

بهبود بهینه سازی گروه ذرات با استفاده از منطق فازی

محمد رضا میبدی

دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران
mmeybodi@aut.ac.ir

محمد حسین نوروزی بیرامی

دانشکده مهندسی برق، رایانه و فناوری اطلاعات
دانشگاه آزاد اسلامی
اسکو ایران
mh.noroozi@gmail.com

چکیده: بهینه سازی گروه ذرات یک تکنیک بهینه سازی مبتنی بر جمعیت می باشد که بر اساس قوانین احتمال کار می کند. در این روش هر یک از ذرات سعی می کنند به سمتی حرکت کنند که بهترین تجربه های فردی و گروهی در آن نقاط روی داده است. دو مشکل اصلی الگوریتم PSO استاندارد، افتادن در دام بهینگی محلی و پایین بودن سرعت همگرایی آن می باشد. یکی از روش های حل مشکلات عنوان شده استفاده از روش هایی ترکیبی مانند منطق فازی و یا الگوریتم های ژنتیک می باشد. در این مقاله دو الگوریتم ترکیبی با استفاده از منطق فازی ارائه می گردد. الگوریتم های پیشنهادی سعی می کند مشکلات عنوان شده در PSO استاندارد را حل نماید. به منظور ارزیابی الگوریتم های پیشنهادی از توابع محک استاندارد استفاده شده است. نتایج آزمایشها نشان از برتری الگوریتم های پیشنهادی دارد.

کلمات کلیدی: بهینه سازی گروه ذرات، بهینگی محلی، منطق فازی

Improving Particle Swarm Optimization using Fuzzy Logic

M. H. Noroozi Beyrami

Electrical and Computer Engineering and Information
Technology Department
Islamic Azad University
Osku, Iran
mh.noroozi@gmail.com

M. R. Meybodi

Computer Engineering and Information Technology
Department
Amirkabir University of Technology
Tehran Iran
mmeybodi@aut.ac.ir

Abstract: Particle Swarm Optimization is a population based optimization technique that based on probability rules. In this technique each particle moves toward their best individual and group experience had occurred. Fundamental problems of a standard PSO algorithm are fall into local optimum trap and the low speed of the convergence. One of the methods to solve these problems is to combine PSO algorithm with other methods such as fuzzy logic and genetic algorithms. In this paper two PSO algorithm based on fuzzy logic are proposed. The proposed algorithms try to solve the above mentioned problems. For evaluation purpose, the proposed algorithms are tested on number of standard optimization functions. The results of experimentations have shown the superiority of the proposed algorithm over standard PSO.

Keywords: Particle Swarm Optimization, Local Optimum, Fuzzy Logic

۱. مقدمه

در بسیاری از مسائل بهینه سازی بویژه مسائل بزرگ، انتخاب بهترین جواب از طریق جستجوی همه جانبه اگر چه غیر ممکن نیست، ولی کاری بسیار مشکل است. هدف مسأله بهینه سازی کاهش زمان این جستجوی می باشد. روش های ابتکاری^۱ راه حل های خوبی برای یافتن جواب بهینه می باشد، ولی تضمینی برای یافتن جواب بهینه نمی دهند. اما امروزه با بزرگ و پیچیده شدن مسائل، استقبال از روش های ابتکاری بطور چشم گیری

^۱ Heuristic

افزایش یافته است [2].

بهینه سازی گروه ذرات^۱ یکی از تکنیک های بهینه سازی ابتکاری می باشد که بر مبنای جمعیت کار می کند. ایده اصلی این روش ابتدا در سال ۱۹۹۵ توسط دکتر کندی و دکتر ابرهارت [4] مطرح گردید که از رفتار دسته جمعی ماهی ها و پرندگان برای یافتن غذا الهام می گیرد. گروهی از پرندگان و ماهی ها در یک فضای تصادفی دنبال غذا می گردند و تنها یک تکه غذا وجود دارد و هیچ یک از پرندگان از محل غذا اطلاعی ندارد و فقط فاصله خود تا غذا را می داند. یکی از بهترین استراتژی ها دنبال کردن پرنده ای می باشد که به غذا نزدیک تر است. این تئوری، استراتژی اصلی الگوریتم PSO می باشد.

در PSO هر پرنده یک جواب ممکن در فضای مسئله می باشد که ذره^۲ نامیده می شود. هر ذره دارای یک مقدار شایستگی می باشد که توسط تابع شایستگی مسئله محاسبه می شود. ذره ای که به جواب نزدیک تر باشد، شایستگی بیشتری دارد. این الگوریتم ماهیت پیوسته ای دارد و در کاربرد های متعددی کارایی خود را اثبات کرده است [1]. در این مقاله دو الگوریتم ترکیبی فازی با PSO به نام های FPSO1 و FPSO2 ارائه شده است که اولاً مشکلات الگوریتم استاندارد را تا حد زیادی حل می کند و از طرفی باعث افزایش سرعت همگرایی آن می شود. روشهای مشابهی نیز وجود دارد، TPSO [7] یک روش ترکیبی فازی می باشد، در صورتی که سرعت ذره از حد آستانه ای کمتر باشد و جواب بدست آمده شایستگی مطلوبی نداشته باشد، شکی به ذرات وارد می کند تا سرعت آنها تغییر نماید. WPSO [8] نیز یک تابع فازی برای بدست آوردن پارامتر وزن میانی الگوریتم PSO استاندارد ارائه می کند. در نهایت الگوریتم های پیشنهادی را با روش های موجود و همچنین الگوریتم PSO استاندارد مورد مقایسه قرار می دهیم. نتایجی که الگوریتم های پیشنهادی ارائه می کنند، نسبت به روش های موجود بهینه بوده و همگرایی آنها نیز سریعتر از الگوریتم های موجود می باشد.

ادامه این مقاله بدین صورت سازماندهی شده است. در بخش دوم این مقاله الگوریتم PSO استاندارد ارائه می شود. در بخش سوم پیش زمینه ای از منطق فازی و کاربرد های آن در زمینه های مختلف ارائه می گردد. در بخش چهارم الگوریتم های ترکیبی فازی با PSO ارائه می شود. در بخش پنجم الگوریتم های پیشنهادی بررسی گردیده و در بخش ششم نتایج ارزیابی های مختلف بر روی الگوریتم های پیشنهادی ارائه می شود و در بخش هفتم نیز به جمع بندی موضوع و کار های آتی می پردازیم.

