

(الگوریتم رقابت استعماری + آتاماتاهای یادگیر)

مرجان عبدچیری^۱، محمدرضا میبدی^۲

دانشکده مهندسی برق، کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی، ایران، قزوین

marjan.abdechiri@qiau.ac.ir

دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، ایران، تهران

mmeybodi@aut.ac.ir

چکیده- در این مقاله الگوریتم ترکیبی، به نام الگوریتم رقابت استعماری با آتاماتای یادگیر (ICALA) معرفی شده است. الگوریتم رقابت استعماری (ICA) از ایده تکامل سیاسی- اجتماعی بشر الهام گرفته است. در این الگوریتم تعدادی کشور استعمارگر همراه با کشورهای مستعمراتشان به جستجو برای یافتن نقطه بهینه عمومی برای حل مسئله بهینه سازی می پردازند. در این مقاله با استفاده از آتاماتاهای یادگیر، شعاع و زاویه حرکت آن ها به سمت استعمارگرها بطور پویا در طی تکرارها تنظیم شده و در نتیجه یک جستجوی هدفمندتر در الگوریتم رقابت استعماری انجام می شود. عملکرد الگوریتم پیشنهادی با چهار تابع محک با نام های (Griewank, Rosenbrock, Sphere, Rastrigin) مورد ارزیابی قرار گرفت. همچنین از الگوریتم پیشنهادی برای حل مسأله SAT نیز استفاده شده است. نتایج تجربی حاصل از اجرای الگوریتم رقابت استعماری با آتاماتای یادگیر نشان دادند که کیفیت راه حل بهینه عمومی و سرعت همگرایی به نقطه بهینه در این الگوریتم در مقایسه با الگوریتم اولیه، الگوریتم ژنتیک و الگوریتم بهینه سازی حرکت گروهی ذرات بهبود قابل توجهی یافته است.

مقدمه

سیاست کشورها استفاده شده است. در الگوریتم پیشنهادی اصلی ICALA بطور همزمان در هر تکرار شعاع و زاویه حرکت با آتاماتاهای تنظیم می شود.

الگوریتم بهبود یافته در چند تابع محک استاندارد، که معمولاً در الگوریتم های تکاملی استفاده می شوند، آزمایش شده است. همچنین از این الگوریتم برای حل کارآمدتر مسأله SAT نیز استفاده شده است. نتایج تجربی بدست آمده از اجرای الگوریتم پیشنهادی در توابع محک و نمونه های مختلف SAT نشان داد که سرعت همگرایی و کیفیت راه حل بدست آمده به مراتب بهتر از الگوریتم ICA، الگوریتم ژنتیک و الگوریتم بهینه سازی حرکت گروهی ذرات است.

این مقاله به صورت زیر سازماندهی شده است: در بخش دوم به معرفی الگوریتم ICA و آتاماتای یادگیر و تعریف مسأله SAT می پردازیم. سپس در بخش سوم الگوریتم پیشنهادی در سه فاز مجزا ارائه می شود. در بخش چهارم نتایج تجربی بدست آمده از اجرای الگوریتم پیشنهادی با نتایج الگوریتم های ICA، PSO و GA مقایسه شده و نتایج را مورد تحلیل و بررسی قرار می دهیم و در بخش آخر به نتیجه گیری و جمع بندی می پردازیم.

الگوریتم رقابت استعماری

الگوریتم رقابت استعماری (ICA) 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 آن ها یک کشور نامیده می شوند، شروع می شود. تعدادی از بهترین عناصر جمعیت (معادل نخبه ها در الگوریتم ژنتیک) به عنوان استعمارگر انتخاب و باقیمانده جمعیت نیز به عنوان مستعمره، در نظر گرفته می شوند. در یک مسأله بهینه سازی N_{var} بعدی، یک کشور، یک آرایه $1 \times N_{var}$ است. این آرایه به صورت زیر تعریف می شود.

$$\text{Country} = [P_1, P_2, P_3, \dots, P_{N_{var}}] \quad (1)$$

الگوریتم های تکاملی گوناگونی تاکنون برای بهینه سازی ارائه شده اند که از آن میان می توان به الگوریتم ژنتیک (GA) [1]، الگوریتم بهینه سازی حرکت گروهی ذرات (PSO) [2]، شبیه سازی سرمایه های فلزات (SA) [3]، الگوریتم های تکامل فرهنگی [4] و سایر الگوریتم های موجود در این زمینه اشاره کرد. اخیراً الگوریتم جدیدی با نام الگوریتم رقابت استعماری^۳، توسط آتش پز گرگری و لوکس در سال ۲۰۰۷ ارائه شده است [5] که نه از پدیده طبیعی، بلکه از یک پدیده اجتماعی- انسانی الهام گرفته است.

در الگوریتم ICA کشورها (مستعمرات) با شعاع و زاویه معین به سمت کشور استعمارگر حرکت می کنند. به همین علت الگوریتم ICA در مینیمم محلی افتاده و توانایی فرار از آن را ندارد. برای رفع این مشکلات تاکنون الگوریتم ها مختلفی ارائه شده اند که از آن جمله می توان به الگوریتم CICA [6] اشاره نمود. باید اشاره شود که الگوریتم ICA در کاربردهای مختلف نیز مورد استفاده قرار گرفته است [7][8].

در این مقاله روشی پیشنهاد شده است که با استفاده از آتاماتاهای یادگیر، زاویه و شعاع حرکت در فرایند جستجوی الگوریتم ICA بطور پویا تنظیم می شود. با این روش برای الگوریتم ICA امکان یک جستجوی هدفمند فراهم شده است، که این امر موجب بهبود عملکرد الگوریتم ICA در حل مسائل بهینه سازی می شود.

