

بهینه‌سازی مسئله تخصیص افزونگی با اجزاء تعمیرپذیر غیرنمایی با استفاده از روش

شبیه‌سازی و شبکه‌های عصبی مصنوعی

مجتبی همتی

دانشجوی دکتری مهندسی صنایع، دانشگاه آزاد اسلامی، واحد قزوین، m.hemmati84@yahoo.com

مقصود امیری

(نویسنده مسئول) استاد، گروه مدیریت صنعتی، دانشکده مدیریت و حسابداری، علامه طباطبایی تهران*

مصطفی زندیه

استاد، گروه مدیریت صنایع، دانشکده مدیریت و حسابداری، دانشگاه شهید بهشتی تهران، m_zandieh@sbu.ac.ir

چکیده در این مقاله، یک مدل جدید دو هدفه همراه با یک روش حل جدید برای ارائه به مساله تخصیص افزونگی غیرنمایی در سیستم های سری موازی با اجزای تعمیرپذیر ارائه شده است. روش پیشنهادی بر اساس بهینه سازی از طریق روش شبیه سازی و تکنیک شبکه عصبی مصنوعی است. علاوه بر این، برای واقع گرایی بیشتر، استراتژی‌های تخفیف برای خرید اجزاء در طول مدل سازی استفاده می‌شود. هدف اصلی این مدل، بیشینه سازی مینیمم مقدار متوسط زمان ماندگاری زیر سیستم‌ها و کمینه سازی هزینه کل سیستم بوده و دارای محدودیت‌هایی مانند حجم کل، وزن کل، و تعداد اجزاء مجاز می‌باشد. برای حل این مشکل، برخی از الگوریتم‌های فرا ابتکاری با روش شبیه‌سازی ادغام شده اند. تعدادی مثال عددی برای بررسی رویکرد پیشنهاد شده انجام شد، نتایج نشان می‌دهد، رویکرد پیشنهادی بسیار واقعی تر از روشهای پیشین است و همچنین راه حل های بهینه مطلوب به دست آمده است.

کلمات کلیدی مساله تخصیص افزونگی؛ الگوریتم های متاهورستیک؛ بهینه سازی از طریق شبیه سازی؛ شبکه های عصبی مصنوعی.

۱- مقدمه

در علم مهندسی قابلیت اطمینان، پارامترهای مختلفی به عنوان معیارهای نشانگر وضعیت سیستم تعریف می شوند. مهمترین شاخص عملکرد یک سیستم، قابلیت اطمینان است که در سیستمهای تعمیرپذیر پارامتر دیگری تحت عنوان دسترس پذیری جایگزین قابلیت اطمینان میشود. یکی از روش های موجود جهت ارتقاء پارامترهای عملکرد سیستم بهسازی عملکرد سیستم از طریق توجه به ارتقاء عملکرد اجزاء یا زیر سیستمها و همچنین نحوه چیدمان اجزاء و عناصر سیستم است. استفاده از افزونگی یکی از روشهای ارتقاء عملکرد سیستم است. استفاده از افزونگی به این معنی است که در مرحله طراحی سیستم برای هر یک از اجزاء سیستم یک یا چند جزء کمکی در نظر گرفته شود تا در صورت خرابی قطعه مزبور، قطعه کمکی مانع از کار افتادگی کل سیستم شود. در فرایند بهسازی پارامترهای عملکرد سیستم توجه به نکاتی همچون: تعمیر پذیری قطعات، تابع توزیع احتمال خرابی و تعمیر اجزاء سیستم، تابع هدف و محدودیت ها و

همچنین ساختار سیستم از اهمیت زیادی برخوردار است. توابع توزیع خرابی در مطالعات مشابه نمائی در نظر گرفته شده تا با کمک مفاهیم و معادلات آماری و ریاضی به راحتی مدلسازی شده و قابل تحلیل و بررسی باشد و این در حالی است که در شرایط واقعی ممکن است توابع خرابی و تعمیر قطعات مختلف از توزیع‌های آماری دیگری پیروی نماید. لذا در این تحقیق تلاش می‌گردد ضمن ورود مقوله تعمیر پذیری، مسئله تخصیص افزونگی را با دو هدف بیشینه سازی متوسط زمان ماندگاری سیستم و کمینه سازی هزینه کل، مدل کرده و با استفاده از روش بهینه سازی مبتنی بر شبیه سازی امکان بررسی و تحلیل توابع توزیع غیر نمائی را فراهم نمائیم. ضمن آنکه لحاظ کردن نرخ تخفیف چند سطحی برای اجزاء افزونه از دیگر ویژگیهای تحقیق حاضر محسوب می‌شود.

بررسی مطالعات گذشته نشان می دهد که مساله تخصیص اجزای مازاد، یک مساله بهینه سازی از کلاس NP-hard است [1] که در نتیجه با روش های دقیق و معمول و در زمان ایده آل قابل

* (Corresponding author) amiri@atu.ac.ir

حل نمی باشد، لذا در سالیان اخیر دانشمندان و محققان روش های

حل مختلفی برای آن پیشنهاد نموده اند که عموماً یا مساله را بصورت تک هدفه تحلیل نموده، یا اجزاء را غیر قابل تعمیر در نظر گرفته و یا نرخ خرابی اجزاء را صرفاً دارای توزیع نمائی فرض نموده اند. اما در این تحقیق علاوه بر این بهره گیری از توزیع های غیر نمائی برای نشان دادن نرخ خرابی و تعمیر عناصر سیستم از نوآوری های این تحقیق محسوب شده که امکان استفاده از روش های ریاضی و تحلیلی را برای حل مدل ارائه شده محدود کرده و مستلزم بهره گیری از تکنیک شبیه سازی و شبکه های عصبی مصنوعی می باشد. این رویکرد با طبیعی فرض کردن رویدادهای تصادفی و لحاظ کردن آن در خود، سعی در بهبود امید ریاضی یا متوسط تابع هدف با تلورانس معقول دارد. بطور کلی ضعف تکنیک های تحقیق در عملیات در مدل سازی مسائل تصادفی از یک طرف و ضعف تکنیک شبیه سازی در بهینه سازی مسائل از طرف دیگر، سبب گردیده ترکیب این روشها تکنیک علاوه بر هم افزایی ناشی از مزایای هر یک، نقاط ضعف آنها را نیز پوشش دهد. بهینه سازی از طریق الگوریتم های فرا ابتکاری انجام می شود. بدین منظور الگوریتم فرا ابتکاری چند هدفه توسعه داده می شود. در تمامی مراحل این الگوریتم به منظور ارزیابی جواب ها از شبکه های عصبی مصنوعی استفاده خواهد شد. شکل ۱ چارچوب کلی روش پیشنهادی را نشان می دهد.

مرحله اول: شناسایی عوامل تاثیر گذار بر قابلیت اطمینان سیستم.

مرحله دوم: تعیین سطوح اولیه هر یک از متغیرهای انتخاب شده. برای رسیدن به این منظور دو سطح حداقل و حداکثر برای هر عامل تاثیر گذار تعریف می کنیم.

مرحله سوم: طراحی آزمایش به منظور ایجاد تعداد مناسبی سناریو از ترکیب متغیرها.

مرحله چهارم: شبیه سازی سیستم تحت سناریوهای ایجاد شده و برآورد قابلیت اطمینان. در مورد آزمایش ها طراحی شده، به منظور یافتن مقادیر قابلیت اطمینان، بایستی مدل شبیه سازی را بر اساس آزمایش ها طراحی شده اجرا کرد و مقادیر تابع هدف (قابلیت اطمینان) را برای هر آزمایش ثبت کرد.

مرحله پنجم: برآورد تابع هدف با استفاده از شبکه های عصبی مصنوعی و توسعه متامدل

مرحله ششم: حل متامدل بدست آمده با استفاده از الگوریتم فرا ابتکاری

شکل ۱. چارچوب روش پیشنهادی

در نتیجه این تحقیق تلاش دارد مساله تخصیص افزونگی در سیستم های سری-موازی با اجزای تعمیرپذیر را با رویکردی چند هدفه مورد تحلیل و بررسی قرار .

۲. مرور ادبیات

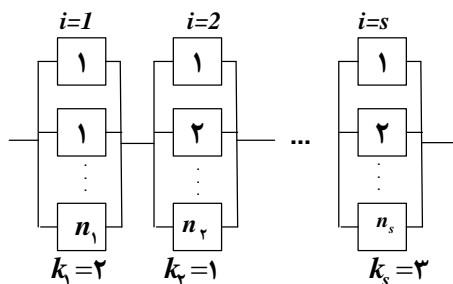
با نگاهی به جریان مطالعات انجام گرفته در مسئله تخصیص افزونگی، به طور آشکار می توان دید که به این مساله در سیستم های تعمیرپذیر پرداخته نشده است. کمبود مطالعات برای مساله تخصیص افزونگی در سیستم های سری-موازی با اجزا و زیرسیستم های تعمیرپذیر در [2] پس از مطالعات کاملی در ادبیات مسئله تخصیص افزونگی آورده شده است. از آنجایی که ساختار سیستم می تواند تاثیر بسزایی روی قابلیت اطمینان سیستم داشته باشد، تعمیرپذیر بودن یا نبودن اجزای سیستم یا زیرسیستم های آن نیز از دیگر موضوعات تاثیرگذار بر عملکرد صحیح یک سیستم در مدت زمان کارکرد آن است [3]. تعمیرپذیر بودن یک سیستم به این معناست که بعد از خرابی می توان با صرف زمان و انجام تعمیرات لازم آن را مجدداً راه اندازی کرد. زمانیکه یک سیستم تعمیرپذیر باشد از واژه «دسترس پذیری» به جای قابلیت اطمینان استفاده می شود. دسترس پذیری به معنای درصدی از زمان است که سیستم تعمیرپذیر به صورتی صحیح وظایف تعریف شده را انجام می دهد [4] با توسعه الگوریتم فرا ابتکاری بر مبنای الگوریتم جستجوی ممنوع، روشی جدید برای حل مسایل بهینه سازی تخصیص افزونگی در سیستم های سری موازی ارائه دادند. این روش برای مدل هایی که امکان انتخاب افزونگی هر یک از زیر سیستم ها از میان چند جزء موجود وجود دارد به کار رفته است. [5] مسایل تخصیص بهینه ای را مدل سازی کرده است که در آنها امکان انتخاب یک گزینه از میان چند گزینه برای هر یک از اجزاء به عنوان افزونگی وجود داشته باشد. در این مطالعه برای حل مدل برنامه ریزی غیر خطی عدد صحیح اینگونه مسایل، از یک رویکرد ساده برنامه ریزی خطی به عنوان روشی برای تقریب جواب بهینه این مسایل استفاده شده است. [6] با اشاره به این موضوع که مسایل بهینه سازی تخصیص افزونگی در سیستم های سری موازی جز مسایل برنامه ریزی غیرخطی عدد صحیح غیر محذب است، الگوریتمی بر اساس روش انشعاب و تحدید جهت حل این مسایل ارائه کردند. این روش بر مبنای حذف فضای جستجو بنا شده است. ها و کو مهمترین مزیت این روش را انعطاف پذیری می دانند که این موضوع باعث می شود این الگوریتم در سایر مسایل بهینه سازی تخصیص افزونگی قابل استفاده باشد. استفاده از مدل در مثال های عددی مسایل تخصیص افزونگی نشان داده که این روش نسبت به

سیستم n جزئی تحت اجزا آماده به کار گرم را مرد بررسی قرار داد به طوریکه در این سیستم اجزا تکی تعمیر پذیر و مابقی سیستم غیر قابل تعمیر در نظر گرفته شده اند. دو مدل جهت محاسبه عملکرد سیستم در این مقاله به کار گرفته شده است که در مدل اول از سیاست جایگزینی بلوک و دیگری بدون استفاده از سیاست جایگزینی بلوک استفاده شده است. [14] در تحقیق خود مدل دو هدفه تخصیص افزونگی در سیستم های سری- موازی تعمیرپذیر را مورد مطالعه قرار داده اند. در این مدل اجزاء دارای نرخ خرابی و تعمیر نمایی هستند مدل ارائه شده توسط الگوریتم رقابت استعماری بهینه سازی شده است.

