

الگوریتم رقابت استعماری ممتیکی

مژده اختری ، محمد رضا میبدی

چکیده

در این مقاله، یک الگوریتم ممتیکی ترکیبی با الگوریتم رقابت استعماری ارائه شده است. در مسائل با پیچیدگی بالا روش رقابت استعماری دچار مشکل می شود و در بینهای های محلی متوقف می شود لذا این مقاله با ارائه روشی ممتیکی در صدد کاهش این مشکل برآمده است. در الگوریتم رقابت استعماری به نوعی از اطلاعات جهت و فاصله در حرکت مستعمرات به سمت استعمارگر استفاده می شود، لذا از مفهوم تکامل تفاضلی برای تعریف مم درجهت بهبود و افزایش کارآیی این الگوریتم استفاده کرده ایم. آزمایشات برای بررسی کارآیی الگوریتم ترکیبی ارائه شده، بر روی مسائل محک استاندارد انجام شد. نتایج آزمایشات، کارآیی بالای الگوریتم ممتیکی ارائه شده را نشان می دهد.

كلمات کلیدی

الگوریتم ممتیک، الگوریتم رقابت استعماری، تکامل تفاضلی، مم، امپراطوری، جهت، فاصله

۱ مدرس دانشگاه فخر رازی ساوه، mojdeh.akhtari@gmail.com

عضو هیئت علمی دانشکده مهندسی کامپیوتر و فناوری اطلاعات دانشگاه صنعتی امیرکبیر، تهران، mmeybodi@aut.ac.ir

C (1)

$$\text{هزینه یک کشور با ارزیابی تابع } f \text{ به ازای متغیرهای } (p_1, p_2, p_3, \dots, p_{N_{var}}) \text{ یافته می شود. بنابراین}$$

$$C (2)$$

در الگوریتم ICA، برای شروع، تعداد $N_{country}$ کشور اولیه ایجاد و N_{imp} تا از بهترین اعضای این جمعیت (کشورهای دارای کمترین مقدار تابع هزینه) به عنوان استعمارگر انتخاب می شوند. باقیمانده N_{col} تا از کشورها، مستعمراتی را تشکیل می دهند که هر کدام به یک امپراطوری تعلق دارند. مستعمرات اولیه متناسب با قدرت استعمارگرها به آنها تخصیص می یابند. برای انجام این کار، با داشتن هزینه همه استعمارگرها، هزینه نرمالیزه آنها را به صورت زیر در نظر می گیریم.

$$C (3)$$

که در آن c_n ، هزینه استعمارگر n است، $\{c_i\}_{i=1}^n$ می باشد، c_n ، هزینه نرمالیزه بیشترین هزینه میان استعمارگرها و c_n ، هزینه نرمالیزه شده این استعمارگرها، می باشد. هر استعمارگری که دارای هزینه بیشتری باشد (استعمارگر ضعیفتری باشد)، دارای هزینه نرمالیزه کمتری خواهد بود. با داشتن هزینه نرمالیزه، قدرت نسبی نرمالیزه هر استعمارگر، به صورت زیر محاسبه شده و بر مبنای آن، کشورهای مستعمره، بین استعمارگرها تقسیم می شوند.

$$p (4)$$

از یک دید دیگر، قدرت نرمالیزه شده یک استعمارگر، تعداد مستعمراتی است که توسط آن استعمارگر اداره می شود. بنابراین تعداد اولیه مستعمرات یک استعمارگر برابر خواهد بود با

$$N (5)$$

که در آن $N.C_n$ تعداد اولیه مستعمرات یک امپراطوری و N_{col} نیز تعداد کل کشورهای مستعمره موجود در جمعیت کشورهای اولیه است. تابع round نیز تابعی است که نزدیکترین عدد صحیح به یک عدد اشاره را می دهد. با درنظر گرفتن $N.C_n$ برای هر امپراطوری، این تعداد از کشورهای مستعمره اولیه به صورت تصادفی انتخاب و به استعمارگر n تخصیص می یابند. کشورهای استعمارگر با اعمال سیاست جذب (همگون سازی) در راستای محورهای مختلف بهینه سازی، کشورهای مستعمره را به سمت خود می کشند. رقابت استعماری که در شکل (1) نشان داده شده است، در کنار سیاست همگون سازی، هسته اصلی این الگوریتم را تشکیل می دهد و باعث می شود کشورها به سمت مینیمم مطلق تابع حرکت کنند. استعمارگران با توجه به قدرتشان، این مستعمرات را با رابطه (6)، به سمت خود می کشند. قدرت کل هر امپراطوری، با محاسبه قدرت هر دو بخش تشکیل دهنده آن یعنی قدرت کشور استعمارگر، به اضافه درصدی از میانگین قدرت مستعمرات آن، تعیین می شود.