در این مقاله برای تحلیل و بررسی کامل عملکرد الگوریتم پیشنهادی ICALA این الگوریتم در سه فاز بیان می شود. در ابتدا الگوریتم ICA_LA_Angle معرفی می شود. در این الگوریتم از آتاماتاهای یادگیر جهت تنظیم زاویه حرکت کشورها در هر تکرار استفاده شده است. در فاز دوم یعنی الگوریتم پیشنهادی ICA_LA_Radius از آتاماتاهای یادگیر جهت تنظیم شعاع حرکت در

هزینه یک کشور با ارزیابی تابع f به ازای متغیرهای $(P_1, P_2, P_3, \dots, P_{N_{var}})$ یافته می‌شود. بنابراین

$$C_i = f(\text{country}_i) = f(P_{11}, P_{12}, P_{13}, \dots, P_{1N_{var}}) \quad (2)$$

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 می‌شوند. باقیمانده N_{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.

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 (۳)، 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.

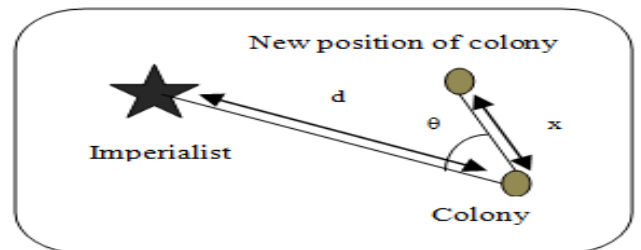
$T.C_n = \text{Cost}(\text{imperialist}_n) + \xi \text{mean}\{\text{Cost}(\text{colonies of empire}_n)\} \quad (3)$
کشور مستعمره، به اندازه x واحد در جهت خط واصل مستعمره به استعمارگر، حرکت کرده و به موقعیت جدید، کشانده می‌شود. در شکل ۱، فاصله میان استعمارگر و مستعمره با d نشان داده شده است و x نیز عددی تصادفی با توزیع یکنواخت (و یا هر توزیع مناسب دیگر) می‌باشد.

$$x \sim U(0, \beta \times d) \quad (4)$$

که در آن β عددی بزرگتر از یک و نزدیک به ۲ می‌باشد. یک انتخاب مناسب می‌تواند $\beta = 2$ باشد. یک انتخاب مناسب می‌تواند $\beta = 2$ باشد. همچنین زاویه حرکت را به صورت توزیع یکنواخت زیر در نظر گرفته شده است.

$$\theta \sim U(-\gamma, \gamma) \quad (5)$$

در الگوریتم ICA، با یک انحراف احتمالی، مستعمره در مسیر جذب استعمارگر پیش می‌رود. این انحراف با زاویه θ نشان داده شده است که θ به صورت تصادفی و با توزیع یکنواخت انتخاب می‌شود.



شکل ۱. حرکت مستعمرات به سمت استعمارگر (سیاست جذب).

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. رقابت استعماری باعث می‌شود که به مرور زمان، به حالتی رسید که

در آن تنها یک امپراطوری در دنیا وجود دارد که آن را اداره می‌کند. این حالت زمانی است که الگوریتم رقابت استعماری با رسیدن به نقطه بهینه تابع هدف، متوقف می‌شود.

آتاماتای یادگیر

آتاماتای یادگیر یک سیستم تصمیم‌گیری تطبیقی است. آتاماتای یادگیر می‌تواند بهترین عمل را در یک محیط ناشناخته یاد بگیرد. این یادگیری بر اساس احتمالات در یک محیط تصادفی ناشناخته انجام می‌شود [9-14]. مبنای کار آتاماتاهای یادگیر بر پایه یافتن عمل بهینه از مجموعه اعمال آتاماتا از طریق تعامل با محیط می‌باشد. در هر مرحله آتاماتای یادگیر از مجموعه اعمال خود بر اساس توزیع احتمال اعمال، عملی مانند α_n را انتخاب نموده و به محیط تصادفی ارسال می‌نماید. محیط تصادفی این عمل را ارزیابی نموده و در پاسخ سیگنال ارزیابی عمل بررسی شده را که سیگنال تقویتی β_n نامیده می‌شود، به آتاماتا بازگشت می‌دهد. آتاماتا یادگیر از سیگنال تقویتی برای بروزرسانی ساختار توزیع احتمال اعمالش استفاده نموده و در تکرار این فرآیند احتمال انتخاب عمل بهینه را افزایش می‌دهد. وضعیت داخلی آتاماتا در لحظه n را با بردار احتمال عمل‌های آتاماتا $P(n)$ که در زیر آمده است، نشان داده می‌شود:

$$P(n) \equiv \{P_1(n), P_2(n), \dots, P_r(n)\} \quad (6)$$

$$\sum_{i=1}^r p_i(n) = 1, \forall n, p_i(n) = \text{Prob}[\alpha(n) = \alpha_i] \quad (7)$$

در آغاز فعالیت آتاماتا، احتمال عمل‌های آن با هم برابر و مساوی $1/r$ می‌باشند (که r تعداد عمل‌های آتاماتا می‌باشد). ورودی محیط یکی از r عمل انتخاب شده آتاماتا است. خروجی (پاسخ) محیط به هر عمل i توسط β_i مشخص می‌شود. در چنین محیطی $\beta_i = 1$ بعنوان پاسخ نامطلوب و $\beta_i = 0$ بعنوان پاسخ مطلوب در نظر گرفته می‌شوند. تغییر احتمال عمل‌ها بصورت زیر می‌باشد.

