

Water Streams Optimization (WSTO): A new Metaheuristic Optimization method in High-Dimensional Problems

Hamed Majani¹ and Mehdi Nasri^{2*}

1- Department of Mechatronic Engineering, Khomeinishahr branch, Islamic Azad University, Isfahan, Iran.

2*- Department of Electrical Engineering, Khomeinishahr branch, Islamic Azad University, Isfahan, Iran.

¹ hamed.majani@iaukhsh.ac.ir, ^{2*} nasri_me@iaukhsh.ac.ir

Corresponding author's address: Mehdi Nasri, Department of Electrical Engineering, Khomeinishahr branch, Islamic Azad University, Isfahan, Iran.

Abstract- In recent years, various heuristic optimization methods have been developed. Many of these methods are inspired by behaviors in nature. In this paper, new nature-inspired algorithm based on behavior of water streams of rain, for solving of real-valued continues optimization problems is introduced. The proposed algorithm does not require the information of the first or second order Derivatives of the object function. Hence, it is a direct method. We investigate the properties of this algorithm. Besides, we apply the proposed algorithm to solve a non-linear optimization problem, where the object function is highly irregular (neither convex nor concave). In addition, the global optimal solution can be found. In the proposed algorithm, the searcher agents are a collection of water currents, which moved based on gravity. The proposed algorithm has been developed from a motivation to find a simpler and more effective search algorithm to optimize multi-dimensional numerical test functions. It is effective in searching and finding an optimum solution from a large search domain within an acceptable CPU time. Statistical analysis compared the solution quality with well-known heuristic search methods. The obtained results confirm the high performance of the proposed method in solving various nonlinear functions.

Keywords- Optimization, Heuristic search algorithms, Natural inspired computation, Water, stream.

الگوریتم بهینه‌سازی جریان‌های آب: روشی جدید در بهینه‌سازی مسائل با بعد بالا

حامد ماجانی^۱، مهدی نصری^{۲*}

^۱ - گروه مهندسی مکترونیک، واحد خمینی‌شهر، دانشگاه آزاد اسلامی، اصفهان، ایران.

^۲ - گروه مهندسی برق، واحد خمینی‌شهر، دانشگاه آزاد اسلامی، اصفهان، ایران.

^{۱*} hamed.majani@iaukhsh.ac.ir ، ^{۲*} masri_me@iaukhsh.ac.ir

* نشانی نویسنده مسئول: مهدی نصری، استان اصفهان، خمینی‌شهر، انتهای بلوار دانشجو، دانشگاه آزاد اسلامی واحد خمینی‌شهر، دانشکده فنی-مهندسی، گروه مهندسی برق.

چکیده- در این مقاله، یک روش بهینه‌سازی فراابتکاری برگرفته از رفتار و حرکت جریان‌های آب بر روی زمین در رسیدن به پست‌ترین مکان ممکن، جهت حل مسائل پیوسته ارائه شده است. حرکت ساده جریان آب بر روی زمین به‌طور مشخصی کارآمد و بهینه می‌باشد و همیشه کوتاه‌ترین و سریع‌ترین مسیر رسیدن به عمیق‌ترین نقطه را در بر دارد. در الگوریتم ارائه شده، حرکت‌های ساده آب در مسیریابی، تغییر جهت و حتی ایجاد تندآب و گرداب به صورت عملگرهای ریاضی مختلف شبیه‌سازی شده است. در ادامه مقاله، جهت بررسی کارایی الگوریتم فراابتکاری پیشنهادی، بیست‌وسه تابع استاندارد مختلف مورد استفاده قرار گرفته و کارایی الگوریتم با برخی روش‌های کلاسیک بهینه‌سازی فراابتکاری مورد مقایسه قرار گرفته است. نتایج آزمایش‌ها مؤید این است که الگوریتم ارائه شده از نظر دقت و سرعت در بیشتر توابع آزمون استاندارد عملکرد بهتری را ارائه می‌دهد، به‌ویژه در بعد بالا این برتری به‌طور معنی‌داری قابل مشاهده است و اختلاف بسیار زیادی با دیگر الگوریتم‌ها دارد، به‌طوری‌که الگوریتم‌های دیگر تقریباً قادر به بهینه‌سازی در ابعاد بالا نیستند. در بعد ۳۰، میانگین زمان اجرای برنامه الگوریتم آب باران نسبت به الگوریتم وراثتی ۱/۶۵۷ و نسبت به الگوریتم جستجوی فاخته ۱/۲۷۴ می‌باشد. در جایی که میانگین خطای الگوریتم ارائه شده به الگوریتم وراثتی ۰/۰۰۶ و نسبت به الگوریتم جستجوی فاخته ۰/۰۰۴ می‌باشد. بنابراین افزایش ۹۴٪ دقت درازای ۶۵٪ درصد زمان بیشتر نسبت به الگوریتم وراثتی و افزایش ۹۹٪ دقت درازای ۲۷٪ زمان بیشتر نسبت به الگوریتم جستجوی فاخته مشاهده می‌شود.

واژه‌های کلیدی: بهینه‌سازی، فراابتکاری، آب، جریان آب.

۱- مقدمه

در عملیات برای اشاره به مباحث بهینه‌سازی به کار می‌رود. از یک منظر می‌توان روش‌های بهینه‌سازی را به دو گروه روش‌های بهینه‌سازی کلاسیک (الگوریتم‌های دقیق) و روش‌های بهینه‌سازی ابتکاری تقسیم کرد. روش‌های بهینه‌سازی کلاسیک شامل روش دیفرانسیل تابع، گرادیان نزولی و نیوتن و ... می‌باشد که جواب بهینه را به‌طور دقیق می‌یابد. این روش‌ها در مورد مسائل بهینه‌سازی سخت کارایی کافی را ندارند و زمان اجرای آن‌ها متناسب با ابعاد مسائل به‌صورت نمایی افزایش می‌یابد و تابع مربوطه حتماً باید مشتق‌پذیر بوده، تعداد

بهینه‌سازی به برگزیدن بهترین عضو از یک مجموعه از اعضای دست‌یافتنی، اشاره می‌کند [۱]. بهینه‌سازی در طراحی و نگهداری بسیاری از سامانه‌های مهندسی و اقتصادی به‌منظور کمینه کردن هزینه یا بیشینه کردن سود کاربرد دارد. به دلیل کاربرد وسیع بهینه‌سازی در علوم متفاوت، این مبحث رشد بسیاری کرده است، به‌طوری‌که در ریاضیات، مدیریت، صنایع و بسیاری از شاخه‌های علوم مورد بررسی قرار می‌گیرد و حتی نام‌های متفاوتی از قبیل برنامه‌ریزی ریاضی و تحقیق

ریشه‌های مشتق تابع نیز متناهی باشد.

در چند دهه‌ی گذشته تحقیقات زیادی بر روی الگوریتم‌های بهینه‌سازی که از فرآیندهای طبیعی یا پدیده‌های اجتماعی الهام گرفته‌اند، انجام شده‌است و یک زمینه‌ی تحقیقاتی بسیار فعال می‌باشد. تحقیقات در زمینه این الگوریتم‌های بهینه‌سازی که عمدتاً تحت عنوان الگوریتم‌های بهینه‌سازی فراابتکاری نام‌گذاری می‌شوند، به دو دسته تقسیم می‌شود. دسته‌ی اول تحقیقات مربوط به کاربرد این الگوریتم‌ها در حل مسائل مختلف اقتصادی و مهندسی است. به صورت عملی نشان داده شده است که این الگوریتم‌ها در کاربردهایی نظیر مدل‌سازی فیلتر [۲]، کنترل سیستم‌ها [۳، ۴]، پردازش تصویر [۵]، [۶]، بازشناسی الگو [۷]، زیست‌سنجی [۸، ۹] و مهندسی قدرت [۱۰-۱۲] در حل مسائل بهینه‌سازی سخت از کارایی لازم برخوردار هستند. دسته‌ی دوم پژوهش‌ها ارائه الگوریتم‌های بهینه‌سازی فراابتکاری جدید است که بتوانند طیف گسترده‌تری از مسائل بهینه‌سازی را با موفقیت حل کنند. با توجه به گستردگی و تنوع این روش‌ها، دسته‌بندی‌های مختلفی از آن‌ها می‌توان داشت که در این مقاله نمی‌گنجد. لیکن در این قسمت، صرفاً مروری مختصر بر آن‌ها خواهیم داشت و یک دسته‌بندی با توجه به قدمت و معروفیت این روش‌ها آورده می‌شود، که این روش‌ها را به سه نسل دسته‌بندی می‌کند.

از نظر تاریخی، الگوریتم وراثتی را می‌توان به عنوان اولین الگوریتم بهینه‌سازی نام برد که با الهام از وراثت طبیعی و اصل تکامل داروین پیشنهاد شده است [۱۳، ۱۴]. از دیگر روش‌های بهینه‌سازی بسیار معروف که می‌توان در کنار الگوریتم وراثتی به عنوان نسل اول دسته‌بندی کرد، الگوریتم‌های کلونی مورچگان [۱۵] و ازدحام ذرات [۱۶] هستند که در زمان خود مسائل بهینه‌سازی زیادی را حل کردند. به عنوان نسل دوم الگوریتم‌های بهینه‌سازی فراابتکاری که عمدتاً متعلق به دهه گذشته هستند، می‌توان به الگوریتم‌های رقابت استعماری [۱۷]، الگوریتم جستجوی گرانشی [۱۸] و الگوریتم جستجوی فاخته [۱۹] اشاره کرد. برای اشاره به اهمیت و فعالیت این زمینه تحقیقاتی، نسل سوم روش‌های بهینه‌سازی ابتکاری به روش‌های ارائه شده در دو سال اخیر اختصاص یافته‌است که از بین آن‌ها می‌توان به الگوریتم بهینه‌سازی شبکه عصبی [۲۰]، الگوریتم کرم خاکی [۲۱] و شب‌پره [۲۲] که در سال ۲۰۱۸ ارائه شده‌اند و همچنین الگوریتم‌های ماهی‌بادبانی [۲۳]، پروانه شهریار [۲۴]، الگوریتم قوش هاوک [۲۵] و تیرانداز خلاقانه [۲۶] در سال ۲۰۱۹ و حتی الگوریتم نسبت طلایی در گیاهان [۲۷] که در سال ۲۰۲۰ ارائه شده‌اند، اشاره کرد. با وجود روش‌های ارائه شده در مقالات، ارائه روش‌های جدید بهینه‌سازی ابتکاری براساس الهام از طبیعت هنوز هم جای کار دارد. از جمله چالش‌های روش‌های ارائه شده تاکنون، می‌توان به این نکته اشاره کرد

که هنوز روشی که بتواند تمامی مسائل را در تمامی انواع آن حل کند وجود ندارد و هر روش فراابتکاری داری کمبودهایی می‌باشد. لذا روش‌هایی که بتوانند پارامترهایی نظیر دقت، سرعت، توان گذر از بهینه‌های محلی و قدرت همگرایی را افزایش دهند می‌توانند بسیار کارا باشند. همچنین به عنوان چالشی بسیار مهم، الگوریتمی که در ابعاد بالا همچنان بتواند به‌طور کارا مسائل را حل کند، می‌تواند بسیار موثر و مفید باشد.

آب و باران و موارد مرتبط با آن از زمره مسائلی است که از منظرهای مختلف می‌توان رویکرد بهینه‌سازی را در آن دید. از چرخه طبیعی آب در طبیعت، ریزش باران و حتی قطراب آب و باران. الگوریتم‌های بهینه‌سازی ابتکاری مختلفی رویکردهای مختلف باران را در بهینه‌سازی در نظر گرفته و روش‌های مختلفی ارائه کرده‌اند. از جمله این روش‌ها می‌توان به الگوریتم بهینه‌سازی ریزش باران [۲۸]، الگوریتم قطره باران [۲۹]، الگوریتم قطرات آب هوشمند [۳۰] و الگوریتم چرخه آب [۳۱] اشاره کرد. با وجود این که هر کدام از این روش‌ها سعی در بهینه‌سازی مسائل مرتبط با آب با رویکرد خودشان دارند، اما وجود روشی که بتواند به‌طور موثر و همه جانبه این پدیده طبیعی را مدل‌سازی کند، خالی است.

بارش باران منجر به ایجاد جریان‌های آب می‌شود. اگر جریان‌های آب باران در مناطق با شیب زیاد جاری شوند، مسیر حرکتشان تا زمانی که شیب منفی است، مستقیم می‌باشد و گرنه مسیر آن‌ها مارپیچ و دائم در جهت شیب منفی (گرانش بیشتر) تغییر خواهد کرد. تندآب زمانی ایجاد می‌شود که مسیر جریان آب به‌طور ناگهانی دچار اختلاف ارتفاع شود. از طرفی، گودال‌های آب پر شده سرریز کرده و جریان‌های آب از یک فاصله مشخص (همسایگی) به گودال‌های سرریز شده، نزدیک نمی‌شوند و به دنبال مسیرهای جدید حرکت می‌کنند.

در این مقاله، براساس این رفتارهای ذکر شده از آب باران، الگوریتم جریان‌های آب ارائه گردیده است که براساس نحوه حرکت و رفتار جریان‌های آب بر روی زمین می‌باشد. در پیاده‌سازی این الگوریتم سعی شده‌است همه ویژگی‌های این پدیده طبیعی طی عملگرهای مختلفی پیاده و به الگوریتم اضافه شوند. برای تایید کارایی این الگوریتم، ۲۳ تابع آزمون استاندارد مورد استفاده قرار گرفته و الگوریتم با برخی از الگوریتم‌های بهینه‌سازی پرکاربرد و متداول مورد مقایسه قرار گرفته است.