$$\xi \text{mean}\{\text{Cost}(\text{colonies of empire}_n)\} (6)$$

۱- مقدمه

الگوریتم های تکاملی که تا کنون برای بهینه سازی ارائه شده اند، با مدل کردن فرآیند تکامل طبیعی، سعی در یافتن جواب بهینه سازی ارائه شده اند. از الگوریتم های تکاملی که تاکنون برای بهینه سازی ارائه شده اند، می توان به الگوریتم ژنتیک (GA) ارائه شده توسط هالند در سال ۱۹۶۲، الگوریتم بهینه سازی حرکت گروهی ذرات (PSO) معرفی شده توسط کندی، و ابرهارت در سال ۱۹۹۵ [1]، شبیه سازی باز پخت فلزات (SA)، الگوریتم های تکامل فرهنگی و سایر الگوریتم های موجود در این زمینه اشاره کرد. اخیرا الگوریتم جدیدی توسط آتش پیز گرگری [2]، در سال ۲۰۰۷ ارائه شده است که نه از پدیده طبیعی، بلکه از یک پدیده اجتماعی - انسانی الهام گرفته است. این الگوریتم به فرآیند استعمار به عنوان مرحله ای از تکامل اجتماعی - سیاسی بشر نگریسته و با مدل سازی ریاضی این پدیده تاریخی، از آن به عنوان منشاء الهام یک الگوریتم قوی در زمینه بهینه سازی بهره گرفته است. الگوریتم رقابت استعماری با داشتن یک دیدگاه کاملاً جدید به مبحث بهینه سازی، میان علوم انسانی و اجتماعی از یک سو و علوم فنی و ریاضی از سوی دیگر ارتباط برقرار می کند. ارتباط میان این دو شاخه از علم به گونه ای است که غالباً ریاضیات به عنوان ابزاری قوی و دقیق در خدمت علوم انسانی قرار گرفته و به درک و تحلیل نتایج آن کمک می کنند. اما این الگوریتم از پدیده سیاسی - اجتماعی استعمار برای حل مسائل بهینه سازی نیز بهره می گیرد. این الگوریتم با افزایش پیچیدگی و ابعاد مسئله و کاهش تنوع و گیر افتادن در بهینه های محلی، توان و کارآیی خود را از دست می دهد. در این مقاله ما روشنی ارائه کرده ایم که با استفاده از جستجو های محلی و بکارگیری یک جهش، سعی در افزایش توان و تنوع الگوریتم داریم. از طرفی الگوریتم تفاضل تکاملی یک الگوریتم تکاملی است که برای تولید نسل جدید، اطلاعات جهت و فاصله را از اعضای جمعیت فعلی و یا حتی بهترین های محلی آنها استخراج می کند و با تفاضل های نمونه برداری شده الگوریتم را بخوبی هدایت می کند. روش این الگوریتم جستجو های محلی خوبی را به همراه دارد. بنابراین از روش تکامل تفاضلی برای بکارگیری میم در الگوریتم پیشنهادی استفاده شده است.

۲- الگوریتم رقابت استعماری

الگوریتم رقابت استعماری یک الگوریتم جدید در زمینه محاسبات تکاملی است که بر مبنای تکامل اجتماعی - سیاسی انسان پایه گذاری شده است. همانند دیگر الگوریتم های تکاملی، این الگوریتم، نیز با تعدادی جمعیت اولیه تصادفی که هر کدام از آنها یک کشور نامیده می شوند، شروع می شود. تعدادی از بهترین عناصر جمعیت، به عنوان استعمارگر انتخاب و باقیمانده جمعیت نیز به عنوان مستعمره، در نظر گرفته می شوند. در یک مسئله بهینه سازی $T.C_{\text{imperialList}}$ (imperialist colonization problem)، یک آرایه $N_{var} \times 1$ است. این آرایه به صورت زیر تعریف می شود.

