



توسعه الگوریتم بهینه‌سازی امواج آب برای حل مسائل بهینه‌سازی چندگانه

بنان برومند^(۱) سید محمد حسین معطر*^(۲)

(۱) گروه مهندسی کامپیوتر، واحد فردوس، دانشگاه آزاد اسلامی، فردوس، ایران

(۲) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران*

(تاریخ دریافت: ۱۴۰۱/۰۷/۲۴ تاریخ پذیرش: ۱۴۰۱/۱۰/۴)

چکیده

پیشنهاد این مقاله، ارائه یک الگوریتم بهینه‌سازی چندگانه تکاملی جدید مبتنی بر الگوریتم امواج آب است. روش‌های گروهی و تکراری بسیاری برای حل مساله بهینه‌سازی چندگانه تاکنون ارائه شده است. مهمترین چالش روش‌های پیشین، نیاز به وجود دانش از تعداد بهینه‌ها و همچنین وجود برخی پارامترهای وابسته به مساله از جمله شعاع همسایگی است. همچنین چگونگی تعیین پایداری و ناپایداری گروه پاسخ‌ها از تصمیمات چالش برانگیز در این زمینه است. در الگوریتم پیشنهادی، بهینه‌ها بوسیله گروه‌های جداگانه‌ای که در حال تکامل هستند جستجو می‌شوند. برای این کار از یک معیار پایداری برای مشخص کردن گروه‌های ناپایدار و پایدار استفاده می‌شود. به این ترتیب اگر گروهی تکامل خود را تا یک آستانه انجام دهد، به این معنی است که دارای یک بهینه است و باید این بهینه در حافظه خارجی ذخیره گردد. بعد از چندین تکرار، جواب‌های ذخیره شده در حافظه، تمام بهینه‌ها را شامل می‌شوند و از این طریق بهینه‌های سراسری و محلی تشخیص داده می‌شوند. الگوریتم پیشنهادی بر روی سیزده تابع معیار، که در مقالات و کارهای پیشین در نظر گرفته شده است، مورد مقایسه قرار گرفته است. با توجه به این مقایسات مشخص شده است که نتایج الگوریتم پیشنهادی نسبت به سایر الگوریتم‌ها دارای برتری است.

کلید واژه‌ها: الگوریتم تکاملی، الگوریتم امواج آب، بهینه‌سازی چندگانه.

* عهده‌دار مکاتبات:

سید محمد حسین معطر

نشانی: گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

پست الکترونیکی: moattar@mshdiau.ac.ir

در مسائل بهینه‌سازی چندگانه^۱ بیش از یک پاسخ بهینه وجود دارد. دو تکنیک برای پیدا کردن بهینه‌های چندگانه در توابع چندگانه وجود دارد: روش‌هایی با تکنیک‌های تکراری و روش‌هایی با زیرجمعیت‌های موازی. محفود^۲ [1] تحقیقاتی برای پیدا کردن راه حل‌های چندگانه که شامل بهینه محلی باشند ارائه کرده است و مقایسه دقیقی بین روش‌های تکراری و موازی انجام داده است. او نشان داده است که الگوریتم‌های با تکنیک تکراری در ترکیب با الگوریتم‌های موازی نتایج بسیار خوبی دارند [1]. در این الگوریتم‌ها گروه‌هایی از جمعیت‌ها، به صورت موازی، بین اجراهای مختلف اما موازی از یک الگوریتم ازدحامی و تکراری توزیع می‌شوند و هر اجرا پاسخ بهینه را جستجو می‌کند. این تکنیک در عین حال که به دلیل ماهیت اجرای موازی زمان محاسبات بالاتری ندارد، منجر به یافتن مناسب‌تر بهینه‌های سراسری می‌شود.

از روش‌هایی که در زمینه بهینه‌سازی چندگانه ارائه شده است می‌توان به روش ازدحام^۳ اشاره کرد [2]. همچنین توسط محفود روش ازدحامی قطعی^۴ [3] ارائه شده است. در [4] روش ازدحامی احتمالی^۵ ارائه گردیده است. نقاط قوت این روش‌ها، ساده و سریع بودن آنها است ولی کارکردشان به معیار شباهتی که محاسبه می‌شود بستگی دارد. همچنین برای بدست آوردن تعداد بهینه‌ها باید جمعیت خروجی با مکانیزمی که وابسته به شعاع بهینه‌ها است ارزیابی شود [5]. روش پاکسازی^۶ توسط پتروفسکی^۷ ارائه گردیده است که از فاصله شعاع پاکسازی^۸ استفاده می‌کند [6].

روش اشتراک برزندگی^۹ توسط گلدبرگ^{۱۰} و ریچاردسون^{۱۱} ارائه شده است که با استفاده از شعاع اشتراک^{۱۲} جمعیت را تقسیم می‌کند [7]. میلر^{۱۳} روش اشتراک برزندگی پویا^{۱۴} را ارائه کرد که به یک تخمین از تعداد بهینه‌ها نیاز دارد [8]. گلدبرگ و وانگ^{۱۵} روش اشتراک تکاملی^{۱۶} را ارائه کردند. این روش مکان و شعاع هر بهینه را برای انطباق با فضاها پیچیده تغییر می‌دهد، ولی باز هم وابسته به شعاع اشتراک است [9]. توسط هریک^{۱۷} روش انتخاب تورنمنت محدود^{۱۸} ابداع گردید [10].

۱ Multi modal

۲ Mahfoud

۳ Crowding

۴ Deterministic crowding

۵ Probabilistic crowding

۶ Clearing

۷ Petrowski

۸ Radius clearing

۹ Fitness sharing

۱۰ Goldberg

۱۱ Richardson

۱۲ Sharing Radius

۱۳ Miller

۱۴ Dynamic niche sharing

۱۵ Wang

۱۶ Coevolutionary sharing

۱۷ Harik

۱۸ Restricted Tournament Selection

همچنین روی^۱ روش تطبیقی انتخاب تورنمنت محدود را ارائه کرد [11]. این روش‌ها به مقدار حد آستانه برای میزان شباهت وابسته هستند و تخمین حد آستانه نیز نیاز به داشتن اطلاعات قبلی از تابع بهینه‌سازی دارد.

رویکرد مبتنی بر خوشه‌بندی توسط بین^۲ ارائه گردید و در آن با استفاده از خوشه‌بندی سعی در رفع وابستگی به تخمین شعاع بهینه بود [12]. لی^۳ و همکاران روش حفاظت از گونه‌ها^۴ را ارائه کردند، که در آن برای هر گونه، پارامتری با نام فاصله گونه^۵ تعریف می‌شود که تخمین درست آن با توجه به دانستن تعداد بهینه‌ها در تابع بهینه‌سازی صورت می‌گیرد [13-14]. بیسلی^۶ روش نیچینگ ترتیبی^۷ را ارائه کرد که نتایج این روش بستگی زیادی به شعاع نیچ دارد [15]. با بهبود نیچینگ ترتیبی توسط ژانگا^۸ سعی شده است نتایج به تعیین مقدار شعاع یا فاصله گروه‌ها حساس نباشد [16]. در [17] از تکنیک همسایه تطبیقی برای بهینه‌سازی چندوجهی استفاده شده است. در [18] نیز از رویکرد مبتنی بر آرشیو برای بهینه‌سازی چندوجهی در الگوریتم تفاضل تکاملی استفاده شده است.