۲. الگوریتم PSO استاندارد

PSO از جمله الگوریتم های جستجوی موازی مبتنی بر جمعیت است که با یک گروه از جواب های تصادفی (ذره ها) شروع به کار می کند، سپس برای یافتن جواب بهینه در فضای مسئله با به روز کردن مکان ذره ها به جستجو ادامه می دهد. هر ذره به صورت چند بعدی (بسته به نوع مسئله) با دو بردار V_{id} و X_{id} که به ترتیب معرف موقعیت مکانی و سرعت بعد d ام از i امین ذره هستند، مشخص می شود. در هر مرحله از حرکت جمعیت، مکان هر ذره با دو مقدار بهترین به روز می شود. اولین مقدار، بهترین تجربه ای است که خود ذره تا به حال بدست آورده است و با p_best نشان داده می شود. دومین مقدار، بهترین تجربه ای است که در بین تمامی ذره ها بدست آمده است و با g_best نشان داده می شود. در برخی ویرایش های PSO، ذره قسمتهایی از جمعیت را که همسایگان توپولوژیکی اش هستند، انتخاب می کند و تنها آنها را در اعمال خود دخیل می کند. در این صورت بهترین راه حلی محلی استفاده می شود و از l_best به جای g_best استفاده می شود و مدل های مختلفی برای بدست آوردن همسایگان توپولوژیکی هر ذره ارائه شده است [20]. در هر تکرار، الگوریتم بعد از یافتن دو مقدار بالا، سرعت و موقعیت جدید ذره را بر اساس معادلات (۱) و (۲) بروز رسانی می کند.

$$v_{id}(t+1) = w.v_{id}(t) + c_1.rand_1(p_best_{id}(t) - x_{id}(t)) + c_2.rand_2(g_best_{id}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

در معادله (۱)، w ضریب اینرسی است که به صورت خطی کاهش می یابد که معمولاً در بازه $[0-1]$ می باشد. c_1 و c_2 ضرایب یادگیری یا شتاب می باشد که در بازه $[0-2]$ انتخاب می شود که در بیشتر موارد برای هر دوی آنها از مقدار ۱،۴۹ و یا ۲ استفاده شده است [2,6,7]. دو عدد $rand_1$ و $rand_2$ نیز اعداد تصادفی در بازه $[0-1]$ می باشد. همچنین مقدار نهایی سرعت هر ذره برای جلوگیری از واگرایی الگوریتم به یک بازه محدود می شود. $v_{id} \in [-v_{max}, v_{max}]$. شرط خاتمه الگوریتم همگرایی تا حد معین و یا توقف بعد از تعداد معینی تکرار می باشد. معادله (۲) نیز بردار

¹ Particle Swarm Optimization - PSO

² Particle

موقعیت فعلی ذره را با توجه به سرعت جدید آن بروز رسانی می کند.

سمت راست معادله (۱) از سه قسمت تشکیل شده است، قسمت اول ضریبی از سرعت فعلی ذره می باشد، قسمت دوم برای تغییر سرعت و چرخش ذره به سمت بهترین تجربه شخصی می باشد و قسمت سوم نیز باعث تغییر سرعت و چرخش ذره به طرف بهترین تجربه گروهی می باشد [3]. در واقع حرکت بهینه سازی گروه ذرات بدون قسمت اول معادله (۱)، فرایندی خواهد بود که طی آن فضای جستجو به تدریج کوچک می شود و جستجوی محلی حول بهترین ذره شکل می گیرد، اما در مقابل قسمت اول معادله (۱) باعث حرکت ذرات در مسیر عادی خود خواهد بود تا به دیواره محدوده جستجو برسد و به نوعی جستجوی سراسری انجام می دهد. با ترکیب این دو عامل سعی شده است تا موازنه ای بین جستجوی محلی و سراسری برقرار گردد. w برای برقراری بهتر این موازنه برای اولین بار در [5] پیشنهاد گردید که ضریب حرکت در جستجوی سراسری را مشخص می کند. دو پارامتر C_1 و C_2 نیز ضریب حرکت در جستجوی محلی را مشخص می کند. با توجه به اینکه w رفته رفته کاهش می یابد ولی C_1 و C_2 همواره ثابت است، بنابراین با ادامه جستجو از میزان جستجوی سراسری کم شده و جستجوی محلی افزایش می یابد تا بهینه ترین جواب بدست آید. هر چه شیب کاهشی w کمتر باشد امکان پیدا کردن جواب بهینه سراسری افزایش پیدا می کند.

یکی از مشکلات شناخته شده الگوریتم PSO این است که برای ذره ای که شامل اطلاعات g_best است، قسمت های دوم و سوم معادله (۱)، صفر خواهد شد. بنابراین ذره در راستای بردار حرکت قبلی خود حرکت خواهد کرد. از طرفی چون به طور معمول w کوچکتر از یک می باشد، این قسمت نیز میرا خواهد بود. بنابراین ذره شامل g_best ثابت خواهد شد و سایر ذرات نیز به این ذره همگرا می شوند [2]. در بسیار از این موارد این موضوع باعث همگرایی زودرس الگوریتم به یک بهینه محلی خواهد شد. برای رفع این مشکل ابتدا در سال ۲۰۰۲، الگوریتم جدیدی با نام GCP SO^۱ ارائه گردید [11] که در این روش یک پارامتر جدیدی به الگوریتم اضافه شده بود که باعث جستجوهای تصادفی ذره g_best حول جواب بهینه خواهد شد. دومین مشکلی که الگوریتم PSO استاندارد دارد، همگرایی خطی آن است که ممکن است بسیار زمان بر باشد. ما در این مقاله دو روش ترکیبی برای حل این مشکلات ارائه می کنیم.

۳. پیش زمینه ای در مورد سیستم های فازی

منطق فازی تکنولوژی نسبتاً جدیدی است که در مقابل روشهای مرسوم برای طراحی و مدلسازی سیستمی که نیازمند ریاضیات و احتمالات پیشرفته و نسبتاً پیچیده می باشد، به کار می رود. از مقادیر و قوانین مبتنی بر متغیرهای زبانی و یا به عبارتی از دانش فرد خبره با هدف ساده، دقیق و کارآمد کردن طراحی سیستم استفاده می کنند. سیستم های کنترل مبتنی بر منطق فازی به دلیل داشتن شفافیت بالا برای تصمیم گیری و همچنین ارائه نتایج و مفاهیم قابل فهم، توسط افراد در سطوح مختلف به کار گرفته می شود. منطق فازی یک تعمیم از منطق بولی می باشد، از آنجایی که در منطق کلاسیک همه چیزها در عبارت باینری بیان شود (۰ یا ۱، سفید یا سیاه، بله یا خیر) منطق فازی گزاره های صحیح بولی را با درجه درستی بیان می کند. تابع عضویت یک مجموعه فازی، مشابه تابع نماینده مجموعه های کلاسیک است. در حالت خاص یک منحنی می باشد که نشان دهنده چگونگی نگاشت شدن هر یک از نقاط فضا به درجه عضویت می باشد، که این درجه عضویت عددی بین ۰ و ۱ خواهد بود. شکل بسیار رایج تر برای توابع عضویت، توابع مثلثی می باشد [13,14].