۳- تکامل تفاضلی^۲ و بکارگیری آن

این الگوریتم بطور تکرار شونده جمعیت خود را تکامل داده و اصلاح می کند. در هر نسل یک جمعیت جدید بر اساس جمعیت فعلی تولید می شود. برای تولید فرزندان جمعیت جدید، الگوریتم اطلاعات جهت و فاصله را از اعضای جمعیت فعلی استخراج می کند و انحراف تصادفی را برای ایجاد تنوع اضافه می کند. اگر یک فرزند مقدار تابع هدف پایین تری نسبت به یک عضو جمعیت از قبل تعیین شده داشته باشد، با این عضو جمعیت جایگزین خواهد شد. این فرآیند تکامل تا زمانی که معیار توقف برآورده شود ادامه می یابد (به عبارت دیگر بهترین مقدار تابع هدف فعلی از مقدار تعیین شده کوچکتر است). الگوریتم تکامل نسل ها برابر با ماکسیمم مقدار تعیین شده است. الگوریتم تکامل تفاضلی به واسطه تواناییش در حفظ تنوع و انجام جستجوی محلی نسبت به الگوریتم های تکاملی (EA) عملکرد بهتری دارد، اما الگوریتم تکامل تفاضلی مکانیزمی برای استفاده مستقیم از اطلاعات عمومی درباره فضای جستجو برای هدایت جمعیت به سمت فضاهای امیدوار کننده ندارد [3]. الگوریتم رقابت استعماری روش خوبی برای کشف اطلاعات عمومی فضای جستجو می باشد.

روند الگوریتم تکامل تفاضلی به صورت زیر تعریف می شود:

در هر نسل برای هر فرد x_i' از جمعیت pop_d سه فرد x_1 , x_2 و x_3 بصورت تصادفی از جمعیت انتخاب می شود. بازترکیب مطابق با منطق تکامل تفاضلی x_{off}' را مطابق با فرمول زیر تولید می کند:

$$x_{off}' = x_1 + k(x_2 - x_3) \quad 1-3$$

که k ثابتی بین $[0,1]$ است. سپس به منظور افزایش exploration (اكتشاف) این عملگر، هر ژن x_{off}' با ژن معادلش در x_i تعویض می شود. هر تعویض با یک احتمال نرخ جهش یکنواخت رخ می دهد. بنابراین فرزند x_{off} تولید می شود:

(10)

آن دیس ژن تحت تخمین است. $\text{rand}(0,1)$ عددی تصادفی بین ۰ و ۱ می باشد که اگر از مقدار CR کوچکتر باشد ژن از x_i انتخاب می شود و در غیر اینصورت از x_{off}' انتخاب می شود. مقدار تابع شایستگی x_{off} محاسبه می شود. اگر x_{off} نسبت به x_i بهبود یافته باشد، جایگزین x_i می شود و در غیر این صورت یعنی $x_i > f(x_{off})$ باشد، جایگزینی رخ نمی دهد [4].

همانطور که در شکل (۲) دیده می شود نقطه جدید، توسط ترکیب نقطه x_i با یک نقطه x_1 که بصورت تصادفی از روی جمعیت فعلی انتخاب می شود و یک بردار از تفاضل های نمونه برداری شده بصورت تصادفی $(x_2 - x_3)$ تولید می شود. فرد انتخاب شده می تواند، بهترین فرد بین افراد x_{best} و یا خود x_i باشد [5]. استفاده از بردار تفاضل های نمونه برداری شده می تواند جستجوهای محلی را به همراه داشته باشد. از اینرو بخوبی می تواند به عنوان یک مم در الگوریتم ممیتیک به کار گرفته شود. از طرفی، چون در الگوریتم رقابت استعماری نیز می