الف- پاسخ مطلوب از محیط

$$p_j(n+1) = \begin{cases} p_j(n) + a[1 - p_j(n)] & j = i \\ (1-a)p_j(n) & \forall j \neq i \end{cases} \quad (8)$$

ب- پاسخ نامطلوب از محیط

$$p_j(n+1) = \begin{cases} (1-b)p_j(n) & j = i \\ (b/r - 1) + (1-b)p_i & \forall j \neq i \end{cases} \quad (9)$$

با توجه به مقادیر a و b در روابط فوق، سه حالت را می‌توان در نظر گرفت. اگر مقادیر a و b برابر باشند، آتاماتای یادگیر L_{RP} نامیده می‌شود. زمانیکه b مساوی با صفر باشد آتاماتای یادگیر L_{RI} نامیده می‌شود. اگر $b < a$ باشد، آتاماتای یادگیر $L_{R\&P}$ نامیده می‌شود.

مسأله صدق‌پذیری

مسأله صدق‌پذیری^۱ (SAT) یکی از مسائل معروف NP-Complete، است. مسأله SAT دارای سه بخش اصلی می‌باشد.

۱. یک مجموعه با n متغیر: x_1, x_2, \dots, x_n

^۱ Satisfiability Problem

۲. یک مجموعه از متغیرها (Literal): یک Literal یک متغیر $(E = x)$ یا یک متغیر منفی $(E = \bar{x})$.

۳. یک مجموعه با m جمله (Clause) مشخص: C_1, C_2, \dots, C_m . هر جمله شامل Literalهایی است که با or منطقی با هم ترکیب شده‌اند.

هدف از مسأله صدق‌پذیری مشخص کردن این است که آیا مقداردی به متغیرها وجود دارد که فرمول CNF را صدق‌پذیر نماید.

$$E = C_1 \wedge C_2 \wedge \dots \wedge C_m = \prod_{j=1}^m C_j \quad (10)$$

همانطور که در رابطه ۱۰، مشاهده می‌شود، نمونه‌های SAT شامل چندین جمله و هر جمله شامل تعدادی متغیر می‌باشند. وقتی هر جمله در E شامل k کلمه باشد این گزاره یک مسأله k -SAT نامیده می‌شود. در این مقاله تمرکز ما روی مسأله 3-SAT است که در واقع هر جمله شامل سه کلمه می‌باشد. هدف مسأله صدق‌پذیری این است که آیا فرمول گزاره‌ای E با مقدار درست ارزیابی می‌شود یا خیر. اگر بتوان با تعیین مقادیر برای تمامی متغیرها به ارزیابی درستی برای گزاره E رسید، آن مسأله صدق‌پذیر است و در غیر این صورت E صدق‌ناپذیر نامیده می‌شود. برای مثال در رابطه $E = x_1 \wedge \bar{x}_2$ در واقع E یک نمونه مسأله 1-SAT با دو متغیر و دو جمله می‌باشد. سائز فضای جستجو S برابر $|S| = 2^n$ است که همان طور که ملاحظه می‌شود، با تغییر تعداد متغیرها، فضای جستجو به صورت نمایی بزرگ می‌شود. به همین دلیل این مسأله زیر مجموعه‌ای از مسائل NP-Complete است. در فرمول‌بندی یک نمونه گسسته از SAT در فضای پیوسته، متغیرهای گسسته از SAT را به فضای پیوسته، تبدیل می‌شوند. بنابراین این تبدیل باعث تغییر در تابع هدف می‌شود که در حالت پیوسته باعث کمتر شدن تعداد مینیمم‌های محلی در مسیر می‌شود. زیرا این روش نسبت به حالت گسسته یکنواختی بیشتری دارد. دو شکل این فرمول‌ها در زیر آورده شده‌اند. برای فرمول‌های بدون محدودیت پیوسته، فرمول‌های زیادی در این قالب وجود دارند مانند فرمول ساده UniSAT [15].

$$\min_{y \in E^n} f(y) = \sum_{i=1}^m C_i(y) \quad (11)$$

که

$$C_i(y) = \prod_{j=1}^n q_{i,j}(y_j) \quad (12)$$

همچنین داریم

$$q_{i,j}(x_j) = \begin{cases} |y_j - T| & \text{if } x_j \text{ in } C_i \\ |y_j + F| & \text{if } \bar{x}_j \text{ in } C_i \\ 1 & \text{otherwise} \end{cases} \quad (13)$$

راه حل فرمول اصلی در $f(y)=0$ می‌باشد. الگوریتم UniSAT5، می‌تواند به طور موثر با الگوریتم‌های جستجوی محلی حریصانه حل شوند. طبق فرمول‌های بالا یک تابع CNF به شکل زیر در می‌آید.

$$(x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \quad (14)$$

به فرم زیر تبدیل می‌شود.

$$f(y) = |y_1 - 1| |y_2 + 1| + |y_1 + 1| |y_2 - 1| |y_3 - 1| \quad (15)$$

راه حل فرمول SAT برابر نقاط مینیمم تابع هدف بالا می‌باشد. یافتن مقدار درست برای F برابر با مقدار صفر رابطه $F(y)$ است. الگوریتم‌های مختلفی در حوزه گسسته و پیوسته برای حل مسأله SAT ارائه شده‌اند [16-20].

