پنجمین کنفرانس سیستمهای فازی ایران
هفدهم الی نوزدهم شهریورماه ۱۳۸۳

بهینه سازی توابع مقید به کمک الگوریتم های ژنتیک و تابع جریمه فازی

محمدحسن شرقی‌گورابی[۱]، محمدابراهیم محمدپوررزندی[۲]، سیدمصطفی رضوی[۳]
مرکز تحقیقات مخابرات ایران
sharghi@itrc.ac.ir

چکیده
بسیاری از مسایل بهینه سازی که بصورت غیر خطی هستند، با روشهای کلاسیک بهینه سازی قابل حل می باشند، اما زمانیکه تعداد متغیرها و محدودیتها زیاد می شوند و شرایط عدم اطمینان نیز به آنها اضافه می گردد، روشهای کلاسیک کارایی خود را از دست می دهند. الگوریتمها و روشهای متعددی برای حل اینگونه مسایل در روند تکاملی ریاضیات پیشنهاد شده است. یکی از این روشها، استفاده از الگوریتمهای تکاملی است. اما مشکل قابل توجهی که در این روش وجود دارد، نحوه برخورد با فرزندان غیرقابل قبول است. روشهای متفاوتی برای حل این مشکل وجود دارد که یکی از آنها استفاده از تابع جریمه فازی است. در این مقاله ضمن معرفی روشهای کلاسیک بهینه سازی توابع مقید (معادله و نامعادله)، تابع جریمه فازی و ویژگیهای آن را با استفاده از سیستم فازی تعیین و سپس به کمک الگوریتم ژنتیک به حل نمونه آن پرداخته ایم.

واژه های کلیدی: برنامه ریزی غیر خطی- الگوریتم ژنتیک- سیستم های فازی- تابع جریمه فازی

مقدمه

در تدوین بسیاری از مسایل بهینه سازی نمی توان فرض خطی بودن، را صادق دانست. از سوی دیگر رویه های عمومی برای حل مسایل غیر خطی وجود ندارد. البته تعدادی الگوریتم های خاص برای حل موارد ویژه ایجاد شده است. بسیاری از این الگوریتمها بر اساس یک نظریه ریاضی مرتبط با ساختار اینگونه مسایل طراحی شده اند. این نظریه را معمولا بهینه سازی کلاسیک(Classical Optimization) گویند.

یکی از مهمترین دستاوردها در زمینه این نظریه توسط دو ریاضیدان بنامهای Kuhn- Tucker (۱۹۵۱) انجام گرفته و به شرایط کان- تاکر مشهور است.

مجموعه فنونی که از این نظریه به وجود آمده‌اند را برنامه ریزی غیر خطی می نامند. علی‌رغم این حقیقت که حل تعداد زیادی از مسایل برنامه ریزی غیر خطی بسیار مشکل است ولی مسایل عملی متعددی را می‌توان بصورت

۱- کارشناس ارشد، مرکز تحقیقات مخابرات ایران
۲- استادیار، دانشگاه آزاد اسلامی واحد مرکزی تهران
۳- استادیار، دانشکده مدیریت، دانشگاه تهران

غیر خطی تدوین و با روشهای موجود حل کرد. بنابراین نظریه بهینه سازی که ماهیتا یک نظریه ریاضی است، معمولا شامل به حداکثر یا حداقل رساندن یک تابع(بعضا نامعلوم) است که بیانگر عملکرد یک سیستم می باشد.
بعضی از مسایل بهینه سازی را می توان از طریق فنون کلاسیک حساب پیشرفته از قبیل روشهای ژاکوبی و ضرایب لاگرانژ حل کرد. اما بسیاری از این مسایل شرایط لازم برای حل با این روشها را ندارند. اینگونه مسایل را با روشهای کاراتری باید حل کرد. خصوصا آنکه شرایط عدم اطمینان و یا متغیرهای غیر کمی نیز به مدل اضافه شوند.
در روند تکامل ریاضیات، مجموعه ای از این روشهای ویژه ایجاد شده است که بعضا به فراموشی سپرده شده و برخی نیز با اختراع و کاربرد وسیع کامپیوترها از لحاظ محاسباتی و سرعت امکانپذیر گشته و مورد استفاده قرار گرفته اند.

