

طبیق پارامترهای گروه ذرات بهینه‌ساز مشارکتی با استفاده از منطق فازی

محمد رضا میبدی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر
mmeybodi@aut.ac.ir

زهرا افصحی

دانشکده مهندسی کامپیوتر، دانشگاه آزاد قزوین
Afsahi_ai@yahoo.com

چکیده: گروه ذرات بهینه‌ساز یک تکنیک بهینه‌سازی است و از حرکت و زندگی گروهی جانورانی مانند ماهی‌ها و پرندگان الهام می‌گیرد. از مشکلات عمدۀ این الگوریتم قرار گرفتن در بهینه‌های محلی و بزرگی ابعاد است. در این مقاله مدل جدیدی با نام $FCPSO-H$ ، که ترکیبی از منطق فازی و الگوریتم گروه ذرات بهینه‌ساز $CPSO-H$ می‌باشد پیشنهاد می‌گردد. در این مدل پارامترهای موثر در بروز رسانی سرعت و مکان هر ذره با استفاده منطق فازی طبیق داده می‌شود. به منظور ارزیابی، الگوریتم پیشنهاد شده روی توابع نمونه استاندارد آزمایش شده است. نتایج به دست آمده نشان از برتری این الگوریتم در مقایسه با PSO و $FPSO$ و $CPSO-H$ دارد.

واژه‌های کلیدی: بهینه‌سازی، گروه ذرات بهینه‌ساز، منطق فازی، یادگیری مشارکتی.

است. این ضریب با توجه به تعداد ذرات بهینه در هر همسایگی و به صورت دستی قابل تنظیم است. در [7] مدل دیگری از ساختار فازی در الگوریتم PSO پیشنهاد شده است. در این مدل بهینه‌ترین ارزش شایستگی بدست آمده در طی تعداد مشخصی از تکرار، فاصله بین بهترین ذره عمومی و بهترین تجربه هر ذره و تعداد نسل‌هایی که در آن بهینه‌ترین شایستگی بدون تغییر است، به عنوان ورودی‌های سیستم فازی و ضرایب c_1 و c_2 و W سه خروجی این سیستم هستند. در [6] مدل دیگری از $FPSO$ ارائه شده است که در آن همسایگی ذرات به صورت فازی تعیین می‌شود. در این مدل نیز پارامتری تحت عنوان درجه عضویت هر ذره با توجه به مکانی که در آن قرار دارد، اضافه شده است. در [8] مدل جدیدی به نام $FATPSO$ ارائه شده است. در این مدل دو پارامتر آستانه به منظور کنترل مکان و سرعت ذرات در فضای جستجو تعریف شده است که خروجی‌های سیستم فازی می‌باشند. بهترین شایستگی بدست آمده و بردار سرعت نظیر به آن، ورودی‌های سیستم فازی در این مدل می‌باشد. در [15] الگوریتمی تحت عنوان $CFPSO$ معرفی شده است. ورودی‌ها در این الگوریتم، فاصله بهترین مکان پیدا شده توسط هر گروه تا مکان ذره جاری، فاصله بهترین تجربه هر ذره در گروه تا مکان ذره جاری و بدترین جواب به دست آمده و خروجی‌ها ضرایب c_1 و c_2 می‌باشند. در این الگوریتم

