

# بهینه‌سازی دوفده‌ی مسئله‌ی تخصیص افزونگی سیستم‌های سری - موازی با اجزای تعمیرپذیر با الگوریتم ژنتیک چندهدفه مبتنی بر شبیه‌سازی تبرید

Original Article

رضا رمضانیان\* (استادیار)

دانشکده‌ی مهندسی صنایع، دانشگاه صنعتی خواجه نصیرالدین طوسی

مهران حسینی (دانشجوی کارشناسی ارشد)

دانشکده‌ی مهندسی صنایع، دانشگاه پیام نور، واحد فنی و مهندسی تهران - شمال

در این مقاله، یک روش جدید برای حل مسئله تخصیص افزونگی در سیستم‌های سری - موازی با اجزای تعمیرپذیر ارائه شده است. در سیستم‌های تعمیرپذیر مفهوم دسترس‌پذیری از اهمیت بسیار بالایی برخوردار است. هدف اصلی این مقاله، بهینه‌سازی دو هدف دسترس‌پذیری و هزینه‌ی سیستم است. با توجه به این که دسترس‌پذیری به دلیل غیرنمایی بودن توزیع نرخ خرابی و تعمیر به صورت ریاضی به سختی قابل برآورد است، برای هر کدام از زیرسیستم‌های دارای اجزای موازی مقدار دسترس‌پذیری با استفاده از شبیه‌سازی جداگانه محاسبه شده و به صورت آرشیو در الگوریتم حل استفاده می‌شود. در این تحقیق، الگوریتم ژنتیک چندهدفه مبتنی بر شبیه‌سازی تبرید برای حل مسئله استفاده می‌شود. الگوریتم ارائه شده بر روی مثال‌های طراحی شده آزمایش شد و بعد از به دست آوردن نتایج از لحاظ آماری با الگوریتم‌های MOPSO و NSGA-II مقایسه شدند که برتری SAMOGA نشان داده شده است.

واژگان کلیدی: مسئله‌ی تخصیص مازاد، سیستم‌های سری - موازی، اجزای تعمیرپذیر، ژنتیک چندهدفه‌ی مبتنی بر شبیه‌سازی تبرید.

Ramezani@kntu.ac.ir  
mehranhoseini10@yahoo.com

## ۱. مقدمه

ابتدا در رابطه با صنایع هوایی و کاربردهای نظامی شکل گرفت ولی به سرعت توسط سایر صنایع مانند صنایع هسته‌یی - که برای تضمین ایمنی و قابلیت اطمینان راکتورهای هسته‌یی در تأمین انرژی الکتریکی شدیداً تحت فشارند - یا صنایع فرایندهای تولید پیوسته مانند صنایع فولاد و صنایع شیمیایی - که هر ساعت از توقف آنها به علت وجود خرابی خسارت‌های بزرگ مالی و جانی و آلودگی محیط زیست را تحمیل کند - مورد توجه قرار گرفت. افزایش قابلیت اطمینان سیستم یکی از حوزه‌های جذاب برای مهندسان و طراحان سیستم‌ها بوده و استفاده از اجزای مازاد یکی از رویکردهای معمول در فرایند توسعه سیستم‌هاست. هر سیستم معمولاً از کنار هم قرار گرفتن اجزاء مختلفی تشکیل شده که نحوه‌ی ارتباط این اجزا با هم، به عملکرد آنها در سیستم بستگی دارد. از انواع ساختار سیستم‌ها می‌توان به سیستم‌های با اجزای موازی، سری، موازی - موازی، موازی - سری و ساختارهای با شبکه‌ی پل اشاره کرد. ساختار سری - موازی یکی از انواع ساختارهای سیستم است که با تخصیص اجزای مازاد به صورت موازی به اجزای یک سیستم با ساختار سری، طراحی شده است.<sup>[۱]</sup> قابلیت اطمینان، احتمال عملکرد صحیح سیستم در یک بازه زمانی مشخص است

پیدایش مهندسی قابلیت اطمینان را می‌توان به سال‌های قبل از سال ۱۹۳۰ ارجاع داد. در آن دوره به دلیل نگرانی از کارکرد صحیح تولیدات و وقوع حوادثی چون غرق شدن کشتی تایتانیک، مطالعه و تحقیقاتی در زمینه‌ی طراحی سیستم‌ها با اجزای موازی یا ذخیره انجام گرفت. سپس در دهه‌ی ۱۹۳۰ همزمان با اختراع توسعه حمل و نقل هوایی، مطالعه و تحقیقاتی در خصوص کمی کردن قابلیت اطمینان انجام شد. در دهه‌ی ۱۹۴۰ با آغاز جنگ جهانی دوم و ساخت ابزارهای پیچیده نظامی، بحث مدل‌سازی قابلیت اطمینان انجام گرفت. در دهه‌ی ۱۹۵۰ با به کارگیری سیستم‌های الکترونیکی گروه‌های مطالعاتی IEEE و Agree تشکیل شده و به تهیه‌ی استانداردهایی برای تولید قطعات با قابلیت اطمینان بالا پرداختند. در دهه‌ی ۱۹۶۰ با پیشرفت صنایع هوایی و انگیزه‌ی ساختن آپولو، اولین کتاب در این زمینه به رشته‌ی تحریر درآمد. شیوه‌های ارزیابی قابلیت اطمینان از نظر تاریخچه‌ی پیدایش

\* نویسنده مسئول

تاریخ: دریافت ۱۳۹۵/۵/۲، اصلاحیه ۱۳۹۵/۱۰/۱۵، پذیرش ۱۳۹۶/۴/۲۵.

DOI:10.24200/J65.2018.20087

بر استراتژی مزاد بهینه ارائه شده است.<sup>[۱۹]</sup> در شکلی دیگر، روشی برای استفاده از زیرسیستم‌های غیرهمگن در سیستم‌های سری - موازی به منظور بهینه‌سازی قابلیت اطمینان ارائه شده است.<sup>[۲۰]</sup>

در ادامه این نوشتار، در بخش دوم به تعریف مسئله و بیان مفروضات آن خواهیم پرداخت و بر پایه‌ی این مفروضات، یک مدل ریاضی برای مسئله تخصیص افزونگی سیستم‌های سری - موازی با اجزای تعمیرپذیر ارائه می‌شود. روش فرایستکاری در بخش سوم تشریح می‌شود. در بخش چهارم نتایج حاصل از حل مدل با استفاده از روش فرایستکاری ارائه شده، بررسی می‌شود. در بخش پایانی نتیجه‌گیری و پیشنهاد برای مطالعات آینده ارائه شده است.

