بهینه سازی چند هدفه با توجه به معادلات رابط فازی

حسن زارعی[i]؛ اسماعیل خرّم[ii]؛صادق رضائی[iii]

چکیده

الگوریتم ژنتیک یکی از الگوریتم‌های فوق ابتکاری است که به‌منظورحل مسایل بهینه سازی مقیّد و نا مقیّد به‌کار می‌رود. این الگوریتم به‌طور تکراری دنباله ای از عملگرهای انتخاب و تکثیر[1]، جهش[2] و تقاطع[3] روی جمعیت اولیه اجرا کرده و به‌جستجوی جواب بهینه می‌پردازد.
این مقاله حل یک مسأله بهینه سازی چند هدفه با توجه به معادلات رابط فازی با ترکیب *max-product* را ارایه می‌دهد. همچنین الگوریتم ژنتیک مورد بازبینی قرار گرفته و برخی از مولفه‌های آن برای حل مسأله تغییر داده می‌شود. افزون بر آن روشی برای کاهش بعد مسأله پیشنهاد می‌شود. در آخر، اجرای این الگوریتم و تاثیر قسمتهای اصلی آن روی مجموعه جواب شدنی انواع مختلف مسایل بررسی می‌شود.

کلمات کلیدی

بهینه سازی چند هدفه؛ معادلات رابط فازی؛ ترکیب max-product؛ الگوریتم ژنتیک.

Multi-purpose optimization regarding constraints of "fuzzy relation equation"

Hassan Zarei; Esmaile. Khorram; Sadegh Rezaei

ABSTRACT

Genetic algorithm is one of the metaheuristic algorithms used for solving constrained and unconstrained optimization problems. This algorithm performs a sequence of Selection/Reproduction, mutation and crossover operators on initial population iteratively and searches the optimal solution.
In this paper, a multiple objective optimization model subjected to a system of fuzzy relation equations with max-product composition and a reduction procedure to reduce problem dimension is presented. Furthermore, a genetic algorithm has been reviewed and some of its component modified to solve the problem. Finally, the algorithm is performed on different problems and its major components role in detecting solution is examined.

KEYWORDS

Multi –objective optimization; fuzzy relation equations; max-product composition; genetic algorithm.

۱- مقدمه

معادلات رابط فازی تعمیم رابط دوتایی[4] [۴] است. برای یک سیستم تعریف شده توسط معادلات رابط فازی پارامترهای ورودی و خروجی به گونه‌ای مرتبط هستند. تصمیم گیرنده قصد دارد تا پارامترهای ورودی را در سطوح خاصی به نحوی قرار دهد، که چند تابع هدف به طور همزمان بهینه شوند. چنین مسأله‌ای را می‌توان به صورت بهینه سازی چند هدفه با توجه به معادلات رابط فازی در نظر گرفت.
ایده معادلات رابط فازی بر پایه ترکیب max-min ابتدا توسط Sanchez مطرح گردید [۱۰] . وی بخشی از قضایا ی

[i] دانشجوی کارشناسی ارشد ریاضی کاربردی؛دانشگاه صنعتی امیر کبیر؛ دانشکده ریاضی و علوم کامپیوتر:hssn_zr@yahoo.com .
[ii] دانشیاردانشگاه صنعتی امیر کبیر؛ دانشکده ریاضی و علوم کامپیوتر:eskhor@aut.ac.ir .
[iii] استادیار دانشگاه شهید چمران اهواز؛ دانشکده ریاضی و علوم کامپیوتر: srezaei@aut.ac.ir .

تعیین وجود جواب‌های بعضی از معادلات رابط فازی خاص را ارایه داده است. علاوه بر آن، محققان بسیاری درشناسایی مسأله و ارایه راه حل‌های مختلف تلاش کرده‌اند.

ترکیب max-min [۱۴] عموما زمانی استفاده می‌شود که یک سیستم احتیاج به جواب‌هایی با این خصوصیت دارد، که خوبی یک جواب نتواند بدی جواب دیگر را جبران کند. در واقع، موقعیت هایی وجود دارد که اجازه داده می‌شود بدی یک جواب را بتوان با بهبود بخشیدن جواب دیگر جبران کرد. در این موقعیت‌ها عملگر max-min چندان مناسب نیست، بلکه در عوض عملگرmax-product نتایج بهتری می‌دهد [۱۲] -[۱۴]-[۱۵]. برخی از معیارهای انتخاب یک عملگر مناسب در موقعیت‌های مختلف توسط Yager ارایه گردیده است [۱۳].

در این مقاله یک سیستم معادلات رابط فازی به شکل رابطه (۱) ارایه می‌شود:

$$xOA = b \qquad (۱)$$

که در آن $A=[a_{ij}], 0 \le a_{ij} \le 1$ یک ماتریس $m \times n$ بعدی و $b=(b_1,\ldots,b_n), 0 \le b_j \le 1$ یک بردار n بعدی می‌باشد و O معرف ترکیب max-product است. مجموعه جواب سیستم (۲) مجموعه تمام بردارهای $x=(x_1,x_2,\ldots,x_m) \in [0,1]^m$ می‌باشد که در رابطه (۲) صدق می‌کنند:

$$\max_{1 \le i \le m}(x_i . a_{ij}) = b_j, \; j = 1,\ldots,n \qquad (۲)$$

Bourke وFisher [۱] حل سیستم معادلات رابط فازی با ترکیب max-product را مطالعه و مجموعه جواب‌های این سیستم و شرایط وجود جواب را ارایه داده‌اند. آنها نشان دادند که این مجموعه جواب می‌تواند به طور کامل توسط یک جواب ماکزیمم و یک تعداد از جواب‌های مینیمال تعین گردد. Li و Fang [۲] مسأله بهینه سازی یک هد فه با توجه به معادلات رابط فازی با ترکیب max-min و تابع هدف خطی را مطالعه کرده اند. تعمیم این مسأله به حالت غیر خطی توسط Fang و Lu [۷] صورت گرفته است. Fang و Loetamonphong [۵] مسأله بهینه سازی یک هدفه با توجه به معادلات رابط فازی با ترکیب max-product و تابع هدف خطی را بررسی کرده اند. به علاوه Fang ، Young و Loetamonphong [۶] مسأله بهینه سازی چند هدفه با توجه به معادلات رابط فازی با عملگر max-min را مورد بررسی قرا دادند. از این رو در تحقیق حاضر با الهام از مسأله بهینه سازی چند هدفه با توجه به معادلات رابط فازی با عملگر max-product مورد مطالعه قرار گرفت. بنابراین مسأله مورد بررسی در رابطه (۳) خلاصه می‌گردد :

$$\begin{aligned} &\min\{f_1(x),\ldots,f_p(x)\} \\ &s.t. \quad xOA = b \\ &\quad 0 \le x_i \le 1, \forall i \end{aligned} \qquad (۳)$$

که در آن f_i ها توابع هدف دلخواه می‌باشند.