۱- مقدمه

از سال ۱۹۹۵ تا کنون، الگوریتم‌های زیادی در زمینه توسعه دادن نسخه اصلی PSO در محیط‌های پیوسته و گسسته [14] و حل مشکلات آن ارائه شده است که می‌توان به الگوریتم‌هایی جهت تنظیم پارامترها [13,17] در معادلات سرعت، الگوریتم‌های مبتنی بر اجتماع که ساختارهای متفاوتی از توپولوژیهای اجتماعی برای همسایگی ذرات مطرح می‌کنند [2]، الگوریتم‌های ترکیبی که از ترکیب گروه ذرات بهینه‌ساز با سایر الگوریتم‌های بهینه‌ساز هوشمند از جمله الگوریتم‌های تکاملی [4]، اوتوماتای یادگیر [12] و منطق فازی به وجود آمده‌اند و استفاده از الگوریتم‌های چند جمعیتی که شامل مجموعه‌ای از گروه‌ها که با یکدیگر در تعامل هستند [3] اشاره کرد. استفاده از منطق فازی [5] در الگوریتم PSO علاوه بر اینکه مشکل اصلی این الگوریتم که قرار گرفتن در بهینه‌های محلی است را بهبود می‌دهد، می‌تواند از دو جنبه مفید باشد. اولاً می‌توان از داشت موجود و تجربیات مثبت ذرات در تعیین روند تغییرات وزن میانی استفاده کرد، ثانیاً این روند با گرفتن بازخورد از اجرای الگوریتم اصلاح می‌شود. مزیت دوم باعث ایجاد پویایی و خودتطبیقی در این الگوریتم می‌شود. در [5] گونه‌ای جدید از PSO مبتنی بر منطق فازی ارائه شده است. در این مدل پارامتری با عنوان ضریب جذب تعریف شده

محسوب می‌شود. ذره λ از گروه موقعیت X_i^d و سرعت V_i^d را در بعد l از فضای جستجو دارد. معادله به روز نمودن سرعت و موقعیت ذره در (۱) و (۲) نمایش داده شده است که جستجو بر اساس این فرمول‌ها برای رسیدن به یک بهینه شروع می‌شود. معادلات به روز رسانی مکان و سرعت برای λ این ذره به شکل زیر به نمایش در می‌آید:

$$\begin{aligned} V_i^d(t+1) &= V_i^d(t) + c_1 \text{rand}1_i^d(t)(pbest_i^d(t) - \\ &\quad X_i^d(t)) + c_2 \text{rand}2_i^d(t)(gbest(t) - X_i^d(t)) \quad (1) \\ X_i^d(t+1) &= X_i^d(t) + V_i^d(t+1) \quad (2) \end{aligned}$$

$pbest_i$ بهترین موقعیت ملاقات شده توسط ذره λ و $gbest$ بهترین موقعیت ملاقات شده توسط کل گروه است. c_1 و c_2 عدد تصادفی در بازه $[0,1]$ می‌باشند. C_1 و C_2 ضرایب مولفه‌های شناختی و اجتماعی می‌باشند.

۳- الگوریتم گروه ذرات بهینه‌ساز مشارکتی

الگوریتم گروه ذرات بهینه‌ساز مشارکتی اولین بار توسط آقای ون-دن-برگ و انگلبرج معرفی شد [9]. در این الگوریتم برای حل یک مسئله D بعدی، به جای تولید یک گروه D بعدی از n ذره، D گروه تک بعدی تشکیل می‌شود و هر گروه شامل n ذره است. در پایان هر گروه باید یک بعد از جواب نهایی را پیدا نماید. به منظور پیدا کردن جواب بهینه‌تر تبادل اطلاعات میان گروه‌ها نیاز است. تبادل اطلاعاتی در بین گروه‌ها نیازمند برداری است که قابل دسترس توسعه تمامی گروه‌ها باشد. این بردار، بردار محتوی نام دارد. برای یک مسئله D بعدی، این بردار نیز D بعد خواهد داشت. زمانیکه گروه j فال است، j سطر از بردار محتوی به وسیله بهترین ذره گروه زام و سایر عناصر این بردار توسط $gbest$ های سایر گروه‌ها ($D-1$ گروهی که در زمان ارزیابی گروه زام ثابت باقی می‌مانند) پر می‌شود. در هر لحظه این بردار به شکل (j,z) است. امتیاز این الگوریتم این است که تابع خطای f ، بعد از به روز رسانی هر عضو از بردار محتوی ارزیابی می‌شود. ارزیابی بردار محتوی برای هر گروه به شکل معادلات (۳) و (۴) است.