تکنیک سرگردانی^۹ ارائه شده توسط لانگ^{۱۰} نشان داد که در الگوریتم‌های مبتنی بر زیر جمعیت استفاده از حافظه خارجی برای ذخیره جواب‌ها می‌تواند در بهینه‌سازی چندگانه موثر باشد [19]. لیانگ^{۱۱} نیز تکنیکی با عنوان جستجوی جمعیت نخبه‌گرا بصورت تطبیقی^{۱۲} ارائه کرد که در آن با تنظیم اپراتورهای الگوریتم ژنتیک و استفاده از فاصله‌ای برای تعیین همسایگی دو جواب، موقعیت دو جواب نسبت به هم مشخص می‌شود [20].

در حالت عمومی الگوریتم‌های بهینه‌سازی چندگانه، برای حل موفقیت‌آمیز باید دو موضوع را در نظر بگیرند: ۱- چگونگی شناخت بهینه‌های چندگانه سراسری و محلی. ۲- چگونگی نگهداری بهینه‌های شناخته شده تا پایان جستجو. همچنین تخمین نامناسب پارامتر شعاع، کارایی الگوریتم‌ها را تحت تاثیر قرار می‌دهد [21, 22].

از مزایای الگوریتم پیشنهادی نسبت به سایر الگوریتم‌ها این است که نیاز به نگهداری جمعیت پایدار در اطراف هر بهینه‌ای که پیدا می‌کند ندارد، زیرا فقط مکان نقاط بهینه نگهداری می‌شود. از طرفی، به علت اینکه هر جواب ذخیره شده در حافظه نشان‌دهنده یک بهینه است، به مکانیزمی که با استفاده از آن بخواهیم تعداد بهینه‌ها را از روی جمعیت بدست بیاوریم، نیاز نداریم. الگوریتم پیشنهادی وابسته به پارامتر شعاع و اطلاعات قبلی در رابطه با تعداد و توزیع بهینه‌ها نیست و در کاربردهایی که این اطلاعات از تابع بهینه‌سازی وجود ندارد می‌تواند بخوبی مورد استفاده قرار گیرد. از جمله این کاربردها می‌توان به نظریه بازی‌ها و حل بازی‌های چند نفره غیر همکارانه اشاره کرد. همچنین الگوریتم پیشنهادی این قابلیت را دارد که هر زمان به تعداد

^۱ Roy

^۲ Yin

^۳ Li

^۴ Species Conservation

^۵ Species distance

^۶ Beasley

^۷ Sequential Niching

^۸ Zhanga

^۹ Roaming Technique

^{۱۰} Lung

^{۱۱} Liang

^{۱۲} Adaptive Elitist-population search technique

مورد نیاز بهینه‌های پیدا شده برای مسئله رسیدیم، متوقف شود و با استفاده از الگوریتم تپه‌نوردی، هر جواب ذخیره شده در حافظه را به بهینه سراسری برساند و به این ترتیب از اجرای بیش از نیاز الگوریتم جلوگیری نماید.

الگوریتم پیشنهادی مبتنی بر زیر جمعیت است و این زیر جمعیت‌ها به طور جداگانه تکامل می‌یابند. از این رو، استفاده از الگوریتم تکاملی که دارای مکانیزم مناسبی برای جستجوی فضای مسئله و همچنین نرخ همگرایی سریع است، می‌تواند باعث جستجوی بهتر و تسریع رسیدن به جوابها شود.

در ادامه در بخش دوم به الگوریتم بهینه‌سازی امواج آب اشاره می‌شود و در بخش سوم ساختار الگوریتم پیشنهادی معرفی می‌گردد. همچنین در بخش چهارم نتایج آزمایش‌ها و در بخش پنجم جمع‌بندی و نتیجه‌گیری ارائه می‌گردد.

۲- الگوریتم بهینه‌سازی امواج آب

در الگوریتم بهینه‌سازی امواج آب [23, 24]، هر موج یک راه حل مسئله است که دارای ارتفاع $h \in Z^+$ و طول موج $\lambda \in R^+$ است و تولید جواب‌های اولیه برای هر موج با یک ارتفاع و طول موج ثابت انجام می‌شود. روال اجرای الگوریتم امواج آب در شکل ۱ نشان داده شده است. در حین تکامل سه نوع حرکت در الگوریتم امواج آب وجود دارد:

۱-۲ انتشار^۱

در هر دوره تکامل هر موج X یکبار انتشار می‌یابد. با استفاده از این عملگر، طبق رابطه (۱) موقعیت جدید برای یک موج در هر بعد ایجاد می‌شود:

$$X'(d) = X(d) + rand(-1,1) \times \lambda L(d) \quad (1)$$

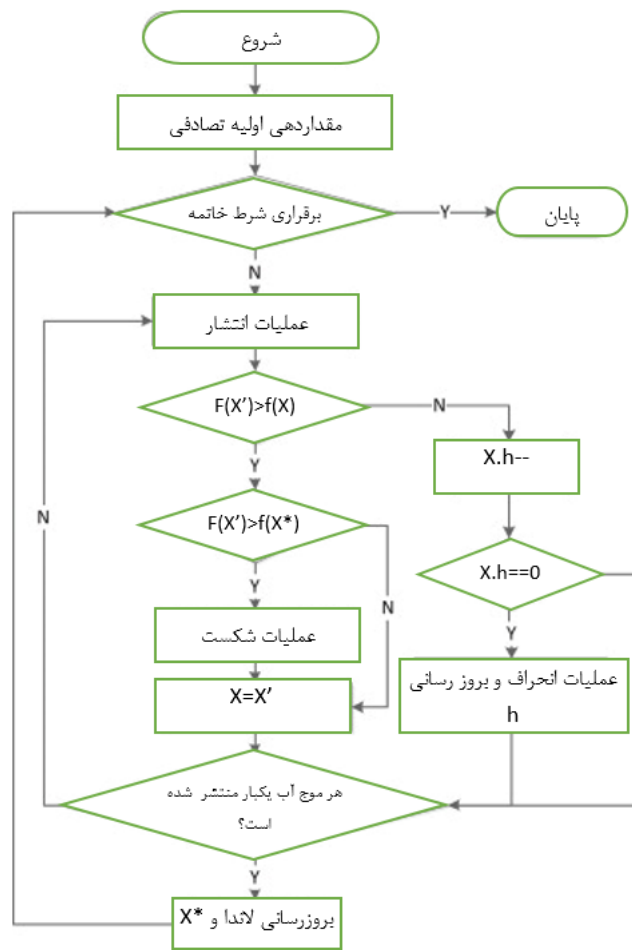
که در آن d شماره بعد، $L(d)$ طول d امین بعد و $rand(-1,1)$ مولد عدد تصادفی حقیقی در بازه $[-1, 1]$ است. تنظیم مقدار صحیح λ تاثیر به سزایی بر روی میزان اکتشاف^۲ الگوریتم امواج آب دارد. اگر مکان جدید موج بیرون از بازه فضا باشد، دوباره با یک موقعیت تصادفی در داخل فضای جستجو جایگزین می‌شود. برازندگی موقعیت جدید موج، بعد از انتشار محاسبه شده و با حالت قبل از انتشار مقایسه می‌شود و در صورتی برازندگی بیشتری داشته باشد جایگزین می‌شود، در غیر اینصورت موج در موقعیت قبل از انتشار باقی می‌ماند. اگر موجی با موقعیت جدید جایگزین شود در اینصورت ارتفاع آن نیز با مقدار $hMAX$ تنظیم شده و اگر موجی با موقعیت قبل باقی بماند از ارتفاع آن یک واحد کم می‌شود. در حالت کلی بعد از هر دوره تکامل طول موج نیز با رابطه (۲) بروزرسانی می‌شود:

$$\lambda = \lambda \times \alpha^{-(f(X)-f_{min}+\epsilon)/(f_{max}-f_{min}+\epsilon)} \quad (2)$$