مرور تحقیقات گذشته نشان می دهد که بررسی مسئله تخصیص افزونگی در حالت چند هدفه، به شکلی که تعمیر مجاز بوده و نرخ خرابی و تعمیر از توزیع غیر نمائی پیروی نماید و همچنین استراتژی تخفیف نیز در مسئله لحاظ شده باشد، کمتر مورد توجه قرار گرفته و فاقد پیشینه مشابه می باشد، لذا در این نوشتار تلاش میگردد با در نظر گرفتن مفروضات اشاره شده، مدلی چند هدفه برای مساله تخصیص افزونگی ارائه و با کمک روشی جدید به نام بهینه سازی مبتنی بر شبیه سازی و شبکه عصبی مدل ارائه شده را حل نماید.

۳. تعریف ساختار مدل:

در این نوشتار سیستمی با ساختار سری- موازی k از n متشکل از s زیرسیستم مستقل از هم مانند شکل ۲ در نظر گرفته شده است. در این ساختار یک سیستم با تعداد s زیر سیستم به صورت سری در کنار یکدیگر قرار گرفته اند و در هر کدام از این زیر سیستمها (s و 2 و 1) جزء n_i مختلف که بصورت موازی قرار گرفته اند تشکیل شده است. برای هر کدام از زیرسیستمها انواع مختلفی از اجزا وجود دارد و در مدل ارائه شده برای هر زیرسیستم فقط از یک نوع جزء می توان استفاده کرد



شکل ۲: ساختار سیستم k از n متشکل از s زیرسیستم مستقل از هم

در مساله تحقیق یک سیستم سری- موازی با n زیرسیستم با مفروضات زیر در نظر گرفته شده است.

روش‌های دقیق برتری‌هایی دارد. برای حل مسایل تخصیص افزونگی در سیستم‌های سری موازی با محدودیت‌های بودجه‌ای و سیستمی و هدف حداکثر سازی قابلیت اطمینان سیستم، از الگوریتم مورچگان استفاده کردند. مساله در نظر گرفته شده در این مدل مساله غیر خطی سخت محسوب می‌شود. در این روش حل از یک روش جستجوی منطقه‌ای برای بهبود کیفیت جواب-های بدست آمده از الگوریتم مورچگان استفاده شده. مثال‌های عددی حل شده در این مطالعه نشان داده‌اند که این الگوریتم از نظر کیفیت جواب‌ها و همچنین میزان زمان مورد نیاز برای حل در وضعیت مناسبی است [7]. [8] الگوریتم جدیدی ارائه کردند که در این الگوریتم محدودیت‌های مختلفی لحاظ شده است. ایده رویکرد ابتکاری ارائه شده در این مطالعه از الگوریتم ژنتیک و الگوریتم آزمندی ریشه گرفته است. ساختار این الگوریتم شامل: ۱- انتخاب تصادفی تعداد اعضای جمعیت اولیه ۲- تخصیص اجزاء بر اساس الگوریتم آزمندی یا انتخاب تصادفی ۳- بهبود جواب‌ها براساس فرایند بازیابی درون سیستمی و بین سیستمی است. در این مطالعه یک مثال عددی مورد استفاده در سایر مطالعات حل شده و نتایج آن با سایر مطالعات مقایسه و ثابت شد که استفاده از این روش نسبت به سایر الگوریتم‌های ابتکاری ارائه شده اثربخش‌تر و کارا تر است. [9] مدل ریاضی جدیدی برای مسئله تخصیص افزونگی را با ساختار سیستم سری- موازی ارائه دادند که در آن امکان تخصیص اجزای غیر یکسان وجود داشت. آنان همچنین سیاست افزونگی فعال و آماده به کار سرد را به عنوان متغیر تصمیم در نظر گرفتند. [10] در مطالعه خود سعی کردند به مبحث بهینه سازی تخصیص افزونگی در سیستم های سری- موازی با در نظر گرفتن قابلیت انتخاب اجزاء متنوع برای هر زیر سیستم و انتخاب اجزاء به صورت فعال یا آماده به کار با لحاظ کردن قابلیت اطمینان کلید های مربوط به سوئیچ قطعات آماده به کار در سیستم پردازند. [11] یک سیستم k -out-of- n با اجزا آماده به کار گرم در حالت تعمیر ناپذیر بررسی کرده اند همچنین در این مقاله طول عمر اجزا نمایی فرض شده است. [12] مساله تخصیص افزونگی را برای یک سیستم سری موازی در نظر گرفته اند. قابلیت انتخاب استراتژی افزونگی به عنوان متغیر تصمیم در این مقاله عنوان شده است که میتوان برای هر زیر سیستم یکی از دو حالت افزونگی فعال یا آماده به کار سرد را انتخاب کرد. آنها در این مقاله با استفاده از الگوریتم تبرید شبیه سازی شده، بدنبال یافتن بهترین استراتژی افزونگی، نوع اجزا و سطح تخصیص افزونگی بوده اند تا با در نظر گرفتن محدودیت های هزینه و وزن کل سیستم، قابلیت اطمینان آنرا بیشینه کند. آنها نتایج بدست آمده از الگوریتم پیشنهادی خود را با نتایج الگوریتم ژنتیک پیشنهاد شده توسط [10] مقایسه کرده اند که نتایج قابل قبولی بدست آورده اند. [13] در تحقیق خود یک

۳-۱ فرضیات مدل:

- فرضیه های مدل چند هدفه این تحقیق به شرح ذیل است:
- اجزاء داخل هر زیرسیستم به صورت موازی قرار دارند.
- طول عمر قطعات (اجزاء) دارای توزیع غیرنمایی است.
- اجزاء تعمیرپذیر بوده و نرخ تعمیر قطعات نیز غیرنمایی است.
- در هر زیرسیستم از انواع مختلف از اجزاء می توان استفاده کرد.
- در داخل یک زیرسیستم فقط از یک نوع جزء می توان استفاده کرد.
- در داخل هر زیرسیستم از تعداد مشخصی از اجزاء می توان استفاده کرد.
- در این سیستم یک تعمیرکار داریم که قادر به تعمیر تمامی اجزاء می باشد.
- هیچ محدودیتی در تعداد اجزای موجود در بازار وجود ندارد.
- سیاست خرید AUD (سیاست تخفیف کلی) استراتژی تخفیف برای خرید برخی از اجزای مورد استفاده قرار می گیرد و سیاست IQD (مقدار نموی) برای خرید سایر اجزای مورد استفاده قرار می گیرد.
- پس از طراحی، تعداد اجزاء در یک زیر سیستم مشخص تغییر نمی کند.

۳-۲ نماد ها، علائم و متغیرهای مساله

با توجه به فرضیه ها، نمادها و شاخص های کاربردی و مدل ریاضی از جمله تابع هدف و محدودیت های این مشکل به شرح زیر است:

- m تعداد زیرسیستم ها
- i اندیس نشانگر شماره زیرسیستم $i = 1, 2, \dots, m$
- n_i تعداد انواع اجزای موجود قابل استفاده برای زیرسیستم i ام
- Π_j تعداد هزینه سناریو برای جزء j ام
- j اندیس نشانگر شماره اجزای $n_i, j = 1, 2, \dots, n_i$
- K تعداد نقاط شکست قیمت در سیاست AUD و IQD
- k شاخص برای نقطه ی شکست قیمت
- C_{ij} هزینه پرداخت شده برای خرید یک جزء از نوع j در زیر سیستم i ام
- X_{ij} تعداد اجزاء انتخاب شده از نوع j ام برای زیرسیستم i ام
- λ_{ij} نرخ خرابی جزء نوع j ام برای زیرسیستم i ام
- μ_{ij} نرخ تعمیر جزء نوع j ام برای زیرسیستم i ام
- q_{ijr} r امین نقطه شکست قیمت جزء نوع j ام برای زیرسیستم i ام

$C_{ijk\pi}$ هزینه مربوط به نقطه شکست k ام قیمت برای خرید جزء j ام در زیر سیستم i ام و سناریوی π

u_{ijk} متغیر باینری است، اگر جزء از نوع j ام در زیر سیستم در نقطه شکست k ام خریداری شود برابر با یک، در غیر این صورت صفر

v_{ij} یک متغیر باینری است، اگر مولفه نوع j ام در زیر سیستم i ام است انتخاب شود برابر با یک در غیر این صورت برابر صفر است

w_{ij} وزن مولفه j ام در دسترس برای زیر سیستم i ام

N_i حد بالای n_i

N_s ماکزیمم تعداد اجزاء بکار در رفته در کل سیستم

B کل بودجه در دسترس

W وزن کل سیستم

M مقدار عددی بزرگ

Z مقدار تابع هدف اصلی یا حداقل MTTFF از سیستم است.

x_{ij} متغیر تصمیم مساله که با استفاده از آن تعداد و نوع اجزای را برای به حداکثر رساندن MTTF و به حداقل رساندن هزینه کل سیستم تعیین می کنیم.

۳-۳ مدل ریاضی:

در این بخش، یک مساله تخصیص افزونگی چند هدفه چند حالتی مدل شده است. برای انجام این کار، از آنجا که یکی از اهداف این است که هزینه کل را به حداقل برسانیم، هزینه خرید اجزای با سیاستهای AUD و IQD اولویت بندی می شود. اجازه دهید رابطه های (۱) و (۲) نقاط شکست قیمت را تحت AUD و IQD تعریف می کنند:

$$c_{ij} = \begin{cases} c_{ij1} & 0 \leq X_{ij} < q_{ij2}, \\ c_{ij2} & q_{ij2} \leq X_{ij} \leq q_{ij3}, \\ \vdots & \vdots \\ c_{ijk} & q_{ijk} \leq X_{ij}, \end{cases} \quad (1)$$

$$\begin{cases} c_{ij1}X_{ij} & 0 \leq X_{ij} \leq q_{ij2}, \\ c_{ij1}q_{ij2} + c_{ij2}(X_{ij} - q_{ij2}) & q_{ij2} \leq X_{ij} \leq q_{ij3}, \\ \vdots & \vdots \\ c_{ij1}q_{ij2} + c_{ij2}(X_{ij} - q_{ij2}) + \dots + c_{ijk}(X_{ij} - q_{ijk}) & q_{ij2} \leq q_{ij3}, \end{cases} \quad (2)$$

هزینه های خرید تحت دو سیاست AUD و IQD در معادلات (۱) و (۲) آمده است. از آنجا که هزینه خرید اجزاء، تصادفی

سیستم بوده، و محدودیت (۹) محدودیت بودجه است. محدودیت تعداد اجزاء کل سیستم در رابطه (۱۰) و محدودیت های ضرورت انتخاب فقط یک نوع جزء برای هر زیرسیستم در رابطه (۱۱) و محدودیت تعداد اجزاء کل سیستم رابطه (۱۲) و همچنین محدودیت رعایت حداقل و حداکثر عضو قابل انتخاب برای هر زیرسیستم برای عملکرد صحیح هر یک از زیرسیستم‌ها لازم است تا حداقل k_i جزء از n_i جزء کارکنند در رابطه (۱۳) بوده و در آخر رابطه های (۱۴) و (۱۵) محدودیت مربوط به نوع متغیرها و حدود آنها را بازگو می کند. معادله (۷) میانگین هزینه خرید اجزاء، بوده بطوریکه قیمت خرید هر واحد کالای احتمالی میباشد. این رابطه برای زمانهای Π_j تکرار می شود و میانگین نتایج به دست آمده در نتیجه معادله محسوب می شود. در این معادله هر واحد کالای نوع j در هر شبیه سازی برای $C_{ijk\pi}$ خریداری شده است.

