

تقویت الگوریتم بهینه ساز گروه جستجو با استفاده از تنظیم هوشمند پارامتر حداکثر فاصله تعقیب

سهیلا عابدینی^۱

^۱ دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، soheylaabedini@aut.ac.ir

چکیده

در این مقاله به معرفی و بهبود یکی از الگوریتم های تکاملی (EA) الهام گرفته از طبیعت می پردازیم. این الگوریتم جدید، بهینه ساز گروه جست و جو (GSO) نام دارد که از رفتار جستجوی حیوانات برای پیدا کردن غذا الهام گرفته است. چارچوب استفاده شده در این الگوریتم مدل تولید کننده - برداشت کننده می باشد. به منظور پیاده سازی این مدل، در بهینه سازی گروه جستجو از یک مکانیسم پویای محیط با زوایای مختلف استفاده شده است. الگوریتم بهینه ساز گروه جستجو شامل سه نوع عضو می باشد: عضو تولیدکننده، عضو برداشت کننده و عضو تکاور. در این الگوریتم پارامتر حداکثر فاصله تعقیب رابطه مستقیم با طول قدم برداشته شده توسط عضوهای تولیدکننده دارد. این پارامتر تأثیر مستقیم در قدرت تولیدکنندگی اعضاء تولیدکننده دارد. در این مقاله، به منظور بهبود الگوریتم بهینه ساز گروه جستجو پارامتر حداکثر فاصله تعقیب به صورت تطبیقی تنظیم شده است. رویکرد تطبیقی جدید، با استفاده از نرخ موفقیت جمعیت به عنوان بازخورد، پارامتر حداکثر فاصله تعقیب را تنظیم می کند. با هدف ارزیابی مستقیم بهینه سازی گروه جستجو، این الگوریتم بر روی مجموعه های غنی از تابع های بهینه سازی که از مسائل دنیای واقعی الهام گرفته اند تست شده است. تابع های محک شامل دو نوع کلی مسائل تک قله ای و چند قله ای می باشند. آزمایش ها در دو مقیاس ۳۰ بعد و ۳۰۰ بعد (ابعاد بالا) انجام شده اند. نتایج آزمایش ها نشان می دهد که مکانیسم پویای محیط الگوریتم بهینه ساز گروه جستجو تابع های محک تک قله ای ۳۰ بعدی را به سادگی، تابع های محک چند قله ای ۳۰ بعدی را با دقت بالا و تابع های محک ۳۰۰ بعدی را نیز به سرعت بهینه می سازد.

واژه های کلیدی

بهینه سازی؛ الگوریتم تکاملی، هوش گروهی؛ بهینه ساز گروه جستجو.

۱- مقدمه

الگوریتم های تکاملی همواره سعی داشته اند از رفتار طبیعی تکامل الهام بگیرند. الهام از طبیعت مهمترین خصیصه این نوع الگوریتم ها می باشد. چندین الگوریتم تکاملی معروف که پایه روش جستجوی آنها از موجودات زنده الهام گرفته است عبارتند از: الگوریتم بهینه ساز گروه ذرات (PSO)^۴ [۱] که از حرکت دسته جمعی پرندگان الهام گرفته است. الگوریتم کلونی مورچه ها (ACO)^۵ [۲] که از رفتار مورچه ها برای جستجوی غذا میان لانه و غذا برگرفته شده است. الگوریتم کلونی زنبورهای مصنوعی (ABC)^۶ [۳] که بر مبنای نحوه جستجوی زنبورها برای غذا بنا شده است.

محاسبات تکاملی (EC)^۱ زیرمجموعه ای از هوش محاسباتی می باشند که شامل مسائل بهینه سازی^۲ می باشند. الگوریتم های تکاملی (EAs)^۳ زیرمجموعه ای از محاسبات تکاملی (EC) می باشد. تکامل در طبیعت باعث به وجود آمدن موجودات مناسب برای محیط می گردد. الگوریتم های تکاملی سعی بر استفاده از الگوی طبیعت برای تولید موجودات برتر دارند. الگوریتم های تکاملی مبتنی بر جمعیت می باشند و از دسته الگوریتم های سعی و خطا می باشند.

^۴ Particle Swarm Optimizer (PSO)

^۵ Ant Colony Optimization (ACO)

^۶ Artificial Bee Colony (ABC)

^۱ Evolutionary Computation (EC)

^۲ Optimization Problems

^۳ Evolutionary Algorithms (EAs)

غذا، جفت، لانه و غیره توصیف شود. از سوی دیگر زندگی گروهی یک دسته از حیوانات باعث اشتراک گذاری اطلاعات و همکاری آنها با یکدیگر می شود. این رفتار گروهی اجازه می دهد که نرخ جستجوی موفق گروه افزایش یابد و همچنین واریانس موفقیت گروه کاهش یابد.

