

الگوریتم بهینه سازی گروه ذرات دو مرحله‌ای

حسام عمرانپور

دانشگاه صنعتی امیرکبیر، دانشکده کامپیوتر

h_omranpour@aut.ac.ir

حمید طاهرپور

دانشگاه صنعتی امیرکبیر، دانشکده کامپیوتر

h_taherpour@aut.ac.ir

محمد مهدی عبادزاده

دانشگاه صنعتی امیرکبیر، دانشکده کامپیوتر

ebadzadeh@aut.ac.ir

چکیده در این مقاله روشی جدید مبتنی بر هوش جمعی برای حل مسائل بهینه سازی ارائه می‌شود. روش پیشنهادی، با استفاده از دو مرحله تحرک و همگرایی جمعیت، به نتایج جالبی در انواع توابع می‌رسد. در این روش جمعیت اولیه ذرات مقداردهی شده و سپس این ذرات در هر مرحله ابتدا خود را از نواحی نامناسب دور کرده و پس از آن به نواحی مناسب مهاجرت می‌کنند و در نهایت در این نواحی سعی در نزدیک شدن به نقاط بهینه را دارند. ویژگی الگوریتم، نتیجه گرفتن در توابع با ابعاد بالا و همچنین توابع دارای اکسترمم‌های محلی زیاد است. حرکت در جهت دور شدن از نواحی نامناسب، باعث می‌شود تا الگوریتم در مواجهه با مسائل با ابعاد بسیار بزرگ و نیز مسائلی که در آنها جمعیت دارای توزیع اولیه نامناسبی است نیز به خوبی عمل کرده و نتایج مناسبی از خود نشان دهد. پراکندگی نامناسب جمعیت اولیه، در الگوریتم بهینه سازی گروه ذرات تاثیر منفی دارد. این الگوریتم با مهاجرت کلی ذرات به سمت فضای مناسب، به نقاط بهینه همگرا می‌شود.

در انتها ضمن آزمودن روش پیشنهادی بر روی چند تابع محک شناخته‌شده و مقایسه با الگوریتم بهینه سازی گروه ذرات مشاهده می‌شود که روش پیشنهادی به نتایج بهتری می‌رسد.

واژه های کلیدی: بهینه سازی، هوش جمعی، بهینه سازی گروه ذرات، توابع با ابعاد بالا، اکسترمم محلی، فضای جستجو، پراکندگی اولیه.

۱ - مقدمه

بهینه سازی گروه ذرات (Particle Swarm Optimization) یک تکنیک بهینه سازی است که بر پایه جمعیتی از پاسخ‌های اولیه عمل می‌کند. این تکنیک اولین بار توسط ابرهارت و کندی در سال ۱۹۹۵ بر اساس رفتار اجتماعی دسته های پرندگان و ماهی ها طراحی شد [1]. [2]

این تکنیک در بسیاری موارد شبیه به تکنیک های محاسباتی تکاملی، مانند الگوریتم های ژنتیکی عمل می‌کند. در این روش هم سیستم با جمعیتی از تعدادی پاسخ اولیه شروع به کار می‌کند و با حرکت دادن این پاسخ ها در طی تکرارهای متوالی، سعی در یافتن پاسخ بهینه دارد.

بر خلاف الگوریتم های ژنتیک ، PSO عملگر های تکاملی مانند جهش و بازترکیبی را ندارد. پاسخ های مسئله (ذرات) در PSO، در فضای جستجو به دنبال یافتن نتایج بهینه سیستم هستند.

PSO در بسیاری زمینه ها مانند یافتن جواب بهینه برای توابع، آموزش شبکه های عصبی، کنترل سیستم های فازی و دیگر مسائلی که الگوریتم های ژنتیکی برای آنها قابل اجراست، از خود عملکرد مناسب و قابل قبولی نشان می‌دهد.

از زمان معرفی الگوریتم بهینه سازی گروه ذرات توسط ابرهارت و کندی در سال ۱۹۹۵، این روش به صورت گسترده ای مورد توجه محققان قرار گرفته و تلاش های بسیاری برای بهبود عملکرد آن از جهات مختلف انجام شده است. اضافه کردن عامل ایستایی (inertia) در فرمول محاسبه سرعت از این تغییرات است [3]. این پارامتر بین جستجوی محلی و جستجوی سراسری فضای مسئله تعادل ایجاد می‌کند. به این صورت که مقادیر بالای این پارامتر برای جستجوی سراسری مناسب بوده و مقادیر پایین تر آن برای جستجوهای محلی. در [3] همچنین کاهش تدریجی این پارامتر نیز بیان می‌شود؛ در [4] بطور کلی بر روی پارامتر های PSO و تاثیر آنها بر روال کار بحث شده است. در [5] کاهش غیر خطی از طریق فازی سازی معرفی شده و در [6] این مقدار، به غیر از زمان های مقدار دهی مجدد (reinitialization) صفر در نظر گرفته می‌شود. در [7] کاهش تدریجی سرعت ماکسیمم نیز معرفی شده است. ایجاد بهبود در PSO از طریق طراحی انواع مختلف مدل های همسایگی از دیگر زمینه های تحقیقاتی فعال است. کندی در [8] و [9] بیان می‌کند که PSO با همسایگی های کوچک می‌تواند در مسائل با پیچیدگی بالاتر بهتر عمل کند و بالعکس [10].