۱- بهینه سازی کلاسیک توابع مقید

تابع $z = f(x_1, x_2, ..., x_n)$ مفروض است. اگر بخواهیم ماکزیمم یا مینیمم این تابع را با فرض اینکه بین متغیرهای مستقل رابطه $g(x_1, x_2, ..., x_n) = c$ برقرار باشد در اینصورت تابع z را تابع مقید و تابع g را قید یا شرط تابع می خوانند. این توابع بر دو نوع قابل تقسیم هستند:

الف) توابع با قیود(محدودیتهای) مساوی (معادله ای)

توابع با قیود (محدودیتهای) نامساوی (نامعادله ای)

در حالت الف، تابع n متغیره $z = f(x_1, x_2, ..., x_n)$ که m محدودیت معادله زیر را دارد، در نظر می گیریم:

$$\begin{aligned}
&z = f(x_1, x_2, ..., x_n) \\
&s.t.: \\
&g_1(x_1, x_2, ..., x_n) = c_1 \\
&g_2(x_1, x_2, ..., x_n) = c_2 \\
&\vdots \\
&g_m(x_1, x_2, ..., x_n) = c_m
\end{aligned} \tag{1}$$

که در این رابطه cها ثابت و $m<n$ است.

برای بهینه سازی آن(پیدا کردن ماکزیمم، مینیمم) از روش ضرایب(نامعین) لاگرانژ استفاده می کنیم یعنی:

$$L = f(X) - \lambda_1[g_1(X) - c_1] - \lambda_2[g_2(X) - c_2] - \cdots - \lambda_m[g_m(X) - c_m] \tag{2}$$

که در آن λها ضرایب لاگرانژ بوده و $X = (x_1, x_2, ..., x_n)$ می باشد.

$$\begin{aligned}
&\frac{\partial L}{\partial x_1} = 0 \\
&\vdots \\
&\frac{\partial L}{\partial x_n} = 0 \\
&\frac{\partial L}{\partial \lambda_1} = 0 \\
&\vdots \\
&\frac{\partial L}{\partial \lambda_m} = 0
\end{aligned} \tag{3}$$

با حل این دستگاه که مشتمل بر $m+n$ معادله و $m+n$ مجهول است مقادیر x و λها تعیین می شود و برای تعیین نوع نقطه از مشتقات مرتبه بالاتر و ماتریس هشین $|H|$ استفاده کرده و نوع تابع را مشخص می کنیم.

در حالت (ب)، تابع z با قیود نامعادله ای، مشروط شده است یعنی:

$$z = f(x_1, x_2, ..., x_n)$$

s.t:

$$\begin{aligned} & g_1(x_1, x_2, ..., x_n) \le c_1 \\ & g_2(x_1, x_2, ..., x_n) \le c_2 \\ & \vdots \\ & g_m(x_1, x_2, ..., x_n) \le c_m \end{aligned} \tag{۴}$$

در این حالت با اضافه کردن متغیرهای کمکی به نامعادلات از شرایط کان- تاکر استفاده می کنیم:

$$F = f(X) - \lambda_1\left[g_1(X) + s_1^2 - c_1\right] - \lambda_2\left[g_2(X) + s_2^2 - c_2\right] - \cdots - \lambda_m\left[g_m(X) + s_m^2 - c_m\right] \tag{۵}$$

آنگاه دستگاه زیر را خواهیم داشت:

(۶)

$$\begin{aligned} & \frac{\partial F}{\partial x_1} = 0 \\ & \vdots \\ & \frac{\partial F}{\partial x_n} = 0 \\ & \frac{\partial F}{\partial \lambda_1} = 0 \\ & \vdots \\ & \frac{\partial F}{\partial \lambda_m} = 0 \\ & \frac{\partial F}{\partial s_1} = 0 \\ & \vdots \\ & \frac{\partial F}{\partial s_m} = 0 \end{aligned}$$

۲- استراتژی جریمه

یکی از روشهای جدید برای بهینه سازی توابع مقید استفاده از الگوریتم های تکاملی می باشد. اما مشکلی که در این روش وجود دارد نحوه برخورد با فرزندان غیر قابل قبول می باشد. روش های متفاوتی برای حل این مشکل وجود دارد. یکی از این روشها استفاده از استراتژی جریمه است.

استراتژی جریمه از فرزندان غیر قابل قبول در جستجوی ژنتیک استفاده می کند. در استراتژی جریمه، با جریمه کردن جوابهای غیر قابل قبول، مساله محدود به یک مساله نامحدود تبدیل می شود، به طوریکه به ازای هر تخلفی از حدود، عبارت جریمه به تابع هدف اضافه می گردد. در این حالت جستجوی ژنتیک از هر دو ناحیه موجه و غیر موجه به سوی جواب بهینه حرکت می کند.