ساختار ادامه مقاله به این صورت است که ابتدا در بخش دوم مروری بر الگوریتم‌های الهام گرفته از پدیده‌های مربوط به آب و باران انجام می‌گیرد. در بخش سوم الگوریتم پیشنهادی به همراه عملگرهای مربوطه ارائه می‌شود، در بخش چهارم نتایج شبیه‌سازی و در بخش پنجم نتیجه‌گیری و پیشنهادها ارائه خواهد گردید.

۲- مروری بر الگوریتم‌های الهام گرفته‌شده از رفتارهای آب و باران

الگوریتم‌های متنوع و متفاوتی با الهام از رفتار قطرات باران و آب باران یا چرخه آن، شکل گرفته است. در این بخش، به اختصار الگوریتم‌های مربوطه معرفی می‌شوند.

۲-۱- الگوریتم آب باران

در الگوریتم آب باران هر قطره باران نماینده مقادیر تصادفی از متغیرهای بهینه‌شده که دارای جرم و ارتفاع است، می‌باشد. این قطرات، با سرعتی که متناسب با شتاب گرانشی و زمان طی مسیر است، با پیروی از قوانین حرکت سقوط آزاد، به زمین می‌افتند. حرکت بعدی، حرکت با شتاب ثابت قطرات باران روی زمین در طول مسیر رسیدن به پایین‌ترین نقطه، در نظر گرفته می‌شود. در حقیقت یافتن پایین‌ترین نقطه بر روی زمین، هدف اصلی این الگوریتم می‌باشد [۳۲].

۲-۲- الگوریتم ریزش باران

الگوریتم ریزش باران، حرکت از یک حدس به سمت بهینه سراسری، مانند حرکت قطرات باران از بلندی‌ها به سمت دریاست. قطره باران در این الگوریتم مانند کروموزوم در الگوریتم وراثتی یک ذره از جمعیت می‌باشد. قبل از حرکت هر قطره، نقاط همسایگی قطره بررسی می‌شوند و در صورت وجود نقطه عمیق‌تر، قطره به سمت عمیق‌ترین همسایگی حرکت کرده و آن نقطه مکان جدید قطره خواهد شد [۲۸].

۲-۳- الگوریتم قطره باران

الگوریتم قطره باران با استفاده از الگوریتم راه رفتن تصادفی البته فقط در جهت بردارهای یکه، شروع به جستجوی همسایگی‌های نقاط می‌کند. اندازه حرکت نقاط با مولفه سرعت تعیین می‌شود و در صورت بهبود، نقطه جدید پذیرفته می‌شود [۲۹].

۲-۴- الگوریتم قطرات آب هوشمند

الگوریتم قطرات آب هوشمند دارای دو مشخصه اصلی سرعت و رسوبات می‌باشد. در طول الگوریتم هر دوی آن‌ها ممکن است دچار تغییر شوند. فرض بر این است که جریان از مکان فعلی به مکان بعدی با گام‌های گسسته حرکت می‌کند، به طوری که میزان سرعت به صورت غیرخطی با معکوس میزان رسوبات در رابطه است. الگوریتم قطرات آب هوشمند با بیان مسئله به صورت گراف با N گره و E یال شروع می‌شود. به میزانی که بهترین راه‌حل بهینه باشد، میزان رسوبات یال‌های آن کاهش پیدا می‌کند. در دور بعد، الگوریتم با جریان‌های جدید روال قبلی را تکرار می‌کند، با این تفاوت که دور جدید با میزان رسوبات دور قبل شروع می‌شود. الگوریتم با رسیدن تعداد تکرارهای دور اصلی به بیشترین حد مجاز یا رسیدن جواب نهایی به کیفیت مورد انتظار، پایان

می‌پذیرد [۳۰].

۲-۵- الگوریتم چرخه آب

در الگوریتم چرخه آب بر اساس میزان شایستگی، دسته‌بندی قطرات صورت می‌گیرد. به طوری که شایسته‌ترین قطرات به عنوان رودخانه و دریا انتخاب می‌شوند. که دریا به طور حتم بهترین آن‌ها است. سپس حرکت جویبارها به رودخانه‌ها و حرکت رودخانه‌ها به دریا آغاز می‌شود و در پایان هر حرکت، قطرات دوباره ارزیابی خواهند شد و در صورت تغییر سلسله مراتب شایستگی، جای جویبار و رودخانه و دریا عوض خواهد شد. عملگر تبخیر در این الگوریتم شبیه به عملگر جهش در الگوریتم وراثتی است و نقاط جدید به صورت تصادفی انتخاب می‌شوند و روندی که در بالا در مورد قطرات باران اشاره شد، برای قطرات جدید نیز اعمال می‌شود و این روال تا پایان الگوریتم ادامه خواهد داشت [۳۱].

۲-۶- چالش‌های الگوریتم‌های ازابیه شده مرتبط با آب و باران

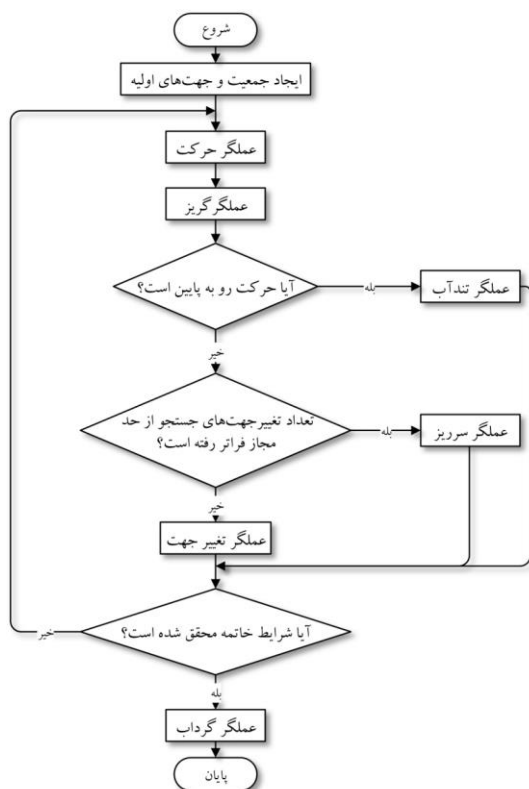
الگوریتم آب باران با تعریف شتاب، سرعت و ارتفاع اولیه و سپس محاسبه موقعیت و پس از آن، محاسبه سرعت و شتاب موقعیت جدید، بسیار شبیه به الگوریتم ازدحام ذرات [۳۳] و الگوریتم جستجوی گرانشی [۱۸]، به سوی نقاط بهینه حرکت می‌کند. تفاوت این روش این است که در هر تکرار، تنها درصد مشخصی (پنج درصد) از قطرات آب به تکرار بعدی الگوریتم می‌روند، که بنابراین می‌توان گفت الگوریتم آب باران یک الگوریتم نخبه‌گرا می‌باشد. الگوریتم ریزش باران با ایجاد تعداد زیادی همسایگی برای هر قطره آب، سعی در جستجوی فضا می‌کند، که حجم محاسبات را بسیار بالا می‌برد و الگوریتم را کند می‌کند. در الگوریتم قطره باران برای هر قطره آب، بردار سرعتی تعریف می‌شود و در صورت عدم بهبود، سرعت نصف می‌شود تا بهینه محلی نزدیک هر قطره پیدا شود. این الگوریتم در فضاهای بزرگ نیاز به تعداد زیاد جمعیت اولیه جهت جستجوی کامل فضا دارد. الگوریتم قطرات آب هوشمند با رسوباتی که جریان‌های آب از خود بجا می‌گذارند، بسیار شبیه به الگوریتم مورچگان [۳۴] رفتار می‌کند و از نظر کاربرد برای مسائل مسیریابی و جایگشتی مناسب می‌باشد. نهایتاً الگوریتم چرخه آب، جهت جستجوی مناسب‌تر به افراز فضا می‌پردازد و نحوه حرکت‌های دسته‌ای آن، نزدیک به الگوریتم رقابت استعماری [۱۷] است. در راستای رفع ایرادات الگوریتم‌های ذکر شده، در بخش بعدی الگوریتم پیشنهادی بهینه‌سازی جریان‌های آب معرفی و عملگرهای آن بررسی می‌شوند.

۳- الگوریتم پیشنهادی بهینه‌سازی فراابتکاری: الگوریتم جریان‌های آب (WSTO)

در این مقاله، الگوریتم جدیدی جهت بهینه‌سازی مسائل پیوسته پیشنهاد شده است، که از نحوه حرکت جریان‌های آب به سمت نقاط

$$X = rand(1, n) \times (High - Low) \quad (1)$$

$$X = (x_1, x_2, \dots, x_n)$$



شکل ۲: نمای الگوریتم بهینه‌سازی جریان‌های آب

به‌طوری‌که X_i یک پاسخ اولیه نام، $rand(1, n)$ تابع تولید یک بردار n تایی (بعد مسئله) از اعداد تصادفی با توزیع یکنواخت، $High$ حد بالای مجاز برای مؤلفه‌های پاسخ اولیه، Low حد پایین مجاز برای مؤلفه‌های پاسخ اولیه و x_i ها مؤلفه‌های پاسخ می‌باشد. مورد بعدی، نحوه تعیین جهت حرکت هر جریان است، که جهت‌های اولیه هر جریان نیز به همان روش تولید جمعیت اولیه ایجاد می‌شود.

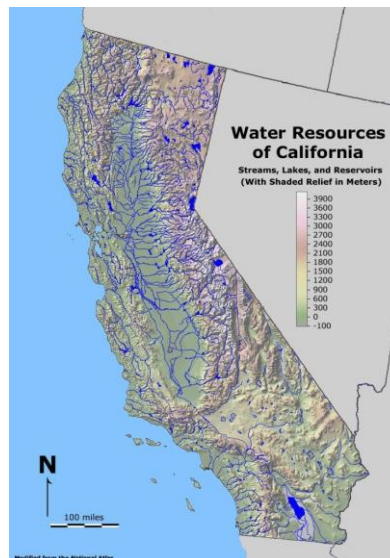
۳-۲- تعریف تابع هزینه

در این قسمت، بر اساس تعریف مسئله تابع هزینه مشخص می‌گردد. در صورتی‌که ماهیت مسئله بهینه‌سازی باشد، با روش‌های ریاضی مثل معکوس یا منفی کردن و غیره، تابع شایستگی را به‌صورت تابع هزینه تعریف می‌کنیم.

۳-۳- عملگر حرکت

پس از تعریف جمعیت اولیه، جریان‌های آب شروع به حرکت در مسیر شیب منفی کرده، این حرکت تا زمانی که شیب منفی بماند ادامه خواهد داشت. در اینجا، هر عضو از جمعیت اولیه به‌عنوان یک جریان آب معرفی می‌شود. بنابراین ما به تعداد جمعیت اولیه، جریان‌های آب داریم که شروع به حرکت کرده و فضا را جستجو می‌کنند. اگر جمعیت اولیه را بردارهایی در فضای پاسخ در نظر

پست‌تر الهام گرفته‌شده است. در شکل (۱)، نمایی از حرکت جریان‌های آب آمده‌است. بارش باران و آب شدن برف در ارتفاعات منجر به ایجاد جریان‌های آب می‌شود. اگر جریان‌های آب در مناطقی با شیب زیاد جاری شوند، مسیر حرکتشان تا زمانی که شیب منفی است، مستقیم می‌باشد و گرنه مسیر آن‌ها مارپیچ و دائم در جهت شیب منفی (گرانش بیشتر) تغییر خواهد کرد. تندآب زمانی ایجاد می‌شود که مسیر جریان آب به‌طور ناگهانی دچار اختلاف ارتفاع شود.



شکل ۱: نمایی از حرکت جریان‌های آب حاصل از باران و آب شدن برف به سمت اقیانوس (بهینه‌سازی) و ایجاد دریاچه‌ها و گودال‌های آب (بهینه‌های محلی) در مسیر^۱

از طرفی گودال‌های آب پر شده (بهینه‌های محلی)، سرریز کرده و جریان‌های آب از یک‌فاصله مشخص (همسایگی) به گودال‌های آب، نزدیک نمی‌شوند و به دنبال مسیرهای جدید حرکت می‌کنند و نهایتاً گرداب برای جریانی که در همسایگی عمیق‌ترین گودال قرار گرفته، شکل می‌گیرد (روند نمای الگوریتم بهینه‌سازی جریان‌های آب را در شکل (۲) نشان داده‌شده است). در زیربخش‌های آتی هر کدام از قسمت‌های الگوریتم به تفکیک معرفی می‌شوند.

۳-۱- تعریف جمعیت اولیه

اگر وسعت ابرهای بارانی را نسبت به وسعت زمین به‌اندازه کافی بزرگ فرض کنیم، می‌توان گفت نحوه ریزش باران بر روی زمین به‌صورت تصادفی با توزیع یکنواخت است که در واقع همان نحوه تعریف جمعیت اولیه در بیشتر الگوریتم‌های فراابتکاری می‌باشد. در الگوریتم بهینه‌سازی جریان‌های آب نیز جمعیت اولیه به همین روال تولید می‌شود. به‌صورت تصادفی قطرات باران در فضای پاسخ پراکنده می‌شوند. حرکت هر قطره نماینده یک جریان آب است. این مجموعه جواب را جمعیت اولیه می‌نامیم. هر جواب را یک جریان آب می‌نامیم، که به‌صورت رابطه (۱) قابل تعریف است.

$$\Delta = \frac{(F(X_{new}) - F(X_{old}))}{\text{norm}_2(X_{new} - X_{old})} \times \beta \times \text{randn}, \quad (3)$$

$$\alpha_{new} = \alpha_{old} \times \Delta,$$

درجایی که، Δ ضریب تغییر، $F(X)$ تابع شایستگی در بردار X ، β ضریب کنترل جستجو، α ضریب وزنی بردار جهت و مکان، norm_2 تابع محاسبه فاصله اقلیدسی (نرم دو) و randn عددی است تصادفی با توزیع گوسی و میانگین صفر و واریانس یک می‌باشد.

