

## ارائه یک الگوریتم ترکیبی شبکه‌های عصبی - تکامل توام ژنتیک، جهت مساله طراحی مقاوم چند متغیره در مهندسی کیفیت

محمدرضا مهرگان<sup>۱\*</sup>، علیرضا فراست<sup>۲</sup>

۱. دانشیار دانشکده مدیریت دانشگاه تهران، ایران

۲. کارشناس ارشد دانشکده مدیریت دانشگاه تهران، ایران

(تاریخ دریافت مقاله: ۱۳۸۷/۱۲/۷، تاریخ تصویب: ۱۳۸۷/۴/۲۴)

### چکیده

در این مقاله یک الگوریتم ترکیبی برای حل مساله طراحی مقاوم با چندین متغیر پاسخ ارائه شده است. الگوریتم ارائه شده، ترکیبی از شبکه‌های عصبی و تکامل توام ژنتیکی است که در آن شبکه‌های عصبی به عنوان تقریب زنده تابع، نگاشت بین متغیرهای فرایند را تقریب زده و الگوریتم تکامل توام مدل ساخته با هدف مقاوم ساختن متغیرهای پاسخ فرایند، را حل می‌نماید و نتایج این الگوریتم با الگوریتم ژنتیک مقایسه می‌شود. روش پیشنهادی در یک مطالعه موردی فرایند ریسندگی اپن اند مورد آزمایش قرار گرفته است.

واژه‌های کلیدی:

## ۱. مقدمه

فرایند را می‌توان ترکیبی از دستگاه‌ها، روش‌ها و افراد تصور کرد بطوریکه ورودی‌ها (غالباً مواد) را تبدیل به خروجی‌هایی که یک یا چند پاسخ قابل مشاهده (متغیرهای پاسخ) دارد، می‌کند. بعضی از متغیرها ( $X_1, X_2, \dots, X_p$ ) در فرآیند کنترل پذیر بوده در صورتیکه متغیرهای دیگر ( $Z_1, Z_2, \dots, Z_q$ ) کنترل ناپذیرند (هر چند برای هدف‌های یک آزمون ممکن است کنترل پذیر باشند) [۳].

بسیاری از مسایل دنیای واقعی، مطلوبست علاوه بر اینکه یک سیستم در سطح بالایی از عملکرد قرار داشته باشد، تا حد زیادی نیز مقاوم باشد به این مفهوم که با تغییر شرایط، عملکرد سیستم دچار اختلال یا افول نشود. به طور عمده در سیستم‌های پویا یک جواب بهینه برای سیستم، نمی‌تواند همواره حالت ایده آل و قابل قبولی برای عملکرد سیستم باشد زیرا با تغییر شرایط حاکم ممکن است جواب بهینه به جواب نزدیک به بهینه یا حتی جوابی نامطلوب تبدیل شود. این دیدگاه از مطالعه سیستم‌های طبیعی نشأت گرفته شده است. یک ویژگی اساسی سیستم‌هایی که قادر هستند به حیات خود ادامه دهند، تطابق‌پذیری و مقاوم بودن است. این ویژگی باعث می‌شود تا عملکرد اینگونه سیستم‌ها در برابر تغییرات محیطی دارای نوسان‌های کمی باشد.

در طراحی سیستم، طراحی مقاوم به این مفهوم است که پارامترهای فرایند طوری طراحی شوند که فرایند نسبت به تغییرات متغیرهای محیطی و پارامترهای فرایند غیر حساس باشد. در اینجا فرض بر این است که بهینه‌سازی مقاوم به طراحی مقاوم می‌انجامد. روش‌های سنتی و کلاسیک در طراحی پارامتر و مقاوم سازی فرایند عمدتاً شامل روشهای آماری مثل طراحی آزمایشات و روش تاگوچی است [۲۵] که بطور بسیار وسیعی در صنایع گوناگون مورد استفاده قرار گرفته است. لازم به ذکر است که روش تاگوچی با انتقاداتی رو به رو بوده است [۱۸]. در اواخر دهه هشتاد بازنگری جامعی بر روش پیشنهادی تاگوچی نشان داد، در واقع فلسفه نهفته در روش تاگوچی کاملاً صحیح است و بحث تنها بر سر تکنیک‌هایی است که او استفاده می‌کند [۱۸].

در این تحقیق نیز یک الگوریتم ترکیبی شبکه‌های عصبی - تکامل توام ژنتیکی (NNs-CGA) برای حل مساله طراحی مقاوم با چندین متغیر پاسخ ارائه شده است که می‌تواند فرایند تولید را شبیه‌سازی و مقاوم‌سازی کند. همچنین الگوریتم تکامل توام ژنتیکی (CGA) نیز با الگوریتم ژنتیک (GA) در بهینه‌سازی چند هدفه مقایسه گردیده

است. الگوریتم پیشنهادی در فرایند ریسندگی نیز مورد آزمایش قرار گرفته است. این الگوریتم در محیط MATLAB 7.4 شبیه‌سازی شده است.

بخش دوم این مقاله مروری بر تحقیقات قبلی صورت گرفته در این زمینه دارد. در بخش سوم به مدل سازی فرایند و متغیرهای آن در مطالعه موردی پرداخته شده و در بخش چهارم تقریب توابع با شبکه‌های عصبی ارائه شده است. بخش پنجم مدل ارائه شده با دو الگوریتم GA و CGA حل گردیده که نتایج حاصل در بخش ششم مورد مقایسه و تحلیل قرار گرفته است. نهایتاً در بخش آخر نتیجه‌گیری و پیشنهاد برای تحقیقات آتی آمده است.