تعداد قوانین فازی مورد استفاده ۴ عدد می‌باشد. در [16] نیز الگوریتمی مشابه ارائه شده است. ورودی‌ها و خروجی‌های سیستم فازی همانند آنچه که در [15] تعریف شده می‌باشد با این تفاوت که الگوریتم پایه در این مدل، PSO استاندارد است. در همگی این روشها پارامترهای جدیدی (ضریب جذب، تعداد نسل‌هایی که در آن شایستگی بدون تغییر است، درجه عضویت، پارامترهای آستانه و بدترین جواب بدست آمده در گروه‌ها) در الگوریتم PSO استاندارد تعریف شده است که سادگی الگوریتم PSO استاندارد تا حدی از بین برده است. از طرفی با اضافه شدن تعداد پارامترها بار محاسباتی الگوریتم نیز بالاتر می‌رود. برای غلبه بر مشکل بزرگی ابعاد نیز الگوریتم‌هایی از جمله الگوریتم‌های CPSO و MCPSO گزارش شده‌اند [18]. در الگوریتم‌های CPSO و MCPSO تبادل اطلاعات بین ذرات از نوع همکاری و یا رقابتی است و یادگیری به شیوه مشارکتی است [9,10]. در این مقاله، یک روش بهینه‌سازی با یک ساختار کنترل کننده فازی روی الگوریتم CPSO-H پیشنهاد شده است. با استفاده از الگوریتم CPSO از تجمع ذرات در یک ناحیه ممانت کرده و تنوع بین ذرات حفظ می‌شود. در واقع در این مدل سعی شده علاوه بر حل مشکل بزرگی ابعاد، بر مشکل اصلی الگوریتم PSO که قرار گرفتن در بهینه‌های محلی است حتی در یک بعد خاص غلبه شود [11]. نتایجی که بر روی ۸ تابع استاندارد نشان می‌دهد که این روش راندمان بالاتری نسبت به الگوریتم‌های PSO، CPSO-H و Fuzzy PSO دارد. در ادامه این مقاله و در بخش ۲ به معرفی الگوریتم PSO می‌پردازد. در بخش ۳ به بررسی الگوریتم CPSO-H می‌پردازد. در بخش ۴ مدل Fuzzy CPSO-H یا FCPSO-H ارائه می‌شود. بخش پنجم اختصاص به ارایه نتایج شبیه سازی‌ها دارد و بخش پایانی نتیجه گیری می‌باشد.

۲- الگوریتم گروه ذرات بهینه‌ساز

بهینه سازی گروه ذرات الگوریتمی است که از رفتار اجتماعی موجوداتی که به صورت انبوه زندگی می‌کنند، الهام گرفته است [1]. جمعیت اولیه این الگوریتم در یک بازه مشخص و به صورت تصادفی مقداردهی می‌شود. هر ذره در فضای D بعدی به عنوان یک راه حل از مساله

الگوریتم PSO استاندارد بهبود می‌بخشد، اما ضرایب C_k قادر به تطبیق خود با تغییرات محیط نیستند [13]. از این جهت توانایی الگوریتم در تطبیق با محیط پویا ضعیف و همگرایی به جواب بهینه را تا حدی محدود می‌شود. بر اساس ورودی‌های سیستم فازی، ضرایب c_1 و c_2 به عنوان خروجی، تحت تاثیر حرکت ذرات و در واقع بهینه عمومی و بهینه‌های محلی هر گروه (با توجه به الگوریتم CPSO-H) هستند. از طرفی این بهینه‌ها در هر مرحله از اجرای الگوریتم و با توجه به تغییرات محیط تغییر می‌کنند. در نتیجه الگوریتم با کمک این ضرایب، خود را با تغییرات محیط تطبیق می‌دهد. با توجه به اینکه کاهش c_1 و c_2 باعث افزایش جستجوی سراسری و افزایش آن باعث افزایش جستجوی محلی می‌باشد، نمودار c_1 و c_2 نموداری صعودی خواهد بود تا رفته رفته از میزان جستجوی سراسری کم شده و جستجوی محلی افزایش پیدا کند. قوانین ارائه شده در جدول ۱ نیز بر این اصل استوار می‌باشد. قوانین ارائه شده به طور واضح مبنای تصمیم‌گیری را مشخص می‌کند. تعداد قوانین فازی در سیستم پیشنهادی ۴۹ عدد است.

