

## یک الگوریتم الهام گرفته از طبیعت مبتنی بر نظریه شرطی سازی کلاسیک

محمدنبی امیدوار<sup>۱</sup>، صمد نجاتیان<sup>۲،۳</sup>، حمید پروین<sup>۴،۵</sup>، وحیده رضایی<sup>۳،۶</sup>، میلاد یثربی<sup>۴</sup>

۱- دانشکده مهندسی کامپیوتر، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران

۲- دانشکده مهندسی برق، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران

۳- باشگاه پژوهشگران جوان و نخبگان، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران

۴- دانشکده مهندسی کامپیوتر، واحد نورآباد ممسنی، دانشگاه آزاد اسلامی، نورآباد، ایران

۵- باشگاه پژوهشگران جوان و نخبگان، واحد نورآباد، دانشگاه آزاد اسلامی، نورآباد، ایران

۶- دانشکده ریاضی، واحد یاسوج، دانشگاه آزاد اسلامی، یاسوج، ایران

**چکیده:** الگوریتم‌های فرا ابتکاری الهام گرفته از طبیعت که به تقلید از طبیعت می‌باشند، یک دوره جدید را در حل مسائل بهینه‌سازی باز کردند. در این مقاله با استفاده از رفتار یادگیری شرطی‌سازی کلاسیک پرندگان، ذرات یاد می‌گیرند یک رفتار طبیعی شرطی را در قبال یک محرک غیرشرطی انجام دهند. ذرات در فضای مسئله به چندین دسته تقسیم خواهند شد و هر ذره اگر تنوع دسته خود را در سطح پایینی دید، سعی خواهد کرد به سمت بهترین تجربه شخصی خود حرکت کند و اگر سطح تنوع دسته بالا بود ذره یاد خواهد گرفت که در این شرایط به سمت بهینه عمومی دسته خود متمایل شود. همچنین با استفاده از ایده حساسیت پرندگان نسبت به فضایی که در آن پرواز می‌کنند، سعی شده که ذرات در فضاهای نامناسب با سرعت بیشتری به حرکت درآمده تا ذره از آن فضا دور گردد و بالعکس در فضاهای پرازش سرعت ذرات جهت جستجوی بیشتر، پایین خواهد آمد. در جمعیت‌دهی اولیه نیز با استفاده از رفتار غریزی پرندگان، یک جمعیت‌دهی براساس شایستگی ذرات انجام خواهد شد. روش پیشنهادی در نرم‌افزار متلب پیاده‌سازی شده و نتایج در چندین بخش با روش‌های مختلف مشابه مقایسه و نتایج حاکی از آن بوده که روش پیشنهادی یک الگوریتم قابل‌انکا در حل مسائل بهینه‌سازی ایستا می‌باشد.

**واژه‌های کلیدی:** الگوریتم پرندگان، بهینه‌سازی، هزینه ذرات، معادله سرعت، شرطی‌سازی کلاسیک.

### A Nature-inspired Algorithm based on Classical-conditioning Theory

M.N. Omidvar<sup>1</sup>, PhD Student; S. Nejatian<sup>2,3</sup>, Assistant Professor; H. Parvin<sup>4,5</sup>, Assistant Professor; V. Rezaie<sup>3,6</sup>, Assistant Professor; M. Yasrebi<sup>4</sup>, Lecturer

1- Department of Computer Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran

2- Department of Electrical Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran, Email: samad.nej.2007@gmail.com

3- Young Researchers and Elite Clubs, Yasooj Branch, Islamic Azad University, Yasooj, Iran

4- Department of Computer Engineering, Noorabad Mamasani Branch, Islamic Azad University, Noorabad Mamasani, Fars, Iran, Email: parvin@iust.ac.ir

5- Young Researchers and Elite Clubs, Noorabad Mamasani Branch, Islamic Azad University, Noorabad Mamasani, Fars, Iran

6- Department of Mathematic, Yasooj Branch, Islamic Azad University, Yasooj, Iran, Email: vahidehrezzaie@gmail.com

**Abstract:** Nature-inspired algorithms are the imitation of nature opened a new era in calculations for solving optimization problems.

In this thesis, we will provide an optimization algorithm inspired by nature using the instinctive behavior of birds. In this thesis, particles learn to have a conditional normal behavior towards an unconditioned stimulus using the classical conditioning learning behavior of birds. Particles will be divided into multiple categories in the problem space. If any particle had a low-level category, it will try to move towards its best personal experience. If any particle had a high-level category, it will learn to move towards the global optimum in its category. Using the idea of birds' sensitivity towards the environment, in which birds are flying, we tried to move particles in incompetent spaces more quickly so that the particle goes far away from that space, and vice versa, we will bring down the particles' speed in valuable spaces to search for more. We selected a population based on the particles' merit in the initial population selection using the instinctive behavior of birds. The proposed method was implemented in MATLAB software, and the results have been compared in several different ways. The results showed that the proposed method is a reliable algorithm to solve the static problems.

**Keywords:** PSO Algorithm, optimization, particles cost, velocity equation, classical conditioning.

تاریخ ارسال مقاله: ۱۳۹۶/۰۷/۲۳

تاریخ اصلاح مقاله: ۱۳۹۶/۱۱/۲۹

تاریخ پذیرش مقاله: ۱۳۹۷/۰۴/۰۶

نام نویسنده مسئول: وحیده رضایی

نشانی نویسنده مسئول: ایران - نورآباد ممسنی فارس - دانشگاه آزاد اسلامی - دانشکده مهندسی برق و کامپیوتر.

## ۱- مقدمه

است. این الگوریتم الگوریتمی بهبودیافته از الگوریتم توده ذرات است که با اعمال ایده‌هایی از رفتار غریزی پرندگان پایه‌ریزی شده است. یکی از اشکالات الگوریتم توده ذرات جمعیت‌دهی تصادفی اولیه می‌باشد، همچنین الگوریتم توده ذرات در حرکت ذرات در فضا توجهی به کیفیت فضای مسئله ندارد. اگر فضایی که در آن ذرات هزینه‌های مناسبی ارائه می‌دهد مورد توجه قرار گیرد و آن فضاها بیشتر جستجو شوند به الگوریتم برای رسیدن به جواب بهینه کمک می‌کند. همچنین الگوریتم توده ذرات استاندارد معادله سرعت ذرات را نیز با این کیفیت تطبیق نمی‌دهد. الگوریتم پرندگان همچنین با انتخاب نقطه‌ای بین بهینه محلی و عمومی ذرات باعث می‌شود ذرات تا رسیدن به جواب بهینه زمان زیادی را سپری کنند. در این مقاله با پوشش همه این نقاط و آن هم به کمک یک رفتار غریزی پرندگان که جان‌مایه آن رفتار شرطی شدن کلاسیک می‌باشد الگوریتمی نو ارائه خواهد شد.

در این مقاله الگوریتمی مبتنی بر الگوریتم توده ذرات یا دسته پرندگان ارائه شده است که در مقایسه با روش‌های گذشته این الگوریتم دارای ویژگی‌هایی نظیر مرحله جمعیت‌دهی مبتنی بر نخگی، در نظر گرفتن ویژگی‌های کیفی فضای مسئله، استفاده از یک ایده مبتنی بر نظریه ذاتی پرندگان برای تعریف نحوه حرکت ذرات، معادله سرعت هدفمندتر می‌باشد.

چهارچوب این مقاله به این شرح است که ابتدا در بخش ۱ شرحی از بهینه‌سازی و الگوریتم‌های بهینه‌سازی آمده است، در بخش ۲ کارهای گذشته دیده می‌شود، در بخش ۳ روش پیشنهادی تشریح شده است، در بخش ۴ نتایج شبیه‌سازی نشان داده شده است و در بخش ۵ به نتیجه‌گیری پرداخته شده است.

## ۲- کارهای مرتبط

الگوریتم ژنتیک یک الگوریتم بهینه‌سازی تکاملی تصادفی با یک پتانسیل جستجوی عمومی می‌باشد که در سال ۱۹۷۵ توسط هالند پیشنهاد شد. الگوریتم ژنتیک در میان یکی از موفق‌ترین الگوریتم‌های تکاملی الهام گرفته از طبیعت قرار گرفت. الگوریتم توسط مقداردهی اولیه جمعیت راه‌حل (کروموزوم) آغاز می‌شود و در ادامه با طی مراحل چندگانه به یافتن جواب بهینه می‌پردازد [۵]. در پژوهشی یک الگوریتم بهینه‌سازی ذرات هیبرید جدید و الگوریتم ژنتیک پیشنهاد شده است تا یک مدل ساده از عملکرد انرژی مولکول را به حداقل برساند. الگوریتم پیشنهادی بهینه‌سازی ذرات ترکیبی و الگوریتم ژنتیکی (HPSOGA) نامیده می‌شود. الگوریتم HPSOGA براساس سه مکانیسم می‌باشد. اولین مکانیسم استفاده از بهینه‌سازی ذره برای تعادل بین اکتشاف و فرآیند بهره‌برداری در الگوریتم پیشنهادی است. مکانیسم دوم فرآیند کاهش ابعاد و فرآیند پراکندگی جمعیت است با تقسیم جمعیت به زیرجمعیت و استفاده از اپراتور متقاطع ریاضی در هر زیرجمعیت به منظور افزایش تنوع جستجو در الگوریتم می‌باشد. آخرین مکانیسم برای جلوگیری از همگرایی زودرس و

بهینه‌سازی یک ابزار ریاضی برای یافتن پاسخ بسیاری از پرسش‌ها در خصوص چگونگی راه‌حل مسائل مختلف می‌باشد [۱]. الگوریتم‌های بسیاری بر پایه ایده‌هایی متفاوت و مختلف در این حوزه ارائه گردیدند. مثلاً در پژوهشی یک الگوریتم بهینه‌سازی ذرات بهبودیافته با استفاده از استراتژی عقاب (ESPSO) برای حل مشکل بهینه‌سازی توان راکتیو برای کاهش تلفات قدرت پیشنهاد شده است. تمام شبیه‌سازی‌ها و تجزیه و تحلیل عددی بر روی سیستم قدرت IEEE 30-bus system power و یک زیرسیستم توزیع قدرت واقعی انجام شده است. علاوه بر این، روش پیشنهادی بر روی برخی از توابع معیار تست شده است [۲]. در مقابل برخی از مسائل دارای جواب‌های بیشینه یا کمینه متعددی هستند که به نام نقاط بهینه یا اکستریم شناخته می‌شوند، بهترین اثر هنری، زیباترین منظره و گوش‌نوازترین قطعه موسیقی از مثال‌هایی هستند که می‌توان برای این‌گونه مسائل بیان کرد [۳]. الگوریتم پرندگان به‌عنوان یکی از پرکاربردترین الگوریتم‌های الهام گرفته از طبیعت می‌باشد که دارای سادگی مکانیزم پیاده‌سازی می‌باشد. لذا این الگوریتم دارای نواقصی نظیر جمعیت‌دهی تصادفی، معادله سرعت ناقص، عدم توجه به فضای مسئله می‌باشد. در این تحقیق این نواقص با ایده‌هایی که در ادامه شرح داده خواهد شد رفع شده و یک الگوریتم الهام گرفته از طبیعت معرفی شده است که در عین سادگی پیاده‌سازی، دارای مکانیزم‌های مناسبی می‌باشد. از رفتار دسته‌جمعی پرندگان هنگام پرواز و رفتارهای غریزی آن‌ها بهره جستجو و یک الگوریتم بهینه‌سازی الهام گرفته از طبیعت که از لحاظ مکانیزم به الگوریتم توده ذرات نزدیک می‌باشد ارائه خواهد شد. انتظار می‌رود که رفتارهای یادگیری پرندگان در قالب یک حرکت اجتماعی ما را به سمت یک مکانیزم برای حل مسائل غیرقطعی پیچیده سوق دهد. ایده اصلی این تحقیق از رفتار یادگیری شرطی‌سازی کلاسیک پرندگان الهام گرفته شده است. شرطی‌سازی کلاسیک یک نوع یادگیری همخوان یا تداعی است. در شرطی‌شدن کلاسیک جانور یاد می‌گیرد که بین یک محرک بی‌اثر، با یک پاداش یا تنبیه کردن ارتباط برقرار کند. اصطلاح شرطی‌شدن کلاسیک نخستین بار توسط ایوان پاولف، فیزیولوژیست روسی، پیشنهاد شد [۴]. مشاهده ابتدایی که پاولف را به سمت این نظریه سوق داد، این بود که بزاق دهان سگ‌ها صرفاً با دیدن کسی که قبلاً چندبار به آن‌ها غذا داده است، ترشح می‌شود؛ حتی اگر شخص اکنون غذایی به همراه نداشته باشد. از این رفتار استفاده خواهد شد و دسته‌های پرندگان موجود در فضای مسئله به این کار ملزم خواهد شد که اگر در یک فضای مسئله تنوع پراکندگی مکانی پرندگان در حد بالایی باشد، پرندگان موجود در آن دسته به سمت بهینه عمومی دسته و اگر بالعکس باشد به سمت بهینه محلی و یا بهینه عمومی کل دسته‌ها حرکت خواهند کرد. چند ایده دیگر در کنار این رفتار را برای حرکت و سرعت پرندگان و همچنین جمعیت‌دهی اولیه در نظر گرفته شده

تصادفی در سال ۲۰۰۷ ارائه گردید [۳۳]. از دیگر تحقیقات جدید در این حوزه می‌توان به الگوریتم MRPSO [۳۴]، الگوریتم EPSDE [۳۵]، الگوریتم CCABC [۳۶]، الگوریتم کرم شب‌تاب [۳۷]، روش‌های TLBO [۳۸] و FFFA [۳۹] اشاره کرد. اغلب این روش‌ها مکانیزم‌هایی ارائه داده‌اند تا بتوانند الگوریتم‌های استاندارد پایه گذشته را با ایده‌هایی نو بهبود دهند. استفاده از این الگوریتم در بخش مقایسه، به‌منظور سنجش دقیق‌تر روش پیشنهادی بوده است. در این مقاله برای صحت نتایج الگوریتم‌ها از تست فریدمن [۴۰] و تست تی [۴۱] استفاده شده است.

