کاربرد الگوریتم ژنتیک آشوبگونه فازی در بهینه سازی توابع پیچیده

محسن فرهادی، محمدرضا جاهدمطلق، ناصر مزینی، حامد رحیم‌اف[۱]

چکیده: مسائل بهینه‌سازی در عمل بسیار پیچیده‌اند و الگوریتم‌های کلاسیک قادر به حل آنها بطور رضایت‌بخش نیستند. آنها دارای دو محدودیت افتادن در تلهٔ مینیمم محلی و صرف وقت زیاد جهت جستجو می‌باشند. از آنجا که یک حرکت آشوبگونه می‌تواند حالات مختلفی از یک دامنه را بدون تکرار طی نماید، لذا استفاده از جستجوی آشوب برای حل مسائل بهینه‌سازی کاربردی گردیده است. از آنجا که اکثر داده‌های اطراف ما بصورت فازی می‌باشند، تغییراتی در روش جستجوی آشوب اعمال نموده به‌طوریکه بتوانند داده‌های فازی را دریافت نمایند، بنابراین این روش بطور ساده‌تر و موثرتری مورد استفاده قرار خواهد گرفت. در نهایت نظر به اینکه سرعت رسیدن به پاسخ و دقت پاسخ تولیدی در روش‌های جستجوی آشوب وابسته به نرخ رشد تابع آشوبگونه می‌باشد، برای بدست آوردن نرخ رشد بهینه، از الگوریتم ژنتیک استفاده نموده و روشی به نام الگوریتم ژنتیک آشوبگونه فازی را معرفی نموده‌ایم. ما با استفاده از روش پیشنهادی اقدام به پیدا نمودن مینیمم واقعی دو تابع معروف Schwefel و Rastrigin که دارای تعداد زیادی مینیمم محلی هستند می‌پردازیم. مقایسهٔ نتایج بدست آمده از حل توابع فوق با پاسخ‌های مسأله در حالت غیر فازی کارایی روش ما را جهت حل توابع پیچیده نشان می‌دهد.

کلمات کلیدی: بهینه‌سازی فازی، جستجوی آشوب ، الگوریتم ژنتیک آشوبگونه فازی.

۱ – مقدمه

آشوب یک پدیدهٔ جهانی در سیستمهای دینامیک غیرخطی می‌باشد. از زمانی که لورنز در سال ۱۹۶۳ مجذوب‌کننده آشوبگونه متعارف را نشان داد، آشوب بطور وسیعی در اجتماع‌های مهندسی و ریاضی مورد مطالعه قرار گرفت. آشوب در ظاهر، یک حرکت نامنظم دارد و ظاهراً رفتار تصادفی غیرقابل پیش‌بینی بوسیلهٔ یک سیستم غیرخطی قطعی وابسته به شرایط اولیه از خود نشان می‌دهد. متغیرهای آشوب طبق قاعده‌اشان می‌توانند از هر حالتی بدون تکرار به ناحیه امن بروند. بعلت خصوصیات دینامیکی و تصادفی متغیرهای آشوب، جستجوی آشوب در روش رسیدن به هدف در سیستم خبره و فرار از بهینهٔ محلی بیشتر از جستجوی تصادفی انجام پذیر می‌باشد[۱]، بنابراین این روش برای بهینه‌سازی محاسبات کاربردی گردیده است. Li و Jiang یک الگوریتم بهینه‌سازی آشوب (COA)[2] را برای حل مسألهٔ بهینه‌سازی پیچیده مطرح نمودند[۲]. Hasegawa یک روش جستجوی آشوب را برمبنای هر دو تأثیر دینامیک آشوب و جستجوی ممنوع (Tabu search) مطرح نمود[۳]. Ji و Tang یک الگوریتم فرآیند شبیه‌سازی بر مبنای آشوب را بوسیلهٔ ادغام کردن یک سیستم اغتشاش و فرآیند شبیه‌سازی گسترش دادند[۴]. Zuo و Fan با ادغام الگوریتم بهینه‌سازی آشوب و الگوریتم انتخاب کلونی یک الگوریتم ایمن جستجوی آشوب (CSIA) طرح‌ریزی نمودند[۵]. ما نیز با اعمال اصلاحاتی در روش COA