مرحله پردازش که به آن موتور استنباط نیز گفته می شود بر اساس مجموعه ای از قوانین فازی بصورت عبارات IF-THEN عمل می کند. هر سیستم فازی تعدادی قانون دارند که در پایگاه دانش آن ذخیره می گردد. به عنوان مثال: اگر تکرار پایین باشد اولویت جابجایی خیلی بالا است که عبارات اولویت و تکرار متغیرهای زبانی می باشند و پایین و خیلی بالا عبارات زبانی می باشند. هر عبارت زبانی مطابق یک تابع عضویت می باشد.

دو رویه عمده برای استنباط وجود دارد. نوع اول مدل استنباط فازی ممدانی می باشد که توسط آقای ابراهیم ممدانی در ۱۹۷۵ ارائه شد و نوع دوم استنباط فازی، روش تاکاگی سوگنو می باشد که در ۱۹۸۵ ارائه شده است. این دو روش در بسیاری از جنبه ها یکسان می باشند، مانند فازی سازی ورودیها و عملگرهای فازی، اما تفاوت اصلی بین روش سوگنو و ممدانی این است که خروجی روش سوگنو عضو توابعی است که می تواند خطی و یا ثابت باشد، ولی در استنباط ممدانی انتظار داریم که خروجی، توابع عضویت مجموعه های فازی باشند [12]. ما در طراحی سیستم مورد نظر، بر حسب نتایج مورد انتظار از روش آقای ممدانی استفاده کرده ایم.

¹Guaranteed Convergence PSO - GCP SO

۴. روش ترکیبی فازی با PSO

با توجه به مطالب ارائه شده در بخش قبل و آشنایی با مشکلات PSO، می توان گفت که از جمله مشکلات مهم روش استاندارد یکی افتادن در دام بهینگی محلی و دیگری پایین بودن سرعت همگرایی آن است. برای حل مشکل اول باید راهکاری اندیشید تا بهینگی محلی تشخیص داده شده و از آن اجتناب شود [16]. روش های مختلفی برای این کار پیشنهاد شده است. در [7] برای اجتناب از افتادن در دام بهینگی محلی در صورتی که سرعت ذره از یک حد مشخصی کمتر شود ولی میزان شایستگی جواب بدست آمده قابل قبول نباشد از یک تابعی برای ایجاد یک شک در سرعت ذره استفاده می شود. در [8,19] از یک تابع غیر خطی فازی برای تغییر ضریب اینرسی استفاده می شود، در صورتی که میزان شایستگی بدست آمده قابل قبول نباشد و ضریب اینرسی نیز کاهش یافته باشد، موجب افزایش این ضریب شده و با افزایش سرعت ذره امکان جستجوی سراسری را بیشتر می کند. در [9,17] با شناسایی دو قله محلی نزدیک به هم تا حد امکان از افتادن در دام بهینگی محلی حفظ می کند. همه این روش های ارائه شده به نوعی مانع از افتادن در دام بهینگی محلی خواهد شد. در این مقاله ما دو روش جدید ارائه می کنیم که در واقع یک تابع غیر خطی برای ضریب اینرسی w و تابع دیگری برای ضرایب شتاب $c1$ و $c2$ می باشند. در ادامه دو روش ترکیبی موجود را به صورت مختصر توضیح می دهیم.

۴.۱. ایجاد آشفتنگی فازی تطبیقی در PSO

در [7] برای اجتناب از افتادن در دام بهینگی محلی، روش جدیدی TPSTO^۱ ارائه گردیده است. در صورتی که سرعت ذره از حد آستانه v_c کمتر باشد، سرعت جدیدی با استفاده از معادله (۳) به ذره انتساب می شود. سرعت فعلی ذره نیز به کمک رابطه (۴) محاسبه می شود.

$$v_{id}(t+1) = w \cdot \hat{v} + c_1 \cdot rand_1(pbest_{id}(t) - x_{id}(t)) + c_2 \cdot rand_2(gbest_{id}(t) - x_{id}(t)) \quad (3)$$

$$\hat{v} = \begin{cases} v_{id} & \text{if } |v_{id}| \geq v_c \\ u(-1,1)v_{\max} / \rho & \text{if } |v_{id}| < v_c \end{cases} \quad (4)$$

در رابطه (۴) $u(-1,1)$ عددی تصادفی با توزیع یکنواخت در بازه $[-1,1]$ می باشد و ρ پارامتر سنجش^۲ دامنه نوسان ذره نسبت به v_{\max} می باشد. تغییر موقعیت هر ذره مستقیماً به دو پارامتر v_c و ρ بستگی دارد. اگر مقدار v_c کوچک باشد، در این صورت نوسان ذره کمتر شده و احتمال افتادن ذره در دام بهینگی محلی بیشتر خواهد شد. اما بزرگ بودن v_c به ذره این امکان را می دهد که به راحتی از موقعیتی به موقعیت دیگر پرش کند. بنابراین مقدار بزرگ v_c امکان جستجوی سراسری و مقدار کوچکتر آن امکان جستجوی محلی را افزایش خواهد داد. ρ نیز مستقیماً دامنه نوسان ذره را تغییر می دهد. مقدار بزرگ ρ باعث کاهش سرعت ذره می شود و به ذره اجازه می دهد که در بهینه های محلی بزرگی که در فضای جستجو وجود دارد و امکان پیدا کردن جواب بیشتر است، پرش انجام ندهد. اما در صورتی که مقدار ρ کوچک باشد، در این صورت مسیر حرکت ذره میزان قابل توجهی حرص خواهد شد. در واقع این پارامتر با کم یا زیاد شدن، میزان جستجوی محلی و سراسری را تنظیم می کند. در این الگوریتم دو متغیر CBPE (بهترین ارزیابی کارایی فعلی^۳) و سرعت فعلی^۴ ذره را به عنوان ورودی سیستم فازی در نظر می گیریم. CBPE میزان شایستگی ذره در حالت فعلی را نشان می دهد و $CBPE_{\min}$ بهترین شایستگی می باشد که ذرات تا به حال بدست آورده است و $CBPE_{\max}$ بدترین شایستگی هست که ذرات تا به حال بدست آورده است. در معادله (۵)، NCBPE مقدار نرمال شده سه مقدار بهترین، بدترین و حالت فعلی مقدار شایستگی تابع می باشد که مقداری در بازه $[0-1]$ به خود می گیرد.

$$NCBPE = \frac{CBPE - CBPE_{\min}}{CBPE_{\max} - CBPE_{\min}} \quad (5)$$

در این سیستم دو پارامتر خروجی نیز داریم که اولی ρ ، پارامتر سنجش کنترل کننده میزان نوسان ذره و V_{ck} که میزان سرعت حد آستانه را به کمک رابطه (۶) بروز رسانی می کند.