به طور خلاصه اصلی‌ترین نوآوری‌های این تحقیق عبارت‌اند از:

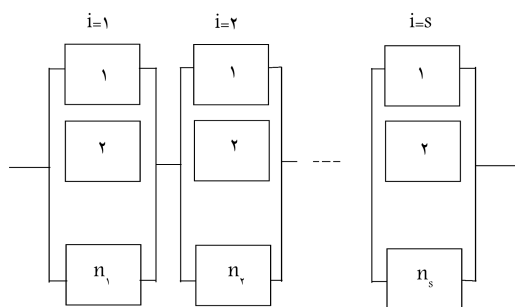
۱. بهینه‌سازی دودهدف‌های مسئله تخصیص افزونگی سیستم‌های سری - موازی با اجزای تعمیرپذیر؛
۲. لحاظ کردن گزینه‌ی تعمیرپذیری در سیستم سری - موازی و استفاده از مفهوم دسترس‌پذیری؛
۳. استفاده از شبیه‌سازی جهت برآورد دسترس‌پذیری به دلیل در نظر گرفتن توزیع‌های غیرنمایی برای نرخ خرابی و تعمیر؛
۴. توسعه‌ی یک الگوریتم ژنتیک چندهدفه مبتنی بر شبیه‌سازی تیرید برای حل کارآمد مسئله‌ی مورد مطالعه.

## ۲. تعریف مسئله

در این نوشتار سیستمی با ساختار سری - موازی  $k$  از  $n$  متشکل از  $s$  زیرسیستم مستقل از هم در نظر گرفته شده است (شکل ۱). از آن‌جا که این نوع ساختار به واقعیت نزدیک‌تر است، مهمترین و پرکاربردترین ساختار در بین تحقیقات مختلف بوده است. در این ساختار یک سیستم با تعداد  $s$  زیرسیستم به صورت سری در کنار یکدیگر قرار گرفته‌اند و هرکدام از این زیرسیستم‌ها از  $n_i$  ( $i = 1, 2, \dots, s$ ) جزء مختلف که به صورت موازی قرار گرفته‌اند تشکیل شده است. برای هر کدام از زیرسیستم‌ها اجزای مختلفی وجود دارد، که در مدل ارائه شده برای هر زیرسیستم فقط از یک نوع جزء می‌توان استفاده کرد.

### ۱.۲. مفروضات

- اجزاء داخل هر زیرسیستم به صورت موازی قرار دارند.
- طول عمر قطعات (اجزا) دارای توزیع غیرنمایی است.

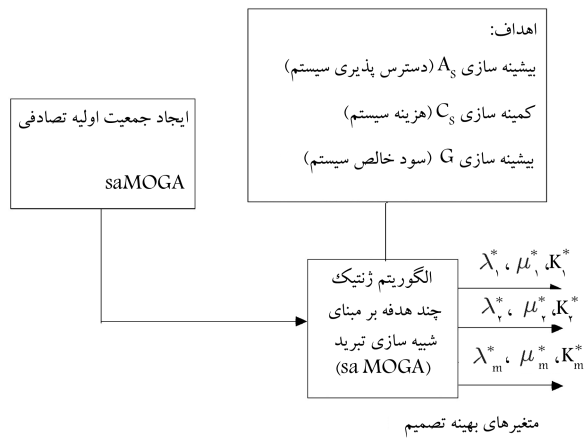


شکل ۱. ساختار سیستم متشکل از  $s$  زیرسیستم مستقل از هم.

و قابلیت اطمینان کل سیستم نیز ترکیبی از قابلیت اطمینان تک‌تک اجزای آن است. دو راهکار عمده به منظور افزایش قابلیت اطمینان سیستم وجود دارد. راهکار اول، بالا بردن قابلیت اطمینان اجزای سیستم و دیگری استفاده از اجزای مزاد در کنار اجزای اصلی سیستم به صورت موازی است. به علت محدودیت‌های اقتصادی و تکنولوژیکی، بهترین و کارآمدترین روش جهت افزایش قابلیت اطمینان سیستم، راهکار دوم یعنی استفاده از اجزای مزاد در کنار اجزای اصلی است. به همین دلیل در این نوشتار نیز این رویکرد به کار گرفته شده است.<sup>[۲۱]</sup> پس از اولین مطالعه<sup>[۲]</sup> که در آن مسئله تخصیص افزونگی در سیستم‌های سری - موازی بررسی شد، محققین بسیاری در مطالعاتشان سعی بر توسعه‌ی این دانش داشته‌اند.<sup>[۳-۱۰]</sup>

سلطانی و همکاران<sup>[۱۱]</sup> نیز مدلی ارائه کردند که در آن قابلیت اطمینان سیستم‌های سری - موازی با اجزای ترکیبی مورد بررسی قرار گرفت و بهبود یافت. لیانگ و همکاران<sup>[۱۲]</sup> از روش جست‌وجوی همسایگی برای تخصیص مزاد بهره بردند. همچنین از مدیریت دانش در بهینه‌سازی و طراحی دسترسی مناسب در سیستم‌های چابک یاد شده است.<sup>[۱۳]</sup> رویکردهای دیگری هم در زمینه مهندسی و ایمنی بر پایه‌ی بهبود قابلیت اطمینان و دسترس‌پذیری و نیز استفاده از شبیه‌سازی تیرید در مسئله تخصیص مزاد مطرح شده است.<sup>[۱۴، ۱۵]</sup>

با نگاهی به جریان مطالعات انجام گرفته در مسئله تخصیص افزونگی، به طور آشکار می‌توان دید که به این مسئله در سیستم‌های تعمیرپذیر کمتر پرداخته شده است. کمبود مطالعات برای مسئله تخصیص افزونگی در سیستم‌های سری - موازی با اجزا و زیرسیستم‌های تعمیرپذیر، در پژوهش محققین و پس از مطالعات کاملی در ادبیات مسئله تخصیص افزونگی آورده شده است.<sup>[۱۶]</sup> از آن‌جا که ساختار سیستم می‌تواند تأثیر به‌سزایی بر قابلیت اطمینان سیستم داشته باشد، تعمیرپذیر بودن یا نبودن اجزای سیستم یا زیرسیستم‌های آن نیز از دیگر موضوعات تأثیرگذار بر عملکرد صحیح یک سیستم در مدت زمان کارکرد آن است.<sup>[۱۷]</sup> تعمیرپذیری بودن یک سیستم به این معناست که بعد از خرابی می‌توان با صرف زمان و انجام تعمیرات لازم آن را مجدداً راه‌اندازی کرد. زمانی که یک سیستم تعمیرپذیر باشد از واژه‌ی «دسترس‌پذیری» به جای قابلیت اطمینان استفاده می‌شود. دسترس‌پذیری به معنای درصدی از زمان است که سیستم تعمیرپذیر به صورتی صحیح وظایف تعریف شده را انجام می‌دهد.<sup>[۱۸]</sup> همان‌طور که برای سیستم‌های تعمیرپذیر واژه‌ی متفاوت از قابلیت اطمینان به عنوان پارامتر عملکردی سیستم تعریف می‌شود، سایر مفاهیم مرتبط با عملکرد سیستم نیز در سیستم‌های تعمیرپذیر متفاوت خواهد بود. میانگین زمان تا خرابی<sup>۱</sup> یا ماندگاری<sup>۲</sup> در سیستم‌های تعمیرناپذیر، نشان‌دهنده‌ی میانگین ماندگاری طول عمر سیستم است در حالی که در سیستم‌های تعمیرپذیر اجزاء قابل تعمیر بوده و پس از تعمیر، سیستم دوباره می‌تواند به کار خود ادامه دهد؛ بنابراین میانگین زمان تا اولین خرابی سیستم (MTTFF) میانگین زمانی است که سیستم برای اولین بار از کار می‌افتد. هدف اصلی این نوشتار نیز تخصیص اجزای مزاد در سیستم‌هایی با اجزای تعمیرپذیر با هدف افزایش مقدار میانگین زمان تا اولین خرابی سیستم (MTTFF) تحت محدودیت‌هایی همچون هزینه‌ی کل و وزن سیستم و مجموع اجزای داخل زیرسیستم‌هاست. مسئله تخصیص اجزای مزاد، یک مسئله بهینه‌سازی چند جمله‌ی غیرخطی<sup>۳</sup> است<sup>[۱۸]</sup> که به روش‌های گوناگون بررسی و حل شده است؛ از آن‌جا که با گسترش فضای جواب، روش‌های دقیق در حل این مسائل ناکارا هستند، در سال‌های اخیر روش‌های فراابتکاری نسبت به روش‌های دقیق برتری یافته‌اند. از جمله روش‌های بهینه‌سازی کلونی مورچه‌ها،<sup>[۷]</sup> بهینه‌سازی گروه ذرات مبتنی بر تیرید<sup>[۸]</sup> و الگوریتم جست‌وجوی همسایگی متغیر<sup>[۹]</sup> را می‌توان در ادبیات دید. برای زیرسیستم‌های با اجزای آماده به کار و فعال مثال‌هایی مبتنی