۳-۶- عملگر تغییر جهت^۵

در صورتی که جریان آب طبق رابطه (۲) حرکت کند و بهبودی نداشته باشد، جهت حرکت تغییر پیدا می‌کند. تعداد این تغییر جهت‌ها محدود و از متغیرهای قابل تنظیم الگوریتم می‌باشد. مشخص است که هرچقدر این تعداد بیشتر شود، کاوش محلی در اطراف هر جریان بهتر انجام می‌شود، اما حجم محاسبات بالا می‌رود. نحوه اعمال این عملگر به این صورت است که تمامی جریان‌هایی که در محدوده همسایگی این جریان قرار دارند (نحوه تعریف همسایگی و اندازه شعاع در سرتاسر الگوریتم برای تمامی عملگرها یکسان در نظر گرفته شده است)، بررسی و شایسته‌ترین آن‌ها انتخاب می‌شود و به‌عنوان جهت جدید جریان، مورد استفاده قرار می‌گیرد. یعنی جهت جریان به سمت پایین‌ترین نقطه موجود در همسایگی تغییر داده می‌شود. البته این عملگر در تکرارهای انتهایی الگوریتم (که در الگوریتم بیست درصد بیشینه تکرارها در نظر گرفته شده است)، جهت بالا بردن همگرایی مورد نیاز، جریان‌ها را به سمت میانگین بهترین جریان‌های تکرارهای قبلی تغییر جهت می‌دهد (استفاده از حافظه الگوریتم).

۳-۷- عملگر سرریز^۶

در صورتی که تعداد جهت‌های مورد بررسی برای یک جریان از حد مذکور فراتر رود و بهبودی حاصل نشود، می‌توان گفت که جریان در یک گودال (بهینه محلی) گیر کرده است. لذا برای فرار جریان از این موقعیت، عملگر سرریز تعریف می‌شود. به این صورت که موقعیت جریان (بردار مکان) به مکانی بیرون از گودال تغییر داده می‌شود. شعاع گودال به اندازه شعاع همسایگی قرار داده می‌شود.

۳-۸- عملگر گرداب^۷

در نهایت، با رسیدن جریان به همسایگی عمیق‌ترین نقطه (بهینه سراسری)، گرداب شکل می‌گیرد. یعنی جریان از حرکت بازمی‌ایستد و شروع به بهبودهای جزئی حول بهینه سراسری می‌کند. این پدیده در الگوریتم به این صورت در نظر گرفته شده است که در تکرار نهایی الگوریتم، بهترین جریان همه تکرارها مشخص می‌شود و تغییرات به صورت مؤلفه‌ای، برای هر بعد از این جریان بررسی می‌شود (بهره‌گیری نهایی). میزان تفکیک‌پذیری و در نتیجه آن دقت جواب

بگیریم، جهت‌ها نیز به صورت بردارهایی هم‌بعد با جریان‌ها تعریف می‌شوند و مکان جدید جریان در ابتدای حرکت هر جریان، بر اساس میانگین حسابی دو بردار (۲) مکان و جهت تعیین می‌گردد. در صورتی که حرکت پذیرفته می‌شود که مکان جدید پایین‌تر از مکان قبلی باشد.

$$X_{new} = X_{old} \cdot (1 - \alpha) + \text{direction} \cdot \alpha \quad (2)$$

به طوری که در رابطه (۲)، X بردار مکان، α ضریب وزنی بردار جهت و مکان و direction بردار جهت می‌باشد.

۳-۴- عملگر گریز^۳

پس از ایجاد موقعیت جدید جریان، در هر نقطه از الگوریتم عملگرهای حرکت و تندآب و قبل از محاسبه شایستگی، عملگر گریز اعمال می‌شود. عملگر گریز فهرستی از گودال‌های آب (بهینه‌های محلی) را در طول اجرای الگوریتم، در خود نگهداری می‌کند و در صورتی که موقعیت جدید در همسایگی بهینه‌های محلی قرار داشته باشد از پذیرفتن آن خودداری می‌کند و این موقعیت را به بیرون از همسایگی انتقال می‌دهد. این عملگر، از تکرار جستجوی ناحیه‌هایی که پیش‌تر مورد بررسی قرار گرفته‌اند، جلوگیری می‌کند. مؤلفه‌های تنظیمی در این عملگر حداکثر تعداد بهینه‌های محلی که ذخیره می‌شود و شعاع همسایگی آن‌ها است. این شعاع همسایگی توسط خود الگوریتم کنترل می‌شود. بدین صورت که در تکرارهای ابتدایی الگوریتم، جهت افزایش قابلیت کاوش، شعاع بزرگ در نظر گرفته می‌شود و به تدریج کاهش پیدا می‌کند (افزایش بهره‌گیری). اما تعداد حداکثر بهینه‌های محلی قابل تنظیم می‌باشد، به طوری که افزایش این مؤلفه، افزایش تنوع و قدرت کاوش الگوریتم را موجب می‌شود.

۳-۵- عملگر تندآب^۴

هر جریانی که در شیب منفی قرار گیرد و در یک جهت، بهبود متوالی داشته باشد، تبدیل به تندآب می‌شود. بدین معنی که اگر حرکت رو به پایین بود، عملگر حرکت در همان جهت تکرار می‌شود. ولی قبل از تکرار، وزن بردار جهت نسبت به وزن بردار جریان آب، بیشتر می‌شود. که این تغییر در وزن، متناسب با میزان بهبود شایستگی، نسبت به تغییرات بردار مکان می‌باشد (تعریف شیب). به طوری که هر چه قدر این نسبت بزرگ‌تر باشد، وزن بردار جهت نسبت به بردار مکان بیشتر می‌شود. البته در الگوریتم، جهت کنترل سرعت همگرایی و قدرت جستجو، یک ضریب در نظر گرفته شده است که می‌تواند بین یک تا هزار باشد، هر چه عدد بزرگ‌تر باشد، همگرایی کمتر و کاوش بهتری انجام می‌شود. جهت بهبود کاوش یک ضریب دیگر، که عددی است تصادفی با توزیع گوسی و میانگین صفر و واریانس یک، اعمال شده است که در (۳) قابل مشاهده است.

۴-۱ شرایط آزمون و معیارهای ارزیابی

پس از معرفی توابع آزمون استاندارد، به تعیین شرایط آزمون و معیارهای ارزیابی پرداخته می‌شود. در بررسی اولیه این توابع بعد از ۳۰ برای توابع آزمون استاندارد در نظر گرفته شده است (توابع چهارده تا بیست و سه (جدول ۳) بر اساس نوع تعریف، هر کدام دارای بعد مشخص هستند) تعداد جمعیت اولیه ۵۰ در نظر گرفته شده و تمامی الگوریتم‌های مورد بررسی در برنامه متلب نوشته و اجرا شده است. آزمون توسط یک رایانه با مشخصات: پردازنده Intel Core(TM) i5-5450M CPU @ 2.50GHz، حافظه RAM ۴ گیگابایت و سیستم عامل ویندوز ۱۰ و نرم افزار MATLAB نسخه (R2016a) اجرا شده است. برنامه الگوریتم وراثتی حقیقی با عملگر تزویج از نوع جمع حسابی و احتمال تزویج ۰/۳ و احتمال جهش ۰/۱ اجرا شده است و جهش گوسی و انتخاب چرخ گردان مورد استفاده قرار گرفته است. الگوریتم ازدحام ذرات [۱۶] با ضریب ۲ برای بهترین موقعیت خود و بهترین موقعیت گروه و ضریب ۰/۲ برای کمترین اینرسی و ضریب ۰/۹ برای بیشترین اینرسی و حداکثر سرعت برابر با یک سوم اختلاف دامنه، مورد آزمون قرار گرفت است. در الگوریتم جریان‌های آب مولفه β در رابطه (۳) برابر با هفتصد و تعداد گودال‌های آب، پنج (بخش ۳-۴) و بیشینه تعداد تغییر جهت برای هر جریان ده‌بار (بخش ۳-۶) در نظر گرفته شد.

جهت بررسی هر یک از روش‌های بهینه‌سازی، برای هر تابع آزمون استاندارد، الگوریتم مربوطه ۱۰۰ بار اجرا خواهد شد و نتایج آزمون با هر تابع، به صورت سه مؤلفه (۱) میانگین بهترین تاکنون، (۲) میانگین بهترین تاکنون^{۱۱} و (۳) انحراف معیار بهترین تاکنون^{۱۱} ارائه خواهد شد.

نیز قابل کنترل در نظر گرفته شده است. بدیهی است که بالا بردن دقت، حجم محاسبات را افزایش می‌دهد و بالعکس.

۴- نتایج آزمون

جهت بررسی و مقایسه الگوریتم‌های بهینه‌سازی توابع استاندارد ارائه شده است که به وسیله آن‌ها می‌توان هر روش بهینه‌سازی را مورد بررسی قرار داد. در جدول (۱) روابط، دامنه و بُعد برای توابع آزمون تک بهینه^۸ تعریف شده است. این دسته از توابع تنها یک نقطه که همان بهینه سراسری است به عنوان نقطه بهینه دیده می‌شود. بنابراین، در بررسی این توابع نرخ همگرایی بیشتر مدنظر است تا نتیجه نهایی. چراکه جهت بهینه‌یابی این دست از مسائل، روش‌های کلاسیک به اندازه کافی دقیق و درست هستند و نیازی به استفاده از روش‌های فراابتکاری نیست. جدول (۲) اطلاعات مربوط به توابع آزمون چند بهینه با بعد بالا را در بر دارد. با توجه به این که این توابع دارای بهینه‌های محلی زیادی هستند، بهینه‌یابی آن‌ها مشکل می‌باشد. در بررسی این توابع، پاسخ نهایی از آنجاکه مشخص کننده توان الگوریتم در فرار از بهینه‌های محلی و رسیدن به جوابی نزدیک به بهینه سراسری است، بسیار مهم می‌باشد. این در شرایطی است که افزایش بعد برای این توابع، تعداد بهینه‌های محلی را به طور تصاعدی افزایش می‌دهد. وجود بهینه‌های محلی متعدد در دامنه این دست از توابع، توان الگوریتم را در یک گذر از بهینه‌های محلی می‌سنگد، که با روش‌های کلاسیک بهینه‌یابی غیرممکن می‌باشد. در ادامه جدول (۳) که دربرگیرنده اطلاعات مربوط به توابع آزمون چندبهینه با بعد ثابت است، مشاهده می‌شود. که پیچیدگی و متعدد بودن نقاط بهینه محلی در این توابع، در نظر گرفته شده است.

جدول ۱: توابع آزمون تک بهینه

تابع	دامنه
$F_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]^n$
$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]^n$
$F_4(x) = \max\{ x_i , 1 \leq i \leq n\}$	$[-100, 100]^n$
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^n$
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	$[-100, 100]^n$
$F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	$[-1.28, 1.28]^n$

جدول ۲: توابع آزمون چندبهمینه با بعد متغیر

تابع	دامنه
$F_8(x) = \sum_{i=1}^n (-x_i \sin(\sqrt{ x_i }))$	$[-500,500]^n$
$F_9(x) = \sum_{i=1}^{n-1} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]^n$
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32,32]^n$
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 + \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-50,50]^n$
$F_{12}(x) = \frac{\pi}{n} \left\{ 10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	
$y_i = 1 + \frac{x_i + 1}{4}$	$[-50,50]^n$
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	
$F_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	$[-50,50]^n$

نتایج به فرآیند می‌تواند تأثیرگذار باشد. بنابراین پس از آن به بررسی مدت‌زمان اجرای هر تابع توسط هر الگوریتم پرداخته می‌شود. در ادامه، آزمون بعدی با افزایش بعد برای توابع آزمون یک تا سیزده انجام خواهد شد. که هدف از آن بررسی و مقایسه رفتار الگوریتم ارائه‌شده با دیگر الگوریتم‌های مورد آزمون، در ارتباط با مسائلی که متغیرهای متعدد تأثیرگذار زیادی، در معادلات آن‌ها وجود دارد،

بررسی مؤلفه‌های مذکور، دقت الگوریتم‌های بهینه‌سازی را در رسیدن به بهینه‌های سراسری را مورد بررسی قرار می‌دهد، اما مؤلفه دیگری که باید موردنظر قرار گیرد مؤلفه زمان است. در برخی از کاربردها مؤلفه زمان نقش بسیار مهمی را ایفا می‌کند، به‌ویژه در فرآیندهایی که از سرعت بالایی برخوردار هستند یا فاقد ناظر انسانی هستند و فرآیند دریافت اطلاعات و بازگرداندن نتیجه به فرآیند به‌طور خودکار انجام می‌شود، تحلیل سریع اطلاعات و بازگرداندن

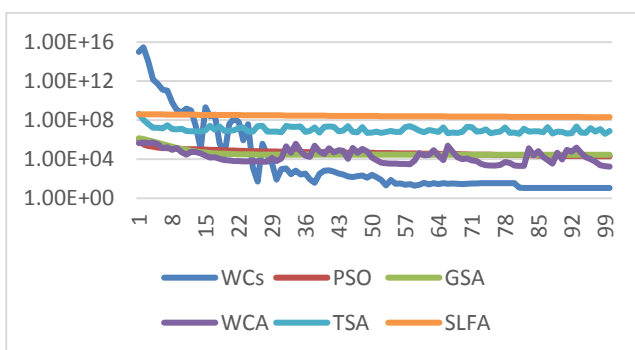
جدول ۳: توابع آزمون چند بهمینه با بعد ثابت

تابع	دامنه
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	$[-65.53,65.53]^2$
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	$[-5,5]^4$
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]^2$
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	$[-5,15]^2$
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \times \left[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right]$	$[-5,5]^2$
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	$[0,1]^3$
$F_{20}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right)$	$[0,1]^6$
$F_{21}(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0,10]^4$
$F_{22}(x) = -\sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0,10]^4$
$F_{23}(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0,10]^4$

مشاهده کرد. در ادامه در جدول (۶) نتایج آزمون برای توابع آزمون چهارده تا بیست و سه مشاهده می‌شود، که این توابع دارای بهینه‌های محلی متعدد هستند اما با بعد ثابت. نقاط بهینه سراسری و مقدار شایستگی هر تابع را می‌توان در جدول (۷) مشاهده کرد. که الگوریتم ارائه شده در بدست آوردن بهینه‌های سراسری عملکرد قابل قبولی در مقایسه با دیگر الگوریتم‌ها از خود نشان می‌دهد.

