

الگوریتم ژنتیک تطبیقی با استفاده از منطق فازی

سجاد یزدانی^۱ و محمدباقر منهج^۱

^۱دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران) ، {Sajjad.Yazdani, Menhaj}@aut.ac.ir

چکیده - این مقاله سعی در کنترل الگوریتم ژنتیک برای ایجاد مصالحه مناسب بین کاوش و بهره‌وری دارد. بر این مبنا ابتدا پارامترهای مناسب برای پایش و کنترل الگوریتم بررسی می‌شوند. سپس یک کنترل‌کننده فازی مناسب ارائه می‌شود که با فیدبک گرفتن از پارامترهای پایش الگوریتم، آن را کنترل می‌کند. نتایج حاصل از پیاده‌سازی بر روی توابع آزمون، این مطلب را نشان می‌دهد که کنترل تطبیقی الگوریتم با استفاده از سیستم فازی باعث بهبود عملکرد آن و ارائه جواب‌های بهتر می‌شود. کلیدواژه - کنترل پارامتر الگوریتم، الگوریتم ژنتیک تطبیقی، کنترل فازی، پایش الگوریتم ژنتیک.

حاصل از پیاده‌سازی و مقایسه الگوریتم کنترل شده با حالت معمولی آن در بخش پنجم مقاله مرور می‌شود و در انتها بخش ششم به نتیجه‌گیری مقاله می‌پردازد.

۱- مقدمه

پیچیدگی روزافزون مسائل مهندسی و زیاد شدن ابعاد مسائل در دنیای امروز، بسیاری از محققان را به استفاده از الگوریتم‌های ابتکاری و فرا ابتکاری تشویق می‌کند. از سوی دیگر دغدغه بسیاری از محققان که از الگوریتم‌های ابتکاری استفاده می‌کنند تنظیم پارامترهای این الگوریتم‌هاست. الگوریتم ژنتیک به عنوان یکی از قوی‌ترین الگوریتم‌های ابتکاری شناخته شده است و ترکیب آن با منطق فازی مورد توجه محققان قرار گرفته است [1,2,3,4,5]. در این مقاله سعی داریم روشی برای کنترل پارامترهای الگوریتم ژنتیک ارائه کنیم. هر چند که تاکنون به روش‌های مشابه دیگر از کنترل‌کننده فازی در الگوریتم ژنتیک استفاده شده است اما در این مقاله کنترل دو پارامتر از الگوریتم به صورت همزمان مورد توجه است. و از طرفی پارامتر جدیدی برای پایش الگوریتم ژنتیک مورد استفاده قرار می‌گیرد. از طرفی نا یقینی زیاد در الگوریتم‌های ابتکاری به عنوان یک سامانه برای کنترل‌کننده‌ها و از سوی دیگر پیچیدگی پارامترها، ما را بر آن می‌دارد تا از یک کنترل‌کننده فازی برای کنترل الگوریتم ژنتیک استفاده کنیم [6].

در ادامه این مقاله، ابتدا به بررسی الگوریتم ژنتیک و نحوه اثرگذاری پارامترهای آن می‌پردازیم سپس در بخش سوم مقاله نحوه پایش الگوریتم ژنتیک مورد بررسی قرار می‌گیرد و با استفاده از متغیرهای مناسب برای پایش الگوریتم در بخش چهارم مقاله یک کنترل‌کننده فازی مناسب ارائه می‌شود. نتایج

۲- پیاده‌سازی الگوریتم ژنتیک و بررسی پارامترهای آن

الگوریتم ژنتیک در سال ۱۹۶۲ توسط جان هلند و بر اساس نظریه تکامل داروین ارائه شده است [7]. در این مقاله از الگوریتم ژنتیک حقیقی که در آن مقادیر کروموزوم‌ها به صورت اعداد حقیقی تعریف می‌شوند استفاده شده است این در حالی است که در الگوریتم ژنتیک باینری مقادیر کروموزوم‌ها به صورت باینری بیان می‌شوند و مسائل با متغیرهای حقیقی، در فضای کد شده، حل می‌شوند. پارامترهایی مانند احتمال وقوع جهش و احتمال عملگر تقاطع نمونه‌های از پارامترهای الگوریتم ژنتیک می‌باشند که در روند الگوریتم اثرگذارند. این پارامترها به تفصیل در مراجع مختلف بررسی شده‌اند.