در رابطه فوق، ϵ یک عدد مثبت بسیار کوچک است و برای رفع مشکل تقسیم بر صفر در نظر گرفته شده است، $f(X)$ برازندگی موج X و f_{min} و f_{max} به ترتیب بیشترین و کمترین برازندگی موج تاکنون هستند. α نیز ضریب کاهش موج است. در رابطه فوق تضمین می‌شود که موجی با برازندگی بالا دارای طول موج کوچکی است و بدین شکل در دامنه کمتری می‌تواند انتشار یابد که این از ملزومات الگوریتم است که باعث همگرایی به جواب‌های بهینه می‌شود [24].

^۱ Propagation

^۲ exploration



شکل ۱: الگوریتم استاندارد امواج آب

۲-۲ انحراف^۱

طبق رابطه (۳) داریم:

$$X'(d) = \text{Gaussian} \left(\frac{X'(d) + X(d)}{2}, \frac{|X^*(d) - X(d)|}{2} \right) \quad (3)$$

که در آن X' بهترین راه حلی است که تا کنون پیدا شده و $\text{Gaussian}(\mu, \sigma)$ مولد عدد تصادفی با توزیع نرمال با میانگین μ و انحراف معیار استاندارد σ است. پس از انحراف، ارتفاع موج با مقدار تنظیم مجدد hMAX و طول موج آن از رابطه (۴) بروزرسانی می‌شود [24]:

$$\lambda' = \lambda \frac{f(x)}{f(x')} \quad (4)$$

۳-۲ شکست^۲

در هر دوره تکامل، هر موج به امواج کوچکتری می‌شکند و این باعث تولید موقعیت جدید برای یک موج می‌شود که از رابطه (۵) محاسبه می‌شود:

$$X'(d) = X(d) + \text{Gaussian}(0,1) \times \beta L(d) \quad (5)$$

۱ Refraction

۲ Breaking

در رابطه فوق β ضریب شکست موج است [24] و تنظیم مقدار آن تاثیر به سزایی بر روی میزان استخراج الگوریتم دارد.

۳- الگوریتم پیشنهادی

در مورد الگوریتم بهینه‌سازی امواج آب تاکنون بهبودهایی انجام شده است [25, 26]، اما به مساله بهینه‌سازی چندگانه توسط این الگوریتم پرداخته نشده است. در الگوریتم پیشنهادی از همگرایی زودرس که در نتیجه از دست دادن تنوع گونه‌ها، ایجاد می‌شود به کمک زیر جمعیت‌هایی که بطور جداگانه تکامل پیدا می‌کنند، اجتناب می‌شود. همچنین در این الگوریتم بهینه‌ها در گروه‌های جداگانه که در حال تکامل و رشد هستند جستجو می‌شوند که این گروه‌ها کاملاً جداگانه هستند. یک معیار پایداری برای مشخص کردن گروه در حال پایدار شدن وجود دارد. این معیار از جمع شدن موج‌ها در اطراف بهینه‌ها جلوگیری می‌کند و به این ترتیب گروهی که تکامل خود را تا این آستانه انجام دهد به این معنی است که دارای بهینه‌ای است و باید این بهینه در حافظه‌ای خارجی ذخیره گردد. تعداد گروه‌های امواج آب پارامتری از الگوریتم است که نسبتی با تعداد بهینه‌ها ندارد و این موضوع انعطاف‌پذیری و مقاوم بودن الگوریتم پیشنهادی را نشان می‌دهد.

الگوریتم پیشنهادی از مکانیزم الگوریتم بهینه‌سازی امواج آب در جستجوی فضای مسئله بهره می‌برد و در کنار آن با الگوریتم تپه‌نوردی ترکیب شده و بدین صورت بعد از چندین تکرار، الگوریتم تپه‌نوردی جواب‌های ذخیره شده در حافظه را به مرکز بهینه می‌رساند. در حقیقت الگوریتم تپه‌نوردی بر روی هر یک از پاسخهای حاصل از الگوریتم بهینه‌سازی امواج آب به صورت جداگانه انجام می‌شود. بدین ترتیب سرعت همگرایی افزایش یافته و از تعداد ارزیابی‌های تابع بهینه‌سازی کاسته می‌شود. به لطف خصوصیت حافظه که در آن هر جواب معادل یک بهینه است، می‌توان شرط توقف الگوریتم را، تعداد تکرار الگوریتم، بدون ذخیره جواب تازه قرار داد. در ادامه طرح مسئله و ساختار الگوریتم ارائه می‌شود.

مسئله بهینه‌سازی به صورت رابطه (۶) است:

$$\begin{cases} \text{maximize } eval(x), \\ x \in S, \end{cases} \quad (6)$$

در آن S فضای راه حل و $eval(x)$ مقدار تابع برازندگی به ازای جواب x است.

تعریف ۱: جواب x از y بهتر است وقتی که مقدار تابع برازندگی x از y بهتر باشد ($eval(x) \geq eval(y)$) و می‌نویسیم:

$$x > y$$

تعریف ۲: گروه در حالت قبل از تکامل را با $group_i$ و گروه در حالت بعد از تکامل را با $group'_i$ نشان می‌دهیم و تعریف می‌کنیم:

x_i^* بهترین موج در گروه $group_i$ است و اپراتور B به صورتی تعریف می‌شود که موج‌های داخل گروه $group'_i$ بهتر باشند از x_i^* :

$$B: R \rightarrow R(R)$$

$$B(x_i^*) = \{x \in group'_i \mid x > x_i^*\}$$

تعریف ۳: میزان پایداری ($stable(group_i)$) با عددی بصورت رابطه (۷) تعیین می‌شود:

$$stable(group_i) = \frac{card B(x_i^*)}{card group_i} \quad (7)$$

در آن x_i^* بهترین جواب در گروه $group_i$ و $Card A$ نشان دهنده عناصر مجموعه A است. طبق رابطه (۷) $0 \leq stable(group) \leq 1$ و $stable(group) = 0$ اگر و تنها اگر هیچ موجی، در حالت بعد از تکامل گروه، بهتر از

بهترین موج گروه در حالت قبل از تکامل نباشد. همچنین اگر $stable(group) = 1$ آنگاه تمام موج‌ها در حالت بعد از تکامل گروه، بهتر از بهترین موج گروه در حالت قبل تکامل گروه بوده‌اند.

تعریف ۴: پارامتر α (پارامتر آرشیو) و پارامتر β (پارامتر پایداری گروه) هستند که $0 \leq \alpha, \beta \leq 1$. اگر $stable(group) \leq \alpha$ آنگاه بهترین موج در گروه، در حافظه ذخیره می‌شود. اگر $stable(group) \leq \beta$ آنگاه امواج آن گروه متزلزل است و در غیر این صورت گروه پایدار است.