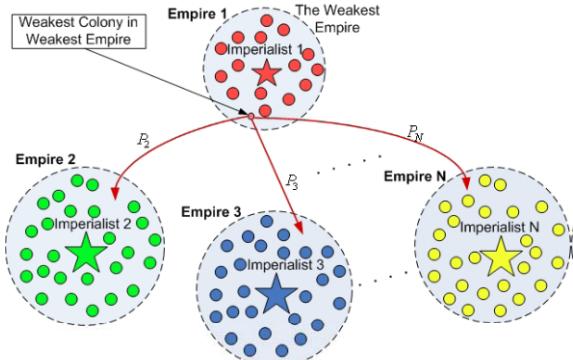
کشور مستعمره، به اندازه X واحد در جهت خط واصل مستعمره به استعمارگر، حرکت کرده و به موقعیت جدید، کشانده می شود. فاصله میان استعمارگر و مستعمره، d و X نیز عددی تصادفی با توزیع یکنواخت (و یا هر توزیع مناسب دیگر) می باشد. یعنی برای X داریم $X \sim U(0, \beta \times d)$

که در آن β عددی بزرگتر از یک و نزدیک به ۲ می باشد. معمولاً $\beta = 2$ در نظر گرفته می شود. این حرکت می تواند با یک انحراف احتمالی همراه باشد. این انحراف به صورت تصادفی و با توزیع یکنواخت انتخاب می شود. کل این حرکت مستعمره به سمت استعمارگر، سیاست جذب نامیده می شود.

$$X = \beta \times \text{rand}(\text{size}(d)).d \quad (8)$$

منظور از $\text{size}(d)$ تعداد ابعاد آن می باشد.

در این الگوریتم انقلاب^۲ و به عبارتی تغییرات ناگهانی در یک کشور اتفاق می افتد. این عمل باعث افزایش قدرت exploration الگوریتم می شود. یعنی در یکی از ابعاد کشور (مثلًا بعد فرهنگی) تغییری رخ دهد. در حین حرکت مستعمرات به سمت کشور استعمارگر، ممکن است بعضی از این مستعمرات به موقعیتی بهتر از استعمارگر برسند در این حالت، کشور استعمارگر و کشور مستعمره، جای خود را با هم عوض می کنند. در این الگوریتم، رقابت استعماری، نقش مهمی دارد. در طی رقابت استعماری، امپراطوری های ضعیف، به تدریج قدرت خود را از دست داده و به مرور زمان با تضعیف شدن، مستعمرات خود را نیز از دست می دهند.



شکل (۱): شمای کلی رقابت استعماری: امپراطوری های بزرگتر، با احتمال بیشتری، مستعمرات امپراطوری های دیگر را تصاحب می کنند برای مدل سازی این رقابت، ابتدا احتمال تصاحب مستعمرات توسط هر امپراطوری با در نظر گرفتن هزینه کل امپراطوری به صورت زیر محاسبه می شود.

N (9)

در این رابطه $T.C.n$ ، هزینه کل امپراطوری n و $N.T.C.n$ نیز، هزینه کل نرمالیزه شده آن امپراطوری می باشد و احتمال تصاحب مستعمره، توسط هر امپراطوری به صورت زیر محاسبه می شود.

P_i (10)

۵- تنظیمات اولیه برای آزمایشات

در این بخش برای ارزیابی عملکرد الگوریتم پیشنهادی در یافتن بهینه عمومی، چندینتابع محک معروف که اغلب در روش‌های بهینه‌سازی تکاملی استفاده می‌شوند، مورد ارزیابی قرار گرفته‌اند. برای بررسی نرخ همگرایی و کیفیت راه حل بهینه‌الگوریتم پیشنهادی، در مقایسه با الگوریتم‌های ICA، CICA، DE [7][6] و GA شبیه‌سازی‌هایی انجام شد که در آن‌ها توابع محک استاندارد با ۳۰ بعد مورد آزمایش قرار گرفتند. هر تابع ۲۰ بار اجرا شد و میانگین مقدار راه حل بهینه بدست آمد. در این‌پیاده‌سازی‌ها، تعداد جمعیت برای تمام الگوریتم‌ها برابر شد. در نظر گرفته شده‌است. تعداد امپراتورها برابر ۱۰ قرار گرفته شد، پارامتر $\beta=2$ ، پارامتر $a=0.5$ و پارامتر $b=0.2$ تنظیم شدند. در الگوریتم DE، پارامتر $\delta=0.0$ و پارامتر $F=0.5$ در نظر گرفته شدند. در الگوریتم GA، نرخ جهش ۰.۰۱ و نرخ ترکیب ۰.۵ در نظر گرفته شدند. تمام پیاده‌سازی‌ها در محیط نرم‌افزار Matlab صورت گرفته‌اند.