بهینه‌ساز گروه جستجو برای اولین بار در سال ۲۰۰۶ توسط هی و همکاران برای آموزش شبکه عصبی مصنوعی (ANN)^۶ استفاده شد [۵]. با استفاده از این الگوریتم تکاملی دو مسأله دسته بندی دنیای واقعی با موفقیت حل شد. در ادامه تحقیقات بر روی این الگوریتم در همان سال نسخه‌ای از این الگوریتم برای بهینه‌سازی تابع^۷ معرفی شد [۶]. در سال ۲۰۰۹ نیز هی و همکاران [۴] نسخه‌ای جامع از بهینه‌ساز گروه جستجو را چاپ نمودند.

پس از معرفی بهینه‌ساز گروه جستجو توسط هی و همکاران [۴]–[۶]، نویسندگان بسیاری به این الگوریتم تکاملی علاقمند شدند. در سال ۲۰۱۰ از الگوریتم بهینه‌سازی گروه جستجو برای طراحی پوسته‌های دولایه توری^۸ استفاده شده است [۷]. همچنین در سال ۲۰۱۱ یک الگوریتم بهینه‌سازی گروه جستجو مبتنی بر درونیایی درجه دو معرفی شد [۸]. بعلاوه در سال ۲۰۱۲ یک الگوریتم بهینه‌سازی گروه جستجو مبتنی بر دنیای کوچک (SW)^۹ معرفی شد. دنیای کوچک یک ساختار شبکه اجتماعی است که برای تبادل اطلاعات میان اعضاء جامعه استفاده می‌شود [۹]. همچنین در سال ۲۰۱۲ یک الگوریتم چند هدفه بهینه‌ساز گروه جستجو به منظور حل مسائل بهینه‌سازی چند هدفه معرفی شد [۱۰]. بعلاوه در سال ۲۰۱۲ از الگوریتم بهینه‌ساز گروه جستجو در کاربرد مخابره اقتصادی (ED)^{۱۰} در سیستم‌های قدرت استفاده شد [۱۱].

۲-۲ الگوریتم بهینه‌ساز گروه جستجو

جمعیت بهینه‌ساز گروه جستجو گروه^{۱۱} و هر فرد یک عضو^{۱۲} نامیده می‌شود. در فضای n بعدی، i امین عضو در k امین تکرار دارای موقعیت $X_i^k \in R^n$ و زاویه سر^{۱۳} $\phi_i^k = (\phi_{i1}^k, \dots, \phi_{i(n-1)}^k) \in R^{n-1}$ می‌باشد. جهت جستجوی^{۱۴} i امین عضو بردار یکه $D_i^k = (d_{i1}^k, \dots, d_{in}^k) \in R^n$ است که از طریق تبدیل قطبی به دکارتی محاسبه می‌شوند.

الگوریتم بهینه‌ساز گروه جستجو (GSO)^۱ [۴] یک الگوریتم تکاملی مبتنی بر جمعیت می‌باشد. تمرکز روش جستجوی این الگوریتم بر نحوه جستجوی غذای حیوانات می‌باشد. موفقیت حیوانات در جستجو به ۳ مورد وابسته است: (۱) رابطه میان در دسترس بودن منابع با توزیع زمانی و مکانی منبع-ها در محیط جستجو؛ (۲) توانایی حیوان در مکان‌یابی منبع‌های قرار گرفته در محیط؛ (۳) توانایی گونه حیوان در تطبیق با تغییرات کوتاه مدت یا طولانی مدت محیط و بهبود توانایی‌های کنونی‌اش. موارد ذکر شده برگرفته از انتخاب طبیعت^۲ می‌باشند.

جمعیت الگوریتم بهینه‌ساز گروه جستجو [۴]، گروه نامیده می‌شود و به هر یک از موجودات عضو گفته می‌شود. گروه از سه نوع عضو تشکیل شده است: (۱) عضو تولید کننده^۳؛ (۲) عضو برداشت کننده^۴؛ (۳) عضو تکاور^۵. هر عضو استراتژی جستجوی خاص خود را دارد: (۱) عضو تولید کننده به دنبال غذا می‌گردد؛ (۲) عضو برداشت کننده از منابع پیدا شده توسط دیگران استفاده می‌کند؛ (۳) عضو تکاور نیز به صورت تصادفی به دنبال منابع خودش می‌گردد. رویکرد جستجوی هر سه نوع عضو گروه بهینه‌ساز گروه جستجو از قدرت بینایی موجودات زنده الهام گرفته است.

الگوریتم بهینه‌ساز گروه جستجو [۴] دارای پارامترهای مختلفی می‌باشد. یکی از کلیدی‌ترین پارامترهای این الگوریتم، پارامتر حداکثر فاصله تعقیب است. این پارامتر تعیین کننده اندازه قدم برداشته شده توسط هر عضو تولید کننده گروه در هر تکرار می‌باشد. این پارامتر در طول دوره تکامل الگوریتم بهینه‌ساز گروه جستجو ثابت می‌باشد. در این مقاله به منظور بهبود این الگوریتم، مقدار این پارامتر در هر تکرار به صورت تطبیقی توسط بازخوردی از موفقیت اعضاء تولیدکننده گروه تعیین می‌شود. آزمایش‌ها نشان می‌دهند که رویکرد تطبیقی به خوبی می‌تواند مقدار پارامتر را تعیین کند

در ادامه این مقاله، در بخش ۲ الگوریتم بهینه‌ساز گروه جستجو معرفی می‌شود. در بخش ۳ رویکرد تنظیم پارامتر تطبیقی معرفی می‌شود. در بخش ۴، نتایج شبیه‌سازی رویکرد پیشنهادی تحلیل می‌شود. در انتها و در بخش ۵ به جمع‌بندی و نتیجه‌گیری مقاله می‌پردازیم.