در این نوشتار در بخش ۲ مفاهیم اساسی مربوط به بهینه سازی چند هدفه مرور می‌شوند. در بخش ۳ روشی ارایه می‌شود که بعد مسأله را به منظور تبدیل مسأله اصلی به یک مسأله کوچک تر کاهش می‌دهد. در بخش ۴ الگوریتم ژنتیک [۶] با اعمال تغییرات لازم برای حل مسأله (۳) ارایه می‌گردد که شامل روش تولید جمعیت اولیه، رویه انتخاب و تکثیر و عملیات تقاطع و جهش است. همچنین یک روش بهبود موضعی برای اصلاح نقاط کارای بدست آمده ارایه می‌شود.

در بخش ۵ الگوریتم ارایه شده با ارایه مثال‌های مختلفی که نتایج آنها از قبل مشخص است مورد آزمایش قرار می‌گیرد و نتیجه گیری نهایی در بخش ۶ می‌آید.

۲- بهینه سازی چند هدفه:

برخی از تعاریف بهینه سازی چند هدفه در اینجا مرور می‌شوند. اگر X ناحیه شدنی مسأله (۳) فرض گردد:

$$X = \{x \in R^m \mid xoA = b, 0 \le x_i \le 1, \forall i\}$$

هر $x \in X$، (۱) یک بردار جواب و $z = (f_1(x),\ldots,f_p(x))$ بردار معیار آن نامیده می‌شود. علاوه بر آن :

$$Z = \{z \in R^p \mid z = (f_1(x),\ldots,f_p(x)), \exists x \in X\}$$

تعریف (۱): نقطه $\overline{x} \in X$ یک جواب کارا یا یک جواب بهینه پارتو برای مسأله (۳) است اگر و فقط اگر $x \in X$ وجود نداشته باشد که $f_k(x) \le f_k(\overline{x}), \forall k = 1,\ldots,p$ و برای حداقل یک k $f_k(x) < f_k(\overline{x})$ باشد. در غیر این صورت $\overline{x}$ یک جواب غیر کارا می‌باشد.

تعریف (۲): اگر $z^1, z^2 \in Z$ دو بردار معیارفرض می شوند. آنگاه z^1 بر z^2 غلبه دارد اگر و فقط اگر $z^1 \le z^2, z^1 \ne z^2$. یعنی $z_k{}^1 \le z_k{}^2 \forall k = 1,\ldots,p$ و برای حداقل یک k رابطه $z_k{}^1 < z_k{}^2$ برقرار باشد.

تعریف (۳): اگر $\overline{z} \in Z$ باشد، آنگاه $\overline{z}$ بردار معیار غیر مغلوب است اگر و فقط اگر $z \in Z$ وجود نداشته باشد که به $\overline{z}$ غلبه داشته باشد. در غیر این صورت $\overline{z}$ یک بردار معیار مغلوب° گفته می‌شود. مجموعه تمام نقاط کارا مجموعه کارا یا مجموعه بهینه پارتو نامیده می‌شود. مجموعه تمام بردارهای غیر مغلوب نیز مجموعه غیر مغلوب نام دارد.

۳- کاهش مسأله:

یک سیستم از معادلات رابط فازی را می‌توان به نحوی دستکاری کرد که حجم عملیات لازم برای الگوریتم ژنتیک ارایه شده کاهش یابد. با توجه به رابطه (۲) برخی از متغیرها که در هر بردار جواب مقادیر ثابتی اختیار کنند و تأثیری در سایر مؤلفه‌های بردار جواب نداشته باشند را می‌توان از مسأله کنار گذاشت. اگر:

$$I=\{1,...,m\}\text{،}\ J=\{1,...,n\}$$

و

$$x^1\le x^2 \Leftrightarrow x_i^1\le x_i^2\ \ i=1,...,m\ \ \forall x^1,x^2\in X$$

باشند به وضوح رابطه $\le$ تشکیل یک رابطه ترتیب جزیی روی X می‌دهد و با این رابطه X دارای جواب ماکزیمم منحصر بفرد $\hat{x}$ به فرم رابطه (۴) می‌باشد [۱]:

$$\hat{x}=A\Theta b=\left[\bigwedge_{j=1}^{n}\left(a_{ij}\Theta b_j\right)\right]_{i\in I} \qquad (۴)$$

که در آن:

$$a\Lambda b=\min(a,b)\ \text{و}\ a_{ij}\Theta b_j=\begin{cases}1 & a_{ij}\le b_j\\ b_j/a_{ij} & a_{ij}>b_j\end{cases}$$

و تعداد متناهی جواب مینیمال می‌باشد. $\tilde{x}\in X$ جواب مینیمال است اگر:

$$\forall x\in X\,(x\le\tilde{x}\Rightarrow x=\tilde{x})$$

اگر $\tilde{X}$ مجموعه جواب‌های مینیمال X باشد،آنگاه:

$$X=\bigcup_{\tilde{x}\in\tilde{X}}\{x\in X\mid \tilde{x}\le x\le\hat{x}\}.$$

اکنون بنا به تعریف:

$$I_j=\{i\in I\mid \hat{x}_i.a_{ij}=b_j\}$$

لم (۱): اگر $\|I_j\|=1$ آنگاه برای $i\in I_j$ خواهد شد:

$$\forall x\in X\left(x_i=\hat{x}_i=\tilde{x}_i=b_j/a_{ij}\right)$$

اثبات:[۵] را ببینید.

اکنون فرض می‌شود که:

$$I'=\{i\in I_j\mid \|I_j\|=1,\ j\in J\},\ J'=\{j\in J\mid I_j\cap I'\neq\phi\}.$$

اگر به ازای یک $q\in J$، $\|I_q\|\ge 2$ و $I_q\cap I'\neq\phi$ گردد، در این صورت هر مولفه $i_0\in I_{j_0}$ مانند $I_q\cap I'$ را از I' و مولفه متناظر آن یعنی j_0 از J' حذف می‌شود. یعنی J',I' به ترتیب به صورت $J'-\{j_0\}$ و $I'-\{i_0\}$ به هنگام می‌شوند. بنابراین می‌توان فرض کرد برای هر $q\in J$ که $\|I_q\|\ge 2$، رابطه $I_q\cap I'=\phi$ حاصل می‌گردد.

حال اگر فرض شود X' مجموعه جواب‌های سیستم $x'oA'=b'$ باشد که در آن A',b' ماتریس و بردار حاصل از حذف تمام سطرهای iام وستونهای jام ماتریس A و مؤلفه های jام بردار b باشند که $i\in I',\ j\in J'$..

با فرض $j\in J-J'$ و $x\in X$ داریم $\max_{i\in I}(x_i.a_{ij})=b_j$.

