

# الگوریتم رقابت استعماری مِمْتیکی

مژده اختری ، محمد رضا میبدی

## چکیده

در این مقاله، یک الگوریتم مِمْتیکی ترکیبی با الگوریتم رقابت استعماری ارائه شده است. در مسائل با پیچیدگی بالا روش رقابت استعماری دچار مشکل می شود و در بهینه های محلی متوقف می شود لذا این مقاله با ارائه روشی مِمْتیکی درصدد 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 مِم درجهت بهبود و افزایش کارایی این الگوریتم استفاده کرده ایم. آزمایشات برای بررسی کارایی الگوریتم ترکیبی ارائه شده، بر روی مسائل محک استاندارد انجام شد. نتایج آزمایشات، کارایی بالای الگوریتم مِمْتیکی ارائه شده را نشان می دهد.

## کلمات کلیدی

الگوریتم مِمْتیک، الگوریتم رقابت استعماری، تکامل تفاضلی، مِم، امپراطوری، جهت، فاصله

## ۱- مقدمه

الگوریتم‌های تکاملی که تا کنون برای بهینه سازی ارائه شده اند، با مدل کردن فرآیند تکامل طبیعی، سعی در یافتن جواب بهینه مسئله دارند. از الگوریتم‌های تکاملی که تاکنون برای بهینه‌سازی ارائه شده‌اند، می‌توان به الگوریتم ژنتیک (GA) ارائه شده توسط هالند در سال ۱۹۶۲، الگوریتم بهینه سازی حرکت گروهی ذرات (PSO) معرفی شده توسط کندی، و ابره‌ه‌ارت در سال ۱۹۹۵ [1]، شبیه‌سازی بازپخت فلزات (SA)، الگوریتم‌های تکامل فرهنگی و سایر الگوریتم‌های موجود در این زمینه اشاره کرد. اخیراً الگوریتم جدیدی توسط آتش پز 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 مرحله‌ای از تکامل اجتماعی - سیاسی بشر نگریسته و با مدل سازی ریاضی این پدیده تاریخی، از آن به عنوان منشأ الهام یک الگوریتم قوی در زمینه بهینه‌سازی بهره گرفته است. الگوریتم رقابت استعماری با داشتن یک دیدگاه کاملاً جدید به مبحث بهینه‌سازی، میان علوم انسانی و اجتماعی از یک سو و علوم فنی و ریاضی از سوی دیگر ارتباط برقرار می‌کند. ارتباط میان این دو شاخه از علم به گونه‌ای است که غالباً ریاضیات به عنوان ابزاری قوی و دقیق در خدمت علوم انسانی قرار گرفته و به درک و تحلیل نتایج آن کمک می‌کنند. اما این الگوریتم از پدیده سیاسی - اجتماعی استعمار برای حل مسائل بهینه‌سازی نیز بهره می‌گیرد. این الگوریتم با افزایش پیچیدگی و ابعاد مسئله و کاهش تنوع و گیر افتادن در بهینه های محلی، توان و کارایی خود را از دست می‌دهد. در این مقاله ما روشی ارائه کرده‌ایم 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 آنها استخراج می‌کند و با تفاضل‌های نمونه برداری شده الگوریتم را بخوبی هدایت می‌کند. روش این الگوریتم جستجوهای محلی خوبی را به همراه دارد. بنابراین از روش تکامل تفاضلی برای بکارگیری مِم 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 پایه‌گذاری شده است. همانند دیگر الگوریتم‌های تکاملی، این الگوریتم، نیز با تعدادی جمعیت اولیه تصادفی که هر کدام از آنها یک کشور نامیده می‌شوند، شروع می‌شود. تعدادی از بهترین عناصر جمعیت، به عنوان استعمارگر انتخاب و باقیمانده جمعیت نیز به عنوان مستعمره، در نظر گرفته می‌شوند. در یک مسئله بهینه‌سازی  $T.C_{\text{empire}} = \text{Cost}(\text{imperialist}_n, \text{colonies}_n)$  یک آرایه‌ی  $1 \times N_{\text{var}}$  است. این آرایه به صورت زیر تعریف می‌شود.