[۱] – مهندس محسن فرهادی، دانشجوی کارشناسی ارشد رشته هوش مصنوعی و رباتیک دانشگاه علم و صنعت ایران narfarhadi@gmail.com ، دکتر محمدرضا جاهد مطلق، دانشیار دانشکدهٔ مهندسی کامپیوتر دانشگاه علم و صنعت ایران jahedmr@iust.ac.ir. دکتر ناصر مزینی، استادیار دانشکدهٔ مهندسی کامپیوتر دانشگاه علم و صنعت ایران mozayani@iust.ac.ir. مهندس حامد رحیم‌اف، عضو هیئت علمی دانشگاه صنعتی شاهرود hrahimov@gmail.com.

[2] chaos optimization algorithm

باعث بهبودی در سرعت در رسیدن به پاسخ و همچنین دقت پاسخ‌های تولیدی در مسائل بهینه‌سازی پیچیده، گردیدیم[۹].

از طرفی اگر سیستم‌های بهینه‌سازی را طوری طراحی کنیم که بتواند داده‌های مبهم را دریافت کنند، این داده‌ها می‌توانند به طور ساده‌تر و موثرتری در اجرا به کار برده شوند. در سالهای گذشته چندین نوع از مسائل بهینه‌سازی فازی مطرح شده‌اند که تفاوت آنها در نوع تجزیه و تحلیلشان می‌باشد. از جمله روشهای مطرح شده عبارتند از: بهینه‌سازی فازی با استفاده از الگوریتم ژنتیک فازی توسط Bukley و Hayashi [۶]، شبیه‌سازی فازی با استفاده از آشوب فازی و استفاده از آن در حل مسائل بهینه‌سازی فازی توسط Bukley و Hayashi [۷]، بهینه‌سازی فازی با استفاده از جستجوی تابو توسط Li ، Liao و Yu [۸]. ما در تحقیقات قبلی با اعمال اصلاحاتی در الگوریتم جستجوی آشوب فازی و ترکیب آن با الگوریتم ژنتیک توانستیم روش جدیدی به نام الگوریتم ژنتیک آشوبگونه فازی برای بهینه سازی فازی ارایه دهیم [۱۱،۱۰] که این روش پاسخ دقیقتری نسبت به سایر روشهای فوق تولید می‌نماید. در این مقاله قصد داریم کاربرد روش پیشنهادی را در بدست آوردن مقدار مینیمم واقعی توابعی که دارای تعداد زیادی مینیمم محلی می‌باشند نشان دهیم.

ادامهٔ این مقاله بدین‌صورت سازماندهی شده‌است. در بخش بعدی ما ابتدا به معرفی برخی از نمادهای استفاده شده در این مقاله پرداخته سپس در بخش سوم به بحث در مورد تولید مجموعه‌های آشوبگونه فازی می‌پردازیم. در بخش چهارم روش بهینه‌سازی فازی با استفاده از الگوریتم ژنتیک آشوبگونه فازی (FCGA)[3] را شرح داده و در بخش پنجم با استفاده از روش FCGA اقدام به پیدا نمودن مینیمم واقعی دو تابع معروف Schwefel و Rastrigin می‌پردازیم. مقایسهٔ نتایج بدست آمده از حل توابع فوق با پاسخ‌های مسأله در حالت غیر فازی کارایی روش ما را جهت حل توابع پیچیده نشان می‌دهد. در نهایت در بخش شش، نتایج و پیشنهاداتی را عنوان خواهیم نمود.