۲- بهینه‌ساز گروه جستجو

۲-۱ معرفی بهینه‌ساز گروه جستجو

بهینه‌ساز گروه جستجو (GSO) [۴] یک مکاشفه بهینه‌سازی مبتنی بر جمعیت می‌باشد. این مکاشفه از رفتار جستجوی حیوانات نشأت گرفته است. رفتار جستجوی حیوانات می‌تواند بوسیله تلاش آنها برای پیدا کردن

^۶ Artificial Neural Network (ANN)

^۷ Function optimization

^۸ Double layer grid shells

^۹ Small World (SW)

^{۱۰} Economic Dispatch (ED)

^{۱۱} Group

^{۱۲} Member

^{۱۳} Head angle

^{۱۴} Search direction

^۱ Group Search Optimizer (GSO)

^۲ Natural selection

^۳ Producer member

^۴ Scrounger member

^۵ Ranger member

کنونی باقیمانده و سرش را به سمت یک زاویه تصادفی می‌چرخاند. در (۵)

$$\alpha_{\max} \in R^1 \text{ حداکثر زاویه چرخش}^3 \text{ می‌باشد.}$$

$$\varphi^{k+1} = \varphi^k + r_2 \alpha_{\max} \quad (5)$$

(۳) اگر تولیدکننده نتوانست موقعیت بهتری پس از a تکرار بدست آورد، سر را به زاویه صفر درجه می‌چرخاند. در (۶) $a \in R^1$ یک ثابت می‌باشد.

$$\varphi^{k+a} = \varphi^k \quad (6)$$

در هر تکرار الگوریتم بهینه‌ساز گروه جستجو، تعدادی از اعضا به عنوان برداشت‌کنندگان انتخاب می‌شوند. برداشت‌کنندگان به دنبال فرصت‌هایی جهت پیوستن به منابع پیدا شده توسط تولیدکنندگان می‌باشند. در بهینه‌ساز گروه جستجو در صورت پیدا شدن یک منبع غذایی توسط عضو تولیدکننده، اعضا برداشت‌کننده از تکنیک کپی‌برداری ناحیه^۴ استفاده می‌کنند. آنها به محض پیدا شدن منبع توسط تولیدکننده به سمت منطقه‌ای که تولیدکننده در آن قرار دارد، حرکت می‌کنند. در k امین تکرار، فرایند کپی‌برداری ناحیه i امین تولیدکننده به صورت قدم زدن تصادفی برداشت‌کننده به سمت تولیدکننده مدل می‌شود.

$$X_i^{k+1} = X_i^k + r_3 \circ (X_p^k - X_i^k) \quad (7)$$

که $r_3 \in R^n$ یک دنباله تصادفی یکنواخت در بازه $(0,1)$ می‌باشد. عملگر "o" ضرب تک تک مؤلفه‌های دو بردار می‌باشد. در حین برداشت، i امین برداشت‌کننده به جستجوی برای پیدا کردن منابع جدیدتر ادامه می‌دهد. این رفتار بوسیله چرخاندن سرش به یک زاویه جدید تصادفی با استفاده از رابطه (۵) مدل می‌شود.

در طبیعت فاز ابتدایی جستجوی تعدادی از حیوانات گشت‌زنی بدون هیچ سرنخی به دنبال منابع می‌باشد. در الگوریتم بهینه‌ساز گروه جستجو اگر i امین عضو گروه تکاور می‌باشد، در صورتیکه گشت‌زنی انجام دهد. حیوانات تکاور در طبیعت به منظور پیدا کردن کارای منابع قدم‌های تصادفی و اصولی برمی‌دارند. قدم زدن تصادفی یکی از روش‌های کارای جستجو می‌باشد که توسط اعضا تکاور انجام می‌شود. در k امین تکرار، i امین عضو تکاور یک زاویه سر تصادفی^۵ φ_i با استفاده از رابطه (۵) تولید می‌کند و سپس یک فاصله تصادفی را با استفاده از (۸) انتخاب می‌کند و با استفاده از (۹) به نقطه جدید حرکت می‌کند.

