمی‌کند. برای تجسم فضایی این مدعی، مجدداً به شکل 2 توجه نمایید. همانطور که واضح است زوج  $(x_1, x_2)$  و همچنین زوج  $(x_2, x_4)$  از نظر ویژگی فنوتیپیک بیان شده در رابطه (3) کاملاً مشابه می‌باشند. اما مسلم است که اعمال عملگر بازترکیب خطی بر روی زوج  $(x_1, x_2)$  نه تنها فرزندان نزدیک‌تر به نقطه بهینه تولید می‌کند بلکه پراکندگی بهتری در نسل آتی بوجود خواهد آورد در حالی که فرزندان تولید شده توسط زوج  $(x_2, x_4)$  چندان تفاوتی با والدین خویش نخواهند داشت و در حقیقت پتانسیل جستجوی

$(0.8, 0.4, 0.2)$  باشد، در این صورت پایگاه قواعد مربوطه برای محاسبه پارامتر  $p_c$  به صورت زیر است،

$$\begin{cases} IF T_{pheno.} \text{ is Similar THEN } p_c = 0.8 \\ IF T_{pheno.} \text{ is Somewhat THEN } p_c = 0.4 \\ IF T_{pheno.} \text{ is Different THEN } p_c = 0.2 \end{cases} \quad (V)$$

در الگوریتم‌های هم‌تکاملی، هر جمعیت بر اساس روش انتخاب نخه، عملگرهای بازترکیب و جهش مربوط به خود و به موازات جمعیت همسایه رشد و تکامل می‌یابد. لذا، همانطور که برای هر یک از کروموزوم‌های جمعیت اصلی، برازندگی (به کمک تابع هدف)



شکل 5: فلوجارت روش CoFGA که در آن جمعیت کروموزوم‌های مسئله بهینه‌سازی و پایگاه قواعد بطور همزمان تکامل می‌یابند.

جدول ۱: پایگاه قواعد در روش ExtFGA. با توجه به اینکه هر یک از ویژگی‌های فنوتیپ و ژنوتیپ به سه مجموعه فازی تقسیم شده‌اند، لذا کنترل‌کننده در این حالت شامل ۹ قاعده فازی است.

$T_{Geno.}$	Far	Middle	Near
$T_{pheno.}$ Different	$c_1$	$c_2$	$c_3$
Somewhat	$c_4$	$c_5$	$c_6$
Similar	$c_7$	$c_8$	$c_9$

یکسان اجرا شده و نتایج آن از جمله متوسط زمان اجرا تا رسیدن به جواب بهینه، متوسط بهترین برازندگی ناشی از اعمال هر روش و انحراف استاندارد مربوط به بهترین برازندگی‌ها در اجراهای متفاوت در جدول ۳ آورده می‌شود. لازم به ذکر است که برای اجرای این الگوریتم‌ها از نرم‌افزار متلب، تحت ویندوز با پردازنده Core i5 520M و چهار گیگابایت RAM استفاده شده است. علاوه بر این، روش پیشنهادی اصلی (یعنی روش ExtCoFGA) بر روی مسأله پرواز تعقیب عوارض زمین اعمال می‌گردد.

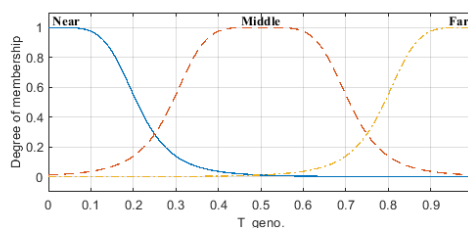
#### ۴-۱- مقایسه روش‌های متفاوت بر روی توابع بهینه‌سازی متفاوت

با توجه به ویژگی‌های روش ارائه شده در این مقاله (که مهمترین آن خودتطبیقی بسته به ناحیه مورد جستجو می‌باشد)، روش‌های پیشنهادی بر روی تعدادی مسأله بهینه‌سازی (مطابق جدول ۲) اعمال و مورد بررسی قرار می‌گیرد. علاوه بر این، روش‌های مطرح شده در مراجع [۱۸] و [۲۲] نیز به همراه نسخه ابتدایی الگوریتم بهینه‌سازی ذرات (PSO) بر روی این مسائل اعمال شده تا بتوان عملکرد روش پیشنهادی را با آنها نیز قیاس نمود.

توابع  $F_1$  الی  $F_3$  همگی تک مودال می‌باشند و دارای یک نقطه بهینه حداقلی هستند. تابع  $F_1$  به شدت مقعر، متقارن و نرم است و یافتن نقطه بهینه حداقلی در آن هیچگونه پیچیدگی خاصی ندارد. در مقابل، تابع روزنبروک ( $F_2$ ) سخت ارزیابی می‌شود چرا که دارای لبه‌های بسیار تیزی است و نقطه بهینه سراسری درون یک دره مسطح طولانی و سهمی‌شکل قرار دارد. این نقطه بهینه، عبارت است از  $(1, 1)$  و به ازای آن، مقدار تابع هدف صفر می‌شود. تابع  $F_3$  نیز

جدول ۲: توابع بهینه‌سازی مورد استفاده برای ارزیابی روش‌ها.