مجموعه قوانین فازی در پروسه فازی کردن ورودی‌ها و دی‌فازی کردن خروجی یکسان انتخاب و تعریف شده است: $(H_n, M_n, L_n, Z, L_p, M_p, H_p)$. که به ترتیب و از چپ به راست "منفی بزرگ"، "منفی میانه"، "منفی کوچک"، "صفر"، "ثبت بزرگ"، "ثبت میانه"، "ثبت کوچک" تعریف شده‌اند. در ادامه مراحل الگوریتم پیشنهادی FCPSO-H ارائه شده است:

- ۱- گروه ذرات بر اساس الگوریتم CPSO-H مقداردهی اولیه می‌شوند.
- ۲- ارزیابی ارزش هر ذره با توجه بهتابع شایستگی مورد آزمایش.
- ۳- محاسبه کردن بهینه‌ترین ذره هر گروه $P_{j,gbest}$ و بهینه‌ترین تجربه هر ذره در گروه $P_{j,pbest_i}$.
- ۴- مشخص کردن مقادیر $D1$ و $D2$ و وارد کردن منطق فازی در الگوریتم (اعمال قوانین فازی).
- ۵- بدست آوردن نرخ تغییرات ضرایب c_1 و c_2 با استفاده از دی‌فازی کردن.

بعد از یکبار تکرار الگوریتم CPSO، بردار محتوی به دست آمده به عنوان ورودی الگوریتم PSO در نظر گرفته می‌شود. برای محاسبه بهینه‌ترین ذره در این مرحله، از بردار محتوی حاصل از الگوریتم CPSO استفاده می‌شود. برای تمامی ذرات جمعیت، الگوریتم PSO اجرا می‌شود و راه حل حاصل از اجرای الگوریتم PSO، بر اساس تعداد ابعاد هر زیرگروه تکه تکه شده و به عنوان بردار محتوی به الگوریتم CPSO باز گردانده می‌شود. نتیجه به کارگیری و ترکیب این دو الگوریتم، افزایش سرعت الگوریتم و کاهش احتمال قرار گرفتن در مینیمم‌های محلی است. در [11] نشان داده شده است که CPSO-H کارآیی بهتری نسبت به CPSO دارد.

$\text{If } f(b(j, P_j \cdot x_i)) < f(b(j, P_j \cdot pbest_i))$ $P_j \cdot pbest_i = P_j \cdot x_i$	(۳)
--	-----

$\text{If } f(b(j, P_j \cdot pbest_i)) > f(b(j, P_j \cdot gbest_i))$ $P_j \cdot gbest_i = P_j \cdot pbest_i$	(۴)
--	-----

۴- الگوریتم پیشنهادی FCPSO-H

الگوریتم H، ترکیبی از بهینه‌سازی گروه ذرات CPSO-H و منطق فازی است. همانند مدل بهینه‌سازی گروه ذرات، دارای n ذره است که هر یک از ذرات دارای یک بردار موقعیت و یک بردار سرعت است. فضای جستجویی که ذرات در آن قرار دارند D بعدی است. در الگوریتم پیشنهادی به جای D زیرگروه تک بعدی، زیرگروه‌هایی در نظر گرفته می‌شوند که بیشتر از یک بعد دارند. به منظور ایجاد پویایی و بالا بردن قابلیت تطبیق با تغییرات محیطی، یک سیستم فازی کنترل کننده پیشنهاد شده است. این سیستم دارای دو ورودی و یک خروجی است. ورودی‌ها به وسیله بهینه‌ترین ذره هر گروه $P_{j,gbest}$ و بهینه‌ترین تجربه هر ذره در گروه $P_{j,pbest_i}$ و طبق فرمول‌های (۵) و (۶) حاصل می‌شود.

$D1_{j,i} = P_{j,pbest_i} - x_i$	(۵)
----------------------------------	-----

$D2_{j,i} = P_{j,gbest} - x_i$	(۶)
--------------------------------	-----

خروچی این سیستم ضرایب C_k است. استراتژی برنامه-نویسی ایستا در طی حجم زیادی از آزمایشات ضرایب C_k مناسبی را بدست می‌آورد. اگر چه این الگوریتم‌ها کارآیی همگرایی به جواب بهینه را نسبت به