۴. بهینه سازی بر پایه شبیه سازی و شبکه های عصبی

یک شبکه عصبی آموزش دیده می تواند برای پیش بینی خروجی های متناسب با مجموعه جدید داده ها بکار رود. از مهم ترین ویژگی های شبکه های عصبی مصنوعی وابسته نبودن آن ها به فرضیه های اولیه درباره داده های ورودی است؛ به این معنا که داده های ورودی می توانند هرگونه توزیع آماری دلخواهی داشته باشند. برای ایجاد شبکه عصبی مصنوعی، سناریوهای زیادی از ترکیب سطوح مختلف متغیرها برای اجرا در شبیه سازی طراحی می شود. سپس هر یک از این سناریوها از طریق شبیه سازی کامپیوتری اجرا و نتایج حاصل از شبیه سازی به عنوان متغیر پاسخ در نظر گرفته می شود. در نهایت با استفاده از متغیرهای ورودی و پاسخ، شبکه های عصبی تربیت می شوند به طوری که می تواند قابلیت اطمینان را به خوبی برآورد کند. شبکه عصبی مصنوعی تنها به منظور برآورد تابع هدف به کار می رود و توانایی بهینه یابی ندارد؛ همچنین با توجه به اینکه متامدل پیشنهادی در دسته مسائل NP-HARD هست بهینه سازی از طریق الگوریتم های فرا ابتکاری انجام می شود.

۴-۱. شبکه عصبی پرسپترون چند لایه (MLP)

در بسیاری از مسائل پیچیده ریاضی که به حل معادلات بفرنج غیرخطی منجر می شود. یک شبکه پرسپترون چند لایه می تواند به سادگی با تعریف اوزان و توابع مناسب مورد استفاده قرار می گیرد. توابع فعالیت مختلفی به فراخور اسلوب مسئله در نرون ها مورد استفاده قرار می گیرد. در این نوع شبکه ها از یک لایه ورودی جهت اعمال ورودی های مسئله یک لایه پنهان و یک لایه خروجی که نهایتاً پاسخ های مسئله را ارائه می نماید. استفاده می-

هستند؛ روابط (۳) و (۴) میانگین هزینه های خرید را تحت سناریوهای پیش تعریف شده محاسبه می کنند.

$$PC_{AUD} = \sum_{i=1}^m \sum_{j=1}^{n_i} X_{ij} \left(\text{Mean}_{\pi \in \Pi_j} \sum_{k=1}^K c_{ijk\pi} u_{ijk} \right), \quad (3)$$

$$PC_{IQD} = \sum_{i=1}^m \sum_{j=1}^{n_i} \sum_{k=1}^{K-1} \text{Mean}_{\pi \in \Pi_j} [c_{ijk\pi} u_{ijk} (q_{ijk+1} - q_{ijk})] + \sum_{i=1}^m \sum_{j=1}^{n_i} \text{Mean}_{\pi \in \Pi_j} [c_{ijk\pi} (X_{ij} - q_{ijk}) u_{ijk}]. \quad (4)$$

بنابراین هزینه کل سیستم (TSC) بصورت ذیل می باشد:

$$TSC = \sum_{i=1}^m \sum_{j=1}^{n_i} X_{ij} \left(\text{Mean}_{\pi \in \Pi_j} \sum_{k=1}^K c_{ijk\pi} u_{ijk} \right) + \sum_{i=1}^m \sum_{j=1}^{n_i} \sum_{k=1}^{K-1} \text{Mean}_{\pi \in \Pi_j} [c_{ijk\pi} u_{ijk} (q_{ijk+1} - q_{ijk})] + \sum_{i=1}^m \sum_{j=1}^{n_i} \text{Mean}_{\pi \in \Pi_j} [c_{ijk\pi} (X_{ij} - q_{ijk}) u_{ijk}]. \quad (5)$$

بنابراین مدل ریاضی اصلی بصورت ذیل می باشد

$$\text{Max} Z = \text{Min} \{ MTTFF_T \} \quad (6)$$

$$\text{Min} C = \text{Min} TSC \quad (7)$$

St:

$$\sum_{i=1}^m \sum_{j=1}^{n_i} w_{ij} \cdot X_{ij} \leq W, \quad (8)$$

$$PC_{AUD} + PC_{IQD} \leq B, \quad (9)$$

$$0 \leq \sum_{j=1}^m X_{ij} y_{ij} \leq M \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n_i, \quad (10)$$

$$\sum_{j=1}^{n_i} y_{ij} = 1 \quad \& \quad 0 \leq x_{ij} \leq N_i \cdot y_{ij} \quad i = 1, \dots, m \quad (11)$$

$$\sum_{i=1}^m \sum_{j=1}^{n_i} X_{ij} \leq N_s \quad (12)$$

$$k_i \leq \sum_{j=1}^{n_i} X_{ij} \leq N_i \quad i = 1, \dots, m \quad (13)$$

$$\sum_{k=1}^K u_{ijk} = \begin{cases} 1 & \text{if } X_{ij} > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (14)$$

$$y_{ij} \in \{0, 1\} \quad k_i, x_{ij} \geq 0, \text{ integer} \quad (15)$$

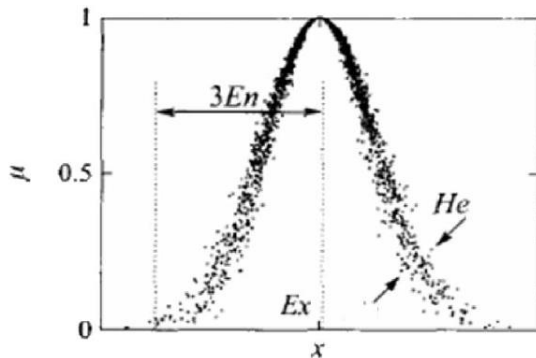
بطوریکه: $MTTFF_T = f(x_{ij}, \lambda_{ij}, \mu_{ij})$

معادله (۶) تابع هدف اصلی یا حداقل MTTFF از سیستم است. تابع هدف دوم این مساله کمینه کردن هزینه کل سیستم در رابطه (۷) نشان داده شده است؛ رابطه (۸) محدودیت حجم

نشان می دهند [15]. Ex مرکز ابر را تعیین می کند، En طیف وسیعی از ابر را تعیین می کند و He نشان دهنده میزان پراکندگی قطرات ابر است. شکل (۵)، نشان می دهد که در حدود ۹۹٫۷۴٪ از تمام قطره ابر بین $Ex - 3En$ و $Ex + 3En$ توزیع شده است. n قطره ابر $(x_1, v_0), \dots, (x_n, v_0)$ می تواند با توزیع نرمال برای یک ابر Y حالت ایجاد شده در صورتی که سه مشخصه دیجیتال (Ex, En, He) و برخی از v_0 شناخته شده است.

INPUT: $\{Ex, En, He\}, n, v_0$
 OUTPUT: $\{(x_1, v_0), \dots, (x_n, v_0)\}$
 FOR $i = 1$ to n
 $En' = randn(En, He)$
 $x_i = Ex \pm En' \sqrt{-2 \ln(v_0)}$
 $drop(x_i, v_0)$
 ENDFOR

جایی که یک عدد تصادفی با توزیع نرمال توسط (En, He) انتظار می رود En و انحراف استاندارد He تولید می شود.



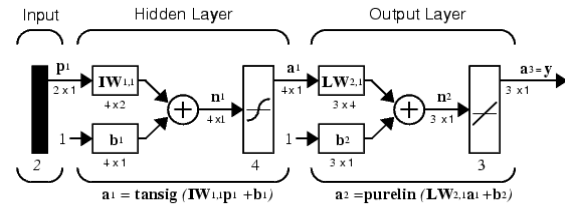
شکل (4): سه ویژگی دیجیتال یک ابر طبیعی [17]

جوابهای بهینه پارتو با استفاده از الگوریتم شبیه سازی شده چندگانه (CBSA) مبتنی بر ابر ساخته می شوند. روش CBSA ما از هر دو رویکرد پارتو و نظریه ابر استفاده می کند. به طور تصادفی یک مجموعه از دمای جدید توزیع شده در اطراف مقدار معینی مانند یک ابر، یک حالت ابر نرمال Y ایجاد می کند، استفاده می شود. به این ترتیب دمای نبرید مداوم تضمین شده و معضل الگوریتم اصلی SA غلبه می شود. بنابراین، ابتدا محدوده ابر، T_k, En را مقداردهی می دهیم؛ درجه پراکندگی قطره ابر، He ، به T_k و مقدار v_0 به $1 - T_k$ سپس معادله (۱۶) و معادله (۱۷) به ترتیب برای محاسبه آنتروپی جدید En' و دمای مرجع جدید T_k' استفاده می شود.

$$En' = En + He - rand(0.1)/3 \quad (16)$$

$$T_k' = En' \sqrt{-2 \ln(v_0)} \quad (17)$$

شود. گره‌هایی که در لایه ورودی هستند. نرون‌های حسی و گره‌های لایه خروجی، نرون‌های پاسخ دهنده هستند. در لایه پنهان نیز نرون‌های پنهان وجود دارند آموزش این شبکه‌ها با روش پس انتشار خطا انجام می شود. نمونه ای از یک شبکه پرسپترون چند لایه در شکل (3) نمایش داده شده است.



شکل (3): ساختار پرسپترون چندلایه با نرون‌های پنهان $tansig$ و نرون‌های خروجی با تابع خطی

شبکه‌های پرسپترون چند لایه می توانند با هر تعداد لایه ساخته و به کار گرفته شوند. ولی قضیه ای که ما در اینجا بدون اثبات می پذیریم. این است که ما بیان می کنیم یک شبکه پرسپترون چند لایه قادر است. هر نوع فضایی را تفکیک کند. این قضیه که قضیه کولوموگروف نامیده می شود. بیانگر مفهوم بسیار مهمی است. که می توان در ساخت شبکه‌های عصبی از آن استفاده کرد. شبکه به کار گرفته شده در این تحقیق یک شبکه پرسپترون چند لایه (MLP) است. که به صورت پیش خور عمل می کند. و در سه لایه، که از ۴۹ ورودی، ۷ لایه پنهان و یک خروجی تشکیل شده است.

۴-۲. الگوریتم شبیه ساز تبرید چند هدفه مبتنی بر تئوری ابر

الگوریتم شبیه ساز تبرید یک روش جستجوی تصادفی براساس استراتژی تکرارشونده مونت کارلو است که از یک هم ارزی قوی بین بهینه سازی ترکیبی و فرایند فیزیکی تبلور بدست می آید. تئوری ابر نوآوری و توسعه تابع عضویت در تئوری فازی است که از دگرگونی عدم اطمینان بین کمیت و کیفیت براساس مفهوم ارزش زمانی به دست می آید. مدل ابر توسط پرفسور دی [15] و بر اساس تئوری فازی سنتی و آمار احتمال ارائه شد. فرض کنید که L ارزش زبانی دامنه v و در مسیر:

$M_D(x): v \rightarrow [0,1], \forall x \in v, x \rightarrow M_D(x)$ اگر توزیع نرمال باشد، آنگاه مدل، ابر نرمال نامیده می شود. این مسیر، گروهی از اعداد تصادفی با گرایش پایا تولید می کند که با امید ریاضی Ex ، بی نظمی En و حد بالای بی نظمی He نمایش داده می شود. این سه پارامتر ویژگی های کمی مفهوم M_D را

یابد. جواب جدید، هنگامی که به دست آمده است، با آنهایی که در آرشیو ذخیره می‌شوند مقایسه می‌شود. اگر هر یک از جوابهای آرشیو آن را غلبه نداشته باشد، می‌توان جابه‌جایی جدیدی را برای ذخیره اضافه کرد، و آن جوابهای که توسط آنهایی که به تازگی افزوده شده اند، حذف خواهند شد. الگوریتم CBSA دمای اولیه T_0 را به عنوان پارامتر ورودی اصلی تعیین می‌کند. برای رسیدن به یک احتمال پذیرش بالا، دمای اولیه مناسب T_0 باید به اندازه کافی بالا باشد. در اینجا T_0 را با توجه به تفاوت بین مقادیر تابع هدف در CBSA تعیین می‌کنیم، زیرا $prob$ به T_0 و تفاوت بین تابع هدف بستگی دارد. اول، یک مجموعه بزرگ از جواب‌ها به طور تصادفی تولید می‌شود. سپس دمای اولیه بر اساس فرمول زیر محاسبه می‌شود:

$$T_0 = \max_{i=1, \dots, nobj} 1 - \text{arc cot} \left(\frac{St.Dev_i}{\pi} \right) \quad (19)$$

بطوریکه $St.Dev_i$ انحراف استاندارد تابع هدف i است و $nobj$ تعداد تابع هدف است. دمای اولیه حداکثر مقدار احتمالی T_0 است، که منجر به افزایش احتمال پذیرش می‌شود که در معادله (۱۹) دیده می‌شود.