در [11] ابرهارت و هو از همسایگی های پویا استفاده می‌کنند. به این معنی که در این روش k نزدیکترین ذره از نظر مقدار تابع شایستگی انتخاب شده و به عنوان همسایه های یک ذره، در محاسبه سرعت آن ذره استفاده می‌شوند. در [12] دو نوع از PSO، یعنی LBest PSO و GBest PSO را با هم ادغام کرده و یک الگوریتم یکنواخت UPSO را معرفی می‌کند. در [13] کندی و مندز الگوریتمی معرفی می‌کنند که در آن ذرات برای محاسبه سرعتشان، بجای استفاده از

بهترین مقدار قبلی خود و بهترین مقداری که در جمعیت دیده شده است، از مقادیر تمام همسایه های خود استفاده می کند.

از دیگر تحقیقات انجام گرفته بروی PSO می توان به معرفی عملگرهای تکاملی مانند انتخاب، بازترکیبی و جهش برای ارتقاء کارایی PSO اشاره کرد. از جدیدترین کارهای انجام شده در این زمینه، می توان الگوریتم CPSO-H [14] را نام برد که در آن ابتدا از روی جمعیت، یکسری جمعیت های تک بعدی ساخته می شود و سپس بر روی هر بعد سعی در رسیدن به هدف را دارد. این الگوریتم در توابع چند هدفه عملکرد خوبی از خود نشان می دهد.

در ادامه این مقاله، در بخش دوم مرور کوتاهی بر روی الگوریتم های مبتنی بر جمعیت و الگوریتم PSO خواهیم داشت. در بخش سوم، الگوریتم پیشنهادی ارائه می شود و مراحل آن با ذکر جزئیات بیان می گردد. در بخش چهارم، نتایج آزمایشی بر روی چند نمونه تابع محک (Benchmark)، ارائه شده است و در انتها، به نتیجه گیری می پردازیم.

۲ - الگوریتم های مبتنی بر جمعیت

هوش جمعی، یک سیستم هوش مصنوعی است که بر اساس رفتار جمعی سیستم های خود سازمانده غیر متمرکز عمل می کند. این بحث اولین بار توسط بانی و وانگ در سال ۱۹۹۸ در زمینه سیستم های روباتیک سلولی معرفی شد [15]. یک سیستم مبتنی بر هوش جمعی شامل جمعیتی از عامل های ساده است که در یک محیط قرار گرفته و با هم در تعامل هستند. با اینکه این عامل ها عملکرد ساده ای داشته و از یک سیستم کنترلی مرکزی پیروی نمی کنند ولی همین تعامل ساده باعث بروز رفتارهای پیچیده از کل سیستم می شود. از نمونه های طبیعی چنین سیستم هایی می توان کلونی مورچه ها، پرواز گروه پرندگان، حرکت گروه ماهی ها را نام برد. بر پایه این سیستم ها، الگوریتم هایی با هدف بهینه سازی معرفی شده اند که از مهمترین آنها می توان به الگوریتم "بهینه سازی کلونی مورچه ها" [16]، الگوریتم "بهینه سازی گروه ذرات" و الگوریتم "بهینه سازی بر اساس حرکت زنبور های عسل" [17] اشاره کرد.

الگوریتم ارائه شده در این مقاله، مبتنی بر هوش جمعی بوده و در مقایسه با دیگر الگوریتم های موجود در این زمینه، شباهت هایی به الگوریتم "بهینه سازی گروه ذرات" (PSO) دارد.

۲ - ۱ بهینه سازی گروه ذرات

PSO بر اساس رفتار اجتماعی دسته های پرندگان عمل می کند. برای درک بهتر این تکنیک، سناریوی زیر را در نظر بگیرید:

"دسته ای پرندۀ بطور تصادفی در محدوده خاصی به دنبال غذا می گردند. در این محدوده فقط یک تکه غذا وجود دارد و پرندگان هم از

محل این غذا خبر ندارند، ولی در هر لحظه فاصله خود با محل غذا را می دانند."

در این حالت یک استراتژی مناسب برای یافتن محل دقیق غذا، دنبال کردن آن پرندۀ ای است که از بقیه پرندگان به غذا نزدیک تر است.