## ۲. مروری بر پیشینه تحقیق

همانطور که بیان گردید، اکثر روش‌های سنتی انتخاب پارامتر کامل نیستند و در آن متغیری که از سایرین مهمتر تشخیص داده شود به عنوان تابع هدف بهینه، و سایر متغیرها به محدودیت تبدیل می‌شوند [۷]. روش‌های ابتکاری متعددی برای مساله طراحی پارامتر با چند متغیر پاسخ ارائه شده است. در یکی از روش‌های ارائه شده از وزن دهی به متغیرهای پاسخ استفاده گردید [۲۶]. Pignatiello یک تابع هزینه بر اساس واریانس مربع انحرافات از هدف، برای چند متغیر پاسخ در طراحی پارامتر پیشنهاد داده است [۲۱]. Layne از یک روش مبتنی بر تاگوچی برای حل مساله طراحی پارامتر با بهینه سازی همزمان متغیر استفاده کرد [۱۲]. در تحقیقی دیگر نیز فرایندی با پنج متغیر پاسخ مورد بررسی قرار گرفت که یکی از متغیرهای مذکور به عنوان متغیر اصلی در نظر گرفته شد و بدون توجه به همبستگی محتمل بین متغیرها سایر آنها به ترتیب بهینه گردید [۱۴]. Hsieh در یک تحقیق شرایط بهینه پارامتر، که محصول را در برابر عوامل محیطی مقاوم می‌کند، در فرایند طراحی پارامتر مشخص نمود و سپس طراحی پارامتر تاگوچی را با دو دسته ویژگی‌های استاتیک و دینامیک در طراحی فرایند با چند متغیر بکار گرفت [۸].

در دهه اخیر بکارگیری روش‌های هوش محاسباتی از قبیل شبکه‌های عصبی و الگوریتم‌های تکاملی در حوزه مهندسی کیفیت و به ویژه طراحی پارامتر گسترش یافته است. شبکه‌های عصبی قابلیت یادگیری نگاشت‌های پیچیده و غیرخطی را دارند [۱] و الگوریتم‌های تکاملی قادرند تا توابع مشکل و حتی غیر دیفرانسیل پذیر را بهینه کنند [۱۷].

در سال ۱۹۹۲، Smith از شبکه‌های عصبی برای مدلسازی فرایند تولید جهت تخمین تطابق محصول تولید شده با مشخصه‌های کیفی تعیین شده بهره جست [۲۴]. Ko در فرایند

شکل دهی فلزات از ترکیب شبکه‌های عصبی و آرایه‌های متعامد تاگوچی برای مینیمم کردن تابع هدف مربوط به فرایند شکل دهی استفاده نمود [۱۰]. Bartarya و Choudhury در پیش بینی فرسودگی ابزار تولید با مطالعه اثر پارامترهای فرایند، مثل دما و روغن کاری سطح از روش ترکیبی شبکه‌های عصبی و طراحی آزمایشات استفاده نمودند [۵]. یکی دیگر از تحقیقاتی که از الگوریتمهای ترکیبی هوشمند و سنتی استفاده نمود، در فرایند جوشکاری صنعتی است. در این فرایند از یک شبکه عصبی با الگوریتم پس انتشار خطا و روش تاگوچی برای یافتن سطوح بهینه سرعت جوشکاری، قدرت لیزر و مکان فاصله کانونی برای جوشکاری لیزر Co2 استفاده گردید [۱۹]. از الگوریتم‌های ژنتیک به طرق مختلف در حل مسئله طراحی پارامتر استفاده شده است. Yeniy از یک الگوریتم ژنتیک که Genrank نام گذاری شده بود برای تعیین سطوح بهینه پارامترهای فرایند استفاده نمود و عملکرد آن را با طراحی آزمایشات مقایسه کرد [۲۷]. همچنین یک روش ترکیبی از مدلسازی و شبیه سازی، الگوریتم ژنتیک، روش آنروپی و ماژول قوام برای حل طراحی پارامتر فرایند ارائه شده است [۴] یک روش ترکیبی دیگر از شبکه‌های عصبی و الگوریتم ژنتیک نیز برای حل مسئله انتخاب پارامترهای فرایند تولید به کار گرفته شد [۱۳].

الگوریتم تکامل توام از مدل‌های الهام گرفته از طبیعت است. محققان از همکاری و رقابت بین جمعیت ارگانیزم‌ها در طبیعت الهام گرفته اند تا پویایی تکامل توام را با GA ادغام کنند و بدین طریق الگوریتم‌های محاسباتی قوی تری را ارائه دهند. عامل مشترک در تمامی الگوریتم‌های تکامل توام وجود بیش از یک جمعیت از افراد است [۲۰].

بر اساس مفهوم بیولوژیکی تکامل توام، به دو گروه تقسیم می‌گردد. گروه اول الگوریتم‌هایی هستند که همکاری بین جمعیت‌های مختلف را مدل می‌نمایند. این گروه از الگوریتم‌ها به الگوریتم‌های تکاملی همکاری کننده معروف هستند. گروه دوم الگوریتم‌هایی هستند که بر اساس مفهوم تعامل رقابتی بین گونه‌های مختلف جانوران، توسعه یافته‌اند. این تعاملات می‌توانند بین دو فرد که مانند یک بازی متقارن در تئوری بازی‌ها با هم رقابت می‌کنند [۲۲، ۲۳]، یا جمعیت‌های متعددی از گونه‌های مختلف که با یکدیگر رابطه شکار و شکارچی دارند، صورت پذیرد [۹، ۱۵].

### ۳. مدل سازی فرایند ریسندگی جهت بهینه سازی مقاوم

جهت مقاوم سازی یک فرایند، ابتدا باید پارامترهای مهم آن را که متغیر پاسخ ( $r$ ) نامیده می‌شوند، مشخص کرد. در برخی فرایندها متغیرهای پاسخ متعددی وجود دارند که ورود همه آنها در مدل سبب پیچیدگی زیادی می‌شود. البته کلیه این متغیرها از اهمیت یکسانی برخوردار نیستند. پس از شناسایی متغیرهای پاسخ باید متغیرهای مستقل و تاثیرگذار بر آنها نیز مشخص شوند.