در مقاله‌ای روشی برای بهینه‌سازی چندهدفه با جبهه‌های مغلوب‌نشده براساس الگوریتم جستجوی گرانشی معرفی شده که از مفاهیم جبهه‌های مغلوب‌نشده برای تخصیص شایستگی و فاصله ازدحامی راه‌حل‌ها برای حفظ تنوع جمعیت استفاده شده است [۴۲].

در روشی ضمن استفاده از روش متداول تبدیل مسئله مکان‌یابی به مسئله بهینه‌سازی، از الگوریتم نوظهور رقابت استعماری استفاده شده و با معرفی مفهومی جدید به‌نام نماینده، روند تولید جواب‌های جدید و بررسی فضای جستجو هدفمند شده است [۴۳].

در تحقیق دیگر یک الگوریتم ژنتیک آشوب‌گونه مبتنی بر خوشه‌بندی و حافظه برای حل مسائل پویا ارائه شده است. یک سیستم آشوب‌گونه پیش‌بینی دقیق‌تری از آینده نسبت به یک سیستم تصادفی دارد و میزان همگرایی را در الگوریتم افزایش می‌دهد. به‌طورمعمول استفاده از اطلاعات گذشته اجازه می‌دهد الگوریتم به‌سرعت بعد از تغییر محیط به سازگاری در شرایط محیطی جدید برسد، بنابراین ایده موردنظر در این زمینه، استفاده از یک حافظه است که با استراتژی مناسبی اطلاعات مفید گذشته را ذخیره نموده و برای استفاده مجدد آن‌ها را بازبازی می‌نماید. خوشه‌بندی در حافظه و جمعیت اصلی، تنوع را در حین اجرای الگوریتم با تبادل اطلاعات میان خوشه‌های متناظر (خوشه‌ها با برچسب شبیه به‌هم) در حافظه و جمعیت اصلی حفظ می‌نماید. در این روش پیشنهادی دو جنبه نوآوری اساسی پیشنهاد شده است. یکی روش خوشه‌بندی استفاده شده که هم جمعیت اصلی و هم جمعیت حافظه را خوشه‌بندی می‌کند و دیگری راه‌کار مناسبی است که برای به‌روزرسانی حافظه استفاده شده است [۴۴].

### ۳- روش پیشنهادی

به‌طورکلی پرندگان را به دسته‌هایی تقسیم کرده و در هر دسته مکانیزم پیشنهادی را اجرا کرده و در نهایت بهترین جواب تولیدشده در کل دسته‌ها به‌عنوان جواب بهینه تعیین خواهد شد. در مرحله نخست بازه متغیرهای موجود در هر جمعیت تعریف خواهد شد. این کار برای این می‌باشد که متغیرهای هر جمعیت در بازه‌هایی مشخص تقسیم شود و آن‌ها مجاز به حرکت در آن محدوده باشند.

$$x = nVar\ section \quad (1)$$

اجتناب از دامداری در حداقل‌های محلی با استفاده از اپراتور جهش ژنتیکی در کل جمعیت استفاده می‌شود [۶]. بهینه‌سازی ازدحام ذرات (PSO) یک الگوریتم الهام‌گرفته از طبیعت و تصادفی می‌باشد که در سال ۱۹۹۵ پیشنهاد شد [۷]. الگوریتم پرندگان با توجه به مکانیزم منحصر به فرد جستجو، مفهوم ساده، بازده محاسباتی، و پیاده‌سازی آسان به‌طور گسترده در بسیاری از حوزه‌های بهینه‌سازی مهندسی کاربرد دارد. در برخی پژوهش‌ها از این الگوریتم برای محاسبه مسیر بهینه در فضای جستجوی دودویی برای حل مشکل کوله‌پشتی (KP) استفاده شده است [۸-۱۰]. ضمناً برای حل برخی مسائل از PSO ترکیب شده با الگوریتم ژنتیک استفاده شده است [۱۱].

الگوریتم کلونی زنبورعسل مصنوعی یک الگوریتم بهینه‌سازی می‌باشد که از رفتار جستجوگرانه زنبورها برای یافتن غذا در طبیعت الگوبرداری شده است [۱۲]. با اقتباس از روش اصلی ABC الگوریتمی ارائه شد که با تغییر رفتار زنبورهای ناظر برای تبدیل مقادیر پیوسته به باینری انجام گرفت [۱۳]. یک الگوریتم ترکیبی زنبور مصنوعی با رمزگذاری گسسته برای TSP ارائه شد [۱۴]. الگوریتم جستجوی هارمونی<sup>۸</sup> [۱۵] یک الگوریتم فرا ابتکاری است که در روند طبیعی عملکرد موسیقی، یک وضعیت خوب در طول بداهه‌نوازی جاز را جستجو می‌کند. الگوریتم زنبورها<sup>۹</sup> ابتدا روی یک تابع ریاضی معرفی شد [۱۶]. الگوریتم زنبورعسل (BA) برای اولین بار برای کارهای زمان‌بندی مورد استفاده قرار گرفت [۱۷] و سپس برای خوشه‌بندی داده‌های باینری مورد استفاده قرار گرفت [۱۸]. الگوریتم رقابت استعماری<sup>۱۰</sup> به‌عنوان یک روش بهینه‌سازی تکاملی است که از رقابت امپریالیستی الهام گرفته شده است و برگرفته از رفتار امپریالیست در تلاش خود برای غلبه بر مستعمرات است [۱۹]. الگوریتم‌های دیگری نیز در سال‌های اخیر ارائه شده‌اند که اغلب در راستای بهبود الگوریتم‌های معروف بهینه‌سازی بوده‌اند، نظیر الگوریتم PBI [۲۰]، الگوریتم NPSO [۲۱]، الگوریتم CS [۲۲]، الگوریتم DSA [۲۳] و الگوریتم BMO [۲۴]، الگوریتم sinDE [۲۵]، الگوریتم JOA [۲۶]، الگوریتم D-PSO-C که با استفاده از یک متد چندجمعیتی پویا برای بهبود الگوریتم توده ذرات ارائه گردید [۲۷]. در الگوریتمی دیگر با استفاده از ایجاد پارامترهای آشوب الگوریتم جستجوی فاخته بهبود داده شد [۲۸]. در سال ۲۰۱۵ با کمک ایده یادگیری انسان برای پیدا کردن راه‌حل، الگوریتم توده ذرات بهبود داده شد [۲۹]. در سال ۲۰۱۶ با ترکیب الگوریتم‌های مورچگان و ژنتیک الگوریتمی بهینه‌سازی ارائه گردید [۳۰]. در تحقیقی نویسندگان با استفاده از الگوریتم ژنتیک تلاش کردند نزدیک‌ترین راه‌حل بهینه عمومی برای حل مسائل بهینه‌سازی مالتی‌مدال غیرخطی پیدا شود [۳۱]. مجموعه تست CEC 2009 از مجموع توابعی می‌باشد که برای آزمایش الگوریتم‌های بهینه‌سازی ارائه شدند [۳۲]. الگوریتمی دیگر به‌نام تکامل

الگو گرفته شده و در هنگام جمعیت‌دهی اولیه، آن ذراتی که دارای شرایط نامناسبی هستند از فضای مسئله حذف خواهند شد و به جای آن‌ها و به همان تعداد جمعیت تصادفی جایگزین می‌شود. برای این کار ابتدا جمعیت تولیدشده اولیه در یک آرایه براساس هزینه آن‌ها مرتب می‌شود.

$$Cost Array = Sort (Particle.Cost) \quad (8)$$

و از آخر آرایه به تعداد تعیین شده عمل جایگزینی انجام خواهد شد. در این تحقیق به تعداد  $\frac{1}{3}$  از آخر آرایه مرتب شده هزینه، ذرات جایگزین خواهند شد.

$$\left( \frac{1}{3} \times nPopulation \right) \rightarrow unifrnd (Var Max, Var Min, Var Size) \quad (9)$$

هزینه جمعیت جدید محاسبه خواهد شد و برای هر ذره در ارزیابی اول، بهینه محلی مکان ذره خواهد بود.

$$PBest = (Particle.Position) \quad (10)$$

بهینه عمومی نیز بهترین مقدار در کل ذرات می‌باشد. برای توابع ایستای این تحقیق مقدار مینیمم بهترین هزینه خواهد بود.

$$GBest = min (Particle.Cost) \quad (11)$$

بعد از مشخص شدن بهینه عمومی کل ذرات، بهینه عمومی هر دسته نیز جداگانه مشخص خواهد شد. ابتدا به‌ازای هر دسته مقدار هزینه ذرات در یک آرایه قرار داده می‌شود.

$$C = (Particle.Cost) \quad (12)$$

$$(13) T = \frac{(i-1) \times Population Number}{nPopulation Section} + 1$$

در پارامتر  $T$  جمعیت‌های هر دسته مشخص شده و در پارامتر  $C$  قرار داده می‌شود تا در هر دسته به‌صورت مجزا جمعیت موجود در هر دسته با هم مقایسه شوند.

$$C = C(T) \quad (14)$$

حالا در هر دسته مینیمم هزینه به‌عنوان بهینه عمومی آن دسته تعیین می‌شود.

$$GBest = min(C) \quad (15)$$

از ایده حساسیت پرنده‌گان نسبت به محیط استفاده خواهد شد و بعد از آنکه ابعاد زمین برای متغیرها به چند بخش تقسیم شدند، ارزش‌گذاری هر فضا بر این اساس که اگر ذره در فضای پرارزش قرار

در معادله ۱ تعداد دسته تعیین شده در متغیر  $X$  تعریف خواهد شد. در این شبیه‌سازی تعداد  $X$  برابر ۱۰ گرفته شده است، یعنی محدوده بازه متغیرها ۱۰ فرض خواهد شد. حال طول هر محدوده با استفاده از رابطه ۲ محاسبه می‌شود.

$$L = \frac{(Var Max - Var Min)}{nVar section} \quad (2)$$

در این معادله  $var max$  و  $var min$  حد بالا و پایین فضای مسئله و  $nvar section$  نیز تعداد بازه متغیرها می‌باشد. در مرحله بعد جمعیت به چند قسمت یا دسته تقسیم می‌شود.

$$x = nPopulation section \quad (3)$$

در این تحقیق جمعیت‌ها به تعداد ۴ دسته تقسیم شده است. طبق رابطه ۴ پارامتر تعداد دسته‌ها در الگوریتم پیشنهادی مقداردهی می‌شود.

$$nPopulation section = 4 \quad (4)$$

در بخش تعریف پارامترها، یک پارامتر آستانه نیز برای تعیین میزان انحراف معیار دسته‌های پرنده‌گان تعریف خواهد شد. در مرحله تولید جمعیت به‌طور تصادفی و به‌اندازه جمعیت تعیین شده براساس حد بالا، حد پایین و تعداد متغیرها جمعیت اولیه ایجاد می‌شود.

$$Particle.Position = unifrnd (Var Max, Var Min, Var Size) \quad (5)$$

در این معادله  $Var Size$  تعداد جمعیت و  $unifrnd$  نیز به این دلیل به‌کاررفته که خواسته شده که اعداد تصادفی یکتا ایجاد شود و مکان ذرات در فضای جستجو تکراری نباشد. جمعیت تولیدشده در تابع هزینه قرار داده می‌شود و هزینه ذرات به دست خواهد آمد.

$$Particle.Cost = Fitness (Particle.Position) \quad (6)$$

در این تحقیق تابع برازندگی را با نام  $Fitness$  تعریف کرده و در آن معادله تابعی که قرار است به‌وسیله الگوریتم پیشنهادی حل گردد، قرار دارد. همچنین مشخص می‌شود که متغیرهای هر جمعیت در کدام ناحیه از فضا قرار دارند. ابعاد فضای مسئله به بخش‌هایی تقسیم خواهد شد تا فضا براساس ارزش آن ناحیه طبقه‌بندی شود. پس لازم است مشخص باشد که هر متغیر از جمعیت در کدام نواحی قرار دارد.

$$\begin{aligned} & \text{if } VarMin + (j-1) * L \leq Particle(i).Position(j) \\ & \text{and } Particle(i).Position(j) < (VarMin + j * L) \\ & \rightarrow Space(i, j) = j \end{aligned} \quad (7)$$