تا کنون متدهای زیادی برای بهبود الگوریتم ژنتیک ارائه شده است. تغییر عملگرهای الگوریتم و اضافه کردن عملگر جدید به الگوریتم ژنتیک نمونه‌ای از این بهبودها می‌باشند. یکی از بهبودهای مناسب الگوریتم، مقیاس بندی خطی مقدار شایستگی کروموزوم‌هاست [7]. در این متد پس از محاسبه مقدار شایستگی کروموزوم‌ها یک تابع خطی درجه یک بر روی آن‌ها اعمال می‌شود (مطابق معادله ۱). مقدار پارامترهای این تابع خطی با توجه به مقادیر تابع شایستگی کروموزوم‌ها و منطبق با روابط ۲ و ۳ تعیین می‌شوند.

$$Fit_i^{scaled} = A * Fit_i + B \quad (1)$$

$$A = Cm \frac{Fit_{ave}}{Fit_{max} - Fit_{min}} \quad (2)$$

$$B = -A Fit_{min} \quad (3)$$

هدف برای متغیرهای تصمیم‌گیری) و یا در سطح ژنوتایپ (مقادیر کروموزم‌ها در فضای کد شده مسئله) باشد. در مرجع [1] پارامتر Var که به خوبی پراکندگی جمعیت در سطح فنوتایپ را مشخص می‌کند، معرفی شده است.

$$Var = \frac{Fit_{max} - Fit_{ave}}{Fit_{max} - Fit_{min}} \quad (5)$$

این پارامتر همگرایی اعضا در سطح فنوتایپ پایش می‌کند. پارامتر دیگری که در اینجا معرفی می‌شود فاصله اقلیدسی اعضا در سطح ژنوتایپ است (معادله ۷).

$$NGD_d = \frac{\sqrt{(\max_{i=1:N} x_i^d)^2 - (\min_{i=1:N} x_i^d)^2}}{\sqrt{up_d^2 - down_d^2}} \quad (6)$$

$$GD = mean(NGD) \quad (7)$$

پارامتر GD پراکندگی جمعیت را در سطح کروموزم‌ها بیان می‌کند. با استفاده از این دو پارامتر می‌توان پراکندگی جمعیت را به صورت نسبی و مطلق در فضای جستجو پایش کرد. مقدار Var متناسب با فاصله اعضا نسبت به فاصله بهترین و بدترین جواب است که به عبارتی پراکندگی نسبی جمعیت را به ما می‌دهد. از سوی دیگر پارامتر GD به صورت مطلق فاصله اعضا از یکدیگر، در فضای مسئله مشخص می‌کند.

پارامتر دیگری که اطلاعاتی از عملکرد الگوریتم را می‌تواند بیان کند تعداد تکرارهایی است که در آن بهترین جواب تغییر نکرده است. این پارامتر را با UC نشان می‌دهیم. بزرگ شدن این پارامتر به این معناست که بهره‌وری در الگوریتم کم است یا آنکه جواب بهینه فرا محلی به دست آمده است. در برخی از کاربردها این پارامتر را به عنوان شرط توقف الگوریتم در نظر می‌گیرند.

۴- پیاده‌سازی کنترل‌کننده فازی

نکته قابل توجه در کنترل الگوریتم‌های ابتکاری این است که سامانه تحت کنترل تقریباً غیرقابل مدل‌سازی، دارای تأخیر زمانی و دینامیک پیچیده است. منطق فازی به خوبی می‌تواند برای کنترل چنین سامانه‌ای استفاده شود. به خصوص که در کنترل فازی نظر فرد خبره به خوبی مدل می‌شود و یکی از بهترین راه‌ها برای کنترل چنین سامانه‌ای مدل کردن اطلاعات یک فرد خبره است. این امر در کنترل‌کننده فازی در قوانین اگر آنگاه پیاده‌سازی می‌شود.