$$l_i = a \times r_1 l_{\max} \quad (8)$$

$$X_i^{k+1} = X_i^k + l_i D_i^k(\varphi^{k+1}) \quad (9)$$

۳- مدل تطبیق پارامتر پیشنهادی

^۳ Maximum turning angle

^۴ Area copying

^۵ Random head angle

$$d_{i1}^k = \prod_{q=1}^{n-1} \cos(\varphi_{iq}^k) \quad (1)$$

$$d_{ij}^k = \sin(\varphi_{i(j-1)}^k) \times \prod_{q=j}^{n-1} \cos(\varphi_{iq}^k) \quad (j = 2, \dots, n-1)$$

$$d_{in}^k = \sin(\varphi_{i(n-1)}^k)$$

یک گروه در الگوریتم بهینه‌ساز گروه جستجو از ۳ نوع عضو تشکیل شده است: (۱) عضو تولیدکننده: این اعضا با حرکت در فضای جستجو به دنبال منابع غذایی جدید می‌گردند. (۲) عضو برداشت‌کننده: با پیدا شدن منبع غذایی توسط عضو تولیدکننده این اعضا به دنبال پیوستن به منبع غذایی جدید می‌باشند. (۳) عضو تکاور: این اعضا به صورت تصادفی حرکت می‌کنند و به دنبال منابع‌های غذایی جدیدتر می‌باشند. این اعضا از گیر کردن الگوریتم در بهینه محلی جلوگیری می‌کنند.

در هر تکرار یک عضو از گروه که در بهترین منطقه واقع شده به عنوان تولیدکننده در نظر گرفته می‌شود. سپس این عضو به پویش محیط می‌پردازد تا منبع (بهینه) جدیدی پیدا کند. پویش کردن محیط یک مؤلفه کلیدی جهت‌گیری جستجو می‌باشد. در الگوریتم بهینه‌ساز گروه جستجو از بینایی که اصلی‌ترین مکانیسم پویش محیط است، استفاده می‌شود.

محدوده پویش حیوانات به صورت دنباله‌ای مخروطی می‌باشد که بوسیله حداکثر زاویه تعقیب $(\theta_{\max})^1$ و حداکثر فاصله تعقیب $(l_{\max})^2$ مشخص می‌شود. رأس این مخروط نیز نقطه‌ای است که حیوان برای شکار ثابت ایستاده است. در الگوریتم بهینه‌ساز گروه جستجو میدان پویش بینایی ساده‌سازی شده و به یک فضای n بعدی تعمیم داده شده است.

در الگوریتم بهینه‌ساز گروه جستجو در k امین تکرار تولیدکننده X_p بدین صورت رفتار می‌کند: (۱) تولیدکننده اقدام به پویش محیط می‌کند و سه نقطه را به صورت تصادفی نمونه برداری می‌کند: یک نقطه در زاویه صفر درجه (۲)، یک نقطه در ابرمکعب دست راست (۳) و یک نقطه در ابرمکعب دست چپ (۴). که در (۲) - (۴) $r_1 \in R^1$ یک عدد نرمال تصادفی با میانگین صفر و انحراف معیار ۱ و $r_2 \in R^{n-1}$ نیز یک دنباله تصادفی یکنواخت در بازه $(0,1)$ می‌باشد.

$$X_z = X_p^k + r_1 l_{\max} D_p^k(\varphi^k) \quad (2)$$

$$X_r = X_p^k + r_1 l_{\max} D_p^k\left(\varphi^k + r_2 \frac{\theta_{\max}}{2}\right) \quad (3)$$

$$X_l = X_p^k + r_1 l_{\max} D_p^k\left(\varphi^k - r_2 \frac{\theta_{\max}}{2}\right) \quad (4)$$

(۲) سپس تولیدکننده بهترین نقطه با بهترین منبع (بهترین شایستگی) را پیدا می‌کند. اگر بهترین نقطه منبع بهتری از موقعیت کنونی داشته باشد، عضو مربوطه به این نقطه حرکت می‌کند. در غیر این صورت در موقعیت

^۱ Maximum pursuit angle

^۲ Maximum pursuit distance

به منظور ارزیابی الگوریتم پیشنهادی، تعداد ۱۲ تابع محک از مجموعه ۲۱ عضوی معرفی شده در [۴]، [۱۴] برای آزمایشها در نظر گرفته شده است. این تابعها شامل دو دسته کلی می‌باشند: (۱) تابعهای تک‌قله‌ای ($f_1 - f_7$) که دارای ساختار ساده‌ای می‌باشند و به راحتی قابل بهینه‌سازی می‌باشند. (۲) تابعهای چندقله‌ای که دارای مینیممهای محلی فراوانی بود به سختی بهینه‌سازی می‌شوند. پارامترهای کلیدی در نظر گرفته شده برای اجرای الگوریتم در جدول ۱ آورده شده است.

پارامترهای در نظر گرفته برای GSO و ADGSO در ادامه ذکر شده‌اند: جمعیت اولیه با توزیع یکنواخت تولید می‌شود. گروه شامل ۴۸ عضو می‌باشد که ۱ عضو آن تولیدکننده می‌باشد. ۸۰٪ باقیمانده اعضا، برداشت‌کننده و ۲۰٪ تکاور می‌باشند. مقدار اولیه زاویه سر $\phi^0 = \left(\frac{\pi}{4}, \dots, \frac{\pi}{4}\right)$ می‌باشد. ثابت a برابر می‌باشد $a = \text{round}(\sqrt{n+1})$ که n تعداد بعدهای مسئله می‌باشد. حداکثر زاویه تعقیب $\theta_{\max} = \frac{\pi}{a^2}$ برابر می‌باشد. حداکثر زاویه چرخش برابر $a_{\max} = \frac{\theta_{\max}}{2}$ می‌باشد. حداکثر زاویه تعقیب برای الگوریتم ADGSO، توسط (۱۰) - (۱۲) محاسبه می‌شود و برای الگوریتم GSO با استفاده از (۱۳) محاسبه می‌شود.

$$l_{\max} = \|U - L\| = \sqrt{\sum_{i=1}^n (U_i - L_i)^2} \quad (13)$$

الگوریتم بهینه‌ساز گروه جستجو (GSO)

$k=0$ مقداردهی اولیه موقعیت‌ها X_i و زاویه سر ϕ_i گروه و محاسبه شایستگی $f(X_i)$.