پرنده‌گانی که انرژی مناسبی داشته باشند و تغذیه خوبی داشته‌اند در هنگام پرواز عملکرد بهتری خواهند داشت، از همین رفتار پرنده‌گان

عمومی حرکت کنند. حال معادله سرعت ذرات بدین صورت تنظیم خواهد شد.

$$velocity(i) = (1 - \frac{spacerate(1, j, spacerate(i, j))}{sum(spacerate(1, j, x))}) \times w$$

$$\times particle.velocity + c1 \times rand(var\ size)$$

$$\times particle(i).PBest - particle(i).position$$

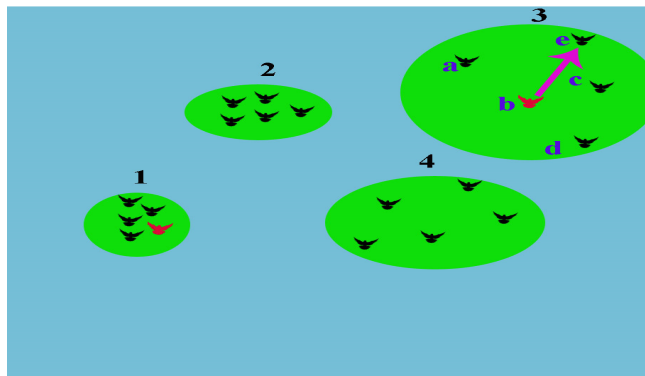
$$+ c2 \times rand(var\ size) \times (particle(i).GBest - particle(i).position)$$

(۱۹)

برای کنترل سرعت حرکت ذرات در فضاهای پرارزش و کم ارزش می باشد. این قسمت معادله سرعت برای ذراتی که در فضایی بالاتر هستند اعداد کمتری تولید می کند. با ضرب این عدد کمتر در وزن اینرسی مقدار سرعت کاهش خواهد یافت. با این مکانیزم این ذرات در آن فضای بالاتر تر با سرعت کمتری حرکت و در آن فضا بیشتر جستجو خواهد داشت.  $w$  وزن اینرسی است که برای تعیین میزان تأثیر سرعت قبلی بر سرعت فعلی می باشد،  $Particle(i).Velocity$  سرعت فعلی ذره،  $c1$  ضریبی است که میزان تمایل ذره به سمت بهینه محلی را مشخص می کند،  $rand(Var\ Size)$  یک عدد تصادفی برای حرکت غیر خطی ذره،  $Particle(i).PBest$  بهینه محلی یا بهترین تجربه شخصی است که بهترین مکانی است که ذره تا الان داشته است.  $Particle(i).Position$  مکان فعلی ذره،  $c2$  ضریبی است که میزان تمایل ذره به سمت بهینه عمومی را مشخص می کند و  $Particle(i).GBest$  بهینه عمومی یا بهترین ذره در بین کل جمعیت پرندگان می باشد. با اعمال این معادله سرعت، ذرات برای حرکت در گام بعدی اگر انحراف استاندارد و تنوع دسته شان بالا باشد به سمت بهینه عمومی حرکت کرده و اگر تنوع دسته پایین باشد و ذرات یک دسته در فضای کمتری به جستجو مشغول باشند، سعی می کند به سمت بهینه محلی یا بهترین تجربه شخصی خود متمایل شود. با مشخص شدن معادله سرعت ذرات، معادله حرکت طبق معادله ۲۰ تعیین خواهد شد.

$$Particle(i).NewPosition = Particle(i).Position + Particle(i).Velocity$$

(۲۰)



شکل ۲: مکانیزم حرکتی ذرات در داخل هر دسته

گرفت و ذره در آن مکان دارای هزینه مناسبی گردید، یک واحد به پارامتر ارزش فضا اضافه شود. هر بار که یک بهینه عمومی در یک زیرفضا قرار گرفت میزان ارزش فضا یک واحد اضافه خواهد شد.

$$if\ GBest\ in\ i \rightarrow Space\ Rate(i) = Space\ Rate(i) + 1$$

(۱۶)

در مرحله بعد میزان انحراف استاندارد هر دسته اندازه گرفته شده است تا بتوان ایده رفتار یادگیری شرطی شدن کلاسیک را پیاده سازی کرد.

$if\ STD \geq Threshold$

$$C1 = 0.9 \times C1 \quad C2 = 1.1 \times C2$$

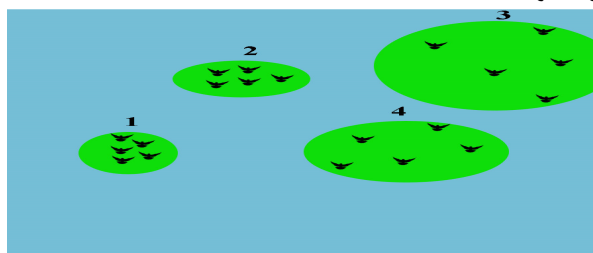
(۱۷)

Else

$$C1 = 1.1 \times C1 \quad C2 = 0.9 \times C2$$

(۱۸)

در این معادله ضریب قبل از بهینه محلی و ضریب قبل از بهینه عمومی در معادله سرعت به کار می روند. طبق معادلات بالا اگر انحراف استاندارد دسته بالا باشد، با قراردادن یک عدد بالاتر از یک برای ضریب قبل از بهینه عمومی و قراردادن یک عدد کمتر از یک قبل از بهینه محلی، سعی خواهد شد که ذره بیشتر به سمت بهینه عمومی سوق داده شود و از بهینه محلی دور گردد. بالعکس اگر تنوع یا انحراف استاندارد دسته پایین باشد با قراردادن ضریب کمتر از یک برای ضریب قبل از بهینه عمومی و ضریب بیشتر از یک برای ضریب قبل از بهینه محلی، ذره از بهینه عمومی دور و به سمت بهینه محلی منتقل شود. این کار به این جهت انجام خواهد شد که اگر تنوع دسته پرندگان زیادتر از حد آستانه تعریف شده باشد، به این دلیل که ذرات در فضای بیشتری جستجو کرده اند رفتن به سمت بهینه عمومی آن سودمندتر خواهد بود.

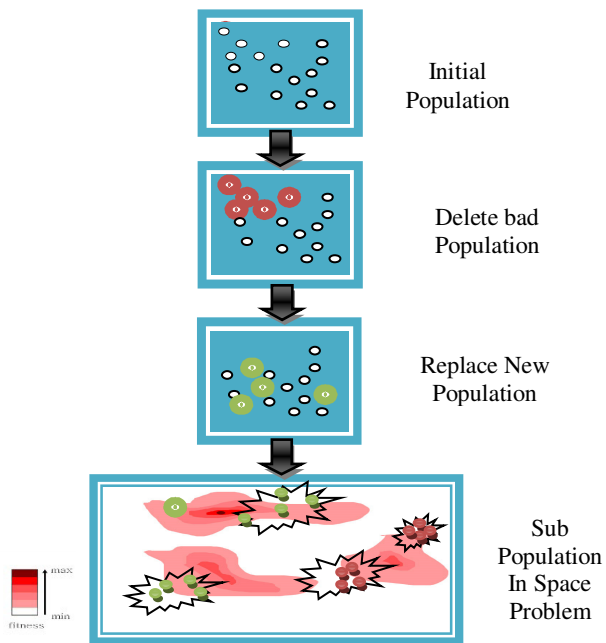


شکل ۱: میزان تنوع دسته ها

در شکل ۱ مشاهده می شود که دسته شماره ۱ دارای کمترین تنوع و دسته ۳ بیشترین تنوع می باشد. در دسته ۱ و ۲ ذرات بیشتر به سمت بهینه محلی حرکت می کنند، در حالی که طبق شکل ۲ ذرات موجود در دسته های ۳ و ۴ طبق مکانیزم تعریف شده موجود در تحقیق بیشتر به سمت بهینه عمومی حرکت خواهند کرد. این ایده در قالب رفتار یادگیری شرطی سازی کلاسیک پرندگان برای ذرات تعریف خواهد شد. بر طبق این رفتار یادگیری، پرندگان یاد می گیرند که هرگاه میزان تنوع دسته خود را بالا حس کردند با میل بیشتری به سمت بهینه

1. جمعیت اولیه را تصادفی تولید کن.
2. جمعیت اولیه را در تابع هزینه قرار بده و هزینه ذرات را محاسبه کن.
3. ذرات را براساس هزینه آن‌ها در یک آرایه مرتب کن.
4. در آرایه مرتب‌شده مرحله ۳ و از آخر تعداد ۱/۳ ذرات را حذف کن و به جای آن‌ها ذرات به‌طور تصادفی جایگزین کن.
5. جمعیت جدید را در تابع هزینه قرار بده و هزینه ذرات را محاسبه کن.
6. ذرات را به چند دسته تقسیم کن.
7. نواحی متغیر هر جمعیت را مشخص کن.
8. برای هر دسته بهینه محلی را مکان اولیه ذرات قرار بده.
9. بهترین بهینه محلی هر دسته را بهینه عمومی دسته قرار بده و ارزش فضای آن ذره را یک واحد اضافه کن.
10. مقادیر پارامترهای  $C1$  و  $C2$  را براساس تنوع و انحراف استاندارد دسته‌ها تنظیم کن.
11. برای هر ذره معادله سرعت را براساس سرعت فضایی که در آن هستند به‌روزرسانی کن.
12. معادله حرکت ذرات را محاسبه کن.
13. جمعیت را به مکان جدید حرکت بده.
14. مقدار هزینه ذرات را برای مکان‌های جدید ذرات محاسبه کن.
15. اگر مکان جدید ذرات از بهینه محلی آن‌ها بهتر بود مکان جدید را به‌عنوان بهینه محلی قرار بده.
16. برای هر دسته بهینه عمومی را پیدا کن.
17. بهترین بهینه عمومی دسته‌ها را به‌عنوان بهترین هزینه انتخاب کن.
18. اگر تکرار جاری کمتر از مقدار تکرار تعیین‌شده است به مرحله ۵ برو در غیر این صورت به مرحله ۱۹ برو.
19. پایان.

شکل ۳: شبه‌کد روش پیشنهادی



شکل ۴: طرح کلی روش پیشنهادی

در این معادله  $Particle(i).NewPosition$  مکان بعدی ذره،  $Particle(i).Position$  مکان فعلی ذره و  $Particle(i).Velocity$  سرعت ذره می‌باشد. بر طبق شکل ۲ اگر فرض گردد ذره  $e$  بهینه عمومی دسته ۳ باشد، آنگاه ذره  $b$  حاضر در دسته ۳ برای حرکت در گام بعد به دلیل تنوع بالای دسته سعی می‌کند معادله سرعت خود را طوری تنظیم کند که به سمت بهینه عمومی دسته حرکت نماید. این کار با تنظیم مقدار اعداد ضرایب  $c1$  و  $c2$  انجام می‌گیرد. در روش پیشنهادی سعی خواهد شد که ذرات در این فضاها با سرعت کمتری به حرکت درآورده شده تا ذرات بتوانند جستجوی بیشتری در این مناطق داشته باشند. قسمت اول معادله سرعت روش پیشنهادی این مکانیزم را برای ما اعمال خواهد کرد.