$$(1) \quad C$$

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

$$(2) \quad C$$

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

$$(3) \quad C$$

که در آن  $c_n$ ، هزینه استعمارگر  $n$ ام است،  $\max_i \{c_i\}$  بیشترین هزینه میان استعمارگرها و  $C_n$ ، هزینه نرمالیزه شده این استعمارگرها، می‌باشد. 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 می‌شوند.

$$(4) \quad p$$

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

$$(5) \quad N$$

که در آن  $N.C_n$  تعداد اولیه مستعمرات یک امپراطوری و  $N_{\text{col}}$  نیز تعداد کل کشورهای مستعمره موجود در جمعیت کشورهای اولیه است. تابع round نیز تابعی است که نزدیک‌ترین عدد صحیح به یک عدد اعشار را می‌دهد. با در نظر گرفتن  $N.C_n$  برای هر امپراطوری، این تعداد از کشورهای مستعمره اولیه به صورت تصادفی انتخاب و به استعمارگر  $n$ ام تخصیص می‌یابند. کشورهای استعمارگر با اعمال سیاست جذب (همگون‌سازی) در راستای محورهای مختلف بهینه‌سازی، کشورهای مستعمره را به سمت خود می‌کشند. رقابت استعماری که در شکل (۱) نشان داده شده است، در کنار سیاست همگون‌سازی، هسته اصلی این الگوریتم را تشکیل می‌دهد و باعث می‌شود کشورها به سمت مینیمم مطلق تابع حرکت کنند. استعمارگران با توجه به قدرتشان، این مستعمرات را با رابطه (۶)، به سمت خود می‌کشند. قدرت کل هر امپراطوری، با محاسبه قدرت هر دو بخش تشکیل‌دهنده آن یعنی قدرت کشور استعمارگر، به اضافه درصدی از میانگین قدرت مستعمرات آن، تعیین می‌شود.

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

### ۳- تکامل تفاضلی<sup>۲</sup> و بکارگیری آن

این الگوریتم بطور تکرار شونده جمعیت خود را تکامل داده و اصلاح می‌کند. در هر نسل یک جمعیت جدید بر اساس جمعیت فعلی تولید می‌شود. برای تولید فرزندان جمعیت جدید، الگوریتم اطلاعات جهت و فاصله را از اعضای جمعیت فعلی استخراج می‌کند و انحراف تصادفی را برای ایجاد تنوع اضافه می‌کند. اگر یک فرزند مقدار تابع هدف پایین‌تری 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 می‌یابد (به عبارت دیگر بهترین مقدار تابع هدف فعلی از مقدار تعیین شده کوچکتر است یا تعداد نسل‌ها برابر با ماکسیمم مقدار تعیین شده است). الگوریتم تکامل تفاضلی به واسطه تواناییش در حفظ تنوع و انجام جستجوی محلی نسبت به الگوریتم‌های 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 می‌باشد.

روند الگوریتم تکامل تفاضلی به صورت زیر تعریف می‌شود:

در هر نسل برای هر فرد  $x_i$  از جمعیت  $pop_k$  سه فرد  $x_1$ ،  $x_2$  و  $x_3$  بصورت تصادفی از جمعیت انتخاب می‌شود. بازترکیبی مطابق با منطق تکامل تفاضلی،  $x'_{off}$  را مطابق با فرمول زیر تولید می‌کند:

$$x'_{off} = x_1 + k(x_2 - x_3) \quad ۱-۳$$

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

$$(۱۰)$$

ژاندیس ژن تحت تخمین است.  $rand(0,1)$  عددی تصادفی بین ۰ و ۱ می‌باشد که اگر از مقدار CR کوچکتر باشد ژن از  $x_i$  انتخاب می‌شود و در غیر این صورت از  $x'_{off}$  انتخاب می‌شود.