۴-۲-۱ عملگر همسایگی

همانند عملگر جهش می‌باشد که در ادامه شرح داده خواهد شد.

۴-۲-۲ برنامه سرد کردن

قسمت‌های تشکیل دهنده برنامه سرد کردن عبارتند از:

۴-۲-۳ درجه حرارت آغازین:

درجه حرارت آغازین باید به اندازه کافی گرم باشد تا حرکت به حالت مجاور را اجازه دهد. پیدا کردن درجه حرارت آغازین مناسب مشکل هست و روش مشخصی برای مسائل مختلف ندارد. در ابتدا تعداد زیادی از جواب‌های تصادفی تولید کرده و تابع هدف آنها را تعیین می‌کنیم؛ سپس میزان انحراف استاندارد موجود در نتایج بدست آمده را محاسبه کرده و از آن برای تعیین دمای اولیه استفاده می‌کنیم. در الگوریتم پیشنهادی از ۱.۵ برابر انحراف استاندارد موجود در جواب‌های اولیه برای تعیین جواب اولیه استفاده شده است. یادآور می‌شود که برای هر یک از اهداف دمای جداگانه در نظر خواهیم گرفت.

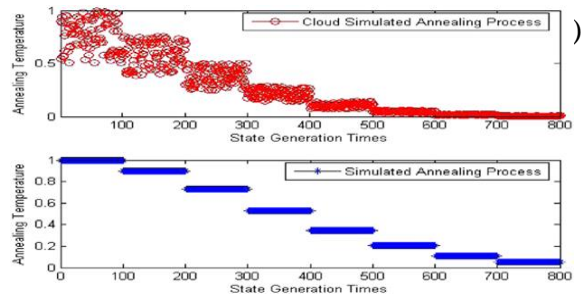
۴-۲-۳-۱ درجه حرارت پایانی:

درجه حرارت کاهش یابد تا زمانی که به صفر حدی برسد. همچنین معیار توقف می‌تواند یک درجه حرارت پایین مناسب باشد.

۴-۲-۳-۲ کاهش درجه حرارت در هر مرحله:

معمولاً می‌توان با یک رابطه خطی ساده (یک تناوب هندسی) کاهش دما را بدست آورد:

معادله (۱۶)، یک عدد تصادفی بین ۰ و ۱ توسط $(0,1)$ تولید می‌شود. دمای تبرید که دمای جدیدی را که در بالا شرح داده شده است را تولید می‌کند، به طور تصادفی در اطراف مرجع داده شده، مانند یک ابر، به تنهایی تغییر می‌کند. بنابراین الگوریتم، حفظ تنوع جستجو، می‌تواند از افتادن در یک بهینه محلی بگریزد [15]. علاوه بر این، بهترین جوابها حفظ می‌شوند و موقعیت‌های بهینه به سرعت از طریق گرایش پایدار دمای تبرید ثابت می‌شوند [17]. در شکل (5) یک فرایند تبرید مبتنی بر ابر و فرایند تبرید عمومی را زمانی که تابع به روز رسانی دما برابر با $\lambda = 0.9$ مقایسه می‌شود. شکل (5) چگونگی تغییرات دما را در حالت ابر نشان می‌دهد. در این شرایط زمانی که سیستم در حالت تعادل قرار دارد، دمای سیستم بر اساس نظریه ابر به شکل متراکم و حول یک نقطه تولید خواهد شد.



شکل (5): چگونگی تغییر دما در الگوریتم شبیه‌سازی تبرید مبتنی بر تئوری ابر

برای ذخیره در جوابهای پارتو در طی فرایند جستجو، یک مجموعه‌ای به نام آرشیو تعریف می‌شود از آنجا که الگوریتم SA قادر به بازگشت پارتو یا بهینه، در یک اجرا نیست. قوانین تصمیم‌گیری زیر برای پذیرش یا رد یک جواب استفاده می‌شود. اصلاح بخش تصادفی انتخاب شده جواب فعلی، X_{cur} در هر تکرار، جواب جدید همسایگی، X_{new} تولید می‌شود. X_{cur} توسط X_{new} غلبه شده و X_{new} به طور خودکار پذیرفته می‌شود، اگر تنها جواب حاصل X_{new} برای هر تابع هدف بهتر از X_{cur} باشد. اگر X_{new} غلبه نداشته باشد بر X_{cur} یا حتی اگر X_{new} توسط X_{cur} تحت سلطه باشد، الگوریتم می‌تواند راه حل X_{new} را با احتمال پذیرش با استفاده از معادله قبول کند.

$$\delta = \exp \left(\frac{- \left(\min_{i=1, \dots, nobj} \left| \frac{F_i(X_{new}) - F_i(X_{cur})}{F_i(X_{cur})} \right| \right)}{T'_k} \right) \quad (18)$$

جایی که $F_i(X)$ ارزش تابع هدف j th جواب X است، $nobj$ تعداد تابع هدف است و T'_k دما در مرحله k است. بر اساس معادله (۱۸) به منظور محاسبه احتمال پذیرش، حداقل اختلاف بین دو جواب می‌تواند مورد استفاده قرار گیرد و به این ترتیب احتمال پذیرش جوابهایی که بهتر از در حال حاضر نیست افزایش می‌یابد و همچنین تنوع راه حل‌های مورد بررسی افزایش می‌یابد.

۴-۳-۶ شرط توقف

معیار توقف در این الگوریتم رسیدن به یک تکرار نسل مشخص می‌باشد. لازم به ذکر است که جهت افزایش کارایی الگوریتم پیشنهادی برای مدل ارایه شده، پارامترهای ورودی الگوریتم با استفاده از روش‌های تنظیم پارامتر در بهترین مقدار خود تنظیم می‌گردند.

۴-۲-۷ نمایش جواب

گام اولیه در اجرای الگوریتم، نمایش جواب یا طراحی کروموزوم است. کروموزوم باید طوری طراحی شود که تا حد ممکن محدودیت‌های اصلی مساله را برآورده سازد. کروموزوم طراحی شده در این تحقیق یک ماتریس $N \times T$ است که T تعداد سطرها مربوط به نوع و تعداد اجزا و N تعداد زیرسیستم‌ها می‌باشد. هر ستون نشان دهنده یک زیرسیستم و مقدار هر ژن در سطر اول نشان دهنده نوع و در سطر دوم نشان دهنده تعداد اجزاء در زیرسیستم مربوطه می‌باشد.

شکل (6): نمایش جواب

به عنوان مثال شکل (6) نشان دهنده الگوی چیدمان در یک سیستم با ۴ زیرسیستم است که در زیرسیستم اول ۴ جزء از نوع دوم، در زیرسیستم دوم ۱ جزء از نوع سوم، در زیرسیستم سوم ۳ جزء از نوع دوم و در زیرسیستم چهارم ۲ جزء از نوع اول قرار دارد.

۴-۳ الگوریتم ژنتیک مرتب سازی نامغلوب (NSGA-II)

این الگوریتم توسط دب و همکاران ارائه گردیده و دارا بودن یک رویکرد روشن برای فراهم آوردن چگالی در بین جواب‌های بهینه پارتو مزیت اصلی الگوریتم محسوب می‌شود [18]. در این روش ابتدا جمعیت فرزندان (Q_t) با استفاده از جمعیت والدین (P_t) ساخته می‌شود. پس از ادغام جمعیت‌ها، از مرتب سازی نامغلوب جهت دسته بندی تمام اعضای جمعیت استفاده می‌شود که این دسته بندی به وسیله صفوف F_1, F_2, \dots, F_n نشان داده می‌شود. اعضای صف F_1 اعضای هستند که اعضای سایر صفوف را مغلوب کرده اند و اعضای آخرین صف اعضایی هستند که در رقابت با اعضای سایر صفوف مغلوب شده اند. حال برای تولید جمعیت جدید (P_{t+1})، اعضای صفوف ابتدایی در جمعیت P_{t+1} قرار داده می‌شود. زمانی که تعداد جمعیت جدید به N رسید، روند قرار دادن اعضای صفوف در جمعیت جدید متوقف می‌شود. در صورتی که صفی وجود داشته باشد که با قرار دادن آن در

$$T_k = \alpha T_{k-1} \quad (20)$$

تجربه نشان می‌دهد باید α بین ۰.۸ تا ۰.۹۹۹ باشد تا بهترین نتیجه بدست آید و الگوریتم طولانی نشود. پارامتر α از طریق تحلیل آزمایشات تعیین می‌شود. البته همان طور که بیان شده در الگوریتم SA مبتنی بر تئوری ابر، دما در هر سطح تعادلی ثابت نبوده و می‌بایست تغییرات جزئی داشته باشد. در این نوشتار از رابطه خطی زیر برای کاهش دما استفاده شده است:

$$En = T_k \cdot He = T_k \cdot v_0 = 1 - T_k \quad (21)$$

$$En' = En + He - rand(0.1)/3$$

$$T'_k = En' \sqrt{-2 \ln(v_0)}$$

در این رابطه T_k مقدار دمای پایه در یک تعادل خاص و T'_k دمای استفاده شده در تابع پذیرش است. تابع $RAND[a, b]$ یک عدد تصادفی در بازه a تا b تولید می‌کند.

Type of component selected	Subsystem			
	2	3	2	1
Number of component	4	1	3	2

۴-۲-۴ تکرار در هر دما:

تعداد تکرار در هر دما از طریق تجزیه و تحلیل آزمایش‌ها تعیین خواهد شد.

۴-۲-۵ معیار پذیرش (یک حرکت)

در الگوریتم‌های بهینه‌سازی محلی، جواب جدید تنها در صورت بهبود تابع هدف پذیرفته می‌شود. این در حالیست که در SA نه تنها جوابی که باعث بهبود تابع هدف می‌شود پذیرفته می‌شود بلکه جواب‌های نامناسب نیز بطور احتمالی پذیرفته می‌شوند، احتمال پذیرفته شدن حرکت بد توسط رابطه‌ی زیر محاسبه می‌شود:

$$P1 = \exp\left(\frac{-\Delta C1}{t1}\right)r \quad (22)$$

که در اینجا $\Delta C1$ اختلاف جواب فعلی و جواب همسایه در توابع ارزیابی است؛ $t1$ دمای فعلی سیستم می‌باشد. r یک عدد تصادفی بین صفر و یک و $p1$ احتمال حرکت به جواب جدید است. در صورتی که جواب جدید از جواب فعلی بهتر باشد و یا مقدار تابع احتمال حرکت از یک عدد تصادفی از دامنه $[0,1]$ بزرگ‌تر باشد، حرکت به جواب جدید انجام خواهد یافت. در غیر اینصورت جستجوگر جواب جدید دیگری را تولید و ارزیابی خواهد نمود. این حرکت گام به گام تا رسیدن به شرط توقف الگوریتم ادامه می‌یابد. یک مسئله‌ی مهم در الگوریتم پیشنهادی SA بررسی شرط تعادل و شرط توقف الگوریتم پیشنهادی است.