۲ – نمادگذاری مقاله

در این بخش به معرفی برخی از نمادهای استفاده شده در این مقاله می‌پردازیم. با قرار دادن یک خط روی نماد یک مجموعهٔ فازی را نشان می‌دهیم. بنابراین $\overline{A}$، $\overline{B}$، $\overline{C}$، . . . ، $\overline{X}$، $\overline{Y}$ و $\overline{Z}$ نشاندهندهٔ مجموعه‌های فازی هستند. تمام مجموعه‌های فازی ما زیرمجموعه‌های فازی از اعداد حقیقی خواهند داشت. اگر $\overline{A}$ یک مجموعه فازی باشد، آنگاه $\overline{A}(x)$ نشاندهندهٔ ارزش تابع عضویت در x است. یک عدد فازی مثلثی $\overline{N}$ بوسیلهٔ سه عدد $a < b < c$ بدین صورت تعریف می‌گردد (الف) $\overline{N}(x) = 0$ برای $x \le a$ و $x \ge c$، (ب) $\overline{N}(b) = 1$ و (ج) نمودار $\overline{N}(x) = y$ یک پاره‌خط مستقیم از $(a,0)$ به $(b,1)$ در $[a,b]$ و یک پاره خط دیگر از $(b,1)$ به $(c,0)$ در $[b,c]$ می‌باشد. ضمنا، عدد فازی مثلثی بصورت $\overline{N}(a,b,c)$ نوشته و از اصل گسترش، جهت محاسبات استاندارد در مجموعه‌های فازی استفاده شده است.

۳ – روش تولید مجموعه‌های آشوبگونه فازی

سیستم های آشوبگونه، سیستم‌هایی هستند که دارای تعداد نامحدودی بستر جذب بوده و به شرایط اولیه حساس هستند. حساسیت به شرایط اولیه (Sensitivity to Initial Conditions) در یک سیستم دینامیک، بر این اساس است که اختلاف بسیار کوچک در مقادیر اولیه سیستم، سبب بوجود آمدن نتایج بسیار متفاوت گردد[۲]. برای نمونه، نگاشت محلی $f(x)$ (رابطه ۱) می‌تواند یک نگاشت لوجستیک معکوس (Revers Logistic) [۹] باشد.

$$f(x_n) = x_{n+1} = ((M - rx_n)/M)^2 \qquad (۱)$$

در رابطه ۱ متغیر r نشاندهنده نرخ رشد تابع آشوبگونه می‌باشد، بطوریکه تابع ۱ به ازای M=1 در بازه $1.4 < r < 2$ مخلوطی از نظم و آشوب می‌باشد و برای مقادیر $r > 1.5$ کل فضای پاسخ مسأله را پوشش می‌دهد.

در اینجا ابتدا اعداد فازی را به اعداد مثلثی محدود کرده و آنرا بصورت $\overline{N} = (a,b,c)$ می‌نویسیم. برای تولید توالی آشوبگونه $\overline{X}_n = (a_n, b_n, c_n)$ را در $[0, M]$، سه مقدار متفاوت برای r انتخاب نموده آنها را r_1, r_2, r_3 می‌نامیم، بطوریکه f

[3] Fuzzy Chaotic Genetic Algorithm

آشوبگونه باشد. اگر فرض نماییم $f_i(x)=((M-r_i x)/M)^2$ و $i=1,2,3$ آنگاه توالی آشوبگونهٔ $\overline{X}_n$ توسط رابطهٔ (۲) ایجاد می‌گردد.

$$\overline{X}_{n+1}=(a_{n+1},b_{n+1},c_{n+1}) \tag{۲}$$

از آن جهت که باید شرط $a<b<c$ برای اعداد فازی ما رعایت گردد، بنابراین مقادیر a_{n+1}، b_{n+1} و c_{n+1} را بترتیب برابر کوچکترین، وسطی و بزرگترین مقدار $f_1(a_n)$ ، $f_2(b_n)$ و $f_3(c_n)$ در نظر می‌گیریم. اگر ما r_iها را بدرستی انتخاب، و مقدار اولیه $\overline{X}_0=(a_0,b_0,c_0)$ را در فاصله $[0,M]$ انتخاب نماییم، آنگاه تابع (۲) یک توالی آشوبگونه از اعداد فازی مثلثی در $[0,M]$ تولید می‌کند.