در این روابط Fit_i^{scaled} مقدار شایستگی پس از مقیاس بندی، Fit_i مقدار شایستگی قبل از مقیاس بندی Fit_{ave} مقدار میانگین شایستگی اعضای جمعیت، Fit_{max} مقدار بیشینه شایستگی اعضا و Fit_{min} مقدار کمینه آن‌ها می‌باشد. مقدار Cm یک پارامتر است که با آن می‌توان فشار انتخاب را در الگوریتم ژنتیک کنترل کرد. به راحتی می‌توان نشان داد که با این مقیاس بندی شایستگی بدترین عضو جامعه برابر صفر و شایستگی بهترین عضو برابر با $Cm.Fit_{ave}$ خواهد شد. با توجه به این، هرچه Cm بزرگ‌تر باشد احتمال انتخاب اعضا با شایستگی بیشتر افزایش میابد و این باعث حذف سریع‌تر اعضا با شایستگی کم می‌شود (که به خوبی فشار انتخاب را بیان می‌کند).

پارامتر مهم دیگری که در این مقاله به آن می‌پردازیم مقدار انحراف معیار هنگام اعمال عملگر جهش است. در این مقاله عملگر جهش، با احتمال ۰.۱ و منطبق با رابطه ۴ اعمال می‌شود.

$$x_{d,i}^{muted} = x_{d,i} + (up_d - down_d) \cdot \mathcal{N}(0, \sigma) \quad (4)$$

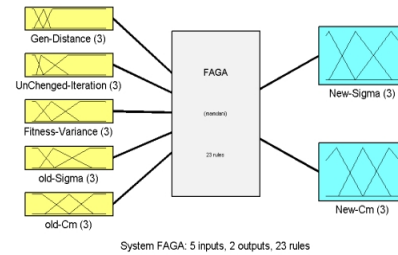
جایی که $x_{d,i}$ و $x_{d,i}^{muted}$ به ترتیب مقادیر کروموزم‌ها بعد و قبل از جهش برای کروموزم‌ها نام در بعد d م نشان می‌دهند. همچنین up_d و $down_d$ به ترتیب مقادیر کمینه مجاز و بیشینه مجاز، برای متغیر تصمیم‌گیری در بعد d را نشان می‌دهند. $\mathcal{N}(0, \sigma)$ بیانگر توزیع نرمال حول مبدا با انحراف معیار σ می‌باشد. مقدار σ یکی از پارامترهای مناسب برای کنترل الگوریتم است. چنانچه میدانیم هنگامی که جمعیت همگرا شده است عملگر جهش باعث کاوش می‌شود و هرچه مقدار σ در این حالت بیشتر باشد کاوش بیشتر صورت می‌گیرد و امکان گیر کردن در بهینه‌های محلی را کاهش می‌دهد. در این مقاله انتخاب والدین توسط روش SUS و عملگر تقاطع با احتمال ۰.۸ و با روش ترکیب خطی والدین اعمال می‌شود [7,1].

۳- پارامترهای پایش وضعیت اجرای الگوریتم

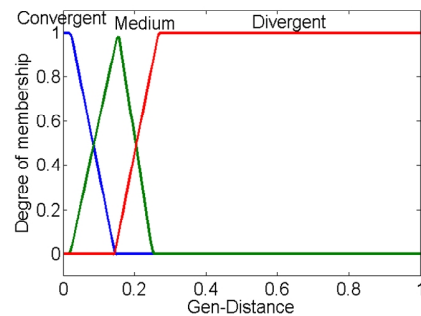
در این قسمت به بررسی و استخراج پارامتری می‌پردازیم که بتواند به خوبی وضعیت جمعیت را در هر تکرار مشخص کند. برای پایش روند اجرای الگوریتم پارامترهای مختلفی را می‌توان بیان نمود. پایش الگوریتم می‌تواند در سطح فنوتایپ (مقدار تابع

همان طور که در شکل ۱ مشاهده می‌شود کنترل‌کننده پیشنهادی، ۲ خروجی، ۵ ورودی و ۲۳ قانون می‌باشد. خروجی‌های کنترل‌کننده مقادیر σ و Cm می‌باشند که با توجه به آنچه در قسمت‌های قبل بیان شد به خوبی می‌توانند روند همگرایی الگوریتم و کاوش و بهره‌وری آن را تحت تأثیر قرار دهند. ورودی‌ها عبارت‌اند از پارامترهای GD ، Var و UC که پارامترهای پایش الگوریتم می‌باشند و همچنین مقادیر σ و Cm که به کنترل‌کننده بازخورد می‌شوند.