در واقع PSO هم از چنین سناریویی الهام گرفته و راه حلی برای مسائل بهینه سازی ارائه می دهد. در PSO هر پرندۀ یک راه حل برای مسئله می باشد. تمام پاسخ های موجود دارای یک مقدار شایستگی (Fitness Value) هستند که آنرا از روی تابع شایستگی تعریف شده برای مسئله بدست می آورند. هدف این تکنیک یافتن محلی است که دارای بهترین مقدار شایستگی در فضای مسئله می باشد. این مقدار شایستگی در جهت و سرعت حرکت این پرندگان (پاسخ های مسئله) به سمت محل غذا (پاسخ بهینه) تاثیر مستقیم دارد.

PSO با تعدادی پاسخ اولیه (ذرات) شروع به کار می کند و با حرکت دادن این پاسخ ها در طی تکرار های متوالی، به دنبال یافتن جواب بهینه برای مسئله است. در هر تکرار دو مقدار GBest و PBest مشخص می شوند.

- PBest : مکان بهترین مقدار شایستگی که هر ذره در طول حرکت خود، به آن رسیده است.
- GBest : مکان بهترین ذره در جمعیت فعلی.

پس از یافتن مقادیر فوق، سرعت حرکت ذرات از رابطه (۱) و نیز مکان بعدی هر ذره از رابطه (۲) محاسبه می شود.

$$v_{i,t+1} = w.v_{i,t} + c_1.r_1.(Pbest_i - P_{i,t}) + c_2.r_2.(Gbest_i - P_{i,t}) \quad (1)$$

$$P_{t+1} = P_t + v_t \quad (2)$$

در این روابط مقادیر r_1 و r_2 عددی تصادفی بین صفر و یک هستند و ضرایب C_1 و C_2 هم که ضرایب یادگیری نامیده می شوند را معمولاً برابر ۲ مقدار دهی می کنیم. گام های الگوریتم PSO را بطور خلاصه در شکل (۱) می بینیم.

در هر تکرار از الگوریتم، سرعت حرکت ذرات (میزان تغییرات برای هر ذره) در هر بعد را می توان با یک مقدار از پیش تعیین شده V_{max} محدود نمود. در این حالت اگر سرعت هر ذره در هر بعد، از این حد فراتر رفت آنرا با V_{max} جایگزین می کنیم.

در این الگوریتم ذراتی را به عنوان جمعیت اولیه تولید می‌کنیم. این ذرات حرکت کرده و به سمت نقاطی که در آنها تابع هدف مقدار بهینه را دارد، همگرا می‌شوند. در مقایسه با PSO، دو مفهوم جدید در الگوریتم خود معرفی کرده‌ایم. "بدترین نقطه در جمعیت فعلی" و "بدترین نقطه در حافظه افراد" که اولی مکان قرارگیری ذره‌ای از جمعیت است که مقدار تابع هدف در آن از تمامی ذرات جمعیت بدتر است. مفهوم دوم نیز بدترین محلی است که هر ذره از جمعیت در طول حرکت خود مشاهده کرده است.

در هر مرحله پس از شناسایی این نقاط خاص، ذرات را در طی ۵ گام مجزا حرکت می‌دهیم. منطق حرکت‌های در نظر گرفته شده نیز، ابتدا "فرار از نقاط و نواحی نامناسب" و سپس همگرایی به نقاط و نواحی مناسب است.

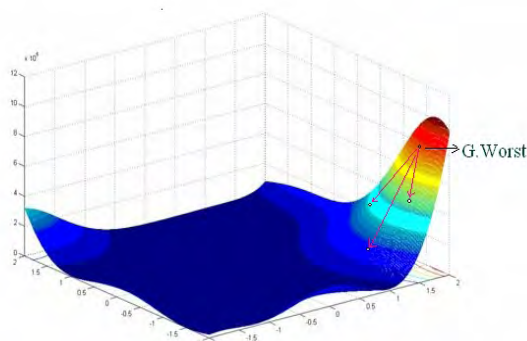
۱-۳ مرحله تحرک جمعیت

در این مرحله با سه نوع حرکت هدفمند، متفاوت و هر یک به منظوری خاص، برای ایجاد تحرک در جمعیت، جمعیت را به حرکت وامی‌داریم. در واقع با دور کردن جمعیت از نقاط بد و مهاجرت جمعیت به سمت نقاط بهینه، فضای مناسب جستجو را می‌یابیم.