در مطالعه موردی پس از شناسایی ایستگاه‌ها و پارامترهای کنترلی آن، از متخصصین نساجی در خصوص اهمیت هر یک از متغیرهای پاسخ نظرسنجی بعمل آمده و بافندگی به عنوان مشتری داخلی ریسندگی در نظر گرفته شده است. با توجه به سوابق موجود، دلایل پارگیها در چله کشی، بروز عیوب در ریسندگی، معایب پارچه و کنترل متراژ که با شاخصه‌های کیفی نخ مرتبط بودند مورد بررسی قرار گرفته و پارامترهایی که بهبود آنها می‌توانست سبب افزایش رضایت مشتری داخلی (بافندگی) گردد تعیین شد. نهایتاً دو پارامتر استحکام ( $r_1$ ) و نایکنواختی نخ ۲۰ پنبه ( $r_2$ ) به عنوان متغیر پاسخ مد نظر قرار گرفت. سایر عوامل موثر بر متغیرهای پاسخ به دو دسته قابل کنترل ( $x$ ) و غیر قابل کنترل ( $Z$ ) تقسیم شدند که در نگاره شماره (۱) نشان داده شده است.

نگاره ۱. عوامل قابل کنترل و غیر قابل کنترل موثر بر استحکام و یکنواختی نخ

تقسیم بندی عوامل موثر بر روی یکنواختی و استحکام نخ ۲۰ پنبه این ند	
عوامل قابل کنترل	عوامل غیر قابل کنترل
تاب نخ $x_1$	میزان مهارت نیروی انسانی
قطر روتور $x_2$	میزان نپ فتیله
رطوبت سالن $x_3$	دمای سالن
نوع پنبه اولیه $x_4$	تمیزی روتور
میزان نایکنواختی فتیله $x_5$	-
درصد ضایعات فتیله $x_6$	-

### تعریف

از نظر ریاضی، جواب  $X$  را یک جواب مقاوم گوئیم هرگاه در یک شعاع مشخص همسایگی از  $X$  با تغییر  $X$  به اندازه  $\Delta X$  تابع هدف  $f(X)$  دارای تغییرات کمی باشد. یعنی  $X^*$  جواب مقاوم است اگر

$$\forall X \in D_f \ni X \in [X^* - \Delta X^*, X^* + \Delta X^*] \Rightarrow \varepsilon = |f(X) - f(X^*)| \text{ is a small value}$$

برای بدست آوردن جواب مقاوم در یک تابع، به معیاری نیاز داریم که با بهینه‌سازی این معیار به نقطه مقاوم تابع دست یابیم. برای بدست آوردن جواب مقاوم در تابع از معیار واریانس تابع استفاده می‌کنیم [۲۸]. فرض می‌کنیم هر بردار  $X = (x_1, x_2, \dots, x_n)$  که یک جواب را در فضای  $n$  بعدی نشان می‌دهد، یک بردار متغیر تصادفی است زیرا می‌تواند بطور شانسی هر مقداری را اتخاذ کند. تابع  $f(X)$  را حول نقطه میانگین  $\mu = (\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n})$  با بسط مرتبه اول سری تیلور به صورت خطی تقریب می‌زنیم. بنابراین داریم:

$$f(X) \approx f(\mu) + \sum_{i=1}^n \left[ \frac{\partial}{\partial x_i} f(\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n}) (x_i - \mu_i) \right] \quad (1)$$

در رابطه فوق  $f(X)$  نیز یک متغیر تصادفی است زیرا هر تابعی از یک متغیر تصادفی نیز خود یک متغیر تصادفی است. بنابراین می‌توان از طرفین رابطه فوق واریانس گرفت.

$$Var(f(X)) \approx Var(f(\mu)) + Var\left(\sum_{i=1}^n \frac{\partial}{\partial x_i} f(\mu_{x_1}, \mu_{x_2}, \dots, \mu_{x_n}) (x_i - \mu_i)\right) \quad (2)$$

می‌دانیم که

$$Var(x + y) = Var(x) + Var(y) + 2Cov(x, y) \quad (3)$$

با استفاده از قوانین مربوط به واریانس متغیرهای تصادفی، جمله دوم عبارت فوق را بسط می‌دهیم. بنابراین خواهیم داشت:

$$\sigma_f^2 \approx \sum_{i=1}^n \left( \frac{\partial f}{\partial x_i} \right)^2 \sigma_{x_i}^2 + \sum_i \sum_{\substack{j \\ i \neq j}} \left( \frac{\partial f}{\partial x_i} \right) \left( \frac{\partial f}{\partial x_j} \right) \sigma_{x_i x_j} \quad (4)$$

$\sigma_f^2$  واریانس تابع  $f$  و  $\sigma_{x_i}$  انحراف معیار  $i$  امین عنصر در بردار جواب می‌باشد. با توجه به اینکه  $x_i$  و  $x_j$  مستقل از یکدیگر فرض می‌شوند که یک فرض منطقی است، آنگاه  $\sigma_{x_i x_j} = 0$  بنابراین تقریب واریانس تابع بدست می‌آید:

$$\sigma_f^2 \approx \sum_{i=1}^n \left( \frac{\partial f}{\partial x_i} \right)^2 \sigma_{x_i}^2 \quad (5)$$

از رابطه ۵ برای محاسبه انحراف معیار تابع استفاده می‌کنیم تا معیاری ایجاد شود که با کمینه کردن این معیار، نقطه مقاوم تابع بدست آید. تابع زیر بعنوان تابع غیرمقاوم پیشنهاد می‌گردد. با کمینه کردن این تابع، جواب مقاوم برای تابع اصلی،  $f(X)$  بدست می‌آید. یعنی هر چه معیار ذکر شده کمتر باشد جواب بدست آمده از قوام بیشتری برخوردار خواهد بود.

$$R(X) = \frac{1}{n} \sum_{j=1}^n \frac{\hat{\sigma}_f}{\hat{\sigma}_{x_j}} \quad (6)$$

در تابع فوق  $\hat{\sigma}_f$  برآورد انحراف معیار تابع  $f$  است که از رابطه ۶ بدست می‌آید.  $\hat{\sigma}_{x_j}$  نیز برآوردی از انحراف معیار  $x_j$  مولفه بردار  $X$  است. در نهایت مدل به صورت زیر در می‌آید:

$$\begin{aligned} \text{Min } R_1 &= \frac{1}{6} \sum_{j=1}^6 \frac{\hat{\sigma}_{f_1}}{\hat{\sigma}_{x_j}} \\ \text{Min } R_2 &= \frac{1}{6} \sum_{j=1}^6 \frac{\hat{\sigma}_{f_2}}{\hat{\sigma}_{x_j}} \end{aligned} \quad (7)$$