۳-۱ مراحل الگوریتم پیشنهادی

مرحله ۱: مقداردهی اولیه: $t, s, N_{opt} = 0$ که t شمارنده تکرار دور الگوریتم و s حداکثر تعداد تکرار الگوریتم است. همچنین N_{opt} تعداد کل جواب‌های ذخیره شده در حافظه، N_{group} تعداد گروه‌های وزش باد، WG تعداد موج هر گروه، N_{gen2} تعداد نهایی دور تکامل الگوریتم و N_{gen1} تعداد تکامل اولیه هستند.

مرحله ۲: تولید امواج آب: تولید جواب‌های اولیه (موج‌ها) با استفاده از نگاشت لوجستیک [27] انجام می‌گیرد. نگاشت لوجستیک به علت آنکه یک بعدی است برای تولید اعدادی در همه ابعاد مسئله مناسب است. در نهایت اعداد تولید شده در کنار یکدیگر قرار می‌گیرند.

هر موج بصورت یک آرایه تعریف می‌شود:

$$W_i = [x_1, x_2, \dots, x_n]$$

مرحله ۳: تعیین برازندگی هر جواب ممکن (موج): برای هر موج میزان برازندگی با FW_i تعریف می‌شود:

$$FW_i = eval(W_i)$$

مرحله ۴: تقسیم موج‌ها در گروه‌های مختلف: در این مرحله موج‌های تولید شده در فضا به تعداد گروهی که تعیین شده است تقسیم شده و هر کدام با نام "عضو گروه امواج آب با وزش باد" شماره ۱ تا N_{group} برچسب‌گذاری می‌شوند. برای سادگی محاسبات در الگوریتم پیشنهادی تعداد امواج هر گروه را ثابت در نظر گرفته‌ایم.

مرحله ۵: انتشار: در هر دوره تکامل، هر موج یکبار انتشار می‌یابد. با استفاده از عملگر رابطه ۱ موقعیت جدیدی برای یک موج در هر بعد ایجاد می‌شود. بعد از انتشار برازندگی موقعیت جدید موج محاسبه شده و با حالت قبل از انتشار مقایسه می‌شود و در صورتی که برازندگی بیشتری داشته باشد، جایگزین می‌شود. در غیر اینصورت موقعیت موج همان مقدار قبل از انتشار باقی می‌ماند. در حالت کلی بعد از هر دوره تکامل، طول موج هر موج نیز با رابطه ۲ بروزرسانی می‌شود.

مرحله ۶: انحراف: به کمک رابطه ۳ اعمال می‌گردد و پس از انحراف ارتفاع و طول موج همانند آنچه در بخش ۲-۲ گفته شد بروز می‌شود.

مرحله ۷: شکست: در هر دوره تکامل هر موج به امواج کوچک‌تری می‌شکند و این باعث تولید موقعیت جدید برای یک موج می‌شود که از رابطه ۵ محاسبه می‌شود.

مرحله ۸: هزینه $eval'(x)$ هر موج در حالت بعد از تکامل برای هر گروه $group'_i(t)$ که $i = 1, \dots, N_{group}$ محاسبه و پایداری گروه طبق تعریف ۳ محاسبه می‌شود.

مرحله ۹: برای هر گروه اگر $stable(group_i(t)) \leq \alpha$ آنگاه x_i^* برای ذخیره به حافظه داده می‌شود.

مرحله ۱۰: برای هر گروه اگر $stable(group_i(t)) \leq \beta$ آنگاه $group_i(t)$ گروه متزلزل است و در غیر این صورت $group_i(t)$ گروه پایدار است.

مرحله ۱۱: در هر کدام از گروه‌های $group_i(t)$ که متزلزل هستند تمام موج‌ها از بین می‌روند و با استفاده از نگاشت آشوب دوباره تولید می‌شوند.

مرحله ۱۲: امواج گروه‌ها در دور بعدی تکامل شامل رابطه (۸) هستند:

$$\begin{aligned} all_{group(t+1)} = & \left\{ group_i(t) | group_i(t) \text{ باشد متزلزل} \right\} \\ & \cup \left\{ group'_i(t) | group_i(t) \text{ باشد پایدار} \right\} \\ & \text{و } t \leftarrow t + 1 \end{aligned} \quad (8)$$

در این مرحله، کل گروه‌هایی که در دور بعدی الگوریتم باید تکامل پیدا کنند مشخص می‌شوند. این گروه‌ها شامل گروه‌های پایداری هستند که هنوز به بهینه‌ای همگرا نشده و هنوز باید تکامل پیدا کنند. همچنین گروه‌های متزلزلی که به بهینه‌ای همگرا شده‌اند و باید حذف شده و دوباره تولید شوند وارد دور بعد می‌شوند.

مرحله ۱۳: اگر $t \leq N_{gen}$ و N_{opt} در S تکرار ثابت بود آنگاه به پایان الگوریتم می‌رویم و در غیر اینصورت به مرحله ۵ باز می‌گردیم. همچنین اگر $t > N_{gen}$ آنگاه به مرحله ۱۴ و پایان الگوریتم می‌رویم.

مرحله ۱۴: هر کدام از جواب‌های ذخیره شده در حافظه را با الگوریتم تپه‌نوردی به قله می‌رسانیم. لازم به توضیح است که الگوریتم تپه‌نوردی بر روی هر یک از پاسخهای حاصل از الگوریتم بهینه‌سازی امواج آب به صورت جداگانه انجام می‌شود.

۲-۳ مرحله ذخیره در حافظه

در حافظه فقط جواب‌هایی ذخیره می‌شوند که با جواب‌های دیگر بر روی یک بهینه قرار نداشته باشند. هر جوابی که می‌خواهد ذخیره شود با تمام جواب‌هایی که از قبل ذخیره شده‌اند مقایسه می‌شود. این مقایسه با نزدیکترین جواب‌ها از نظر فاصله اقلیدسی انجام می‌گیرد و بدین ترتیب اگر این جواب با هیچ یک از جواب‌های دیگر موجود در حافظه بروی یک بهینه قرار نداشته باشد، به حافظه اضافه می‌شود. در غیر این صورت اگر این جواب از نزدیکترین جواب موجود در حافظه بهینه‌تر باشد، جایگزین آن می‌شود و اگر بهتر نباشد، نادیده گرفته می‌شود. برای تعیین اینکه آیا دو نقطه در یک بهینه هستند و یا خیر، از روش تپه - دره استفاده می‌شود [19].