۱-۱-۳ گام اول

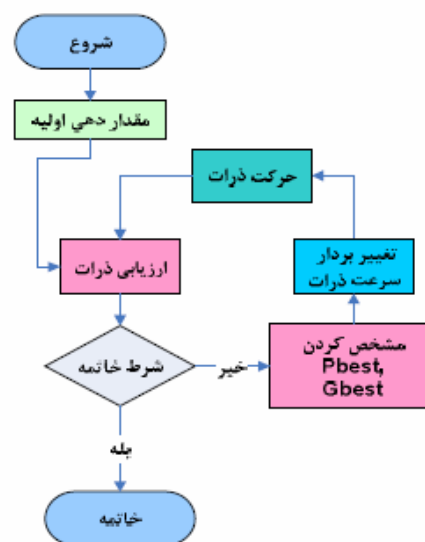
در این گام، جمعیت بصورت هدایت شده، گسترده می‌شود. در واقع هر شخص از بدترین ذره‌ی موجود دور می‌شود. این دور شدن در تمامی ابعاد و بر طبق رابطه ۳ انجام خواهد شد تا جمعیت، در فضای جستجوی بهتری، جستجو را انجام دهد (شکل ۳).

$$v_{i,t+1} = w.v_i + c_1.r.(P_{i,t} - G_{worst,i,t}) \quad (3)$$



شکل (۳): حرکت در جهت دور شدن از بدترین ذره

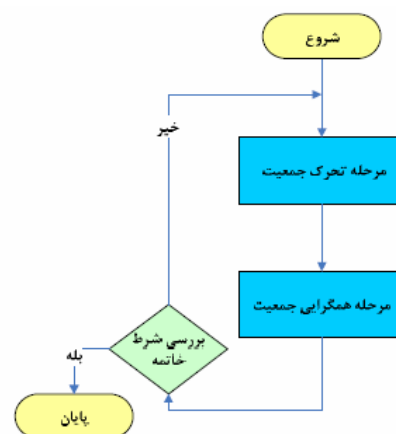
ذرات با استفاده از این حرکت، به فضای گسترده‌تری برای جستجو هدایت می‌شوند، به این امید که در آن فضا، از مکان‌های بهتری عبور کنند.



شکل (۱): الگوریتم PSO

۳- الگوریتم پیشنهادی

الگوریتم پیشنهادی بر اساس دو مرحله، بنا نهاده شده است. در هر مرحله یک نوع حرکت انجام می‌شود. مرحله اول مرحله فرار از اکستریم‌های محلی و مهاجرت جمعیت برای دور شدن از مکانهای نامناسب است. مرحله دوم، مرحله همگرایی جمعیت به بهینه سراسری است. این دو مرحله به تناوب تکرار می‌شوند تا شرط خاتمه برقرار شود (شکل ۲).



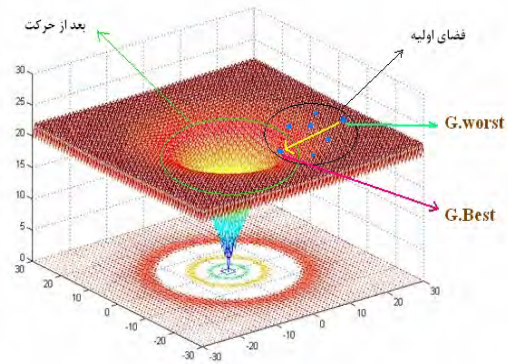
شکل (۲): فلوچارت روش پیشنهادی

هر مرحله از چند گام حرکتی تشکیل شده که هر یک به منظور خاصی انجام می‌شوند. مرحله اول شامل سه حرکت و مرحله دوم شامل دو حرکت می‌شود.

۳-۱-۲ گام دوم

در این قسمت کل ذرات با هم در جهت بدترین ذره به بهترین ذره موجود، حرکت می کنند (رابطه ۴) تا در نتیجه مانند شکل (۴) به فضای جستجوی مناسب رسیده و در آن فضا حرکات خود را ادامه دهند.

$$v_{i,t+1} = w.v_{i,t} + c_1.r.(Gbest_{i,t} - Gworst_{i,t}) \quad (4)$$



شکل (۴): مهاجرت جمعیت به سمت فضای جستجوی بهتر

در واقع با مهاجرت جمعیت از فضای جستجوی نامناسب به فضای جستجوی بهتر، ذرات در نواحی بهتری به دنبال نقاط بهینه می گردند.

۳-۱-۳ گام سوم

در این گام ذرات در محیط جدیدی که به آن مهاجرت کرده اند، مجدداً با استفاده از حافظه شخصی خود و در جهت دور شدن از بدترین نقطه مشاهده شده در طول مسیرشان، حرکت می کنند (رابطه ۵).

$$v_{i,t+1} = w.v_{i,t} + c_1.r.(P_{i,t} - Pworst_{i,t}) \quad (5)$$

در اینصورت اگر مهاجرت گام پیشین هم باعث نزدیکی ناخواسته برخی ذرات به نواحی نامناسب شده باشد و یا در اثر آن مهاجرت به اکستریم محلی رسیده باشیم، با این حرکت ذرات به امید عبور از اکستریم های محلی از اینچنین نواحی فاصله می گیرند.