۶- نتایج آزمایشات

جدول (۱)، میانگین نتایج حاصل از اعمال الگوریتم‌های مورد مقایسه را برای ۲۰ اجرا در ۳۰ بعد نشان می‌دهد. نتایج برای مقایسه با الگوریتم‌های دیگر در ۱۰۰۰ نسل بدست آمده‌اند. بهترین پاسخ بصورت بلد برگسته شده است.

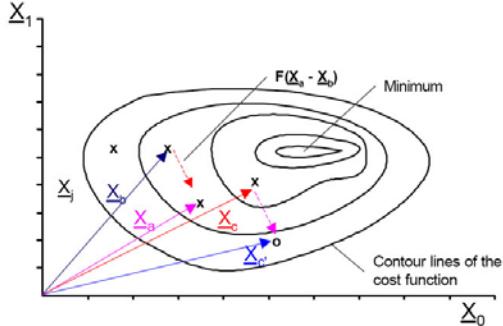
جدول (۱) : مقایسه میانگین نتایج حاصل از ۲۰ بار اجرای الگوریتم

MemeticICA برای توابع محک ۳۰ بعدی

الگوریتم					ایستاده
GA	DE	CICA	ICA	MemeticICA	نام تابع
36.6644	789.0188	10.8793	۱۰.۸۰۰-۰-۱۳	5.6843e-014	Rastrigin
17.6300	4.0132	1.1585e-7	۱۰.۱۷۷۵-۰-۱۵	۰.۳۱۵-۲.۵۲۵۳	Sphere
193.3368	962.0613	23.1916	۲۸.۲۲۵۶	۲۴.۲۲۶۹	Rosenbrock
-1.9314	-0.3131	-2.3712	۰.۰۱۱۱	۰.۰۰۵۰	Griewank
3.2076	5.0587	2.4958	۲.۰۶۳۴e-۱۰	2.3093e-014	Ackley

در تابع Sphere الگوریتم که نتایج عددی نشان می‌دهند، الگوریتم پیشنهادی به بهینه عمومی همگرا شده است و الگوریتم ICA و CICA نیز به نتایج خوبی رسیده‌اند. در تابع Type equation here Rstrigrin الگوریتم پیشنهادی با تفاوت کمی نسبت به ICA نتیجه بهتری در مقایسه با دیگر الگوریتم‌ها بدست آورده است. در تابع Rosenbrock الگوریتم CICA توانسته پاسخ مناسب‌تری بیابد. در تابع Greiwank الگوریتم پیشنهادی عملکرد مناسبی ندارد و الگوریتم CICA نتیجه بسیار بهتری بدست آورده است. در تابع چند ریختی Ackley عملکرد الگوریتم پیشنهادی بسیار خوب بوده و پس از آن الگوریتم ICA به پاسخ مناسبی دست یافته است. در شکل (۳) نمودارهای همگرایی الگوریتم‌های مورد مقایسه برای توابع محک مذکور با ۳۰ بعد نشان داده شده‌اند.

توان گفت به نوعی از جهت و فاصله در حرکت مستعمرات به سمت استعمارگر استفاده می‌شود، برآن شدید تا از تکامل تفاضلی به عنوان مِم برای بهبود این الگوریتم استفاده کیم.