ادامه مرتب سازی نامغلوب نیاز نیست. این روش به کاهش زمان اجرای الگوریتم کمک می‌کند. در مراحل ابتدایی تکامل، ممکن است جواب‌ها در تعداد زیادی مرز پارتو متفاوت قرار داشته باشند. اما در نسل‌های بعدی ممکن است جواب‌های زیادی متعلق به بهترین مرز ناچیره باشند و حتی تعداد آنها از N تجاوز کند. زمانی که کل جمعیت به سوی مرز بهینه پارتو همگرا می‌شود، الگوریتم وجود یک توزیع خوب در بین جواب‌های برگزیده را تضمین می‌کند.

۴-۳-۱ استراژی تقاطع

عملکرد اصلی جهت تولید کروموزوم‌های جدید در الگوریتم ژنتیک، عملگر تقاطع می‌باشد. یکی از نقاط قابل توجه، استفاده از یک عملگر مناسب است که جواب را از حالت موجه خارج نکند. برای انجام عمل تقاطع ضمن رعایت شرط موجه بودن فرزندان ایجاد شده، مراحل زیر را دنبال می‌کنیم:

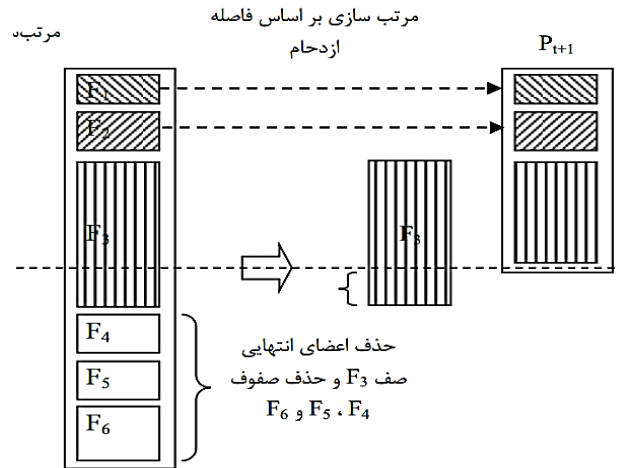
- اولین کروموزوم موجود در جمعیت انتخاب می‌شود.
- یک عدد تصادفی r بین صفر و یک تولید کنید.
- اگر $r < PC$ باشد، آن کروموزوم را برای انجام عمل تقاطع انتخاب کنید.
- اکنون کروموزوم‌های انتخابی را به صورت تصادفی با یکدیگر جفت می‌زنیم: برای هر جفت کروموزوم یک عدد تصادفی $Cross\ point$ در فاصله $\{1...N\}$ تولید می‌کنیم که N تعداد زیر سیستم‌ها است.
- بیت‌های ۱ تا $Cross\ point$ موجود در کروموزوم والد اول را مستقیماً در ژن‌های فرزند اول کپی می‌کنیم.
- بیت‌های $Cross\ point+1$ تا N والد دوم، بر طبق ترتیب قرارگیری آنها در والد دوم، به فرزند اول انتقال داده می‌شوند.
- گام‌های ۵ و ۶ متقابلاً برای تولید فرزند دوم تکرار می‌شوند.
- گام‌های فوق برای تولید سایر کروموزوم‌های فرزند تکرار می‌شوند.

Offspring 1	<table border="1"><tr><td>3</td><td>1</td><td>2</td><td>2</td></tr><tr><td>1</td><td>5</td><td>4</td><td>3</td></tr></table>	3	1	2	2	1	5	4	3	Parent 1	<table border="1"><tr><td>3</td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>5</td><td>4</td><td>2</td></tr></table>	3	1	2	3	1	5	4	2
3	1	2	2																
1	5	4	3																
3	1	2	3																
1	5	4	2																
Offspring 2	<table border="1"><tr><td>3</td><td>1</td><td>3</td><td>3</td></tr><tr><td>3</td><td>4</td><td>2</td><td>2</td></tr></table>	3	1	3	3	3	4	2	2	Parent 2	<table border="1"><tr><td>3</td><td>1</td><td>3</td><td>2</td></tr><tr><td>3</td><td>4</td><td>2</td><td>3</td></tr></table>	3	1	3	2	3	4	2	3
3	1	3	3																
3	4	2	2																
3	1	3	2																
3	4	2	3																

شکل (۸) عملگر تقاطع

به عنوان مثال شکل (۸) در تقاطع دو کروموزوم به عنوان والد، عدد ۳ بصورت تصادفی به عنوان نقطه تقاطع انتخاب شده است.

جمعیت جدید، اعضای جمعیت به عددی بیش از N برسد، باید اعضای آن صف بر اساس فاصله ازدحام مرتب شده و برای قرار گرفتن در جمعیت جدید اعضای با فاصله ازدحام بیشتر در اولویت قرار خواهند داشت. شکل (۷) روند انجام الگوریتم را نشان می‌دهد [18].



شکل (۷): روش کار الگوریتم ژنتیک مرتب سازی نامغلوب

بنابراین جهت انجام الگوریتم ژنتیک مرتب سازی نامغلوب گام‌های زیر انجام می‌شود:

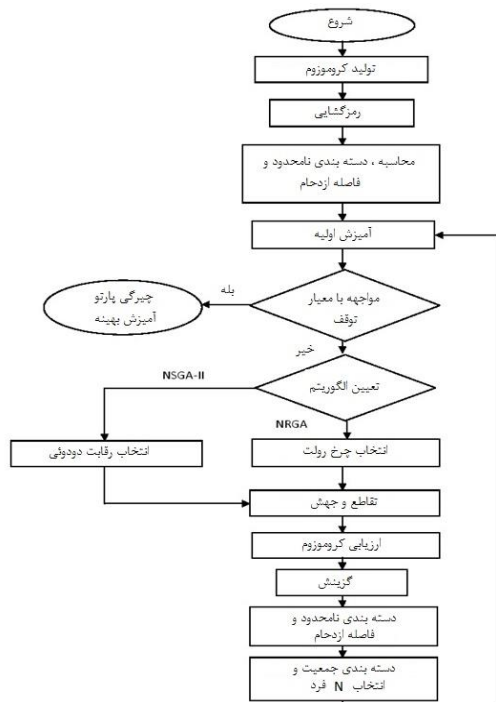
- جمعیت والدین (P_t) و جمعیت فرزندان (C_t) با هم ادغام شده و جمعیت (R_t) را پدید می‌آورند. اکنون باید N جواب برتر از R_t انتخاب شده و جمعیت والدین جدید P_{t+1} را تشکیل دهند.
 - عملیات مرتب سازی نامغلوب روی R_t و تعیین مرزهای مختلف آن که با ($F_i : i=1,2,\dots$) نشان داده می‌شوند، مشخص می‌گردند.
 - روند قرار گرفتن جواب‌ها در جمعیت، از جواب‌های متعلق به مرز نخست یعنی F_1 آغاز می‌شود و تا زمانی که با اضافه شدن اعضای هر مرز، تعداد اعضای جمعیت همچنان از N کمتر باشد، این روند ادامه می‌یابد.
 - اگر مجموع اعضای مرز i ام با تعداد اعضای کنونی جمعیت از N بیشتر شود یعنی: $|F_i| + |P_{t+1}| > N$ فاصله تراکم هر جواب محاسبه می‌شود و تعداد ($N - |P_{t+1}|$) تا از جواب‌ها که دارای بیشترین فاصله تراکم هستند به جمعیت افزوده می‌شوند.
 - جمعیت فرزند جدید (Q_{t+1}) با به کار گیری عملگرهای انتخاب، تقاطع و جهش روی P_{t+1} پدید می‌آید.
- البته می‌توان گام‌های اول و دوم را با هم انجام داد. در این روش مرزهای پارتو و جواب‌های نامغلوب به ترتیب پیدا شده و بررسی می‌شود که آیا با اضافه شدن اعضای آن مرز هنوز اندازه جمعیت از N کوچک‌تر است یا نه؛ اگر این گونه نباشد، دیگر به

۴-۳-۲ استراتژی جهش

در این معادله $rank_i$ رتبه مرز i ام، Nf تعداد مرزهای مشخص شده در مرحله مرتب کردن غیر مغلوب ها می باشد. واضح است که جواب هایی که در مرزهای بهتر قرار دارند احتمال انتخاب بیشتری به خود می گیرند. احتمال انتخاب جواب j ام موجود در مرز غیر مغلوب i ام به صورت رابطه ۲۴ محاسبه می شود:

$$p_{ji} = \frac{2 \times rank_{ji}}{N_j \times (N_j + 1)} = \frac{rank_{ji}}{\sum_{j=1}^p rank_{ji}} \quad (24)$$

در این معادله، N_j نشان دهنده تعداد جواب های موجود در مرز j ام و $rank_{ji}$ نشان دهنده رتبه جواب j ام در مرز i ام بر اساس فاصله ازدحامی است. در این معادله نیز واضح است که جواب های با فاصله ازدحام بیشتر، احتمال انتخاب بیشتری پیدا می کنند. چرخ رولت بدین صورت است که ابتدا بر روی دو بازه حقیقی $[0, S1]$ و $[0, S2]$ ، $S1 = \sum_{i=1}^n p_i$ و $S2 = \sum_{j=1}^m p_{0j}$ تعریف می شود. سپس جواب های موجود در هر مرز نیز بر اساس مقدار احتمال انتخابشان مقدار از بازه $[0, S1]$ و $[0, S2]$ را اشغال می کنند. سپس دو عدد تصادفی بین ۰ و ۱ انتخاب شده و عدد تصادفی اول برای انتخاب مرز در بازه $[0, S1]$ و عدد تصادفی دوم برای انتخاب یکی از جواب های موجود در مرز انتخاب شده در بازه $[0, S2]$ به کار می رود. با توجه به توضیحات داده شده درباره دو الگوریتم NSGA-II و NPGA فلوجارت خلاصه شده این دو الگوریتم به صورت شکل (۱۰) می باشد.



مهمترین وظیفه عملگر جهش اجتناب از همگرایی و بهینه محلی و جستجو در فضاهای دست نخورده مساله است. برای اعمال جهش بر روی کروموزوم ها، از گام های زیر پیروی می کنیم:

- اولین کروموزوم موجود در جمعیت انتخاب می شود.
- یک عدد تصادفی حقیقی در فاصله صفر و یک تولید کنید.
- اگر $r < P_m$ باشد کروموزوم مورد نظر تحت عملگر جهش قرار می گیرد. به این صورت که یکی از ژن های مربوط به نوع و یا تعداد اجزاء در یک زیرسیستم انتخاب شده و با مقدار دیگری که به صورت تصادفی و موجه تولید شده، جایگزین می گردد.

2	1	2	3
1	5	4	2

➔

2	1	2	3
1	5	2	2

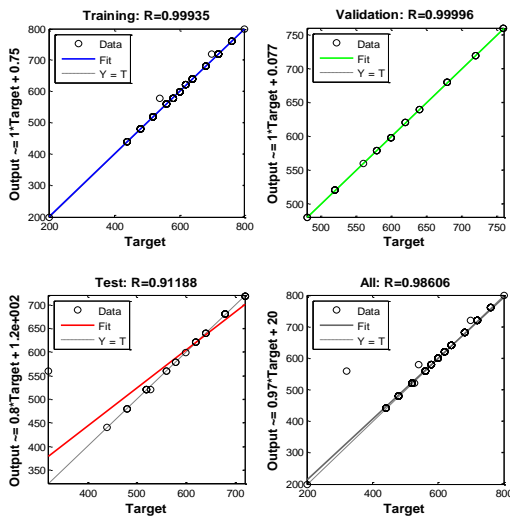
شکل (۹) عملگر جهش

در شکل (۹) عملگر جهش به یکی از ژن های زیر سیستم سوم اعمال شده و مقدار آن را از ۴ به ۲ تغییر داده است.