¹ Turbulent Particle Swarm Optimization - TPSTO

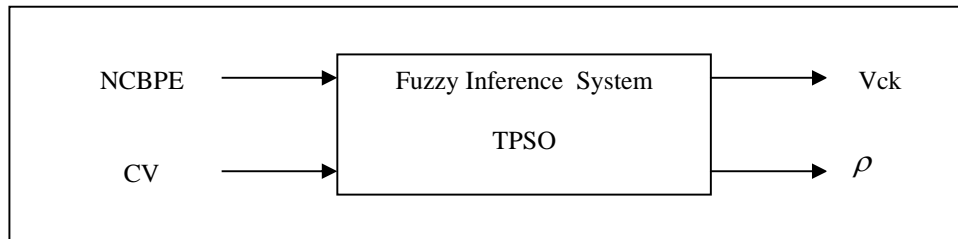
² scaling factor

³ Current Best Performance Evaluation - CBPE

⁴ Current Velocity - CV

$$v_c = e - [10(1 + Vck)] \quad (6)$$

سیستم استنباط فازی مورد نظر به روش ممدانی پیاده سازی گردیده است، این سیستم دو ورودی و دو خروجی دارد که در شکل (۱) نشان داده شده است.

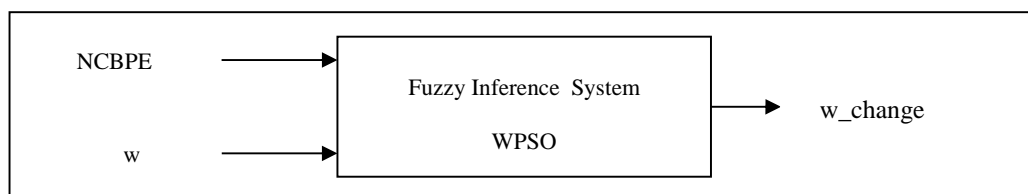


شکل (۱) سیستم استنباط فازی TPSO

۲.۴. ارائه یک تابع غیر خطی برای وزن میانی w

با توجه به بخش های قبلی، یکی از مشکلاتی که می توان برای PSO استاندارد عنوان کرد، همگرایی خطی الگوریتم بود. این کار باعث می شود که سرعت همگرایی به مراتب کاهش پیدا کند. برای حل این مشکل می توان از توابع غیر خطی برای پارامتر وزن میانی استفاده کرد. چون در این سیستم ها چندین پارامتر برای تصمیم گیری وجود دارد، بنابراین توابع فازی می تواند از جایگاه ویژه ای برخوردار باشد. این نوع توابع برای اولین بار در [8] ارائه گردیده است.

در این روش که WPSO می نامیم، اساس تصمیم گیری با روش قبل متفاوت است. با توجه به تعریف پارامتر NCBPE، می توان گفت که میزان همگرایی با استفاده از این پارامتر مشخص می گردد و این پارامتر در صورتی که مقداری نزدیک به صفر داشته باشد می توان گفت که نتیجه همگرایی مطلوب است. از طرفی چون w به صورت خطی عمل می کند، بنابراین تا زمانی که w به اندازه گفتمانی کاهش نیافته باشد میزان همگرایی مطلوب نخواهد شد. بنابراین در این الگوریتم اساس تصمیم گیری بر اساس دو پارامتر w و NCBPE می باشد. شکل (۲) نمایی از این سیستم را نشان می دهد. در این روش با توجه به پارامتر w و اندازه NCBPE، مقدار جدیدی برای w تعیین می گردد.

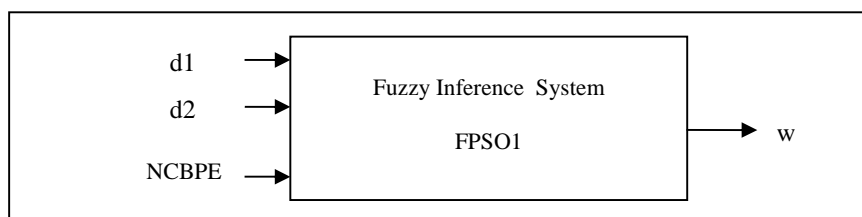


شکل (۲) سیستم استنباط فازی WPSO

۵. الگوریتم های پیشنهادی

۵.۱. روش ترکیبی FPSO1^۱

اولین الگوریتم پیشنهادی را FPSO1 می نامیم که ضریب اینرسی w برای این الگوریتم یک تابع فازی است که سه پارامتر $d1$ ، $d2$ و NCBPE به عنوان ورودی این تابع و w خروجی آن می باشد که در شکل (۳) نشان داده شده است.



^۱ Fuzzy PSO

شکل (۳) تابع فازی مربوط به FPSO1

$$d1=|p_best-x| \quad d2=|g_best-x| \quad (۷)$$

در روابط (۷) دو پارامتر $d1$ و $d2$ معرف میزان نزدیکی ذره به بهترین تجربه محلی و سراسری را نشان می دهد که می تواند میزان نزدیکی به بهینه محلی و سراسری را بازگو نماید و NCBPE که در بخش ۴-۱ توضیح داده شد. برای معرفی این سه پارامتر به سیستم از برچسب زبانی کم، متوسط و زیاد استفاده کرده ایم که محدوده $d1$ و $d2$ بر حسب اندازه فضای جستجو مشخص می شود. بنابراین این سه پارامتر مبنای تصمیم گیری سیستم فازی است تا مقدار w را تعیین نماید که در بازه $[0 \ 1]$ قابل انتخاب می باشد. اما نکته مهم، انتخاب قوانین فازی می باشد که تاثیر مستقیم بر روی نتایج بدست آمده دارد. جدول (۱) تعدادی از قوانین استفاده شده در این سیستم را نشان می دهد.