St:

$$850\text{rpm} \leq x_1 \leq 950\text{rpm}$$

$$x_2 = \begin{cases} 0 & \text{if Rotor Diameter} = 35\text{mm} \\ 1 & \text{if Rotor Diameter} = 40\text{mm} \end{cases}$$

$$60\% \leq x_3 \leq 75\%$$

$$x_4 = \begin{cases} 0 & \text{for Cotten type 1} \\ 1 & \text{for Cotten type 2} \end{cases}$$

$$3 \leq x_5 \leq 7$$

$$0\% \leq x_6 \leq 20\%$$

#### ۴. تقریب توابع با شبکه‌های عصبی:

در مدل ارائه شده (رابطه ۷) باید توابع  $f_1$  و  $f_2$  با استفاده از داده‌های تجربی تقریب زده شود. در اینجا قضیه ای مطرح می‌گردد که از نظر تئوری بسیار مهم است. این قضیه به قضیه تقریب ساز جهانی موسوم بوده و توسط افرادی چون سینکو و هورنیک اثبات گردید [۱] قضیه تقریب ساز جهانی ادعا می‌کند که شبکه عصبی پیش خور با یک لایه پنهان که تعداد محدودی نرون دارد با هر تابع تبدیل دلخواه، می‌تواند هر تابعی را در فضای "n" تقریب زده و به عنوان یک تقریب ساز جهانی عمل کند.

#### ۴-۱. پیش پردازی و نرمال سازی داده ها

با استفاده از اطلاعات نمونه ای که از آزمایشات طراحی شده بدست آمده است، ابتدا داده‌ها هم مقیاس و استاندارد می‌شوند. در این قسمت از فرمول نرمال ساز دو قطبی (روابط ۴-۲) استفاده می‌گردد [۱۱]:

$$\hat{X} = mX_i + b \quad (2)$$

$$m = \frac{2}{\text{Max}(X) - \text{Min}(X)} \quad (3)$$

$$b = \frac{\text{Max}(X) + \text{Min}(X)}{2} \quad (4)$$

فرمول فوق کلیه داده‌ها را در فاصله [۱ و -۱] استاندارد می‌نماید.

#### ۴-۲. تعیین ساختار و توپولوژی شبکه عصبی

برای تقریب توابع مطرح شده در فرایند از یک شبکه دو لایه پیشخور که تابع تبدیل لایه پنهان تانزانت هایپربولیک و تابع تبدیل لایه خروجی خطی است، استفاده می‌شود. تعداد ورودی‌ها و خروجی‌های شبکه بر اساس تابعی که در مدل سازی فرایند مطرح گردید مشخص شده است که تعیین کننده تعداد نرون‌های ورودی و خروجی است. در این تحقیق با آزمون و خطا تعداد نرون‌های لایه میانی تعیین شده است.

#### ۴-۳. الگوریتم‌های یادگیری شبکه و مقایسه آنها

الگوریتم‌های متعددی برای آموزش شبکه وجود دارد. در اینجا از الگوریتم پس انتشار خطا استفاده می‌شود. الگوریتم BP استاندارد، یک الگوریتم گرادیان کاهشی است. روش



گرایان کاهش در بهینه سازی در برخورد با اولین بهینه محلی متوقف می‌شود. بنابراین روشهای بر اساس همین الگوریتم توسعه یافتند که به روشهای مبتنی بر جستجوی گرایان مشهور می‌باشند که در BP نیز مورد استفاده قرار گرفته اند. اشکال عمده این الگوریتم‌ها این است که در نقاط بهینه محلی گرفتار می‌شوند. در انتخاب الگوریتم یادگیری، الگوریتم‌های کلاسیک برای سطوح مختلف تعداد نرون با شرایط اولیه یکسان با یکدیگر مقایسه شده است (نگاره شماره ۲ و ۳).

نگاره ۲. مقایسه الگوریتم‌های یادگیری شبکه عصبی برای تقریب استحکام نخ

الگوریتم	تعداد تکرار	معیار mse	msereg <sup>1</sup>	MSE	فاصله اقلیدسی
گرایان کاهش با ممتم	۵۰۰۰	۰,۰۰۵۲۶	۰,۰۱۵۳	۱,۸۱۳۸	۵,۲۱۶۱
گرایان کاهش با نرخ یادگیری متغیر	۵۰۰۰	۰,۰۰۱۳۲	۰,۰۱۰۲	۱,۴۷۶۹	۴,۸۹۶۷
پس انتشار ارتجاعی	۳۴۹۶	۰,۰۰۱	۰,۰۱۰۲	۱,۴۵۸۶	۴,۷۸۹۹
گرایان مزدوج	۲۸۴	۰,۰۰۱	۰,۰۱۰۲	۱,۴۱۹۵۰	۴,۱۶۲۹
شبه نیوتنی	۳۵۶	۰,۰۰۱	۰,۰۱۰۲	۱,۴۴۹۵۰	۴,۶۶۲۹
لونبرگ-مارکوارد	۲۵	۰,۰۰۱	۱,۰۴۲۲	۲۲,۸۳۶۹	۱۸,۵۰۸۱۹۸