کران‌ها	فرمول تابع	نام
$-5 \leq x_i \leq 5$	$f_1 = x_1^2 + x_2^2$	$F_1$
$-2 \leq x_i \leq 2$	$f_2 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	$F_2$
$-5 \leq x_i \leq 5$	$f_3 = \text{int}(x_1) + \text{int}(x_2)$	$F_3$
$-500 \leq x_i \leq 500$	$f_4 = 800 - \sum_{i=1}^2 x_i \sin(\sqrt{ x_i })$	$F_4$
$-5 \leq x_i \leq 5$	$f_5 = \sum_{i=1}^2 x_i^2 - 10 \cos(2\pi x_i)$	$F_5$



شکل ۶: افزایش معیار نزدیکی دو کروموزوم به سه مجموعه فازی. با توجه به ویژگی ژنوتیپ تعریف شده، نزدیکی دو والد به سه مجموعه فازی تقسیم می‌شود.

جدیدی در نسل بعد اضافه نخواهد شد. لذا، این ایده مطرح می‌گردد که با منظور کردن ویژگی ژنوتیپ  $T_{Geno.}$  مطابق (۴)، روش FGA را توسعه داده و روش ExtFGA را ارائه دهیم که در بخش مقدم از پایگاه قواعد هر دو ویژگی  $T_{Geno.}$  و  $T_{pheno.}$  دیده می‌شود. برای این منظور، ویژگی  $T_{Geno.}$  به سه مجموعه فازی «Near»، «Middle» و «Far» مطابق شکل ۶ تقسیم می‌گردد. پایگاه قواعد کنترل‌کننده فازی تاکاگی-سوگنو مرتبه صفر، در روش ExtFGA، به صورت جدول ۱ می‌باشد که در آن پارامترهای  $c_1$  الی  $c_9$  اعداد ثابتی و حقیقی متعلق به بازه  $(0, 1)$  هستند.

مسلم است که به منظور حفظ پراکندگی جمعیت در نسل‌های متوالی و جلوگیری از همگرایی زودرس، هر چه فاصله اقلیدسی کمتر باشد، میزان پارامتر احتمال بازترکیب کمتر در نظر گرفته می‌شود. بطور مثال در بخش ۴، به منظور شبیه‌سازی روش ExtFGA ما از پایگاه قواعدی با پارامترهای زیر استفاده می‌نماییم.

$$(c_1, \dots, c_9) = (0.5, 0.3, 0.1, 0.7, 0.5, 0.3, 0.9, 0.7, 0.5)$$

در اینجا نیز به منظور بهبود دادن کنترل‌کننده، مشابه آنچه در بخش ۲-۳ آورده شد، از الگوریتم هم‌تکمالی می‌توان بهره گرفت با این تفاوت که ساختار هر کروموزوم در جمعیت مربوط به پایگاه قواعد با توجه به ساختار جدول ۱، بایستی مطابق با ساختار نمایش داده شده در شکل ۷ باشد. ما این روش را که بر اساس الگوریتم هم-تکمالی بوده و در ورودی کنترل‌کننده فازی، هر دو ویژگی فنوتیپ و ژنوتیپ وجود دارد، روش ExtCoFGA می‌نامیم.

#### ۴- نتایج شبیه‌سازی

در این بخش، ابتدا الگوریتم‌های ارائه شده بر روی چندین مسأله بهینه‌سازی اعمال می‌گردد. برای این منظور، به ازای هر یک از مسائل بهینه‌سازی مورد نظر، هر الگوریتم صد مرتبه و با شرایط

$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$
-------	-------	-------	-------	-------	-------	-------	-------	-------

شکل ۷: ساختار کروموزوم در جمعیت پایگاه قواعد مربوط به روش ExtCoFGA



پارامترهای الگوریتم FGA:  $(c_1, c_2, c_3) = (0.9, 0.5, 0.1)$

پارامترهای الگوریتم ExtFGA:

$(c_1, \dots, c_9) = (0.5, 0.3, 0.1, 0.7, 0.5, 0.3, 0.9, 0.7, 0.5)$

همانطور که در جدول ۳ مشاهده می‌شود، در مورد توابع تک مودال ساده، هنگامی که اندازه جمعیت در هر نسل به قدر کافی باشد، عملکرد روش‌های پیشنهادی و الگوریتم ژنتیک استاندارد یکسان بوده و تغییر محسوسی در مقدار حداقل یافته شده وجود ندارد و تقریباً همه روش‌ها در هر ۱۰۰ مرتبه اجرا، جواب بهینه یکسانی را تولید می‌نمایند. اما کاملاً مشهود است که در هنگام محدود بودن اندازه جمعیت، روش‌های پیشنهادی به مراتب بهتر رفتار کرده و نه تنها میانگین جواب‌های بهینه بهتر شده است بلکه پراکندگی این جواب‌ها در یکصد اجرای مختلف از روش استاندارد بهتر است. به هر حال با مقایسه زمان اجرا، در صورتی که اطمینان وجود دارد که مسأله تک مودال و ساده است، پیشنهاد می‌گردد که از الگوریتم ژنتیک استاندارد با اندازه جمعیت مناسب استفاده شود.

اما در مورد توابع تک مودال پیچیده و به خصوص توابع چند مودال، بطور واضح دیده می‌شود که روش‌های پیشنهادی، نه تنها هنگامی که اندازه جمعیت کم است بلکه در زمان زیاد بودن اندازه جمعیت، بخوبی عمل کرده و مقدار بهینه تابع و پراکندگی این مقدار در

شامل سطوح مسطح و به صورت تابع پله و تکه‌ای پیوسته است. این سطوح مسطح به عنوان یک مانع در فرآیند بهینه‌سازی محسوب می‌شوند چرا که اطلاعات درستی برای جهت حرکت به منظور رسیدن به نقطه بهینه در اختیار قرار نمی‌دهند. علاوه بر این موارد، توابع  $F_4$  و  $F_5$ ، چند مودال بوده و دارای تعداد زیادی نقطه بهینه محلی هستند ولی تنها یک نقطه بهینه سراسری دارند. این دو تابع به ترتیب به تابع Schwefel و تابع Rastrigin معروف می‌باشند. برای جزئیات بیشتر در مورد توابع جدول ۲ به [۳۴] مراجعه نمایید.

در این مقاله، به منظور یافتن نقطه بهینه مربوط به توابع  $F_1$  الی  $F_5$ ، روش‌های معرفی شده در بخش ۳ اجرا شده و نتایج آن با الگوریتم ژنتیک استاندارد مقایسه می‌شود. این نتایج در جدول ۳ آورده شده است. پارامترهای مورد استفاده در این شبیه‌سازی عبارتند از،

اندازه جمعیت =  $(10, 50, 100)$

پارامتر احتمال جهش:  $P_m = 0.1$

روش انتخاب کروموزوم‌ها: چرخ رولت

نوع باز ترکیب: خطی

نوع جهش: غیر یکنواخت

پارامتر احتمال باز ترکیب در الگوریتم ژنتیک استاندارد:  $P_c = 0.9$

جدول ۳: مقایسه عملکرد نسخه‌های مختلف الگوریتم ژنتیک با یکدیگر و الگوریتم PSO.

تابع هدف	اندازه جمعیت	میانگین (میانگین)	SGA	FGA	CoFGA	ExtFGA	ExCoFGA	مرجع [۸]	مرجع [۲۱]	PSO
$F_1$	10	زمان اجرا	0.03	0.08	0.50	0.13	0.86	0.09	0.26	0.003
		نسل همگرایی	55.8	69.1	79.70	79.80	90.50	64.4	57.9	54.18
		$f_{min}$	0.024	0.004	0.000	0.001	0.000	0.000	0.003	0.033
		std.dev.	0.054	0.009	0.000	0.003	0.000	0.000	0.001	0.152
	50	زمان اجرا	0.04	0.10	3.32	0.23	3.96	0.16	0.47	0.004
		نسل همگرایی	27.5	25.9	42.4	30.5	29.6	33.3	31.5	32.2
		$f_{min}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		std.dev.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	100	زمان اجرا	0.06	0.18	8.65	0.35	14.08	0.20	0.89	0.007
		نسل همگرایی	24.7	24.2	30.7	27.9	28.6	30.1	29.5	32.3
		$f_{min}$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
		std.dev.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$F_2$	10	زمان اجرا	0.07	0.11	0.62	0.18	1.10	0.11	0.30	0.003
		نسل همگرایی	98.2	73.6	100.3	97.1	115.9	90.6	76.8	60.5
		$f_{min}$	1.058	0.633	0.456	0.705	0.229	0.388	0.238	0.982
		std.dev.	1.383	1.024	0.813	1.049	0.265	0.800	0.202	1.437
	50	زمان اجرا	0.09	0.33	11.95	0.65	17.39	0.27	0.69	0.008
		نسل همگرایی	69.9	75.7	164.7	111.1	136.8	104.4	98.4	68.8
		$f_{min}$	0.047	0.004	0.002	0.040	0.012	0.050	0.018	0.071
		std.dev.	0.057	0.008	0.005	0.060	0.017	0.051	0.019	0.468
	100	زمان اجرا	0.07	0.23	34.98	0.44	49.30	0.36	2.89	0.009
		نسل همگرایی	31.4	33.2	129.9	36.4	103.6	43.1	71.6	46.6
		$f_{min}$	0.021	0.015	0.000	0.000	0.000	0.000	0.000	0.003
		std.dev.	0.034	0.031	0.000	0.000	0.000	0.000	0.000	0.030