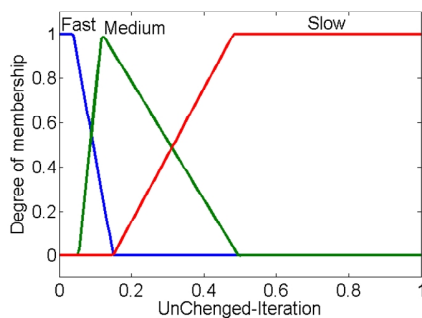
برای هر ورودی و خروجی ۳ متغیر زبانی در نظر گرفته شده است. شکل ۲ توابع عضویت برای ورودی و خروجی سیستم فازی را نشان می‌دهد. همان طور که مشاهده می‌شود دامنه هر یک از متغیرها متناسب با پارامتر ورودی آن در نظر گرفته شده است. برای ورودی‌ها توابع ابتدایی و انتهایی بازه‌ها از نوع ذوزنقه‌ای می‌باشند و تابع عضویت میانی از نوع مثلثی. برای خروجی سیستم فازی همه توابع مثلثی در نظر گرفته شده‌اند. این توابع و



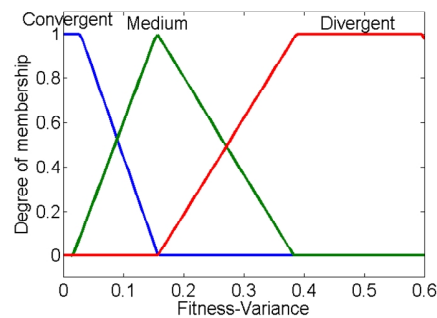
شکل ۱: نمای کلی کنترل‌کننده فازی ارائه شده



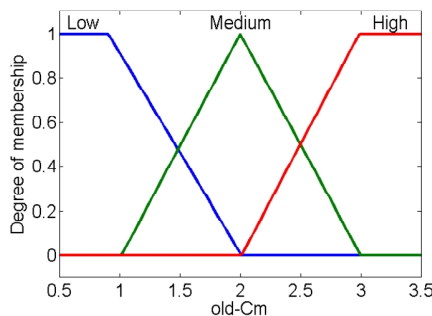
(الف)



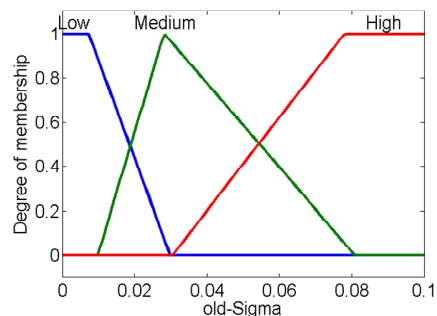
(ب)



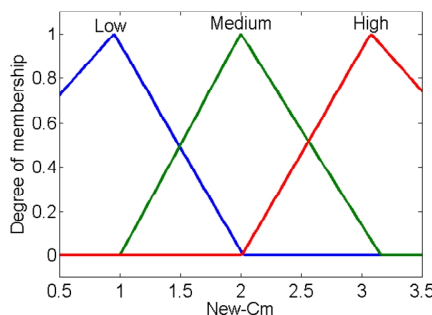
(ج)



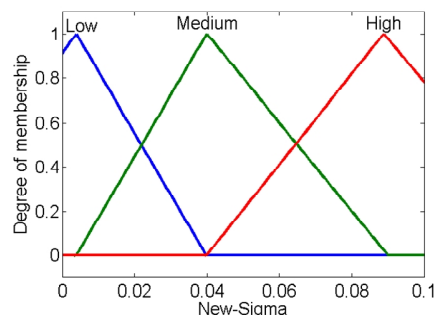
(د)



(ه)



(و)



(ز)

شکل ۲: توابع عضویت برای ورودی‌های سیستم فازی پیشنهاد شده. (الف) GD (ب) Var (ج) UC (د) σ^t (ه) Cm^t (و) σ^{t+1} (ز) Cm^{t+1}

پارامترهای مربوطه با سعی و خطا به دست آمده‌اند.