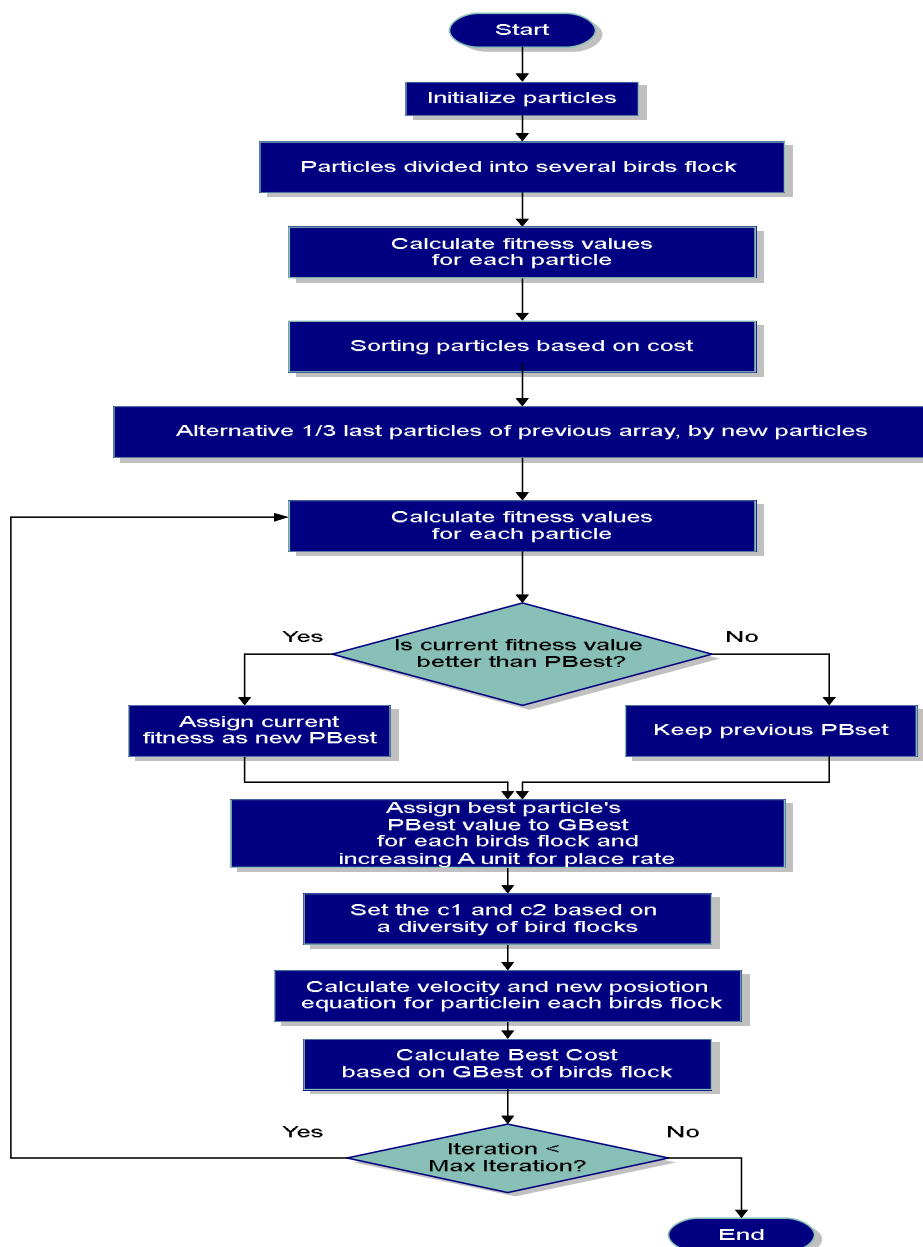
بعد از حرکت ذرات طبق معادله ۲۰ و قرارگرفتن در مکان جدید، دوباره در تابع هزینه قرار گرفته و هزینه ذرات به دست خواهد آمد. دوباره بهینه‌های عمومی و محلی هر دسته مشخص خواهد شد. بهینه عمومی طبق معادلات گفته‌شده قبلی خواهد بود و یافتن بهینه محلی یا بهترین تجربه شخصی هر ذره طبق معادله ۲۱ به دست می‌آید.

$$if \text{Cost} (Particle.NewPosition) < Pbest \rightarrow Pbest = Particle.NewPosition \quad (21)$$

در این معادله  $Cost(Particle(i).NewPosition)$  هزینه مکان جدید پرنده،  $Pbest(i)$  بهینه محلی پرنده می‌باشد. پیش‌تر گفته شد که در مرحله اول بهینه محلی هر پرنده مکان اولیه پرنده می‌باشد، و از تکرار دوم به بعد طبق معادله ۲۲ بهینه محلی هر ذره به‌روزرسانی می‌شود. در پایان تعداد ارزیابی‌های تعیین‌شده با مقایسه بهینه‌های عمومی کل دسته‌ها بهترین هزینه به دست می‌آید.

$$Best \text{ Cost} = \min (Gbest (Population \text{ Section})) \quad (22)$$

در شکل ۳ شبه‌کد روش پیشنهادی آمده است که روش پیشنهادی را به صورت مرحله‌ای تشریح کرده است. در شکل مشخص می‌باشد که تا مرحله ۱۹ مراحل مربوط به یک تکرار است که باید انجام گردد. در شکل ۴ طرح کلی روش پیشنهادی آمده است. جمعیت اولیه تصادفی تولید شده، سپس یک سوم جمعیت بد از جمعیت حذف خواهد شد. در مرحله بعد به اندازه جمعیت حذف‌شده جمعیتی تصادفی جایگزین خواهد شد. سپس جمعیت به تعدادی دسته ذرات تقسیم می‌شود. در شکل نمونه ذرات حاضر در دسته قرمز به دلیل تنوع پایین بیشتر به سمت بهترین تجربه شخصی خود حرکت خواهند کرد. این حرکت که با یک حرکت آزادانه و تصادفی نیز ترکیب خواهد شد شاید همین دسته را به یک دسته با تنوع بالاتر در ارزیابی‌های بعدی تبدیل کند. ذرات دسته سبز رنگ به دلیل تنوع بالای دسته بیشتر به بهینه عمومی دسته خود همگرا خواهند شد. در شکل ۵ نیز روندنمای روش پیشنهادی آمده است که نحوه انجام مراحل روش را به صورت مختصر و نمادین نشان داده است.



شکل ۵: روندنمای روش پیشنهادی

#### ۴- شبیه‌سازی و نتایج

ژنتیک [۵]، تکامل تصادفی [۳۳]، توده ذرات [۷]، الگوریتم زنبورها [۱۶]، الگوریتم PBI [۲۰]، الگوریتم NPSO [۲۱]، الگوریتم MRPSO [۳۴]، الگوریتم EPSDE [۳۵]، الگوریتم CCABC [۳۶] و الگوریتم کرم شب‌تاب [۳۷] مقایسه شده است. نتایج با پارامترهایی برابر برای همه روش‌ها، جمعیت ۱۰۰، تعداد ۳۰ متغیر، ۵۰۰۰۰۰ بار ارزیابی انجام شده است. نتایج براساس دو معیار میانگین بهترین هزینه ذرات (Mean) و انحراف استاندارد ذرات (StdDev) روی ۲۶ تابع معرفی شده در جدول ۱ تنظیم شده است. نتایج به صورت جامع در جدول ۲ آمده‌اند. در ادامه همین جدول تست فریدمن<sup>۱</sup> [۴۰] برای آزمایش معناداری نتایج برای همه روش‌ها آورده شده است. طبق آنچه در نتایج

روش پیشنهادی در نرم‌افزار متلب پیاده‌سازی شده و شبیه‌سازی در بخش‌های متفاوت و با روش‌های مشابه مختلف و تحت مجموعه تست‌های متفاوت سنجیده شده است. نتایج در دو بخش انجام داده و در هر بخش روش‌های مختلف برای مقایسه استفاده شده است. ابتدا در بخش اول نتایج در مجموعه تست‌های CEC 2009 [۳۲] به دست آمده‌اند. در بخش دوم از مجموعه تست‌های CEC 2013 [۳۶] استفاده شده است.

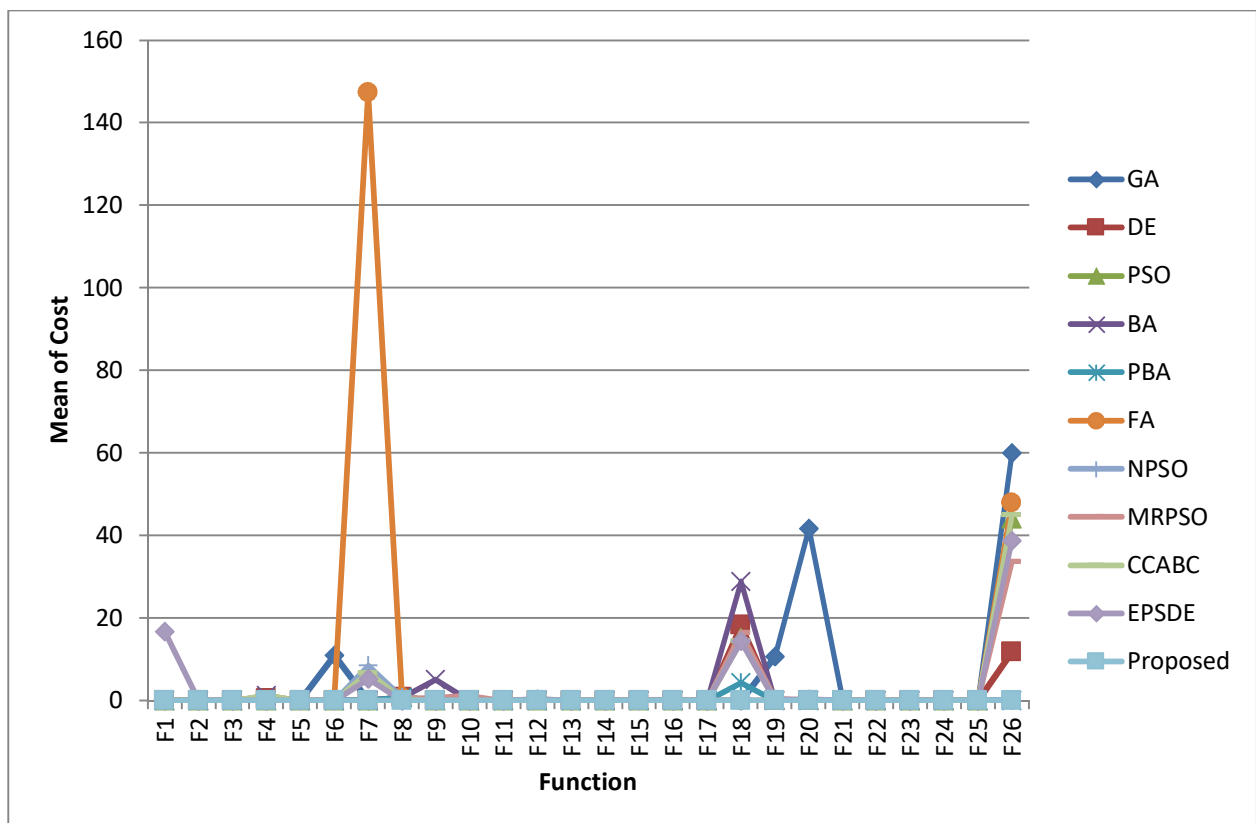
#### ۴-۱- بخش اول نتایج

در بخش اول الگوریتم‌ها روی مجموعه تست‌های CEC 2009 [۳۲] پیاده‌سازی انجام شده است. روش پیشنهادی در این بخش با روش‌های

F15 و F13 به جز الگوریتم ژنتیک همه روش‌ها به نقطه بهینه همگرا شدند. در تابع‌های F3، F14، F17 و F21 همه روش‌ها توانسته‌اند به نقطه بهینه همگرا شوند. روش پیشنهادی در توابع F8، F12 و F18 به تنهایی به جواب بهینه دست پیدا کند. تست فریدمن انجام گرفته بر روی نتایج کل روش‌ها نشان‌دهنده روند طبیعی نتایج و عدم وابستگی به روند تصادفی بوده است.

به دست آمده است روش پیشنهادی در همه ۲۶ تابع شاخص توانسته به نقطه بهینه مورد نظر تابع همگرا شود.

نتایج نشان می‌دهد به دلیل اینکه میانگین هزینه ذرات ملاک قرار داده شده است، می‌توان گفت که الگوریتم پیشنهادی در همه ۲۶ تست توانسته کاملاً به بهینه همگرا گردد. با این تفصیل منطقی به نظر می‌رسد که در پایان ارزیابی تعیین شده که اینجا ۱۰۰۰ بوده است، انحراف معیار ذرات هم صفر بوده است. انحراف معیار به این دلیل صفر شده که همه ذرات توانسته‌اند روی جواب بهینه همگرا شوند و روی هم قرار گیرند. در روی چند تابع روش پیشنهادی به طور مشترک با چند روش دیگر بهترین عملکرد را در کیفیت نتایج داشته‌اند. در توابع F11،



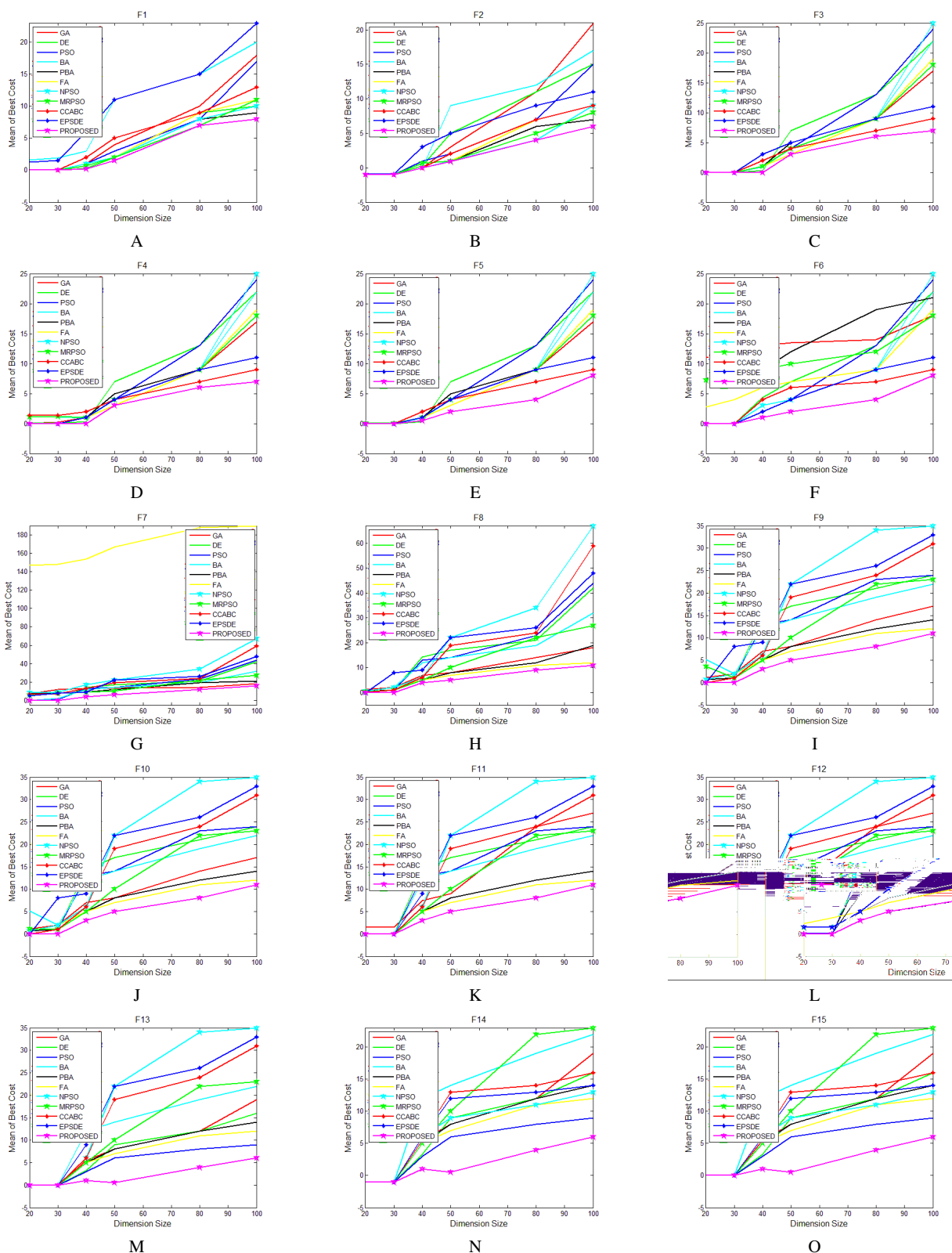
شکل ۶: نمودار مقایسه بهترین هزینه روش‌ها در مجموعه تست CEC 2009

افزایش تعداد متغیرها یا افزایش ابعاد مسئله از کیفیت بهتر به سمت کیفیت بدتر به دلیل پیچیده‌تر شدن مکانیزم مسئله مورد بحث تغییر کرده است.