اگر $\|I_j\|=1$ و $i_0\in I_j$ در این صورت از لم (۱) نتیجه می‌شود $x_{i_0}=\hat{x}_{i_0}=b_j/a_{i_0 j}$. از طرفی چون $j\in J-J'$ پس $i_0\notin I'$ و بنابراین $\max_{i\in I-I'}(x_i.a_{ij})=b_j$.

حال فرض می‌شود که $\|I_j\|\ge 2$ و $x_p.a_{pj}=b_j$. با توجه به $x_p\le\hat{x}_p$ و

$$\hat{x}_p a_{pj}\le\max_{i\in I}\hat{x}_i a_{ij}=b_j=x_p a_{pj}$$

به وضوح $x_p=\hat{x}_p$. بنابراین $p\in I_j$ و چون $I_j\cap I'=\phi$ پس $p\notin I'$. بنابراین در این حالت هم $\max_{i\in I-I'}(x_i.a_{ij})=b_j$.

بر عکس اگر $x'\in X'$ با تعریف مؤلفه‌های x به صورت زیر به وضوح $x\in X$ خواهد

$$x_i=\begin{cases}x_i' & i\in I-I'\\ \hat{x}_i & i\in I'\end{cases}$$

مبحث فوق نشان می‌دهد فرایند کاهش مسأله که به آن اشاره شد جواب‌های مسأله اصلی را تحت تأثیر قرار نمی‌دهد.

مثال (۱): فرض کنید سیستم معادلات رابط فازی (۱) داده شده باشد که در آن:

$$A=\begin{bmatrix}0.2 & 0.3 & 0.8 & 0\\ 0.6 & 0.9 & 0.7 & 1\\ 0.15 & 0.25 & 0.9 & 0.45\end{bmatrix},$$

$$b=(0.18\quad 0.27\quad 0.9\quad 0.45)$$

در این صورت با استفاده از رابطه (۴)، $\hat{x}=(0.9\quad 0.3\quad 1)$ و محاسبه‌ای ساده نشان می‌دهد:

$$I_1=\{1,2\},I_2=\{1,2\},I_3=\{3\},I_4=\{3\}$$

$$J'=\{3,4\}\ \text{و}\ I_2\cap I'=I_1\cap I'=\phi,I'=\{3\}$$

لذا با حذف سطر ۳ و ستون های ۳ و ۴ ماتریس A و با حذف مولفه‌های ۳ و ۴ بردار b و تخصیص مقدار $x_3=1$ سیستم فوق به سیستم زیر تقلیل می‌یابد:

$$x'oA'=b',0\le x_i'\le 1\forall i\in I-I'$$

که در آن:

$$A'=\begin{bmatrix}0.2 & 0.3\\ 0.6 & 0.9\end{bmatrix},b'=(0.18\quad 0.27)$$

۴- الگوریتم ژنتیک:

فرض بر این است که ابعاد مسأله با به کار بردن روش ارایه شده در بخش قبل کاهش یافته است. مؤلفه‌های اصلی الگوریتم ژنتیک ارایه شده شامل عملگرهای تقاطع و جهش و عملیات انتخاب و تکثیر می‌باشند که در این بخش می‌آیند.

۴-۱- معرفی[۶] :

اگر از رشته دودویی[۷] [۳] برای معرفی هر فرد استفاده شود، برای مسایل با دقت مطلوب بالا باید وجود فضای جستجوی بزرگ را لحاظ نمود و بدین جهت الگوریتم برای اجرا احتیاج به زمان زیادی دارد. برای غلبه بر این مشکل می‌توان از صورت ممیز شناور[۸] [۸] که امکان اجرای ساده‌تر و کاراتر عملگرهای تقاطع و جهش را فراهم می‌آورد استفاده کرد. به این صورت که در الگوریتم ارایه شده هر فرد توسط یک m-بردار $x=(x_1,\ldots,x_m)$ معرفی می‌گردد، که در آن $x_i\in[0,1]$.

۴-۲- تولید جمعیت اولیه:

جمعیت اولیه باید به نحوی تولید شود که به طور مناسب فضای جستجو را پوشش دهد. برای این منظور از ایده به کار رفته در [۶] بهره گرفته شده و الگوریتم‌های ۱ و ۲ ارایه می‌شوند.

الگوریتم (۱):

(اصلاح جواب x برای شدنی بودن)

فرض می‌گردد که E_j مجموعه اند یس مؤلفه‌هایی باشد که می‌توانند در محدودیت j ام صدق کنند. یعنی $E_j=\{i\in I\mid a_{ij}\ge b_j\}$.

۱- محدودیت j را که جواب x در آن صدق نمی‌کند انتخاب کن.

۲- به تصادف $k\in E_j$ را انتخاب کن و قرار بده $x_k=\hat{x}_k$.

۳- شدنی بودن جواب جدید را امتحان کن. در صورت نشدنی بودن به مرحله ۱ برو. در غیر این صورت توقف کن.

الگوریتم(۲):

(تولید جمعیت اولیه)

۱- اگر b' تهی است به تعداد $2\times m\times d$ فرد با انتخاب تصادفی مؤلفه i ام هر فرد از بازه $[0,\hat{x}_i]$، تولید کن.

۲- اگر b' ناتهی است نیمی از جمعیت را به شکل زیر تولید کن:

برای هر $i\in I$، d فرد با قرار دادن $x_{i'}=\hat{x}_{i'}$ برای هر $i'\in I, i'\neq i$ و انتخاب تصادفی x_i از بازه $[0,\hat{x}_i]$، تولید کن.

برای تولید نیمه دوم جمعیت اولیه:

- $m\times d$ فرد را با انتخاب تصادفی x_i از بازه $[0,\hat{x}_i]$، تولید کن.
- شدنی بودن هر فرد را امتحان کن. در صورت نشدنی بودن فرد آن را با استفاده از الگوریتم (۱) اصلاح کن.

۴-۳- رویه انتخاب و تکثیر

به منظور جلوگیری از بروز مشکلاتی مانند علامت توابع هدف و پدیده فرد برتر[۹] [۸] که باعث ایجاد همگرایی زودرس[۱۰] [۸] می‌شود می‌توان از ایده تخصیص رتبه[۱۱] به هر فرد استفاده کرد. مشابه با[۶] رتبه فرد p ام در جمعیت جاری، یعنی r_p به صورت زیر در نظر گرفته می‌شود:

r_p = pop-size-تعدادغالبها

که در آن تعداد غالبها به تعداد افراد در جمعیت جاری دلالت دارد که بردار معیارشان به بردار معیار فرد p ام غلبه دارد. سپس از ایده roulette wheel [۸] برای تولید نسل بعدی به کمک احتمال انتخاب فرد p ام توسط رابطه

$$r_p\Big/\left(\sum_{i=1}^{pop-size} r_i\right)$$

استفاده می‌شود.