مقدار تابع شایستگی  $x_{off}$  محاسبه می‌شود. اگر  $x_{off}$  نسبت به  $x_i$  بهبود یافته باشد، جایگزین  $x_i$  می‌شود و در غیر این صورت یعنی  $f(x_{off}) > f(x_i)$  باشد، جایگزینی رخ نمی‌دهد. [4]

همانطور که در شکل (۲) دیده می‌شود نقطه جدید، توسط ترکیب نقطه  $x_i$  با یک نقطه  $x_1$  که بصورت تصادفی از روی جمعیت فعلی انتخاب می‌شود و یک بردار از تفاضل‌های نمونه برداری شده بصورت تصادفی ( $x_2 - x_3$ ) تولید می‌شود. فرد انتخاب شده می‌تواند، بهترین فرد بین افراد  $x_{best}$  و یا خود  $x_i$  باشد [5]. استفاده از بردار تفاضل‌های نمونه برداری شده می‌تواند جستجوهای محلی را به همراه داشته باشد. از اینرو بخوبی می‌تواند به عنوان یک میم در الگوریتم میمیک 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 می‌شود. فاصله میان استعمارگر و مستعمره،  $d$  و  $x$  نیز عددی تصادفی با توزیع یکنواخت (و یا هر توزیع مناسب دیگر) می‌باشد. یعنی برای  $x$  داریم

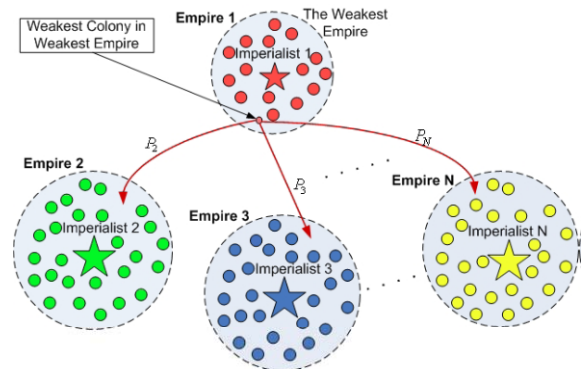
$$x \sim U(0, \beta \times d) \quad (۷)$$

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

$$x = \beta \times rand(size(d)).d \quad (۸)$$

منظور از  $size(d)$  تعداد ابعاد آن می‌باشد.

در این الگوریتم انقلاب<sup>۳</sup> و به عبارتی تغییرات ناگهانی در یک کشور اتفاق می‌افتد. این عمل باعث افزایش قدرت exploration الگوریتم می‌شود. یعنی در یکی از ابعاد کشور (مثلاً 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 می‌کنند. در این الگوریتم، رقابت استعماری، نقش مهمی دارد. در طی رقابت استعماری، امپراطوری‌های ضعیف، به تدریج قدرت خود را از دست داده و به مرور زمان با تضعیف شدن، مستعمرات خود را نیز از دست می‌دهند.



شکل (۱): شمای کلی رقابت استعماری: امپراطوری‌های بزرگتر، با احتمال بیشتری، مستعمرات امپراطوری‌های دیگر را تصاحب می‌کنند

برای مدل سازی این رقابت، ابتدا احتمال تصاحب مستعمرات توسط هر امپراطوری با در نظر گرفتن هزینه کل امپراطوری به صورت زیر محاسبه می‌شود.

$$N \quad (۹)$$

در این رابطه  $T.C_n$ ، هزینه کل امپراطوری  $n$ ام و  $N.T.C_n$  نیز، هزینه کل نرمالیزه شده آن امپراطوری می‌باشد و احتمال تصاحب مستعمره، توسط هر امپراطوری به صورت زیر محاسبه می‌شود.

$$P_i \quad (۱۰)$$

## ۵- تنظیمات اولیه برای آزمایشات

در این بخش برای ارزیابی عملکرد الگوریتم پیشنهادی در یافتن بهینه عمومی، چندین تابع محک معروف که اغلب در روش های بهینه سازی تکاملی استفاده می شوند، مورد ارزیابی قرار گرفته اند. برای بررسی نرخ همگرایی و کیفیت راه حل بهینه الگوریتم پیشنهادی، در مقایسه با الگوریتم های ICA، CICA [6][7]، DE و GA شبیه سازی هایی انجام شد که در آن ها 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 این پیاده سازی ها، تعداد جمعیت برای تمام الگوریتم ها برابر ۱۰۰ در نظر گرفته شده است. تعداد امپراطورها 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 0.01 و نرخ ترکیب 0.5 در نظر گرفته شدند. تمام پیاده سازی ها در محیط نرم افزار Matlab صورت گرفته اند.