۳-۲ مرحله همگرایی جمعیت

پس از دور شدن ذرات از نواحی و نقاط نامناسب که در اثر تحرک جمعیت طی سه گام پیشین انجام شد، اکنون نوبت به حرکت در جهت نزدیکی به نواحی مناسب و همگرایی به نقاط هدف می رسد.

این مرحله خود شامل دو گام است و ذرات در این مرحله با دو نوع حرکت مختلف و متفاوت، به نقاط بهینه هدایت می شوند. در واقع در انتهای این مرحله امید آن می رود که جمعیت به یک نقطه اکستریم سراسری همگرا شود.

۳-۲-۱ گام اول

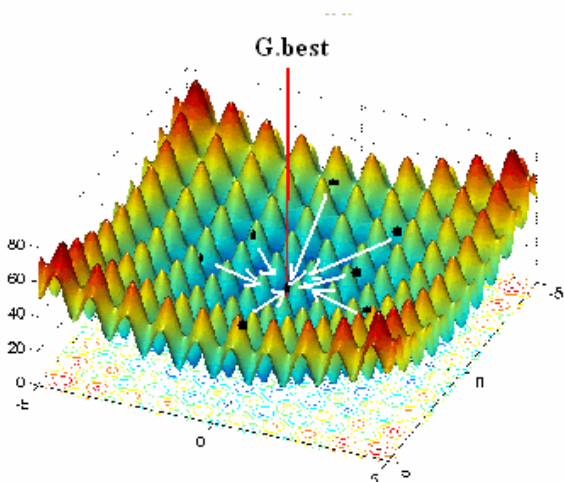
در این حرکت، هر ذره با هدف ایجاد بهبود در وضعیت کنونی و بر اساس حافظه شخصی خود، به سمت بهترین نقطه مشاهده شده در طول مسیرش هدایت می شود (رابطه ۶)، تا در نهایت با چند تکرار این گام، همه ذرات در نقاط اکستریم محلی یا سراسری که در طول مسیر حرکتی مشاهده کرده اند، نزدیک شوند.

$$v_{i,t+1} = w.v_{i,t} + c_2.r.(Pbest_{i,t} - P_{i,t}) \quad (6)$$

۳-۲-۲ گام دوم

با اطمینان از فاصله گرفتن از نواحی نامناسب طی سه گام نخست و سپس حرکت به سمت نقاط مناسب مشاهده شده توسط هر ذره (گام قبل) اکنون نوبت به حرکت تمامی افراد حاضر در جمعیت به سمت بهترین ذره موجود در جمعیت و در نهایت همگرایی به نقطه بهینه سراسری است. در این گام فرض بر این است که ذرات جمعیت در حرکات قبل، به اکستریم محلی یا سراسری نزدیک شده اند. این حرکت برای هر ذره در راستای محل قرارگیری بهترین ذره جمعیت (رابطه ۷) انجام می شود تا همگرایی به نقطه بهینه (با امید بهینه سراسری) صورت گیرد (شکل ۵).

$$v_{i,t+1} = w.v_{i,t} + c_2.r.(Gbest_{i,t} - P_{i,t}) \quad (7)$$



شکل (۵): همگرایی نهایی جمعیت به سمت نقطه بهینه موجود

۳-۳ - تحلیل

در الگوریتم pso مشاهده شد که این الگوریتم برای جابجایی و مهاجرت جمعیت قدرت چندانی نداشته و همچنین در مواجهه با مسائل بهینه سازی در ابعاد بالا امکان نرسیدن به پاسخ بهینه زیاد است. از این رو در پی آن بر آمدیم تا با دو مرحله ای نمودن pso و ارائه حرکاتی که از رسیدن به حالت سکون جلوگیری می کند، روش جدیدی ارائه کنیم. این روش با افزودن حرکاتی که در حالت کلی نامناسب نیستند، قدرت تحرک و تکاپوی بیشتری در تمام فضای جستجو را به الگوریتم می دهد. در واقع تکرارهای پیاپی و متناوب مرحله تحرک پس از مرحله همگرایی، علاوه بر حرکت دادن جمعیت و افزایش قدرت جستجو در فضای مسئله، باعث افزایش توانایی الگوریتم در فرار از اکسترمم های محلی می شود. این مسئله را با بررسی عملکرد روش پیشنهادی در مواجهه با مسائل در ابعاد بالا و پراکندگی نامناسب جمعیت اولیه بخوبی مشاهده می کنیم. لازم به ذکر است مرحله اول ۱۸٪ مرحله دوم ۸۲٪ تکرارها را شامل می شود تا علاوه بر اینکه تکاپو و تحرک را به جمعیت داده باشیم از فضاهای مناسبی که یافته ایم، دور نشویم.