تا زمانیکه (شرایط خاتمه ملاقات نشدند)

برای (هر عضو i در گروه)

انتخاب تولیدکنندگان: تولیدکننده X_p گروه را پیدا کنید.

انجام تولیدکنندگی:

(۱) تولیدکننده با زاویه صفر درجه پویس می‌کند و ۳ نقطه از میدان پویس را به صورت تصادفی با استفاده از رابطه‌های (۲) الی (۴) نمونه‌برداری می‌کند.

(۲) پیدا کردن بهترین نقطه با بهترین منبع. اگر بهترین نقطه منبع بهتری از موقعیت کنونی داشت، به این نقطه حرکت می‌کند. در غیر اینصورت در موقعیت کنونی‌اش سرش را با استفاده از رابطه (۵) به یک زاویه جدید می‌چرخاند.

(۳) اگر تولیدکننده پس از a تکرار نتوانست ناحیه بهتری پیدا کند، با استفاده از رابطه (۶) سرش را به سمت زاویه صفر درجه خواهد چرخاند.

انجام برداشت‌کنندگی: ۸۰٪ از باقیمانده اعضا به برداشت کردن می‌پردازند.

انجام گشت‌زنی: باقیمانده اعضا پراکنده می‌شوند تا عمل گشت‌زنی را انجام دهند:

(۱) تولید یک زاویه سر اولیه با استفاده از رابطه (۵).

(۲) انتخاب یک فاصله تصادفی l_i از توزیع نرمال.

(۳) تکاورها با استفاده از رابطه (۹) به نقطه جدید منتقل می‌شوند. سپس ناحیه جستجوی آنها چک شده و شایستگی آنها ارزیابی می‌شود؛

محاسبه شایستگی: شایستگی عضو کنونی محاسبه می‌شود: $f(X_i)$.

در الگوریتم بهینه‌ساز گروه جستجو پارامتر حداکثر فاصله تعقیب (l_{\max}) در طول دوره تکامل گروه ثابت در نظر گرفته شده است. این پارامتر در (۲) - (۴) به صورت مستقیم بر اعضای تولیدکننده و در (۹) به صورت غیر مستقیم به اعضا تکاور اعمال می‌شود. به منظور تسریع، تعدیل و کاهش حرکت اعضا به سمت بهینه تابع، در تکرارهای اولیه، میانی و پایانی الگوریتم یک رویکرد تطبیقی برای تنظیم این پارامتر پیشنهاد می‌شود.

به منظور داشتن یک استراتژی تطبیق پارامتر کارا و پویا باید بازخوردی مناسب از سمت گروه برای الگوریتم بهینه‌ساز گروه جستجو تعریف شود. در این راستا از تکنیک نرخ موفقیت استفاده شده است. این رویکرد تطبیقی در [۱۲] معرفی شده و همچنین در الگوریتم بهینه‌سازی گروه ذرات توسط نیک‌آبادی و همکاران [۱۳] استفاده شده است. نرخ بالای موفقیت معرف این است که شایستگی بهترین عضو جمعیت بهبود یافته است و عضوهای برداشت‌کننده به تدریج به سمت آن حرکت می‌کنند. همچنین نرخ پایین موفقیت معرف این است که شایستگی درصد کمی از اعضا بهبود یافته است. موفقیت عضو i در تکرار k ام یک مسئله بهینه‌سازی با استفاده از (۱۰) تعریف می‌شود که $f()$ تابع هدف می‌باشد.

$$S_i^k = \begin{cases} 1 & f(X_i^k) < f(X_i^{k-1}) \\ 0 & f(X_i^k) \geq f(X_i^{k-1}) \end{cases} \quad (10)$$

در ادامه با استفاده از (۱۱)، نرخ موفقیت گروه محاسبه می‌شود. که در (۱۱) m اندازه گروه و $P_s \in [0, 1]$ درصد موفقیت اعضا گروه می‌باشد.