## ۶- نتایج آزمایشات

جدول (۱)، میانگین نتایج حاصل از اعمال الگوریتم های مورد مقایسه را برای ۲۰ اجرا در ۳۰ بعد نشان می دهد. نتایج برای مقایسه با الگوریتم های دیگر در ۱۰۰۰ نسل بدست آمده اند. بهترین پاسخ بصورت بلد برجسته شده است.

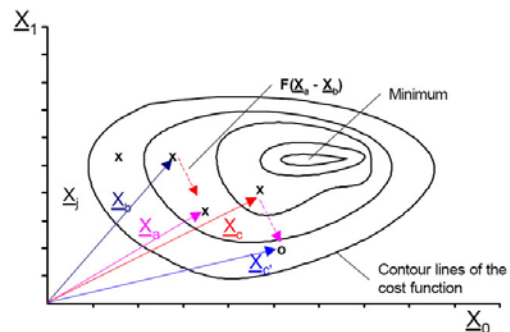
جدول (۱): مقایسه میانگین نتایج حاصل از ۲۰ بار اجرای الگوریتم

MemeticICA برای توابع محک ۳۰ بعدی

ایستا	الگوریتم	نام تابع	GA	DE	CICA	ICA	MemeticICA
Rastrigrin	5.6843e-014	۱.۰۸۰۰e-۰۱۳	36.6644	789.0188	10.8793	۱۰.۸۰۰e-۰۱۳	5.6843e-014
Sphere	۰.۳۱e-۲.۵۳۵۳	۱۰.۱۷۷e-۰۱۵	17.6300	4.0132	1.1585e-7	۱۰.۱۷۷e-۰۱۵	۰.۳۱e-۲.۵۳۵۳
Rosenbrock	۲۴.۲۲۶۹	۲۸.۲۲۵۶	193.3368	962.0613	23.1916	۲۸.۲۲۵۶	۲۴.۲۲۶۹
Griewank	۰.۰۰۵۰	۰.۰۱۱۱	-1.9314	-0.3131	-2.3712	۰.۰۱۱۱	۰.۰۰۵۰
Ackley	2.3093e-014	۲.۰۶۳۴e-۰۱۰	3.2076	5.0587	2.4958	۲.۰۶۳۴e-۰۱۰	2.3093e-014

در تابع Sphere همانطور که نتایج عددی نشان می دهند، الگوریتم پیشنهادی به بهینه عمومی همگرا شده است و الگوریتم ICA و CICA نیز به نتایج خوبی رسیده Type equation here. اند. در تابع Rstrigrin الگوریتم پیشنهادی با تفاوت کمی نسبت به ICA نتیجه بهتری در مقایسه با دیگر الگوریتم ها بدست آورده است. در تابع Rosenbrock الگوریتم CICA توانسته پاسخ مناسب تری بیابد. در تابع Griewank 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 الگوریتم های مورد مقایسه برای توابع محک مذکور با ۳۰ بعد نشان داده شده اند.

توان گفت به نوعی از جهت و فاصله در حرکت مستعمرات به سمت استعمارگر استفاده می شود، برآن شدیم تا از تکامل تفاضلی به عنوان مِم برای بهبود این الگوریتم استفاده کنیم.



شکل (۲): نمایی از روند الگوریتم تکامل تفاضلی،  $x$  بردار فردی از جمعیت و  $0$  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 می تواند به عنوان یک مِم در الگوریتم مِمیتیک به کار گرفته شود. از طرفی دیگر، الگوریتم رقابت استعماری نیز به نوعی از جهت و فاصله در حرکت مستعمرات به سمت استعمارگر استفاده می کند. 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 می شود به آن افزوده شده است.

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

اگر مستعمره تغییر یافته (مکان جدید) نسبت به قبل بهبود یافته باشد، جایگزین آن می شود و درغیراین صورت، جایگزینی رخ نمی دهد. ضریب  $k$  را می توان عددی بین صفر و یک و یا معادل  $\beta$  در سیاست جذب در نظر گرفت.