به منظور حفظ مجموعه جوابهای کارا، اگر E مجموعه جواب‌های کارا فرض شود که در طول اجرای الگوریتم تا به حال حاصل شده است. در هر تکرار در مرحله انتخاب بعد از محاسبه بردار تناسب[۱۲] هر فرد متعلق به جمعیت جاری بررسی می‌شود که آیا بردار معیار آن می‌تواند به یکی از افراد متعلق به E غلبه کند یا نه. سپس E با بیرون کشیدن فرد مغلوب و اضافه کردن فرد غالب به هنگام می‌شود.

از آنجا که در برخی از موارد E مجموعه‌ای همبند می‌شود لذا، در طول اجرای الگوریم اندازه این مجموعه به سرعت بزرگ می‌شود و این، کارایی الگوریتم را تحت تأثیر قرار داده و زمان اجرای الگوریتم را به سرعت بالا می‌برد. با اعمال کنترل روی اندازه E باید این مشکل را بر طرف کرد.

به این ترتیب که بعد از چند تکرار اگر اندازه E از اندازه مطلوب، که با متغیر max-size نشان داده می‌شود، بالاتر بود با استفاده از ایده دسته بندی فازی[۱۳] [۱۴] اعضای E را در C دسته جای داده و درجه تعلق هر عضو به هر دسته محاسبه می‌شود. سپس یک عضو متعلق به دسته خاص گرفته می‌شود. اگر درجه عضویت آن عضو در دسته مذکور در مقایسه با سایر دسته‌ها بیشترین مقدار باشد. در نهایت تعداد مطلوب

نقاط را از دسته ها به طور یکنواخت انتخاب کرده و سایر نقاط از E حذف می‌شوند. در اینجا C= ۵ انتخاب شده است.

٤-٤- تقاطع و جهش

با توجه به شباهت ناحیه شدنی سیستم معادلات فازی با ترکیبات max -min وmax-product ، عملیات تقاطع و جهش ارایه شده در[۶] به کار برده می‌شود.

الگوریتم (۳): (عملیات جهش)

اگر فرض شود که ξ مقدار آستانه مجاز برای جهش باشد. برای هر فرد در جمعیت به صورت زیر انجام می‌گردد:

۱- m-بردار r را از اعداد تصادفی در بازه [0,1] تولید کن.

۲- برای i=1,...,m اگر $r_i \le \xi$ آنگاه x_i را بطور تصادفی از بازه $[0,\hat{x}_i]$ انتخاب کن.

۳- در صورت نشدنی بودن فرد جهش یافته با استفاده از الگوریتم (۱) آن را اصلاح کن.

الگوریتم (۴): (تقاطع)

۱- قرار بده $p = 1$.

۲- به تصادف دو فرد متمایز x^1, x^2 را از جمعیت جاری انتخاب کن.

۳- عدد تصادفی $r \in [0,1]$ انتخاب کن. اگر $r \ge 0.5$ به مرحله ۵ و در غیر اینصورت به مرحله ۴ برو.

۴- عدد تصادفی $\lambda \in [0.5,1]$ را انتخاب کن. قرار بده $\bar{x}^1 = \lambda x^1 + (1-\lambda) x^2$ و به مرحله ۶ برو.

۵- عدد تصادفی $\gamma \in [1, 0.5]$ را انتخاب کن. قرار بده $\bar{x}^1 = \gamma x^1 - (\gamma - 1) x^2$.

۶- عدد تصادفی $r \in [0,1]$ انتخاب کن. اگر $r \ge 0.5$ به مرحله ۸ و در غیر اینصورت به مرحله ۷ برو.

۷- عدد تصادفی $\lambda \in [0.5,1]$ را انتخاب کن. قرار بده $\bar{x}^2 = \lambda x^1 + (1-\lambda) x^2$ و به مرحله ۹ برو.

۸- عدد تصادفی $\lambda \in [0.5,1]$ را انتخاب کن و قرار بده $\bar{x}^2 = \gamma x^1 - (\gamma - 1) x^2$.

۹- عدد تصادفی $r_1 \in [0,1]$ را انتخاب کن. اگر $r_1 \ge \delta$ به مرحله بعد برو. در غیر این صورت عدد تصادفی $r_2 \in [0,1]$ را انتخاب کن. اگر $r_2 < 0.5$ عدد جدید $\lambda \in [0.5,1]$ را انتخاب کن و قرار بده $\bar{x}^1 = \lambda x^1 + (1-\lambda)\hat{x}$. در غیر اینصورت عدد جدید $\gamma \in [1, 0.5]$ را انتخاب کن و قرار بده $\bar{x}^1 = \gamma x^1 - (\gamma - 1)\hat{x}$.

۱۰- عدد تصادفی $r_1 \in [0,1]$ را انتخاب کن. اگر $r_1 \ge \delta$ به مرحله بعد برو. در غیر این صورت عدد تصادفی $r_2 \in [0,1]$ را انتخاب کن. اگر $r_2 < 0.5$ عدد جدید $\lambda \in [0.5,1]$ را انتخاب کن و قرار بده $\bar{x}^1 = \lambda x^1 + (1-\lambda)\hat{x}$. در غیر اینصورت عدد جدید $\gamma \in [1, 0.5]$ را انتخاب کن و قرار بده $\bar{x}^1 = \gamma x^1 - (\gamma - 1)\hat{x}$.

۱۱- اگر برای یک $i \in I$، $\bar{x}_i^1$ (یا $\bar{x}_i^2$) خارج از بازه $[0, \hat{x}_i]$ باشند مقدار $\bar{x}_i^1$ (یا $\bar{x}_i^2$) را با یک عدد تصادفی در بازه $[0, \hat{x}_i]$ عوض کن.

۱۲- در صورت نشدنی بودن $\bar{x}_i^1$ یا $\bar{x}_i^2$ آنها را با استفاده از الگوریتم (۱) اصلاح کن.

۱۳- $\bar{x}^1$ و $\bar{x}^2$ را در جمعیت جدید قرار بده.

۱۴- قرار بده $p = p + 1$. اگر $p \le pop - size/2$ به مرحله ۲ برو در غیر این صورت توقف کن.

٤-٥- عملیات بهبودموضعی[۱٤]

از ایده بهبود موضعی[۹] برای بهبود جواب‌های بدست آمده با الگوریتم ژنتیک استفاده می‌شود. به این ترتیب که با اضافه کردن یک بردار، که مؤلفه‌های آن از یک همسایگی کوچک صفر مثلا[0.05 , 0.05-] به تصادف انتخاب شده‌اند، به یکی از نقاط کارا، یک نقطه جدید در همسایگی آن حاصل می‌شود. سپس در صورت نشدنی بودن نقطه جدید، شدنی بودن آن توسط الگوریتم(۱) برقرار می‌شود. اگر بردار معیار نقطه جدید حاصل به بردار معیار یکی از نقاط کارای داده شده غلبه کند، مجموعه نقاط کارای E با اضافه کردن نقطه جدید و حذف نقطه مغلوب به هنگام می‌شوند.