۳-۳ روش تپه-دره

تابع تناسب توپولوژیکی^۱ توسط اورسم^۲ [28] معرفی گردید. همچنین لین^۳ تکنیکی برای تشخیص نیچ ارائه کرد [29]. از طرفی برای تعیین اینکه آیا دو نقطه بر روی یک بهینه هستند در [30] تحقیقاتی انجام گرفته است. در روش تپه-دره فاصله بین دو نقطه در حافظه به چند قسمت تقسیم می‌شود و با بدست آوردن مقدار تابع بهینه این نقاط و مقایسه آن با دو نقطه اولیه، این مسئله که آیا این دو نقطه اولیه بر روی یک بهینه هستند و یا خیر تعیین می‌شود. اگر i_q و i_p دو نقطه دلخواه در فضای جستجو باشند، آنگاه شبه کد تپه-دره $HV(i_q, i_p)$ به شکل زیر تعریف شود:

$Hill - valley(i_q, i_p, samples)$

1. $min_{fit} = \min(fitness(i_p), fitness(i_q))$

2. for $j = 1$ to $sampleslength$

point i_p and i_q . 3. Calculate point $i_{interior}$ on the line between the

4. if ($min_{fit} > fitness(i_{interior})$)

5. return 1

6. end if

^۱ fitness-topology function

^۲ Ursem

^۳ Lin

7. end for
8. return 0

در این تابع اگر همه نقاط درونی بزرگتر از مینیمم برازندگی i_q و i_p باشند، پاسخ صفر و در غیر این صورت یک است. آرایه $samples$ برای محاسبه نقاط داخلی^۱ در تابع تپه-دره استفاده می‌شود. نقطه $i_{interior}$ در تابع برای دو بعد به صورت رابطه (۹) محاسبه می‌شود:

$$i_{interior} = (i_{px} + (i_{qx} - i_{px}).samples[j]), \quad (9)$$

$$i_{py} + (i_{qy} - i_{py}).samples[j])$$

که در آن j ، بیاتگر j امین عضو آرایه است.

۳-۴ الگوریتم تپه نوردی

در الگوریتم تپه نوردی از محل جواب ذخیره شده در حافظه، تعدادی همسایه برای حالت فعلی تولید می‌شود و از میان حالت‌های همسایه بهترین آن‌ها انتخاب شده و جایگزین حالت فعلی می‌گردد. این روند تا رسیدن به مرکز بهینه ادامه پیدا می‌کند [1].

برای پیاده‌سازی روش تپه‌نوردی نیاز به تابع Neighbor است که همسایه‌های حالت فعلی را تولید می‌کند. الگوریتم نیز زمانی به اتمام می‌رسد که حالت بهتری نسبت به حالت فعلی تولید نشود. شبه کد زیر نحوه پیاده‌سازی الگوریتم تپه‌نوردی را نشان می‌دهد:

Procedure HillClimbing

Generate a solution (s')

Best = S'

Loop

S = Best

S' = Neighbors (S)

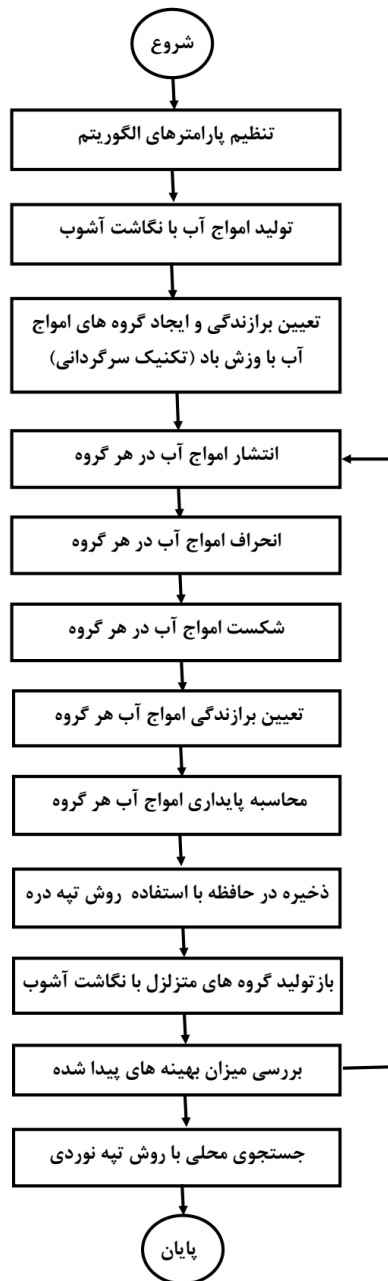
Best = SelectBest (S')

Until stop criterion satisfied

End

شکل ۲ شمای کلی الگوریتم پیشنهادی را نمایش می‌دهد.

^۱ interior



شکل ۲: شمای کلی اجرای مراحل الگوریتم پیشنهادی

۴- نتایج

الگوریتم پیشنهادی بر روی سیزده تابع محک جدول ۱ مورد ارزیابی قرار گرفته است [31].

جدول ۱: توابع محک در آزمایش‌ها

نام تابع	تعداد کل بهینه ها	تعریف ریاضی و شروط
۱ Goldberg F2	۵	$f_1(x) = e^{-2\ln(2)\left(\frac{x-0.1}{0.8}\right)^2} \sin^6(5\pi x), x \in [0,1];$
۲ function Ten optima	۱۰	$f_2(x) = \sum_{j=1}^5 j \cos((j+1)x + j), x \in [0,10];$

$f_3(x) = \ln(x) \cdot (\sin(e^x) + \sin(3x)), x \in [0,4];$	۹	Optimal non-uniform distribution function	۳
$, x \in [0,10]; f_4(x) = \sum_{i=1}^{10} \frac{1}{(K_i(x-a_i))^2 + C_i}$	۸	Optimal difficult function	۴
$f_5(x, y) = \left(y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6\right)^2 + 10 \left(1 - \frac{1}{8\pi}\right) \cos(x) + 10, x \in [-5,10] \text{ and } y \in [-5,10]$	۳	Branin RCOS	۵
$f_6(x, y) = x^2 + 2y^2 - 0.3 \cos(3\pi x) - 0.4 \cos(4\pi y) + 0.7$ $x \in [-2,2] \text{ and } y \in [-2,2]$	۹	Bohachevsky F1	۶
$f_7(x, y) = - \left[\left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2) \cdot y^2 \right],$ $x \in [-1.9,1.9] \text{ and } y \in [-1.1,1.1]$	۶	Six-hump Camel Back	۷
$f_8(x, y) = 3 \sin(0.5\pi x + 0.5\pi) \cdot \frac{2 - \sqrt{(x^2 + y^2)}}{4},$ $x \in [-2,2] \text{ and } y \in [-2,2]$	۵	Ursem F4	۸
$f_9(x, y) = -(x^2 + y - 11)^2 - (x + y^2 - 7)^2,$ $x \in [-6,6] \text{ and } y \in [-6,6]$	۴	Himmelblau	۹
$f_{10}(x_1, x_2) = 1 + \sin^2 x_1 + \sin^2 x_2 - 0.1 \cdot e^{(-x_1^2 - x_2^2)}, x_1 \in [-5,5] \text{ and } x_2 \in [-5,5]$	۱۶	Sixteen optima function	۱۰
$f_{11}(x) = \sin^6(5\pi x), x \in [0,1];$	۵	Deb's function	۱۱
$f_{12}(x) = e^{-2 \ln(2) \left(\frac{x-0.1}{0.9}\right)^2} \sin^6(5\pi x), x \in [0,1];$	۵	Deb's decreasing function	۱۲
$f_{13}(x) = \frac{1}{1 + x^6 - 1 }, x \in \mathbb{C}, x = x_1 + ix_2 \in [-2,2]$	۶	Roots function	۱۳