۳- ترکیب تابع هدف با عبارت جریمه

دو روش برای ترکیب تابع هدف با عبارت جریمه وجود دارد:

روش اول استفاده از عمل جمع می باشد. اگر فرض کنیم که *f(x)* تابع هدف و *p(x)* تابع جریمه باشد آنگاه خواهیم داشت :
$$eval(x) = f(x) + p(x)$$

به طوریکه برای مسایل بیشینه سازی خواهیم داشت:

$p(x) = 0$ ؛ اگر x کروموزم موجه است.

$p(x) <0$ ؛ اگر x کروموزم غیر موجه است.

و همینطور برای مسایل کمینه سازی خواهیم داشت:

$p(x) = 0$ ؛ اگر x کروموزم موجه است.

$p(x) >0$ ؛ اگر x کروموزم غیر موجه است.

روش دوم استفاده از عمل ضرب می باشد. به طوریکه در این روش خواهیم داشت:

$$eval(x) = f(x).p(x)$$

که برای مسایل بیشینه سازی : $p(x) = 1$ ؛ اگر x کروموزم موجه باشد.

$0<p(x) <1$ ؛ اگر x کروموزم غیر موجه باشد.

و برای مسایل کمینه سازی : $p(x) = 1$ ؛ اگر x کروموزم موجه باشد.

$p(x) >1$ ؛ اگر x کروموزم غیر موجه باشد.

۴- معرفی چند تابع جریمه

۵-۱: *Homaifar, Qi & Lai* : در این روش مساله برنامه ریزی غیر خطی به شکل زیر است:

$$Min \;\; f(x) \tag{۷}$$

$$s.t \qquad g_i(x) \ge 0 \quad , i= 1,2,...,m$$

تابع $eval(x) = f(x) + p(x)$ برای ارزیابی در نظر گرفته شده و تابع جریمه نیز به شکل زیر است:

اگر x کروموزم قابل قبول باشد: $p(x) = 0$

در غیر اینصورت:

$$p(x) = \sum_{i=1}^{m} r_i g_i^2(x) \tag{۸}$$

بطوریکه r_i ضریب جریمه برای محدودیت i ام و $g_i(x)$ مقدار محدودیت i ام برای کروموزم x می باشد. انتخاب مقادیر مناسب برای r_i کار ساده ای نیست و به مساله وابسته است.

۵-۲: *Houck & Joines* : در این روش مساله برنامه ریزی غیر خطی به شکل زیر می باشد:

$$Min \qquad f(x) \tag{۹}$$

$$s.t \qquad g_i(x) \ge 0 \quad , i= 1,2,...,m_1$$

$$h_i(x) = 0 \quad , i=m_1+1,m_1+2,...,m$$

تابع $eval(x) = f(x) + p(x)$ برای ارزیابی در نظر گرفته شده و تابع جریمه نیز به شکل زیر است:

$$p(x) = \rho_t^{\alpha} \sum_{i=1}^{m} d_i^{\beta}(x) \tag{۱۰}$$

t تکرار الگوریتم ژنتیک است. α و β پارامترهایی هستند که برای تعدیل جریمه بکار می روند. جریمه برای محدودیت iام به شکل زیر محاسبه می شود:

اگر x کروموزم قابل قبول باشد , $d_i(x) = 0$

$$d_i(x) = |g_i(x)| \qquad , \quad 1<i<m_1$$
$$d_i(x) = |h_i(x)| \qquad , \quad i=m_1+1,m_1+2,...,m$$

$\rho = ct$ بطوریکه c مقدار ثابتی است. جواب بهینه به سه پارامتر α و βو ρ بسیار حساس است. تعیین این سه پارامتر کاملا وابسته به مساله می باشد.

۵-۳: *Attia & Michalwics* : در این روش مشخصات مساله برنامه ریزی غیر خطی مثل روش قبل می باشد. اما تابع جریمه به شکل زیر تعریف می شود:

$$p(x) = \frac{1}{2T}\sum_{i\in A} d_i^2(x) \tag{۱۱}$$

A مجموعه ای از محدودیتهای فعال است. محدودیتهای فعال محدودیتهایی هستند که مقدار آنها به ازای کروموزم x بزرگتر از مقدار δ باشد. δ پارامتری است که نشان می دهد محدودیت فعال است. *T* هم جزء متغیر جریمه است. در این روش *$d_i(x)$* به روش زیر محاسبه می شود:

$$d_i(x) = \max\{0, g_i(x)\} \qquad ; \ 1<i<m_1$$
$$d_i(x) = |h_i(x)| \qquad ; \ m_1+1 < i < m$$

این روش به مقادیر پارامترها بسیار حساس است و چگونگی تعیین این پارامترها برای یک مساله کاملا اختیاری می باشد.