شکل ۲. ساختار پیشنهادی برای سیستم سری - موازی.

(شکل ۲) عبارت‌اند از: نرخ خرابی، نرخ تعمیر و تعداد اجزای بهینه در زیرسیستم؛ یعنی  $s = 1, 2, \dots$ ،  $\lambda_i^*, \mu_i^*, k_i^*$  توابع هدف  $A_s$  و  $C_s$  مورد استفاده قرار گرفته‌اند. ما روش SAMOGA را ارائه دادیم که برای پرداختن به مسائل شرح داده شده در بخش ۲، سه ویژگی دارد: ۱. دارای متد بهینه‌سازی پارتو برای مسائل بهینه‌سازی چند هدفه است؛ ۲. ویژگی شبیه‌سازی تبرید را با استراتژی جست‌وجویی ترکیب می‌کند؛ ۳. از طراحی همسایگی برای سازوکار تکاملی بهره گرفته است. جزئیات طراحی SAMOGA در ادامه ارائه خواهد شد.

### ۱.۳. طراحی SAMOGA

الگوریتم ژنتیک که در ابتدا توسط هولاند در سال ۱۹۷۵ ارائه شد، نمونه‌ی قدرتمند، با هدف بهینه‌سازی عمومی ارائه می‌دهد که در آن فرایند محاسباتی از نظریه‌ی تکامل زیستی پیروی می‌کند؛ کاربرد این الگوریتم در بسیاری از حوزه‌ها موفقیت‌آمیز بوده است. برای پرداختن هم‌زمان به چندین تابع هدف، تعدادی روش بهینه‌سازی چندهدفه ایجاد شده است. می‌توان رویکرد MOGA را به دو دسته تقسیم کرد: ۱. روش غیر پارتو مانند متد VEGA؛ ۲. رویکردهای مبتنی بر پارتو مانند روش NPGA.

در این نوشتار با استفاده از مفاهیم بهینه‌سازی پارتو، گونه‌ی جدیدی از الگوریتم ژنتیک چندهدفه با نام «ژنتیک چندهدفه مبتنی بر شبیه‌سازی تبرید» ارائه شده است. بهینه‌سازی پارتو که توسط ویلفرد پارتو در ۱۹۹۵ ارائه شد یک اصطلاح اقتصادی برای توصیف ویژگی جواب‌های توابع چندهدفه است. جواب‌های پارتو غیر غالب‌اند و این بدان معناست که بدون بدتر کردن یک قسمت دیگر از جواب نمی‌توان هیچکدام از قسمت‌های جواب پارتو را بهبود بخشید. روش SAMOGA ارائه شده برای حل مسائل تخصیص دسترس‌پذیری و بهینه‌سازی مورد استفاده قرار گرفته است. ساختار SAMOGA پیشنهادی، مفاهیم شبیه‌سازی تبرید و طراحی همسایگی معرفی شده در بخش ۴.۴ را ترکیب می‌کند. این الگوریتم با ۵ ویژگی ایجاد شده است: ۱. رویکرد مبتنی بر بهینه‌سازی پارتو؛ ۲. استراتژی جست‌وجو بر مبنای شبیه‌سازی تبرید؛ ۳. توابع چندهدفه؛ ۴. طراحی همسایگی برای سازوکار تکاملی؛ ۵. عملگرهای ژنتیک چندگانه.

### ۲.۳. سازوکار نمایش و ایجاد جمعیت اولیه

اولین گام اساسی در به کارگیری الگوریتم ژنتیک و پیاده سازی آن، نمایش جواب‌های مسئله به صورت کروموزوم است. در حقیقت این عمل، یک مفهوم کلیدی در

- اجزاء تعمیر پذیر بوده و نرخ تعمیر قطعات نیز غیرنمایی است.
- در هر زیرسیستم از انواع مختلف از اجزاء می‌توان استفاده کرد.
- در داخل یک زیرسیستم فقط از یک نوع جزء می‌توان استفاده کرد.
- در داخل هر زیرسیستم از تعداد مشخصی از اجزا می‌توان استفاده کرد.
- در این سیستم یک تعمیرکار داریم که قادر به تعمیر تمامی اجزاست.

### ۲.۲. فرموله کردن مسئله

یک سیستم سری با  $S$  زیرسیستم مختلف را در نظر بگیرید. برای زیرسیستم  $i$ ،  $N_i$  تکنولوژی با ویژگی‌های مختلف، از جمله هزینه و دسترس‌پذیری، وجود دارد  $C_{ij}$ ،  $w_{ij}$  و  $R_{ij}$  را به ترتیب هزینه، وزن و دسترس‌پذیری زیرسیستم  $i$  (وقتی که از تکنولوژی  $j$  استفاده می‌شود) تعریف می‌کنیم. در کل، سیستم سری - موازی دارای دو هدف است: ۱. بیشینه‌سازی دسترس‌پذیری سیستم  $(A_s)$ ؛ ۲. کمینه‌سازی هزینه سیستم  $(C_s)$ . کل بودجه‌ی در دسترس را  $B$  تعریف می‌کنیم. مسئله‌ی بهینه را انتخاب تنها یک تکنولوژی برای هر زیرسیستم با بیشترین مقدار دسترس‌پذیری سیستم به شرط در دسترس بودن بودجه تعریف می‌کنیم. به منظور فرموله کردن مسئله در یک بیان ریاضی، متغیر تصمیم‌گیری  $X_{ij}$  را چنین تعریف می‌کنیم:  $X_{ij}$ : متغیر باینری که اگر زیرسیستم  $i$  تکنولوژی  $j$  (جزء) را استفاده کند معادل ۱، در غیر این صورت صفر خواهد بود. سپس مسئله را به صورت مسئله‌ی برنامه‌ریزی عدد صحیح باینری با تابع هدف غیرخطی فرموله می‌کنیم:

$$\text{Max } A_s(\lambda, \mu, k)$$

$$\text{Max } C_s(\lambda, \mu, k)$$

s. t.