۳-۴ مقایسه نتایج از نظر مدت زمان اجرا

شرایط اجرای برنامه همان شرایط آزمون پیشین است با این تفاوت که هر الگوریتم تنها یک بار برای هر تابع آزمون اجرا گردیده و میانگین، میان و انحراف معیار زمان اجرای برنامه محاسبه شده است. نتایج این آزمون در جدول (۸) قابل مشاهده است. الگوریتم بهینه‌سازی جریان‌های آب، پس از الگوریتم وراثتی و الگوریتم جستجوی فاخته رتبه سوم را از نظر زمان اجرا داراست. شاید مقایسه اگر هم‌زمان به دو مؤلفه دقت و زمان توجه داشته باشد، نتایج معنی‌دارتر باشند. لذا در جدول (۹) به بررسی خطای نسبی بین الگوریتم‌های وراثتی و الگوریتم جستجوی فاخته با الگوریتم ارائه‌شده برای پنج تابع اول پرداخته می‌شود و در جدول (۱۰) همین نوع مقایسه از نظر زمانی انجام شده است. میانگین زمان اجرای برنامه الگوریتم بهینه‌سازی جریان‌های آب نسبت به الگوریتم وراثتی ۱/۶۵۷ و نسبت به الگوریتم جستجوی فاخته ۱/۲۷۴ می‌باشد. درجایی که میانگین خطای الگوریتم ارائه‌شده به الگوریتم وراثتی ۰/۰۶ و نسبت به الگوریتم جستجوی فاخته ۰/۰۴ می‌باشد. بنابراین افزایش ۹۴٪ دقت در ازای ۶۵٪ درصد زمان بیشتر نسبت به الگوریتم وراثتی و افزایش ۹۹/۶٪ دقت در ازای ۲۷/۵٪ زمان بیشتر نسبت به الگوریتم جستجوی فاخته مشاهده می‌گردد.



شکل ۳: مقایسه همگرایی الگوریتم بهینه‌سازی جریان‌های آب، الگوریتم ازدحام ذرات، الگوریتم جستجوی گرانشی و الگوریتم چرخه آب در کمینه‌یابی بر روی تابع آزمون شماره ۳ با بعد ۳۰

می‌باشد. با توجه به این‌که در دسته دوم توابع آزمون (توابع هشت تا سیزده) تعداد بهینه‌های سراسری با افزایش بعد، به صورت تصاعدی افزایش می‌یابد، این آزمون توان الگوریتم‌ها را در برخورد با این دست از مسائل به چالش می‌کشد.

۴-۲ مقایسه نتایج از نظر دقت جواب

با اعمال الگوریتم ارائه‌شده بر توابع استاندارد^{۱۳} و مقایسه با الگوریتم وراثتی حقیقی [۳۵]، الگوریتم جستجوی گرانشی [۳۶]، الگوریتم ازدحام ذرات [۳۷]، الگوریتم جستجوی فاخته [۳۸]، الگوریتم آبدزدک دریایی [۳۹]، الگوریتم جهش قورباغه [۴۰] و الگوریتم چرخه آب که در شرایط برابر اجرا شده‌اند، به بررسی عملکرد این الگوریتم پرداخته شده است. جدول (۴) نتایج آزمون برای هفت تابع استاندارد اول را نشان می‌دهد که این توابع دارای تنها یک بهینه سراسری هستند و هدف از این آزمون بررسی توان همگرایی الگوریتم می‌باشد. نقطه بهینه سراسری تمامی توابع آزمون تک بهینه به غیر از تابع آزمون شماره پنج، بردار 0^n و عضوی از فضای R^n می‌باشد که در آن، n بعد فضا است و برای تابع آزمون شماره پنج بردار 1^n بهینه سراسری تعریف شده است. در آزمون برای تمامی توابع مورد بررسی بردار بهینه سراسری به اندازه ۰.۷ جابجا شده است. میزان شایستگی نقطه بهینه برای توابع آزمون جدول (۴) برابر با صفر می‌باشد. نتایج ارائه‌شده با صد بار اجرای هر الگوریتم استخراج گردیده است و سه مؤلفه میانگین بهترین تاکنون و میان بهترین تاکنون و انحراف معیار بهترین تاکنون مورد نظر بوده است. جدول (۴) مشاهده می‌شود به طور مشخص میزان همگرایی الگوریتم بهینه‌سازی جریان‌های آب در توابع آزمون استاندارد شماره ۳ و ۴ و ۵ و ۶ بالاتر از دیگر الگوریتم‌هاست. نتایج بررسی الگوریتم‌های بهینه‌سازی برای توابع با بهینه‌های متعدد محلی و بعد بالا را می‌توان در جدول (۵) مشاهده کرد. نقطه بهینه سراسری برای تابع آزمون شماره هشت بردار $1^n [۴۲۰.۹۶]$ به طوری که n بعد فضا در نظر گرفته می‌شود و برای توابع آزمون شماره دوازده و سیزده بردار 1^n بهینه سراسری در نظر گرفته شده است و مابقی توابع آزمون جدول (۵) در بردار 0^n بهینه می‌شوند. که در آزمون برای تمامی توابع مورد بررسی بردار بهینه سراسری به اندازه ۰.۷ جابجا شده است. میزان شایستگی نقطه بهینه برای توابع آزمون جدول (۵) به استثنای تابع آزمون شماره هشت برابر با صفر می‌باشد و میزان شایستگی تابع آزمون شماره هشت n برابر عدد $۴۱۸/۹۸۲۹-$ می‌باشد که در آن n بعد فضا می‌باشد. در این بررسی می‌توان برتری مشخص الگوریتم ارائه‌شده را در توابع آزمون استاندارد شماره ۹، ۱۱، ۱۲ و ۱۳

جدول ۴: نتایج آزمون برای توابع آزمون یک تا هفت

	WSTO	TSA	SLFA	RGA	GSA	PSO	Cuckoo	WCA	
	۲.۲۴E-۱۰	۸.۶۲E-۵۲	۱.۷۹E-۰۷	۴.۹۳۰۸	۳.۸۳E-۱۷	۰.۰۰۱	۶۴۰.۶۳۹	۱.۴۰E-۰۶	میانگین
1	۲.۲۳E-۱۰	۲.۴۴E-۵۳	۱.۰۵E-۰۷	۴.۷۷۵۵	۳.۶۲E-۱۷	۰.۰۰۰۶	۶۱۰.۳۹۹۳	۱.۰۴E-۰۷	میانه
	۳.۲۸E-۱۱	۶.۰۱E-۵۱	۲.۱۵E-۰۷	۱.۱۵۰۹	۱.۲۶E-۱۷	۰.۰۰۱۱	۱۷۹.۶۸۶۷	۸.۶۸E-۰۶	انحراف
	۱.۴۷E-۰۵	۴.۲۰E-۳۱	۱.۱۳E-۰۴	۲.۳۵۸۹	۳.۱۱E-۰۸	۴.۳۰۸۵	۵۵.۴۴۷۸	۰.۶۰۰۴	میانگین
2	۱.۴۹E-۰۵	۱.۰۷E-۳۱	۷.۶۵E-۰۵	۱.۹۵۷۶	۳.۰۷E-۰۸	۰.۰۰۰۹	۵۵.۷۶۷۶	۰.۰۰۰۱	میانه
	۱.۳۵E-۰۶	۹.۳۹E-۳۱	۹.۵۹E-۰۵	۱.۰۴۳	۵.۸۹E-۰۹	۶.۸۵۱۵	۱۰.۴۲۶	۲.۸۳۵۳	انحراف
	۳.۹۲۱۱	۳.۰۷E-۱۵	۲۵۴.۹۱	۱۹۴۴.۱۶	۶۰.۶۵۹۸	۵۲۲۰.۵	۹۳۵۰.۱۶	۶۵.۴۴۴۳	میانگین
3	۰.۹۴۰۹	۳.۲۵E-۱۹	۲۰۷.۰۹۴	۱۸۱۰.۳۱	۵۹۱.۴۳۱	۳۵۱۵.۳۰۶	۹۲۳۰.۶۳۵۲	۱۲.۲۸۲۳	میانه
	۶.۵۰۸۷	۲.۱۵E-۱۴	۱۷۷.۳۰۹	۱۴۷۱.۹۵	۱۸۸.۴۴۴	۴۱۸۷.۳۲۷	۱۷۴۵.۲۷۹	۴۹۵.۳۲۹	انحراف
	۰.۹۲۰۷	۰.۰۰۰۱۷۸۴	۰.۹۲۳۶	۱.۷۶۷۲	۳.۲۲۰۹	۹.۶۷۱۵	۲۵.۶۵۳۸	۱۰.۶۳۴۱	میانگین
4	۰.۸۶۵۲	۵.۲۳E-۰۵	۰.۸۴۳	۱.۳۲۵۵	۲.۹۷۹۱	۹.۳۳۵۷	۲۵.۷۴۶۵	۱۰.۲۸۶۹	میانه
	۰.۴۳۶۹	۰.۰۰۰۴	۰.۳۵۶۶	۲.۸۹۲۲	۱.۴۶۷۴	۲.۲۷۹۱	۲.۵۷۵۷	۳.۷۳۸۵	انحراف
	۸.۷۵۷۲	۲۸.۴۱۱	۵۱.۱۹۴۴	۱۰۴.۱۳۶	۳۶.۹۸۷۵	۹۴۶۳.۶۰۶	۸۵۱۵۶.۶۹۹	۷۳.۴۸۲۷	میانگین
5	۰.۴۷۲۹	۲۸.۷۴۲۶	۲۷.۷۶۲۳	۷۵.۱۳۶۹	۲۷.۵۲۵۹	۱۰۵.۱۴۰۸	۷۴۴۰.۸۵۹	۷۵.۷۳۵۱	میانه
	۱۲.۷۷۹۲	۰.۶۹۹۴	۳۷.۱۹۸۵	۱۳۰.۹۹۷	۴۴.۱۶۸۴	۲۷۰.۱۹۶	۴۴۶۶۷.۳۷۴	۵۷.۱۲۲۳	انحراف
	.	۰.۰۴	۰.۹۱	۶.۴۸	۰.۴۹	۰.۲۸	۷۵۸.۱۳	۷.۲۲	میانگین
6	.	.	۱	۶	.	.	۷۵۳.۵	۶	میانه
	.	۰.۳۱۵۳	۱.۲۲۳۵	۱.۷۶۶۶	۱.۲۴۳۱	۰.۵۳۳۳	۲۱۰.۷۰۰۷	۵.۶۶۵۶	انحراف
	۰.۲۷۳۹	۰.۰۰۳۱	۰.۰۲۵۲	۰.۰۱۴	۰.۱۱۸۱	۰.۰۵۷۷	۰.۴۳۳۱	۰.۰۸۸۴	میانگین
7	۰.۰۲۲۱	۰.۰۰۲۷	۰.۰۲۳۳	۰.۰۱۳۶	۰.۰۹۶۲	۰.۰۵۴۹	۰.۴۱۲	۰.۰۷۸۳	میانه
	۰.۸۲۲۹	۰.۰۰۱۵	۰.۰۰۹۹	۰.۰۰۵۱	۰.۰۶۵۷	۰.۰۱۹۴	۰.۱۹۴۶	۰.۰۵۱	انحراف