شکل (۲) : نمایی از روند الگوریتم تکامل تفاضلی، x بردار فردی از جمعیت و x_0 بردار فرد جدید تولید شده است

۴- الگوریتم پیشنهادی

همانطور که در بالا بیان شد، روش تکامل تفاضلی به واسطه تواناییش در حفظ تنوع و انجام جستجوی محلی بخوبی می‌تواند به عنوان یک مِم در الگوریتم مُمتیک به کار گرفته شود. از طرفی دیگر، الگوریتم رقابت استعماری نیز به نوعی از جهت و فاصله در حرکت مستعمرات به سمت استعمارگر استفاده می‌کند. بنابراین در الگوریتم پیشنهادی، از روش تفاضل تکاملی به منظور جستجوی محلی استفاده شده است، به این ترتیب که مبنای الگوریتم همان الگوریتم رقابت استعماری است و یک مِم که به صورت زیر تعریف می‌شود به آن افزوده شده است.

در هر حوزه‌ی امپراتوری به همراه مستعمراتش، علاوه بر سیاست جذب و حرکت مستعمرات به سمت امپراتوری، اجازه یک جستجوی محلی در آن حوزه و به نوعی تقلید از بهترین مستعمره داده می‌شود. به این صورت که برداری حاصل از تفاضل بهترین مستعمره و یک مستعمره و یا مستعمره موردنظر تشکیل می‌شود و به آن اضافه می‌شود و به عبارتی راهنمای آن در جستجوی محلی می‌شود.

اگر مستعمره تغییر یافته (مکان جدید) نسبت به قبل بهبود یافته باشد، جایگزین آن می‌شود و در غیراین صورت، جایگزینی رخ نمی‌دهد. ضریب k را می‌توان عددی بین صفر و یک و یا معادل β در سیاست جذب در نظر گرفت.

با بکارگیری مِم بیان شده، مستعمرات رقابت سخت تری برای قویتر شدن با یکدیگر خواهد داشت و کاهش تعداد امپراتوری‌ها کنترل صورت خواهد پذیرفت و فضای مسئله بهتر از قبل مورد کاوش قرار خواهد گرفت.

البته برای آنکه روند کاهش تعداد امپراتوری‌ها و غلبه آنها بر یکدیگر خیلی کند نشود و فاصله بهترین مستعمره تا بقیه حفظ شود، می‌توان از یک جهش استفاده کرد، یعنی به بهترین مستعمره حوزه اجازه جهش به سمت بهترین امپراتوری موجود داده شود.

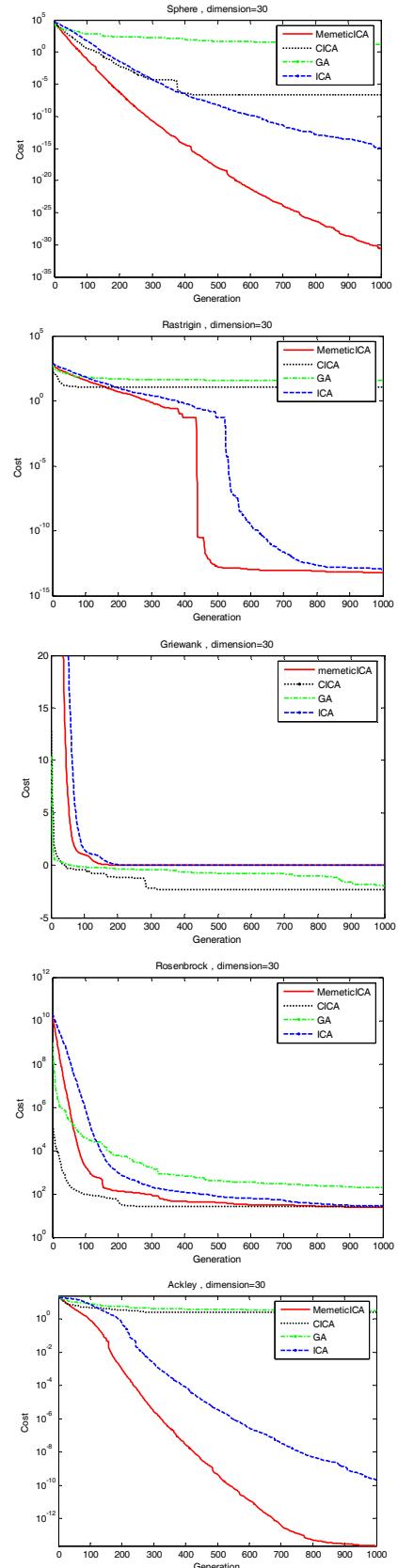
نتایج عددی بدست آمده در جدول ۲، با یکدیگر مقایسه شده است. مقداری که پرنگ شده، بهترین پاسخ می باشد. الگوریتم پیشنهادی در توابع تک ریختی Sphere و Rosenbrock نتایج خوبی را بدست آورده است. در توابع چند ریختی نیز با اختلاف زیاد نسبت به ICA حتی در تابع Schwefel با پیچیدگی بالا، نتایج بسیار خوبی را بدست آورده است.