نگاره ۳. مقایسه الگوریتم‌های یادگیری شبکه عصبی برای تقریب یکنواختی نخ

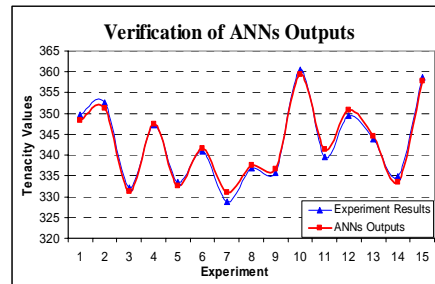
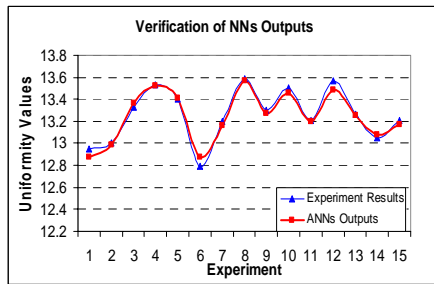
الگوریتم	تعداد تکرار	معیار mse	msereg	MSE	فاصله اقلیدسی
گرایان کاهش با ممتم	۵۰۰۰	۰,۰۰۰۷۸	۰,۰۱۸۷	۰,۰۰۲۵۳	۰,۱۷۵۴
گرایان کاهش با نرخ یادگیری متغیر	۵۰۰۰	۰,۰۰۰۳۷	۰,۰۱۱۶	۰,۰۰۱۸۵	۰,۱۶۶۸
پس انتشار ارتجاعی	۳۱۶۷	۰,۰۰۰۲۶	۰,۰۱۱۶	۰,۰۰۱۹۶	۰,۱۶۷۵
گرایان مزدوج	۲۳۸	۰,۰۰۰۲۵	۰,۰۱۱۶	۰,۰۰۱۷۱	۰,۱۵۹۳
شبه نیوتنی	۳۰۴	۰,۰۰۰۲۶	۰,۰۱۱۶	۰,۰۰۱۸۷	۰,۱۶۷۵
لونبرگ-مارکوارد	۲۵	۰,۰۰۰۲۶	۱,۱۸۰۶۴	۰,۱۹۷۴	۱,۳۹۷۰

۱. معیاری است که برای افزایش قدرت تعمیم پذیری شبکه از آن استفاده می‌شود

$$msereg = \gamma mse + (1 - \gamma) \frac{1}{n} \sum_{j=1}^n w_j^2$$

#### ۴-۴. ارزیابی اعتبار شبکه‌های تقریب زنده

در این تحقیق ارزیابی اعتبار شبکه از اهمیت ویژه‌ای برخوردار است، زیرا در واقع اعتبار سنجی الگوریتم پیشنهادی طراحی فرایند نیز به شمار می‌آید نمودارهای شماره ۳ و ۴ مقایسه بین پاسخ مقادیر واقعی شبکه و مقادیر واقعی استحکام و نایکنواختی که از طریق انجام آزمایش به دست آمده را نشان می‌دهد.



نمودار ۲. سنجش اعتبار شبکه تقریب زنده  
یکنواختی برا مقادیر واقعی

نمودار ۱. سنجش اعتبار شبکه تقریب زنده  
استحکام برا مقادیر واقعی

مقادیر MSE و فاصله اقلیدسی به ازای داده‌های واقعی (نه نرمال شده) برای تمامی الگوریتم‌هایی که در آموزش شبکه‌ها آزمایش شده اند در جداول شماره ۲ و ۳ آورده شده است.

#### ۵. حل مدل‌های فرایند با الگوریتم ژنتیک و تکامل توام

پس از اینکه توابع مورد نظر تقریب زده شدند، به الگوریتمی نیاز است تا بر اساس مدل سازی‌های انجام شده قادر به حل این مدل‌ها باشد. در اینجا الگوریتم‌های ریاضی و کلاسیک بهینه‌سازی کارایی چندانی ندارد زیرا در واقع توابع مورد نظر به صورت جعبه سیاه می‌باشند که بردار متغیرهای تصمیم را دریافت و مقدار تابع را پاسخ می‌دهند. به همین علت به الگوریتم‌هایی نیاز است که قابلیت کار با توابع گسسته را به خوبی داشته باشند که در این بین الگوریتم‌های تکاملی گزینه مناسبی می‌باشند.

### ۱-۵. حل مدل طراحی پارامتر با GA

تکنیک‌های متعددی وجود دارد تا بتوان توابع چندگانه را به یک تابع هدف تبدیل نمود. در اینجا از ساده‌ترین حالت استفاده می‌شود و توابع به صورت ترکیب خطی از یکدیگر در می‌آیند [۲].

برای بهینه‌سازی مساله زیر از GA استفاده می‌شود:

$$\text{Min } f = -\lambda R_1 + (1 - \lambda) R_2$$

St :

$$X \in B$$

$$\lambda \in [0, 1]$$

(۵)

به منظور حل مدل فوق، ابتدا باید ساختار هر کروموزم که یک بردار جواب را نشان می‌دهد، مشخص نمود. در مدل فوق ۴ متغیر پیوسته و ۲ متغیر باینری وجود دارد. متغیرهای پیوسته دارای کران بالا و پایین بوده و باید موجه بودن آنها در طول بهینه‌سازی حفظ شود. از آنجا که تابع ارزیابی برازندگی در این مدل با استفاده از شبکه‌های عصبی محاسبه می‌شود و ورودی شبکه‌های عصبی باید نرمال باشد، لازم است که متغیرهای هر ژن قبل از محاسبه برازندگی کروموزم نرمال گردد. بنابراین در ژن‌هایی که متغیر آنها پیوسته است به جای قرار دادن مقدار واقعی، از مقدار نرمال شده آنها استفاده می‌گردد.

پس از تعیین نوع بازنمایی، مراحل بهینه‌سازی توسط GA به صورت زیر است:

۱. مقدار دهی اولیه: در مقدار دهی اولیه، مقدار پارامترهایی مانند تعداد جمعیت، تعداد نسل، نرخ تقاطع، نرخ جهش و غیره مشخص می‌شود. همچنین یک جمعیت اولیه تصادفی ایجاد می‌شود.

۲. ارزیابی برازندگی: در هر نسل GA باید برازندگی کروموزم‌ها ارزیابی گردد. این ارزیابی توسط شبکه‌های عصبی صورت گرفته و با استفاده از رابطه ۵ برازندگی کروموزم بدست می‌آید.