۵-۴: *Smith,Tate &* Coit : تابع جریمه تطبیقی اولین بار توسط *Smith, Tate* پیشنهاد شد. در این تابع مقدار جریمه به طور پویا و مقایسه کردن با بهترین جواب تاکنون بدست آمده ، تغییر می نماید.

مفهوم *NFT(Near - Feasibility Thershold)* : فاصله آستانه ای از منطقه موجه است که از آن نقطه به بعد کاربر احساس می کند که جستجوی ژنتیک به جواب نزدیک می شود.

تابع جریمه، الگوریتم ژنتیک را تشویق می کند در حوزه منطقه موجه و همسایگی *NFT* منطقه موجه جستجو کند و آن را از جستجوی ورای این آستانه باز می دارد. تابع $eval(x) = f(x) + p(x)$ برای ارزیابی بکار می رود و تابع جریمه به شکل زیر تعریف می گردد:

$$p(x) = -\sum_{i=1}^{m}\left(\frac{\Delta b_i(x)}{\Delta b_i^{nef}}\right)^{\alpha}\left(f_{all}^* - f_{feas}^*\right) \tag{۱۲}$$

α پارامتری است که برای تعدیل دقت بکار می رود. $\Delta b_i(x)$ میزان تخلف از محدودیت i ام و Δb_i^{nef} نیز برابر با *NFT* محدودیت i می باشد. انتخاب مقدار مناسب برای *nef* به مساله وابسته است.

f^*_{feas} : ارزش تابع هدف بهترین جواب موجهی است، که تاکنون پیدا شده است.

f^*_{all} : مقدار تابع هدف جریمه نشده است، از بهترین جوابی که تاکنون پیدا شده است.

۵-۵: *Yakota , Gen , Ida & Taguchi* : در این روش، مساله برنامه ریزی غیر خطی مشابه روش قبل در نظر گرفته شده اما برای ارزیابی تابع *eval(x) = f(x) . p(x)* بکار رفته است. تابع جریمه به صورت زیر ساخته می شود:

$$p(x) = 1 - \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\Delta b_i(x)}{b_i}\right)^{\alpha} \quad , \quad \Delta b_i(x) = Max\{0, g_i(x) - b_i\} \tag{۱۳}$$

$\Delta b_i(x)$ میزان تخلف از محدودیت i ام است. این تابع جریمه را می توان به عنوان یک حالت خاص روش قبل در نظر گرفت با این تفاوت که $\Delta b_i^{nef} = b_i$ است. بنابراین جریمه ای که بدست می آید تا حدی متعادلتر از روش قبل است.

۵-۶: *Gen & Cheng* : در این روش اگر *x* کروموزمی در جمعیت فعلی باشد، تابع جریمه زیر بکار گرفته می شود:

$$p(x) = 1 - \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\Delta b_i(x)}{\Delta b_i^{\max}}\right)^{\alpha} \tag{۱۴}$$

$$\Delta b_i(x) = Max\{0, g_i(x) - b_i\} \;,\quad \Delta b_i^{\max} = Max\{\varepsilon, \Delta b_i(x)\}$$

$\Delta b_i(x)$ میزان تخلف از محدودیت i مربوط به کروموزم x است. $\Delta b_i^{\max}$ حداکثر تخلف از محدودیت i در جمعیت فعلی می باشد و ε عدد مثبت کوچکی است که برای اجتناب از تقسیم بر صفر شدن جریمه در نظر گرفته شده است.

۵- تابع جریمه فازی

در اینجا تابع جریمه ای که پارامترهای آن با استفاده از سیستم فازی تعیین می شوند را ارایه می دهیم و مساله را نیز به شکل زیر در نظر می گیریم:

$$\begin{aligned} & Min \quad f(x) \\ s.t.\; & g_j(x) \le 0 \;;\; j=1,\dots,m \\ & h_L(x) = 0 \;;\; L=1,\dots,k \\ & a(i) \le x_i \le b(i) \quad i=1,\dots,n \end{aligned} \qquad (۱۵)$$

a(i),b(i) مرزهای فضای جستجو می باشند. با اعمال تابع جریمه به مساله با محدودیت، مساله به یک مساله بهینه سازی بدون محدودیت تبدیل می شود که تابع هدف جدید *F(x)* به شکل زیر است:

$F(x) = f(x)$; اگر x یک کروموزم قابل قبول باشد.