Fuzzy PSO دست پیدا کرده است. البته لازم به ذکر است که در رسیدن به این جواب‌ها دارای سرعت بالاتری نسبت به Fuzzy PSO می‌باشد. در کورد الگوریتم پیشنهادی باید گفت که در تابع F6 به عدد صفر رسیده است. در تابع F8 و F3 و F4 هم به بهترین جواب رسیده است. البته خیلی انتظار رسیدن به جواب‌های بهتری نسبت به Fuzzy PSO نمی‌رفت. زیرا تعداد ابعاد در این آزمایش پایین است و این امر باعث شده تا این الگوریتم در برخی از توابع به جواب‌های بهتری برسد. همانطور که انتظار می‌رفت با بالا رفتن ابعاد فضای جستجو الگوریتم پیشنهادی به جواب‌های بهینه‌تری در تمامی توابع رسیده است. الگوریتم پیشنهادی در توابع F6 و F4 به عدد صفر رسیده است. و همانطور که در نمودارها مشخص است از سرعت بالای الگوریتم CPSO و کارآئی الگوریتم Fuzzy PSO استفاده کرده و هم سرعت بالاتری نسبت به سایر روش‌ها دارد وهم به جواب‌های بهتری رسیده است. در این آزمایش الگوریتم FuzzyPSO همچنان در بعضی از توابع به نتایج بهتری نسبت به CPSO دست یافته است اما سرعت پایین‌تری دارد.

۶- نتیجه‌گیری

در این فصل یک الگوریتم بهینه‌سازی جدید به نام FCPSO-H که ترکیبی از منطق فازی و نوع خاصی از بهینه‌سازی گروه ذرات به نام CPSO-H می‌باشد، ارائه گردید. با توجه به نتایج بدست آمده، می‌توان گفت که استفاده از منطق فازی این امکان را برای الگوریتم PSO استاندارد فراهم می‌کند تا در مقابل تغییرات محیطی، عکس‌العمل مناسبی نشان دهد. و بتواند خود را با تغییرات محیط تطبیق دهد. از طرفی با استفاده از یادگیری مشارکتی در الگوریتم CPSO-H سرعت همگرایی افزایش و مشکل قرار گرفتن در بهینه‌های محلی را تا حدودی حل شده است. در مجموع با استفاده از ویژگی‌های مثبت این دو الگوریتم توانسته به جواب‌های قابل قبولی دست پیدا کند. کارآئی این الگوریتم در فضای جستجو با ابعاد بالا، بیشتر است، هرچند که در فضای جستجو با ابعاد پایین نیز به نتایج خوبی دست پیدا کرده است.

۶- در ادامه سرعت و مکان هر ذره در صورت بهینه‌تر شدن ذره (با توجه به ارزیابی تابع شایستگی مورد بررسی) به روز می‌شوند.