۴ – الگوریتم ژنتیک آشوبگونه‌فازی

برای پیاده‌سازی الگوریتم ژنتیک آشوبگونه فازی (FCGA) بصورت زیر عمل می‌نماییم:

برای هر کروموزوم $3k$ ژن در نظر گرفته، بطوریکه هر سه ژن، نمایندهٔ r_1,r_2,r_3 (مقادیر نرخ رشد تابع آشوبگونه) بوده و می‌تواند مقداری بین ۱/۵ تا ۲ اختیار نمایند و k برابر تعداد متغیر مستقل ورودی می‌باشد. برای ارزیابی کروموزوم‌ها، با استفاده از هر سه ژن به تولید مجموعه‌های آشوبگونه فازی به ازای هر متغیر ورودی نموده و با جایگذاری آنها در مسأله، خروجی را محاسبه می‌نماییم و برای مقایسه خروجی‌های بدست آمده جهت پیدا نمودن پاسخ بهینه، مرکز ثقل آنها را محاسبه نموده و با مقایسه آنها بهترین مرکز ثقل خروجی بعنوان پاسخ بهینه انتخاب می‌گردد که در واقع خروجی تابع ارزیابی الگوریتم ژنتیک ما نیز می‌باشد. سپس به تولید نسل جدید و اعمال جهش می‌پردازیم. الگوریتم را تا هنگامی‌که بهترین کروموزوم در S مرتبه تکرار شود، ادامه می‌دهیم. در انتها ژن‌های بهترین کروموزوم، مقادیر بهینهٔ نرخ رشد و خروجی تابع ارزیابی معادل آن، پاسخ بهینهٔ مسألهٔ ما می‌باشد.

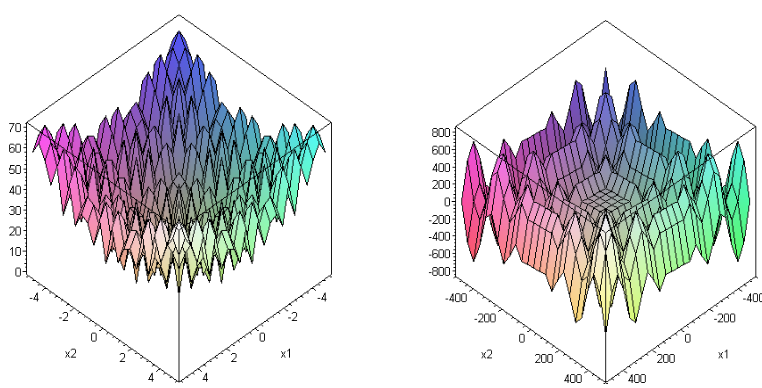
۵ – حل مسائل Schwefel و Rastering

توابع Schwefel (۳) و Rastering (۴) دو تابع پیچیده با تعداد زیادی مینیمم محلی می‌باشند.

$$f_3(x)=\sum_{i=1}^{n} x_i.\sin(\sqrt{|x_i|}) \qquad -500\le x_i\le 500 \tag{۳}$$

$$f_4(x)=10.n+\sum_{i=1}^{n}(x_i^2-10.\cos(2.\pi.x_i)) \qquad -5.12\le x_i\le 5.12 \tag{۴}$$

می‌دانیم پاسخ مینیمم واقعی تابع Schwefel در حالت قطعی (غیرفازی) n*۴۱۸/۹۸۲۹- بوده (n مشخص کنندهٔ تعداد متغیرهای ورودی تابع Schwefel می‌باشد) و این پاسخ به ازای ورودی‌های $x_i=-420.9687, i=1:n$ بدست می‌آید و پاسخ مینیمم واقعی تابع Rastrigin در حالت قطعی برابر صفر بوده و این پاسخ به ازای ورودی‌های $x_i=0, i=1:n$ بدست می‌آید. شکل ۱-الف نمودار تابع Schwefel وشکل ا-ب نمودار تابع Rastrigin در حالت قطعی به ازای $n=2$ می‌باشند.