جدول ۲) مقایسه میانگین نتایج حاصل از ۲۰ بار اجرای الگوریتم

پیشنهادی و ICA برای توابع محک ۱۰۰ و ۲۰۰ بعدی

الگوریتم		ایستا	
ICA	MemeticICA	بعد	تابع
$4.00019 \pm 8.0381e-004$	$4.7874e-006 \pm 3.2280e-006$	100	Ackley
2.5483 ± 0.3782	0.1490 ± 0.1714	200	
5.9567 ± 2.4723	0.8458 ± 0.8086	100	Rastrigin
94.5156 ± 8.7734	28.4534 ± 6.1319	200	
0.0104 ± 0.0180	0.0140 ± 0.0238	100	Griewank
0.2893 ± 0.0833	0.0210 ± 0.0269	200	
165.8159 ± 169.2508	48.1194 ± 70.4080	100	Schwefel
$3.5569e+003 \pm 770.1312$	$1.7674e+003 \pm 460.9391$	200	
$983.4173 \pm 1.7104e+003$	297.6517 ± 308.0428	100	Rosenbrock
$1.1016e+004 \pm 3.4685e+003$	$2.1907e+003 \pm 2.2211e+003$	200	
$9.3698e-006 \pm 4.2799e-006$	$1.1168e-009 \pm 8.5769e-010$	100	Sphere
1.3077 ± 0.3407	0.0023 ± 0.0026	200	

همانطور که در نمودارهای شکل (۵) نمایان است، الگوریتم پیشنهادی روند همگرایی شبیه به الگوریتم اصلی رقابت استعماری دارد ولذا الگوریتم پیشنهادی خصوصیات اصلی الگوریتم پایه خود یعنی رقابت استعماری را حفظ کرده و بعلاوه باعث بهبود آن نیز شده است و با اختلاف خوبی در نسل هایی زودتر از ICA به پاسخ های بهینه تر دست یافته است. هرچه پیچیدگی مسئله بیشتر می شود الگوریتم پیشنهادی کارآیی خود را بهتر نشان می دهد. همانطور که مشخص است در ۲۰۰ بعد نسبت به ۱۰۰ بعد اختلاف دو الگوریتم بسیار بیشتر است. الگوریتم ICA با افزایش ابعاد مسئله تاحدودی کارآیی خود را از دست می دهد و روش پیشنهادی با جستجوهای محلی خود و با بکارگیری مفهوم میتیک و بالا بردن توان مستعمرات و افزایش رقابت، باعث هرچه بهتر شدن فضای جستجوی فضای می شود و به نوعی بر پیچیدگی فضای غلبه می کند. این بهینگی و دست یافتن به جواب های بهتر در نسل های کمتر در تمامی توابع مورد آزمایش قابل مشاهده است.



شکل (۳): نمودار مقایسه شایستگی توابع محک (۳۰ بعد و ۱۰۰۰ نسل) در آزمایشی دیگر، برای مشخص شدن کارآیی الگوریتم پیشنهادی در ابعاد بالا، مقایسات به بطور جداگانه در ۱۰۰ بعد و ۲۰۰ بعد بررسی شده اند.