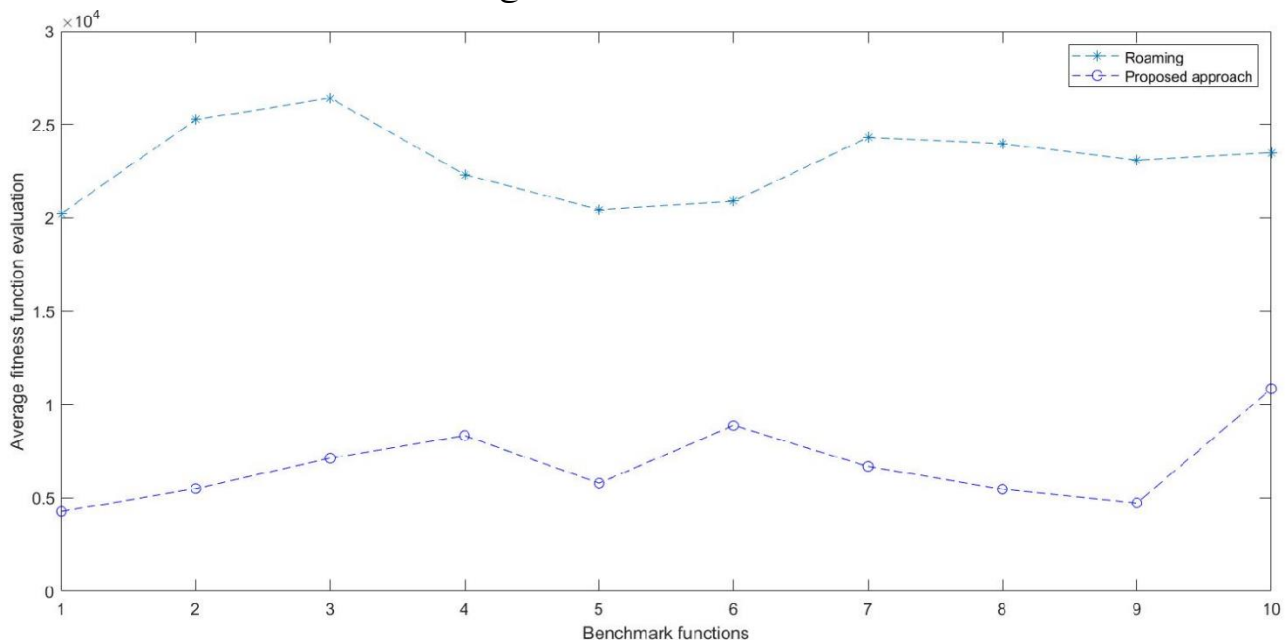
تابع یک تاده از توابع استاندارد در بهینه‌سازی چندگانه هستند و توابع یازده تا سیزده نیز بخاطر فضای مسئله خاصی که دارند مورد توجه هستند. پارامترهای این الگوریتم به این صورت است که، تعداد موج ۱۰۰ و تعداد گروه‌ها ۱۰ در نظر گرفته شده است. تعداد تکرار اولیه الگوریتم را ۱۰ و تعداد تکرار نهایی را ۵۰ دور و حداکثر تعداد تکرار الگوریتم که در آن جوابی به حافظه اضافه نگردد را ۱۰ تکرار تنظیم کرده‌ایم. حداقل میزان پایداری را ۰ و حداکثر تعداد بهینه‌های پیدا شده را بی‌نهایت در نظر گرفتیم. در روش پیشنهادی تعداد جوابهای بهینه حاصل پس از الگوریتم تپه نوردی به عنوان تعداد نقاط بهینه در نظر گرفته می‌شود. به این ترتیب که اگر نقاط در مجاورت هم باشند، پس از تپه نوردی بر روی یک پاسخ نگاشت می‌شوند و پاسخهای تکراری حذف می‌شوند. مقادیر پارامترهای فوق با توجه به تنظیماتی که معمولاً مسائل چندوجهی انجام می‌شود در نظر گرفته شده است. همچنین ضریب کاهش موج ۱ و ضریب شکست موج از ۰,۲۵ با کاهش خطی تا ۰,۱ و بیشترین ارتفاع موج ۵ و طول موج ۰,۵ در نظر گرفته شده است. این پارامترها نیز با توجه به تنظیمات پایه مقاله [23] مقداردهی شده است. در جدول (۲) نتایج الگوریتم برای توابع تست در ۳۰ بار اجرا با $s = 10$ نشان داده شده است.

جدول ۲: نتایج الگوریتم پیشنهادی شامل میانگین و انحراف معیار تعداد بهینه‌ها و میانگین و انحراف معیار تعداد ارزیابی تابع در مقایسه با تکنیک سرگردانی با $s = 10$ در ۳۰ اجرا

توابع ارزیابی	تکنیک سرگردانی [19]		الگوریتم پیشنهادی	
	میانگین بهینه‌ها	میانگین ارزیابی تابع	میانگین بهینه‌ها	میانگین ارزیابی تابع
Goldberg F2	0 ± 5	3 ± 20253	0 ± 5	0 ± 4270
function Ten optima	0 ± 10	0 ± 25250	0 ± 10	5 ± 5493
Optimal non-uniform function distribution	0 ± 9	4 ± 26482	0 ± 9	2 ± 7103
Optimal difficult function	0 ± 8	4 ± 22281	0 ± 8	4 ± 8326
Branin RCOS	0 ± 3	4 ± 20488	0 ± 3	5 ± 5783
Bohachevsky F1	0 ± 9	2 ± 20874	0 ± 9	0 ± 8873
Six-hump Camel Back	0 ± 6	4 ± 24325	0 ± 6	0 ± 6684
Ursem F4	0 ± 5	3 ± 24007	0 ± 5	4 ± 5468
Himmelblau	0 ± 4	4 ± 23158	0 ± 4	4 ± 4655
Sixteen optima function	0 ± 16	5 ± 23463	0 ± 16	3 ± 10863

بدون وابستگی به تخمینی از فاصله بهینه‌ها از یکدیگر

شکل ۳ میانگین تعداد ارزیابی تابع برازندگی را به ازای دو روش سرگردانی [19] و روش پیشنهادی بر روی ۱۰ تابع ارزیابی نشان می‌دهد. در این شکل شماره‌های محور افقی بیانگر شماره تابع ارزیابی است. در شکل زیر جدا از تفاوت قابل توجه روش پیشنهادی نسبت به روش سرگردانی، مشخص است که هر دو روش بر روی توابع مختلف نسبتاً یکسان است.