۱–ب ا–الف

شکل (۱-الف) نمودار تابع Schwefel و (ا-ب) نمودار تابع Rastrigin به ازای $n = 2$ (در حالت قطعی)

ما می‌خواهیم مینیمم واقعی توابع ۳ و ۴ در حالت فازی را توسط الگوریتم FCGA بدست آورده و با مقایسه پاسخ‌های بدست آمده با پاسخ‌های مسأله در حالت غیرفازی، کارایی روش پیشنهادی خود را نشان دهیم. پس ما با در نظر گرفتن $n = 2$ یا دو متغیر ورودی مستقل برای توابع Schwefel و Rastrigin الگوریتم FCGA را بصورت زیر پیکربندی می‌نماییم.

ما برای هر کروموزوم ۶ ژن در نظر گرفته و تعداد جمعیت اولیه را برابر ۵۰۰ در نظر می‌گیریم. برای ارزیابی کروموزوم‌ها ابتدا ژن‌های هر کروموزوم را معادل rهای تابع آشوبگونه در نظر گرفته و به صورت گفته شده در بخش دوم به تولید مجموعه‌های آشوبگونه فازی به ازای هر متغیر مستقل ورودی می‌نماییم. سپس برای بدست آوردن $\overline{Y}_n$ از رابطهٔ ۵ برای تابع Schwefel و رابطه ۶ برای تابع Rastrigin استفاده می‌نماییم.

$$\overline{Y}_n = \sum_{i=1}^{2} \overline{X}_n . \sin(\sqrt{|\overline{X}_n|}) \qquad (۵)$$

$$\overline{Y}_n = 20 + \sum_{i=1}^{2} (\overline{X}_i^2 - 10.\cos(2.\pi.\overline{X}_i)) \qquad (۶)$$

سپس θ_n را برابر مرکز ثقل عدد فازی $\overline{Y}_n$ قرار می‌دهیم، در اینجا برای محاسبه θ_n از روش توزیع نسبی استفاده نموده و در حالت محدود نمودن اعداد فازی به اعداد مثلثی، مرکز ثقل از طریق رابطهٔ (۷) قابل محاسبه می‌باشد.

$$\theta_n = (a_{yn} + 2 * b_{yn} + c_{yn}) / 4 \qquad (۷)$$

در اینجا مسالهٔ ما پیدا کردن مینیمم $\overline{Y}_n = (a_{yn}, b_{yn}, c_{yn})$ است بطوریکه یکی از این $\overline{Y}_n$ها کوچکترین مرکز ثقل را داشته باشد. تا مرحله $n = 20$، $\overline{Y}_i$، $0 \le i \le n$ که کوچکترین مرکز ثقل را داراست بجای مقدار $\overline{Y}^*$ قرار می‌دهیم. در هر مرحله از الگوریتم اگر θ_{n+1} کوچکتر یا مساوی با مرکز ثقل $\overline{Y}^*$ باشد، مقدار $\overline{Y}^*$ برابر $\overline{Y}_{n+1}$ خواهد بود، در غیر اینصورت در $\overline{Y}^*$ تغییری حاصل نمی‌شود. درنهایت بهترین $\overline{Y}^*$ پاسخ تابع ارزیابی ما می‌باشد.