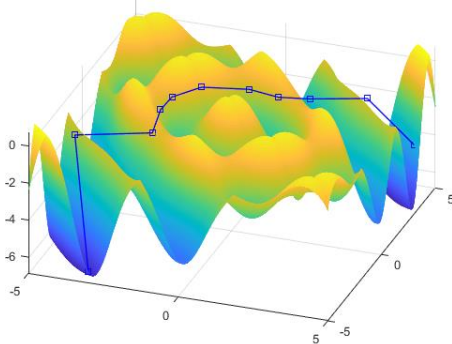
ادامه جدول ۳: مقایسه عملکرد نسخه‌های مختلف الگوریتم ژنتیک با یکدیگر و الگوریتم PSO.

تایم هدف	اندازه جمعیت	میانگین (میانگین)	SGA	FGA	CoFGA	ExtFGA	ExtCoFGA	مرحله [۸]	مرحله [۲۱]	PSO
F <sub>3</sub>	10	زمان اجرا	0.02	0.03	0.11	0.04	0.25	0.03	0.07	0.003
		نسل همگرایی	13.2	16.8	13.1	13.3	16.9	13.0	13.8	49.7
		$f_{min}$	-8.30	-8.50	-8.60	-8.90	-9.00	-8.80	-8.86	-4.79
		std. dev.	2.406	1.581	1.955	1.729	1.633	1.802	1.012	10.67
	50	زمان اجرا	0.02	0.05	0.93	0.08	1.69	0.06	0.93	0.006
		نسل همگرایی	11.0	11.0	11.7	11.9	12.0	11.9	11.9	58.3
		$f_{min}$	-10.0	-10.0	-10.0	-10.0	-10.0	-10.0	-10.0	-5.76
		std. dev.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	14.86
	100	زمان اجرا	0.03	0.08	3.32	0.14	5.66	0.12	2.00	0.014
		نسل همگرایی	10.5	10.9	10.6	11.0	10.6	10.7	10.8	62.9
		$f_{min}$	-10.0	-10.0	-10.0	-10.0	-10.0	-10.0	-10.0	-7.80
		std. dev.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	19.22
F <sub>4</sub>	10	زمان اجرا	0.06	0.09	0.59	0.23	1.27	0.20	0.18	0.004
		نسل همگرایی	87.0	68.2	80.7	107.7	121.3	100.1	89.6	81.5
		$f_{min}$	-731	-659	-696	-721	-745	-701	-712	-680
		std. dev.	104.1	101.6	92.1	94.5	87.8	93.1	96.0	281.8
	50	زمان اجرا	0.06	0.24	5.85	0.37	9.96	0.32	1.73	0.008
		نسل همگرایی	41.9	53.4	76.7	60.2	76.8	60.0	60.9	69.0
		$f_{min}$	-789	-802	-805	-804	-814	-802	-808	-791
		std. dev.	60.3	61.2	57.1	72.9	49.9	77.1	51.4	327.2
	100	زمان اجرا	0.08	0.28	21.11	0.51	27.55	0.49	3.18	0.013
		نسل همگرایی	32.5	39.6	76.2	41.1	57.0	43.8	42.1	57.8
		$f_{min}$	-809	-837	-842	-831	-846	-829	-841	-819
		std. dev.	48.9	57.2	37.5	48.9	37.5	50.4	30.4	365.9
F <sub>5</sub>	10	زمان اجرا	0.04	0.09	0.42	0.16	0.89	0.12	0.11	0.003
		نسل همگرایی	73.1	83.7	66.9	87.1	88.5	75.1	70.8	46.6
		$f_{min}$	-16.8	-17.4	-17.5	-17.6	-16.8	-17.0	-17.1	-14.1
		std. dev.	2.99	2.19	2.08	2.52	1.67	1.92	1.01	4.56
	50	زمان اجرا	0.06	0.20	6.29	0.30	7.37	0.22	1.01	0.004
		نسل همگرایی	43.3	48.5	83.2	47.7	56.9	49.9	53.6	32.0
		$f_{min}$	-19.5	-19.2	-19.9	-19.0	-19.7	-19.0	-19.3	-17.7
		std. dev.	0.63	0.52	0.31	0.44	0.47	0.30	0.28	2.15
	100	زمان اجرا	0.09	0.26	19.04	0.46	25.96	0.38	2.09	0.007
		نسل همگرایی	37.8	37.2	70.5	39.1	54.9	40.7	50.5	31.7
		$f_{min}$	-19.4	-19.7	-19.8	-20.0	-20.0	-20.0	-20.0	-18.4
		std. dev.	1.26	0.48	0.42	0.12	0.00	0.10	0.07	1.67