$$\sum_{i=1}^s \sum_{j=1}^{N_i} C_{ij} X_{ij} \leq B \quad (1)$$

$$\sum_{j=1}^{N_i} X_{ij} = 1; \quad i = 1, 2, \dots, s \quad (2)$$

$$\sum_{i=1}^s \sum_{j=1}^{N_i} W_{ij} X_{ij} \leq W; \quad \forall i, j \quad (3)$$

$$X_{ij} \in \{0, 1\}; \quad \forall i, j \quad (4)$$

محدودیت ۱ بیان‌گر بودجه، محدودیت ۲ بیان‌گر محدودیت‌های چند گزینه‌ی، محدودیت ۳ بیان‌گر رعایت وزن و محدودیت ۴ بیان‌گر متغیرهای تصمیم‌گیری هستند.

### ۳. الگوریتم بهینه‌یابی ژنتیک چندهدفه مبتنی بر

#### شبیه‌سازی تبرید

برای حل مسائل با تخصیص دسترس‌پذیری و بهینه‌سازی مطابق مدل ریاضی ارائه شده در بخش قبل، در اینجا «رویکرد ژنتیک چندهدفه مبتنی بر شبیه‌سازی تبرید» (SAMOGA) ارائه شده است. متغیرهای تصمیم بهینه که باید مشخص شوند

رتبه‌بندی به سه قسمت تقسیم شده‌اند: ۱. جداسازی تمام موارد در یک جمعیت و تقسیم آنها به دو گروه شدنی، که محدودیت‌ها را ارضا می‌کنند (A)؛ ۲. رتبه‌بندی موارد شدنی/نشدنی با استفاده از رویکرد دسته‌بندی حیابی و بر اساس اولویت توابع هدف (B)؛ ۳. یکی کردن موارد شدنی و نشدنی رتبه‌بندی شده (C).

دو الگوریتم رتبه‌بندی به نام‌های رتبه‌بندی I و رتبه‌بندی II دارای ساختاری مشابه در قسمت A هستند. اگر یک مورد حداقل بتواند به سطح قابل قبولی از تمامی توابع هدف مانند  $\mathcal{R}_{S1}$ ,  $\mathcal{R}_{S2}$ ,  $C_{max}$  دست پیدا کند، این مورد به مجموعه مقادیر قابل قبول  $fit_{pass}$  اضافه می‌شود. در غیر این صورت در گروه موارد غیر قابل قبول  $fit_{unpass}$  قرار می‌گیرد. بعد از آن، قسمت B با استفاده از روش دسته‌بندی حیابی و بر اساس اولویت توابع هدف این موارد را رتبه‌بندی می‌کند. ترتیب اولویت توابع به شکل  $C_s \succ A_s$  برای مورد I و  $G \succ A_s$  برای مورد II است و نماد « $\succ$ » به معنای برتری است. در قسمت C موارد شدنی و نشدنی با هم ترکیب می‌شود.

### ۴.۳. طراحی همسایگی

در استراتژی عمومی تکاملی، تمام کروموزوم‌های موجود در یک جمعیت باید عملگرهای ژنتیک را انجام دهند. اما با توجه به سازوکار تکاملی طراحی همسایگی، تنها کروموزوم‌های موجود در مرز اطراف کروموزوم‌های خاص باید عملگرهای ژنتیک را انجام دهند تا به این ترتیب کارایی اجرا افزایش چشمگیری داشته باشد. همسایگی به معنای مرز اطراف کروموزومی است به نام «کروموزوم مرکزی،  $Y_c$ » در یک جمعیت. بدین منظور مراحل زیر را دنبال می‌کنیم:

- اولین جواب موجود در جمعیت انتخاب می‌شود.
- یک عدد تصادفی حقیقی در فاصله‌ی صفر و ۱ تولید می‌کنیم.
- اگر  $r < Pm$  آنگاه جواب مورد نظر تحت عملگر همسایگی قرار می‌گیرد؛ به این صورت که یکی از ژن‌های مربوط به نوع یا تعداد اجزاء در یک زیرسیستم انتخاب می‌شود و جایگزین مقدار دیگری می‌شود که به صورت تصادفی و موجه تولید شده‌اند.

### ۵.۳. عملگرهای ژنتیک

#### ۱.۵.۳. عملگر تقاطع

بخشی از فرایند تکامل در طبیعت بدین صورت است که کروموزوم‌هایی به عنوان والدین انتخاب شده و با هم ترکیب می‌شوند. در واقع باید کروموزوم‌های انتخاب شده را تقاطع دهیم. برای تولید فرزند جدید در هر تکرار از الگوریتم، عملگر تقاطع یکنواخت<sup>۴</sup> پیاده‌سازی شده است. از این عملگر برای انتقال خصوصیات مناسب بعضی از ژن‌ها که در طول کروموزوم پراکنده شده‌اند، استفاده می‌شود. در تقاطع یک و دونقطه‌یی، انتقال خصوصیات به صورت جابه‌جایی قسمتی از رشته مورد نظر انجام می‌گیرد. در شرایطی که ژن‌ها با خصوصیات مناسب در طول رشته پراکنده شده باشند، استفاده از تقاطع یکنواخت می‌تواند مفیدتر از سایر روش‌های تقاطع باشد. عملگر تقاطع یکنواخت، به صورت تصادفی بعضی از بیت‌ها را در طول رشته‌های والدین به منظور تولید فرزندان جابه‌جا می‌کند. در شکل ۴ نحوه‌ی اجرایی عملگر تقاطع یکنواخت با یک مثال عددی نشان داده شده است.

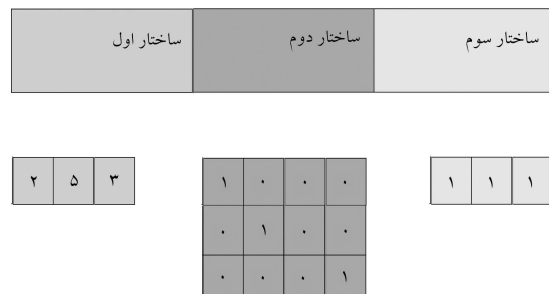
الگوریتم ژنتیک است که در موفقیت و نحوه‌ی پیاده‌سازی الگوریتم تأثیر فراوانی دارد. روش‌های مختلفی برای کد کردن جواب‌های مسئله و طراحی کروموزوم وجود دارد. در حالت کلی هنگام طراحی یک روش نمایش کروموزوم باید موارد زیر را در نظر گرفت:

- کروموزوم باید چنان طراحی شود که تا حد امکان از اطلاعات مسئله برای کاهش تعداد حالات فضای حل استفاده کند.
- در یک کروموزوم نباید اعمال محدودیت‌های مسئله به راحتی انجام شود، و در اثر اعمال مراحل دیگر الگوریتم، محدودیت‌های مسئله نباید نقض شود.
- تا حد امکان کروموزوم نباید در اثر اجرای مراحل تقاطع و جهش به حالت نامعتبر تبدیل شود. در برخی از مسائل گریز از حالت‌های نامعتبر در نحوه‌ی نمایش امکان‌پذیر نخواهد بود و باید پس از اجرای روش تقاطع و جهش در کروموزوم‌ها تغییر ایجاد کنیم.