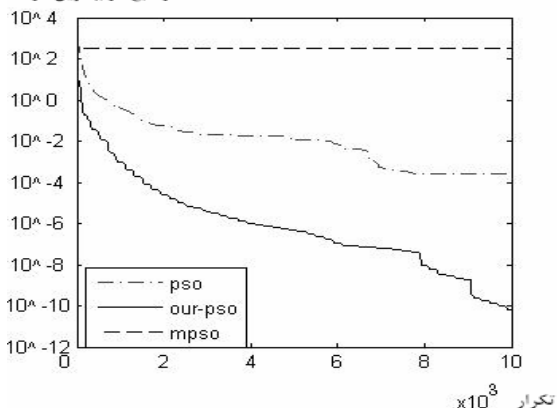
۴ - نتایج آزمایشی

در این بخش مقایسه الگوریتم پیشنهادی با روش بهینه سازی گروه ذرات و همچنین روش $mpso$ [3] در انواع توابع پیچیده ارائه شده است. آزمایشات با استفاده از نرم افزار $Matlab$ انجام شده است. این آزمایشات بر روی چهار تابع محک شناخته شده، انجام شده اند. نتایج به تفکیک در دو قسمت، ابعاد بالا و پراکندگی نامناسب جمعیت اولیه، ارائه شده است. نتایج بر اساس میانگین ده مرتبه اجرای الگوریتم ها در نمودارها آمده است. نمودار بصورت لگاریتم در مبنای ۱۰ تابع هدف بر حسب تعداد تکرار نمایش داده شده است. با توجه به شکل (۲) شرط خاتمه الگوریتم ۱۰۰۰۰ تکرار است، که گام های مرحله اول هر یک به میزان ۶٪ و گام های مرحله دوم هر کدام ۴۱٪ تکرارها را شامل می شوند. همچنین، مقادیر C_1 و C_2 به ترتیب برابر ۱ و ۲ در نظر گرفته شده است. مقدار پارامتر ایستایی (w) در تمامی مراحل الگوریتم، برابر با ۰/۷ است.

۴-۱ بررسی ابعاد بالا

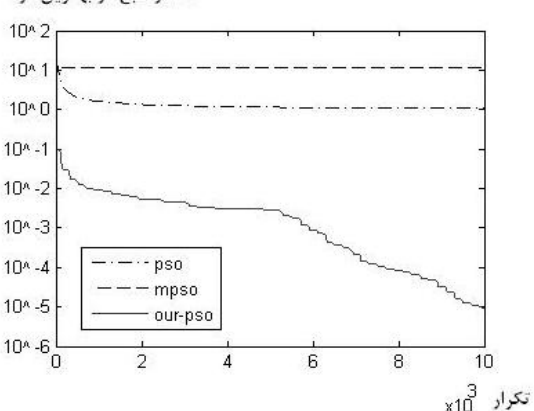
در این قسمت از توابع با ابعاد بالا برای مقایسه الگوریتم پیشنهادی با الگوریتم pso استفاده شده است؛ تا قدرت یافتن نقاط بهینه در فضای جستجوی بزرگ و با ابعاد بالا نمایش داده شود. نتایج در نمودارهای (۴-۱) بر روی چهار نوع تابع مختلف و متنوع، بررسی شده است. ابعاد مسئله در این آزمایش ها برابر ۱۰۰ در نظر گرفته شده است. همانگونه که مشاهده می شود، الگوریتم پیشنهادی با ویژگی های خاص خود به نتایج بهتری دست یافته است.

مقدار تابع در بهترین ذره



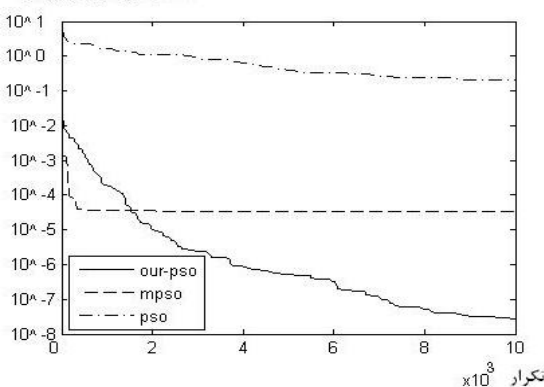
نمودار (۱): مقایسه نتایج الگوریتم ها در تابع $sphere$ با ابعاد بالا

مقدار تابع در بهترین ذره



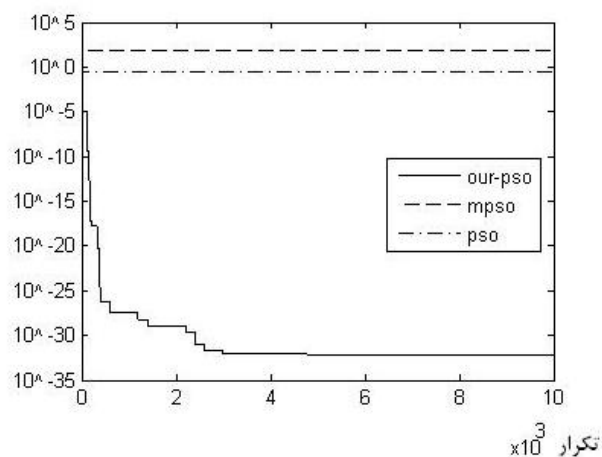
نمودار (۲): مقایسه نتایج الگوریتم ها در تابع $Ackley$ با ابعاد بالا