۷- بررسی شرط خاتمه

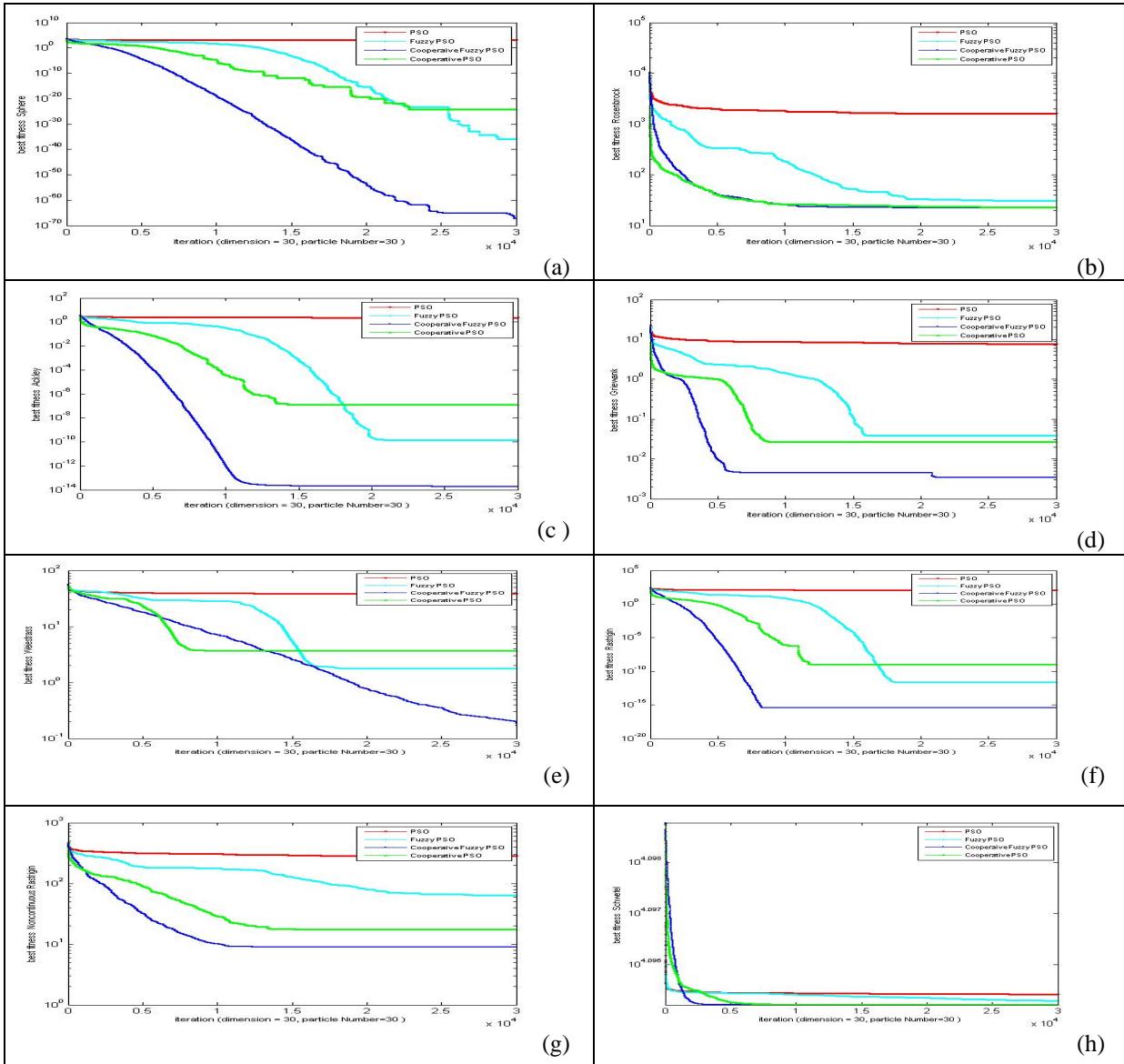
باید به این نکته توجه شود که با اعمال سیستم فازی کنترل کننده به الگوریتم CPSO-H در هر یک از گروه‌ها با ابعاد کوچکتر انجام می‌پذیرد.

جدول ۱: ارزش‌های منطقی

E_1/E_2	Hn	Mn	Ln	Z	Lp	Mp	Hp
Hn	Lp	Lp	Mn	Ln	Ln	Z	Lp
Mn	Mp	Lp	Ln	Ln	Ln	Z	Lp
Ln	Hp	Hp	Z	Mn	Z	Lp	Mp
Z	Hp	Hp	Mp	Z	Mp	Hp	Hp
Lp	Hp	Lp	Z	Mn	Lp	Lp	Hp
Mp	Mp	Lp	Z	Ln	Z	Mp	Hp
Hp	Lp	Z	Ln	Ln	Ln	Hp	Hp