در این تحقیق برای نمایش جواب‌های مسئله از یک ساختار استفاده شده است. ساختار اول مربوط به یک بردار  $1 \times n$  است. بدین ترتیب که  $n$  بیانگر تعداد زیرسیستم‌هاست و عدد موجود در ژن اول نشان‌دهنده‌ی تعداد اجزای دهنده‌ی در زیرسیستم اول، و عدد موجود در ژن دوم نشان‌دهنده‌ی تعداد اجزای دهنده‌ی در زیرسیستم دوم و ... الی آخر است. ساختار دوم مربوط به یک ماتریس  $m \times n$  است که در آن  $n$  بیانگر تعداد زیرسیستم‌هاست و  $m$  بیانگر تعداد نوع اجزاست. در هر سطر این ماتریس یک عدد ۱ وجود دارد که نشان‌دهنده‌ی نوع جزء انتخاب شده در زیرسیستم است. ساختار سوم نشان‌گر یک بردار  $1 \times n$  است که در آن  $n$  بیانگر تعداد زیرسیستم‌هاست و عدد موجود در ژن اول نشان‌دهنده‌ی نوع سیاست افزونگی انتخاب شده برای زیرسیستم اول، عدد موجود در ژن دوم نشان‌دهنده‌ی نوع سیاست افزونگی انتخاب شده برای زیرسیستم دوم است و ... الی آخر. به عنوان مثال شکل ۳ نشان‌دهنده‌ی کروموزومی است که بر اساس آن گاهی در زیرسیستم اول ۲ عدد جزء نوع ۱، در زیرسیستم دوم ۵ عدد جزء نوع ۲، و در زیرسیستم سوم ۳ عدد جزء نوع ۴ قرار دارد.

### ۳.۳. رتبه‌بندی جواب‌ها

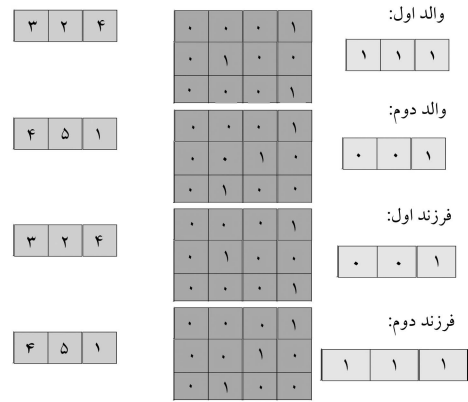
ما برای یافتن جواب‌های بهینه‌ی پارتو در یک جمعیت، دو الگوریتم رتبه‌بندی ارائه می‌کنیم. الگوریتم رتبه‌بندی می‌تواند در زمینه‌ی انجام مقایسه میان دو مورد و شناسایی موارد غیرغالب به ما کمک کند. به منظور یافتن تعداد بیشتری از جواب‌های بهینه‌ی پارتو، باید تنوع فضای جواب را افزایش داد. بنابراین جواب‌های نشدنی فعلی در الگوریتم‌های رتبه‌بندی باقی می‌مانند. این کار به جواب‌های نشدنی امکان بهینه شدن و تبدیل آن به جواب‌های قابل قبول‌تر را می‌دهد. قواعد پایه‌ی الگوریتم



شکل ۳. نحوه‌ی نمایش جواب.

### ۶.۳. ارزیابی تابع هدف دسترس پذیری

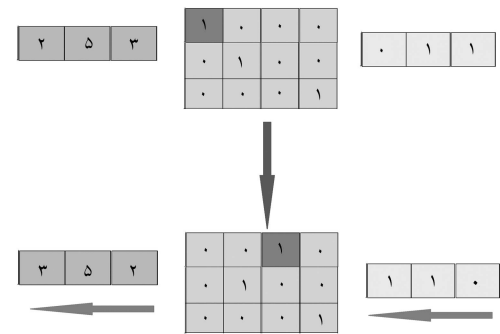
یک سیستم سری - موازی برای اولین بار زمانی از کار می افتد که یکی از زیرسیستم های آن از کار بیفتد؛ به عبارتی هر کدام از زیرسیستم ها که زودتر از دیگر زیرسیستم ها از کار بیفتد کل سیستم خراب می شود. تمامی روابط و معادلات برای محاسبه ی As یک سیستم با احتمالات خرابی و تعمیر نمایی به کمک تحلیل های ریاضی و آماری امکان پذیر است و در صورت غیرنمایی بودن این امکان وجود ندارد؛ از این رو از رویکرد شبیه سازی برای محاسبه ی این مقدار استفاده شده است. یادآور می شود که برازندگی هر کروموزوم برابر با میانگین برازندگی های آن در  $n$  بار شبیه سازی آن کروموزوم است. بدین ترتیب قادریم میزان شایستگی (برازندگی) هر کروموزوم را بدون در نظر گرفتن هیچگونه فرض غیرواقعی محاسبه کنیم.



شکل ۴. نمونه یی از عملگر تقاطع اعمالی.

### ۴. آزمایشات محاسباتی

در این بخش با استفاده از مسائل نمونه تصادفی تولید شده، عملکرد الگوریتم ها را در حل مدل ارزیابی می کنیم. دو الگوریتم فراابتکاری ژنتیک با مرتب سازی نامغلوب (NSGA-II) و بهینه سازی اجتماع ذرات چندهدفه (MOPSO) به منظور ارزیابی کارایی الگوریتم پیشنهادی مورد استفاده قرار می گیرند. برنامه نویسی، محاسبات و اجرای برنامه ی الگوریتم ها در نرم افزار Matlab ۷/۸ و ED ۸/۱ (برای انجام شبیه سازی) و توسط یک PC با چهار گیگابایت RAM و پردازشگر Core i۷, ۲ duo ۲٫۲ GHz انجام شده است.



شکل ۵. نمونه یی از کاربرد عملگر جهش پیشنهادی.