شکل ۸ نمودار تست نتایج همه روش‌ها را بر روی مقادیر متفاوت ارزیابی نشان داده است. ارزیابی‌های متفاوت نشان می‌دهد که در این نمودار همگرایی الگوریتم پیشنهادی در مقادیر متفاوت ارزیابی به مرور بیشتر شدن تعداد ارزیابی به سمت جواب بهینه همگرا شده است. مقدار ارزیابی‌ها برای همه روش‌ها روی ۱۰۰، ۲۰۰، ۵۰۰، ۱۰۰۰، ۱۰۰۰۰ و ۵۰۰۰۰۰ تست شده‌اند. نتایج ۲۶ تابع نشان داده که الگوریتم پیشنهادی بهترین عملکرد را روی همه مقادیر متفاوت ارزیابی‌ها داشته است و توانسته در همه ارزیابی‌ها بهترین هزینه را پیدا کند. روند نتایج با افزایش مقدار ارزیابی به سمت بهتر بودن پیش رفته است.

نتایج روش‌های مورد مقایسه براساس ابعاد متفاوت اندازه گرفته شده و در شکل ۶ آمده است. ابعاد روی مقادیر ۲۰، ۳۰، ۴۰، ۵۰، ۸۰ و ۱۰۰ تغییر پیدا کرده و نتایج روی ۲۶ تابع شاخص برای همه روش‌ها تست شده است. در اینجا منظور از ابعاد تعداد متغیرهای الگوریتم می‌باشد. نتایج حاکی از آن است که همگرایی به بهینه تقریباً در تمامی ابعاد برای الگوریتم پیشنهادی یکسان بوده است. نتایج نشان می‌دهد که الگوریتم پیشنهادی در همه تست‌های انجام شده دارای بهترین کیفیت نتایج بوده است. استفاده از مکانیزم پیشنهادی در اینکه ذرات از جای مناسب‌تری نسبت به جواب مسئله شروع به جستجو کنند و در مکان‌های مناسب جستجوی بیشتری کنند در رسیدن به این میزان از کیفیت نتایج مؤثر بوده است. به طور طبیعی روند نتایج با





شکل ۷: نمودار بهترین هزینه روش‌های مورد مقایسه در ابعاد ۲۰، ۳۰، ۴۰، ۵۰، ۸۰ و ۱۰۰ بر روی ۱۵ تابع از ۲۸ تابع cec 2009

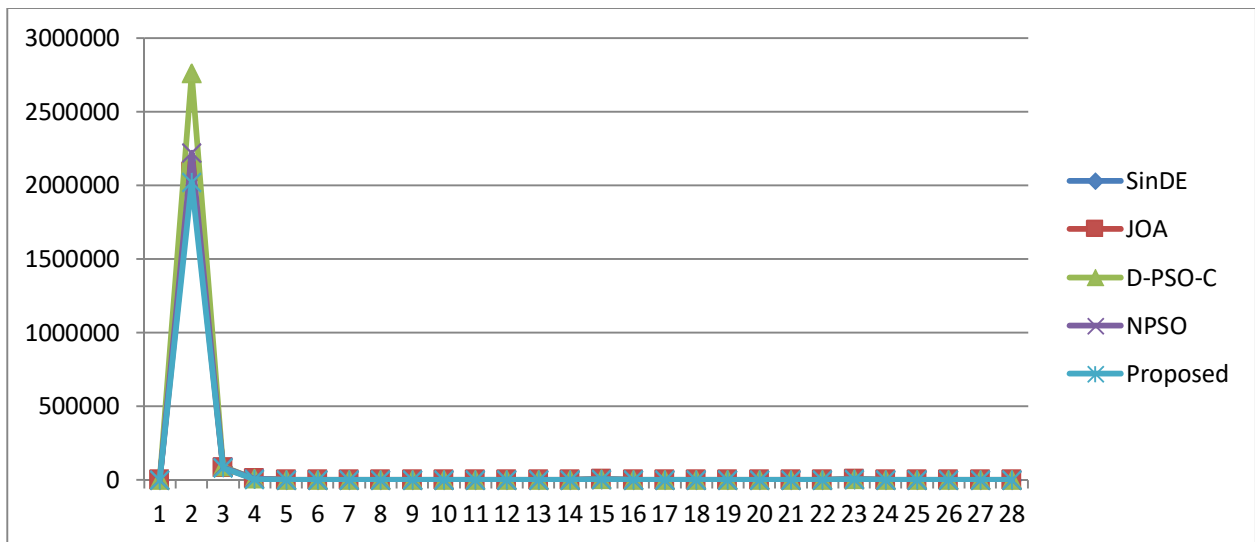
شکل ۷ شامل ۱۵ بخش است که با عنوان‌های A تا O جدا شده‌اند، نموداری را نشان می‌دهد که براساس آن همه روش‌ها با ابعاد که به صورت صعودی افزایش یافته‌اند تست شده‌اند. این نمودار برای اثبات پایداری روش‌های موردنظر نیز بوده است، بدین‌منظور که هر چه

[۲۷] هستند و در تعداد ۴۰ جمعیت، تعداد ارزیابی ۱۰۰۰۰، تعداد ۳۰ متغیر و از میانگین ۵۱ بار اجرا در روی ۲۸ تابع مورد نظر می‌باشد. به جز تابع F7 که روش JOA بهترین هزینه را در پایان تعداد ارزیابی‌های تعیین شده به خودش اختصاص داده است، در بقیه ۲۷ تابع هزینه روش پیشنهادی بهترین هزینه را پیدا کرده است. در تابع F28 نیز الگوریتم JOA هم‌رتبه با الگوریتم پیشنهادی بوده و این روش نیز در این تابع نقطه بهینه را پیدا کرده است. در ۲۶ تابع نیز روش پیشنهادی به‌تنهایی بهترین عملکرد را داشته است. میزان انحراف استاندارد روش پیشنهادی که در اغلب توابع کم بوده نشان داده روش توانسته از یک پراکندگی تصادفی شروع کند و پس از افزایش پراکندگی در اواخر ارزیابی با کاهش میزان تنوع توانسته اغلب ذرات را به سمت بهینه عمومی همگرا کند.

تعداد متغیرها بالاتر می‌رود کیفیت مطلوب حفظ شده روش پیشنهادی نسبت به سایر روش‌ها، اثبات پایداری روش پیشنهادی در تمامی شرایط را نشان داده است. روش پیشنهادی در ابعاد و شرایط محیطی پیچیده همان کیفیتی را نسبت به سایر روش‌ها حفظ کرده که در ابعاد پایین و ساده‌تر نشان داده است. در بخش دوم روش پیشنهادی با الگوریتم‌هایی مشابه که در دو سال اخیر به چاپ رسیده‌اند، مقایسه شده است. در پیوست ۳ برای اثبات پایداری روش پیشنهادی، این روش را تحت پارامترها و شرایط متفاوت مسئله آزمایش کرده و نتایج حاکی از یک ثبات کیفیت نتایج بوده است.

#### ۴-۲- بخش دوم نتایج

در این بخش مجموعه تابع مورد تست مجموعه CEC 2013 [۳۶] می‌باشد. روش‌هایی که مورد مقایسه قرار گرفته‌اند، روش sinDE [۲۵]، روش JOA [۲۶]، روش NPSO [۲۱] و روش D-PSO-C



شکل ۸: نمودار مقایسه بهترین هزینه روش‌ها در مجموعه تست CEC 2009

تست T نتایج مربوط به هزینه ذرات الگوریتم‌ها آمده است که نتایج تست نشان‌دهنده عدم تصادفی بودن نتایج و میزان معناداری بالای اعداد حاصل از بهترین هزینه ذرات بوده است.

#### ۴-۳- تست T و تحلیل پیچیدگی زمانی با الگوریتم‌های بخش

۴-۲

در قسمت آخر بخش نتایج تست T [۴۱] و شایستگی زمانی روش‌ها در آزمایش انجام شده در بخش ۴-۲ سنجیده شده است. در جدول ۱

جدول ۱: تست T برای نتایج الگوریتم‌های مورد مقایسه بخش ۴-۲

Algorithm	Confidence interval 95%		Test Value	p-value	or $H_0 H_1$
	Low	upper			
sinDE	۷.۲۹	۲۵.۳۱	۱۷	.	$H_1$
JOA	۰.۸۴۷	۶.۳۰۱	۳	.	$H_1$
D-PSO-C	۳.۱۸	۸.۲۶۱	۴.۵	.	$H_1$
NPSO	۶.۲۳	۲۵.۰۲	۱۵.۵	.	$H_1$
Proposed	۷.۳۳	۲۲.۷۷	۱۵.۵	.	$H_1$

داده است. با توجه به پیچیدگی زمانی تقریباً برابر روش‌ها می‌توان نتیجه گرفت که روش پیشنهادی در فاز شایستگی زمانی نیز بهتر عمل کرده است. در ادامه در دو نمودار جدا میانگین بهترین هزینه و میانگین زمان اجرا برای روش‌ها در ۲۸ تابع بخش ۳-۴ آمده است.

در جدول ۲ میانگین بهترین هزینه به دست آمده از روش‌ها به همراه میانگین زمان اجرای الگوریتم براساس ساعت و همچنین پیچیدگی زمانی الگوریتم‌ها آمده است. نتایج نشان داده که روش پیشنهادی میانگین بهترین هزینه کمتری نسبت به بقیه روش‌ها داشته است، در حالی که در میانگین زمان کمتری تعداد تکرارها (۱۰۰۰۰) را انجام

جدول ۲: نتایج زمان اجرا و پیچیدگی زمانی

الگوریتم	sinDE	JOA	D-PSO-C	NPSO	Proposed
پیچیدگی زمانی	O(n2logn)	O(nlogn)	O(nlogn)	O(nlogn)	O(nlogn)
میانگین زمان اجرا	H ۴۲/۲۶	H ۴۶/۴۲	H ۴۱/۱۱	H ۴۰/۹۸	H ۴۰/۳۱
میانگین بهترین هزینه	۹۸۲۳۴	۹۵۰۲۱	۱۲۴۶۴۵	۱۰۰۸۹۱	۸۸۰۴۸

### ۵- نتیجه

براساس شایستگی ارزش گذاری شده و سعی گردید در معادلات سرعت ذرات تغییراتی صورت داده تا در هنگام حرکت ذرات در فضاهای پرارزش سرعت ذرات کمتر باشد تا جستجوی بیشتری صورت پذیرد. بالعکس در فضاهای کم ارزش ذرات با سرعت بیشتری به حرکت درآمده‌اند تا ذرات زودتر از آن فضاها دور گردند. شبیه سازی در نرم افزار متلب انجام داده و نتایج روش پیشنهادی در سه بخش مجزا با روش‌های گوناگون مشابه پایه و جدید مقایسه شده است. نتایج به دست آمده نشان داد الگوریتم پیشنهادی توانسته در توابع ایستا الگوریتمی کارا و قابل اتکا باشد.

براساس آنچه در این مقاله آمد برای کارهای آتی می‌توان چنین پیشنهادهایی را مطرح ساخت: استفاده از نظریه آشوب در جمعیت‌دهی اولیه، استفاده از ایده ذرات کوانتوم در الگوریتم فوق، استفاده الگوریتم فوق برای حل مسائل بهینه سازی پویا.