جدول ۵: نتایج آزمون برای توابع آزمون هشت تا سیزده

	WSTO	TSA	SLFA	RGA	GSA	PSO	Cuckoo	WCA	
	-۸۶۸۸.۵۵	-۶۲۴۹.۵۳	-۴۸۳۶.۲۸	-۳۵۶۴.۲۹	-۲۷۸۰.۵۹	-۹۷۴۴.۳۵	-۷۶۷۳.۰۲۴	-۸۴۰۶.۱۶	میانگین
8	-۸۸۱۶.۵۷	-۶۳۱۲.۲۹	-۴۵۵۳.۰۲	-۳۵۲۹.۱۳	-۲۷۶۹	-۹۷۷۲.۶۲	-۷۶۸۴.۷۵۷	-۸۳۴۰.۹	میانه
	۹۲۷.۰۴۳۸	۶۱۰.۶۳۷	۹۱۲.۷۵۳۶	۴۷۲.۸۱۸	۳۶۷.۷۴۲۹	۵۸۶.۷۰۲۶	۲۰۱.۷۳۸	۸۳۱.۳۲۶	انحراف
	۱.۹۸۹۹	۱۶۷.۹۱۸	۷۰.۴۹۲۷	۸۹.۱۰۴۷	۱۷.۹۰۹۳	۵۹.۹۸۷۴	۱۴۷.۸۹۲۲	۸۱.۹۶۵۱	میانگین
9	۴.۹۴E-۱۰	۱۷۲.۴۵	۶۹.۱۵۰۷	۸۸.۱۳۶۴	۱۷.۹۰۹۳	۵۹.۴۴۵۸	۱۴۷.۶۶۱	۷۹.۵۹۶۴	میانه
	۷.۵۷E+۰۰	۳۷.۹۸۹	۱۸.۷۶۶۶	۲۹.۲۵۸۷	۴.۷۷۰۶	۱۸.۴۵۰۹	۱۲.۵۰۷۷	۳۰.۶۴۱۱	انحراف
	۷.۰۸E-۰۶	۲.۱۳۳	۰.۳۲۰۲	۱۷.۹۴۹۹	۴.۸۲E-۰۹	۰.۳۲۶۴	۱۱.۴۴۹	۳.۵۶۱۹	میانگین
10	۷.۰۲E-۰۶	۲.۹۸۱۱	۰.۰۰۰۱	۱۸.۰۵۸۹	۴.۷۶E-۰۹	۰.۰۱۷۳	۱۱.۵۰۶۶	۳.۰۲۸۳	میانه
	۵.۴۵E-۰۷	۱.۴۹۵۲	۰.۵۴۸۷	۰.۷۴۴۲	۷.۱۸E-۱۰	۰.۶۱۰۹	۰.۹۸۱۷	۲.۱۸۱۲	انحراف
	۰.۰۰۲۴۲	۰.۰۰۵۱	۰.۰۱۰۲	۱.۲۱۷۹	۱۷.۵۶	۰.۰۱۷۸	۶.۹۳۹۲	۰.۰۲۵	میانگین
11	۹.۷۰E-۰۶	.	۰.۰۰۷۴	۱.۲۱۸	۱۷.۵۵۷۹	۰.۰۱۱	۶.۴۸۹۶	۰.۰۱۷۲	میانه
	۰.۰۰۸۲	۰.۰۰۶۴	۰.۰۱۲۳	۰.۰۴۹۴	۴.۱۱۶۴	۰.۰۲۲۹	۱.۶۱۶۳	۰.۰۲۶۴	انحراف
	۶.۷۰E-۱۲	۷.۱۹۸	۰.۱۳	۰.۰۰۷۹	۰.۵۰۳۵	۰.۳۶۹۱	۱۷.۲۱۷۱	۰.۳۹۱	میانگین
12	۴.۴۹E-۱۲	۶.۸۳۹۱	۵.۸۷E-۰۷	۰.۰۰۷۶	۰.۳۷۰۷	۰.۲۰۷۵	۱۶.۷۹۹۴	۰.۰۰۹۹	میانه
	۴.۹۰E-۱۲	۳.۴۲۰۳	۰.۲۷۴۹	۰.۰۰۲	۰.۴۸۳۹	۰.۴۹۵	۴.۴۰۶۶	۰.۸۷۸۲	انحراف
	۰.۰۰۱۸	۲.۹۱۱	۰.۰۰۴۵	۰.۲۳۱۹	۲.۱۲۱۶	۰.۱۵۹۳	۵۷۹۲.۲۴۸	۰.۱۷۶۶	میانگین
13	۱.۰۵E-۱۰	۲.۸۵۸۳	۲.۴۹E-۰۷	۰.۲۲۳۹	۰.۶۶۵۴	۰.۰۴۸۸	۲۷۵۰.۳۵	۱.۶۱E-۰۵	میانه
	۰.۰۰۴۲	۰.۶۸۲۵	۰.۰۰۸۶	۰.۰۵۲۱	۲.۸۲۷	۰.۳۱۳۸	۶۸۱۱.۴۸۲	۱.۶۷۹۳	انحراف

جدول ۶: نتایج آزمون برای توابع چهارده تا بیست و سه

WSTO	TSA	SLFA	RGA	GSA	PSO	Cuckoo	WCA	
۱.۴۸۳۸	۶.۹۶۰۹	۱.۲۴۵۷	۰.۹۹۸	۴.۶۹۸۹	۰.۹۹۸	۰.۹۹۸	۰.۹۹۸	میانگین
۰.۹۹۸	۳.۹۶۸۳	۰.۹۹۸	۰.۹۹۸	۴.۰۹۸۵	۰.۹۹۸	۰.۹۹۸	۰.۹۹۸	میانه
۰.۹۴۸۵	۴.۹۵۳۸	۰.۷۶۱۸	۴.۱۰E-۰۹	۲.۸۹۳۸	۲.۵۸E-۱۵	۴.۱۷E-۰۹	۱.۷۹E-۱۵	انحراف
۰.۰۰۰۹	۰.۰۰۶۵	۰.۰۰۱۲	۰.۰۰۱۳	۰.۰۰۴۶	۰.۰۰۰۳	۰.۰۰۰۷	۰.۰۰۱۶	میانگین
۰.۰۰۰۷	۰.۰۰۰۵	۰.۰۰۰۳	۰.۰۰۱۴	۰.۰۰۳۹	۰.۰۰۰۷	۰.۰۰۰۷	۰.۰۰۰۸	میانه
۰.۰۰۰۶	۰.۰۱۱۹	۰.۰۰۳۹	۰.۰۰۲۶	۰.۰۰۲۸	۰.۰۰۶۲	۰.۰۰۰۱	۰.۰۰۳۵	انحراف
-۱.۰۰۳۱	-۱.۰۲۹۷	-۱.۰۰۳۱۶	-۱.۰۱۳۲	-۱.۰۰۳۱۶	-۱.۰۰۳۱۶	-۱.۰۰۳۱۶	-۱.۰۰۳۱۶	میانگین
-۱.۰۰۳۱	-۱.۰۰۳۱۶	-۱.۰۰۳۱۶	-۱.۰۰۲۰۸	-۱.۰۰۳۱۶	-۱.۰۰۳۱۶	-۱.۰۰۳۱۶	-۱.۰۰۳۱۶	میانه
۵.۲۷E-۱۳	۰.۰۰۷۵	۱.۵۶E-۱۵	۰.۰۰۲۹	۱.۳۷E-۱۵	۱.۵۴E-۱۵	۱.۶۹E-۱۲	۱.۴۱E-۱۵	انحراف
۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	میانگین
۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	۰.۳۹۷۹	میانه
۰.۰۰۰۲	۲.۵۳E-۰۵	۱.۰۶E-۱۵	۰.۴۲۳۱	۱.۰۶E-۱۵	۱.۰۶E-۱۵	۷.۸۰E-۰۹	۱.۰۶E-۱۵	انحراف
۳	۶.۵۱	۳	۴۶.۸۴۹۴	۳	۳	۳	۳	میانگین
۳	۳	۳	۶۹.۰۸۵۸	۳	۳	۳	۳	میانه
۸.۳۲E-۰۶	۱۴۶۹۱	۸.۳۴E-۱۶	۵۱.۴۶۶۲	۵.۱۲E-۱۵	۹.۸۲E-۱۶	۲.۰۷E-۱۱	۴.۷۵E-۱۵	انحراف
-۳.۸۵۶	-۳.۸۰۵۱	-۳.۸۶۲۸	-۳.۵۳۱۸	-۳.۸۶۲۸	-۳.۸۶۲۷	-۰.۹۸۶۷	-۳.۸۶۲۸	میانگین
-۳.۸۶۲	-۳.۸۶۲۸	-۳.۸۶۲۸	-۳.۵۸۴۳	-۳.۸۶۲۸	-۳.۸۶۲۸	-۰.۹۸۶۷	-۳.۸۶۲۸	میانه
۰.۰۱۶	۰.۴۰۲۶	۶.۲۵E-۱۵	-۴.۳۰E-۰۵	۶.۵۹E-۱۵	۰.۰۰۰۸	۲.۰۹E-۱۴	۶.۵۴E-۱۵	انحراف
-۳.۳۰۳	-۳.۲۲۵۷	-۳.۲۴۷۱	-۱.۶۹۲۲	-۳.۲۸۷	-۳.۱۸۷۲	-۱.۱۰۵۳	-۳.۲۵۵۴	میانگین
-۳.۳۲۲	-۳.۳۲۰۳	-۳.۲۰۳۱	-۱.۶۹۴۲	-۳.۳۲۲	-۳.۱۹۷۴	-۱.۱۰۵۳	-۳.۲۰۳۱	میانه
۰.۰۴۷۷	۰.۲۵۶۶	۰.۰۵۷۷	-۱.۰۰E-۰۶	۰.۰۵۵	۰.۲۱۱۹	۳.۹۸E-۰۹	۰.۰۵۹۳	انحراف
-۱.۰۰۱۰۲	-۷.۳۴۲۱	-۶.۵۷۶۸	-۰.۸۰۸۲	-۷.۰۱۱۴	-۷.۹۲۵۸	-۱.۰۱۵۱۵	-۶.۶۶۱۷	میانگین
-۱.۰۱۵۳	-۹.۹۲۹۲	-۵.۱۰۰۸	-۰.۷۰۵۹	-۱۰.۱۵۳۲	-۱۰.۱۵۳۲	-۱۰.۱۵۳۳	-۵.۱۰۰۸	میانه
۰.۵۰۹	۳.۱۳۳۷	۳.۵۹۰۸	-۰.۰۴۲۹	۳.۵۸۷۸	۳.۰۰۷۷	۰.۰۰۲۴	۳.۳۰۰۱	انحراف
-۱.۰۰۴۰۲	-۷.۹۰۹	-۷.۱۸۲۷	-۱.۱۳۴۴	-۶.۳۳۶۵	-۹.۲۷۶۹	-۱.۰۳۹۸۹	-۷.۹۷۹۹	میانگین
-۱.۰۰۴۰۲	-۱۰.۲۲۹	-۱۰.۰۴۰۲۹	-۰.۹۸۷	-۶.۳۳۶۵	-۱۰.۰۴۰۲۹	-۱۰.۰۴۰۱۴	-۱۰.۰۴۰۲۹	میانه
۱.۵۲E-۱۱	۳.۲۲۴	۳.۵۴۸۷	-۰.۰۵۷۹	۰	۲.۴۹۹۵	۰.۰۰۶۷	۳.۲۵۱۱	انحراف
-۱۰.۵۳۶	-۷.۵۲۴۳	-۷.۲۱۶۱	-۱.۳۳۹۴	-۱۰.۵۳۶۴	-۹.۶۳۷۷	-۱۰.۵۲۷۷	-۱۰.۵۳۶۴	میانگین
-۱۰.۵۳۶	-۱۰.۲۳۳	-۱۰.۵۳۶۴	-۱.۲۱۰۵	-۱۰.۵۳۶۴	-۱۰.۵۳۶۴	-۱۰.۵۲۳۳	-۱۰.۵۳۶۴	میانه
۱.۳۲E-۱۱	۳.۶۱۳۱	۳.۸۲۸۳	-۰.۰۸۷۸	۱.۶۳E-۱۴	۲.۲۸۴۱	۰.۰۱۰۵	۰	انحراف

جدول ۷: نقاط بهینه سراسری و مقدار شایستگی برای توابع چهارده تا بیست و سه

شماره تابع	بردار بهینه	مقدار بهینه تابع
۱۴	(-۳۲,۳۲)	۱
۱۵	(۰/۱۹۲۸, ۰/۱۹۰۸, ۰/۱۲۳۱, ۰/۱۳۵۸)	۰.۰۰۰۳
۱۶	(۰/۰۸۹, ۰/۷۱۲), (-۰/۰۸۹, ۰/۷۱۲)	-۱.۰۰۳۱۶
۱۷	(۳/۱۴, ۱۲/۲۷), (۳/۱۴, ۲/۲۷۵), (۹/۴۲, ۲/۴۲)	۰.۳۹۸
۱۸	(۰, -۱)	۳
۱۹	(۰/۱۱۴, ۰/۵۵۶, ۰/۸۵۲)	-۳.۸۶
۲۰	(۰/۲۰۱, ۰/۱۵, ۰/۴۷۷, ۰/۲۷۵, ۰/۳۱۱, ۰/۶۵۷)	-۳.۳۲
۲۱	مینیمم محلی در $a_{ij}, i=1 \dots 5$	-۱۰.۱۵۳۲
۲۲	مینیمم محلی در $a_{ij}, i=1 \dots 7$	-۱۰.۰۴۰۲۸
۲۳	مینیمم محلی در $a_{ij}, i=1 \dots 10$	-۱۰.۵۳۶۳

جدول ۸: مدت زمان اجرا (ثانیه)