### ۲.۵.۳. عملگر جهش

اعمال جهش به هنگام حرکت از جمعیت حاضر به جمعیت جدید باعث افزایش میزان تنوع در جمعیت جدید می شود و این تنوع اساس تکامل و پیشرفت در رسیدن به جواب نهایی است. عملگر جهش تضمین می کند که الگوریتم ژنتیک به جواب بهینه عمومی یا جواب های نزدیک به آن برسد. زیرا با اعمال این عملگر، نقاط اطراف کروموزوم های جمعیت نیز در مجموعه فضای جست و جو قرار خواهند گرفت و در واقع یک جست و جوی همسایگی انجام گرفته و فضای جست و جو را برای رسیدن به بهینه مطلق توسعه می دهد. این فرایند به جمعیت اولیه ویژگی هایی می دهد که ممکن است در کروموزوم های جمعیت وجود نداشته باشد؛ همچنین از همگرایی سریع الگوریتم ژنتیک در بهینه های محلی، قبل از آن که تمام سطوح هزینه  $5$  را جست و جو کند، ممانعت می کند. عملگر جهش مورد استفاده از نوع تصادفی است. بدین صورت که برای هر کروموزوم یک عدد تصادفی بین صفر و ۱ با توجه به ابعادش تولید کرده، آنگاه چنانچه اعداد تولید شده ی هر ژن کوچک تر از مقداری خاص بود، روی آن ژن مربوطه جهش صورت می گیرد و مجدداً مقدار خواهد گرفت. شکل ۵ بیانگر نحوه ی اجرای عملگر جهش است.

گام های اعمال جهش روی کروموزوم ها عبارت اند از:

۱. اولین کروموزوم موجود در جمعیت انتخاب می شود.

۲. یک عدد تصادفی حقیقی در فاصله ی صفر و ۱ تولید کنید.

اگر  $r < P_m$ ، کروموزوم مورد نظر تحت عملگر جهش قرار می گیرد؛ به این صورت که تمامی ژن های موجود در ساختار اول و سوم انتخاب شده و به طور معکوس مرتب می شوند. یکی از ژن های ساختار سوم انتخاب، و با مقدار دیگری که به صورت تصادفی و موجه تولید شده، جایگزین می شود.

### ۱.۴. شاخص های ارزیابی عملکرد الگوریتم های چندهدفه

چنانچه گفته شد، الگوریتم های چندهدفه ی پیشنهادی بر مبنای پارتو جست و جو و کار می کنند. از طرفی می دانیم پارتو نهایی یافت شده توسط الگوریتم ها باید دارای دو خاصیت همگرایی و تنوع گرایی خوب باشند. برای این منظور لازم است تا شاخص های مختلفی بررسی شوند تا شناخت جامعی از عملکرد یک الگوریتم چندهدفه حاصل شود. در این بخش، به چند نمونه از این معیارها اشاره می کنیم.

البته پیش از تعریف این شاخص ها لازم است تا پارامترهای زیر تعریف شوند:

$Q$ : مجموعه حل های نامغلوب سطح اول

$|Q|$ : تعداد حل های موجود در مجموعه  $Q$

$M$ : تعداد اهداف

$f_i^z$ : مقدار  $z$  امین تابع هدف از جواب  $i$  ام

### ۱.۱.۴. فاصله گذاری

این شاخص یک معیار اندازه گیری برای سنجش فاصله نسبی جواب های متوالی است. در رابطه ی ۴ نحوه ی محاسبه ی این شاخص ارائه شده است:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \bar{d})^2} \text{ where}$$

$$d_i = \min_{k \in n, k \neq i} \sum_{m=1}^M |f_m^i - f_m^k| \text{ and } \bar{d} = \frac{1}{n} \sum_{i=1}^n d_i \quad (5)$$

فاصله ی اندازه گیری شده برابر است با کمترین مقدار مجموع قدر مطلق تفاضل در مقادیر تابع هدف بین  $i$  امین جواب و جواب های واقع در مجموعه ی نامغلوب نهایی. این معیار، انحراف معیارهای مقادیر مختلف  $d_i$  را اندازه گیری می کند. زمانی

زیرسیستم بین ۲ تا ۵ زیرسیستم، بیشترین تعداد اجزا در هر زیرسیستم بین ۲ تا ۴، تعداد انواع مختلف اجزا برای هر زیرسیستم بین ۲ تا ۵ و برای وزن و هزینه به ترتیب اعداد بین ۵۰۰ تا ۱۰۰۰ و ۱۰۰ تا ۵۰۰ انتخاب شده است. در شکل ۸ کارایی الگوریتم‌ها در تمامی شاخص‌ها نشان داده شده است. از آنجا که معیار تعداد جواب پارتو هرچه بیشتر باشد بهتر است می‌توان نتیجه گرفت که الگوریتم NSGA-II بهترین عملکرد را در بین سایر الگوریتم‌ها دارد و می‌تواند تعداد جواب‌های بیشتری را در صف اول پارتو پیدا کند. مطابق شکل ۸ می‌توان گفت که در معیار تنوع الگوریتم SAMOGA عملکرد بهتری در مقایسه با الگوریتم‌های NSGA-II و MOPSO دارد. شکل ۸ برتری الگوریتم SAMOGA نسبت به دو الگوریتم دیگر را در مدت زمان اجرا نشان می‌دهد. طبق این شکل الگوریتم SAMOGA در مسائل اندازه کوچک عملکرد مشابه با الگوریتم NSGA-II و MOPSO دارد ولی با افزایش اندازه مسئله عملکرد الگوریتم SAMOGA به شدت افزایش می‌یابد، به طوری که اختلاف محسوسی با دو الگوریتم دیگر ایجاد می‌کند.

### ۳.۴. تحلیل واریانس

به منظور بررسی و مقایسه‌ی دقیق‌تر از تحلیل‌های آماری استفاده کرده‌ایم. همان‌طور که اشاره شد، در این حیطة از تحلیل واریانس بهره جست‌ه‌ایم. به این صورت که الگوریتم‌ها در رابطه با هر معیار توسط نرم‌افزار مورد آنالیز واریانس قرار می‌گیرند و نتایج آن‌ها تحلیل می‌شود. در این راستا ابتدا نتایج الگوریتم‌ها در اندازه‌های مختلف با استفاده از معیار درصد انحراف نسبی (RPD) نرمال‌سازی می‌شوند. مقدار RPD نشان می‌دهد که جواب‌ها در هر الگوریتم تا چه اندازه از بهترین جواب حاصله فاصله دارند. درصد انحراف نسبی بر طبق فرمول ۶ محاسبه می‌شود:

$$RPD_{ij} = \frac{Sol_{ij} - Sol_{jmin}}{Sol_{jmin}} \quad (7)$$

که در آن  $i$  شماره الگوریتم و  $j$  شماره سائز مثال است. سپس الگوریتم‌ها در رابطه با هر معیار مورد آنالیز واریانس قرار می‌گیرند و نتایج آن‌ها تحلیل می‌شود. کمتر از ۰/۰۵ درصد بودن مقدار p-value نشان‌دهنده‌ی معنادار بودن اختلاف بین پاسخ‌های الگوریتم‌ها در رابطه با آن معیار خاص است و در غیر این صورت می‌توان گفت بین عملکرد الگوریتم‌ها در رابطه با آن معیار تفاوت معناداری وجود ندارد. ابتدا بررسی می‌شود که آیا میانگین الگوریتم‌ها در هر یک از شاخص‌ها با هم متفاوت است یا خیر (فرض صفر و ۱):

$$H_0 : \mu_1 = \mu_2 = \mu_3$$

$$H_1 : \mu_i \neq \mu_j, i, j = 1, 2, 3 \text{ and } i \neq j$$

نتایج به دست آمده از تحلیل واریانس در جدول ۱ آمده است. در این جداول خروجی تحلیل واریانس به صورت F-value و p-value های به دست آمده گزارش

جدول ۱. آنالیز واریانس برای نتایج.