$$P_s = \frac{\sum_{i=1}^n S_i^k}{m} \quad (11)$$

مقدار کم P_s معرف این است که اعضا گروه پراکنده می‌باشند و مقدار زیاد آن معرف نزدیکی اعضا می‌باشد. در صورتیکه که موقعیت اعضا بهبود پیدا نکند، $P_s = 0$ خواهد شد. بنابراین کلیه ذرات ساکن خواهند شد. در این مقاله به منظور جلوگیری از رکود اعضا این نرخ به صورت وزن دار اعمال می‌شود. بنابراین مقدار نهایی حداکثر فاصله تعقیب (l_{\max}) با استفاده از نگاشت مقدار P_s به یک بازه قابل قبول برای حداکثر فاصله تعقیب به صورت (۱۲) انجام می‌شود. که در (۱۲) $w_{\min}, w_{\max} \in [0, 1]$ (پارامترهای وزن)، $P_s \in [0, 1]$ و l_{\max} هر مقدار قابل قبولی می‌تواند باشد. فلوجارت الگوریتم تطبیقی حداکثر فاصله تعقیب بهینه‌ساز گروه جستجوی (ADGSO)^۱ در الگوریتم ۱ آورده شده است.

$$l_{\max}^k = (w_{\max} - w_{\min}) P_s^k + w_{\min} \quad (12)$$

۴- شبیه‌سازی

۴-۱ معرفی تابع های محک و تنظیمات آزمایش ها

^۱ Adaptive maximum pursuit distance (D) GSO (ADGSO)

محاسبه موفقیت: با استفاده از (۱۰) موفقیت عضو X_i محاسبه می‌شود: S_i^k .

انتهای برای

محاسبه نرخ موفقیت: با استفاده از (۱۱) نرخ موفقیت گروه محاسبه می‌شود: P_g .

محاسبه حداکثر فاصله تعقیب: با استفاده از (۱۲) بدست می‌آید: J_{max} .

$$k = k + 1$$

انتهای تازمانیکه

جدول ۱ میانگین و واریانس نتایج $f_{12}(x)^{30} - f_1(x)^{30}$.

Function Optimizer	f_1	f_2	f_3	f_4
GA	3.17E+00±1.66E+00	5.77E-01±1.31E-01	9.75E+03±2.59E+03	7.96E+00±1.51E+00
PSO	3.69E-37±2.45E-36	2.92E-24±1.14E-23	1.20E-03±2.11E-03	4.12E-01±2.50E-01
GSO	1.94E-08±1.16E-08	3.70E-05±8.62E-05	5.78E+00±3.68E+00	1.08E-01±4.00E-02
ADGSO	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00	0.00E+00±0.00E+00
Function Optimizer	f_5	f_6	f_7	f_8
GA	3.39E+02±3.61E+02	3.70E+00±1.95E+00	1.05E-01±3.62E-02	-12566.1±2.11E+00
PSO	3.74E+01±3.21E+01	1.46E-01±4.18E-01	9.90E-03±3.54E-02	-9659.7±4.64E+02
GSO	4.98E+01±3.02E+01	1.60E-02±1.33E-01	7.38E-02±9.26E-02	-12569.5±2.21E-02
ADGSO	2.89E+01±3.83E-02	2.21E+00±1.78E+00	2.24E-04±2.17E-04	-25821.8±1.23E+04
Function Optimizer	f_9	f_{10}	f_{11}	f_{12}
GA	6.51E-01±3.59E-01	8.68E-01±2.81E-01	1.00E+00±6.75E-02	4.36E-02±5.06E-02
PSO	2.08E+01±5.94E+00	1.34E-03±4.23E-02	2.32E-01±4.43E-01	3.95E-02±9.14E-02
GSO	1.02E+00±9.51E-01	2.65E-05±3.08E-05	3.08E-02±3.09E-02	2.76E-11±9.17E-11
ADGSO	0.00E+00±0.00E+00	8.88E-16±0.00E+00	0.00E+00±0.00E+00	7.20E-02±1.23E-02

تابع‌ها دارای مینیمم‌های محلی اندکی می‌باشند. جمعیت PSO در هر تکرار به سمت دو موقعیت بهترین تجربه فردی ($pbest$) و بهترین تجربه سراسری ($gbest$) حرکت می‌کنند. حرکت جذبی ذرات PSO به سمت این دو موقعیت باعث می‌شود تا در تابع‌های با ساختار ساده الگوریتم توازن میان جستجوی محلی و سراسری را حفظ کرده و پاسخ‌های خوبی بدست آورد. این در حالیست که در تابع‌های چند قله‌ای، به دلیل پیچیده‌تر شدن ساختار تابع بهینه‌سازی الگوریتم PSO توانایی تولید پاسخ مناسبی نمی‌باشد. الگوریتم GA نیز در هر یک از تابع‌های تست بدترین جواب‌ها را بدست آورده است. ساختار جهش و بازترکیبی این الگوریتم مناسب برای بهینه‌سازی عددی نمی‌باشد و این الگوریتم از همگرایی زودرس رنج می‌برد. در الگوریتم GSO، پارامتر حداکثر فاصله تعقیب (l_{max})، تعیین کننده اندازه قدم برداشته شده توسط عضو تولید کننده در هنگام حرکت به سمت بهینه می‌باشد. با توجه به نتایج جدول ۱، با مقایسه نتایج الگوریتم‌های ADGSO و GSO متوجه می‌شویم که استراتژی تعیین تطبیقی حداکثر فاصله تعقیب تأثیر بسزایی در کارایی الگوریتم ADGSO از خود نشان داده است. این استراتژی در هر تکرار بازخوردی از گروه تحت عنوان نرخ موفقیت دریافت می‌کند، که بر مبنای آن پارامتر l_{max} تعیین می‌کند. نرخ