روش‌های FGA و CoFGA تولید می‌نمایند. لذا، در مجموع پیشنهاد می‌گردد که در مواردی که دیدی نسبت به ساختار مسأله بهینه-سازی نداشته و یا ساختار آن پیچیده و چند مودال است از روش ExtCoFGA استفاده گردد که نه تنها پاسخ‌های مطلوبی را نتیجه می‌دهد بلکه پارامترهای کنترلی آن نیز به صورت هم‌تکاملی تعیین می‌گردد.

از طرف دیگر، با مقایسه با الگوریتم PSO مشاهده می‌شود که سرعت روش پیشنهادی به صورت محسوسی نسبت به PSO بالاتر بوده ولی در مقایسه با آن در تعداد عامل‌های مساوی، رفتار بهتری در یافتن نقطه بهینه دارد. علت این رفتار را می‌توان در جستجوی هوشمندانه همه فضای جستجو دانست. این در حالی است که با محدود بودن

اجراهای مختلف را به طور محسوسی بهبود می‌دهند. از طرف دیگر، کاملاً مشهود است که روش‌های پیشنهادی به دلیل ساختار مورد استفاده دیرتر همگرایی را تجربه می‌کنند که این به معنی حفظ پراکندگی جمعیت تا رسیدن به پاسخ مطلوب می‌باشد. در مقایسه بین روش‌های فازی و روش‌های فازی-هم‌تکاملی در اغلب موارد مشاهده می‌گردد که به دلیل تطبیق پارامترها در طول روند تکامل، روش‌های CoFGA و ExtCoFGA به مراتب رفتار بهتری نسبت به روش‌های FGA و ExtFGA دارند. همچنین روش‌های ExtCoFGA و ExtFGA علی‌رغم اینکه روند طولانی‌تری را به دلیل حفظ پراکندگی جمعیت (با لحاظ کردن هر دو ویژگی فنوتیپ و ژنوتیپ) دارد ولی مجموعه جواب‌هایی به مراتب مناسب‌تر نسبت به



شکل ۹: مسیر استخراج شده در مسأله پرواز تعقیب عوارض زمین بوسیله ExtCoFGA.

ارتفاع موشک دارای محدودیت و کران است، کروموزوم شکل ۸ برای تعیین مختصات و ارتفاع موشک در  $n$  نقطه میانی پیشنهاد می‌شود. نقاط مطلوب، نقاطی هستند که ضمن بهینه‌کردن یک تابع هدف، قیود مسأله را نیز برآورده نماید. در اینجا، از تابع هدف و قیود مطرح شده در [۳۶] بهره گرفته می‌شود. این تابع هدف،

$$f = \sum l_i^2 + h_i^2 \quad (10)$$

است که  $l_i$  طول مسیر بین هر دو نقطه میانی و  $h_i$  متوسط ارتفاع از سطح زمین در طول قطعه نام از مسیر می‌باشد. در حقیقت در نظر گرفتن این دو پارامتر، تضمین‌کننده حداقل زمان رسیدن به هدف و مخفی ماندن حداکثری از دید رادارهاست. البته هر موشک محدودیت‌هایی دارد که پیچیدگی مسأله را بیشتر می‌نماید. در اینجا ما قیودی همچون حداکثر زاویه اوج/افروید یا حداکثر زاویه چرخش به راست یا چپ را  $60^\circ$  درجه در نظر می‌گیریم. همچنین فرض می‌کنیم که موشک باید در طول مسیر نسبت به سطح زمین حداقل ارتفاع  $0.1$  واحد و در نقاط میانی حداکثر ارتفاع  $1$  واحد داشته باشد و هیچ‌گاه نباید از محدوده نقشه خارج گردد. علاوه بر این، فاصله بین دو نقطه میانی بایستی از  $1$  واحد بیشتر و حداکثر طول مسیر از  $40$  واحد کمتر باشد. شکل ۹، نتیجه اجرای روش ExtCoFGA را نشان می‌دهد که بیان‌گر عملکرد مطلوب روش پیشنهادی در این مسأله چند مودال می‌باشد. مقدار تابع هدف بدست آمده پس از  $200$  نسل برابر  $77/246$  است. همچنین، نتیجه اجرای الگوریتم ژنتیک ساده، روش مرجع [۲۲] و الگوریتم PSO بر روی مسأله بهینه‌سازی پرواز تعقیب عوارض زمین خلاصه شده است.