Test Results	P - Value	F - Value	Metric
فرض صفر رد می‌شود.	۰/۰۰۰	۷۵/۳۶	Diversity
فرض صفر رد می‌شود.	۰/۰۰۰	۳۷/۷۵	NOS
فرض صفر رد می‌شود.	۰/۰۰۰	۱۱/۵۹	Spacing
فرض صفر رد می‌شود.	۰/۰۰۰	۱۸/۵۸	Time

که جواب‌ها به‌طور یکنواخت در کنار هم باشند آنگاه مقدار  $S$  کوچک خواهد بود. بنابراین الگوریتمی که جواب‌های نامغلوب نهایی آن دارای مقدار  $S$  کمتری است، بهتر خواهد بود.

### ۲.۱.۴. تنوع

این معیار طول قطر مکعب فضایی را که با مقادیر انتهایی اهداف برای مجموعه‌ی جواب‌های نامغلوب کاربرد دارد، اندازه‌گیری می‌کند.

$$D = \sqrt{\sum_{j=1}^m (\max_i f_i^m - \min_i f_i^m)^2} \quad (6)$$

در حالت دوهدفه، این معیار برابر فاصله اقلیدسی بین دو جواب مرزی در فضای هدف است. هرچه این معیار بزرگ‌تر باشد، بهتر است. در شکل ۶ تفاوت فاصله اندازه گرفته شده در دو روش فاصله‌گذاری و تنوع نشان داده شده است.

### ۳.۱.۴. معیار تعداد جواب‌های پارتو (NOS)

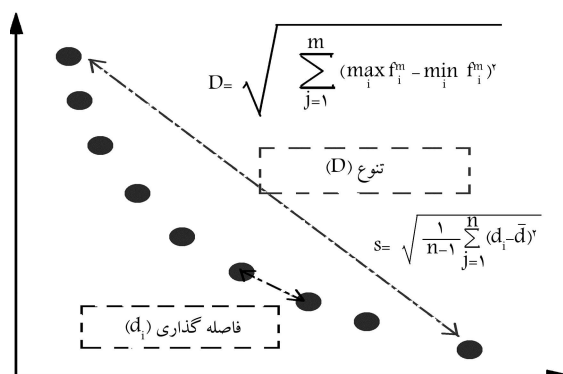
هرچه تعداد جواب‌های پارتو یک روش بیشتر باشد، مطلوب‌تر است. تعداد جواب‌های پارتو را با NOS نشان می‌دهیم. شکل ۷ اشاره به جواب‌های پارتو دارد.

### ۴.۱.۴. زمان محاسباتی

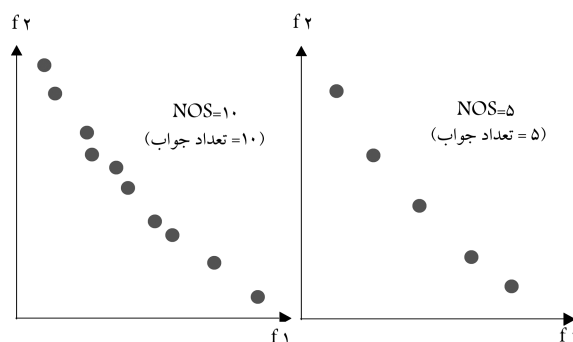
زمان رسیدن به جواب‌های نزدیک به همیشه همواره یکی از مهم‌ترین معیارهای بررسی کارایی الگوریتم‌های فراابتکاری بوده است.

### ۲.۴. ارزیابی الگوریتم‌ها

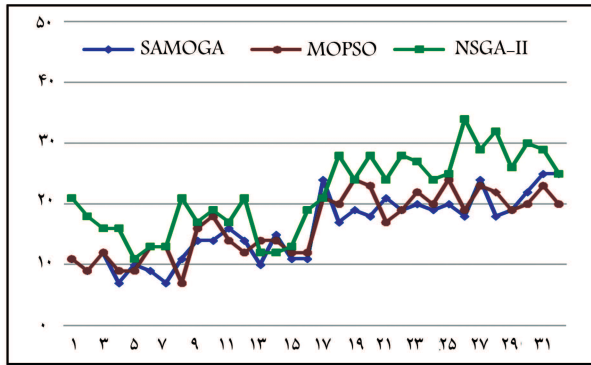
به‌منظور تشریح کارایی الگوریتم‌های پیشنهادی مثال‌های عددی به صورت ۳۲ آزمایش مشابه<sup>[۲۱]</sup> طراحی شده است. در این آزمایش‌ها، تعداد زیرسیستم‌ها در هر



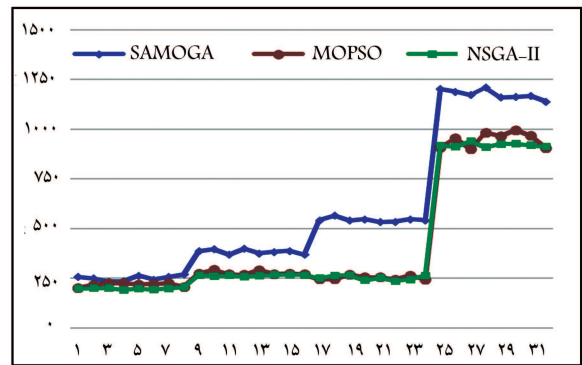
شکل ۶. معیارهای بیشترین گسترش و فاصله گذاری.



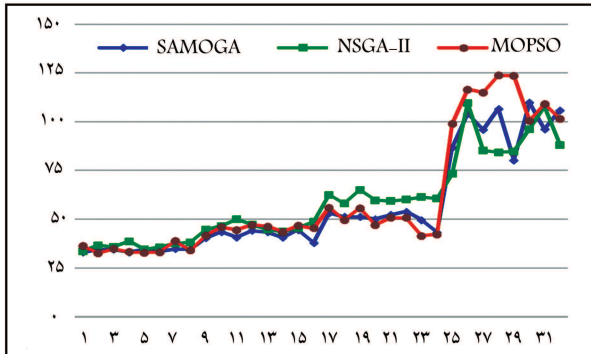
شکل ۷. تعداد جواب‌های پارتو در مسئله چندهدفه.



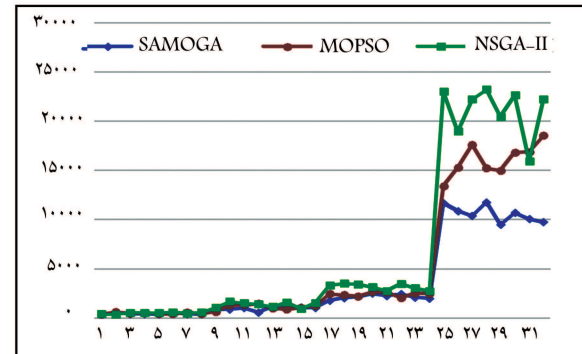
ب) NOS؛



الف) تنوع؛

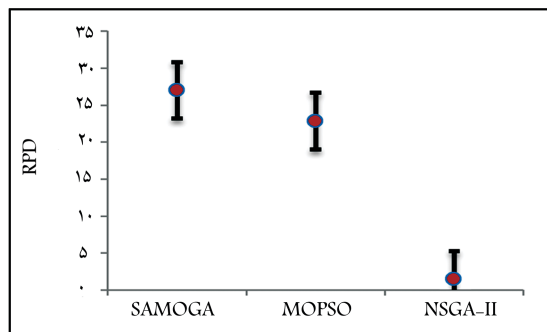


د) فاصله گذاری.