در ادامه ۱۰ درصد از بهترین کروموزومها را به جمعیت جدید منتقل می‌نماییم و برای تولید سایر کروموزوم‌های نسل جدید، بعد از انتخاب تصادفی دو کروموزوم با استفاده از ترکیب دو نقطه‌ای به ایجاد فرزندان می‌پردازیم و اینکار را تا تولید کامل نسل جدید ادامه می‌دهیم. نرخ جهش را پنج درصد انتخاب نموده و با درنظر گرفتن S=۱۰ و مادامیکه در ده تکرار پیاپی در بهترین پاسخ، بهبودی حاصل نگردد مراحل را تکرار می‌نماییم.

نتایج بدست آمده از حل مسألهٔ (۳) با استفاده از تنظیمات فوق برای روش FCGA در جدول (۱) و نتایج حاصل از مسأله (۴) در جدول (۲) نمایش داده شده‌اند.

جدول ۱ نتایج بدست آمده از حل مسألهٔ (۳) با استفاده از روش FCGA

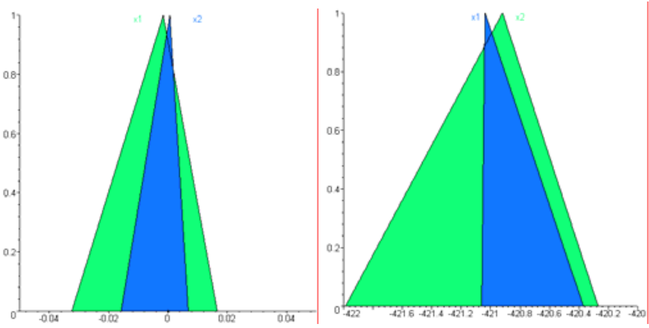
θ	c^*	b^*	a^*	
-۴۲۰/۸۷۷۵	-۴۲۰/۳۶۹۳	-۴۲۱/۰۳۹۴	-۴۲۱/۰۶۱۹	$\overline{X}_1^*$
-۴۲۱/۰۲۴۳	-۴۲۰/۲۶۸۹	-۴۲۰/۹۱۹۷	-۴۲۱/۹۸۸۹	$\overline{X}_2^*$
-۸۳۷/۸۸۹۹	-۸۳۷/۷۸۹۱	-۸۳۷/۹۰۲۹	-۸۳۷/۹۶۴۸	$\overline{Y}^*$

جدول۲ نتایج بدست آمده از حل مسألهٔ (۴) با استفاده از روش FCGA

θ	c^*	b^*	a^*	
-۰/۰۰۴۷۸۳	۰/۰۱۶۵۹۲	-۰/۰۰۱۶۳۰	-۰/۰۳۲۴۶۶	$\overline{X}_1^*$

$\overline{X}_2^*$	−۰/۰۱۵۸۳۳	۰/۰۰۰۶۰۷	۰/۰۰۶۷۹۹	−۰/۰۰۱۹۵۴
$\overline{Y}^*$	۰/۰۰۰۶۰۲	۰/۰۶۳۷۴۴	۰/۲۵۸۰۹۱	۰/۰۹۶۵۴۳

اعداد فازی $\overline{X}_1^*$ و $\overline{X}_2^*$ از جداول ۱ و ۲ به ترتیب در شکل‌های (۲-الف) و (۲-ب) نمایش داده شده‌اند. همانگونه کـه مشاهده می‌گردد، اعداد فازی بدست آمده از روش FCGA، پاسخ‌های مسأله در حالت قطعی را پوشش می‌دهد. از طرفـی مرکز ثقل اعداد فازی $\overline{X}_1^*$ و $\overline{X}_2^*$ نیز مقادیری نزدیک به پاسخ مسأله در حالت قطعی را نمایش می‌دهند. لذا با استفاده از روش FCGA می‌توان پاسخ تقریباً دقیقی برای توابع پیچیده بدست آورد.