جدول ۴: مقایسه روش‌های مختلف از نظر بهینگی تابع هدف پرواز تعقیب عوارض زمین که در رابطه (۱۰) تعریف شده است.

ExtCoFGA	مرجع [22]	PSO	SGA	
77.246	85.930	103.414	90.734	$f_{min}$

تعداد عامل‌ها در الگوریتم PSO خطر گرفتاری در بهینه‌های محلی کاملاً وجود دارد. شاهد این مدعی، انحراف استاندارد بسیار زیاد مربوط به جواب بهینه در مورد مسائل بهینه‌سازی چند مودال می‌باشد.

در پایان، لازم به ذکر است که اگرچه این مقاله تنها بر روی فازی کردن پارامتر  $p_c$  تمرکز دارد ولی به راحتی می‌توان همین ایده را برای سایر پارامترها از جمله پارامتر  $p_m$  نیز توسعه داد. در ادامه، نتیجه اعمال روش ExtCoFGA بر روی مسأله پرواز تعقیب عوارض زمین که یک مسأله چند مودال محسوب می‌گردد، آورده می‌شود.

#### ۴-۲- مسأله پرواز تعقیب عوارض زمین

مسأله هدایت و کنترل به کمک تعقیب عوارض زمین یکی از زمینه‌های مورد علاقه تحقیقاتی در رابطه با روش‌های ناوبری و کمک ناوبری در دهه‌های اخیر بوده است. امروزه اهمیت مانور تعقیب عوارض و یا اجتناب از موانع، توسط موشک‌های کروز و هواپیماهای بدون سرنشین، از جنبه‌های مهم و شناخته شده در عملیات‌های نظامی است. موضوع اصلی در این دسته مانورها، تنظیم ارتفاع ثابت کمینه تا سطح زمین و طی کردن سریع مسیر مشخص شده به منظور کاهش احتمال شناسایی و تعقیب است. در اولین گام برای این مسأله، بایستی مسیر بهینه یا زیربهینه حرکت موشک طراحی گردد بگونه‌ای که محدودیت‌های عملیاتی نیز در نظر گرفته شود. اگر چه در ادامه بطور مختصری چنین کاربردی در یک مثال آورده شده است ولی برای اطلاعات بیشتر، خواننده می‌تواند به مرجع [۳۵] مراجعه نماید.

فرض کنید که هدف هدایت یک موشک زمین به زمین، از نقطه  $(-5, -3)$  به نقطه  $(4.75, 3.75)$  در میان ناهمواری‌هایی باشد که ارتفاع هر نقطه از سطح آب‌های آزاد، از معادله زیر تبعیت می‌کند،

$$z = \cos(\sqrt{x^2 + y^2}) + 0.1|a|^{\cos(a)} + 0.9|a|^{\sin(a)} \quad (9)$$

و در آن  $a = x + y$  و  $x, y \in [-5, 5]$  است. برای این منظور، بایستی تعدادی نقطه راهنما بین مبدأ و مقصد تعیین گردد و موشک مسیر مستقیم بین این نقاط را بپیماید. هدف ما در این بخش یافتن این نقاط با استفاده از الگوریتم ژنتیک است. همان‌گونه که در [۳۶] اشاره شده است در اولین گام نیاز است تا ساختار کروموزوم تعیین گردد. از آنجا که هدف تعیین مختصات این نقاط بوده و در هر نقطه

$x_1$	$y_1$	$h_1$	$x_2$	$y_2$	$h_2$	...	$x_n$	$y_n$	$h_n$
-------	-------	-------	-------	-------	-------	-----	-------	-------	-------

شکل ۸: ساختار کروموزوم در مسأله تعیین نقاط میانی پرواز مربوط به تعقیب عوارض زمین. ژن‌ها در این کروموزوم بگونه‌ای هستند که طول و عرض نقطه و ارتفاع موشک از سطح زمین را شامل می‌شوند.

## ۵- نتیجه گیری

تعیین مناسب پارامترهای کنترلی در الگوریتم ژنتیک به منظور عملکرد مطلوب آن و جلوگیری از همگرایی زودرس و گرفتار شدن در یک بهینه محلی حائز اهمیت است. در این مقاله روش جدیدی ارائه گردید که با استفاده از آن و با توجه به ویژگی‌های فنوتیپ و ژنوتیپ هر کروموزوم (و نه کل یک نسل)، پارامتر احتمال باز ترکیب برای هر زوج والد انتخابی به صورت مستقل و فازی تعیین می‌شود. علاوه بر این، بخش نتیجه در کنترل کننده فازی از طریق یک فرآیند هم‌تکاملی و به صورت همزمان با متغیرهای تصمیم در مسأله بهینه سازی اصلی رشد داده می‌شود.