$$F_3(\mathbf{X}) = 418.9829.n + \sum_{i=1}^n -x_i \sin(\sqrt{|x_i|}), \mathbf{X} \in [-500, 500]^n \quad (10)$$

$$F_4(\mathbf{X}) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e, \mathbf{X} \in [-32, 32]^n \quad (11)$$

$$F_5(\mathbf{X}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2], \mathbf{X} \in [-30, 30]^n \quad (12)$$

$$F_6(\mathbf{X}) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|, \mathbf{X} \in [-10, 10]^n \quad (13)$$

$$F_7(\mathbf{X}) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2, \mathbf{X} \in [-100, 100]^n \quad (14)$$

$$F_8(\mathbf{X}) = \max_i \{|x_i|, 1 \leq i \leq n\}, \mathbf{X} \in [-100, 100]^n \quad (15)$$

در پیاده‌سازی شرایط را برای تمامی الگوریتم‌های مورد آزمون به صورت یکسان در نظر گرفته شده است. در این مقاله بعد مسائل ۳۰، تعداد اعضای جمعیت ۱۰۰ و تعداد تکرار ۱۰۰۰ در نظر گرفته شده است. مقدار احتمال جهش ۰.۱ و مقدار احتمال عملگر تقاطع ۰.۸ در نظر گرفته شده است. مقدار Cm در حالتی که کنترل نمی‌شود برابر با ۳ در نظر گرفته شده است.

جدول ۱ نتایج حاصل از این پیاده‌سازی در ۵۰ بار اجرای مستقل الگوریتم نشان می‌دهد.

جدول ۱ - نتایج حاصل از پیاده‌سازی برای ۸ تابع آزمون در ۵۰ تکرار مستقل با الگوریتم‌های جستجوی پرندگان (PSO) ژنتیک حقیقی (RGA) و ژنتیک تطبیقی ارائه شده (FAGA)

الگوریتم تابع	بهترین جواب دیده‌شده				میانگین جواب‌های دیده‌شده
	FAGA	RGA	FAGA	GA	PSO[8]
F ₁	5.3211E-76	7.6145E-67	2.9313E-39	5.6233E-23	1.12E-47
F ₂	5.6292E-17	1.7680E-1	8.7966	93.4323	61.7205
F ₃	20.0767	157.7887	2796.3	5209.7	5793.69
F ₄	2.7154E-14	0.1987	5.5922	12.4164	7.19459
F ₅	4.53951E-14	6.7165E-2	1.3889	4.3148	11.924
F ₆	9.2169E-46	2.8875E-35	2.8907E-7	3.5039E-3	0.001128
F ₇	2.5442E-12	1.5194E-15	9.4527E-4	3.4259E-5	3.12E-05
F ₈	8.6598E-13	1.0986E-22	6.6329E-5	2.8404E-3	0.062396

بر مبنای جدول ۱ به خوبی واضح است که کنترل‌کننده مناسبی برای الگوریتم ارائه شده است.

با توجه به توابع عضویت بیان‌شده، ۲۳ قانون اگر-آنگاه برای کنترل‌کننده در نظر گرفته شده است. قوانین بر این مبنا بیان شده‌اند که هرگاه جمعیت همگرا می‌شود و به مدت زیاد تغییری در بهترین جواب یافته شده ایجاد نشود، با کاهش فشار انتخاب و افزایش کاوش، جمعیت پراکنده شود تا از احتمال گیر افتادن در بهینه محلی جلوگیری کند. از سوی دیگر هنگامی که جمعیت واگرا شده است با افزایش فشار انتخاب آن را همگرا می‌کنیم تا بهره‌وری در الگوریتم افزایش یابد. افزایش یا کاهش فشار انتخاب با افزایش و کاهش مقدار Cm نسبت به حالت قبل صورت می‌گیرد. علاوه بر این هنگام همگرایی جمعیت مقدار σ یا به عبارتی مقدار جهش را افزایش می‌دهیم تا پراکندگی اعضا بیشتر شود و از همگرایی حول نقطه بهینه محلی جلوگیری کنیم. و هنگامی که اعضا پراکنده شده‌اند با کاهش آن به همگرایی الگوریتم کمک می‌کنیم.

در این مقاله، غیرفازی ساز مورد استفاده از نوع مرکزی می‌باشد همچنین روابط "و" و "یا" با استفاده از کمینه و بیشینه محاسبه می‌شوند.