روال فوق را در مورد هر جواب کارای بدست آمده توسط الگوریتم ژنتیک می‌توان به تعداد ثابت مثلا ۱۰۰ بار تکرار کرد. روال مذکور یک روش موثر در تصحیح نقاط کارا می‌باشد.

٤-٦- جمع بندی الگوریتم ژنتیک ارایه شده:

الگوریتم ژنتیک ارایه شده برای حل مسأله بهینه سازی چند هدفه با توجه به محدودیت های معادلات رابط فازی با ترکیب max-product به صورت زیر جمع بندی می‌گردد.

الگوریتم (۵):

اگر فرض شود که max-gen ماکزیمم تعداد تکرارها باشد.

۱- قرار بده k=1.

۲- جمعیت اولیه را با استفاده از الگوریتم (۲) تولید کن.

۳- عملیات انتخاب و تکثیر را اجرا کن. مجموعه کارا ی E را با قرار دادن مجموعه نقاط کارای جمعیت اولیه در آن تشکیل بده.

۴- عملیات جهش را اجرا کن.

۵- عملیات تقاطع را اجرا کن.

۶- عملیات انتخاب و تکثیر را اجرا کن. مجموعه کارا ی E را به هنگام کن. برای هر تعداد خاص تکرار مثلا ۵ تکرار اگر اندازه E بزرگتر از max-size است روش دسته بندی فازی را برای کاستن اندازه E به max-size به کار ببر.

۷- اگر k < max-gen قرار بده k=k+1 و به مرحله ۳ برو.

۸- عملیات بهبود موضعی را روی مجموعه کارای حاصل اجرا کن.

در این مقاله نتایج بر پایه max-gen=200 و d=20 است. همچنین برنامه به زبان MATLAB نوشته شده و در کامپیوتر پنتیوم 4 با سرعت CPU برابر 3 GHz اجرا شده است.

۵- امتحان الگوریتم ژنتیک:

در اینجا الگوریتم روی چند مسأله با توابع هدف خطی و غیر خطی امتحان شده و نشان داده می‌شود این الگوریتم به طور کارا توانایی یافتن نقاط بهینه پارتو را دار است. به منظور بررسی توانایی الگوریتم در تعیین مجموعه کارا تعریف و قضیه زیرآورده می‌شود:

تعریف (۴): زمانی که مسأله بهینه سازی چند هدفه با توابع هدف خطی مطرح است، مخروط قطبی نیمه مثبت [۱۰] تولید شده توسط بردارهای گرادیان توابع هدف به صورت زیر تعریف می‌شود:

$$C^{\geq}=\{c\in R^m \mid a_k^{\,T}.c\geq 0, a_k^{\,T}.c\neq 0\forall k\}\cup\{0\}$$

که در آن $a_k=\nabla Z_k(x)$ گرادیان kامین تابع هدف است. علاوه بر این برای یک مسأله ماکزیمم سازی با ناحیه شدنی X، قضیه زیر برقرار است.

قضیه (۱): $x\in X$ کاراست اگر و تنها اگر

$$(\{x\}\oplus C^{\geq})\cap X=\{x\}.$$

اثبات: [۱۱] را ببینید.

تذکر: با توجه به اینکه مسأله مورد بررسی در این مقاله یک مسأله مینیمم سازی است پس به منظور استفاده از قضیه (۱) در تعیین نقاط کارا باید در تعریف (۴)، به جای a_k ها، منفی بردارهای گرادیان در نظر گرفته شوند.

۵-۱- توابع هدف خطی:

مجموعه بهینه پارتو با توجه به توابع هدف مورد نظر دارای موقعیت‌های مختلفی در ناحیه شدنی است. نشان داده می‌شود الگوریتم ارایه شده توانایی تعیین نقاط کارا را در موقعیت‌های مختلف داراست. برای حالتی که توابع خطی دو متغیره مطرح است می‌توان از قضیه (۱) برای تعیین مجموعه کارا استفاده کرد.

۵-۱-۱- دو تابع هدف دو متغیره

سیستم معادلات رابط فازی مثال (۱) موردنظر می‌باشد. با توجه به اینکه ابعاد مسأله کاهش داده شد ناحیه شدنی را می‌توان در صفحه x_1-x_2 به صورت شکل (1) رسم کرد. بررسی ساده نشان می‌دهد ناحیه شدنی عبارت است از $S^1\bigcup S^2$ که در آن:

$$S^1=\{(x_1,x_2,x_3)\mid x_1=0.9, 0\leq x_2\leq 0.3, x_3=1\}$$
$$S^2=\{(x_1,x_2,x_3)\mid 0\leq x_1\leq 0.9, x_2=0.3, x_3=1\}$$

ابتدا الگوریتم را روی ٤ مسأله که هر کدام دارای دو تابع هدف دو متغیره خطی با محدودیت سیستم معادلات رابط فازی max-product با اعمال کنترل max-size=50 آزمایش می‌شود. مخروط‌های قطبی نیمه مثبت تولید شده توسط بردار منفی گرادیان های دو تابع در هر ۴ حالت در شکل (۸) نشان داده شده است. v^1,v^2 به ترتیب معرف منفی بردار گرادیان توابع هدف اول و دوم و y^1,y^2 معرف بردارهای عمود بر v^1,v^2 هستند.

حالت I-۱: فرض کنید توابع هدف به صورت زیر داده شده‌اند:

$$\begin{cases} f_1(x)=-0.1x_1-0.3x_2-0.5x_3 \\ f_2(x)=-0.2x_1-0.3x_2+x_3 \end{cases}$$

همانطورکه قبلا بررسی شد، در هر بردار جواب متغیر x_3 مقدار ثابت $x_3=1$ را اختیار می‌کند، از این رو می‌توان این متغیر را نادیده گرفت. به طور نظری با استفاده از قضیه (۱) این مسأله دارای یک نقطه بهینه پارتو (0.9,0.3,1) است. جواب بدست آمده توسط الگوریتم ژنتیک هم دقیقا همان نقطه‌ای را نشان می‌دهد که در شکل (۲) آمده است.