۳. عملگرهای ژنتیک: انتخاب عملگرهای ژنتیک نیز از اهمیت بالایی برخوردار است. در این تحقیق، از عملگرهای زیر استفاده شده است:

الف) عملگر انتخاب: عملگر انتخاب مورد استفاده، ترکیبی از روش‌های انتخاب تورنمنت و نخه‌پروری است. در شبیه‌سازی‌های صورت گرفته این روش نسبت به سایر روش‌ها، نقش موثرتری در سرعت همگرایی الگوریتم داشته است. ضمن

اینکه فشار انتخابی نیز تا حدودی حفظ می‌شود. تعداد نخبه نیز به طور متغیر از ۵ تا ۱۰ عدد در هر نسل تغییر می‌کند.

ب) عملگر تقاطع: تقاطع در این الگوریتم از یک تا  $n$  نقطه صورت می‌گیرد. بدین صورت که یک عدد تصادفی در بازه [1,5] تولید می‌شود که تعداد نقاطی را که تقاطع باید صورت گیرد، مشخص می‌کند. ژن اول به تصادف از والد اول یا دوم انتخاب شده و سپس با توجه با اینکه تقاطع از چه نقاطی صورت می‌گیرد، بقیه ژن‌های فرزند تشکیل می‌شود. در این تقاطع هر دو کروموزم والد یک کروموزم فرزند می‌دهند.

ج) عملگر جهش: عملگر جهش با متغیرها دو قطبی و پیوسته متفاوت رفتار می‌کند. ابتدا یک ژن به طور تصادفی انتخاب می‌شود، اگر ژن دوم یا چهارم باشد که متغیرهای دو قطبی هستند با اعمال جهش مقدار آنها از ۱ به -۱ و بالعکس تغییر می‌کند. برای سایر ژن‌ها نیز با یک عدد تصادفی با توزیع یکنواخت که در بازه [-0.2,0.2] است به مقدار متغیر ژن مربوطه اضافه می‌شود.

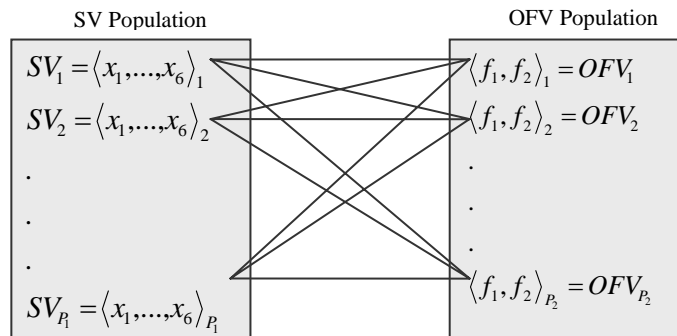
## ۲-۵. حل مدل طراحی پارامتر با CGA

الگوریتمی که در این قسمت بیان می‌شود الگوریتم ژنتیک تکامل توام رقابتی نام داشته و الگوریتم پایه استفاده شده در MOO و برخی از کاربردهای سخت افزاری است (Lohn et al, 1999, 2001). در این تحقیق تعدیلاتی بر روی الگوریتم اولیه صورت گرفته است تا کارایی آن را افزایش دهد. الگوریتم تکامل توام از دو جمعیت مجزا بر خوردار است. یک جمعیت بردار جواب که با جمعیت SV نام گذاری شده است و جمعیت بردار توابع هدف که با جمعیت OFV نشان داده می‌شود. جمعیت SV همان جمعیت کروموزم‌های جواب است که در GA نیز از آن استفاده می‌شود، اما جمعیت OFV کروموزم‌هایی هستند که هر ژن آنها مقدار یک تابع هدف را نشان می‌دهد.

مکانیسم عملکرد این الگوریتم به زبان ساده، تعامل بین دو جمعیت فوق است. در این الگوریتم دو جمعیت به طور مجزا تکامل می‌یابند و عملگرهای تقاطع و جهش بر روی هر کدام به طور جداگانه و همزمان عمل می‌کنند. تعامل آنها به گونه ای است که تکامل آنها را هدفمند کرده و تکامل هر جمعیت بر تکامل دیگر تاثیر گذار است. در این الگوریتم، انتخاب، بر اساس برازندگی صورت می‌پذیرد. تعامل بین این دو جمعیت در

اندازه گیری برازندگی کروموزمها در هر جمعیت ظاهر می‌شود. تعداد اعضای جمعیت SV را  $P_1$  و تعداد اعضای جمعیت OFV را با  $P_2$  نشان داده می‌شود. نمودار شماره (۳) تعامل بین این دو جمعیت را نشان می‌دهد.

در مدل فوق، جمعیت OFV در واقع میزان سختی هدف برای SVها را مشخص می‌کند. جمعیت OFVها در یک طیف آسان تا سخت قرار می‌گیرد. تعیین سختی و آسانی هر OFV بر اساس توان SV در حل آن است. برای محاسبه برازندگی به صورت زیر عمل می‌شود:



نمودار ۳. تعامل دو جمعیت SV و OFV

در جمعیت SVها، به هر یک از افرادی که بتوانند OFVهای سخت‌تری را حل کنند امتیاز بیشتری داده می‌شود. سخت‌ترین OFV، OFVای است که تنها یک فرد از جمعیت SVها بتواند آن را حل کند و به این OFV بالاترین امتیاز تعلق می‌گیرد. OFVهایی که توسط هیچ یک از افراد جمعیت SV حل نمی‌شوند یا آنهایی که براحتی حل می‌شوند، برازندگی خوبی کسب نمی‌کنند. برای محاسبه برازندگی افراد در جمعیت جواب‌ها، ابتدا از رابطه زیر برای فرد  $i$ ام امتیاز  $s_i$  محاسبه می‌شود:

$$s_i = \sum_{j \in \Omega_i} \frac{1}{m_j} \quad ; 0 \leq s_i \leq P_2 \quad (6)$$

در رابطه فوق،  $m_j$  تعداد افراد در جمعیت جواب‌ها است که  $OFV_j$  را حل می‌کنند.  $\Omega_i$  مجموعه‌ای از اندیس‌های OFV است که توسط فرد  $i$ ام در جمعیت SV حل می‌شود. به عبارت ساده هر چه فرد  $i$ ام بتواند تعداد بیشتری از OFVها را حل کند و از طرفی همان OFVها توسط افراد کمتری در جمعیت SV حل شوند، امتیاز بالاتری کسب می‌کند.