الگوریتم‌های پیشنهادی

در مدل ICA استاندارد بعثت ماهیت الگوریتم، توانایی جستجوی محلی بسیار بیشتر از توانایی جستجوی عمومی می‌باشد. مشکل این الگوریتم بهینه‌سازی، این است که در طول فرایند جستجو در فضای مسئله احتمال افتادن در مینیمم محلی وجود دارد. دو دلیل این مشکل: ۱- در الگوریتم استاندارد ارائه شده، پارامتر θ ، زاویه حرکت کلونی‌ها به سمت کشورهای استعمارگر، دارای توزیع یکنواخت و ثابت می‌باشد. ۲- یکی دیگر از دلایل گیر افتادن در مینیمم محلی، شعاع حرکت کلونی‌ها به سمت کشورهای استعمارگر می‌باشد که در الگوریتم ICA استاندارد، بصورت توزیع یکنواخت در نظر گرفته شده است. این پارامتر نیز مانند پارامتر θ برای جستجو در فضای مسئله باعث گیر افتادن در مینیمم محلی می‌شود.

در این گزارش سه مدل جدید برای الگوریتم ICA به نام‌های ICA-LA-Angle، ICA-LA-Radius و ICALA پیشنهاد می‌شوند. در الگوریتم‌های پیشنهادی برای هر کشور استعمارگر یک مجموعه آتاماتای یادگیر در نظر گرفته شده است. هر آتاماتا نیز دارای دو عمل می‌باشد. در این الگوریتم‌ها از آتاماتای یادگیر با دو عمل «ادامه مسیر در راستای قبل» و «ادامه مسیر در راستای بهترین کشور استعمارگر» استفاده شده است.

مدل پیشنهادی اول ICA-LA-Angle

اولین مدل پیشنهادی ICA-LA-Angle است که ترکیبی از آتاماتای یادگیر و بهینه‌سازی کشورها می‌باشد. در این مدل از آتاماتای یادگیر جهت کنترل حرکت کلونی‌ها استفاده می‌شود و آتاماتای یادگیر دارای دو عمل «ادامه مسیر در راستای قبل» و «ادامه مسیر در راستای بهترین کشور استعمارگر» می‌باشد. در الگوریتم پیشنهادی در صورتیکه آتاماتای یادگیر عمل «ادامه مسیر در راستای بهترین کشور استعمارگر» را انتخاب نماید زاویه اولیه کلونی در محاسبه نحوه حرکت جدید کلونی نادیده گرفته می‌شود. بدین نحو کلونی‌ها به جستجوی سراسری در فضای جستجو می‌پردازند. در صورتیکه آتاماتای یادگیر عمل «ادامه مسیر در راستای قبل» را انتخاب نماید، جهت محاسبه زاویه حرکت جدید کلونی از زاویه قبلی کلونی استفاده می‌شود. چنین عملکردی که در آن از اطلاعات مفید مرحله قبل استفاده می‌شود توانایی exploit نامیده می‌شود. در این صورت کلونی‌ها به جستجوی محلی می‌پردازند. الگوریتم ICA-LA-Angle در زیر ارائه شده است:

در ابتدا، موقعیت کشورهای استعمارگر و همچنین بردار احتمالات انتخاب اعمال آتاماتای یادگیر مقداردهی اولیه می‌شوند. سپس تا زمانیکه حداکثر تعداد گام‌ها انجام گردد و یا هدف مورد نظر حاصل شود، مراحل زیر تکرار می‌شوند:

- ۱- آتاماتای یادگیر یکی از اعمالش را بر طبق بردار احتمال اعمالش، انتخاب می‌کند.

۲- با توجه به عمل انتخاب شده، نحوه بروزسانی زاویه کلونی‌ها تعیین می‌شود و سپس کلونی‌ها، راستای حرکت و موقعیت خود را بروز می‌کنند.

۳- بر اساس نتایج بروزسانی موقعیت کلونی‌ها، عمل آتاماتا یادگیر، ارزیابی می‌شود و بردار احتمال انتخاب اعمال آتاماتای یادگیر اصلاح می‌شود.

عملی که آتاماتای یادگیر در هر گام برمی‌گزیند، تعیین کننده شیوه بروزکردن زاویه کلونی‌ها در آن گام می‌باشد. در صورت انتخاب عمل «ادامه مسیر در

می‌پردازد. در عمل «ادامه مسیر در راستای بهترین کشور استعمارگر» بصورت زیر است.

$$r_{iter} = r_{(iter-1)best_colony} + r_{iter-1} \quad (21)$$

این جستجوی عمومی بدلیل اینکه از اطلاعات مفید مرحله قبل استفاده نمی‌کند توانایی explore الگوریتم را بالا می‌برد.

مدل پیشنهادی سوم ICA-LA

مدل پیشنهادی سوم ICA-LA است که ترکیبی از آتاماتای یادگیر و بهینه‌سازی کشورها می‌باشد. در این مدل از آتاماتای یادگیر جهت کنترل شعاع و زاویه حرکت مستعمرات به سمت کشورهای استعمارگر بطور همزمان استفاده می‌شود.

عملی که آتاماتای یادگیر در هر گام برمی‌گزیند، تعیین‌کننده شیوه بروز کردن شعاع و زاویه حرکت کلونی‌ها در آن گام می‌باشد. بروزسانی شعاع و زاویه حرکت در هر گام با روابط (16,18) و (19,21) بطور همزمان انجام می‌شوند. سیگنال تقویتی هم با رابطه زیر تولید می‌شود.

$$\beta = \begin{cases} 0 & \text{if } cost_{i+1}(empire) > cost_i(empire) \\ 1 & \text{otherwise} \end{cases} \quad (22)$$