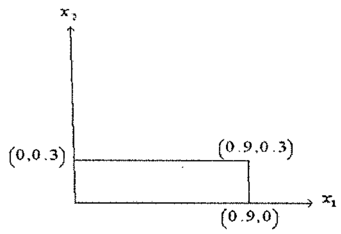
حالت I-۲: فرض کنید توابع هدف به صورت زیرداده شده‌اند:

$$\begin{cases} f_1(x)=x_1+x_2+x_3 \\ f_2(x)=-0.5x_1+x_2+0.3x_3 \end{cases}$$

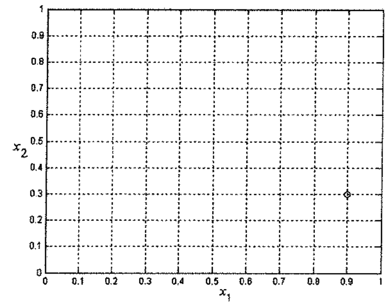
به طور نظری با استفاده از قضیه (۱) جوابهای کارای این مسأله به صورت

$$\{(x_1,x_2,x_3)\mid 0\leq x_1\leq 0.6, x_2=0.3, x_3=1\}\cup\{(0.9,0,1)\}$$

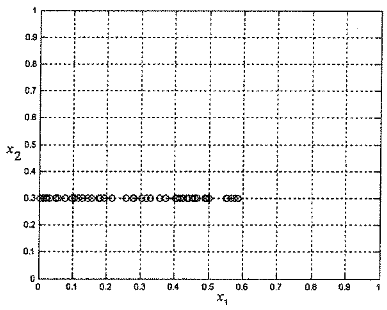
است. جواب‌های بدست آمده توسط الگوریتم ژنتیک در شکل (۳) نشان داده شده است.



شکل (۱): فضای شدنی برای مسأله دو بعدی.



شکل (۲): مجموعه کارا برای حالت I-۱.



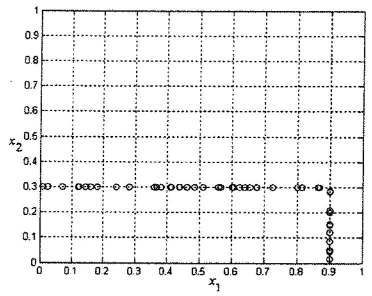
شکل (۳): مجموعه کارا برای حالت I-۲.

حالت I-۳: فرض کنید توابع هدف به صورت زیر داده شده‌اند:

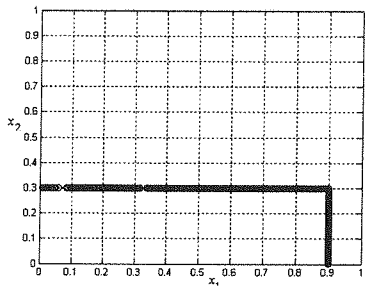
$$\begin{cases} f_1(x)=0.1x_1-0.3x_2+x_3 \\ f_2(x)=-0.2x_1+0.3x_2-x_3 \end{cases}$$

با توجه به قضیه (۱) تمام ناحیه شدنی تشکیل مجموعه کارا را می‌دهد. جواب‌های بدست آمده توسط الگوریتم ژنتیک هم مبین همان نقاطی است که در شکل (۴) رسم شده است.

حالت I-۳': به منظورروشن شدن لزوم اعمال کنترل روی اندازه مجموعه کارا، الگوریتم روی مسأله فوق بدون اعمال کنترل روی اندازه E اجرا شده است، ولی با ۲۰ تکرار و بدون به کار بردن روش بهبود موضعی. جوابهای حاصل در این حالت هم در شکل (۵) نشان داده شده است. زمان اجرای الگوریتم در این حالت به شدت بالا رفته و کارایی الگوریتم تحت تأثیر قرار می گیرد.زمان اجرای الگوریتم و اندازه نقاط کارا در حالات مختلف در جدول (۱) آمده است.



شکل (۴): مجموعه کارا برای حالت I-۳.



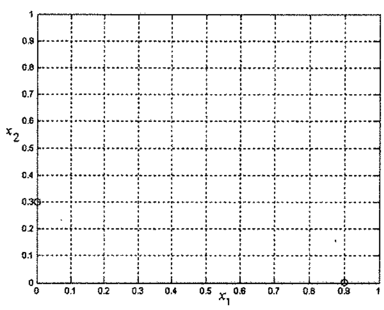
شکل (۵): مجموعه کارا برای حالت I-۳'.

حالت I-۴: فرض کنید توابع هدف به صورت زیر داده شده‌اند:

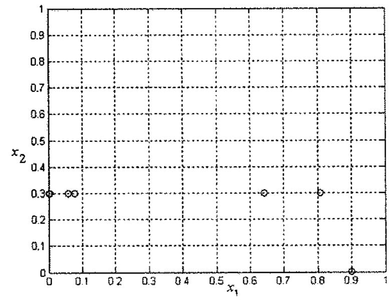
$$\begin{cases} f_1(x)=x_1+x_3 \\ f_2(x)=x_2-x_3 \end{cases}$$

به طور نظری با استفاده از قضیه (۱) مجموعه جوابهای کارای این مسأله به صورت $\{(0,0.3,1),(0.9,0,1)\}$ است. نتایج بدست آمده توسط الگوریتم ژنتیک در شکل (۶) رسم شده است.

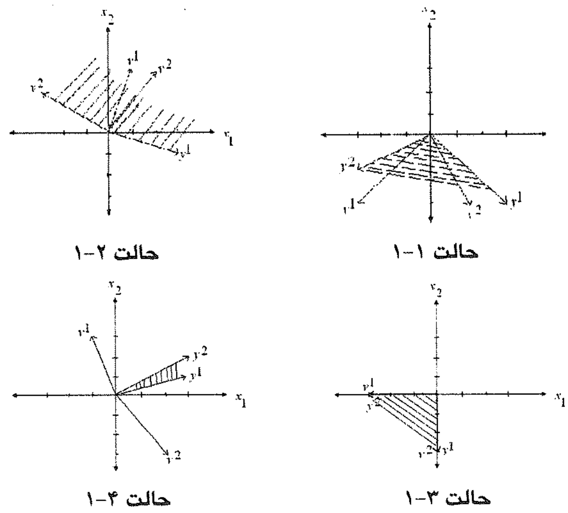
حالت I-۴': به منظور روشن شدن نقش بهبودی موضعی ، الگوریتم روی مسأله فوق بدون به کار بردن عملیات بهبودی موضعی روی نقاط کارای حاصل اجرا شده است. نتایج حاصل در جدول (۱) و شکل (۷) نشان داده شده اند. مشخص است که باحذف عملیات بهبودی موضعی احتمالا تعداد زیادی از نقاط به عنوان نقاط کارا به دست می‌آید که باعث نا کارایی الگوریتم می‌شود.



شکل (۶): مجموعه کارا برای حالت I-۴.



شکل (۷): مجموعه کارابرای حالت '۳-I.



شکل (۸): مخروطهای قطبی نیمه مثبت برای هر ۴ حالت.

جدول (۱): زمان اجراواندازه مجموعه کارا برای مسایل I.