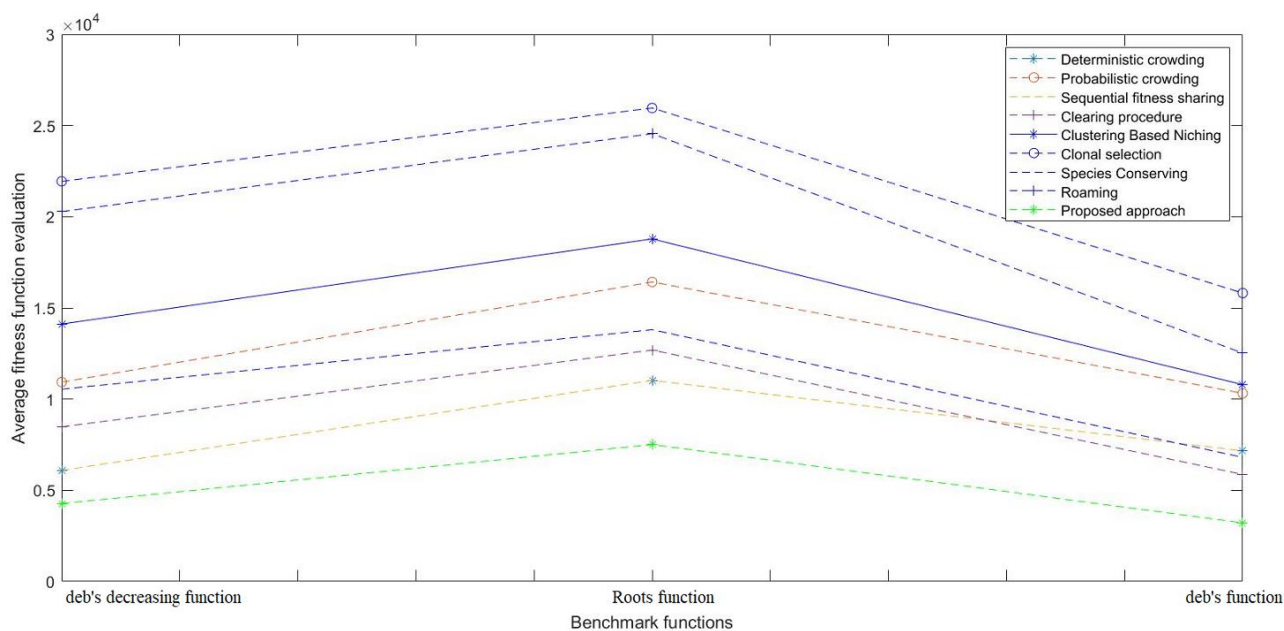
شکل ۳: مقایسه روش پیشنهادی و روش سرگردانی بر اساس متوسط تعداد ارزیابی تابع برازندگی بر روی ۱۰ تابع ارزیابی

در جدول ۳ و شکل ۴ مقایسه نتایج بدست آمده برای توابع Deb's deceasing, Deb's و Roots با سایر روش‌های ارائه شده در [19, 32, 33] نشان داده شده است. مقایسه الگوریتم پیشنهادی بر روی توابع تست در مقایسه با سایر روش‌های

بهینه‌سازی چندگانه نشان می‌دهد که روش پیشنهادی نسبت به روش سرگردانی که ساختاری شبیه به الگوریتم پیشنهادی دارد، نتایج بسیار خوبی دارد. همچنین در مقایسه با سایر روش‌های پایه در بهینه‌سازی چندگانه نتیجه بهتری بدست آمده است. در مقایسه با روش تطبیقی جمعیت نخبگان که از الگوریتم‌های موفق در بهینه‌سازی چندگانه است، نتیجه قابل قبولی داشته است و این با در نظر گرفتن این مورد است که الگوریتم پیشنهادی وابسته به پارامتر حداقل فاصله بین بهینه‌ها از یکدیگر که از اطلاعات قبلی از تابع بهینه‌سازی بدست می‌آید نیست. این مورد الگوریتم پیشنهادی را قادر می‌سازد تا در مسائل مختلف دنیای واقعی که اطلاعات کاملی از مسئله بهینه‌سازی وجود ندارد، به خوبی عمل نماید.

جدول ۳: مقایسه نتایج بدست آمده از روش پیشنهادی برای توابع Deb's decreasing و Roots با سایر روشها (میانگین و انحراف معیار برای ۳۰ بار اجرا)

روش‌ها	Deb's decreasing function		Roots function		Deb's function		
	میانگین بهینه‌ها	میانگین ارزیابی تابع	میانگین بهینه‌ها	میانگین ارزیابی تابع	میانگین بهینه‌ها	میانگین ارزیابی تابع	
وابسته به تخمینی از فاصله بهینه‌ها از یکدیگر	Deterministic crowding [3]	۰,۱۳±۲,۵۴	۱۲±۶۰,۲۲	۰,۱±۴,۲۱	۲۰±۱۱۰,۲۳	۰±۵	۸±۷۱۵۲
	Probabilistic crowding [4]	۰,۰۵±۴,۷۲	۳۲±۱۰,۹۴۳	۰,۲±۴,۹۸	۲۷±۱۶۳,۹۵	۰±۵	۶±۱۰۳,۰۸
	Sequential fitness sharing [32]	۰,۰۲±۴,۷۷	۱۸±۱۲۷,۹۰	۰,۲۵±۴,۸۵	۱۵±۱۴۴,۱۸	۰±۵	۹±۹۹۳۲
	Clearing procedure [32]	۰,۲۱±۴,۷	۸±۸۴,۶۹	۰±۶	۱۰±۱۲۶,۹۳	۰±۵	۶±۵۸,۶۱
	Clustering Based Niching [33]	۰,۱۴±۴,۷	۲۱±۱۴۱,۲۰	۰,۰۵±۴,۷۲	۱۶±۱۸۷,۵۰	۰±۵	۴±۱۰۷,۸۵
	Clonal selection [33]	۰±۵	۱۲±۲۱۹,۲۵	۰,۳±۵,۶۳	۱۵±۲۵۹,۴۷	۰±۵	۹±۱۵۸,۱۰
	Species Conserving [33]	۰,۲±۴,۸۵	۱۵±۱۰۵,۵۲	۰±۶	۱۳±۱۳۸,۰۹	۰±۵	۱۲±۶۷,۸۴
بدون وابستگی	Roaming [19]	۰±۵	۱۸±۲۰۲,۶۲	۰±۶	۲۰±۲۴۵,۴۱	۰±۵	۴±۱۲۵,۲۱
	الگوریتم پیشنهادی	۰±۵	۸±۴۲,۷۵	۰±۶	۱۱±۷۵,۳۰	۰±۵	۳±۳۲,۱۵



شکل ۴: مقایسه روش پیشنهادی با برخی دیگر از روشهای بهینه‌سازی چندگانه بر روی توابع Deb's، Deb's decreasing و Roots

۵- جمع‌بندی

در این مقاله، الگوریتم جدید بهینه‌سازی چندگانه تکاملی مبتنی بر بهینه‌سازی امواج آب ارائه شده است. در این الگوریتم بهینه‌ها در قالب گروه‌های امواج آب جداگانه که در حال تکامل هستند جستجو می‌شوند. برای این کار از یک معیار پایداری برای مشخص کردن اینکه گروه امواج دارای بهینه است یا نه استفاده شده است. گروه‌ها جداگانه تکامل می‌یابند تا بهینه‌ای که تا به حال شناخته نشده را بیابند. بهینه‌های شناخته شده در حافظه ذخیره می‌شوند و گروه موحی که دیگر نمی‌تواند بهینه جدیدی را پیدا کند، در ناحیه‌ای دیگر از فضای جستجو با نگاشت آشوب پخش می‌شود. الگوریتم پیشنهادی در ترکیب با الگوریتم تپه‌نوردی استفاده می‌شود. به این ترتیب که تا تعداد تکراری مشخص الگوریتم اجرا شده و بعد از آن الگوریتم تپه‌نوردی جواب‌ها را به قله می‌رساند. بدین صورت میتوان کنترلی بروی اجرای الگوریتم و تعداد بهینه‌های بدست آمده داشت. الگوریتم پیشنهادی با توجه به ساختاری که دارد می‌تواند در کاربردهای دیگری از جمله در حل مسئله سینماتیک معکوس و بهینه‌سازی الکترومغناطیس استفاده شود [34]. همچنین به نظر می‌رسد الگوریتم پیشنهادی با ترکیب روش‌های حوشه‌بندی از جمله حوشه‌بندی بر اساس چگالی جواب‌ها [35, 36]، بتواند با به حداقل رساندن ارزیابی تابع، زمان اجرا را در توابع پرهزینه پایین‌تر بیاورد.