۵- نتایج شبیه‌سازی‌ها

برای مقایسه الگوریتم پیشنهادی FCPSO-H با الگوریتم‌های Fuzzy PSO ، CPSO-H و PSO شبیه‌سازی‌هایی بر روی هشت تابع استاندارد صورت گرفته است که معمولاً به عنوان معیار سنجش الگوریتم‌های بهینه‌سازی مورد استفاده قرار می‌گیرند. آزمایش بر روی تابع تست و در دو قسمت ۱۰ بعدی و ۳۰ بعدی انجام شده است. نتایج ارائه شده در شکل ۱ میانگین ۲۵ بار اجرای الگوریتم‌ها می‌باشد. نتایج ارائه شده در جدول ۲ و ۳ به ترتیب در فضای جستجوی ۱۰ بعدی و ۳۰ بعدی، برتری الگوریتم پیشنهادی را در برابر روش‌های موجود نشان می‌دهد. از مقایسه این نتایج می‌توان نتیجه گرفت که رفتار تابع پیشنهادی خیلی بهتر از سه الگوریتم PSO و CPSO-H و Fuzzy PSO می‌باشد. همانطور که در جدول ۲ دیده می‌شود، با توجه به اینکه تعداد ابعاد پایین است الگوریتم Fuzzy PSO در برخی از توابع نظری F1 و F7 به نتایج بهتری رسیده است. در سایر توابع نیز به جواب‌های قابل قبولی رسیده است. روند خطی تغییرات در نمودارهای الگوریتم PSO نشان دهنده قرار گرفتن این الگوریتم در بهینه‌های محلی است. الگوریتم CPSO با توجه به کوچک بودن ابعاد به نتایجی بدتر از



شکل ۱ : میانگین بهترین تابع شایستگی ذرات نسبت به تعداد تکرارهای الگوریتم ۲۵ تکرار، ۳۰ ذره، ۳۰۰۰۰ تکرار ۸ تابع (a) Sphere (b).Schwefel(h), Noncontinuous Rastrigin(g). Rastrigin (f) . Weierstrass (e) . Greiwank (d) . Ackley (c).Rosenbrock

- Subpopulation," *Proceedings of the Third Genetic and Evolutionary Computation Conference*, Vol. 1, pp. 469-476, 2001.
- [5] A. M. Abdelbar, S. Abdelshahid and D. C. Wunsch, "Fuzzy PSO: A Generalization of Particle Swarm Optimization," *Proceedings of International Joint Conference on Neural Networks*, Montreal, Canada, July, 2005.
- [6] H. Liu and A. Abraham, "A Hybrid Fuzzy Variable Neighborhood Particle Swarm Optimization Algorithm for Solving Quadratic Assignment Problems," *Journal of Universal Computer Science*, Vol. 13, No. 7, pp. 1032-1054, 2007.

مراجع - ۷

- [1] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, pp. 1942-1948, 1995.
- [2] J. Kennedy, S. Worlds and M. Minds, "Effects of Neighborhood Topology on Particle Swarm Performance," *Proceedings of the IEEE Congress on Evolutionary Computation*, Vol. 3, pp. 1931-1938, July 1999.
- [3] F. V. Bergh and A. P. Engelbrecht, "Effects of Swarm Size on Cooperative Particle Swarm Optimizers," *Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, USA, pp. 892-899, 2001.
- [4] M. Løvberg, T. Rasmussen and T. Krink, "Hybrid Particle Swarm Optimizer with Breeding and

جدول ۲: مقایسه میانگین و واریانس الگوریتم CPSO-H، PSO و الگوریتم FCPSO-H در ۱۰ بار تکرار بعد و بعد از ۲۵ بار تکرار

10D	F1	F2	F3	F4
Original PSO	207.12 \pm 61.6	97.387 \pm 68.824	0.019495 \pm 0.7199	0.0026229 \pm 1.5058
Fuzzy PSO	0 \pm 3.3297e-314	1.2629 \pm 0.11752	1.7895e-028 \pm 2.1316e-014	0.0015774 \pm 0.59075
CPSO-H	0 \pm 1.0872e-311	2.8252 \pm 3.9871	5.9603e-029 \pm 7.1054e-015	0.0025699 \pm 0.23848
Fuzzy CPSO-H	0 \pm 9.1231e-214	1.2453 \pm 2.5446	0 \pm 3.5527e-015	0.00011212 \pm 0.012316
10D	F5	F6	F7	F8
Original PSO	0.39671 \pm 8.8185	1.2521 \pm 6.2292	15.034 \pm 34.066	0.011978 \pm 4151.1
Fuzzy PSO	0.13468 \pm 0.52964	2.9801e-030 \pm 0	2.1778 \pm 8.8818e-015	0.001993 \pm 4150.8
CPSO-H	0.078251 \pm 0.0021739	1.7744e-028 \pm 0	0.23333 \pm 1	2.6164e-017 \pm 4150.4
Fuzzy CPSO-H	1.4817e-006 \pm 0.0039687	0 \pm 0	0.44444 \pm 1	9.1909e-026 \pm 4150.3