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

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

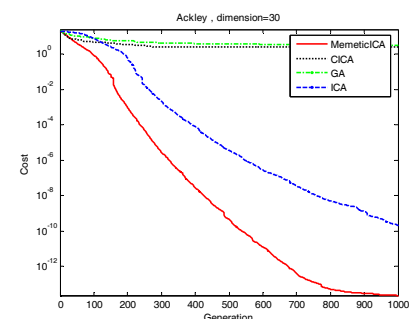
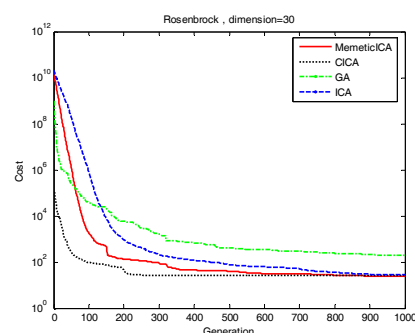
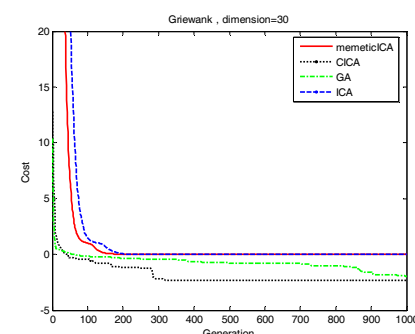
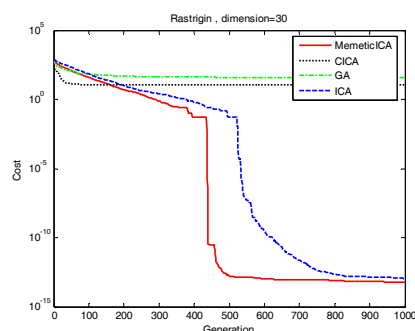
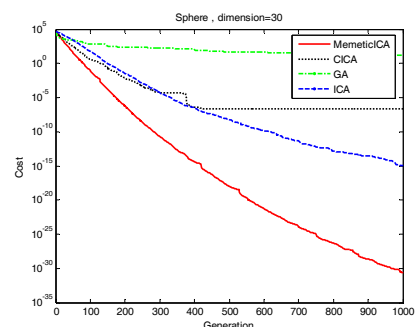
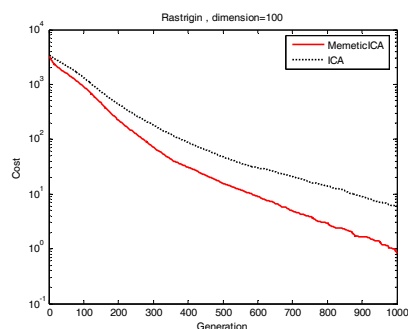
نتایج عددی بدست آمده در جدول ۲، با یکدیگر مقایسه شده است. مقادیری که پررنگ شده، بهترین پاسخ می باشد. الگوریتم پیشنهادی در توابع تک ریختی Sphere و Rosenbrock نتایج خوبی را بدست آورده است. در توابع چند ریختی نیز با اختلاف زیاد نسبت به ICA حتی در تابع Schwefel با پیچیدگی بالا، نتایج بسیار خوبی را بدست آورده 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
$0.0019 \pm 8.0381e-004$	<b><math>4.7874e-006 \pm 3.2280e-006</math></b>	100	Ackley
$2.5483 \pm 0.3782$	<b><math>0.1490 \pm 0.1714</math></b>	200	
$5.9567 \pm 2.4723$	<b><math>0.8458 \pm 0.8086</math></b>	100	Rastrigin
$94.5156 \pm 8.7734$	<b><math>28.4534 \pm 6.1319</math></b>	200	
<b><math>0.0104 \pm 0.0180</math></b>	$0.0140 \pm 0.0238$	100	Griewank
$0.2893 \pm 0.0833$	<b><math>0.0210 \pm 0.0269</math></b>	200	
$165.8159 \pm 169.2508$	<b><math>48.1194 \pm 70.4080</math></b>	100	Schwefel
$3.5569e+003 \pm 770.1312$	<b><math>1.7674e+003 \pm 460.9391</math></b>	200	
$983.4173 \pm 1.7104e+003$	<b><math>297.6517 \pm 308.0428</math></b>	100	Rosenbrock
$1.1016e+004 \pm 3.4685e+003$	<b><math>2.1907e+003 \pm 2.2211e+003</math></b>	200	
$9.3698e-006 \pm 4.2799e-006$	<b><math>1.1168e-009 \pm 8.5769e-010</math></b>	100	Sphere
$1.3077 \pm 0.3407$	<b><math>0.0023 \pm 0.0026</math></b>	200	