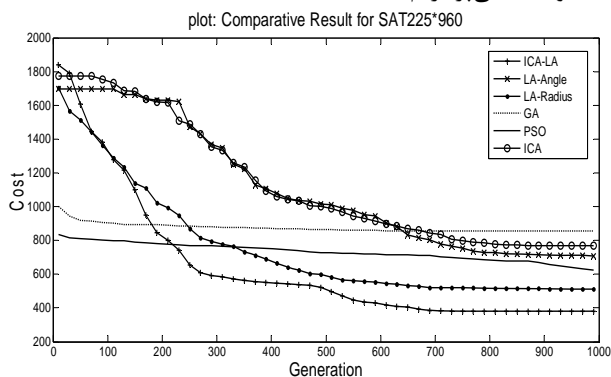
شیوه ارزیابی عمل انتخاب شده، به این صورت است که اگر قدرت کشور استعمارگر نسبت به گام قبل بیشتر باشد، عمل انتخاب شده مثبت و در غیر اینصورت منفی ارزیابی می‌شود. هر دو الگوریتم پیشنهادی را بر روی چند دیتابیس از قبل معرفی شده تست و نتایج الگوریتم‌های پیشنهادی را در ادامه مشاهده می‌کنید. در جدول ۸، پارامترهای مربوط به الگوریتم‌های پیشنهادی و الگوریتم‌های ICA، PSO، GA آورده شده است.

جدول ۱. پارامترهای تنظیم شده برای هر الگوریتم.

	Parameters
ICA	NumOfCountries = 30, NumOfInitialImperialists = 5, NumOfDecades = 1000, RevolutionRate = 0.3, AssimilationCoefficient = 2, Zeta = 0.02, actionNumber = 2, LRP: Epsi = 0.01, a = 0.1, b = Epsi*a for Angle. LRI: a=0.1, b=0. LRP: a=b=0.01.
GA	PopSize = 30, MaxGenerations = 1000, CrossPercent = 70/100, SelectionMode: Tournament
PSO	C1 = 1.5, C2 = 1.5, Inertia = .1, ParticleSize = 30, MaxIter = 1000

تحلیل و بررسی نتایج

در این بخش به تحلیل و مقایسه نتایج الگوریتم‌های پیشنهادی و الگوریتم‌های PSO، GA و ICA می‌پردازیم.



شکل ۲. میانگین بهترین تابع شایستگی در ۱۲۰ اجرا.

راستای قبل»، بروزسانی سرعت ذرات مطابق با فرمول زیر انجام می‌گردد. در این صورت $\alpha = 0.01$ در نظر گرفته شده است. در ادامه مسیر در راستای قبل از رابطه احتمالی زیر استفاده می‌شود.

$$\theta_{iter} = 0.15(\theta_{iter-1} + \alpha) + 0.85(\theta_{iter-1} - \alpha) \quad (16)$$

در الگوریتم رقابت استعماری استاندارد پارامتر زاویه حرکت همه کلونی‌ها مقدار ثابت ۲ را دارند.

$$\theta \sim U(-\gamma, \gamma) \quad (17)$$

در حالی که در الگوریتم پیشنهادی در هر گام که پاسخ دریافتی از محیط مناسب باشد. در واقع با این عملکرد کلونی در اطراف کشور استعمارگر خود به جستجو می‌پردازد. این جستجوی محلی بدلیل اینکه از اطلاعات مفید مرحله قبل استفاده می‌کند توانایی exploit الگوریتم را بالا می‌برد. در عمل «ادامه مسیر در راستای بهترین کشور استعمارگر» بصورت زیر است.

$$\theta_{iter} = |\theta_{(iter-1)best_im} - \theta_{iter-1}| \quad (18)$$

در واقع انتخاب عمل «ادامه مسیر در راستای قبل» انجام یک جستجوی محلی را در پی خواهد داشت و انتخاب عمل «ادامه مسیر در راستای بهترین کشور استعمارگر»، باعث جستجوی سراسری و کشف قسمت‌هایی ناشناخته از فضای جستجو می‌گردد. شیوه ارزیابی عمل انتخاب شده، به این صورت است که قدرت کل امپراطوری و فرمانروایی را با مرحله قبل مقایسه می‌کنیم. اگر نسبت به گام قبل بیشتر باشد، عمل انتخاب شده مثبت و در غیر اینصورت منفی ارزیابی می‌شود.

مدل پیشنهادی دوم ICA-LA-Radius

مدل پیشنهادی دوم ICA-LA-Radius است که ترکیبی از آتاماتای یادگیر و بهینه‌سازی کشورها می‌باشد. در این مدل از آتاماتای یادگیر جهت کنترل شعاع حرکت مستعمرات به سمت کشورهای استعمارگر استفاده می‌شود. هر آتاماتای یادگیر مانند الگوریتم قبل دارای دو عمل است. روند کلی الگوریتم نیز مانند قبل است.

عملی که آتاماتای یادگیر در هر گام بر می‌گزیند، تعیین‌کننده شیوه بروز کردن شعاع کلونی‌ها در آن گام می‌باشد. در صورت انتخاب عمل «ادامه مسیر در راستای قبل»، بروزسانی موقعیت کلونی‌ها به این صورت است که اگر شعاع حرکت از یک حد آستانه radius بیشتر باشد، برای مرحله قبل مقدار شعاع حرکت را از مقدار ثابت α کم می‌کنیم. در غیر این صورت به شعاع حرکت مقدار α را اضافه می‌کنیم. این بخش از الگوریتم مطابق فرمول‌های زیر انجام می‌شود. آستانه در نظر گرفته شده $\beta = 0.5$ است. درواقع جستجو با اطلاعات مکان کشورها در تکرار قبلی انجام می‌شود بنابراین این مرحله یک جستجوی محلی برای الگوریتم ایجاد می‌کند. در نتیجه توانایی exploit بالاتر می‌رود.

$$r_{iter} = 0.85(r_{iter-1} + \alpha) + 0.15(r_{iter-1} - \alpha) \quad (19)$$

در الگوریتم رقابت استعماری استاندارد پارامتر شعاع حرکت همه کلونی‌ها مقدار ثابت ۰.۵ می‌باشد.

$$x \sim U(0, \beta \times d) \quad (20)$$

در حالی که در الگوریتم پیشنهادی در هر گام که پاسخ دریافتی از محیط مناسب باشد با مقدار احتمالی بیشتر همان شعاع قبلی را ادامه می‌دهیم. در واقع با این عملکرد کلونی در اطراف کشور استعمارگر خود به جستجو

نتایج الگوریتم ICALA با الگوریتم یادگیر LRP در آتاماتاهای یادگیر دارای عملکرد بهتری برای نمونه مسائل SAT می‌باشد.