میانگین	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳			
WSTO	۱.۴۰	۴.۱۹	۴.۵۹	۴.۰۲	۴.۱۹	۸.۸۷	۳.۶۴	۴.۷۴	۴.۷۵	۴.۸۸	۴.۸۹	۴.۴۸	۳.۹۷	۴.۱۶	۴.۹۵	۶.۲۶	۷.۵۸	۳.۳۰	۳.۱۲	۳.۰۵	۳.۲۱	۳.۵۶	۳.۶۶	۴.۱۸	۴.۶۱	۵.۳۹
RGA	۱.۵۵	۲.۴۶	۳.۱۳	۲.۳۵	۲.۳۸	۶.۷۲	۲.۲۵	۲.۵۲	۲.۲۸	۳.۱۱	۲.۷۶	۲.۲۷	۲.۴۴	۲.۵۸	۳.۳۹	۳.۴۰	۷.۶۸	۲.۱۱	۱.۸۵	۱.۸۳	۱.۸۳	۲.۳۸	۲.۴۶	۳.۶۴	۴.۳۲	۵.۵۲
GSA	۱.۹۴	۸.۴۲	۸.۳۹	۸.۶۸	۸.۶۵	۱۳.۱۴	۸.۳۴	۹.۴۴	۸.۳۵	۸.۶۴	۸.۷۴	۸.۴۲	۸.۳۸	۸.۴۹	۹.۷۵	۹.۷۵	۱۲.۸۹	۶.۵۶	۵.۵۸	۵.۷۷	۵.۷۸	۶.۳۱	۶.۳۹	۷.۶۳	۸.۰۱	۹.۳۲
PSO	۱.۶۱	۵.۱۵	۵.۸۲	۴.۷۸	۴.۸۷	۹.۵۲	۴.۸۵	۵.۱۰	۵.۳۷	۵.۲۳	۴.۹۸	۵.۱۵	۶.۰۱	۵.۴۲	۶.۴۴	۶.۲۱	۹.۸۵	۴.۴۵	۴.۲۱	۴.۲۶	۴.۲۴	۵.۰۲	۵.۰۸	۶.۹۲	۸.۱۱	۷.۸۲
Cuckoo	۱.۵۹	۳.۶۵	۴.۱۸	۲.۸۷	۳.۰۴	۸.۲۵	۳.۶۶	۳.۱۲	۲.۷۹	۳.۵۷	۴.۱۶	۴.۰۲	۳.۶۵	۳.۶۱	۴.۴۴	۴.۱۵	۸.۹۵	۳.۱۲	۳.۹۸	۳.۲۶	۳.۰۹	۳.۴۵	۳.۳۵	۴.۴۹	۵.۲۵	۶.۰۱
WCA	۱.۸۲۱	۴.۸	۵.۴۴۸	۵	۵	۱۰.۲	۵.۲	۴.۶	۴.۱	۴.۵	۴.۶	۴.۸	۴.۸	۴.۷	۵.۴	۵.۴	۱۰.۸	۴.۴	۴.۵	۴	۴	۴.۲	۴.۶	۶.۲	۶.۵	۷.۸

جدول ۹: مقایسه خطای نسبی الگوریتم بهینه‌سازی جریان‌های آب و الگوریتم وراثتی و الگوریتم جستجوی فاخته برای توابع آزمون یک تا پنج

WSTO/RGA	WSTO/Cuckoo	WSTO	RGA	Cuckoo
۹.۲۱E-۱۲	۷.۰۹E-۱۴	۴.۵۴E-۱۱	۴.۹۳۰۸۰۱	۶۴۰.۶۳۸۹۶۲
۲.۶۵E-۰۶	۱.۱۳E-۰۷	۰.۰۰۰۰۰۰۶	۲.۳۵۸۹۴۹	۵۵.۴۴۷۷۷۹
۱.۰۴E-۰۶	۲.۱۶E-۰۷	۰.۰۰۲۰۱۷	۱۹۴۴.۱۶۶۸۵۷	۹۳۵۰.۱۶۸۶۶۰
۲.۹۵E-۰۱	۲.۰۳E-۰۲	۰.۵۲۰۹۵۵	۱.۷۶۷۲۳۷	۲۵.۶۵۳۷۹۴
۸.۰۸E-۰۴	۹.۸۸E-۰۷	۰.۰۸۴۰۹۳	۱۰۴.۱۳۶۰۱۴	۸۵۱۵۶.۶۹۹۱۲۹
۰.۰۵۹۱۱۹	۰.۰۰۴۰۶۲	۰.۱۲۱۴۱۴	۴۱۱.۴۷۱۹۷۲	۱۹۰۴۵.۷۲۱۶۶۵

جدول ۱۰: مقایسه زمانی (ثانیه) الگوریتم بهینه‌سازی جریان‌های آب و الگوریتم وراثتی و الگوریتم جستجوی فاخته برای توابع آزمون یک تا پنج

WSTO/RGA	WSTO/Cuckoo	WSTO	RGA	Cuckoo	میانگین
۱.۷۰۹	۱.۴۰۰	۴.۰۱۶	۲.۳۴۹	۲.۸۶۸	۱
۱.۷۵۷	۱.۳۸۰	۴.۱۸۸	۲.۳۸۴	۳.۰۳۵	۲
۱.۳۲۰	۱.۰۷۶	۸.۸۷۱	۶.۷۱۸	۸.۲۴۸	۳
۱.۶۱۹	۰.۹۹۴	۳.۶۳۶	۲.۲۴۷	۳.۶۵۷	۴
۱.۸۸۰	۱.۵۲۱	۴.۷۴۱	۲.۵۲۱	۳.۱۱۷	۵
۱.۶۵۷	۱.۲۷۴	۵.۰۹۱	۳.۲۴۴	۴.۱۸۵	میانگین

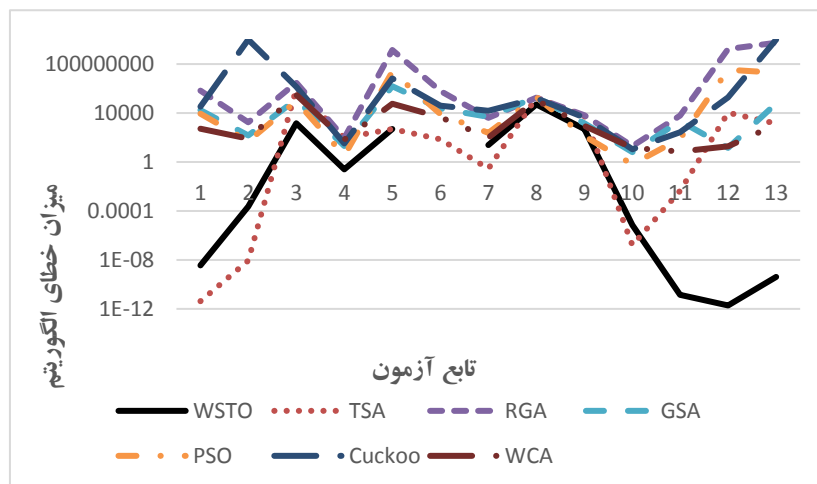
۴-۴ آزمون توابع استاندارد با بعد بالا

افزایش بعد به صورت تصاعدی افزایش می‌یابد و قدرت الگوریتم‌های بهینه‌سازی را در فرار از بهینه‌های محلی مورد بررسی قرار می‌دهد. میزان شایستگی تابع آزمون شماره هشت Π برابر عدد ۴۱۸/۹۸۲۹- می‌باشد که در آن Π بعد فضا می‌باشد بنابراین میزان شایستگی تابع هشت، در نقطه بهینه سراسری در بعد ۵۰۰ برابر با ۲۰۹۴۹۱/۴۵- است. که نزدیک‌ترین نقطه را در میان الگوریتم‌های مورد آزمون الگوریتم بهینه‌سازی جریان‌های آب به دست می‌آورد. در مابقی توابع غیر از تابع شماره نه که الگوریتم ازدحام ذرات بهترین عملکرد را دارد، می‌توان برتری مشخص الگوریتم ارائه‌شده را مشاهده کرد. جهت مشاهده بهتر، نتایج ارائه‌شده به صورت خطای محاسبه در شکل (۴) رسم شده است. همان‌طور که مشاهده می‌شود خطای الگوریتم ارائه‌شده در توابع با بعد بالا نسبت به مابقی الگوریتم‌های مورد بررسی در وضعیت مناسبی قرار دارد. به طوری که در توابع آزمون یک، دو، سه، پنج، شش، دوازده و سیزده اختلاف زیادی وجود دارد

در این آزمون، با افزایش بعد، روش‌های بهینه‌سازی با یکدیگر مقایسه می‌گردند. معمولاً الگوریتم‌ها در بعد پایین به خوبی رفتار کرده و موفق به یافتن بهینه سراسری می‌شوند، اما با افزایش بعد و رشد تصاعدی فضای پاسخ، عملکرد الگوریتم‌های بهینه‌سازی به شدت تضعیف می‌شوند. در ادامه، به ارائه نتایج این آزمون پرداخته می‌شود به طوری که بعد متغیرهای ورودی به ۵۰۰ افزایش داده می‌شود و نتایج، مورد بررسی و مقایسه قرار می‌گیرد. نتایج آزمون برای توابع یک تا هفت برای بعد ۵۰۰ در جدول (۱۰) ارائه شده است. شرایط آزمون و نقاط و بردارهای بهینه مانند آزمون اول در نظر گرفته شد. همان‌طور که مشاهده می‌شود، خطای نتیجه نهایی برای الگوریتم ارائه‌شده در تمامی توابع کمترین می‌باشد. توجه شود که بهینه مطلق برای تمامی توابع این رسته عدد صفر می‌باشد. در ادامه، نتایج آزمون با ابعاد بالا برای توابع هشت تا سیزده ارائه شده است. شایان ذکر است که بهینه‌های محلی در این رسته از توابع آزمون با

جدول ۱۰: اجرای الگوریتم‌های مورد آزمون با متغیرهای ورودی با بعد ۵۰۰ برای توابع یک تا هفت

	WSTO	TSA	SLFA	RGA	GSA	PSO	Cuckoo	WCA	
1	۳.۵۰E-۰۹	۴.۲۷E-۱۲	۱۳۷۰۲۴۴.۱	۶.۸۹E+۰۵	۱.۷۱E+۰۴	۹۰۴۶.۸	۳۳۸۲۹.۶	۵۱۳.۱۲	بهترین تاکنون
	۶۴۳۳۴	۷۰۰۰۰	۷۰۰۰۵۰	۷۰۰۰۵۰	۷۰۰۰۰۰	۷۰۰۰۰۰	۷۰۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۱۴.۸۸	۷.۴۲۳۵	۴۳.۶۲۴	۱۶.۲۴۲۴	۱۵۲.۶۴۲	۳۰.۰۰۶	۱۷.۲۸۲۱	۱۷.۸۸	زمان اجرا(ثانیه)
2	۲.۳۵E-۰۴	۹.۰۵۸E-۰۹	۱.۷۰E+۲۴۲	۱.۷۲E+۰۳	۱۴۸.۴۶	۵۰.۹۱	۱.۰۰E+۱۰	۸۱.۱۱۸۹	بهترین تاکنون
	۶۲۱۰.۸	۷۰۰۰۰	۷۰۰۰۵۰	۷۰۰۰۵۰	۷۰۰۰۰۰	۷۰۰۰۰۰	۷۰۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۱۴.۴۰۲	۷.۷۲	۵۱.۲۴۳	۱۸.۰۹۵۲	۱۵۲.۲۹	۳۱.۴۶۳	۲۰.۳۶۷۶	۱۹.۵۹۴	زمان اجرا(ثانیه)
3	۱.۴۴E+۰۳	۹.۲۹E+۰۵	۱.۱۲E+۰۷	۳.۱۰E+۰۶	۱.۲۸E+۰۵	۹.۲۳E+۰۴	۱.۴۲E+۰۶	۳.۱۱E+۰۵	بهترین تاکنون
	۶.۹۶E+۰۴	۷۰۰۰۰	۷۰۰۰۵۰	۷۰۰۰۵۰	۷۰۰۰۰۰	۷۰۰۰۰۰	۷۰۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۴۴۳.۷۵	۷۵.۵۴۱	۶۲.۳۲۲	۴۵۰.۳۷	۵۸۰.۷۴	۴۵۱.۷۷	۴۲۶.۸۳۶	۴۳۲.۴۹	زمان اجرا(ثانیه)
4	۰.۲۵۰۳	۹۹.۱۲۳۱۶	۹۳.۴۵۲	۸۹.۱۱۵۳	۱۹.۸۱۳	۳.۲۵۴۶	۳۲.۴۰۳	۶۶.۵۱	بهترین تاکنون
	۵۹۰۵۷	۷۰۰۰۰	۷۰۰۰۵۰	۷۰۰۰۵۰	۷۰۰۰۰۰	۷۰۰۰۰۰	۷۰۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۱۲.۱۴	۷.۵۲۲	۳۹.۲۵۱	۱۵.۷۹۲۱	۱۵۱.۲۶	۲۹.۴۷۶	۱۷.۱۰۷	۱۷.۴۸۹	زمان اجرا(ثانیه)
5	۴۹۳.۳۸۵۴۳	۴۹۷.۸۱۴	۵.۹۱E+۰۹	۱.۴۲۱E+۰۹	۱.۵۴E+۰۶	۲.۳۹E+۰۷	۶.۵۹E+۰۶	۵.۶۲E+۰۴	بهترین تاکنون
	۶۲۸۴۱	۷۰۰۰۰	۷۰۰۰۵۰	۷۰۰۰۵۰	۷۰۰۰۰۰	۷۰۰۰۰۰	۷۰۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۱۶.۸۶۱	۸.۲۱۲	۲۳.۲۱۴	۱۹.۲۱۶۹	۱۵۳.۹۶	۳۲.۰۱۵	۱۹.۵۱۵۹	۲۰.۷۲۶	زمان اجرا(ثانیه)
6	۰	۷۰.۲۶	۱.۳۷E+۰۶	۶۳۷۸۷۷	۲۲۷۱۲	۸۲۰۹.۷	۳۸۹۴۴	۴۶۰۳	بهترین تاکنون
	۶۰۵۰۱	۷۰۰۰۰	۷۰۰۰۵۰	۷۰۰۰۵۰	۷۰۰۰۰۰	۷۰۰۰۰۰	۷۰۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۱۷.۸۲۴	۷.۸۱۵	۳۳.۸۵۱	۱۷.۱۰۸	۱۵۴.۷۲	۳۱.۷۲	۱۹.۴۸۲۸	۲۶.۲۸۲	زمان اجرا(ثانیه)
7	۲۳.۵۰۶	۰.۲۸۳۷	۴۷۲۰۴.۳۰۶۲	۳۹۹۶.۷۴	۴۷۴۴.۵	۲۴۰.۱	۱۵۰۳۳۶	۹۷.۹۴۶	بهترین تاکنون
	۷۸۸۴۵	۷۰۰۰۰	۷۰۰۰۵۰	۷۰۰۰۵۰	۷۰۰۰۰۰	۷۰۰۰۰۰	۷۰۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۲۴.۵۷۸	۱۱.۳۳۹	۳۵.۲۶۱	۲۰.۶۱۱۱	۱۶۰.۰۳	۳۵.۱۵۱	۲۲.۶۳۷۷	۲۳.۲۳۲	زمان اجرا(ثانیه)