جدول ۳: مقایسه میانگین و واریانس الگوریتم CPSO-H، PSO و الگوریتم FCPSO-H در ۳۰ بار تکرار بعد و بعد از ۲۵ بار تکرار

30D	F1	F2	F3	F4
Original PSO	13119 \pm 941.31	7127.1 \pm 1610.6	0.012453 \pm 2.1537	7.3611 \pm 0.30691
Fuzzy PSO	1.1967e-040 \pm 8.7323e-072	567.57 \pm 77.241	3.3604e-020 \pm 3.4204e-010	0.022151 \pm 0.0007942
CPSO-H	1.2981e-048 \pm 4.7755e-033	6.7224 \pm 23.7	4.5419e-014 \pm 1.676e-007	0.031942 \pm 0.00046773
Fuzzy CPSO-H	1.8117e-1134 \pm 3.0918e-077	21.261 \pm 14.574	2.5819e-028 \pm 6.3949e-014	0 \pm 3.6682e-005
30D	F5	F6	F7	F8
Original PSO	0.82859 \pm 36.845	40.833 \pm 109.4	184.2 \pm 291.79	\pm 12456 0.21194
Fuzzy PSO	0.76157 \pm 1.2692	1.2733e-023 \pm 6.2172e-012	1154.2 \pm 50	\pm 12454 0.13615
CPSO-H	4.975 \pm 1.4738	5.981e-018 \pm 1.5987e-014	367.34 \pm 25	\pm 12451 0.0010138
Fuzzy CPSO-H	0.026626 \pm 0.031958	5.6097e-031 \pm 0	11.556 \pm 9	\pm 12451 9.8048e-022

- [15] M. H. Noroozibeyrami and M. R. Meybodi, "Improving Particle Swarm Optimization using Fuzzy Logic", *Proceedings of the Second Iranian Data Mining Conference*, Amir Kabir University of Technology, Tehran, Iran, Sept. 21-22, 2008.
- [16] M. H. Noroozibeyrami and M. R. Meybodi, "Cooperative Fuzzy Particle Swarm Optimization", *Proceedings of the 2nd Joint Congress on Fuzzy and Intelligent Systems*, Malek Ashtar University of Technology, Tehran, Iran, 28-30 October, 2008.
- [17] Y. Zheng, L. Ma, L. Zhang and Qian, "Empirical Study of Particle Swarm Optimizer with Increasing Inertia Weight," *In Proceeding of IEEE Congress on Evolutionary Computation*, pp. 221-226, 2003.
- [18] B. Niu, Y. Zhu, X. He and H. Wu, "MCPSO: A Multi-Swarm Cooperative Particle Swarm Optimizer," *Elsevier Mathematics and Computation*, No. 185, pp. 1050–1062, 2007.
- [7] S. H. Zahiri and S. A. Seyedin, "Swarm Intelligence Based Classifiers," *Journal of the Franklin Institute* 344, pp. 362–376, 2007.
- [8] H. Liu and A. Abraham, "A Fuzzy Turbulent Particle Swarm Optimization," *Int. J. Innovative Computing and Applications*, vol. 1, no. 1, 2007.
- [9] J. J. Liang and A. K. Qin, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, June 2006.
- [10] F. V. Berg and A. P. Engelbrecht, "A Cooperative Approach to Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 8, no. 3, June 2004.
- [11] F. V. Bergh, "An Analysis of Particle Swarm Optimizers," Survey, November 2001.
- [12] R. Rastegar, M. R. Meybodi and K. Badie, "A New Discrete Binary Particle Swarm Optimization based on Learning Automata," *Proceedings of International Conference on Machine Learning and Applications (ICMLA2004)*.
- [13] A. Ratnaweera, S. Halgamuge, and H. Watson, "Particle Swarm Optimization with Self-Adaptive Acceleration Coefficients," *Proceedings of the First International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 2411-2418, December 2003.
- [14] J. Kennedy and R. C. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm," *Conference on System and Cybernetics*, pp. 4104-4109, 1997.