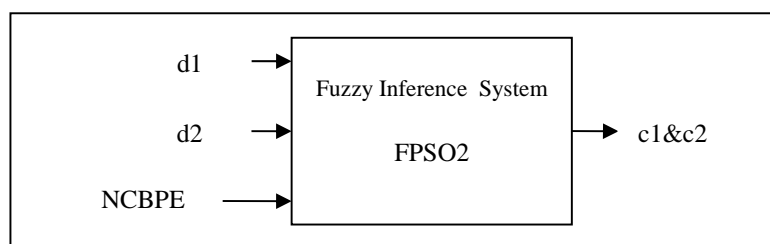
جدول (۱) تعدادی از قوانین فازی الگوریتم FPSO1

Rules	Input			Output
	d1	d2	NCBPE	w
1	High	High	High	High
2	Low	Low	Low	Low
3	Low	Low	Not Low	High
4	Low	Low	Medium	Medium

قوانین متعددی می توان برای این تابع در نظر گرفت ولی تجربه نشان داده است که زیاد بودن قوانین تاثیر زیادی بر روی نتیجه بدست آمده ندارد، بلکه نقطه مهم کیفیت قوانین انتخابی می باشد که نزدیک ترین جواب را تولید نماید. می توان قوانین جدول (۱) را به صورت زیر بیان کرد. قانون یک: اگر $d1$ و $d2$ کم باشد به این معنی است که ذره به بهترین تجربه محلی و سراسری نزدیک می باشد و از طرف دیگر میزان شایستگی آن نیز در حد قابل قبولی است، که در این صورت مقدار w نیز کم انتخاب می شود تا ذره حول این موقعیت به جستجو بپردازد. قانون دوم: در صورتی که $d1$ و $d2$ کم باشد ولی NCBPE کم نباشد (یعنی می تواند متوسط یا زیاد باشد)، بدین معنی است که ذره به بهترین تجربه محلی و سراسری نزدیک است ولی میزان شایستگی آن در حد قابل قبولی نمی باشد (این نکته دقیقا همان افتادن در دام بهینگی محلی می باشد) که در این صورت مقدار w زیاد انتخاب می شود که سرعت ذره بیشتر شده و موقعیت آن تغییر کند. قانون سوم: اگر $d1$ کم باشد و $d2$ کم نباشد و مقدار NCBPE متوسط باشد به این معنی است که ذره به بهترین تجربه محلی نزدیک است ولی فاصله مناسبی با بهینه سراسری ندارد، در این صورت ذره در شرایط مناسبی قرار ندارد و مقدار w متوسط انتخاب می شود. قانون چهارم: در این قانون به دلیل کم نبودن $d1$ و $d2$ ، باید سرعت آن زیاد شود، بنابراین w بزرگتر انتخاب می شود.

۵.۲. روش ترکیبی FPSO2

الگوریتم پیشنهادی دوم را FPSO2 می نامیم که در این تابع C را به عنوان ضریب شتاب، یک تابع غیر خطی فازی در نظر می گیریم، که در شکل (۴) نشان داده شده است. در این تابع نیز $d1$ و $d2$ اختلاف بین موقعیت فعلی ذره را به ترتیب با بهترین تجربه شخصی و بهترین تجربه گروهی نشان می دهد و NCBPE مقدار شایستگی فعلی نرمال شده ذره می باشد و C نیز در بازه $[1 \ 2]$ تعریف شده است. با توجه به توضیحاتی که در بخش های قبلی ارائه شد، چون C ضریب شتاب برای افزایش جستجوی محلی حول بهترین تجربه گروهی و محلی می باشد. بنابراین کاهش C باعث افزایش جستجوی سراسری و افزایش آن باعث افزایش جستجوی محلی می باشد. در نتیجه در این تابع، نمودار C نموداری افزایشی خواهد بود تا رفته رفته از میزان جستجوی سراسری کم شده و جستجوی محلی افزایش پیدا کند. جدول (۲) تعدادی از قوانین استفاده شده در این تابع را نشان می دهد. قانون دوم، قانونی است که با کاهش C و در نتیجه آن افزایش جستجوی سراسری، مانع از افتادن در دام بهینگی محلی می شود. در قانون سوم به دلیل اینکه مقادیر پارامترها متوسط می باشد، مقدار C نیز متوسط انتخاب می شود و در قانون چهارم هم به دلیل شرایط نامناسب $d1$ و $d2$ نیازی به بررسی میزان شایستگی ذره نمی باشد و باید C کاهش پیدا کند تا جستجوی سراسری افزایش پیدا کرده و ذره از حالت کنونی خارج شود.



شکل (۴) تابع فازی مربوط به FPSO2

در این تابع مقدار W یک تابع خطی گرفته می شود و مقدار آن نزولی است تا رفته رفته از میزان جستجوی سراسری کم شده و جستجوی محلی، برای بهبود جواب بدست آمده افزایش پیدا نماید.

جدول (۲) تعدادی از قوانین فازی الگوریتم FPSO2

Rules	Input			Output
	d1	d2	NCBPE	
1	High	High	High	Low
2	Low	Low	Low	High
3	Low	Low	High	Low
4	Low	Low	Medium	Medium

۶. ارزیابی الگوریتم های پیشنهادی

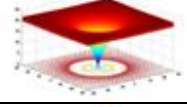
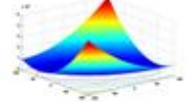
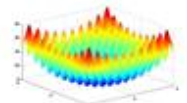
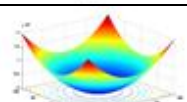
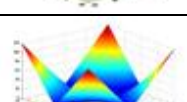
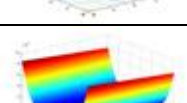
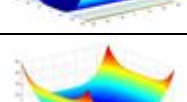
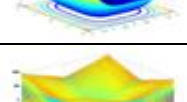
در هر دو الگوریتم پیشنهادی اساس کار بر روی تغییر وضعیت فعلی ذره می باشد. الگوریتم های پیشنهادی و الگوریتم PSO استاندارد و الگوریتم های TPSO و WPSO با استفاده از نرم افزار MATLAB 7.1 شبیه سازی شده و با استفاده از توابع محک^۱ استاندارد مورد ارزیابی قرار گرفته است. این توابع، توابع پرکاربردی می باشد که در ارزیابی های مشابه مورد استفاده قرار گرفته است [2,7,9,10,11,15]. در جدول (۳) توابع محک استفاده شده نشان داده می شود. همچنین جدول (۳) نمودار های هر یک از توابع محک را برای دو بعد نشان می دهد.

در این ارزیابی، ۲۰ ذره با ۵، ۱۰ و ۱۵ بعد و $c_1=c_2=2$ در نظر گرفته شده است. W برای توابع TPSO، FPSO2 و SPSO در بازه [0.9 0.1] در نظر گرفته شده است و بعد از ۴۰۰۰ بار تکرار نتایج مورد نظر بدست آمده است. الگوریتم های بالا را ۱۰ بار به طور مستقل اجرا کرده و نتایج ارائه شده میانگین نتایج بدست آمده از این ده مرتبه می باشد [18]. جداول (۴) الی (۱۱) نتایج این شبیه سازی ها را برای توابع محک مختلف و ابعاد مختلف را نشان می دهد. بهترین جواب و متوسط جواب بدست آمده برای الگوریتم های پیشنهادی و الگوریتم PSO استاندارد^۲ و الگوریتم های موجود مورد مقایسه قرار می گیرد. تمامی توابع محک استفاده شده مینیمم شونده هستند. شکل (۳) مقایسه نتایج بدست آمده برای الگوریتم های موجود SPSO، TPSO، WPSO و الگوریتم های پیشنهادی FPSO1 و FPSO2 را برای توابع محک مختلف با ۱۵ بعد نشان می دهد.