زمان اجرا (ثانیه)	اندازه مجموعه کارا	حالت
۳٤۲.۸۸	۱	I-1
۸۳۳.۲۵	۵۰	I-2
۸۳۰.۷٦٦	۵۰	I-3
۱۰۹۴.۰۱۶	۱۰۱۸	I-3'
۳۷٤.۱۵٦	۲	I-4
۳٤٦.۷۰۳	۸	I-4'

۵-۱-۲- توابع هدف سه گانه سه متغیره

برای بررسی الگوریتم ارایه شده روی مسایل با توابع هدف سه متغیره فرض بر این است که در هر حالت سیستم معادلات رابط فازی به صورت $xoA=b, 0\le x_i\le 1\forall i$ برقرار است که در آن:

$$A=\begin{bmatrix}0.8 & 0 & 0.18 & 0.3\\ 1 & 0.9 & 0.225 & 0.25\\ 0.5 & 0.8 & 0.2 & 0.127\\ 0.3 & 0.4 & 0.019 & 1\end{bmatrix}, b=(0.8,0.72,0.18,0.5)$$

محاسبه ساده‌ای نشان می‌دهد که جواب ماکزیمم سیستم فوق برابر با $\hat{x}=(1,0.8,0.9,0.5)$ است.

با به کار بردن عملیات کاهشی که در بخش ۳ توضیح داده شد در هر بردار جواب $x=(x_1,x_2,x_3,x_4)$ متغیر x_4 مقدار ثابت $x_4=0.5$ را اختیار می‌کند.

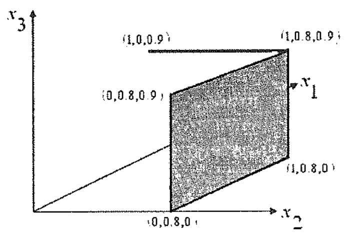
بررسی ساده نشان می‌دهد ناحیه شدنی عبارت است از $S^1\bigcup S^2$ که در آن:

$$S^1=\{(x_1,x_2,x_3,x_4)\mid x_1=1, 0\le x_2\le 0.8, x_3=0.9, x_4=0.5\},$$
$$S^2=\{(x_1,x_2,x_3,x_4)\mid 0\le x_1\le 1, x_2=0.8, 0\le x_3\le 0.9, x_4=0.5\}$$

در واقع با سیستم کاهشی $x'oA'=b', 0\le x_i'\le 1$ مواجه بوده که درآن:

$$A'=\begin{bmatrix}0.8 & 0 & 0.18\\ 1 & 0.9 & 0.225\\ 0.5 & 0.8 & 0.2\end{bmatrix}, b'=(0.8,0.72,0.18)$$

ناحیه شدنی این سیستم که از یک صفحه وخط متصل تشکیل شده در شکل شکل (۹): ناحیه شدنی برای مسأله سه بعدی.. رسم شده است.



شکل (۹): ناحیه شدنی برای مسأله سه بعدی. .

حال الگوریتم ژنتیک روی مسایل زیر با محدود یت‌های فوق با اعمال کنترل max-size=120 اجرا می‌شود. به منظور بالا بردن کارایی الگوریتم در هر ۲ تکرار اندازه مجموعه کارا بررسی می‌شود.

حالت II-۱: فرض کنید توابع هدف به صورت زیر داده شده‌اند:

$$\begin{cases} f_1(x)=x_1+x_4\\ f_2(x)=x_2-x_4\\ f_3(x)=x_3+2x_4\end{cases}$$

با توجه به ضابطه توابع هدف به سهولت می‌توان تحقیق کرد که مجموعه کارا به صورت $\{(1,0,0.9,0.5),(0,0.8,0,0.5)\}$ می‌باشد. نتیجه حاصل از الگوریتم هم دقیقا این نقاط می‌باشد که در شکل (۱۰) نشان داده شده است.

حالت II-۲: فرض کنید توابع هدف به صورت زیر داده شده‌اند:

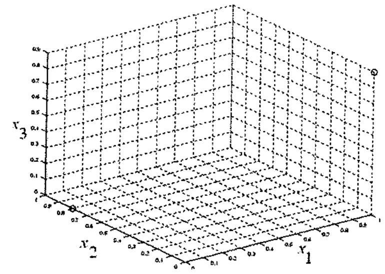
$$\begin{cases} f_1(x) = x_1 + x_2 + x_3 + x_4 \\ f_2(x) = -x_1 - x_2 - x_3 \\ f_3(x) = 2x_1 - x_3 - 0.8x_4 \end{cases}$$

با توجه به دو تابع f_1, f_2 ، تقریبا تمام ناحیه شدنی تشکیل مجموعه کارا می‌دهد. نتایج حاصل از اجرای الگوریتم با ۱۰ تکرار و بدون اعمال کنترل روی اندازه مجموعه کارا و بدون عملیات بهبود موضعی که در شکل (۱۱) نشان داده شده است کاملا با این مطلب سازگاری دارد.

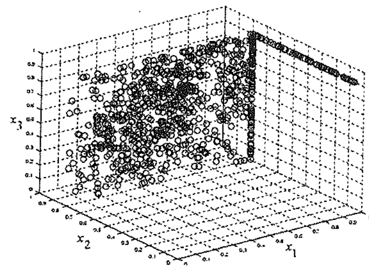
حالت II-3: فرض کنید توابع هدف به صورت زیر داده شده‌اند:

$$\begin{cases} f_1(x) = x_1 + 2x_2 + 2x_3 + 6x_4 \\ f_2(x) = 2x_1 - 3x_2 + 6x_3 - x_4 \\ f_3(x) = 3x_1 + 8x_2 - 3x_3 + 4x_4 \end{cases}$$

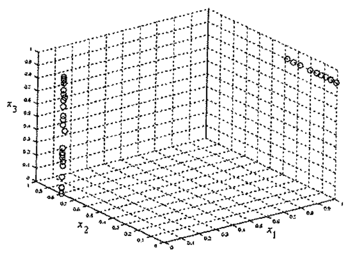
شکل (۱۲) نتیجه حاصل از اجرای الگوریتم را نشان می‌دهد.



شکل (۱۰): مجموعه کارا برای حالت II-1.



شکل (۱۱): مجموعه کارا برای حالت II-2.



شکل (۱۲): مجموعه کارا برای حالت II-3.

جدول (۲):زمان اجراواندازه مجموعه کارا برای مسایل II.