۴-۴ الگوریتم ژنتیک رتبه بندی کردن نامغلوب ها (NRGA)

با وجود دست آوردهای خوب در توسعه الگوریتم های چند هدفه محققان هنوز در توسعه الگوریتم های جدید و یا بهبود الگوریتم های موجود، معمولاً یا اپراتور جدیدی به الگوریتم افزوده می شود و یا اپراتور موجود بهبود داده می شود. یکی از اپراتورهای محلی و نزدیکی هر چه بیشتر به بهینه های جهانی شود. در الگوریتم های چند هدفه نیز این مهم می تواند نقش خود را در قالب نزدیک سازی هر چه بیشتر الگوریتم به پارتوی بهینه ایفا کند. در همین راستا و به منظور ارتقای عملکرد الگوریتم NSGA-II، الگوریتم انتخاب بر اساس چرخ رولت را بکار برده که در آن به هر جواب یک مقدار برازشی مساوی با رتبه آن جواب در جمعیت تخصیص داده می شود. همانطور که در بالا توضیح داده شد هر عضو جمعیت دارای دو مشخصه رتبه مرز غیر مغلوبی که در آن قرار دارد و رتبه خودش درون مرز که بر اساس فاصله ازدحام می باشد. بنابراین برای انتخاب یک جواب، ابتدا باید یک مرز غیر مغلوب انتخاب شود و سپس درون آن مرز یک جواب انتخاب شود. احتمال انتخاب مرز غیر مغلوب i ام به صورت رابطه ۲۳ محاسبه می شود:

ما یک شبکه عصبی مصنوعی تربیت شده در اختیار داریم که می‌تواند مقادیر تابع هدف را به خوبی تخمین بزند.



شکل (11): کارایی شبکه عصبی در برازش داده‌ها در مدل تحقیق

۵. نتایج محاسباتی و مقایسه‌ها

در این بخش، برخی از نمونه‌های عددی برای اعتبار سنجی نتایج حاصل از CBSA و نیز مقایسه عملکرد آن‌ها با نتایج حاصل از NSGA-II و NPGA مورد ارزیابی قرار می‌گیرد. اجرای الگوریتم‌ها بر اساس چند معیار سنجش روشهای چندهدفه اندازه‌گیری می‌شود، که در آخر بر اساس نتایج آماری و گرافیکی مقایسه می‌شوند. مسائل نمونه شامل ۱۴ و ۲۰ زیر سیستم است که هر کدام شامل سه یا چهار جزء بوده و حداکثر تعداد اجزاء در یک زیرسیستم شش یا هشت عدد می‌باشد. با در نظر گرفتن حداکثر تعداد اجزاء در یک زیرسیستم، $4 = 2 \times 2$ اندازه متفاوت وجود دارد. هشت مساله برای هر اندازه مساله وجود دارد. بنابراین، ۳۰ مساله آزمایشی وجود دارد که مسایل ۱-۸ مربوط به ۱۴ زیر سیستم است که هر زیرسیستم به ۶ جزء محدود می‌شود، مسائل ۹-۱۶ مربوط به ۱۴ زیر سیستم است که در آن هر زیرسیستم محدود به هشت جزء و غیره می‌باشد.

اطلاعات دقیق در مورد مساله نمونه در جدول (3) ارائه شده است. علاوه بر این، نقطه شکست اجزاء در استراتژی AUD زیر نشان داده شده است. نقطه شکست قیمت در استراتژی IQD شبیه به IQD است:

$$\begin{cases} 2.850 \leq X_{ij} < 2 \\ 2.52 \leq X_{ij} < 4 \\ 24 \leq X_{ij} \end{cases} \quad (26)$$

با توجه به وجود 5 زیرسیستم نیازمند 5 متغیر تصمیم برای تعیین تعداد اجزاء در هر زیرسیستم و 5 متغیر تصمیم برای تعیین نوع اجزاء در هر زیرسیستم می‌باشیم. سطوح متغیرهای

شکل (10): فلوجارت پیشنهادی برای الگوریتم‌های NSGA-II و NPGA

۴-۵. ارزیابی تابع هدف اول:

یک سیستم سری - موازی برای اولین بار زمانی از کار می‌افتد که یکی از زیرسیستم‌های آن از کار بیفتد؛ به عبارتی هر کدام از زیرسیستم‌های سیستم که زودتر از دیگر زیرسیستم‌ها از کار بیفتد کل سیستم خراب می‌شود. از این رو برای محاسبه MTTFF سیستم کافی است MTTFF هر یک از زیرسیستم‌ها شبیه‌سازی شده و مقدار MTTFF زیرسیستمی که کمترین زمان قبل از اولین خرابی را دارد به عنوان MTTFF کل سیستم در نظر گرفته شود. از آنجایی که تمامی روابط و معادلات برای محاسبه MTTFF یک سیستم با احتمالات خرابی و تعمیر نمایی به کمک تحلیل‌های ریاضی و آماری امکانپذیر می‌باشد و در صورت غیر نمایی بودن این امکان وجود ندارد، از این رو از رویکرد شبیه‌سازی برای محاسبه این مقدار استفاده شده است. پس از اجرای تمامی آزمایشات در محیط شبیه‌سازی تکنیک شبکه‌های عصبی به کار گرفته می‌شود تا یک رابطه بین متغیرهای تصمیم و متغیر پاسخ تعیین شود. این رابطه بیان‌کننده چگونگی تاثیر تعداد و میانگین زمان خرابی اجزاء بر دسترس پذیری سیستم می‌باشد. بر اساس رابطه (25) تعداد آزمایشات باید برابر $180 \leq N \leq 18$ باشد. لازم بذکر است که ما از شاخص R^2 بعنوان معیار انتخاب کارایی شبکه عصبی استفاده نموده ایم و دلیل این موضوع این است که هدف یافتن مدلی است که بتواند توسط فاکتورهای اطلاعات را بخوبی توضیح دهد. مقادیر R^2 در شکل (11) نشان داده شده است. شبکه‌ها جهت همگرایی بهتر و رسیدن به حداقل خطا، آموزش در ۸ سیکل انجام شده است. بررسی قابلیت تعمیم شبکه عصبی آموزش‌دیده، آخرین مرحله در توسعه مدل شبکه عصبی مصنوعی است. برای یافتن شبکه‌ای با معماری مناسب به کمک الگوریتم‌های آموزشی، از معیار خطای مربعات میانگین استفاده می‌شود:

$$MSE = \frac{\sum_{p=1}^M \sum_{i=1}^N (S_{ip} - T_{ip})^2}{n_p \times n_o} \quad (25)$$

که در آن، MSE خطای مربعات میانگین، S_{ip} خروجی شبکه در نرون i ام و الگوی p ام، T_{ip} خروجی هدف در نرون i ام و الگوی p ام، n_p تعداد الگوها، n_o تعداد نرون‌های لایه خروجی، N تعداد نرونهای خروجی و M تعداد الگوهای آموزشی است. مقادیر بالای شاخص R^2 که در شکل (11) آمده است نشان می‌دهد که کارایی شبکه عصبی طراحی شده در حد رضایت بخش بوده و می‌توان از آن برای پیش‌بینی مقادیر متغیر پاسخ استفاده کرد. حال

تصمیم در جدول 1 آمده است که در آن متغیر x_i نشان‌دهنده نوع جزء به کار رفته در زیرسیستم i و متغیر n_i نشان‌دهنده تعداد اجزاء به کاررفته در زیرسیستم i می‌باشد.

جدول 1: فاکتورها (متغیرهای تصمیم) و سطوح آنها

		سطوح		
		سطح پایینی	سطح میانی	سطح بالایی
نوع جزء انتخابی زیرسیستم اول	x_1	1	3	5
نوع جزء انتخابی زیرسیستم دوم	x_2	1	3	5
نوع جزء انتخابی زیرسیستم سوم	x_3	1	3	5
نوع جزء انتخابی زیرسیستم چهارم	x_4	1	3	5
نوع جزء انتخابی زیرسیستم پنجم	x_5	1	3	5
تعداد اجزاء زیرسیستم اول	n_1	1	4	7
تعداد اجزاء زیرسیستم دوم	n_2	1	4	7
تعداد اجزاء زیرسیستم سوم	n_3	1	4	7
تعداد اجزاء زیرسیستم چهارم	n_4	1	4	7
تعداد اجزاء زیرسیستم پنجم	n_5	1	4	7

با توجه به اینکه در جدول (1) متغیر تصمیم نوع اول بین (5، 1) و متغیر تصمیم نوع دوم (7، 1) تغییر می‌کند آزمایشات لازم طراحی می‌شوند. بر اساس آزمایشات طراحی شده، سناریوهای گوناگونی در محیط شبیه‌سازی تولید و اجرا می‌شود. سپس مقادیر MTTFF هریک از این سناریوها به عنوان نتیجه آزمایش استخراج می‌شود. آزمایش‌های طراحی شده و نتایج آنها در در ضمیمه A ارائه شده است. به عنوان مثال در آزمایش اول برای زیرسیستم یک تعداد 3 عدد جزء نوع 1 به کار رفته است.

لازم به ذکر است که هر یک از آزمایشات در محیط شبیه‌سازی شده سیستم به اندازه 30 تکرار اجرا و میانگین مقادیر قابلیت اطمینان هریک از این سناریوها به عنوان نتیجه آزمایش استخراج می‌شود. در این بخش، عملکرد روش‌های پیشنهادی چند هدفه شده تنظیم شده مورد ارزیابی قرار گرفته و با استفاده از معیارهای چند هدفه مقایسه می‌شود. علاوه بر این، برای از بین بردن عدم اطمینان از جوابهای به دست آمده، هر یک از مسائل نمونه را در محیط‌های تصادفی مختلف سه بار اجرا شده اند. سپس، میانگین این سه اجرا به عنوان پاسخ نهایی محسوب می‌شود. سپس الگوریتم CBSA پیشنهادی با دو الگوریتم NSGA-II و NRGا را به عنوان بیشترین کاربرد MOEA های مبتنی بر پارتو در مسئله نمونه مقایسه می‌کنیم تا نشان داده شود عملکرد الگوریتم پیشنهاد شده برای حل مساله بهینه سازی چند هدفه عملکرد بهتری داشته است.

۱-5. معیارهای مقایسه ای

در این روشها غالباً مبنای اصلی طراحی هر معیار عملکرد بر مبنای یکی از سه حالت زیر است:

- تعداد حل‌های غیرغالب بدست آمده توسط الگوریتم
- تراکم و نزدیکی بین حل‌های بدست آمده و حل‌های بهینه (اگر حل‌های بهینه موجود باشد)
- منطقه پوشش داده شده توسط حل‌های بدست آمده به عنوان مثال توزیع و پراکندگی حل‌های غیر مغلوب.

برای ارزیابی عملکرد CBSA پیشنهادی، جدول (4) مقادیر معیارهای مقایسه ای چند هدفه را در 32 مساله آزمایشی گزارش می‌دهد که در آن "NAS" نشان می‌دهد که الگوریتم می‌تواند در زمان گزارش پارتو را پیدا نکند.