از آنجا که تغییرات حاصل از کنترل‌کننده بر روی پارامترها به سرعت بر روی روند الگوریتم اثرگذار نیست و به عبارتی سامانه تحت کنترل دارای تأخیر زمانی است نمی‌توان کنترل‌کننده را در هر تکرار مورد استفاده قرارداد. در صورت تغییر پارامترها در هر تکرار در برخی موارد کنترل‌کننده یا الگوریتم به سمت ناپایداری پیش می‌رود. بدین منظور بعد از هر بار تغییر پارامترها توسط کنترل‌کننده تعداد مشخصی تکرار (۱۵ تکرار) از الگوریتم پیش می‌رود و سپس این کنترل‌کننده فراخوانی می‌شود.

۵- نتایج پیاده‌سازی

روش معرفی‌شده برای کنترل الگوریتم ژنتیک را بر روی ۸ تابع آزمون برگرفته از مرجع [8] که دارای خواص متفاوتی هستند و دارای چندین بهینه محلی می‌باشند پیاده‌سازی شده است. روابط (۸) الی (۱۵) این توابع را بیان می‌کنند.

$$F_1(\mathbf{X}) = \sum_{i=1}^n x_i^2, \mathbf{X} \in [-100, 100]^n \quad (8)$$

$$F_2(\mathbf{X}) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10], \mathbf{X} \in [-5.12, 5.12]^n \quad (9)$$

۶- نتیجه‌گیری

در این مقاله اثر پارامترهای الگوریتم ژنتیک بر کاوش و بهره‌وری به اختصار مورد بررسی قرار گرفت و همچنین پارامترهای مناسب جهت پایش الگوریتم ژنتیک معرفی شد. یک کنترل‌کننده فازی مناسب با توجه به پارامترهای بررسی‌شده ارائه شد و تغییرات الگوریتم با ۸ تابع آزمون مختلف مورد بررسی قرار گرفت. نتایج حاصل از شبیه‌سازی به خوبی نشان می‌دهد که کنترل‌کننده ارائه‌شده پارامترهای الگوریتم را به خوبی کنترل می‌کند و به دلیل ایجاد مصالحه مناسب بین کاوش و بهره‌وری جواب‌های به دست آمده از الگوریتم بهبود یافته‌اند.

مراجع

- GENETIC ALGORITHM," *Iranian Journal of Fuzzy Systems*, vol. 8, no. 3, pp. 1-21, 2011.
- [2] I.G. Damousis, K.J. Satsios, D.P. Labridis, and P.S. Dokopoulos, "Combined fuzzy logic and genetic algorithm techniques—application to an electromagnetic field problem," *Fuzzy Sets and Systems*, vol. 129, pp. 371–386, 2002.
- [3] M. Lozano F. Herrera, "Adaptive genetic operators based on coevolution with fuzzy behaviors," in *IEEE Trans. Evolution.*, vol. 5 (2), 2001, pp. 1–18.
- [4] Hongbo Liu, Zhanguo Xu, and Ajith Abraham, "Hybrid Fuzzy-Genetic Algorithm Approach for Crew Grouping," in *5th International Conference on Intelligent Systems Design and Applications (ISDA'05). Proceedings.*, 2005, pp. 332–337.
- [5] Mark Last and Shay Eyal, "A fuzzy-based lifetime extension of genetic algorithms," *Fuzzy Sets and Systems*, vol. 149, pp. 131–147, 2005.
- [6] F. Herrera and M. Lozano, "Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions," *Soft Computing*, vol. 7, pp. 545–562, 2003.
- [7] حسین نظام آبادی پور، الگوریتم وراثتی-مفاهیم پایه و مباحث پیشرفته، چاپ نخست. کرمان، ایران: دانشگاه شهید باهنر کرمان، ۱۳۸۹.
- [8] Changhe Li and Shengxiang Yang, "An Adaptive Learning Particle Swarm Optimizer for Function Optimization," *IEEE Congress on Evolutionary Computation, CEC'09.*, pp. 381–388, 2009.
- [1] Hossein Nezam-Abadipour, Sajjad Yazdani, Malihe Maghfouri Farsangi, and Mahdi Neyestani, "A SOLUTION TO AN ECONOMIC DISPATCH PROBLEM BY A FUZZY ADAPTIVE