### پیوست‌ها

پیوست ۱: نتایج مقایسه روش‌ها در مجموعه تست CEC 2009

	Criteria	GA	DE	PSO	BA	PBA	FA	NPSO	MRPSO	CCABC	EPSDE	Proposed
F1	Mean	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	- ۱/۸۸ E-۵ ‡	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	- ۱/۴۸ E-۵ ‡	۰/۰۰
	StdDev	۰/۰۰	۰/۰۰	۰/۰۰	- ۱/۹۴ E-۵	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	- ۱/۶۴ E-۵	۰/۰۰
F2	Mean	- ۱ \$	- ۱ \$	- ۱ \$	- ۱/۴۹۹۹۹ ‡	- ۱ \$	- ۱ \$	- ۱ \$	- ۱ \$	- ۱ \$	- ۱/۸۹۶۵ ‡	- ۱
	StdDev	۰/۰۰	۰/۰۰	۰/۰۰	- ۴/۵۰ E-۵	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	- ۳/۵۶ E-۵	۰/۰۰
F3	Mean	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰
	StdDev	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F4	Mean	۰/۰۱۴۹۴ ‡	۰/۴۰۹۱ ‡	۰/۰۰\$	۱/۱۱۶۷۰ ‡	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۱/۱۳۵۶۰ ‡	۱/۳۱۷۶۰ ‡	۰/۰۰\$	۰/۰۰
	StdDev	۰/۰۰۷۳۶	۰/۰۸۱۹۸	۰/۰۰	۰/۴۶۶۲۳	۰/۰۰	۰/۰۰	۰/۰۰	۰/۴۵۸۲۳	۰/۷۶۶۴۳	۰/۰۰	۰/۰۰
F5	Mean	۰/۱۲۳۶۶ ‡	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۰\$	۰/۰۱۱۲۵ ‡	۰/۰۰\$	۰/۰۰\$	۰/۰۰

	StdDev	۰/۰۰۰۴۵۳	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰۰۳۶۵	۰/۰۰	۰/۰۰	۰/۰۰
F6	Mean	۱/۱۰۲۵۴ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۷/۵۹ E-۱۰ ‡	۲/۱۷۳۹ E-۱۰ ‡	۰/۰۰§	۷/۲۶ E-۱۰ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰
	StdDev	۱/۳۸۶۸	۰/۰۰	۰/۰۰	۰/۰۰	۷/۱۰ E-۱۰	۱/۱۵ E-۱۱	۰/۰۰	۷/۲۹ E-۱۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F7	Mean	۷/۴۰ E+۳ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰	۱۴۷/۴۰ ‡	۸/۵۱۴ ‡	۵/۵۴۴ ‡	۶/۵۶۷ ‡	۵/۲۶۲ ‡	۰/۰۰	۰/۰۰
	StdDev	۱/۱۴ E+۳	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۴۴۸/۵۷	۷/۷۶۸	۵/۷۱۲	۶/۱۲۸	۵/۰۳۵	۰/۰۰	۰/۰۰
F8	Mean	۱/۳۲ E+۳ ‡	-۰/۶۶۶۶۷ ‡	-۰/۶۶۶۶۷ ‡	-۰/۶۶۶۶۷ ‡	-۰/۶۶۶۶۷ ‡	-۰/۶۶۶۶۷ ‡	-۰/۶۶۶۶۷ ‡	-۰/۶۶۶۶۷ ‡	-۰/۶۶۶۶۷ ‡	-۰/۶۶۶۶۷ ‡	۰/۰۰§	۰/۰۰
	StdDev	۲/۶۶ E+۲	۲/۴۹ E+۲	۲/۹۱ E+۲	۱/۱۶ E-۹	۵/۶۵ E-۱۰	۰/۰۰	۰/۰۰	۲/۱۳ E+۲	۲/۶۹ E-۱۰	۰/۰۰	۰/۰۰	۰/۰۰
F9	Mean	۱/۱۷ E+۳ ‡	۰/۰۰§	۰/۰۰§	۵/۱۲۳۷ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	-۰/۵۴۹۷ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰
	StdDev	۷۹/۵۹	۰/۰۰	۰/۰۰	-۰/۳۹۲۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F10	Mean	۱/۱۱ E+۳ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰
	StdDev	۷۴/۲۱	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F11	Mean	۱/۴۸ E+۳ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰
	StdDev	۱۲/۴۰۹	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F12	Mean	-۰/۱۸۰۷ ‡	-۰/۰۰۱۳ ‡	-۰/۰۰۱۱ ‡	۱/۷۲ E-۶ ‡	-۰/۰۰۶۷ ‡	۳/۶۶ E-۳ ‡	۹/۷۰ E-۴ ‡	۳/۵۲ E-۵ ‡	-۰/۰۰۴۲ ‡	-۰/۰۰۲۳ ‡	۰/۰۰	۰/۰۰
	StdDev	-۰/۰۲۷۱	-۰/۰۰۰۴۲	-۰/۰۰۰۲۸	۱/۵۸ E-۶	-۰/۰۰۱۳۳	-۰/۰۰۱۶۰	-۰/۰۰۱۲۵	۱/۷۵ E-۵	-۰/۰۰۷۶۵	-۰/۰۱۵۵۵	۰/۰۰	۰/۰۰
F13	Mean	-۰/۰۰۴۲۴ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰
	StdDev	-۰/۰۰۴۶۷	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F14	Mean	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §	-۱/۰۰۳۱۶۳ §
	StdDev	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F15	Mean	-۰/۰۶۸۲ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰
	StdDev	-۰/۰۷۸۲	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F16	Mean	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰
	StdDev	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F17	Mean	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §	-۱/۸۶/۷۳ §
	StdDev	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F18	Mean	۱/۹۶ E+۵ ‡	۱/۸۲۰۲۹ ‡	۱/۵۰۸۸ ‡	۲/۸۱۳۴ ‡	۴/۲۸۳۱ ‡	۲/۰۲ E+۱ ‡	۱/۰۴ E-۷ ‡	۱/۶۳۲۶ ‡	۱/۴۴۳۸ ‡	۱/۶۶۶۸ ‡	۰/۰۰	۰/۰۰
	StdDev	۳/۸۵ E+۴ ‡	۵/۰۳۶۱	۳۴/۱۷۰	-۱/۰۰۵۹	۵/۷۸۷۷	۱/۱۴۷۹	۲/۹۵ E-۷	۲۲/۴۳۱	۲۹/۵۷۰	۲۱/۸۷۰	۰/۰۰	۰/۰۰
F19	Mean	۱/۰۶۳۳ ‡	-۰/۰۰۱۴۸ ‡	-۰/۰۱۷۳ ‡	۰/۰۰§	-۰/۰۰۴۶ ‡	۰/۰۰§	۰/۰۰§	-۰/۴۲۷ ‡	-۰/۰۹۷۳ ‡	-۰/۰۸۲۴ ‡	۰/۰۰	۰/۰۰
	StdDev	-۰/۱۶۱۴	-۰/۰۰۲۹	-۰/۰۰۲۰۸	۰/۰۰	-۰/۰۰۶۷	۰/۰۰	۰/۰۰	-۰/۷۲۰۸	-۰/۰۹۷۷	-۰/۰۸۶۹	۰/۰۰	۰/۰۰
F20	Mean	۴/۱۶۷۱ ‡	۰/۰۰§	-۰/۱۶۴۶ ‡	۰/۰۰§	۳/۱۲ E-۸ ‡	۶/۵۶ E-۱۰ ‡	۰/۰۰§	۰/۰۰§	-۰/۱۱۴۳ ‡	-۰/۲۱۳۶ ‡	۰/۰۰	۰/۰۰
	StdDev	-۰/۱۷۸۱	۰/۰۰	-۰/۴۹۳۸	۰/۰۰	۳/۹۸ E-۸	۱/۳۴ E-۹	۰/۰۰	۰/۰۰	-۰/۳۳۵۸	-۰/۵۴۹۳	۰/۰۰	۰/۰۰
F21	Mean	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰
	StdDev	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F22	Mean	۰/۰۰§	۰/۰۰§	۰/۰۰§	-۰/۰۰۰۵۳ ‡	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	۰/۰۰§	-۰/۰۰۱۲۳ ‡	۰/۰۰
	StdDev	۰/۰۰	۰/۰۰	۰/۰۰	-۰/۰۰۰۷۴	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	-۰/۰۰۱۴۹	۰/۰۰
F23	Mean	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §	-۱/۵۷۲۸ §	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §	-۱/۸۰۱۳ §
	StdDev	۰/۰۰	۰/۰۰	-۰/۱۱۹۸۶	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰	۰/۰۰
F24	Mean	-۴/۶۴۴ ‡	-۴/۶۸۳ ‡	-۲/۴۹۰۸ ‡	-۴/۶۸۷۷ §	-۴/۶۸۷۷ §	۴/۶۰ E+۰ ‡	-۴/۶۸۷۷ §	-۱/۳۵۷۷ ‡	-۳/۶۴۴ ‡	-۲/۴۴۴ ‡	-۴/۶۸۷۷ §	-۴/۶۸۷۷ §

F25	StdDev	۰/۰۹۷۸	۰/۰۱۲۵	۰/۳۵۶۹	۰/۰۰	۰/۰۰	۰/۰۹۲۶	۰/۰۰	۰/۷۸۶۱	۰/۰۰۷۸	۰/۰۳۵۹	۰/۰۰
	Mean	-۹/۴۹۶±	-۹/۵۹۱±	-۴/۰۰۷۱±	-۹/۶۰۲۲±	-۹/۶۰۲۲±	-۹/۲۹۵۲±	-۹/۶۵۳۵±	-۹/۷۷۵۲±	-۹/۱۱۵۲±	-۹/۶۵۳۵±	-۹/۶۶۰۲
F26	StdDev	۰/۱۴۱۱	۰/۰۶۴۲	۰/۰۵۲۶	۰/۰۰	۰/۰۰	۰/۳۸۲۰	۰/۰۲۴۹	۰/۰۱۹۹	۰/۰۲۸۹	۰/۰۱۴۹	۰/۰۰
	Mean	۵۹/۹۲±	۱۱/۷۱۶±	۴۳/۹۷۷±	۰/۰۰±	۰/۰۰±	۴۷/۸۸±	۰/۰۰±	۳۳/۵۵±	۴۴/۹۶۶±	۳۸/۷۸±	۰/۰۰
	StdDev	۴/۵۶۴	۲/۵۳۸	۱۱/۷۲۸	۰/۰۰	۰/۰۰	۱۶/۱۳۲	۰/۰۰	۲۱/۷۸۸	۱۱/۸۸۶	۱۱/۰۹۸	۰/۰۰

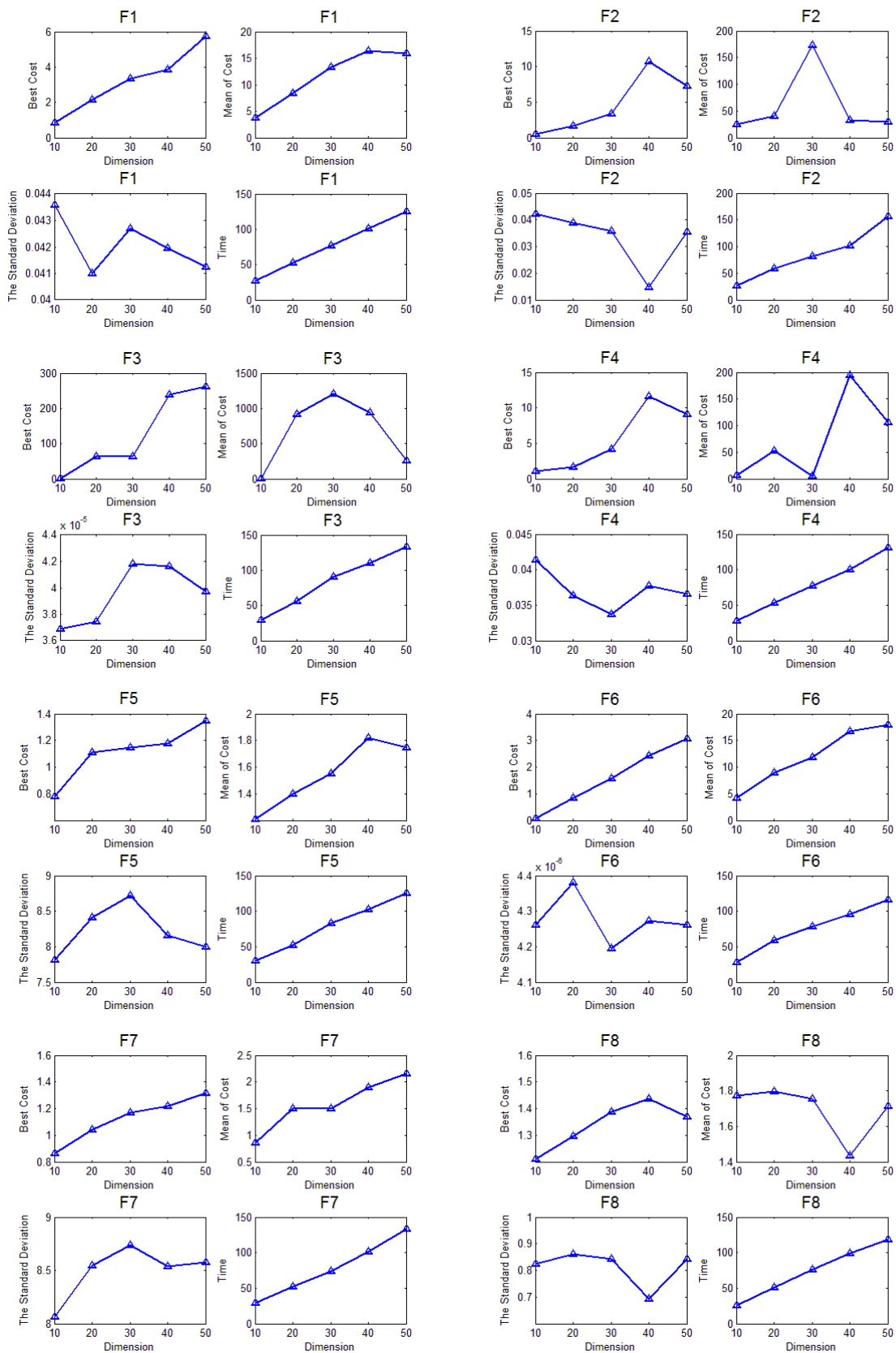
علامت "±" یعنی عملکرد الگوریتم پیشنهادی بهتر بوده، علامت "+" یعنی عملکرد پیشنهادی بدتر بوده و علامت "§" به معنی عملکرد یکسان بوده است.