جدول ۱۱. مقایسه نتایج در نمونه‌های مختلف..

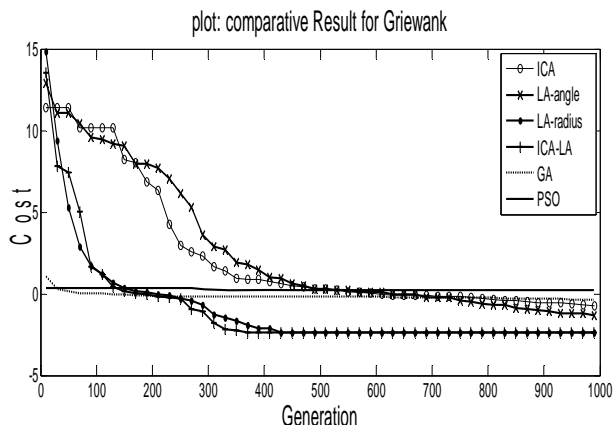
Formula		ICA		ICA_LA	
V	C	Time	Rate of satisfy	Time	Rate of satisfy
100	430	407.37	100%	500.75	100%
200	860	604.29	81%	607.44	95%
400	1700	709.67	66%	609.22	89%
600	2550	804.23	62%	703.77	80%
800	3400	809.73	50%	708.45	76%
1000	4250	903.72	49%	802.56	54%

همان‌طور که در جدول ۱۱ مشاهده می‌شود، الگوریتم پیشنهادی ICALA دارای نرخ موفقیت بیشتری در ۱۰۰ نمونه متفاوت با سایز یکسان، در مقابل الگوریتم ICA می‌باشد. از نظر زمان اجرا تفاوت چندین زیادی ندارند ولی برای نمونه‌های با سایز کوچک الگوریتم ICA دارای زمان اجرای کمتری می‌باشد در حالی که الگوریتم ICALA برای نمونه‌های بزرگ‌تر دارای تکامل و زمان اجرای مناسب می‌باشد. در ادامه این بخش برای تحلیل بیشتر الگوریتم‌های پیشنهادی و عملکردهای آن‌ها از توابع محک زیر نیز استفاده شده است.

جدول ۱۲. توابع محک مورد استفاده برای ارزیابی عملکرد الگوریتم‌ها.

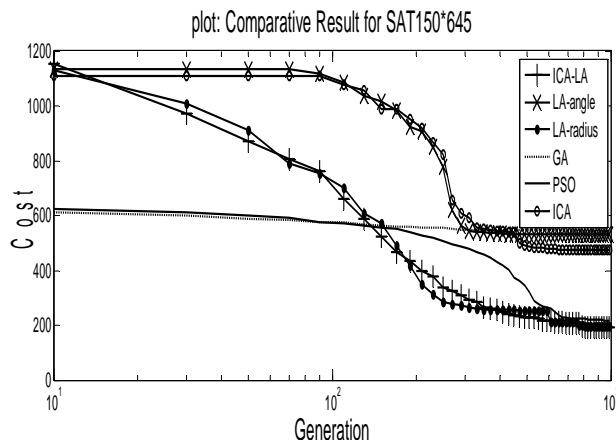
نام تابع	نمایش ریاضی تابع	محدوده	ابعاد	نوع تابع
Sphere	$f1(x) = \sum_{i=1}^D x_i^2$	(-100,100)	30	UM*
Rastrigrin	$f2(x) = \sum_{i=1}^D (x_i^2 - 10 \times \cos(2\pi x_i) + 10)$	(-10,10)	30	MM*
Rosenbrock	$f3(x) = \sum_{i=1}^{D-1} (100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	(-100,100)	30	UM
Griewank	$f4(x) = \frac{1}{4000} \times \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	(-600,600)	30	MM

در شکل‌های ۴، ۵، ۶ و ۷ توابع بالا در ۳۰ بعد اجرا شده‌اند، روند عملکرد الگوریتم‌ها به خوبی قابل مشاهده است. همان‌طور که دیده می‌شود ماهیت پویایی الگوریتم‌های پیشنهادی، سرعت یافتن بهینه محلی را نسبت به سایر الگوریتم‌ها افزایش داده است. الگوریتم پیشنهادی ICALA بهترین الگوریتم برای فرار از مینیمم محلی است. الگوریتم‌های ICA_LA_Angle و ICA_LA_Radius نیز برای نمونه‌های خاصی خوب عمل می‌کنند ولی برای همه نمونه‌ها دارای عملکرد مناسبی نمی‌باشند.



شکل ۴. میانگین بهترین تابع شایستگی در ۱۰ اجرا.