۷- نتیجه گیری

در این بخش الگوریتم رقابت استعماری ممتدیکی ارائه شده است که از مفهوم تکامل تفاضلی برای تعریف مم آن استفاده شده است. نتایج برروی توابع محک استاندارد با ابعاد مختلف بررسی شد. با توجه به نتایج بدست آمده می توان بیان کرد که در مسائل با ابعاد و پیچیدگی های بالا بکارگیری روش پیشنهادی بسیار موثر است ولی در غیر این صورت بکارگیری الگوریتم ICA و CICA موثرتر است. به عبارتی الگوریتم پیشنهادی هزینه اضافی جستجوهای محلی را به همراه دارد که در حالتی که پیچیدگی کمتر است متحمل شدن این هزینه هر چند کوچک، لزومی ندارد کما اینکه الگوریتم های دیگر نتایج بهتری را عرضه می کنند ولی در موقعی که پیچیدگی مسئله بالاست و الگوریتم های دیگر کارآیی خود را از دست می دهند، بکارگیری روش پیشنهادی مفید می باشد.

مراجع

- [1] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", in proc.IEEE Int. Conf.Neural Networks, vol.5, pp.1942-1948, Australia, 1995.
- [2] E. Atashpaz-Gargari and C. Lucas, "Imperialist Competitive Algorithm: An Algorithm for Optimization inspired by Imperialistic Competition", IEEE Congress on Evolutionary Computation, Singapore, 2007.
- [3] J. Sun, Q. Zhang, E. Tsang, "DE/EDA: A New Evolutionary Algorithm for Global Optimization", 2004.
- [4] A. Caponio, F. Neri, V. Tirronen "Super-fit control adaptation in memetic differential evolution frameworks", Soft Comput Fusion Found Methodol Appl, vol. 13, no. 8, pp. 811–831, 2009.
- [5] A. K. Qin, P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization" In: Proceedings of the IEEE congress on evolutionary computation, vol. 2, pp. 1785–1791, 2005.
- [6] H. Bahrami, K. Faez, M. Abdechiri, "Imperialist Competitive Algorithm using Chaos Theory for Optimization (CICA)" Intelligent System and Applications(ISA), 2010.
- [7] M. Akhtari, K. Faez, "The Application of a CICA Neural Network on Farsi License Plates Recognition, Hybrid Intelligent Systems (HIS)", IEEE, 2010.

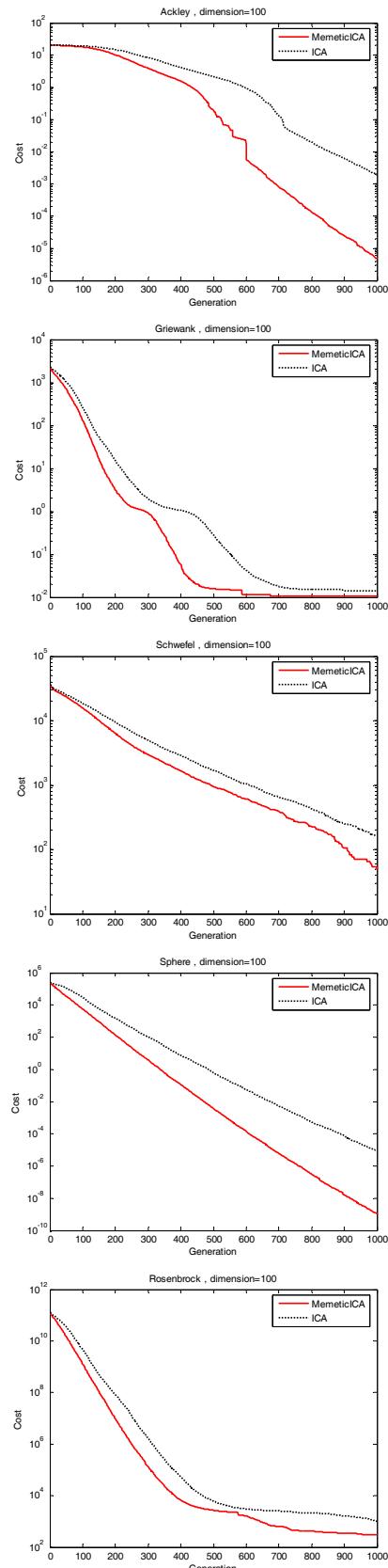
زیرنویس‌ها

¹ Imperialist Competitive Algorithm (ICA)

² Revolution

³ Differential Evolution

⁴ Individual



شکل(۵): نمودار مقایسه شایستگی توابع محک(۱۰۰۰ بعد و ۱۰۰۰ نسل)