زمان اجرا (ثانیه)	اندازه مجموعه کارا	حالت
۱۰۰۰.۹۴۸	۲	I-1
۲۷۶۹.۸۲۸	۱۲۰	I-2
۱۳۱۵.۴۲۳	۳۴	I-3

۵-۲- توابع هدف غیر خطی

الگوریتم ژنتیک را می‌توان روی انواع توابع هدف از جمله توابع غیر خطی بکار برد. برای روشن شدن مطلب الگوریتم روی مسأله زیر پیاده می‌شود:

$$\min \begin{cases} f_1(x) = 10(x_1 - 0.4)^2 + 10(x_2 - 0.3)^2 \\ f_2(x) = 10(x_1 - 0.7)^2 + 10(x_2 - 0.3)^2 \end{cases}$$

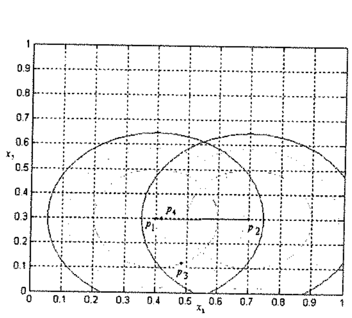
$$s.t. \quad xo\begin{pmatrix} 0.2 & 0.3 \\ 0.6 & 0.9 \end{pmatrix} = (0.18, 0.27)$$

$$0 \le x_i \le 1, \forall i$$

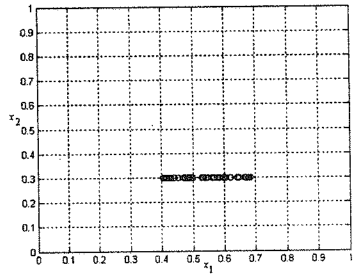
باید توجه داشت که همان محدودیت‌های مسایل I برقرار می‌باشد که ناحیه شدنی آن در شکل (۱) رسم شده است. منحنی‌های تراز این توابع در شکل (۱۳)رسم شده اند که ترازها از درون به بیرون به ترتیب معرف مقادیر 0.4 و 0.8 و 1.2 برای توابع هستند.

در نقطه p_1 مقدار اولین تابع مینیمم و در نقطه p_2 مقدار دومین تابع مینیمم می‌شود. نقاط p_3, p_4 روی تراز مشابهی قرار دارند، بنابراین دو تابع دارای مقدار مشترکی در این نقاط هستند. با این حال مقدار تابع اول در نقطه p_4 کمتر از مقدار تابع دوم در این نقطه است. لذا p_4 بر p_3 غلبه دارد. به بیان مشابه نقاط بین p_1 و p_2 نقاط کارا هستند. پس مجموعه کارا به فرم $\{(x_1, x_2) \mid x_2 = 0.3, 0.4 \le x_1 \le 0.7\}$ است.

الگوریتم روی این مسأله با اعمال کنترل max-size=50 اجرا شده است که زمان اجرای الگوریتم برابر با ۱۲۰۶.۰۶۲ ثانیه واندازه مجموعه کارای حاصل برابر ۵۰ شد. نقاط مورد انتظاربطور کامل حاصل شد که درشکل (۱۴)نشان داده شده است.



شکل (۱۳): منحنی های تراز دو تابع غیر خطی



شکل (۱۴): مجموعه کارا برای حالت غیر خطی

۶- نتیجه

از انجام امتحان الگوریتم نتیجه می‌شود که الگوریتم توانایی تعیین جواب‌های کارا را در فضای دو بعدی و سه بعدی تحت انواع توابع هدف خطی و غیر خطی با موقعیت‌های مختلف نقاط کارا در ناحیه شدنی را داراست.

همچنین نتایج حاصل از اجرای الگوریتم نشان می‌دهد که به منظور افزایش کارایی الگوریتم باید روی اندازه مجموعه کارا کنترل داشته و از رشد سریع اندازه این مجموعه جلوگیری کرد. همچنین جهت اصلاح نقاط کارا و دست یابی به جواب‌های مطلوب به کارگیری عملیات بهبود موضعی ضروری است. بررسی زمان اجرا در حالت‌های مختلف بر این مطلب دلالت دارد که زمان اجرای الگوریتم رابطه مستقیم بااندازه مجموعه کارا دارد.

۷- مراجع

[۱] Bourke, M. M.; Fisher, D. G.; "*Solution algorithms for fuzzy relational equations with max-product composition*", Fuzzy sets and Systems, vol. 94, p.p. 61-69, 1998.

[۲] Fang, S.C.; Li, G.; "*Solving fuzzy relation equations with a linear objective function*", Fuzzy sets and Systems, vol. 103, p.p. 107-113, 1999.

[۳] Holland, J.H.; *Adaptation in Natural and Artificial Systems*, University of Michigan, 1975.

[۴] Klir, G. J.; Folger, T.; *Fuzzy sets, Uncertainty and information*, Prentice-Hall, NJ, 1998.

[۵] Loetamonphong, J.; Fang, S.C.; "*Optimization of fuzzy relation equations with max-product composition*", Fuzzy sets and Systems, vol .118, p.p. 509-517, 2001.

[۶] Loetamonphong, J.; Fang, S.C.; Young, R.E.; "*Multi-objective optimization problems with fuzzy relation equation constraints*", Fuzzy sets and Systems, vol. 127, p.p. 141-164, 2002.

[۷] Lu, J.; Fang, S.C.; "*Solving nonlinear optimization problems with fuzzy relation equations constraints*", Fuzzy sets and Systems, vol. 119, p.p. 1-20, 2001.

[۸] Michalewicz, Z.; *Genetic Algorithms+Data Structures=Evolution Program*, Springer, Berlin/Heidelbeg/New York, 1996.

[۹] Reeves, C.R.; *Modern Heuristic Techniques for Combinatorial Problems*, Wiely, New York, 1993.

[۱۰] Sanchez, E.; "*Resolution of composite fuzzy relation equations*", Inform And Control, vol. 30, p.p. 38-48, 1976.

[۱۱] Steuer, R.E.; *Multiple Criteria Optimization: Theory, Computation, and Application*, Wiley, New York, 1996.

[۱۲] Thole, U.; Zimmermann, H. J.; Zysno, P.; "*On the suitability of minimum and product operators for intersection of fuzzy Sets*", Fuzzy sets and Systems, vol. 2, p.p. 167-180, 1979.

[۱۳] Yager, R.R.; "*Some procedures for selecting fuzzy set-theoretic operators*", Internat. J. General Systemes, vol. 8, p.p. 235-242, 1982.

[۱۴] Zimmermann, H. J.; Fuzzy Set Theory and its Applications, Kluwer Academic Publishers, Boston/Dordrecht/London, 1991.

[۱۵] Zimmermann, H. J.; Zysno, P.; "*Latent connectives in human decision-making*", Fuzzy sets and Systems, vol. 4, p.p. 37-51, 1980.

۸- زیرنویس ها

[۱] Selection/reproduction
[۲] Mutation
[۳] Crossover
[۴] Boolean relation
[۵] Dominated
[۶] Representation
[۷] Binary string
[۸] Floating point representation
[۹] super-individual
[۱۰] Premature convergence
[۱۱] Rank
[۱۲] fitness
[۱۳] Fuzzy clustering
[۱۴] Local improvement
[۱۵] semi-positive polar cones