(ب) (الف)

شکل ۲-الف- ورودیهای بدست آمده از روش FCGA جهت حل مسألهٔ Schwefel و ۲-ب برای مساله Rastring

۶ - خلاصه و نتیجه‌گیری

در این مقاله ابتدا به بحث در مورد تولید آشوبگونه مجموعه‌های فازی پرداخته، سپس الگوریتم ژنتیک آشوبگونه فازی را توضیح داده و روش بدست آوردن مینیمم واقعی توابع پیچیده با استفاده از این روش را بیان نمودیم. سپس با مقایسـه نتـایج بدسـت آمـده از روش پیشنهادی با پاسخ‌های مسأله در حالت غیرفازی به این نتیجه رسیدیم که روش پیشنهادی، روشی کارآمد جهت بدست آوردن مینیمم واقعی توابع پیچیده می‌باشد.

از آنجا که دقت پاسخ تولیدی و سرعت رسیدن به آن در روش جستجوی آشوب فازی وابسته به مقادیر انتخابی بـرای نـرخ رشـد تـابع آشوبگونه می‌باشد، پس انتخاب مقادیر بهینه با روش‌های مختلف بهینه‌سازی و مقایسه نتایج بدست آمده از آنها می‌تواند تحقیقی برای آینده باشد.

مراجع

[1] Zhou, C., Chen, T., ”Chaotic annealing for optimisation.”, Phys Rev E; vol. 55(3), p.p. 2580–7, 1997.

[2] Li, B., Jiang, W., “Optimizing complex functions by chaos search.”, Int J Cybernet Syst”, vol. 29(4), p.p. 409–19, 1998.

[3] Lu, Z., Shieh, L.S., Chen, G. , Coleman, N.P., “Simplex sliding mode control form on linear uncertain systems via chaos optimization.”, Chaos, Solitons & Fractals, vol. 23, p.p. 747–55,2005.

[4] Hasegawa, M. , Ikeguchi, T. ,Aihara, K. , Itoh, K., “A novel chaotic search for quadratic as signment problems.”, Eur J Oper Res, vol. 534, p.p. 56-139, 2002.

[5] Ji, M.J. , Tang, H.W., “Application of chaos in simulated annealing.”, Chaos, Solitons & Fractals, vol. 21, p.p. 33-41, 2004.

[6] Buckley, J.J.; Hayashi, Y.; “Fuzzy genetic algorithm and application”, Fuzzy Sets Syst.,vol. 61, pp. 129-146, 1994.

[7] Buckley, J.; Hauashi, Y.; “Applications of fuzzy chaos to fuzzy simulation”, Fuzzy sets and systems, Vol. 99, pp. 151-157, 1998.

[8] Li ,C.; Liao, X.; Ya, J.; “Tabu search for fuzzy optimization and applications”, Information Sciences, Vol. 158, pp. 3-13, 2004.

[۹] فرهادی، محسن، جاهدمطلق، مزینی، رحیم‌اف، "پیاده سازی جستجوی آشوب گونه اصلاح شده به منظور بهینه سازی توابع پیچیده"، سومین کنفرانس بین‌المللی تحقیق در عملیات،دانشگاه صنعتی امیرکبیر، ۱۳۸۹.

[۱۰] فرهادی، محسن، جاهدمطلق، مزینی، رحیم‌اف، "بهینه‌سازی تابع هدف خطی با استفاده از جستجوی آشوب فازی و الگوریتم ژنتیک"، سومین کنفرانس بین‌المللی تحقیق در عملیات، دانشگاه مازندران، ۱۳۸۸.

[۱۱] فرهادی، محسن، جاهدمطلق، مزینی، رحیم‌اف، "حل مسائل رگرسیون فازی با استفاده از الگوریتم ژنتیک آشوبگونه فازی"، سومین کنفرانس داده‌کاوی، دانشگاه علم و صنعت ایران، ۱۳۸۸.