الگوریتم‌ها با استفاده از تجزیه و تحلیل واریانس (ANOVA) بر اساس معیارهای ارزیابی به دست آمده از نظر آماری مقایسه می‌شوند. در این مورد، ارزش هر معیار به یک معیار عملکرد نرمال شده تبدیل می‌شود به نام انحراف درصد نسبی (RPD)، که توسط فرمول زیر بدست می‌آید:

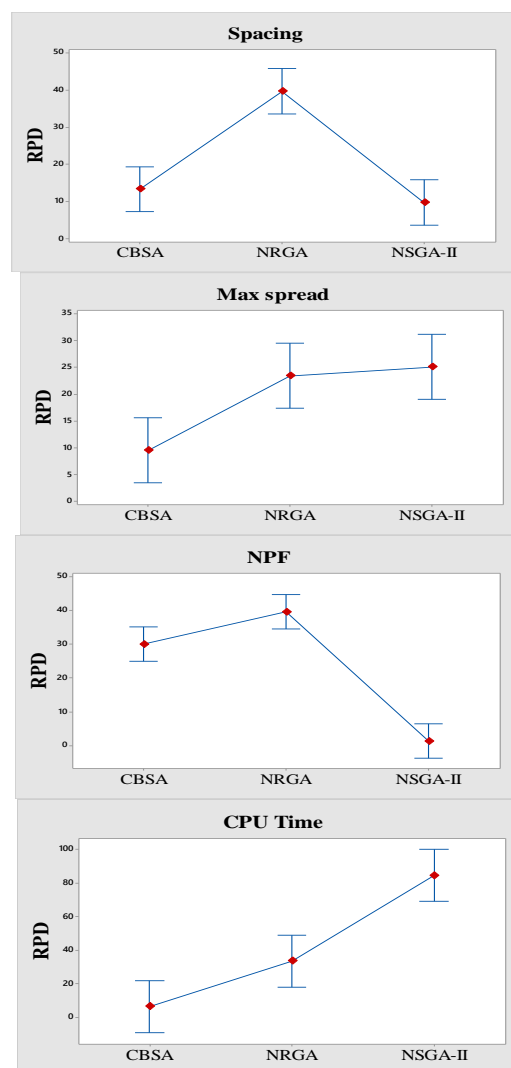
$$RPD(i, k) = \frac{Alg_{sol}(i, k) - min_{sol}(k)}{min_{sol}(k)} \times 100 \quad (27)$$

در معادله (27) $Alg_{sol}(i, k)$ مقدار اندازه‌گیری عملکرد را در تعداد k مساله توسط الگوریتم i را نشان می‌دهد و $min_{sol}(k)$ بهترین مقدار اندازه‌گیری عملکرد برای مساله شماره k بین تمام الگوریتم‌ها است. نتایج ANOVA در جدول (5) نشان داده شده است، در حالی که رد فرضیه Null نشان می‌دهد که بین الگوریتم‌ها تفاوت معنایی وجود دارد. نتایج ANOVA در جدول (5) نشان داده شده است، در حالی که رد فرضیه Null نشان می‌دهد که بین الگوریتم‌ها تفاوت معنی داری وجود دارد.

بر اساس خروجی‌های آماری در جدول (4) همراه با شکل (12)، NSGA-II عملکرد بهتر را از لحاظ NOS نشان می‌دهد؛ در حالی که NRGا دارای عملکرد بهتر از لحاظ زمان CPU است. علاوه بر این، شکل (12) مقایسه قابلیت CBSA در مقایسه با NSGA-II و NRGا را بر اساس معیار اندازه‌گیری‌های فاصله و تنوع نشان می‌دهد که در آن الگوریتم‌ها اختلاف معنی داری ندارند و از لحاظ آماری کار مشابهی دارند. لازم به ذکر است که این نتیجه‌گیری در سطح اطمینان 95٪ تایید شده است. بر

جوابهای CBSA بهترین نتیجه را برای هر تابع هدف ارائه می‌دهد، که نشان دهنده کارایی CBSA از نظر فاصله است. به عبارت دیگر، حداقل MTTFF و به ویژه حداقل هزینه توسط CBSA به دست آمده است.

اساس خروجی‌های جدول (۵)، با افزایش اندازه مساله، در مسائل ۲۲ و ۳۰ و در مسئله 32، NSGA-II و NPGA نمی‌توانند جوابهای پارتو مناسبی ارائه دهند. با این حال، در اندازه‌های بزرگ، CBSA می‌تواند جوابهای پارتو مناسب را بیابد. الگوریتم CBSA عملکرد بهتری براساس معیار CPU TIME را اجرا می‌کند. این ویژگی‌ها نتیجه می‌شود CBSA پیشنهاد شده در زمینه مسائل بهینه‌سازی چند هدفه در مسائل بزرگ عملکرد بهتری نسبت با سایر الگوریتمهای پیشنهادی دارد.



شکل (۱۲): مقایسه CBSA در مقایسه با NPGA و NSGA-II براساس معیارهای تنوع، زمان CPU، فاصله گذاری، و معیارهای NOS

نتایج نشان داده شده در شکل (۱۳) به وضوح نشان می‌دهد که تفاوت بین CBSA و دو الگوریتم دیگر از لحاظ فاصله و تنوع وجود دارد. به منظور تجزیه و تحلیل گرافیکی عملکرد CBSA، جوابهای پارتو تمام الگوریتم‌های حاصل از چهار مساله نمونه در ابعاد مختلف در شکل (۱۴) ارائه شده است. توجه کنید که

جدول (۲): داده‌های ورودی مدل تحقیق

subsystem	Number of states	Component type	availability	Performance rates	Cost (\$)	Weight (kg)
U [3, 5]	U [3, 4]	U [1, 5]	U [0.05, 0.95]	U [0, 2.5]	U [1.5]	U [0.5, 2]

جدول (۳): اطلاعات دقیق ورودی مدل تحقیق

	Failure rate of components				Repair rate of Components			
	λ_1	λ_2	λ_3	λ_4	μ_1	μ_2	μ_3	μ_4
subsystem 1	0.0112	0.0123	0.0119	0.0123	0.0065	0.0060	0.0054	0.0053
subsystem 2	0.0123	0.0123	0.0106	0.0119	0.0054	0.0058	0.0063	0.0058
subsystem 3	0.0108	0.0125	0.0114	0.0122	0.0066	0.0055	0.0053	0.0061
subsystem 4	0.0110	0.0115	0.0119	0.0118	0.0060	0.0053	0.0062	0.0065
subsystem 5	0.0108	0.0104	0.0109	0.0104	0.0056	0.0063	0.0053	0.0060

جدول (۴) گزارش مقادیر معیارهای ارزیابی چند هدفه را در ۳۲ مساله آزمایشی

	Spacing			Diversity			NOS			CPU Time		
	CBSA	NRGA	NSGA-II	CBSA	NRGA	NSGA-II	CBSA	NRGA	NSGA-II	CBSA	NRGA	NSGA-II
P1	4114	3391	2573	33426	29518	34155	14	12	21	219	199	229
P2	3233	5084	3167	29405	30792	20061	13	13	15	213	195	207
P3	4449	4926	3199	30338	28844	19474	10	7	19	191	230	206
P4	4458	5293	3493	29540	16388	32405	10	8	15	247	240	225
P5	2944	3393	2619	11145	28784	25587	14	13	20	214	210	248
P6	3904	3018	3582	20331	22170	17184	10	14	18	237	194	257
P7	3686	3485	3778	11626	22794	33155	15	7	17	237	257	219
P8	2579	4711	3313	33071	29373	21691	12	13	12	229	251	222
P9	3235	4757	2578	34007	28531	17814	15	11	18	326	621	605
P10	2940	3766	3363	41583	21435	26662	8	10	20	348	622	512
P11	2643	3681	2066	38960	15764	21584	15	11	18	437	352	687
P12	3395	4104	3430	21892	23473	34681	13	7	12	486	662	436
P13	3490	4724	4418	45878	43307	38702	10	8	14	490	322	490
P14	3429	3891	2564	39790	35333	20648	12	7	17	357	408	646
P15	3395	4026	2924	30989	25257	27697	9	7	17	446	633	416
P16	2591	4719	3817	32907	25076	25679	10	10	17	483	332	643
P17	7549	7819	5422	23564	29870	27461	15	7	20	643	1034	1938
P18	5219	6883	6591	27262	32996	40017	15	12	17	615	1077	2090
P19	7506	6656	6284	31883	26333	16343	8	13	14	687	1024	1991
P20	5503	7654	5730	38947	28868	26863	14	9	17	690	1028	1781
P21	5275	7187	6753	19782	24038	36302	11	7	12	615	1023	1769
P22	6138	8226	5957	49494	32604	28381	15	12	21	617	1100	1869
P23	5467	6350	6648	34937	45981	40848	11	8	19	631	1016	1779
P24	5034	7944	6967	49909	27029	46649	10	12	15	636	1030	1975
P25	8551	12224	7704	51890	45976	28189	10	9	14	3561	4695	6494
P26	8034	12490	10645	56949	30739	45566	15	12	12	3862	4526	6463
P27	9272	10704	10745	60567	42278	37127	9	14	20	3648	4654	6462
P28	9169	11849	9988	64031	43910	39791	14	12	13	3559	4552	6456
P29	10605	11540	8849	46022	48133	32100	7	9	21	3622	4590	6292
P30	10003	10496	8154	60669	43883	45921	8	8	21	3525	4848	6089
P31	8431	11051	7965	64308	44316	38391	11	14	17	3721	4647	6370
P32	10816	11467	9024	68622	35700	42636	10	10	16	3594	4773	6290

جدول (۵): نتایج ANOVA مدل

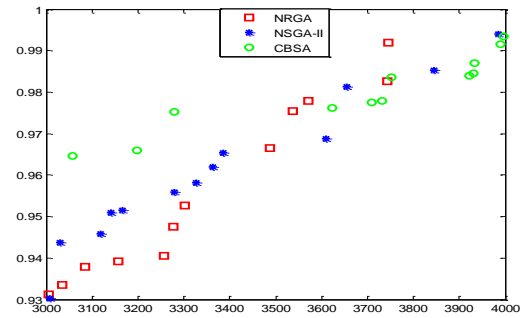
Metric's name	F-value	P-value	Test results
Spacing	28.60	5.27e-10	Null hypothesis is rejected
Diversity	7.75	1e-03	Null hypothesis is rejected
NOS	59.97	9.25e-15	Null hypothesis is rejected
CPU Time	25.47	5.30e-13	Null hypothesis is rejected

نمائی برای نشان دادن نرخ خرابی و تعمیر عناصر سیستم از نوآوری‌های این تحقیق محسوب شده که امکان استفاده از روش‌های ریاضی و تحلیلی را برای حل مدل ارائه شده محدود کرده و مستلزم بهره‌گیری از تکنیک شبیه‌سازی و شبکه‌های عصبی مصنوعی می‌باشد. به همین منظور رویکرد بهینه‌سازی مبتنی بر شبیه‌سازی و شبکه‌های عصبی مصنوعی در این تحقیق مورد استفاده قرار گرفته است. الگوریتم‌های NSGA-II، NPGA و CBSA برای ارزیابی عملکرد پیشنهاد شده است. نتایج نشان می‌دهد که الگوریتم CBSA برای حل مساله چند هدفه توانایی بالایی دارد.

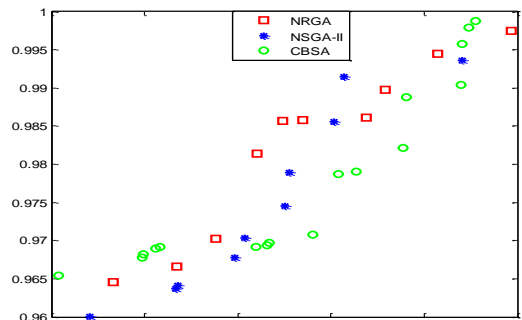
از پیشنهاد‌های قابل‌ارایه جهت انجام تحقیقات آتی اضافه کردن ویژگی‌های جدید به سیستم و بازنگری مدل بهینه‌سازی تحقیق و حل مدل جدید، خرابی اجزا به صورت فازی باشند، لحاظ کردن محدودیت‌های مربوط به تعداد تعمیرکاران موجود، و در آخر لحاظ کردن بحث تنوع در خدمات موجود در تعمیر زیرسیستم‌ها و هزینه‌های مترتب بر این تنوع در تابع هدف هزینه سیستم است.