همان‌طور که در شکل ۲، مشاهده می‌شود، الگوریتم‌های مورد نظر برای مقایسه را در ۱۰۰۰ تکرار مورد بررسی قرار داده‌ایم. نتایج نشان داده شده، حاصل از میانگین ۲۰ بار اجرای نمونه‌های مختلف می‌باشد. واضح است که الگوریتم پیشنهادی سوم یعنی ICALA دارای توانایی بالاتری برای جستجوی عمومی می‌باشد. الگوریتم ICA_LA_Angle در مقایسه با الگوریتم ICA تفاوت زیادی در توانایی جستجو ندارد. همچنین مشاهده می‌شود که الگوریتم ICA_LA از الگوریتم ICA_LA_Radius نیز بهتر عمل می‌کند. علت این امر نیز بخاطر پویایی بالاتر الگوریتم و تنظیم همزمان شعاع و زاویه حرکت در کلونی‌ها می‌باشد. نتایج در شکل ۳، بر روی نمونه با سایز کوچک‌تر می‌باشد. بنابراین فضای مسأله دارای مینیمم محلی کمتری در طول فرایند جستجو می‌باشد. در این نمونه نیز الگوریتم ICALA دارای توانایی جستجوی بالاتری می‌باشد و به بهینه عمومی نزدیک‌تر می‌شود. در نمونه‌های با اندازه کوچک‌تر الگوریتم ICA_LA_Radius بهتر از الگوریتم ICA_LA_Angle عمل می‌کند.



شکل ۳. میانگین بهترین تابع شایستگی در ۲۰ بار اجرا

جدول ۹. مقایسه نتایج در نمونه‌های مختلف و آتاماتاهای متفاوت در الگوریتم ICALA.

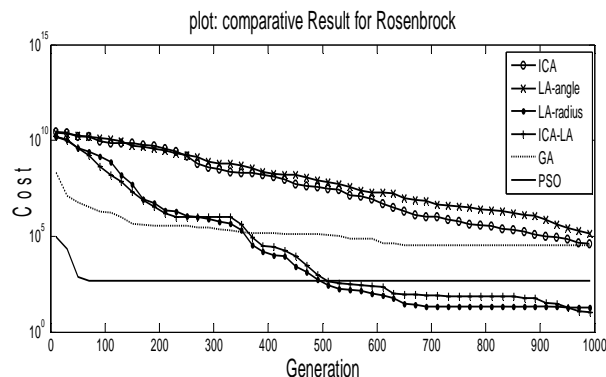
		BestCost		BestCost	
SAT125*538	LIP	145.0926	flat100-239	LIP	600.9345
	LRεP	230.5671		LRεP	633.6633
	LRI	155.2949		LRI	664.2557
	LRP	66.4430		LRP	530.7654
SAT150*645	LIP	310.4966	flat125-301	LIP	604.3134
	LRεP	394.4540		LRεP	599.6785
	LRI	343.3092		LRI	642.2393
	LRP	195.6315		LRP	665.3676
SAT200*860	LIP	390.5325	flat150-360	LIP	654.5325
	LRεP	402.7041		LRεP	702.5459
	LRI	416.9380		LRI	600.1020
	LRP	370.6733		LRP	719.6564
SAT225*960	LIP	428.0629	flat200-479	LIP	772.6729
	LRεP	383.0548		LRεP	770.0556
	LRI	480.1577		LRI	721.9812
	LRP	495.1522		LRP	718.4322

این نتایج از الگوریتم ICALA با میانگین ۲۰ بار اجرا برای هر نوع الگوریتم یادگیر می‌باشد. همان‌طور که در نتایج مشخص است، الگوریتم ICALA برای نمونه‌های متفاوت دارای نتایج متفاوت می‌باشد. الگوریتم ICALA با الگوریتم یادگیر LRP در بیشتر نمونه‌های SAT دارای عملکرد بهتری می‌باشد. بنابراین

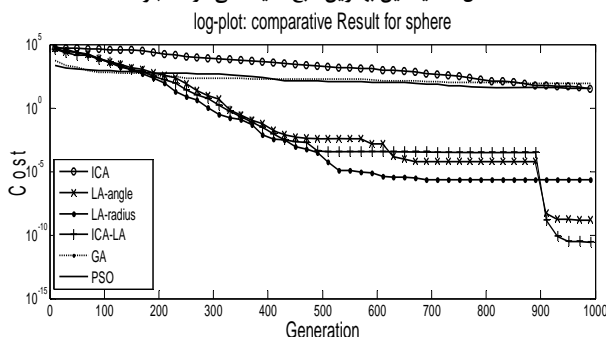
استفاده از آتاماتای یادگیر دارای دو مزیت عمده می‌باشد: اولاً می‌توان از دانش موجود در تعیین زاویه و شعاع حرکت کلونی‌ها استفاده نمود و ثانیاً این روند با گرفتن بازخورد از اجرای الگوریتم اصلاح می‌گردد. بنابر نتایج حاصل شده، الگوریتم پیشنهادی ICALA توانایی explore و exploit مناسبی دارد.