آزمایش‌ها در دو دسته تابع‌های محک ۳۰ - بعدی ($f_1 - f_{13}$) و ۳۰۰ - بعدی ($f_8 - f_{13}$) انجام شده‌اند. در آزمایش‌ها علاوه بر مقایسه با الگوریتم GSO استاندارد، الگوریتم ADGSO با الگوریتم ژنتیک (GA) و PSO نیز مقایسه می‌شود. تنظیمات مربوط به این الگوریتم‌ها و تعداد ارزیابی کارایی مربوط به هر تابع محک مشابه [۴] در نظر گرفته شده است. الگوریتم ADGSO، $w_{min} = 0.4$ و $w_{max} = 0.9$ می‌باشند. نتایج بدست آمده از الگوریتم ADGSO حاصل ۵۰ بار اجرای این الگوریتم می‌باشد.

۴-۲ آزمایش اول: تابع های ۳۰ - بعدی

هدف از این آزمایش بررسی توانایی الگوریتم‌های ADGSO، GSO، PSO و GA در بهینه‌سازی تابع‌های محک می‌باشد. نتایج حاصل از اجرای این الگوریتم‌ها بر روی تابع‌های $f_1 - f_{12}$ در جدول ۱ ذکر شده است. با توجه به نتایج الگوریتم ADGSO در تابع‌های تک‌قله‌ای ($f_1 - f_7$) بهترین پاسخ‌ها را بدست آورده است. این الگوریتم توانسته است در ۴ مورد به جواب بهینه سراسری تابع تست برسد. همچنین در تابع‌های محک چند قله‌ای ($f_8 - f_{12}$) توانسته است در ۳ تابع از ۵ تابع تست بهترین پاسخ را بدست آورد. با توجه به نتایج جدول ۱، الگوریتم PSO توانسته است در تابع‌های تک-قله‌ای پاسخ بهتری نسبت به الگوریتم GSO استاندارد بدست آورد. این

محاسبه می‌کند. تنظیم پارمتر I_{max} موجب برقراری تعادل میان جستجوی محلی و جستجوی سراسری الگوریتم ADGSO می‌شود.

الگوریتم ADGSO تحت مجموعه‌ای از ۱۲ تابع محک مقایسه و ارزیابی شد. تابع‌های محک شامل ۳ دسته مختلف می‌باشند: تابع‌های تک قله‌ای، تابع‌ها چندقله‌ای پیچیده، تابع‌های چند قله‌ای با مقیاس بزرگ. نتایج آزمایش‌ها نشان می‌دهند که این الگوریتم در مینیمم‌سازی این تابع‌ها بسیار خوب عمل می‌کند. رویکرد تطبیقی الگوریتم ADGSO کمک می‌کند تا الگوریتم در هنگام بهینه‌سازی تابع‌های مختلف تعادل میان قدرت اکتشاف (استراتژی تعبیه شده در اعضا تولیدکننده) و استخراج (استراتژی تعبیه شده در اعضا برداشت کننده) را حفظ کند.

الگوریتم GSO استاندارد دارای پارامترهای کنترلی کلیدی از جمله حداکثر زاویه تعقیب، حداکثر زاویه چرخش، حداکثر فاصله تعقیب می‌باشد. به کار بردن رویکرد تطبیقی نرخ موفقیت به منظور تنظیم همزمان این ۳ پارامتر در حین تکامل الگوریتم GSO می‌تواند به عنوان پیشنهادی برای ادامه این مقاله در نظر گرفته شود.

مراجع

- [1] B. Y. Qu, J. J. Liang, and P. N. Suganthan, "Niching PSO with local search for multi-modal optimization," *Information Sciences*, 2012.
- [2] F. E. B. Otero, A. A. Freitas, and C. G. Johnson, "Inducing decision trees with an ant colony optimization algorithm," *Applied Soft Computing*, 2012.
- [3] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, pp. 120–142, 2012.
- [4] S. He, Q. Wu, and J. Saunders, "Group search optimizer: an optimization algorithm inspired by animal searching behavior," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 5, pp. 973–990, 2009.
- [5] S. He, Q. Wu, and J. Saunders, "A group search optimizer for neural network training," *Computational Science and Its Applications-ICCSA 2006*, pp. 934–943, 2006.
- [6] S. He, Q. Wu, and J. Saunders, "A novel group search optimizer inspired by animal behavioural ecology," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 1272–1278.
- [7] Q. Guang, L. Feng, and L. Lijuan, "A Quick Group Search Optimizer and Its Application to the Optimal Design of Double Layer Grid Shells," *AIP Conference Proceedings*, vol. 1233, no. 1, pp. 718–723, May 2010.
- [8] J. Yao, Z. Cui, Z. Wei, and Y. Tan, "Hybrid group search optimiser with quadratic interpolation method and its application," *International Journal of Wireless and Mobile Computing*, vol. 5, no. 1, pp. 98–106, 2011.
- [9] X. Yan, W. Yang, and H. Shi, "A group search optimization based on improved small world and its application on neural network training in ammonia

موفقیت رابطه مستقیم با بهبود شایستگی اعضا دارد، لذا این پارمتر بوسیله یک دید جامع از حرکت گروه به سمت بهینه محلی تعیین می‌شود.