$F(x) = f(x) + rG(x)$; اگر x یک کروموزم غیر قابل قبول باشد.

بطوریکه : $G(x) = \sum_{j=1}^{m} g_j(x) + \sum_{l=1}^{k} |h_l(x)|$ و r نیز ضریب جریمه می باشد.

F(x) تابع هدف جدید یا همان تابع *fitness* ما خواهد بود. تعیین مقدار r کار راحتی نیست و کاملا به مساله وابسته است. ما اکنون سیستم فازی را توضیح می دهیم که می تواند مقدار بهینه ای برای r در هر نسل پیدا کند.

فرض می شود *f(x),G(x)* متغیر های فازی ورودی باشند که آنها را با f,g نشان می دهیم. از طرفی فرض می شود که دامنه تغییرات f، $[f_{min}, f_{max}]$ و دامنه تغییرات g ، $[0, G_{\max}]$ باشد. بطوریکه:

$$G_{max} = \mathrm{Max}\, G_l^{(t)}(x) \quad ، \quad f_{max} = \mathrm{Max}\; f_l^{(t)}(x) \quad ، \quad f_{min} = \mathrm{Min}\; f_l^{(t)}(x)$$

t شماره نسل فعلی و l شماره کروموزم در نسل فعلی می باشد. r را متغیر فازی خروجی در نظر می گیریم که دامنه تغییرات آن بازه $[0, r_{\max}]$ می باشد و $r_{max} = f_{max} - f_{min}$.

برای هر یک از متغیرهای فازی ورودی سه مجموعه فازی **کوچک**، **متوسط**، **بزرگ** و برای متغیر فازی خروجی نیز پنج مجموعه فازی **خیلی کوچک**، **کوچک**، **متوسط**، **بزرگ**، **خیلی بزرگ** را در نظر می گیریم. اکنون قوانین را به شکل زیر می‌توان تعریف کرد:

اگر f و g ***بزرگ*** *هستند آنگاه r* ***خیلی کوچک*** *است.*

اگر f ***بزرگ*** *و g* ***متوسط*** *است آنگاه r* ***کوچک*** *است.*

اگر f ***بزرگ*** *و g* ***کوچک*** *است آنگاه r* ***متوسط*** *است.*

اگر f ***متوسط*** *و g* ***بزرگ*** *است آنگاه r* ***کوچک*** *است.*

اگر f و g ***متوسط*** *هستند آنگاه r* ***متوسط*** *است.*

اگر f ***متوسط*** *و g* ***کوچک*** *است آنگاه r* ***بزرگ*** *است.*

اگر f ***کوچک*** *و g* ***بزرگ*** *است آنگاه r* ***متوسط*** *است.*

اگر f ***کوچک*** *و g* ***متوسط*** *است آنگاه r* ***بزرگ*** *است.*

اگر f و g ***کوچک*** *هستند آنگاه r* ***خیلی بزرگ*** *است.*

از سیستم فازی با فازی ساز منفرد و موتور استنتاج ضرب و غیر فازی ساز میانگین مراکز استفاده می شود. اکنون الگوریتمی برای بهینه سازی با محدودیت به کمک الگوریتمهای ژنتیک و تابع جریمه فازی به شکل زیر می توان ارایه نمود:

ابتدا یک جمعیت اولیه در نظر می گیریم که کروموزم های آن مقدار دهی اولیه شده باشند. بطوریکه مقدار اولیه هر ژن در کروموزم به شکل مقابل باشد: $x_i^j \in (a^k, b^k)$, j=1,...,k

برای هر کروموزم مقدار تابع هدف *f(x)* و مقدار تابع محدودیت *G(x)* را محاسبه می کنیم.

مقدار اولیه جریمه فازی بر مبنای تعداد مجموعه فازی و قوانین تعیین می شود.

برای هر کروموزم مقدار تابع هدف جدید *F(x)* محاسبه می شود.

عملگرهای ژنتیک ، *Cross Over* و *Mutation* را بر روی جمعیت فعلی اعمال می کنیم.