شکل ۴: خطای الگوریتم‌های مورد بررسی برای توابع آزمون یک تا سیزده با بعد ۵۰۰

جدول ۱۱: اجرای الگوریتم‌های مورد آزمون با متغیرهای ورودی با بعد ۵۰۰ برای توابع هشت تا سیزده

	WSTO	TSA	SLFA	RGA	GSA	PSO	Cuckoo	WCA	
	-۱۵۹۲۸۵.۹	-۳.۴۷E+۰۴	-۱۷۷۳۳.۱	-۲۹۰۲۷.۵	-۹۱۲۵.۸	-۲۴۹۶.۶	-۵۳۳۰۳.۵	-۱۰۰۱۷۰	بهترین تاکنون
8	۷۴۸۵۵	۷۰۰۰۰	۷۰۰۵۰	۷۰۰۵۰	۷۰۰۰۰	۷۰۰۰۰	۷۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۲۴.۸۸۵	۹.۱۰۴	۴۴.۵۹۶	۲۱.۲۹۶	۱۶۰.۵۳	۳۴.۴۴۲	۲۳.۰۱۱۷	۲۳.۳۰۳	زمان اجرا(ثانیه)
	۴۹۷.۴۷	۵۶۹۶.۷۹	۸۲۹۷.۲۳	۶۵۰۹.۲۷	۱۴۷۲.۳	۱۳۰۰.۳	۴۲۲۵.۳۲	۸۷۱.۵۸	بهترین تاکنون
9	۶۲۴۱۰	۷۰۰۰۰	۷۰۰۵۰	۷۰۰۵۰	۷۰۰۰۰	۷۰۰۰۰	۷۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۱۸.۵۰۵	۸.۶۲۶	۲۳.۳۱۸	۱۷.۷۴۵۱	۱۵۳.۸۶	۳۱.۹۹۵	۲۰.۱۹۵۶	۲۰.۳۸۸	زمان اجرا(ثانیه)
	۷.۰۰E-۰۶	۱.۵۶E-۰۷	۲۰.۹۸۱۶۸	۲۰.۰۰۳۳	۶.۳۸۴	۰.۶۴۸۲	۱۰.۹۷۷۲	۱۳.۸۴۶	بهترین تاکنون
10	۶۰۴۴۶	۷۰۰۰۰	۷۰۰۵۰	۷۰۰۵۰	۷۰۰۰۰	۷۰۰۰۰	۷۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۱۵.۹۶۵	۸.۳۴۷	۵۶.۳۱۹	۱۹.۳۲۹۳	۱۵۳.۸۱	۳۳.۴۵۸	۲۰.۷۲۶۷	۲۱.۲۳۴	زمان اجرا(ثانیه)
	۱.۴۰E-۱۱	۰.۰۰۴۳۵۳	۱۲۲۶۳.۲۵	۶۲۲۸.۵۹	۲۱۷۵.۴	۸۱.۳۰۴	۲۶۴.۳	۷.۱۴۳	بهترین تاکنون
11	۷۰۱۶۱	۷۰۰۰۰	۷۰۰۵۰	۷۰۰۵۰	۷۰۰۰۰	۷۰۰۰۰	۷۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۲۲.۰۷۲	۹.۴۸۳	۲۴.۲۶۵	۲۲.۰۲۵۲	۱۵۷.۵۷	۳۵.۴۴۴	۲۲.۴۳۲۴	۲۴.۸۷۷	زمان اجرا(ثانیه)
	۱.۹۰E-۱۲	۱.۰۳E+۰۴	۱.۳۹E+۱۰	۱.۷E+۰۹	۱۴.۰۶	۳.۴۵E+۰۷	۱.۸۷E+۰۵	۱۹.۲۸۱	بهترین تاکنون
12	۶۱۵۸۸	۷۰۰۰۰	۷۰۰۵۰	۷۰۰۵۰	۷۰۰۰۰	۷۰۰۰۰	۷۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۳۴.۱۱۸	۲۲.۳۸۱	۶۸.۲۲۷	۳۷.۹۲۳	۱۷۵.۲۹	۵۲.۲۰۶	۴۰.۱۴۵۱	۴۲.۶۶	زمان اجرا(ثانیه)
	۴.۱۱E-۱۰	۱۳۵۶.۹۶۳	۲.۶۲E+۱۰	۵.۵E+۰۹	۶.۱۷E+۰۴	۱.۹۴E+۰۷	۱E+۱۰	۱۱۳۸.۵	بهترین تاکنون
13	۶۲۶۴۳	۷۰۰۰۰	۷۰۰۵۰	۷۰۰۵۰	۷۰۰۰۰	۷۰۰۰۰	۷۰۰۵۰	۷۰۶۱۰	تعداد فراخوانی
	۳۵.۰۱۸	۲۲.۲۹۶	۴۸.۵۷۹	۳۸.۸۳۸۳	۱۷۵.۶۹	۵۱.۹۵۷	۴۰.۳۸۱۲	۳۹.۸۱۲	زمان اجرا(ثانیه)

۴-۵ بررسی آماری نتایج

جهت بررسی و مقایسه دقیق‌تر نتایج حاصل شده، در این بخش از آزمون‌های ناپارامتریک ویلکاکسون^{۱۳} و فریدمن^{۱۴} استفاده شده است. نتایج آزمون ویلکاکسون برای بعد ۳۰ در جدول ۱۲ قابل مشاهده است. در این آزمون، الگوریتم‌های مورد آزمون یک به یک با الگوریتم پیشنهادی جریان‌های آب (WSTO) مورد مقایسه قرار گرفته‌اند. در این مقایسه هر جا که الگوریتم پیشنهادی بازنده بوده است، خالی گذاشته شده است و هنگام برتری آن علامت تیک در جدول قرار گرفته است. برتری نسبی الگوریتم پیشنهادی در این جدول مشاهده می‌شود.

جدول ۱۲. نتایج آزمون ویلکاکسون الگوریتم پیشنهادی در مقابل سایر

الگوریتم‌های ابتکاری

تابع	TSA	SLFA	WCA	Cuckoo	PSO	GSA	RGA
F1		✓	✓	✓	✓		✓
F2			✓	✓	✓		✓
F3			✓	✓	✓	✓	✓
F4			✓	✓	✓	✓	✓
F5	✓	✓	✓	✓	✓	✓	✓
F6	✓	✓	✓	✓	✓	✓	✓
F7							
F8	✓	✓	✓	✓	✓	✓	✓
F9	✓	✓	✓	✓	✓	✓	✓
F10	✓	✓	✓	✓	✓		✓

آماره‌های آزمون ویلکاکسون در جدول ۱۳ ارائه شده است. در این جدول، رتبه منفی و رتبه مثبت این آزمون بیان‌کننده وزن اختلاف بین الگوریتم‌ها نسبت به نتیجه مقایسه می‌باشد. در آزمون فریدمن تمامی الگوریتم‌های بهینه‌سازی همزمان مورد بررسی و رتبه‌بندی می‌شوند. با توجه به این‌که این آزمون بر اساس میزان خطای الگوریتم‌های بهینه‌سازی انجام شده است هرچه رتبه الگوریتم کمتر باشد نشان از کارایی بالاتر الگوریتم می‌باشد. نتایج این آزمون را برای بعد سی در جدول ۱۴ و آماره‌های آن در جدول ۱۵ و برای بعد پانصد در جدول ۱۶ و آماره‌های آن در جدول ۱۷ مشاهده می‌شود. برتری الگوریتم ارائه شده در این آزمون نیز مشاهده

مکان یافته شده برای همه ذرات، موقعیت بعدی خود را می‌یابد. به بیان دیگر، سرعت و جهت حرکت ذره حاصل جمع وزنی سرعت قبلی ذره، بهترین موقعیت ذره و بهترین موقعیت همه ذرات است. در الگوریتم بهینه‌سازی جریان‌های آب نیز در ابتدا هر جریان در یک جهت تصادفی با یک سرعت تصادفی شروع به حرکت می‌کند، ولی حرکت هر ذره تنها با توجه به بهبود خود ذره اصلاح می‌شود. به طوری که، جهت آن در ابتدا در جهت خط واصل بین دو بردار است (بردار مکان و بردار جهت) و در صورت عدم بهبود تغییر داده خواهد شد. این تغییر در ابتدای الگوریتم به صورت تصادفی صورت می‌گیرد و در اجراهای پایانی جهت به سمت میانگین بهترین موقعیت‌های دیگر جریان‌ها تغییر داده می‌شود و سرعت حرکت هر جریان نیز تابع میزان بهبود متوالی است. این بدین معنی که هر چه در جهتی بهبود بیشتری صورت گیرد، سرعت حرکت با عملکرد تندآب افزایش بیشتری خواهد داشت که این خود منجر به کاهش حجم عملیات و تعداد فراخوانی تابع می‌گردد.

جدول ۱۵. آماره‌های آزمون فریدمن برای بعد ۳۰

تعداد توابع آزمون	بعد
۲۳	۳۰
مربع کای	۳۲.۱۱۵
درجه آزادی	۷
سطح معنی داری	۰.۰۰۰

جدول ۱۶. آزمون فریدمن برای بعد ۵۰۰ و رتبه‌بندی الگوریتم پیشنهادی WSTO در مقایسه با الگوریتم‌های ابتکاری دیگر

رتبه	الگوریتم
۱.۵۸	WSTO
۲.۵۸	TSA
۳.۵۸	WCA
۴.۵۸	PSO
۴.۴۲	GSA
۵.۱۷	Cuckoo
۶.۵۸	RGA
۷.۵۰	SLFA

می‌شود و با افزایش بعد اختلاف بیشتر می‌شود. کمتر بودن سطح معنی‌داری از عدد ۰/۰۵ در این آزمون‌ها نشان از رد فرضیه صفر، یعنی عدم وجود تفاوت معنی‌دار می‌باشد.

جدول ۱۳. آماره‌های آزمون ویلکاکسون

الگوریتم در مقابل	R+	R-	سطح معنی داری	فرضیه صفر
RGA	۱۲.۴۳	۷.۵	۰.۰۰۰	رد شد
GSA	۱۳.۶۷	۶.۰۰	۰.۰۰۱	رد شد
PSO	۱۱.۸۹	۱۲.۵۰	۰.۰۰۷	رد شد
Cuckoo	۱۴.۴۱	۵.۱۷	۰.۰۰۱	رد شد
WCA	۱۲.۴۰	۹.۳۳	۰.۰۰۱	رد شد
SLFA	۱۲.۳۰	۱۰.۰۰	۰.۰۰۱	رد شد
TSA	۱۳.۴۱	۸.۰۰	۰.۰۰۶	رد شد

۴-۶- مقایسه الگوریتم بهینه‌سازی جریان‌های آب با الگوریتم وراثتی

در این قسمت به بررسی عملکرد الگوریتم بهینه‌سازی جریان‌های آب و شباهت‌ها و تفاوت‌های آن با الگوریتم پرکاربرد وراثتی پرداخته می‌شود. عملکرد تزویج و عملکرد جهش، عملگرهای اصلی الگوریتم وراثتی شمرده می‌شوند. می‌توان گفت، عملکرد جهش در الگوریتم وراثتی به دلیل بالا بردن ویژگی تنوع الگوریتم (در اجراهای پایانی) و به جهت یافتن نقاط نادیده و جستجوی تصادفی خارج از سازوکارهای منظم الگوریتم، در نظر گرفته شده است. عملکرد گریز در الگوریتم بهینه‌سازی جریان‌های آب عملکرد مشابهی را اعمال می‌کند، به‌ویژه در فرار از بهینه‌های محلی. با این تفاوت که در عملکرد جهش الگوریتم وراثتی تنها تغییر جزئی در برخی مؤلفه‌ها اعمال می‌شود و این فرآیند به‌طور کامل تصادفی انجام می‌شود (با توزیع گاوسی). در صورتی که عملکرد گریز تنها در شرایط رسیدن جریان آب به گودال آب (بهینه محلی) اعمال شده و شعاع گودال آب (میزان تغییر مکان بردار) دقیقاً مشخص است (در ابتدای الگوریتم شعاع بزرگ و به تدریج کوچک می‌شود).