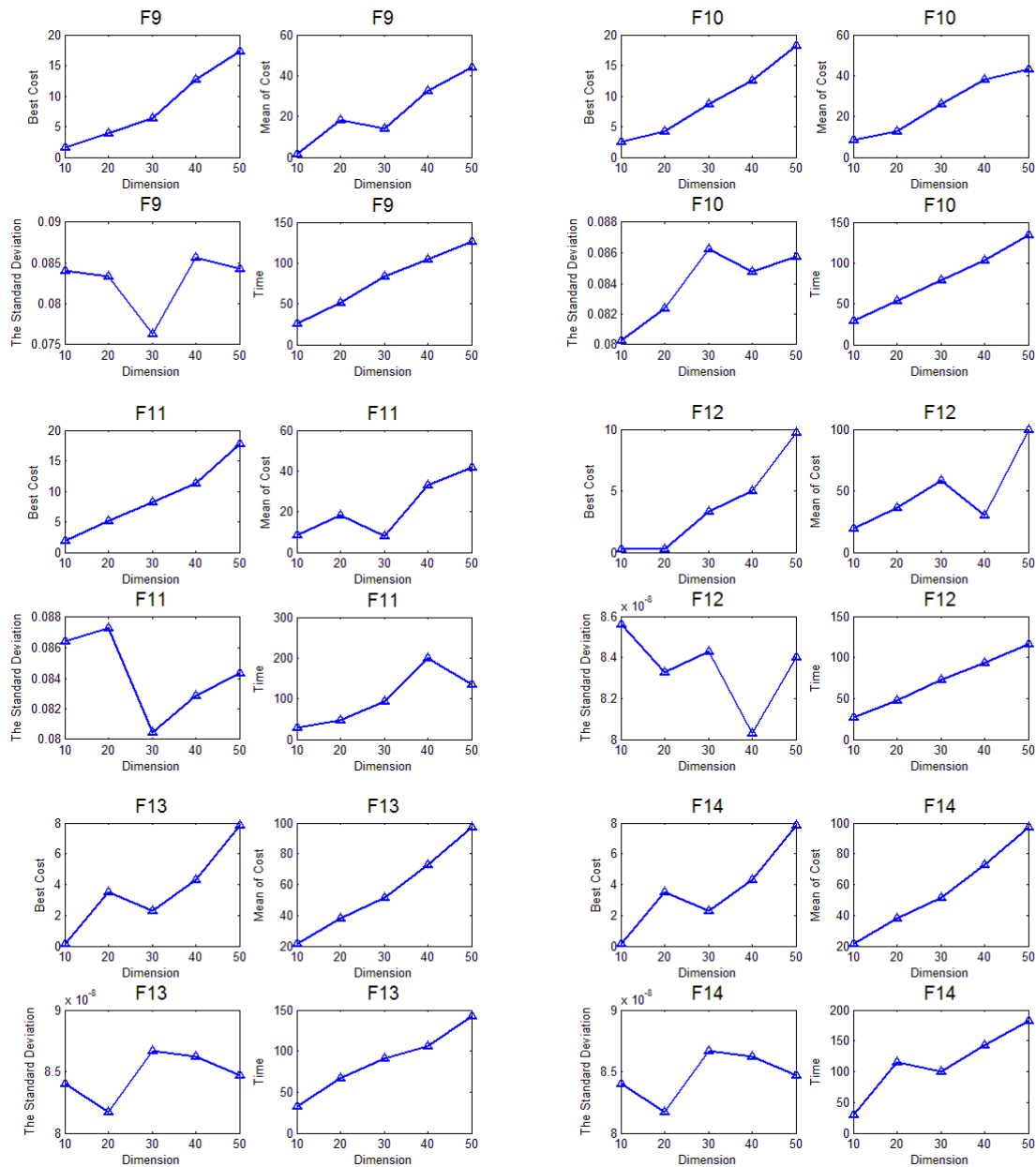
Friedman Test												
		۵/۵۶	۳/۶۵	۴/۰۸	۳/۸۵	۳/۵۶	۴/۰۶	۳/۱۵	۳/۱۱	۴/۱۲	۳/۷۷	۴/۰۲
p-value		۳/۴۵ E-۰۷	۲/۱۵ E-۰۷	۵/۶۶ E-۰۷	۲/۷۰ E-۰۷	۳/۸۱ E-۰۷	۳/۰۵ E-۰۷	۴/۰۹ E-۰۷	۲/۱۱ E-۰۷	۵/۴۰ E-۰۷	۴/۴۷ E-۰۷	۵/۳۱ E-۰۷
Statistic		۴۰/۶۰۹	۳۶/۱۲۳	۳۹/۱۰۹	۴۱/۱۱۲	۴۶/۷۱۲	۴۴/۶۱۱	۳۸/۷۵۵	۴۲/۵۴۳	۳۹/۴۲۱	۴۴/۱۰۲	۴۰/۵۴۹

پیوست ۲: نتایج روش پیشنهادی و ۴ روش جدید بهینه‌سازی بر روی ۲۸ تابع شاخص

Function	SinDE		JOA		D-PSO-C		NPSO		Proposed	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1	۲/۲۳ E-۱۴	۶/۸۳ E-۱۴	۱/۷۸ E-۱۴	۳/۳۳ E-۱۴	۲/۴۴ E-۱۴	۶/۱۳ E-۱۱	۲/۵۲ E-۱۴	۲/۷۷ E-۱۲	۱/۳۶ E-۱۴	۲/۱۸ E-۱۴
2	۲/۱۶ E+۰۶	۶/۱۵ E+۰۵	۲/۰۹ E+۰۶	۴/۳۶ E+۰۵	۲/۷۶ E+۰۶	۱/۱۰ E+۰۵	۲/۲۲ E+۰۶	۱/۳۲ E+۰۵	۲/۰۲ E+۰۶	۳/۴۶ E+۰۵
3	۸/۴۹ E+۰۵	۲/۱۱ E+۰۵	۸/۱۹ E+۰۴	۴/۳۱ E+۰۵	۹/۱۱ E+۰۴	۱/۶۲ E+۰۵	۸/۵۷ E+۰۴	۳/۶۵ E+۰۵	۸/۱۳ E+۰۴	۱/۳۳ E+۰۵
4	۶/۳۸ E+۰۳	۲/۰۰ E+۰۳	۵/۸۲ E+۰۳	۱/۸۰ E+۰۳	۶/۵۰ E+۰۳	۳/۴۹ E+۰۳	۶/۴۷ E+۰۳	۳/۰۹ E+۰۳	۴/۷۱ E+۰۳	۱/۷۷ E+۰۳
5	۱/۱۴ E-۱۳	۷/۶۵ E-۲۹	۱/۱۱ E-۱۳	۵/۱۴ E-۲۹	۲/۴۷ E-۱۳	۶/۱۵ E-۲۹	۱/۲۲ E-۱۳	۱/۸۵ E-۲۹	۱/۰۴ E-۱۳	۲/۶۴ E-۲۹
6	۱/۴۶ E+۰۱	۲/۴۲ E+۰۰	۱/۵۲ E+۰۱	۴/۳۸ E+۰۰	۳/۵۴ E+۰۱	۳/۵۲ E+۰۰	۱/۷۶ E+۰۱	۱/۷۲ E+۰۰	۱/۳۲ E+۰۱	۱/۲۴ E+۰۰
7	۱/۲۱ E-۰۱	۱/۵۷ E-۰۱	۱/۰۹ E-۰۱	۲/۱۸ E-۰۱	۱/۸۱ E-۰۱	۴/۳۸ E-۰۱	۱/۳۴ E-۰۱	۲/۱۷ E-۰۱	۴/۵۱ E-۰۱	۲/۴۵ E-۰۱
8	۲/۰۹ E+۰۱	۴/۹۶ E-۰۲	۱/۷۸ E+۰۱	۲/۵۴ E-۰۲	۲/۶۷ E+۰۱	۳/۷۷ E-۰۲	۲/۱۳ E+۰۱	۶/۰۶ E-۰۲	۱/۴۷ E+۰۱	۲/۱۴ E-۰۲
9	۱/۵۲ E+۰۱	۳/۰۵ E+۰۰	۱/۴۸ E+۰۱	۲/۳۱ E+۰۰	۱/۷۸ E+۰۱	۲/۱۳ E+۰۰	۱/۵۰ E+۰۱	۲/۸۷ E+۰۰	۱/۳۳ E+۰۱	۱/۰۹ E+۰۰
10	۲/۰۴ E-۰۲	۱/۳۰ E-۰۲	۱/۶۶ E-۰۲	۲/۱۹ E-۰۲	۲/۶۵ E-۰۲	۳/۵۲ E-۰۲	۲/۱۹ E-۰۲	۲/۴۳ E-۰۲	۱/۵۲ E-۰۲	۱/۲۲ E-۰۲
11	۱/۹۵ E-۰۲	۱/۳۹ E-۰۱	۱/۹۸ E-۰۲	۲/۵۵ E-۰۱	۲/۳۴ E-۰۲	۲/۴۴ E-۰۱	۲/۰۰ E-۰۲	۴/۳۵ E-۰۱	۰/۰۰ E+۰۰	۰/۰۰ E+۰۰
12	۳/۰۲ E+۰۱	۸/۶۵ E+۰۰	۲/۷۸ E+۰۱	۲/۴۴ E+۰۰	۳/۶۶ E+۰۱	۴/۳۳ E+۰۰	۳/۱۸ E+۰۱	۴/۲۲ E+۰۰	۲/۲۵۴ E+۰۱	۲/۷۵ E+۰۰
13	۷/۳۳ E+۰۱	۲/۰۷ E+۰۱	۷/۱۴ E+۰۱	۱/۸۷ E+۰۱	۷/۷۶ E+۰۱	۴/۴۷ E+۰۱	۷/۴۰ E+۰۱	۳/۲۷ E+۰۱	۷/۰۲ E+۰۱	۲/۰۹ E+۰۱
14	۵/۰۴ E+۰۱	۱/۹۲ E+۰۱	۵/۰۰ E+۰۱	۳/۵۲ E+۰۱	۵/۷۸ E+۰۱	۵/۰۲ E+۰۱	۵/۱۶ E+۰۱	۶/۴۸ E+۰۱	۴/۷۸ E+۰۱	۲/۰۲ E+۰۱
15	۲/۹۵ E+۰۳	۴/۸۶ E+۰۲	۲/۸۲ E+۰۳	۳/۲۱ E+۰۲	۳/۱۳ E+۰۳	۳/۲۲ E+۰۲	۲/۹۷ E+۰۳	۴/۸۶ E+۰۲	۲/۴۸ E+۰۳	۲/۰۰ E+۰۲
16	۱/۴۷ E+۰۰	۲/۵۲ E-۰۱	۱/۵۹ E+۰۰	۱/۴۵ E-۰۱	۱/۸۷ E+۰۰	۳/۵۵ E-۰۱	۱/۷۵ E+۰۰	۲/۲۸ E-۰۱	۱/۴۶ E+۰۰	۱/۵۰ E-۰۱
17	۳/۳۷ E+۰۱	۷/۹۷ E-۰۱	۳/۲۳ E+۰۱	۵/۷۹ E-۰۱	۴/۰۹ E+۰۱	۳/۲۳ E-۰۱	۳/۵۹ E+۰۱	۴/۵۵ E-۰۱	۳/۰۸ E+۰۱	۲/۱۰ E-۰۱
18	۷/۸۶ E+۰۱	۱/۴۲ E+۰۱	۷/۴۸ E+۰۱	۳/۵۶ E+۰۱	۷/۸۹ E+۰۱	۳/۴۵ E+۰۱	۷/۵۵ E+۰۱	۴/۴۴ E+۰۱	۷/۱۶ E+۰۱	۲/۳۳ E+۰۱
19	۲/۴۲ E+۰۰	۳/۷۹ E-۰۱	۲/۶۴ E+۰۰	۲/۱۲ E-۰۱	۳/۷۵ E+۰۰	۲/۷۳ E-۰۱	۲/۳۳ E+۰۰	۲/۲۸ E-۰۱	۲/۰۶ E+۰۰	۱/۲۳ E-۰۱
20	۹/۹۹ E+۰۰	۵/۵۰ E-۰۱	۹/۷۵ E+۰۰	۳/۲۳ E-۰۱	۹/۸۳ E+۰۰	۴/۸۲ E-۰۱	۹/۸۱ E+۰۰	۴/۸۸ E-۰۱	۹/۳۴ E+۰۰	۲/۷۱ E-۰۱
21	۲/۸۷ E+۰۲	۶/۴۰ E+۰۱	۲/۸۱ E+۰۲	۳/۹۷ E+۰۱	۲/۹۹ E+۰۲	۶/۴۰ E+۰۱	۲/۹۱ E+۰۲	۳/۹۱ E+۰۱	۲/۷۷ E+۰۲	۲/۳۸ E+۰۱
22	۱/۴۹ E+۰۲	۱/۷۶ E+۰۱	۱/۳۲ E+۰۲	۲/۹۰ E+۰۱	۱/۵۵ E+۰۲	۴/۷۵ E+۰۱	۱/۵۷ E+۰۲	۳/۲۲ E+۰۱	۱/۲۰ E+۰۲	۳/۰۹ E+۰۱
23	۳/۱۴ E+۰۳	۵/۳۱ E+۰۲	۳/۲۸ E+۰۳	۳/۳۲ E+۰۲	۳/۸۳ E+۰۳	۴/۱۱ E+۰۲	۳/۲۰ E+۰۳	۲/۳۹ E+۰۲	۳/۱۲ E+۰۳	۲/۸۸ E+۰۲
24	۲/۰۰ E+۰۲	۷/۱۶ E-۰۳	۲/۱۱ E+۰۲	۳/۴۴ E-۰۳	۲/۶۷ E+۰۲	۴/۴۶ E-۰۳	۲/۵۲ E+۰۲	۶/۸۶ E-۰۳	۱/۸۹ E+۰۲	۴/۷۹ E-۰۳
25	۲/۴۹ E+۰۲	۶/۸۵ E+۰۰	۲/۳۷ E+۰۲	۲/۶۳ E+۰۰	۲/۴۷ E+۰۲	۳/۰۵ E+۰۰	۲/۵۱ E+۰۲	۵/۷۳ E+۰۰	۲/۰۸ E+۰۲	۳/۶۶ E+۰۰
26	۲/۰۲ E+۰۲	۱/۴۰ E+۰۱	۱/۸۸ E+۰۲	۴/۴۳ E+۰۱	۲/۳۷ E+۰۲	۵/۶۵ E+۰۱	۲/۱۲ E+۰۲	۴/۴۷ E+۰۱	۱/۷۹ E+۰۲	۲/۷۲ E+۰۱
27	۳/۰۱ E+۰۲	۲/۳۶ E+۰۰	۳/۰۹ E+۰۲	۴/۶۵ E+۰۰	۳/۴۵ E+۰۲	۷/۶۶ E+۰۰	۳/۲۰ E+۰۲	۶/۴۷ E+۰۰	۲/۸۱ E+۰۲	۳/۳۱ E+۰۰
28	۳/۰۰ E+۰۲	۰/۰۰ E+۰۰	۰/۰۰ E+۰۰	۰/۰۰ E+۰۰	۳/۲۲ E+۰۲	۲/۰۰ E+۰۰	۳/۵۴ E+۰۲	۳/۳۹ E+۰۰	۰/۰۰ E+۰۰	۰/۰۰ E+۰۰