¹ Benchmark Function

² Standard PSO - SPSO

جدول (۳) توابع محک استاندارد

نام تابع	فرمول	دامنه	نمودار (n=2)
Ackley	$f(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)}$	$[-32,32]^n$	
Quadric	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100,100]^n$	
Rastrigin	$f(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12,5.12]^n$	
Sphere	$f(x) = \sum_{i=1}^n x_i^2$	$[-100,100]^n$	
Schwefel's Problem 2.22	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10,10]^n$	
Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$	
Quadric i.e. Noise	$f(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1)$	$[-1.28,1.28]^n$	
Generalized Schwefel's Problem 2.26	$f(x) = \sum_{i=1}^n x_i \sin(x_i)$	$[-500,500]^n$	

جدول (۴) نتایج بدست آمده از ابعاد مختلف برای تابع Ackley

۱۵ بعد		۱۰ بعد		۵ بعد		الگوریتم
مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	
0.219	2.15 e-12	0.076	2.66 e-15	0.056	8.88 e-16	FPSO1
0.175	3.82 e-11	0.051	2.66 e-15	0.032	8.88 e-16	FPSO2
1.198	4.25 e-2	0.044	1.25 e-5	0.019	4.26 e-8	TPSO
4.890	2.43	4.011	1.96	3.349	1.64	WPSO
7.664	4.72	6.457	3.05	5.488	2.31	SPSO

جدول (۵) نتایج بدست آمده از ابعاد مختلف برای تابع Quadric

الگوریتم	۵ بعد		۱۰ بعد		۱۵ بعد	
	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط
FPSO1	4.88 e-149	88.03	2.83 e-101	73.20	4.46 e-46	47.93
FPSO2	1.67 e-63	190.38	3.71 e-56	53.18	8.73 e-40	31.56
TPSO	3.49 e-15	11.73	4.34 e-10	16.97	3.56 e-8	35.13
WPSO	415.29	89.42	7.51	63.58	6.35 e-1	49.67
SPSO	2.47 e-15	9.31	1.13 e-13	18.18	2.55 e-8	33.73

جدول (۶) نتایج بدست آمده از ابعاد مختلف برای تابع Rastrigin

الگوریتم	۵ بعد		۱۰ بعد		۱۵ بعد	
	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط
FPSO1	0.301	0.777	0.388	2.328	1.203	2.031
FPSO2	0.406	1.235	0.535	1.412	1.148	2.134
TPSO	1.479	3.710	1.638	3.912	1.923	6.560
WPSO	1.549	1.773	2.063	2.219	2.845	3.599
SPSO	6.324	14.586	6.532	15.012	6.991	15.558

جدول (۷) نتایج بدست آمده از ابعاد مختلف برای تابع Sphere

الگوریتم	۵ بعد		۱۰ بعد		۱۵ بعد	
	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط
FPSO1	9.18 e-275	47.55	3.66 e-223	50.51	2.11 e-84	120.65
FPSO2	3.67 e-170	286.13	3.45 e-90	294.49	7.61 e-61	349.79
TPSO	2.07 e-116	261.12	6.81 e-60	280.19	7.70 e-41	300.86
WPSO	8.86 e+2	183.49	8.55 e+2	125.71	6.26 e+2	73.06
SPSO	4.16 e-133	262.86	1.35 e-45	287.23	2.43 e-26	295.48

جدول (۸) نتایج بدست آمده از ابعاد مختلف برای تابع Schwefel's Problem 2.22

الگوریتم	۵ بعد		۱۰ بعد		۱۵ بعد	
	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط
FPSO1	1.46 e-159	1.02	2.45 e-123	7.37	6.12 e-99	128.25
FPSO2	4.21 e-103	2.02	8.77 e-43	22.29	2.58 e-40	382.89
TPSO	1.88 e-67	3.85	4.41 e-34	40.95	1.81 e-31	667.03
WPSO	2.79 e-3	0.23	0.313	8.68	0.854	32.13
SPSO	5.26 e-10	6.01	6.61 e-2	60.66	0.402	8194.24

جدول (۹) نتایج بدست آمده از ابعاد مختلف برای تابع Rosenbrock

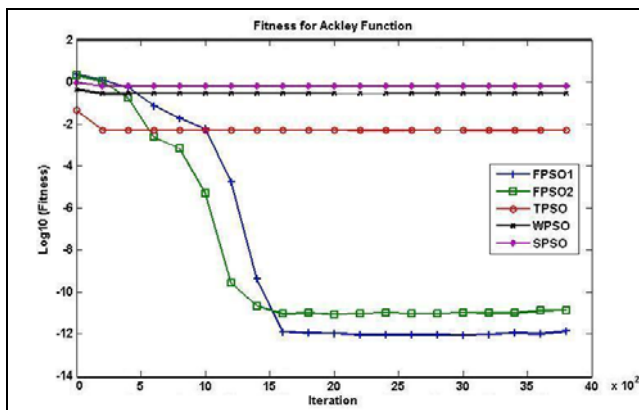
الگوریتم	۵ بعد		۱۰ بعد		۱۵ بعد	
	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط
FPSO1	0.0278	20677	0.3214	24551	0.7943	62924
FPSO2	0.1309	138578	0.8850	209051	1.0185	178012
TPSO	0.1628	451038	1.3962	525734	2.6661	657842
WPSO	4.8573	6919	7.6965	7124	9.3214	7854
SPSO	33.4833	38405	51.2411	99851	125.1253	175491

جدول (۱۰) نتایج بدست آمده از ابعاد مختلف برای تابع Quadric i.e. Noise

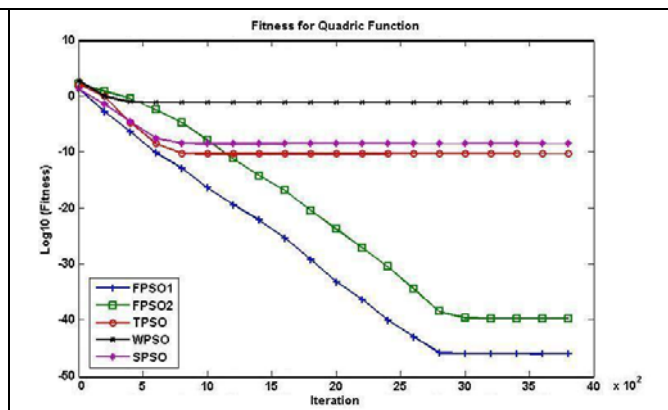
۱۵ بعد		۱۰ بعد		۵ بعد		الگوریتم
مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	
0.066	6.77 e-3	0.079	1.47 e-3	0.106	1.78 e-4	FPSO1
0.106	7.14 e-4	0.108	1.88 e-4	0.128	1.01 e-4	FPSO2
0.105	3.06 e-3	0.112	1.54 e-3	0.139	2.52 e-4	TPSO
0.065	2.70 e-3	0.081	1.58 e-3	0.112	6.08 e-4	WPSO
0.183	1.15 e-2	0.210	3.92 e-3	0.561	9.85 e-4	SPSO