۶- مراجع

- [1] S.W. Mahfoud, "A comparison of parallel and sequential niching method", in: Proceedings of the Sixth International Conference on Genetic Algorithms, ICGA, pp. 136-143, 1995.
- [2] Jong, K.A.D, An analysis of the behaviour of a class of genetic adaptive systems. PhD thesis, University of Michigan, 1975
- [3] S. Mahfoud, "Crowding and preselection revisited", in: Parallel Problem Solving from Nature, vol. 2, pp. 27-37, 1992

- [4] O. Mengsheel, D. Goldberg, "Probabilistic crowding: deterministic crowding with probabilistic replacement", in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 99, pp. 409–416, 1999.
- [5] Ole J. Mengshoel, David E. Goldberg, *The Crowding Approach to Niching in Genetic Algorithms*, by the Massachusetts Institute of Technology, 2000
- [6] A. Pérowski, "A clearing procedure as a niching method for geneticalgorithms", in: Proceedings of Third IEEE International Conference on Evolutionary Computation, ICEC'96, IEEE Press, Piscataway, NJ, pp. 798–803, 1996
- [7] D.E. Goldberg, J. Richardson, "Genetic algorithms with sharing for multimodal function optimization", in: Proceedings of the second International Conference on Genetic Algorithms, pp. 41–49, 1987
- [8] Miller, B.L.; Shaw, M.J.; , "Genetic algorithms with dynamic niche sharing for multimodal function optimization," *Evolutionary Computation*, 1996., Proceedings of IEEE International Conference on , vol., no., pp.786-791, 20-22 May 1996
- [9] D.E. Goldberg, L. Wang, "Adaptive niching via coevolutionary sharing", in: *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, pp. 21–38, 1998
- [10] G. Harik, "Finding multi-modal solutions using restricted tournament selection", in: Proceedings of the Sixth International Conference on Genetic Algorithms, ICGA-95, pp. 24–31, 1997.
- [11] R. Roy, I.C. Parmee, "Adaptive restricted tournament selection for the identification of multiple sub-optima in a multi-modal function", in: *Selected Papers from AISB Workshop on Evolutionary Computing*, Springer-Verlag, London, vol. 1143, pp. 236–256, 1996
- [12] X. Yin, N. Gernay, "A fast genetic algorithm with sharing scheme using cluster analysis methods in multi-modal function optimization", in: Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, pp. 450–457, 1993.
- [13] J.-P. Li, M.E. Balazs, G.T. Parks, P.J. Clarkson, "A species conserving genetic algorithm for multi-modal function optimization", *Evolutionary Computation*, Vol.10, No.3, pp. 207–234, 2002.
- [14] J.-P. Li, A. Wood, "Random search with species conservation for multimodal functions", in: Proceedings of the Eleventh Conference on Congress on Evolutionary Computation, Trondheim, Norway, pp. 3164–3171, 2009.
- [15] D. Beasley, D.R. Bull, R.R. Martin, "A sequential niche technique for multimodal function optimization", vol. 1, No.2, pp. 101–125, 1993.
- [16] Jun Zhanga, De-Shuang Huang, Tat-Ming Lok, Michael R. Lyu, "A novel adaptive sequential niche technique for multimodal function optimization", *Neurocomputing*, vol.1, pp.75-80, 2006.
- [17] S. Xiong, W. Gong, K. Wang, "An adaptive neighborhood-based speciation differential evolution for multimodal optimization", *Expert Systems with Applications*, vol. 211, 2023, 118571.
- [18] S. Agrawal, A. Tiwari, "Solving multimodal optimization problems using adaptive differential evolution with archive", *Information Sciences*, vol. 612, 2022, Pages 1024-1044.
- [19] Lung, R.I.; Dumitrescu, D.; , "A new subpopulation model for evolutionary multimodal optimization," *Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2005. Seventh International Symposium on* , vol., no., pp. 4 pp., 25-29 Sept. 2005.
- [20] K.S.Leung, Y. Liang , "Adaptive Elitist-Population Based Genetic Algorithm for Multimodal Function Optimization", in: GECCO, pp. 1160–1171, 2003.

- [21] Yu Liu, Zhaofa Yan, Wentao Li, Mingwei Lv, and YuanYao, “An Automatic Niching Particle Swarm for Multimodal Function Optimization”, Springer-Verlag Berlin Heidelberg ,2010.
- [22] Swagatam Das, Sayan Maity, Bo-Yang Qu, P.N. Suganthan, “Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art”, Swarm and Evolutionary Computation, 2011.
- [23] Yu-Jun Zheng and Bei Zhang.2015. A Simplified Water Wave Optimization Algorithm. IEEE,978-1-4799-7492-4/15/\$31.00.
- [24] Y. J. Zheng, “Water wave optimization: A new nature-inspired metaheuristic,” Computers & Operations Research, vol. 55, pp. 1–11, 2015.
- [25] Z. Yan, J. Zhang, J. Tang, Path planning for autonomous underwater vehicle based on an enhanced water wave optimization algorithm, Mathematics and Computers in Simulation, vol. 181, 2021, Pages 192-241.
- [26] F Zhao, D. Shao, L. Wang, T. Xu, N. Zhu, An effective water wave optimization algorithm with problem-specific knowledge for the distributed assembly blocking flow-shop scheduling problem. Knowledge-Based Systems, vol. 243, 2022, 108471.
- [27] S. Taghavi Zadeh Forotan.2014. Cuckoo improved algorithm based on chaos theory. Master thesis artificial intelligence. Islamic Azad University of Mashhad.
- [28] R.K. Ursem, “Multinational evolutionary algorithms”, in: Proceedings of the Congress on Evolutionary Computation, vol. 3, pp. 1633–1640, 1999.
- [29] C.Y.lin,W.H.Wu, Niche Identification Techniques in multimodal Genetic search with Sharing Scheme. Advances In Engineering, vol.33,No.11,pp.779-791,2002.
- [30] J. Zhang, et al., Multi-sub-swarm particle swarm optimization algorithm for multimodal function optimization. Proc. Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp. 3215-3220, 2007.
- [31] Nadia Nekouie, Mahdi Yaghoobi. 2016. A new method in multimodal optimization based on firefly algorithm. DOI 10.1007/s10462-016-9463-0
- [32] K.S.Leung, Y. Liang, genetic algorithm with Adaptive elitist-population strategies for multimodal function optimization. In: Applied soft Computing,Vol.11,Issue.2,pp. 2017–2034 ,2011.
- [33] G. Singh, K. Deb, Comparison of multi-modal optimization algorithms based on evolutionary algorithms. in: GECCO,pp.1305-1312,2008.
- [34] F. Ye, W. Qi, J. Xiao. Research of niching genetic algorithms for optimization in electromagnetics. Science Direct, Procedia Engineering. Vol.16. pp. 3215-3220.2011.
- [35] de Arruda Pereira M, Davis Júnior CA, Gontijo Carrano E, de Vasconcelos JA. A niching genetic programming-based multi-objective algorithm for hybrid data classification. Science Direct, Neuro computing. vol.133, pp.342–357.2014.
- [36] H.Z. Yang, F.C. Li, C.M. Wang. A density clustering based niching genetic algorithm for multimodal optimization. Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 3, pp.1599–1604.2005.