مقادیر *f(x)* و *G(X)* را برای فرزندان متولد شده ، بدست می آوریم.

از بین والدین و فرزندان تعدادی را انتخاب می کنیم.

برای جمعیت فعلی الگوریتم فوق را از گام دوم تکرار می کنیم.

مراحل فوق را تکرار می کنیم تا به تعداد نسل مورد نظر یا اینکه به یک دقت از پیش تعیین شده برسیم.

۶- شبیه سازی

الگوریتم فوق را بر روی مساله زیر پیاده سازی نمودیم:

$$Min \quad f(x) = (x_1 - 2)^2 + (x^2 - 1)^2 \tag{۱۶}$$

subject to:

$$g_1(x) = x_1 - 2x^2 + 1 = 0$$

$$g_2(x) = -\frac{{x_1}^2}{4} + x_2^2 - 1 \le 0$$

برای پیاده سازی سیستم فازی از Fuzzy logic toolbox نرم افزار Matlab استفاده شده است. بطوریکه پارامترهای آن برای هر نسل از داخل برنامه تنظیم می شود.

با اتخاد نرخ جهش 0.8، نرخ *cross over* ، 0.5 و تعداد کروموزمها ۵۰ عدد و تعداد ۱۰۰ نسل، نتایج زیر بدست آمده است:

$$x_1 = 0.95 \qquad x_2 = 0.62 \qquad f(x) = 1.23 \qquad g_1(x) = 0.7 \qquad g_2(x) = 0.83$$

برای همین مساله که با الگوریتم ژنتیک و با استفاده از روش تابع جریمه *Homaifar, Qi & Lai* نیز حل شده، نتایج زیر بدست آمده است:

$$x_1 = 0.8 \qquad x_2 = 0.88 \qquad f(x) = 1.43 \qquad g_1(x) = 0.37 \qquad g_2(x) = 0.52$$

۷- نتیجه گیری

روشهای کلاسیک برای بهینه سازی توابع مقید هنگامی که با مسایلی مواجه می شوند که تعداد متغیرها و محدودیتها زیاد است، کارایی‌شان را از دست می دهند. از اینرو روشهای جدیدی که مبتنی بر الگوریتمهای ژنتیک هستند، ارایه شده است. اما در روش الگوریتمهای ژنتیک نحوه برخورد با فرزندان غیر قابل قبول بسیار مهم است. روشهایی برای برخورد با فرزندان غیر قابل قبول وجود دارد که یکی از آنها استفاده از تابع جریمه است. در این مقاله علاوه بر بررسی مختصر روشهای موجود برای پیاده سازی تابع جریمه، روشی که مبتنی بر سیستمهای فازی است بطور مفصل مورد بررسی قرار گرفته و الگوریتم ژنتیکی مبتنی بر آن ارایه و پیاده سازی شده است.

مراجع

[1] Mitsuo Gen and Runwei Cheng , (1996) , "Genetic Algorithms & Engineering Design" , Wiley.
[2] Rita Almeida Ribeiro, Fernando Moura Pires, (1999), Fuzzy Linear Programming via Simulated Annealing,, Universidade Nova Lisboa.
[3] Milan Cisty, Advantages of using genetic algorithms over deterministic methods in optimal design of the water networks rehabilitation, Proceedings of ALGORITMY 2000 Conference on Scientific Computing, pp. 293 – 300.
[4] Kalyanmoy Deb and Samir Agrawal, (1997), A Niched-Penalty Approach for Constraint Handling in Genetic Algorithms , Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering, Indian Institute of Technology Kanpur, PIN 208 016, India.
[5] Rakesh Angira1 & B. V. Babu, (2002), evolutionary computation for global optimization of non-linear chemical engineering processes , Department of Chemical Engineering Birla Institute of Technology & Science Pilani - 333 031(Rajasthan) India.
[۶] د.اینتریلییگیتور، میشل، ترجمه حسین علی پور کاظمی، بهینه سازی ریاضی. انتشارات دانشگاه شهید بهشتی.
[۷] لی وانگ. ترجمه محمد تشنه لب- نیما صفارپور- داریوش افیونی، (۱۳۷۸)، سیستمهای فازی و کنترل فازی. ، انتشارات دانشگاه صنعتی خواجه نصیرالدین طوسی.
[۸] بوجادزیف، جرج و بوجادزیف، ماریا، ترجمه سید محمد حسینی، (۱۳۸۱)، منطق فازی و کاربردهای آن در مدیریت، انتشارات ایشیق.