به طور مثال اگر  $SV_i$  تمامی OFV ها را حل کند یعنی  $\Omega_i$  شامل تمامی اندیس های افراد در جمعیت OFV باشد  $\Omega_i = \{1,2,3,\dots,P_2\}$  و از طرف دیگر هیچ یک از افراد موجود در جمعیت SV نتوانند هیچ یک از OFV ها را حل کنند، آنگاه  $SV_i$  بالاترین امتیاز ممکن یعنی  $s_i = P_2$  را کسب می کند. به هر صورت برای محاسبه برازندگی  $SV_i$  در جمعیت SV از رابطه ۷ استفاده می شود. در این رابطه مقدار صفر بهترین برازندگی و یک، بدترین برازندگی است.

$$F(SV_i) = \frac{P_2 - s_i}{P_2} \quad ; \quad 0 \leq F(SV_i) \leq 1 \quad (7)$$

مشاهده می شود که  $s_i$  بین حدود بالا و پایین خود به صورت خطی نرمال شده است. رابطه فوق به گونه ای است که اگر OFV توسط تعداد زیادی SV حل شود آنگاه امتیاز بدست آمده بین تمامی SV هایی که OFV مورد نظر را حل نموده اند، تقسیم می شود. برای محاسبه برازندگی افراد در جمعیت OFV،  $y_j$  تعداد اعضایی در جمعیت SV فرض می شوند که  $OFV_j$  را حل می کند و  $P_1$  نیز تعداد اعضای جمعیت SV را مشخص می کند. برازندگی  $OFV_j$  از رابطه ۸ بدست می آید:

$$F(OFV_j) = \begin{cases} 1 & \text{if } y_j = 0 \\ \frac{1}{(P_1 - 1)}(y_j - 1) & \text{if } y_j \geq 1 \end{cases} \quad (8)$$

این رابطه سبب می شود تا فشار انتخاب بر روی کروموزم هایی در جمعیت OFV باشد که سخت هستند و توسط تعداد کمی از افراد جمعیت SV حل می شوند. OFV های آسان برازندگی پایینی دارند.

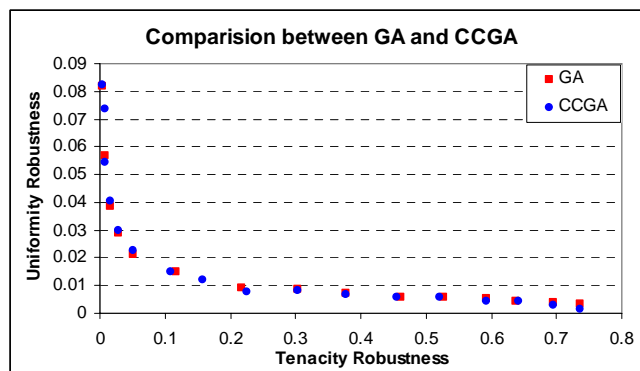
در مطالعه موردی، جهت بهینه سازی مدل اول، CGA نیز مورد آزمایش قرار گرفت. با ایجاد تبادل میان استحکام و یکنواختی نخ، همانند روشی که برای GA مورد استفاده قرار گرفت.

با توجه به اینکه CV معیار یکنواختی نخ است هر چه این معیار کمتر باشد مقدار یکنواختی نخ بهتر است. بنابراین نقاط پایین مرز کارایی را تشکیل می دهند و نقاط بالای آن جواب های غیر مسلط یا غیر کارا هستند زیرا جوابی وجود دارد که بر آنها مسلط است.

### ۶. مقایسه دو الگوریتم GA و CGA در مساله طراحی پارامتر

دو الگوریتم، CGA و GA در یافتن مرز کارایی در طراحی پارامتر با چند متغیر پاسخ با یکدیگر مقایسه شده است (نمودار شماره ۱۹). همانطور که در نمودار ذیل نیز مشخص است با گذشت ۱۰۰۰ نسل از هر دو الگوریتم، CGA مرز کارایی را با تعداد نقاط بیشتر و تا حدی نیز بهتر از GA پیدا کرده است.

در مقایسه دو الگوریتم CGA و GA، ۵۰ شرایط اولیه یکسان، زمان اجرا توسط CGA کوتاه‌تر از GA است. همچنین جواب‌های کارایی که توسط CGA استخراج گردید نسبت به جواب‌های GA تا حدودی مطلوب‌تر است.



نمودار ۴. مقایسه مرز کارایی تولید شده CGA و GA

### ۷. نتیجه‌گیری و پیشنهاد برای تحقیقات آتی

در این مقاله با استفاده از ابزارهای هوش محاسباتی، الگوریتم ارائه شده برای مساله طراحی مقاوم، در یک فرایند واقعی با دو متغیر پاسخ، مورد آزمایش قرار گرفت که نتایج حاصل مورد تایید کارشناسان قرار گرفت. با توجه به نقش بسیار مهم شبکه‌های مورد استفاده جهت تقریب توابع متغیرهای پاسخ، با شرایط اولیه متعدد وزنه‌های سیناپسی و الگوریتم‌های مختلف یادگیری، شبکه‌هایی ایجاد گردید که قابلیت تعمیم‌پذیری خوبی از خود نشان دادند و به خوبی توانستند توابع استحکام و نا یکنواختی را یاد بگیرند. سپس با استفاده از یک الگوریتم تکاملی نوین که تکامل توام را مدل می‌کند مساله مورد نظر حل و نتایج حاصل از این الگوریتم نیز با نتایج الگوریتم ژنتیک مقایسه گردید که برتری نسبی عملکرد تکامل توام را نشان می‌دهد. همچنین بر اساس الگوریتم ارائه شده جداول تصمیمی (مانند نگاره ۵) برای فرایند ریسندگی استخراج گردید که بر اساس آن می‌توان برای هر سطح

تبادل بین دو متغیر پاسخ مورد نظر، ترکیب بهینه عوامل را مشخص نمود علاوه بر آن این جدول مقایسه بین وضع موجود و مطلوب را نشان می‌دهد. همچنین مدل قابلیت تحلیل حساسیت جهت تصمیم سازی مهندسی فرایند را نیز دارد.