همانطور که در نمودارهای شکل (۵) نمایان است، الگوریتم پیشنهادی روند همگرایی شبیه به الگوریتم اصلی رقابت استعماری دارد و لذا الگوریتم پیشنهادی خصوصیات اصلی الگوریتم پایه خود یعنی رقابت استعماری را حفظ کرده و علاوه 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 کارایی 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 کارایی خود را از دست می دهد و روش پیشنهادی با جستجوهای محلی خود و با بکارگیری مفهوم مِمْتیک و بالا بردن توان مستعمرات و افزایش رقابت، باعث هرچه بهتر شدن جستجوی فضا می شود و به نوعی بر پیچیدگی فضا 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 کارایی الگوریتم پیشنهادی در ابعاد بالا، مقایسات به طور جداگانه در ۱۰۰ بعد و ۲۰۰ بعد بررسی شده اند.

## ۷- نتیجه گیری

در این بخش الگوریتم رقابت استعماری مبتنی بر ارائه شده است که از مفهوم تکامل تفاضلی برای تعریف میم آن استفاده شده است. نتایج بر روی 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 مواقعی که پیچیدگی مسئله بالاست و الگوریتم های دیگر کارایی 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.

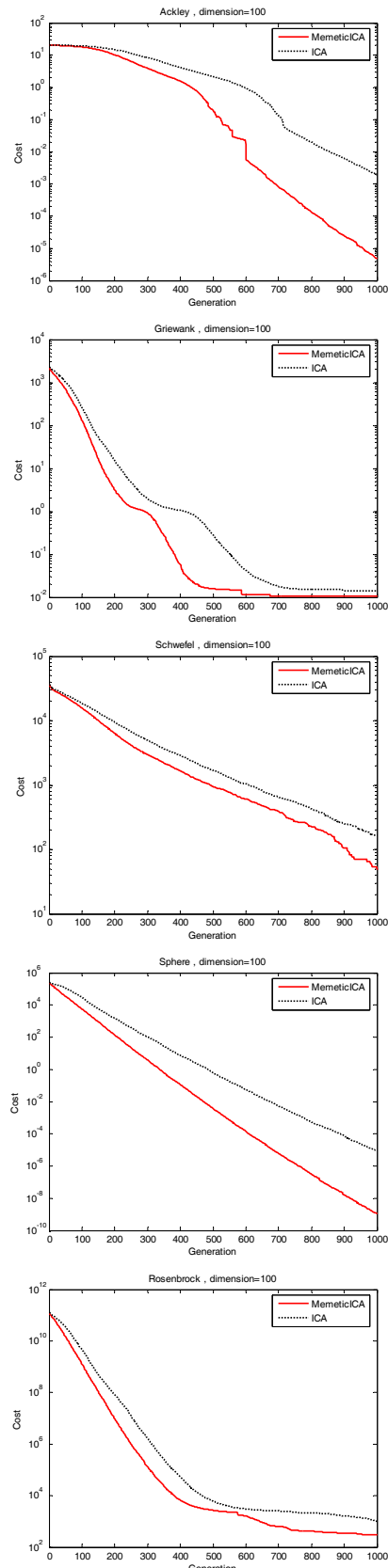
## زیر نویس ها

<sup>1</sup> Imperialist Competitive Algorithm (ICA)

<sup>2</sup> Revolution

<sup>3</sup> Differential Evolution

<sup>4</sup> Individual



شکل (۵): نمودار مقایسه شایستگی توابع محک (۱۰۰ بعد و ۱۰۰۰ نسل)