۴-۷- مقایسه الگوریتم ارائه‌شده با الگوریتم ازدحام ذرات

در الگوریتم ازدحام ذرات در ابتدا هر ذره در یک جهت تصادفی با یک سرعت تصادفی شروع به حرکت می‌کند و در ادامه هر ذره با توجه به حرکت قبلی خود و بهترین مکان یافته شده خود و بهترین

جدول ۱۴. آزمون فریدمن برای بعد ۳۰ و رتبه‌بندی الگوریتم پیشنهادی WSTO در مقایسه با الگوریتم‌های ابتکاری دیگر

الگوریتم	WSTO	SLFA	TSA	PSO	WCA	GSA	Cuckoo	RGA
رتبه	۲.۲۶	۴.۳۰	۴.۳۵	۴.۴۳	۴.۵۴	۴.۵۹	۵.۴۷	۵.۷۸

جدول ۱۷. آماره‌های آزمون فریدمن برای بعد ۵۰۰

تعداد توابع آزمون	۱۳
بعد	۵۰۰
مربع کای	۴۹.۶۳۹
درجه آزادی	۷
سطح معنی داری	۰.۰۰۰

۴-۸ مؤلفه‌های تنظیم الگوریتم

مصالحه بین بهره‌گیری^{۱۰} (قدرت همگرایی) و کاوش^{۱۱} (قدرت جستجو)، همواره چالش طراحان الگوریتم‌های فراابتکاری می‌باشد. در رویه‌های بهینه‌سازی باید الگوریتم‌هایی ایجاد کرد که حدود مناسب، میزان و حجم محاسبات و نزدیکی به بهینگی را تضمین کند و تعادل خواسته‌شده بین آن‌ها را ارضا نماید. این حد از تعادل، می‌تواند برای مسائل و کاربردهای گوناگون، متفاوت باشد. بنابراین با توجه به سطح پیچیدگی مسئله موردنظر، دقت موردنیاز و کاربرد آن ممکن است الگوریتم بهینه‌سازی برای تمامی مسائل به‌اندازه کافی متعادل رفتار نکند. یکی از راهکارها شاید این باشد که الگوریتم، مؤلفه‌های قابل تنظیم داشته باشد تا کاربر بتواند با تنظیم آن مؤلفه‌ها، تعادل مطلوب خود را، بین دقت جواب به‌دست‌آمده و میزان محاسبات برقرار نماید [۴۱]. در ادامه به توضیح پیرامون مؤلفه‌های قابل تنظیم الگوریتم ارائه‌شده پرداخته می‌شود.

اولین مؤلفه‌ای که در الگوریتم بهینه‌سازی جریان‌های آب در تعادل بین همگرایی و قدرت جستجو تأثیرگذار است، مؤلفه β می‌باشد که این مؤلفه ضریبی است در رابطه (۳) که مربوط به عملگر تندآب است. همان‌طور که در رابطه (۳) مشخص است، هر چه این مؤلفه بزرگ‌تر باشد سرعت حرکت تندآب افزایش تصاعدی بیشتری خواهد داشت و موجب افزایش قدرت جستجو و کاهش همگرایی الگوریتم می‌گردد و بالعکس. دومین مؤلفه، تعداد تغییر جهت‌های جستجو توسط جریان‌ها می‌باشد، که مربوط به وضعیتی است که حرکت جریان در جهت مشخص موجب بهبود نمی‌شود، لذا عملگر تغییر جهت اعمال می‌شود. حال این مسئله که چند بار یک جریان می‌تواند در یک نقطه مشخص جهت‌های مختلف را موردبررسی قرار دهد، مورد تأثیر این مؤلفه می‌باشد. به‌طوری‌که هر چه تعداد این تغییر جهت‌ها بیشتر شود قدرت جستجو کاهش و میزان همگرایی الگوریتم افزایش می‌یابد و بالعکس. سومین مؤلفه، تعداد گودال‌های مسئله است. همان‌طور که در بخش سوم توضیح داده شد، در صورت رسیدن جریان آب به عمیق‌ترین موقعیت خود و عدم ایجاد بهبود در جهت‌های مختلف، موقعیت فعلی به‌عنوان گودال آب شناخته‌شده و یک همسایگی از آن با شعاع متغیر (تابعی از تکرار حلقه اصلی الگوریتم) به‌عنوان منطقه ممنوعه تعریف‌شده و هیچ جریانی در این همسایگی حرکت نخواهد کرد. بنابراین این سازوکار باعث افراز فضا

(به‌ویژه در تکرارهای ابتدایی الگوریتم) و جلوگیری از محاسبات تکراری در فضاها بررسی‌شده و افزایش قدرت جستجوی الگوریتم خواهد شد. لذا در الگوریتم تعداد بیشینه گودال‌ها به‌عنوان یک مؤلفه کنترلی در نظر گرفته‌شده است، به‌طوری‌که هر چه تعداد این گودال‌ها بیشتر در نظر گرفته شود قدرت جستجو افزایش و همگرایی الگوریتم کاهش می‌یابد و بالعکس.

۵- جمع‌بندی و پیشنهادها برای آینده

طراحی یک روش قطعی برای رویارویی با مسائلی که با روش‌های کلاسیک قابل بررسی نیستند، به نظر غیرممکن می‌رسد و الگوریتم برای حل این مسائل باید هم‌زمان با هوشمندی، رفتار نامشخص و تصادفی داشته باشد تا بتواند قسمت‌هایی از فضای امکان‌پذیر که توسط روش‌های دیگر موردبررسی قرار نگرفته است، شناسایی کند. در این مقاله، الگوریتم بهینه‌سازی فراابتکاری جریان‌های آب پیشنهاد شد و توانایی آن در بهینه‌سازی توابع استاندارد به خصوص در بعد بالا مورد بررسی و تایید قرار گرفت. در جستجوی فضای جواب در الگوریتم‌های فراابتکاری، به دو ویژگی کاوش و بهره‌گیری توجه ویژه‌ای وجود دارد. با توجه به اهمیت مصالحه بین این دو ویژگی، سعی در ایجاد تعادل بین آن‌ها همیشه چالشی برای روش‌های بهینه‌سازی است. عملگر تندآب در الگوریتم بهینه‌سازی جریان‌های آب، به‌نوعی این رابطه را تضعیف می‌کند (همگرایی با سرعت تصاعدی که هم‌زمان هم جستجوی وسیع‌تر را ایجاد می‌کند هم همگرایی بیشتر). ارایه نسخه‌های دودویی و چندهدفه این الگوریتم و همچنین بررسی کارایی این الگوریتم بهینه‌سازی در کاربردهای عملی از موضوعات آینده پژوهشی می‌باشد.

مراجع

- [1] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department 2005.
- [2] Y.-L. Lin, W.-D. Chang, and J.-G. Hsieh, "A particle swarm optimization approach to nonlinear rational filter modeling," *Expert Systems with Applications*, vol. 34, pp. 1194-1199, 2008.
- [3] M. Nasri, H. Nezamabadi-Pour, and M. Maghfoori, "A PSO-based optimum design of PID controller for a linear brushless DC motor," *World Academy of Science, Engineering and Technology*, vol. 26, pp. 211-215, 2007.
- [4] R. Bahramipour-Esfahani, M. Nasri, and S. M. Tabatabaei, "Designing a Metaheuristic Multi-objective Fractional-order PID Controller for TRMS system," *Computational Intelligence in Electrical Engineering*, pp. -, 2020.
- [5] H. Nezamabadi-Pour, S. Saryazdi, and E. Rashedi, "Edge detection using ant algorithms," *Soft Computing*, vol. 10, pp. 623-628, 2006.
- [6] E. Gharaati and M. Nasri, "A new band selection method for hyperspectral images based on constrained optimization," in *2015 7th Conference on Information and Knowledge Technology (IKT)*, 2015, pp. 1-6.
- [7] Y. Liu, Z. Yi, H. Wu, M. Ye, and K. Chen, "A tabu search approach for the minimum sum-of-squares clustering problem," *Information Sciences*, vol. 178, pp. 2680-2704, 2008.
- [8] F. Saadat and M. Nasri, "A multibiometric finger vein verification system based on score level fusion strategy," in *2015 International Congress on Technology, Communication and Knowledge (ICTCK)*, 2015, pp. 501-507.

- [25] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawk optimization: Algorithm and applications," *Future generation computer systems*, vol. 97, pp. 849-872, 2019.
- [26] P. Pijarski and P. Kacejko, "A new metaheuristic optimization method: the algorithm of the innovative gunner (AIG)," *Engineering Optimization*, vol. 51, pp. 2049-2068, 2019.
- [27] A. F. Nematollahi, A. Rahiminejad, and B. Vahidi, "A novel meta-heuristic optimization method based on golden ratio in nature," *Soft Computing*, vol. 24, pp. 1117-1151, 2020.
- [28] S. H. A. Kaboli, J. Selvaraj, and N. Rahim, "Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems," *Journal of Computational Science*, vol. 19, pp. 31-42, 2017.
- [29] Z. Wei, "A Raindrop Algorithm for Searching The Global Optimal Solution in Non-linear Programming," *arXiv preprint arXiv:1306.2043*, 2013.
- [30] H. Shah-Hosseini, "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm," *International Journal of Bio-inspired computation*, vol. 1, pp. 71-79, 2009.
- [31] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm-A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. 110, pp. 151-166, 2012.
- [32] T. R. Biyanto, G. P. Dienanta, T. O. Angrea, I. T. Utami, L. Ayurani, M. Khalil, et al., "Optimization of supersonic separation (3S) design using rain water algorithm," in *AIP conference proceedings*, 2018, p. 050008.
- [33] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153-165, 2015.
- [34] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, pp. 28-39, 2006.
- [35] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of genetic algorithms*. vol. 2, ed: Elsevier, 1993, pp. 187-202.
- [36] E. Rashedi, H. Nezamabadi-Pour, and S. J. I. s. Saryazdi, "GSA: a gravitational search algorithm," vol. 179, pp. 2232-2248, 2009.
- [37] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, 1995, pp. 1942-1948.
- [38] A. H. Gandomi, X.-S. Yang, and A. H. J. E. w. c. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," vol. 29, pp. 17-35, 2013.
- [39] S. Kaur, L. K. Awasthi, A. Sangal, and G. Dhiman, "Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103541, 2020.
- [40] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering optimization*, vol. 38, pp. 129-154, 2006.
- [41] J. Pearl, "Intelligent search strategies for computer problem solving," *Addison Wesley*, 1984.
- [9] F. Saadat and M. Nasri, "A GSA-based method in human identification using finger vein patterns," in *2016 1st Conference on swarm Intelligence and Evolutionary Computation (CSIEC)*, 2016, pp. 102-106.
- [10] S. M. Koloushani, M. Nasri, and M. M. Rezaei, "Strategic management of stochastic power losses in smart transmission grids," *International Transactions on Electrical Energy Systems*, vol. 29, p. e12032, 2019.
- [11] J. Ebrahimi, M. Abedini, M. M. Rezaei, and M. Nasri, "A two-step approach to energy management in smart micro-grids aimed at improving social welfare levels and the demand side management effect," *Iranian Electric Industry Journal of Quality and Productivity*, vol. 9, pp. 56-67, 2020.
- [12] J. Ebrahimi, M. Abedini, M. M. Rezaei, and M. Nasri, "Optimum design of a multi-form energy in the presence of electric vehicle charging station and renewable resources considering uncertainty," *Sustainable Energy, Grids and Networks*, vol. 23, p. 100375, 2020.
- [13] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT press, 1992.
- [14] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning," *Machine Learning*, vol. 3, pp. 95-99, 1988/10/01 1988.
- [15] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, 1999, pp. 1470-1477.
- [16] J. Kennedy and R. Eberhart, "Particle swarm optimization (PSO)," in *Proc. IEEE International Conference on Neural Networks, Perth, Australia*, 1995, pp. 1942-1948.
- [17] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *2007 IEEE congress on evolutionary computation*, 2007, pp. 4661-4667.
- [18] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information sciences*, vol. 179, pp. 2232-2248, 2009.
- [19] R. Rajabioun, "Cuckoo optimization algorithm," *Applied soft computing*, vol. 11, pp. 5508-5518, 2011.
- [20] A. Sadollah, H. Sayyaadi, and A. Yadav, "A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm," *Applied Soft Computing*, vol. 71, pp. 747-782, 2018.
- [21] G.-G. Wang, S. Deb, and L. dos Santos Coelho, "Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems," *IJBIC*, vol. 12, pp. 1-22, 2018.
- [22] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, pp. 151-164, 2018.
- [23] S. Shadravan, H. Naji, and V. K. Bardsiri, "The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 80, pp. 20-34, 2019.
- [24] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural computing applications*, vol. 31, pp. 1995-2014, 2019.

پاورقی‌ها:

⁹ Average best-so-far

¹⁰ Median best-so-far

¹¹ Standard Deviation best-so-far

¹² Standard Benchmark Functions

¹³ Wilcoxon test

¹⁴ Friedman test

¹⁵ Exploitation

¹⁶ Exploration

¹ http://gotbooks.miracosta.edu/geology/images/CA_water_relief.jpg

² Movement Operator

³ Escape Operator

⁴ Waterfall Operator

⁵ Turn Operator

⁶ Overflow Operator

⁷ Vortex Operator

⁸ Unimodal test Functions