نتایج شبیه سازی بر روی مجموعه‌ای از توابع استاندارد نشان می‌دهد که اگرچه زمان همگرایی به دلیل بار محاسباتی بالاتر، طولانی تر است ولی نسبت به الگوریتم ژنتیک استاندارد، یا روش‌هایی که فقط ویژگی‌های ژنوتیپ یا فنوتیپ در یک نسل را به منظور تطبیق استفاده می‌نمایند عملکرد بهتری داشته و در مسائل بهینه سازی آفلاین قابل استفاده می‌باشد. همچنین، نشان داده شده است که تنظیم کنترل کننده بوسیله الگوریتم هم‌تکاملی سبب بالا رفتن عملکرد روش پیشنهادی نسبت به حالتی شده است که در آن پارامترهای کنترل کننده ثابت می‌باشد.

به عنوان گام بعدی در این تحقیق می‌توان به تنظیم سایر پارامترهای الگوریتم ژنتیک توسط یک سیستم فازی خود تنظیم اشاره داشت که در این سیستم فازی نه تنها بخش نتیجه، بلکه بخش مقدم نیز به صورت هم‌تکاملی ساخته می‌شود.

## مراجع

- [6] K. A. De Jong and W. M. Spears, "An analysis of the interacting roles of population size and crossover in genetic algorithms," 1991, pp. 38-47.
- [7] F. Herrera, M. Lozano, and J. L. Verdegay, "Dynamic and heuristic fuzzy connectives-based crossover operators for controlling the diversity and convergence of real-coded genetic algorithms," *Int. J. Intell. Syst.*, vol. 11, no. 12, pp. 1013-1040, Dec. 1998.
- [8] J. Hesser and R. Männer, "Towards an optimal mutation probability for genetic algorithms," in *Parallel Problem Solving from Nature*, Berlin/Heidelberg: Springer-Verlag, pp. 23-32.
- [9] F. Herrera, M. Lozano, and J. L. Verdegay, "Fuzzy connectives based crossover operators to model genetic algorithms population diversity," *Fuzzy Sets Syst.*, vol. 92, no. 1, pp. 21-30, Nov. 1997.
- [10] J. E. Smith, "Self-Adaptation in Evolutionary Algorithms for Combinatorial Optimisation," in *Adaptive and Multilevel Metaheuristics. Studies in Computational Intelligence*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 31-57.
- [11] E. Rodriguez-Tello and J. Torres-Jimenez, "Improving the Performance of a Genetic Algorithm Using a Variable-Reordering Algorithm," 2004, pp. 102-113.
- [12] X. Yan, "An Improved Genetic Algorithm and Its Application in Classification," *Int. J. Comput. Sci. Issues*, vol. 10, no. 1, pp. 337-346, 2013.
- [13] V. R. Kalatjari and M. H. Talebpour, "Optimization of Skeletal Structures Using Improved Genetic Algorithms Based on Proposed Sampling Search Space Idea," *Iran Univ. Sci. Technol.*, vol. 8, no. 3, pp. 415-432, Oct. 2018.
- [14] K. A. De Jong and W. M. Spears, "A formal analysis of the role of multi-point crossover in genetic algorithms," *Ann. Math. Artif. Intell.*, vol. 5, no. 1, pp. 1-26, Mar. 1992.
- [15] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive Operator Selection With Bandits for a Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114-130, Feb. 2014.
- [16] M. Jalali Varnamkhasti, L. S. Lee, M. R. Abu Bakar, and W. J. Leong, "A Genetic Algorithm with Fuzzy Crossover Operator and Probability," *Adv. Oper. Res.*, vol. 2012, pp. 1-16, 2012.
- [17] D. E. Goldberg and K. Sastry, "A Practical Schema Theorem for Genetic Algorithm Design and Tuning," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 2001, pp. 328-335.
- [18] H. Guo, Y. Feng, F. Hao, S. Zhong, and S. Li, "Dynamic Fuzzy Logic Control of Genetic Algorithm Probabilities," *J. Comput.*, vol. 9, no. 1, Jan. 2014.
- [19] H. C. Lau, D. Nakandala, and L. Zhao, "Development of a hybrid fuzzy genetic algorithm model for solving transportation scheduling problem," *J. Inf. Syst. Technol. Manag.*, vol. 12, no. 3, Jan. 2016.
- [20] E. Khmeleva, A. A. Hopgood, L. Tipi, and M. Shahidan, "Fuzzy-Logic Controlled Genetic Algorithm for the Rail-Freight Crew-Scheduling Problem," *KI - Künstliche Intelligenz*, vol. 32, no. 1, pp. 61-75, Feb. 2018.
- [21] L. Yao, Y. L. Jiang, and J. Xiao, "An Improved Fuzzy Adaptive Genetic Algorithm for Function Optimization," *Adv. Mater. Res.*, vol. 403-408, pp. 2598-2601, Nov. 2011.
- [1] H. Sun, Y. Ge, W. Liu, and Z. Liu, "Geometric optimization of two-stage thermoelectric generator using genetic algorithms and thermodynamic analysis," *Energy*, vol. 171, pp. 37-48, Mar. 2019.
- [2] Y. Jiang, P. Wu, J. Zeng, Y. Zhang, Y. Zhang, and S. Wang, "Multi-parameter and multi-objective optimisation of articulated monorail vehicle system dynamics using genetic algorithm," *Veh. Syst. Dyn.*, pp. 1-18, Jan. 2019.
- [3] S. Suri and R. Vijay, "A Bi-objective Genetic Algorithm Optimization of Chaos-DNA Based Hybrid Approach," *J. Intell. Syst.*, vol. 28, no. 2, pp. 333-346, Apr. 2019.
- [4] R. Pereira and L. Aelenei, "Optimization assessment of the energy performance of a BIPV/T-PCM system using Genetic Algorithms," *Renew. Energy*, vol. 137, pp. 157-166, Jul. 2019.
- [5] A. Shrestha and A. Mahmood, "Improving Genetic Algorithm with Fine-Tuned Crossover and Scaled Architecture," *J. Math.*, vol. 2016, pp. 1-10, 2016.