جدول (۱۱) نتایج بدست آمده از ابعاد مختلف برای تابع Generalized Schwefel's Problem 2.26

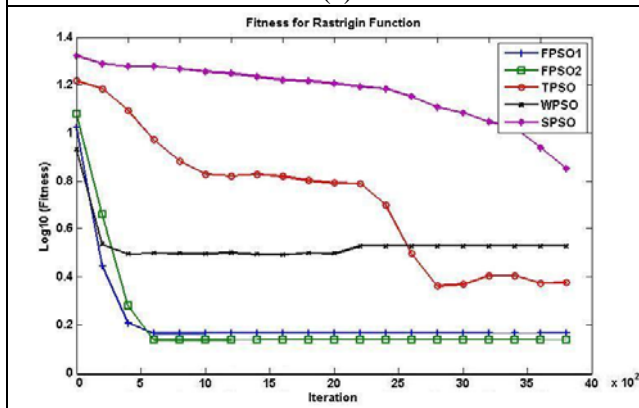
۱۵ بعد		۱۰ بعد		۵ بعد		الگوریتم
مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	مقدار متوسط	بهترین مقدار	
-269.28	-3020.19	-366.30	-2796.61	-416.52	-2323.21	FPSO1
-351.03	-3812.01	-386.27	-3419.63	-408.60	-2458.30	FPSO2
-238.83	-2901.24	-288.98	-2756.32	-330.50	-1949.02	TPSO
-170.06	-2844.72	-184.74	-1739.35	-245.57	-1540.86	WPSO
-185.77	-2741.21	-270.95	-2252.12	-283.04	-2037.16	SPSO



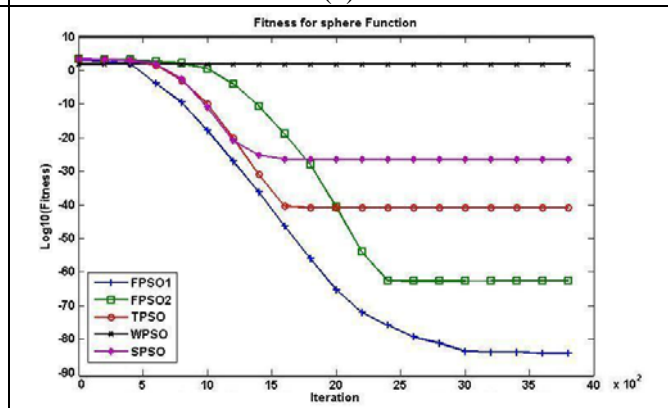
(a)



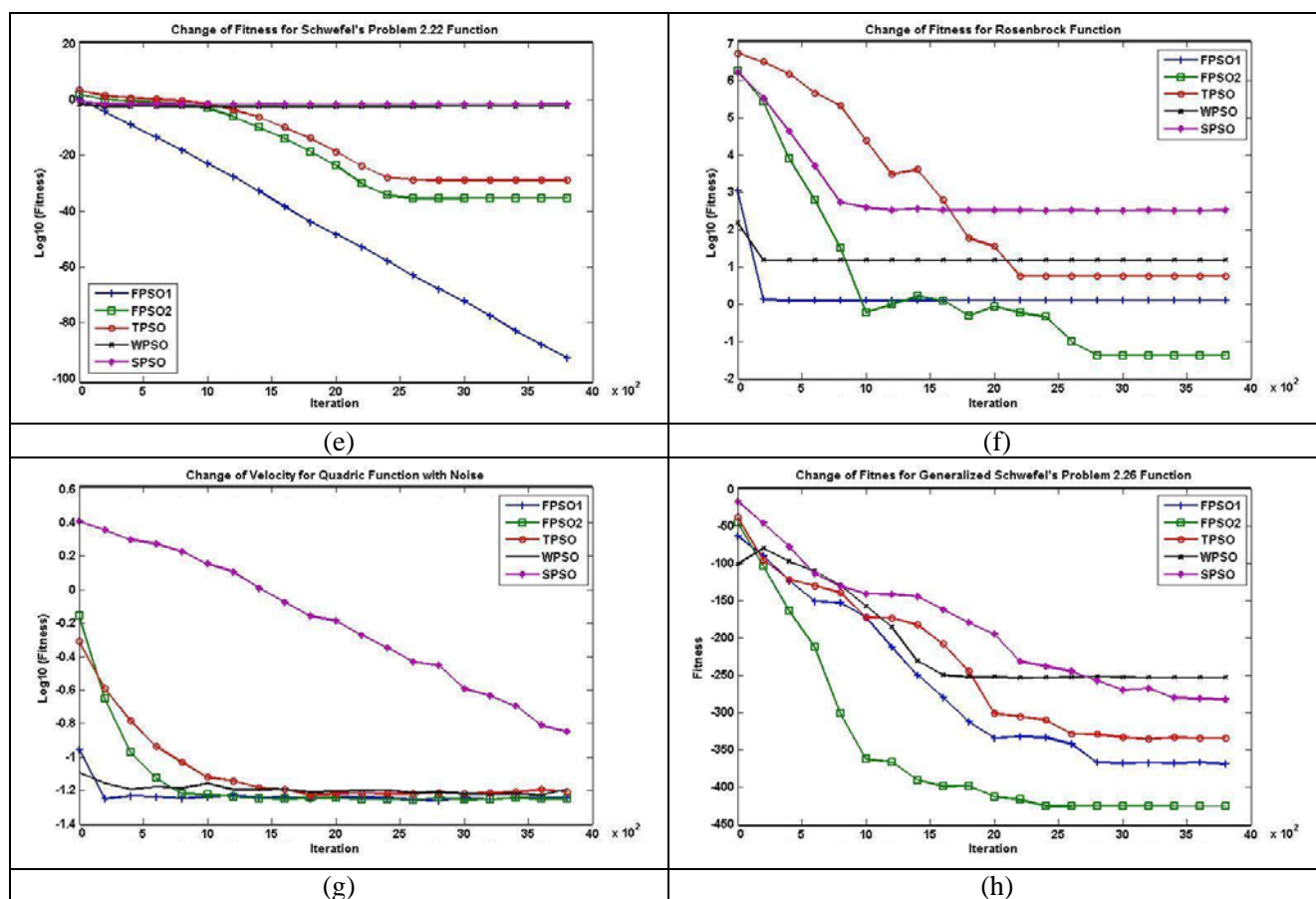
(b)