References

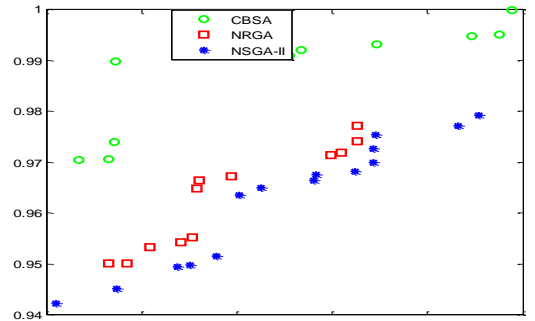
- [1] Chern, M. S., (1992), On The computational complexity of reliability redundancy allocation in a series system. *Operations Research Letters*, 11(5), 309-315.
- [2] Kuo, W., Wan, R. (2007), "Recent Advances in Optimal reliability Allocation". *IEEE Transactions on systems, man and cybernetics-Part A: systems and humans*. 37(2): 143 - 156.
- [3] Gnedenko, B., Ushakov, I.(1995). *Probabilistic Reliability Engineering*. New York: John Willy & Sons, Inc.
- [4] Coit. D. W. Smith. A. E.(2002). "Genetic algorithm to maximize a lowerbound for system time-to-failure with uncertain component Weibull parameters". *Computers and Industrial Engineering*. 41(4):423-440.
- [5] Hsieh. Y. C. (2003). "A linear approximation for redundant reliability problems with multiple component choices". *Computers and Industrial Engineering*. 44(1): 91-103.
- [6] Ha. C. Kuo. W.(2006). "Reliability redundancy allocation: An improved realization for nonconvex nonlinear programming problems". *European Journal of operation research*. 171(1): 24-38.
- [7] Nahas, N., Nourelfath M. (2005). "Ant system for reliability optimization of a series system with multiple-choice and budget constraints". *Reliability Engineering & System Safety*. 87(1) 1-12.
- [8] You, P. S., Chen, T. C. (2005). "An efficient heuristic for series-parallel redundant reliability



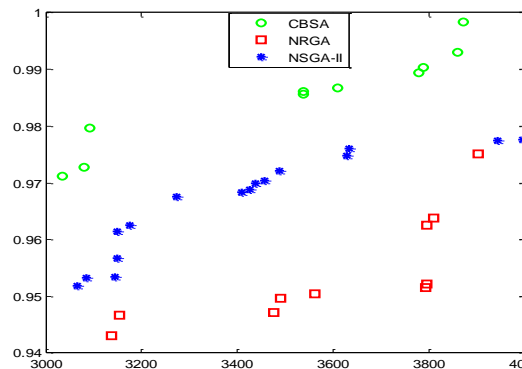
Problem 8



Problem 16



Problem 24



Problem 32

نمودار (13). مرز پارتو بدست آمده برای چهار مساله نمونه ۸، ۱۶، ۲۴ و ۳۲.

۶. نتیجه‌گیری و پیشنهادات

در این مقاله، مسئله تخصیص افزونگی با توجه به دو هدف حداکثر سازی MTTFF و به حداقل رساندن هزینه کل مدل سازی شده است. علاوه بر این بهره‌گیری از توزیع‌های غیر

- problems". *Computers & Operations Research*. 32(8):2117–2127.
- [9] Tavakkoli-Moghaddam, R.; Safari, J., (2007), A New mathematical model for a redundancy allocation problem with mixing components redundant and choice of redundancy strategies. *Applied Mathematical Sciences*, 45(1), 2221-2230.
- [10] Tavakkoli-Moghaddam, R.; Safari, J.; Sassani, F., (2008), Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. *Reliability Engineering and System Safety*, 93(4), 550-556.
- [11] Amari, S. V. Pham. H. Dill. G.(2012). "Optimal design of k-out-of-n : G subsystems subjected to imperfect fault-coverage". *IEEE Transactio on Reliability*. 53(4):567–575.
- [12] Chambari, A., Rahmati, S.H.A., Najafi, A.A., karimi, A.(2012) "A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies". *Computers & Industrial Engineering* 63:109–119.
- [13] Charles E. Wells (2014), Reliability analysis of a single warm-standby system subject to repairable and nonrepairable failures, *European Journal of Operational Research*. 235:180-186.
- [14] Amiri M., Sadeghi M.R., Khatami Firoozabadi A., Mikaili F., (2014), A Multi-Objective Optimization for Redundancy Allocation Problems in Series-Parallel Systems with Repairable Components, *International Journal of Industrial Engineering & Production Research*, 71-81.
- [15] Deyi L, Haijun M, Xuemei S (1995). Membership clouds and membership cloud generators. *J Comput Res Dev*, 32:15–20.
- [16] Deyi L., Yi D., (2005), *Artificial intelligence with uncertainty*, Chapman and Hall.
- [17] Lv, P., Yuan, L., & Zhang, J. Cloud theory-based simulated annealing algorithm and application. *Engineering Applications of Artificial Intelligence* 2009; 22:742–749.
- [18] Deb, K., (2001). *Multiobjective Optimization Using Evolutionary Algorithms*. New York: Wiley

ضمیمه A: آزمایشات طراحی شده و نتایج اجرای آزمایشات

فاکتورها (متغیرهای تصمیم)

متغیر پاسخ

آزمایش	x_1	x_2	x_3	x_4	x_5	n_1	n_2	n_3	n_4	n_5	MTTFF
1	1	1	1	1	1	۳	7	7	1	7	0.516
2	5	1	5	5	1	1	7	1	7	7	0.888
3	1	1	1	1	5	1	7	1	7	7	0.624
4	1	1	5	1	1	1	7	7	1	7	0.616
5	1	1	1	5	1	1	7	7	7	1	0.636
6	1	1	5	5	5	7	7	1	7	1	0.94
7	1	5	1	1	1	7	1	7	7	1	0.476
8	1	5	5	1	1	7	1	7	7	7	0.708
9	1	1	1	1	1	7	1	7	7	7	0.492
10	5	5	1	5	1	7	7	7	7	7	0.904
11	5	1	5	1	5	7	7	7	1	1	0.828
12	5	5	1	1	1	1	1	1	1	1	0.352
13	1	5	5	1	1	1	1	7	7	1	0.576
14	1	1	5	5	5	7	7	7	7	7	0.978
15	5	1	5	5	5	7	1	7	1	1	0.888
16	1	5	1	1	5	7	1	7	1	1	0.536
17	5	5	5	5	1	7	7	1	1	1	0.848
18	5	5	1	1	1	1	7	7	1	1	0.52
19	1	1	1	5	1	1	1	7	1	1	0.444
20	1	1	5	1	1	7	1	1	1	1	0.436
21	5	5	1	5	5	7	1	1	7	1	0.808
22	1	5	5	5	1	7	1	1	1	1	0.66
23	1	1	5	1	1	1	1	7	1	1	0.436
24	3	3	3	1	3	4	4	4	4	4	0.628
25	1	1	1	1	1	7	7	7	7	1	0.528
26	3	3	3	3	5	4	4	4	4	4	0.784
27	5	1	1	5	5	7	1	7	7	1	0.812
28	1	1	1	5	5	7	7	7	1	7	0.828
29	1	5	5	5	1	1	1	1	7	1	0.684
30	1	5	5	1	5	7	1	1	7	7	0.792
31	1	1	1	5	5	1	7	1	1	7	0.708
32	5	1	1	1	5	7	1	1	1	7	0.572
33	1	5	1	5	5	7	1	1	7	7	0.8
34	1	5	1	5	5	1	1	1	7	1	0.668
35	5	5	1	1	1	7	1	7	1	1	0.472
36	5	5	1	1	5	1	7	1	7	7	0.76
37	3	3	5	3	3	4	4	4	4	4	0.792
38	1	5	5	5	1	1	1	7	7	7	0.816
39	5	1	1	1	1	7	1	1	1	1	0.356
40	1	1	1	1	1	7	7	1	1	1	0.384
41	1	5	1	5	1	1	1	1	1	1	0.44
42	1	5	1	1	5	7	1	1	7	1	0.56
43	5	5	5	5	5	1	7	7	7	1	0.997
44	5	5	5	5	5	7	7	7	1	1	0.968
45	1	5	5	5	1	1	7	7	7	1	0.852
46	5	5	5	1	5	7	1	1	1	7	0.788
47	3	3	3	3	3	4	1	4	4	4	0.658
48	1	5	1	1	1	7	7	7	7	7	0.656
49	5	1	1	1	1	1	7	1	7	7	0.56
50	5	5	1	5	1	1	7	1	7	7	0.784
51	5	5	1	5	5	7	1	1	1	7	0.796
52	5	1	1	5	5	1	1	7	7	7	0.824
53	5	1	5	1	5	1	1	7	7	7	0.816
54	5	1	5	5	5	7	7	7	1	7	0.968
55	5	1	1	1	5	7	7	1	1	1	0.608
56	1	1	5	1	5	1	1	1	1	1	0.52
57	1	1	1	1	1	7	1	7	1	1	0.336

58	1	1	1	5	5	1	7	1	7	1	0.72
59	1	5	1	5	1	7	1	1	1	7	0.572
60	5	5	5	1	5	7	1	7	1	1	0.776
61	1	1	1	1	1	1	1	7	7	1	0.36
62	5	5	1	5	1	7	1	7	1	7	0.712
63	5	1	5	1	5	7	7	1	7	1	0.852
64	1	1	5	5	1	7	7	1	7	7	0.868
65	1	1	1	5	5	7	1	7	7	7	0.804
66	1	1	5	1	1	1	7	1	1	1	0.484
67	5	1	5	1	1	1	7	7	7	7	0.78
68	1	5	5	1	5	1	1	1	7	1	0.66
69	1	1	5	5	5	7	1	7	1	7	0.88
70	5	5	1	1	5	7	1	7	7	1	0.7
71	5	1	1	5	5	7	7	1	7	1	0.86
72	1	5	1	1	5	1	1	7	1	7	0.548
73	5	5	5	5	1	1	7	1	1	7	0.86
74	5	5	1	1	5	7	7	7	1	1	0.724
75	1	1	1	1	1	1	7	7	7	7	0.54
76	1	5	1	5	1	1	7	7	7	7	0.764
77	5	1	1	1	5	7	7	1	7	7	0.764
78	3	3	3	3	3	4	4	4	4	4	0.712
79	5	1	1	5	1	1	1	1	7	7	0.62
80	5	1	5	5	1	7	1	1	1	1	0.684
81	1	1	5	1	5	1	1	7	7	1	0.664
82	5	5	5	5	1	1	7	7	1	1	0.848
83	1	5	1	1	5	7	7	7	1	7	0.716
84	1	5	5	1	1	7	1	1	1	7	0.564
85	5	5	5	1	5	7	1	7	7	7	0.932
86	1	1	5	1	1	7	7	7	7	7	0.76
87	5	5	1	1	5	7	1	1	1	1	0.556
88	1	5	1	1	5	1	7	7	7	7	0.74
89	5	5	5	5	5	1	7	1	7	7	0.988
90	5	5	1	5	1	1	1	1	1	7	0.592
91	5	5	1	5	1	7	7	7	1	1	0.748
92	1	1	1	5	5	7	1	1	7	1	0.672
93	5	1	1	1	5	1	1	7	1	7	0.572
94	5	5	1	5	5	1	7	1	7	1	0.856
95	5	1	5	5	1	1	7	1	1	1	0.732
96	5	5	5	5	5	7	1	1	1	1	0.884
97	5	5	5	5	1	7	1	7	1	1	0.8
98	1	1	1	1	5	1	1	7	7	7	0.576
99	5	5	5	5	5	1	7	1	1	1	0.932
100	1	5	5	5	5	1	1	1	1	1	0.744