- [30] C.-W. Lee and B.-Y. Lin, "Applications of the Chaotic Quantum Genetic Algorithm with Support Vector Regression in Load Forecasting," *Energies*, vol. 10, no. 11, p. 1832, Nov. 2017.
- [31] R. Lahoz-Beltra, "Quantum Genetic Algorithms for Computer Scientists," *Computers*, vol. 5, no. 4, p. 24, Oct. 2016.
- [32] F. X. Blasco Ferragud, "Control predictivo basado en modelos mediante técnicas de optimización heurística. Aplicación a procesos no lineales y multivariados," *Universitat Politècnica de València, Valencia (Spain)*, 1999.
- [33] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [34] J. G. Digalakis and K. G. Margaritis, "On benchmarking functions for genetic algorithms," *Int. J. Comput. Math.*, vol. 77, no. 4, pp. 481–506, Jan. 2001.
- [35] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and Decision-Making for Autonomous Vehicles," *Annu. Rev. Control. Robot. Auton. Syst.*, vol. 1, no. 1, pp. 187–210, May 2018.
- [36] Changwen Zheng, Lei Li, Fanjiang Xu, Fuchun Sun, and Mingyue Ding, "Evolutionary route planner for unmanned air vehicles," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 609–620, Aug. 2005.
- [22] J. Zhang, H. S.-H. Chung, and W.-L. Lo, "Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 326–335, Jun. 2007.
- [23] M. Yan et al., "Improved adaptive genetic algorithm with sparsity constraint applied to thermal neutron CT reconstruction of two-phase flow," *Meas. Sci. Technol.*, vol. 29, no. 5, p. 055404, May 2018.
- [24] A. Aleti and I. Moser, "A Systematic Literature Review of Adaptive Parameter Control Methods for Evolutionary Algorithms," *ACM Comput. Surv.*, vol. 49, no. 3, pp. 1–35, Oct. 2016.
- [25] F. Herrera and M. Lozano, "Fuzzy Genetic Algorithms: Issues and Models," 1999.
- [26] L. J. Eshelman, R. A. Caruana, and J. D. Schaffer, "Biases in the Crossover Landscape," in *Proceedings of the Third International Conference on Genetic Algorithms*, 1989, pp. 10–19.
- [۲۷] م. قلعه‌نوئی، "بهبود الگوریتم ژنتیک با استفاده از ترکیب منطق فازی و الگوریتم هم‌تکاملی و کاربرد آن در مسأله پرواز تعقیب عوارض زمین،" چهارمین کنفرانس تکنولوژی در مهندسی برق و کامپیوتر، ۱۳۹۸.
- [28] H. Wang, J. Liu, J. Zhi, and C. Fu, "The Improvement of Quantum Genetic Algorithm and Its Application on Function Optimization," *Math. Probl. Eng.*, vol. 2013, pp. 1–10, 2013.
- [29] O. Montiel, Y. Rubio, C. Olvera, and A. Rivera, "Quantum-Inspired Acromyrmex Evolutionary Algorithm," *Sci. Rep.*, vol. 9, no. 1, p. 12181, Dec. 2019.

#### زیر نویس‌ها:

- <sup>5</sup> Phenotype  
<sup>6</sup> Genotype  
<sup>7</sup> Hamming distance  
<sup>8</sup> Binary codification

- <sup>1</sup> Crossover  
<sup>2</sup> Mutation  
<sup>3</sup> Multi-point crossover  
<sup>4</sup> Single-point crossover