(c)



(d)



شکل (۳) نمودار میزان شاسستگی الگوریتم های مختلف با استفاده از توابع محک استاندارد. (a) تابع Ackley (b) تابع Quadric (c) تابع Rastrigin (d) Sphere (e) Schwefel's Problem 2.22 (f) Rosenbrock (g) Quadric i.e. Noise (h) Generalized Schwefel's Problem 2.26

نتایج بدست آمده از جداول (۴) الی (۱۱) نشان می دهد که رفتار الگوریتم های پیشنهادی FPSO1 و FPSO2 در تمامی توابع محک استفاده شده بهینه می باشد و نمودار مربوط به این دو الگوریتم پایین تر از الگوریتم های TPSON و WPSO و SPSO قرار دارد. اختلاف محسوسی که در نمودار هایی شکل (۳) وجود دارد برتری الگوریتم های پیشنهادی را نشان می دهد. این دو الگوریتم در تشخیص بهینه های محلی موفق بوده و توانسته اند خود را از دام بهینه های محلی رها کرده و جواب های بهینه ای را پیدا کنند. اما در شکل (۳) (g)، نمودار های بدست آمده برای تابع محک Quadric with Noise هم تراز می باشد، این همترازی به خاطر مقدار تصادفی است که در فرمول این تابع وجود دارد. ولی با توجه به جدول (۱۰) بهترین مقداری که الگوریتم های پیشنهادی برای تابع Quadric with Noise ارائه می کنند، بهتر از سایر الگوریتم ها می باشد. در کل می توان نتیجه گرفت که پارامتر های الگوریتم های پیشنهادی برای تصمیم گیری فازی استفاده می کنند (d1 و d2)، بهتر می تواند بهینه های محلی را تشخیص داده و سریع تر به جواب بهینه همگرا شوند.

۷. نتیجه گیری و کارهای آتی

با توجه به نتایج بدست آمده، می توان گفت که استفاده از توابع فازی رفتار الگوریتم PSO استاندارد را به مراتب بهبود می بخشد و این امکان را برای الگوریتم PSO فراهم می کند که در مقابل موقعیت های غیر عادی، عکس العمل مناسبی نشان دهد. همچنین، چون چندین پارامتر در تصمیم گیری دخیل می باشد، منطق فازی یکی از بهترین روش ها برای توابع غیر خطی در PSO می باشد. در آینده می توان از منطق فازی برای ارائه توابع غیر خطی جدید برای ضریب اینرسی و ضرایب شتاب استفاده کرد که مبنای تصمیم گیری آن پارامتر های دیگری مثل سرعت فعلی ذره، C_1 و C_2 یا اعداد تصادفی باشد.

مراجع

- [۱] محمد شیبانی، "مدلهایی ترکیبی برای بهینه سازی بر پایه اتوماتای یادگیر سلولی و محاسبات تکاملی و کاربرد های آنها"، پایان نامه کارشناسی ارشد مهندسی کامپیوتر، دانشگاه صنعتی امیر کبیر، ۱۳۸۵.
- [۲] حسین نظام آبادی پور و مجید رستمی شهر بابکی، "تعمیمی بر الگوریتم **GCBPSO**"، دوازدهمین کنفرانس مهندسی کامپیوتر ایران، صفحات ۲۹-۳۵، ۱۳۸۵.
- [۳] محمد شیبانی و محمد رضا میبدی، "**PSO-LA**: یک مدل جدید برای بهینه سازی"، دوازدهمین کنفرانس مهندسی کامپیوتر ایران، صفحات ۱۱۶۲-۱۱۶۹، ۱۳۸۵.
- [4] J. Kennedy, R. Eberhart, "*Particle Sswarm Ooptimization*", IEEE, pp. 1942-1948, 1995.
- [5] Y. Shi, R. Eberhart, "*A Modified Particle Swarm Optimizer*", IEEE International Conference on EvolutionaryComputation, USA, pp. 255- 262, 1998.
- [6] X. Cui, "*Document Clustering using Particle Swarm Optimization*", IEEE, pp. 185-191, 2005.
- [7] L. Hongbo, M. Abraham, "*Fuzzy Adaptive Turbulent Particle Swarm Optimization*", IEEE, pp. 39-47, 2005.
- [8] R. Eberhart, S. Yuhui, "*Fuzzy Adaptive Particle Swarm Optimization*", IEEE, pp. 101-106, 2001.
- [9] Q. Kang, L. Wang, Q. Wu, "*Research on Fuzzy Adaptive Optimization Strategy of Particle Swarm Algorithm*", International Journal of Information Technology, pp. 65-76, 2006.
- [10] X. Feng, J. Zhang, Z. Yang, "*Adaptive Particle Swarm Optimization on Individual Level*", IEEE, China, pp. 1215- 1218, 2002.
- [11] F. Bergh, A. Engelbrecht, "*A new locally convergent particle swarm optimizer*", IEEE, 2002.
- [12] M. Sugeno, "*Industrial Applications of Fuzzy Control*", Elsevier , New York, 1985.
- [13] J. Liebowitz, "*The Handbook Of Applied Expert Systems*",CRC Press, ISBN: 0849331064, 1999.
- [14] C. Krishnamoorthy, S. Rajeev, "*Artificial Intelligence and Expert Systems for Engineers*", CRC Press LLC, ISBN: 849391253, 1996.
- [15] D. Bratton, J. Kennedy, "*Defining a Standard for Particle Swarm Optimization*", Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007), 2007.
- [16] T. Kiink, J. S. Vesterstroem, J. Riget, "*Particle Swam Optimization with Spatial Particle Extension*", Proceedings of the IEEE Congress on Evolutionary Computation, pp. 1474-1479, 2002.
- [17] R. Brits, A. P. Engelbrecht, F. V. D. Bergh, "*Locating Multiple Optima using particle Swarm Optimization*", Applied Mathematics and Computation 189, Elsevier, pp. 1859–1883, 2007.
- [18] Z. Li, Y. Huan, H. Shang, "*Optimal Choice of Parameters for Particle Swarm Optimization*", Journal of Zhejiang University SCIENCE ISSN 1009-3095, pp. 528-534, 2005.
- [19] H. Liu, A. Abraham, W. Zhang, "*A Fuzzy Adaptive Turbulent Particle Swarm Optimization*", Int. J. Innovative Computing and Applications, pp.39–47, 2007.
- [20] J. Kennedy, R. Mendes, "*Population Sstructure and Particle Swarm Performance,*" in Proc. IEEE Congr. Evol. Comput., Honolulu, HI, pp. 1671–1676, 2002.