مراجع

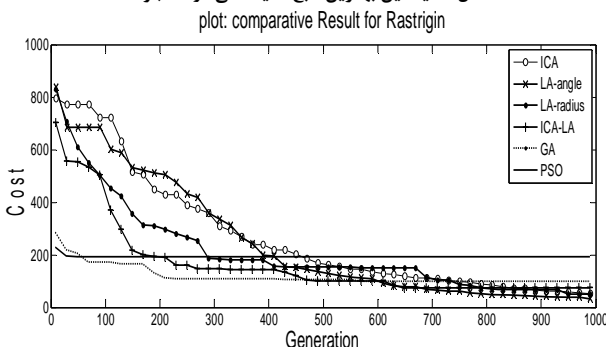
- [1] M. Melanie, "An Introduction to Genetic Algorithms, Massachusetts: MIT Press, 1999.
- [2] J. Kennedy, R.C. Eberhart, "Particle Swarm Optimization," Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp.1942-1948, 1995.
- [3] S. Kirkpatrick and C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," Science, Vol 220, Number 4598, pp. 671-680, 1983.
- [4] B. Franklin and M. Bergerman, "Cultural Algorithms: Concepts and Experiments", In Proceedings of the IEEE Congress on Evolutionary Computation, volume 2, pp. 1245-1251, 2000.
- [5] E. Atashpaz-Gargari and C. Lucas, "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition," IEEE Congress on Evolutionary Computation (CEC 2007). pp 4661-4667, 2007.
- [6] H. Bahrami, K. Faez and M. Abdechiri, "Imperialist Competitive Algorithm using Chaos Theory for Optimization", UKSim-AMSS 12th International Conference on Computer Modeling and Simulation Cambridge, UK, pp. 98-103, 2010, ISBN: 978-0-7695-4016-0.
- [7] M. Abdechiri, K. Faez and H. Bahrami, "Neural Network Learning based on Chaotic Imperialist Competitive Algorithm", The 2nd International Workshop on Intelligent System and Applications (ISA2010), 2010.
- [8] R. Rajabioun, F. Hashemzadeh, E. Atashpaz-Gargari, B. Mesgari and F. Rajaei Salmasi, "Identification of a MIMO Evaporator and Its Decentralized PID Controller Tuning Using Colonial Competitive Algorithm", Accepted to be presented in IFAC World Congress, 2008.
- [9] K. S. Narendra and K. S. Thathachar, "Learning Automata: An Introduction", New York, PrenticeHall, 1989.
- [10] M. A. L. Thathachar and B. R. Harita, "Learning Automata with Changing Number of Actions", IEEE Transactions on Systems, Man, and Cybernetics, 17: pp. 1095-1100, 1987.
- [11] H. Beigy and M. R. Meybodi, "Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problems", International Journal of Uncertainty, Fuzziness and KnowledgeBased Systems, 14: pp. 591-615, 2006.
- [12] M. R. Meybodi and H. Beigy, "A Note on Learning Automata Based Schemes for Adaptation of BP Parameters", Journal of Neurocomputing, 48(4): pp. 957-974, October 2002.
- [13] H. Beigy and M. R. Meybodi, "A Learning Automata Based Algorithm for Determination of Minimum Number of Hidden Units for Three Layers Neural Networks", Journal of Amirkabir, 12(46): pp. 111-136, 2001.
- [14] M. Sheybani, and M.R. Meybodi, "PSO-LA: A New Model for Optimization", Proceedings of 12th Annual CSI Computer Conference of Iran, Shahid Beheshti University, Tehran, Iran, pp. 1162-1169, 2007.
- [15] J. Gu, "The UniSAT Problem Models (appendix)", IEEE Trans, on Pattern Analysis and Machine Intelligence, 14(8):865, 1992.
- [16] J. Gu, "Global Optimization for satisfiability (SAT) problems", IEEE Trans, on Knowledge and Data Engineering, 6(3):361-381, jun, 1994, and 7(1):192, 1995.
- [17] M. Bertran, S. Lakhdar and G. Eric, "Tabu search for SAT", In Proc. of the 14th National Conference on Artificial Intelligence and 9th Innovative application of Artificial Intelligence Conference (AAAI-97/IAAI-97), pp. 281-285, 1997.
- [18] W. M. Spears, "Simulated annealing for hard satisfiability problems", In second DIMACS implementation challenge: cliques, coloring and dsatisfiability, volume 26 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pp.533-558, 1996.
- [19] A. Kenneth, P. Andre and W. M. Spears, "Using genetic algorithm to solve NP-complete problems", In Proc. of the Third Int. Conf. on Genetic Algorithms, pp. 124-132, San Mateo, CA, 1989.
- [20] O. C. Granmo and N. Bouhmala, "Solving the Satisfiability Problem Using Finite Learning Automata", IJCSA 4(3): 15-29, 2007.



شکل ۵. میانگین بهترین تابع شایستگی در ۱۰ اجرا.



شکل ۶. میانگین بهترین تابع شایستگی در ۱۰ اجرا



شکل ۷. میانگین بهترین تابع شایستگی در ۱۰ اجرا

جدول ۱۳. مقایسه نتایج در نمونه‌های مختلف و آتاماتاهای متفاوت.

	PSO	GA	ICA	Angle	Radius	ICA_LA
F1	25.54	67.99	27.57	1.50×10^{-9}	1.91×10^{-6}	2.34×10^{-11}
F2	518.7	3.52×10^4	3.26×10^4	1.11×10^5	20.0514	10.94
F3	189.7	95.24	49.78	21.80	49.75	48.63
F4	0.234	-0.38	-0.79	-1.33	-2.37	-2.37

همانطور که در جدول ۱۳ مشاهده می‌شود الگوریتم پیشنهادی ICALA راه‌حل‌های بهتری را در توابع Rosenbrock, Sphere, Griewank, Ackley و Rastrigin در ۳۰ بعد بدست آورد.

نتیجه‌گیری

در الگوریتم پیشنهادی ICALA، از آتاماتای یادگیر برای تنظیم شعاع و زاویه حرکت کشورها به سمت امپراطور بصورت پویا استفاده شد. الگوریتم پیشنهادی عملکرد کارآمدتر و همگرایی مناسب‌تری نسبت به الگوریتم ICA دارا می‌باشد. وظیفه آتاماتا برقراری تعادل بین جستجوی سراسری و جستجوی محلی است.