مقدار تابع در بهترین ذره



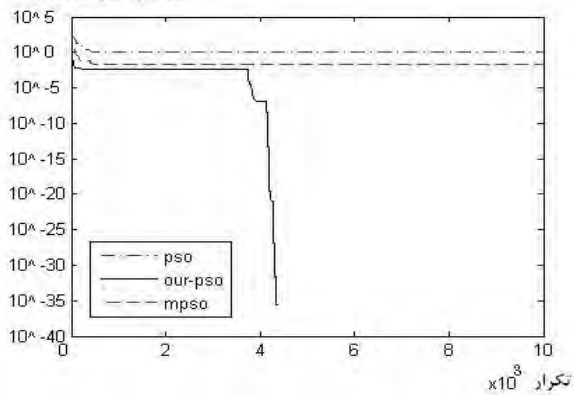
نمودار (۳): مقایسه نتایج الگوریتم ها در تابع $Rastrigin$ با ابعاد بالا

مقدار تابع در بهترین ذره



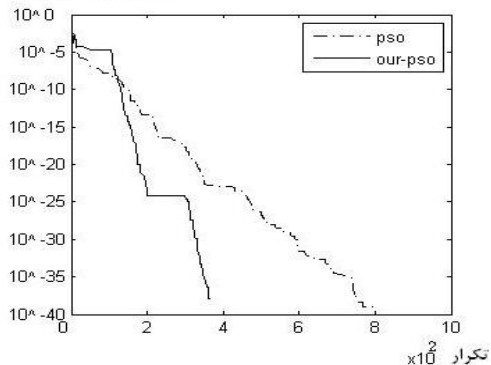
نمودار (۶): مقایسه نتایج الگوریتم‌ها در تابع Ackley با پراکندگی نامناسب جمعیت اولیه

مقدار تابع در بهترین ذره



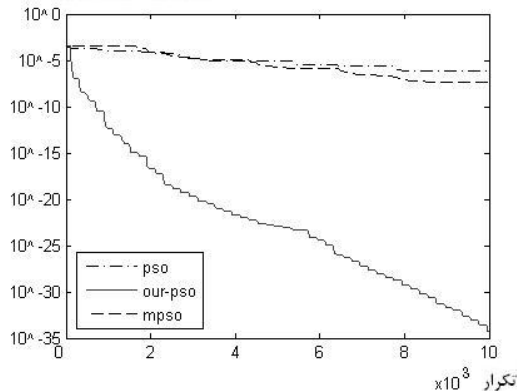
نمودار (۷): مقایسه نتایج الگوریتم‌ها در تابع Rastrigin با پراکندگی نامناسب جمعیت اولیه

مقدار تابع در بهترین ذره



نمودار (۸): مقایسه نتایج الگوریتم‌ها در تابع Griewank با پراکندگی نامناسب جمعیت اولیه با ۱۰۰۰ تکرار

مقدار تابع در بهترین ذره

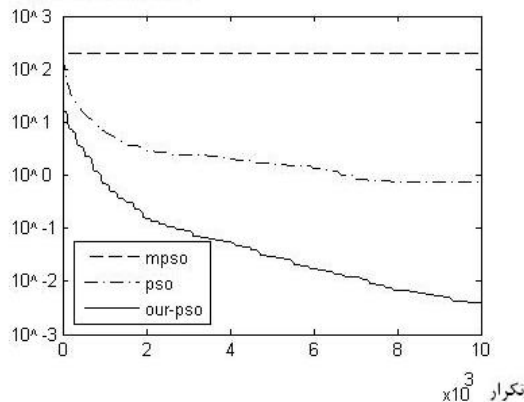


نمودار (۴): مقایسه نتایج الگوریتم‌ها در تابع Griewank با ابعاد بالا

۴-۲ بررسی پراکندگی نامناسب جمعیت اولیه

اگر جمعیت در ابتدا به خوبی پراکنده نشود و یا در حول نقطه دیگری به جز بهینه سراسری پراکنده شود نیز الگوریتم جستجو باید به نتایج خوبی برسد. همانطور که در نمودارهای (۵-۸) مشاهده می‌شود، الگوریتم پیشنهادی با استفاده از رابطه (۴) با مهاجرت جمعیت به فضای مناسب، به نتایج بهتری دست یافته است. در این آزمایش پراکندگی اولیه جمعیت حول نقطه ای به فاصله ۵ واحد از بهینه سراسری توابع (برای این توابع مرکز مختصات) است. در تابع Sphere اندازه جمعیت ۱۰ و ابعاد مسئله برابر ۱۰۰، در تابع Griewank اندازه جمعیت ۱۰ و ابعاد مسئله برابر ۵ و در تابع Ackley و Rastrigin اندازه جمعیت ۵ و ابعاد مسئله برابر ۲ در نظر گرفته شده است.