تکراه ۴. جدول تصمیم برای تبادل بین استحکام و نایکنواختی و مقایسه با وضع موجود

شرح	تاب نخ	قطر روتور	رطوبت	نوع پنبه	نایکنواختی فتیله	ضایعات فتیله	استحکام	نایکنواختی
وضع موجود	۸۶۰	۳۵mm	٪۵۴	ایرانی	۳	٪۶	۳۶۹,۳۹	۱۲,۷۷۳
وضع پیشنهادی	۸۶۵	۴۰Mm	٪۷۱	ایرانی	۳,۵	٪۵	۳۷۲,۷۶	۱۲,۵۶۷

## منابع

۱. منہاج، محمدباقر "مبانی شبکه‌های عصبی (جلد اول)" انتشارات دانشگاه صنعتی امیرکبیر، چاپ سوم، ۱۳۸۴.
۲. مهرگان، محمد رضا "تصمیم‌گیری با چندین هدف" انتشارات دانشکده مدیریت دانشگاه تهران، ۱۳۸۶.
۳. مونت گمری، داگلاس سی. ترجمه غلامحسین شاهکار "طرح و تحلیل آزمایش‌ها" مرکز نشر دانشگاهی، ۱۳۸۰.
4. Al-Aomar Raid, (2002), "A Robust Simulation-Based Multicriteria Optimization Methodology", Proceedings of the 2002 Winter Simulation Conference.
5. Choudhury S.K., Bartarya G., (2003), "Role of temperature and surface finish in predicting tool wear using neural network and design of experiments", International Journal of Machine Tools & Manufacture 43, pp.747-753.
6. Csáji Balázs Csanádm, (2001), "Approximation with Artificial Neural Networks", MSc thesis, Faculty of Sciences Eötvös Loránd University Hungary.
7. Das P., (1999), "Concurrent optimization of multi-response product performance", Quality Engineering 11 (3), pp.365-368.



8. Hsieh Kun-Lin, Tong Lee-Ing, Chiu Hung-Pin, Yeh Hsin-Ya, (2005), "Optimization of a multi-response problem in Taguchi's dynamic system", *Computers & Industrial Engineering* 49, pp.556–571.
9. Juille H., Pollack J., (1998), "Coevolving the "ideal" trainer: Application to the discovery of cellular automata rules", *Proc.Third Annual Genetic Programming Conf.*
10. Ko Dae-Cheol, Kim Dong-Hwan, Kim Byung-Min, Choi J.C., (1998), "Methodology of preform design considering workability in metal forming by the artificial neural network and Taguchi method", *Journal of Materials Processing Technology* 80–81, pp.487–492.
11. Larose D.T, "Discovering Knowledge in Data", John Wiley, (2005).
12. Layne K.L., (1995), "Method to determine optimum factor levels for multiple responses in the designed experimentation", *Quality Engineering* 7 (4), pp.649–656.
13. Li T.S., Su C.T., Chiang T.L., (2003), "Applying robust multi-response quality engineering for parameter selection using a novel neural–genetic algorithm", *Computers in Industry* 50, pp.113–122.
14. Logothetis N. A. (1998), "Haigh Characterizing and optimizing multi-response processes by the Taguchi method", *Quality and Reliability Engineering International* 4 (2), pp.159–169.
15. Lohn J.D., W. Kraus, G. Haith, (2002), "Comparing a Coevolutionary Genetic Algorithm for Multiobjective Optimization", *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pp. 1157-1162.
16. Lohn J.D., G.L. Haith, S.P. Colombano, D. Stassinopoulos, (1999), "A Comparison of Dynamic Fitness Schedules for Evolutionary Design of Amplifiers" *Proc. of the First NASA/DoD Workshop on Evolvable Hardware*, Pasadena, CA, IEEE Computer Society Press, pp. 87-92.
17. Mitchel M, "An introduction to genetic algorithms", MIT Press, (1999).
18. Montgomery D.C, "Design and Analysis Of Experiments", JohnWiley And SonsInc,fifth Edition .
19. Olabi A.G., Casalino G., Benyounis K.Y., Hashmi M.S.J., (2006), "An ANN and Taguchi algorithms integrated approach to the optimization of CO2 laser welding", *Advances in Engineering Software* 37, pp.643–648.
20. Paredis J., (1995), "The symbiotic evolution of solutions and their representations". Pages 359 365 of: Eshelman, L. (ed), *Proceedings of the sixth international conference on genetic algorithms*. San Mateo, CA: Morgan Kaufmann.

21. Pignatiello J.J., (1993), "Strategy for robust multi-response quality engineering", IIE Transactions 25 (3), pp.5-15.
22. Pollack J., Blair A., Land M., (1996), "Coevolution of a backgammon player", Proc. of Artificial Life 5, Langton, C. (ed), MIT Press.
23. Rosin C.D., (1997), "Coevolutionary Search Among Adversaries", Ph.D. Thesis, University of California, San Diego.
24. Smith Alice E., (1992), "Impacts of Intelligent Process Control on Product Design", ASME Press.
25. Taguchi G., (1987), "System of Experimental Design : Engineering Methods to Optimize Quality and Minimize Cost", UNIPUB, White Plains, New York.
26. Tai C.Y., Chen T.S., Wu M.C., (1992), "An enhanced Taguchi method for optimizing SMT processes", Journal of Electronics Manufacturing 2, pp. 91-100.
27. Yeniay Ozgur, (2001), "A Comparison of the Performances between a Genetic Algorithm and the Taguchi Method over Artificial Problems", Turk J Engin Environ Sci 25, pp.561-568.
28. Jin Yaochu, Sendhoff Bernhard, (2003), "Trade-off between Performance and Robustness: An Evolutionary Multiobjective Approach", Proceedings of the Second International Conference on Evolutionary Multicriterion Optimization, LNCS 2632, Springer, pages 237-251.