۴-۳ آزمایش دوم: تابع ۳۰۰ - بعدی

همواره مسائل موجود در دنیای واقعی بسیار پیچیده می‌باشند و پارامترهای بسیاری در آنها درگیر می‌باشند. در صورت مدل‌سازی آنها به صورت یک مسئله بهینه‌سازی، این مسائل تبدیل به مسائل بهینه‌سازی با تعداد ابعاد بالا می‌شوند. هدف از این آزمایش بررسی کارایی الگوریتم‌ها در بهینه‌سازی تابع‌های محک چندقله‌ای (f_8-f_{12}) با تعداد بعد بالا (۳۰۰ بعد) می‌باشد.

جدول ۲ نتایج بدست آمده از اجرای الگوریتم ADGSO, GSO, PSO و GA می‌باشد. تابع‌های تست استفاده شده در این آزمایش همه از نوع چند قله‌ای بوده و دارای ۳۰۰ بعد می‌باشند. الگوریتم ADGSO توانسته است در تابع‌های f_{10} و f_{11} بهترین جواب‌ها را بدست آورد. البته الگوریتم GSO استاندارد توانسته است، در تابع‌های محک f_8, f_9, f_{12} بهترین پاسخ را بدست آورد.

جدول ۲ میانگین و واریانس نتایج $f_{12}(x)^{300} - f_8(x)^{300}$

Func. Opti.	8	9	10	11	12
GA	-117275	1.21E+02	6.24E+00	3.70E-01	5.28E+01
PSO	-87449.2	4.27E+02	3.95E-06	1.81E+00	1.46E+01
GSO	-125351.2	9.89E+01	1.35E-03	1.82E-07	8.26E-08
ADGSO	-1471242	6.54E+02	8.88E-16	0.00E+00	7.09E-02

در الگوریتم‌های بهینه‌سازی اگر با افزایش تعداد بعدها، کارایی الگوریتم کاهش یابد، اصطلاحاً الگوریتم دچار طلسم/ابعادی شده است. با توجه به جدول ۲ این پدیده در تمامی الگوریتم‌های شبیه‌سازی شده مشاهده می‌شود، بجز تابع‌های f_{10} و f_{11} بهینه شده توسط الگوریتم ADGOS. رویکرد تطبیقی الگوریتم ADGSO باعث می‌شود تا الگوریتم در هنگام برخورد با تابع‌های به شدت پیچیده با گرفتن به موقع بازخورد از گروه، کارایی خود را حفظ کند.

۵- نتیجه گیری

در این مقاله الگوریتم ADGSO معرفی شد. الگوریتم ADGSO دارای ۳ عضو است، که هر یک معرف یک تکنیک جستجوی منحصر به فرد می‌باشد: (۱) عضو تولیدکننده که به دنبال مینیمم‌های محلی می‌گردد؛ (۲) عضو برداشت‌کننده که با پیدا شدن مینیمم محلی توسط عضو تولیدکننده به سمت آن مینیمم محلی حرکت می‌کند؛ (۳) عضو تکاور که با قدم زدن تصادفی به دنبال فضاها امیدبخش جدید می‌گردد. پایه روش جستجوی اعضا الگوریتم ADGSO، توانایی بینایی اعضا می‌باشد. در هر نسل از این الگوریتم با استفاده از یک تبدیل قطبی به کارترین زاویه دید عضو تولیدکننده به جهت حرکت آنها در فضای جستجو تبدیل می‌شود و سپس عضو مربوطه با یک اندازه قدم تطبیقی (I_{max}) به سمت موقعیت بعدی حرکت می‌کند. رویکرد تطبیقی پیشنهادی با استفاده از نرخ موفقیت که معرف درصد بهبود شایستگی گروه می‌باشد، بازخوردی را برای تنظیم I_{max}

- synthesis,” *Neurocomputing*, vol. 97, pp. 94–107, Nov. 2012.
- [10] L. Wang, X. Zhong, and M. Liu, “A novel group search optimizer for multi-objective optimization,” *Expert Systems with Applications*, vol. 39, no. 3, pp. 2939–2946, Feb. 2012.
- [11] K. Zare, M. T. Haque, and E. Davoodi, “Solving non-convex economic dispatch problem with valve point effects using modified group search optimizer method,” *Electric Power Systems Research*, vol. 84, no. 1, pp. 83–89, Mar. 2012.
- [12] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies – A comprehensive introduction,” *Natural Computing*, vol. 1, no. 1, pp. 3–52, Mar. 2002.
- [13] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, “A novel particle swarm optimization algorithm with adaptive inertia weight,” *Applied Soft Computing*, vol. 11, no. 4, pp. 3658–3670, Jun. 2011.
- [14] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 2, pp. 82–102, 1999.