پیوست ۳- نمودار بهترین هزینه، میانگین هزینه، انحراف استاندارد و زمان اجرای روش پیشنهادی بر روی ۱۴ تابع تست





[6] F. Ali and M. Tawhid, "A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems", *Ain Shams Engineering Journal*, doi.org/10.1016/j.asej.2016.07.008, 2016.

[7] J. Kennedy and R. Eberhar, "Particle Swarm Optimization", *Proceedings of the 4<sup>th</sup> IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.

[8] N F. Wan and L. Nolle, "Solving a multi-dimensional knapsack problem using hybrid particle". *23rd European Conference on Modelling and Simulation*, 2008.

[9] K B. Deep, "A socio-cognitive particle swarm optimization for multi-dimensional". *First International Conference on Emerging Trends in Engineering and*, 355-360, 2008.

[10] X. Shen, Y. Li, C. Chen, J. Yang, D. Zhang, "Greedy continuous particle swarm optimisation algorithm for the knapsack problems". *International Journal of*

مراجع

[1] R. Haupt and S. E. Haupt, "Practical Genetic Algorithms", 2nd Edition, John Wiley & Sons Inc, 2004.

[2] H. Yapıcı and N. Çetinkaya, "An Improved Particle Swarm Optimization Algorithm Using Eagle Strategy for Power Loss Minimization", *Hindawi, Mathematical Problems in Engineering*, doi.org/10.1155/2017/1063045, 2017.

[3] W. Sun and Y. Yuan, "Optimization Theory and Methods: Nonlinear Programming", Springer Science Business Media, LLC Press, 2006.

[4] Classical conditioning, "The Gale encyclopedia of psychology", Gale Group, p. 124, 2001.

[5] J. Holland, "Genetic algorithms and the optimal allocation of trials". *SIAM J. Comput.* 2, 88-105, 1979.

- cooperative learning strategy”, *Appl. Soft Comput.* 29, 169–183, 2015.
- [28] J. Wang, B. Zhou, Sh. Zhou, “An Improved Cuckoo Search Optimization Algorithm for the Problem of haotic Systems Parameter Estimation”, *Hindawi Publishing Corporation, Computational Intelligence and Neuroscience, Volume 2016, 10.1155/2016/2959370*, 2016.
- [29] E R. Tanweer, S. Suresh, N. Sundararajan, “Self regulating particle swarm optimization algorithm”, *Innovative Applications of Artificial Neural Networks in Engineering, Volume 294, 182–202*, 2015.
- [30] F T. Zhao, Zh. Yao, J. Luan, X. Son, “A Novel Fused Optimization Algorithm of Genetic Algorithm and Ant Colony Optimization”, *athematical Problems in Engineering, Volume 2016 (2016), Article ID 2167413*, 2016.
- [31] M. Thankur, “A new genetic algorithm for global optimization of multimodal continuous functions”, *Journal of Computational Science, 298–311*, 2014.
- [32] Q. Zhang, A. Zhou, Sh. Zhao, P. Suganthan, W. Liu, S. Tiwari, “Multiobjective optimization test instances for the CEC 2009 Special Session and Competition”, *Technical Report CES-487*, 2009.
- [33] R. Storn and K. Price, “Differential evolution a simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization, 11 (1997) 341–359*, 1997.
- [34] A. Gao and W B. Xu, “A new particle swarm algorithm and its globally convergent modifications”, *IEEE Trans. Syst. Man. Cy. B, vol. 41, no. 5, 1334–1351*, 2011.
- [35] R. Mallipeddi, P N. Suganthan, Q. Pan, M. Tasgetiren, “Differential evolution algorithm with ensemble of parameters and mutation strategies”, *Appl. Soft. Comput.* 1679-1696, 2011.
- [36] Y. Liang, Y. Liu, L. Zhang, “An Improved Artificial Bee Colony (ABC) Algorithm for Large Scale Optimization”, *2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), IEEE, 978-1-4799-2716-6/13/\$31.00*, 2013.
- [37] X S. Yang, “Nature-Inspired Metaheuristic Algorithms: Second Edition”, *Luniver Press*, 2011.
- [38] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, “GSA: A Gravitational Search Algorithm”, *Inform. Sciences, 2232-2248*, 2009.
- [39] R M. Rizk Allah, “Hybridization of Fruit Fly Optimization Algorithm and Firefly Algorithm for Solving Nonlinear Programming Problems”, *International Journal of Swarm Intelligence and Evolutionary Computation*, 2016.
- [40] J. G. Villegas, “Using Nonparametric Test to Compare the Performance of Metaheuristics”, *friedman-test-24062011.pdf*, 2001.
- [41] Statistical Consultant for Doctoral Students and Researchers, <http://www.statisticallysignificantconsulting.com/Ttest.htm>.
- [42] م. امیرعباسیان، ح. نظام‌آبادی پور، «الگوریتم جستجوی گرانشی چندهدفه مبتنی بر مرتب‌سازی چپه‌های مغلوب‌نشده»، *مجله مهندسی برق دانشگاه تبریز، شماره ۱ جلد ۴۱، ص ۶۱–۸۱، ۱۳۹۱*.
- [11] H S. Lopes and L S. Coelho, “Particle swarn optimization with fast local search for the blind traveling salesman problem”. *International Conference on Hybrid Intelligent Systems, 245–250*, 2005.
- [12] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”. *Journal of Global Optimization 39*, 2007.
- [13] A. Banharsakun and B. Sirinaovakul, “T. Achalakul, Job shop scheduling with the best-so-far ABC”. *Engineering Applications of Artificial Intelligence 25 (3), 583–593*, 2012.
- [14] D. Karaboga and B. Gorkemli, “A combinatorial artificial bee colony algorithm for traveling salesman problem”. *International Symposium on Intelligent Systems and Applications, pp. 50–53*, 2011.
- [15] Z. Geem, J. Kim, G. Loganathan, “A new heuristic optimization algorithm: Harmony search”. *Simulation, 60*, 2011.
- [16] DT. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, “The bees algorithm”. *Technical note, Cardiff University, UK: Manufacturing Engineering Center*, 2005.
- [17] D T. Pham, S. Otri, A. Afify, M. Mahmuddin, H. Al-Jabbouli, “Data clustering using the bees algorithm”. *40th CIRP International Seminar on Manufacturing Systems, 2007*.
- [18] D. Pham, E. Koc, J. Lee, J. Phruksanant, “Using the bees algorithm to schedule jobs for a machine”. *Proceedings of Eighth International Conference on Laser Metrology, 430–439, CMM and Machine, 2007*.
- [19] X. Miao, J. Chu, L. Zhang, J. Qiao, “An Evolutionary Neural Network Approach to Simple Prediction of Dam Deformation”, *Journal of Information & Computational Science, 1315–1324*, 2013.
- [20] M. Cheng and L. Lien, “Hybrid artificial intelligencebased pba for benchmark functions and facility layout design optimization”. *Journal of Computing in Civil Engineering, 26, 612–624*, 2012.
- [21] W. Feng and Ch. Liu, “A Novel Particle Swarm Optimization Algorithm for Global Optimization”, *Hindawi Publishing Corporation Computational Intelligence and Neuroscience Volume 2016, Article ID 9482073, 9 pages*, 2016.
- [22] X. S. Yang and S. Deb, “Cuckoo search via Levy flights”, in *Proc. NaBIC 2009, IEEE Publications, 210-214, Dec. 2009. 18 / Information Sciences XX 1–22 19, 2014*.
- [23] P. Civicioglu, “Transforming geocentric cartesian coordinates to geodeticcoordinates by using differential search algorithm”. *Comput, Geosci, 229-247*, 2012.
- [24] A. Gandomi, “Bird mating optimizer: An optimization algorithm inspired by birdmating strategies”. *Commun Nonlinear Sci, 1213-1228*, 2014.
- [25] A. Draa, S. Bouzoubia, I. Boukhalfa, “A sinusoidal differential evolution algorithmfor numerical optimization”, *Appl. Soft Comput.* 99–126, 2015.
- [26] G. Sun, R. Zhao, Y. Lan, “Joint operations algorithm for large-scale global optimization”. *Applied Soft Computing, 38: 1025-1039*, 2016.
- [27] X. Xu, Y. Tang, J. Li, CC. Hua, X P. Guan, “Dynamic multi-swarm particle swarmoptimizer with



[44] م. محمدپور، ح. پروین، « الگوریتم ژنتیک آشوب گونه مبتنی بر حافظه و خوشه بندی برای حل مسائل بهینه سازی پویا»، مجله مهندسی برق دانشگاه تبریز، شماره ۳ جلد ۴۶، ص ۷۷، ۱۳۹۵.

[43] ش. جمالی، س. ملک تاجی، م. آنالویی، « مکان یابی ماشین های مجازی با استفاده از الگوریتم رقابت استعماری»، مجله مهندسی برق دانشگاه تبریز، شماره ۱ جلد ۴۶، ص ۷۵، ۱۳۹۵.

#### زیر نویس ها

- <sup>1</sup> Maximum
- <sup>2</sup> Minimum
- <sup>3</sup> Extremum
- <sup>4</sup> Genetic Algorithm
- <sup>5</sup> Holland
- <sup>6</sup> Particle Swarm Optimization
- <sup>7</sup> Artificial Bee Colony
- <sup>8</sup> Harmony Search
- <sup>9</sup> Bee Algorithm
- <sup>۱۰</sup> Imperialist Competitive Algorithm
- <sup>۱۱</sup> Friedman