مقدار تابع در بهترین ذره



نمودار (۵): مقایسه نتایج الگوریتم‌ها در تابع sphere با پراکندگی نامناسب جمعیت اولیه

۵ - نتیجه گیری

بهینه سازی گروه ذرات از روشهای هوشمند حل مسائل بهینه سازی است. در این مقاله روشی بر اساس بهینه سازی ذرات ارائه شد که دارای قدرت زیادی در گذر از اکستریم های محلی، به خصوص در توابع پیچیده و دارای اکستریم محلی زیاد و همچنین توابع با ابعاد بالا است. استفاده از گام های حرکتی در خلاف جهت بدترین نقاط در این الگوریتم، باعث می شود جمعیت در حالت سکون گیر نکند. همانگونه که در روی انواع توابع مشاهده شد، الگوریتم پیشنهادی بخوبی از اکستریم محلی گذشته و جواب بهینه سراسری را می یابد. کار آینده ما، پویا نمودن ضرایب ثابت در معادلات مربوط به محاسبه سرعت ذرات بر حسب شرایط کنونی جمعیت و مقدار دهی مناسب آنها در هر گام است.

مراجع

- [6] Ratnaweera Asanga, Halgamuge Saman K., Watson Harry C., "Self-organizing hierarchical particle swarm optimizer with time varying accelerating coefficients", *IEEE Trans. Evol. Comput.*, vol. 8, pp. 240-255, Jun. 2004.
- [7] Fan Howard, Shi Yuhui., "Study on Vmax of particle swarm optimization", in *Proc. Workshop Particle Swarm Optimization*, Indianapolis, IN, 2001.
- [8] Kennedy James, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance", in *Proc. Congr. Evol. Comput.*, 1999, pp. 1931-1938.
- [9] Kennedy James, Mendes Rodriguez, "Population structure and particle swarm performance", in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, 2002, pp. 1671-1676.
- [10] Suganthan Ponnuthurai N., "Particle swarm optimizer with neighborhood operator", in *Proc. Congr. Evol. Comput.*, Washington, DC, 1999, pp. 1958-1962.
- [11] Hu Xiaohui, Eberhart Russel C., "Multiobjective optimization using dynamic neighborhood particle swarm optimization", in *Proc. Congr. Evol. Comput.*, Honolulu, HI, 2002, pp. 1677-1681.
- [12] Parsopoulos Konstantinos E., Vrahatis Michael N., "UPSO—A unified particle swarm optimization scheme", in *Lecture Series on Computational Sciences*, 2004, pp. 868-873.
- [13] Mendes Rodriguez, Kennedy James, Neves J., "The fully informed particle swarm: Simpler, maybe better", *IEEE Trans. Evol. Comput.*, vol. 8, pp. 204-210, Jun. 2004.
- [14] V.d.Bergh Frans, Engelbrecht Andries P., "A cooperative approach to particle swarm optimization", *IEEE Trans. Evol. Comput.*, vol. 8, pp. 225-239, Jun. 2004.
- [15] Beni Gerardo, Wang Jing, "Swarm Intelligence in Cellular Robotic Systems", *Proceed. NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, Italy, June 26-30 (1989)
- [16] Dorigo M., *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italy, 1992.
- [17] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S., Zaidi M. "The Bees Algorithm". *Technical Note, Manufacturing Engineering Centre*, Cardiff University, UK, 2005.
- [1] Eberhart Russel C., Kennedy James, "A new optimizer using particle swarm theory", in *Proc. 6th Int. Symp. Micromachine Human Sci.*, Nagoya, Japan, 1995, pp. 39-43.
- [2] Kennedy James, Eberhart Russel C., "Particle swarm optimization", in *Proc. IEEE Int. Conf. Neural Networks*, 1995, pp. 1942-1948.
- [3] Shi Yuhui, Eberhart Russel C., "A modified particle swarm optimizer", in *Proc. IEEE Congr. Evol. Comput.*, 1998, pp. 69-73.
- [4] Shi Yuhui, Eberhart Russel C., "Parameter selection in particle swarm optimization", in *Proc. 7th Conf. Evol. Programming*, New York, 1998, pp. 591-600.
- [5] Shi Yuhui, Eberhart Russel C., "Particle swarm optimization with fuzzy adaptive inertia weight", in *Proc. Workshop Particle Swarm Optimization*, Indianapolis, IN, 2001, pp. 101-106.