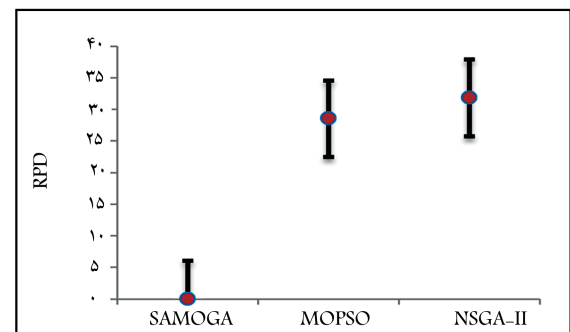


ج) زمان حل؛

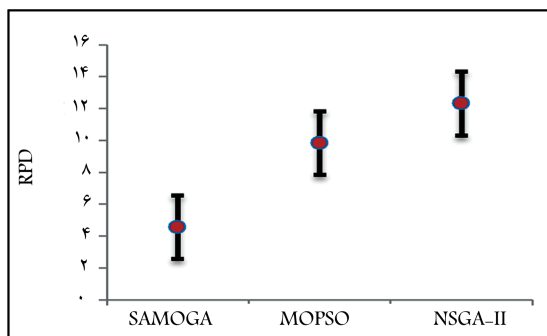
شکل ۸. مقایسه‌ی کارایی الگوریتم‌های پیشنهادی در معیارهای مورد بررسی.



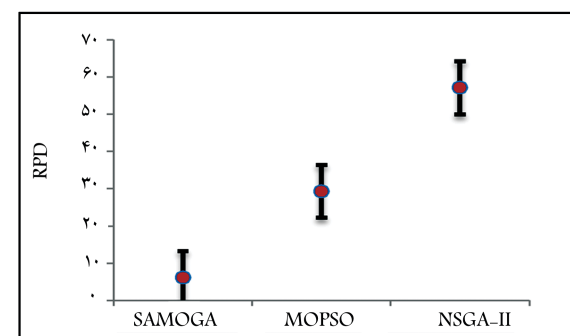
ب) NOS؛



الف) تنوع؛



د) فاصله گذاری.



ج) زمان حل؛

شکل ۹. مقایسه‌ی الگوریتم‌های پیشنهادی در معیارهای مورد بررسی بر اساس آزمون توکی.

افزونگی برای سیستم‌هایی با ساختار سری - موازی ارائه شد که در آن تا حدود زیادی حجم محاسبات کاهش یافته و دست‌یابی به یک جواب مطلوب با سهولت بیشتر را امکان‌پذیر کرده است. از آن‌جا که توزیع‌های در نظر گرفته شده برای خرابی و تعمیر غیرنمایی است، تابع هدف دسترس‌پذیری مسئله با روش شبیه‌سازی محاسبه می‌شود.

برای بهینه‌سازی مسئله‌ی تخصیص افزونگی که به صورت یک مدل دوهدفه ارائه شد، از الگوریتم فراابتکاری SAMOGA استفاده شد. الگوریتم ارائه شده روی مثال‌های طراحی شده آزمایش شد و بعد از به دست آوردن نتایج از لحاظ آماری با دو الگوریتم MOPSO و NSGA-II مورد مقایسه قرار گرفت. روش پیشنهادی توانست در کمترین زمان ممکن به با کیفیت‌ترین جواب‌ها دست یابد. یکی از پیشنهادات قابل ارائه برای انجام تحقیقات آتی اضافه کردن ویژگی‌های جدید به سیستم و بازنگری مدل بهینه‌سازی تحقیق و حل مدل جدید است. از جمله این ویژگی‌ها می‌توان اشاره کرد به:

۱. امکان تخصیص هم‌زمان انواع مختلف از اجزا در هر زیرسیستم؛
۲. تعریف خرابی و تعمیر اجزا به صورت اعداد فازی؛
۳. انواع تخصیص‌های مختلف خدمات تعمیر و تعداد تعمیرکاران؛
۴. امکان تخصیص اجزای چندحالتی در زیرسیستم‌ها.

شده است. لذا خروجی‌های آماری بیان می‌دارد که در توزیع نمای الگوریتم‌ها در تمامی معیارها کاملاً قابلیت رقابت با یکدیگر را دارند.

به منظور تعیین کارایی الگوریتم‌ها در شاخص‌هایی که دارای تفاوت معنادار هستند از آزمون توکی استفاده می‌کنیم. نتایج به دست آمده از آزمون توکی به شکل فواصل اطمینان هم‌زمان توکی در شکل ۹ آمده است. بر این اساس، می‌توان گفت که الگوریتم SAMOGA در معیارهای زمان محاسباتی، بیشترین گسترش و فاصله‌گذاری عملکرد بالاتری در مقایسه با الگوریتم‌های NSGA-II و MOPSO دارد. در صورتی که در معیار تعداد جواب پارتو، الگوریتم NSGA-II عملکرد بهتری در مقایسه با الگوریتم‌های MOPSO و SAMOGA دارد.

## ۵. نتیجه‌گیری

در بررسی مدل‌های قابلیت اطمینان، نرخ‌های خرابی عناصر متفاوت است و برای کل سیستم، حالت‌های در نظر گرفتن انواع نرخ خرابی به طور فزاینده‌ی افزایش می‌یابند. فضای جواب این‌گونه مسائل بسیار بزرگ است و محاسبه‌ی مقداری دقیق برای قابلیت اطمینان در هر لحظه از زمان سخت است. در این تحقیق، یک روش جدید برای انتخاب بهترین اجزا با نرخ خرابی و تعمیر غیرنمایی و بهترین سیاست

## پانویس‌ها

1. Mean Time to Failure
2. Survivability
3. NP-hard
4. Uniform crossover
5. Cost surface

## منابع (References)

1. Tekiner-Mogulkoc, H. and Coit, D.W. "System reliability optimization considering uncertainty: minimization of the coefficient of variation for series-parallel systems", *Reliability, IEEE Transactions on Reliability*, **60**(3), pp. 667-674 (2011).
2. Kuo, W. and Prasad, V.R. "An annotated overview of system-reliability optimization", *IEEE Transaction on Reliability*, **2**, pp. 176-187 (2000).
3. Fyffe, D.E., Hines, W.W. and Lee, N.K. "System reliability allocation and a computational algorithm", *IEEE Transaction on Reliability*, **R-17**(2), pp. 64- 69 (1968).
4. Coelho, L.D.S. "Reliability-redundancy optimization by means of a chaotic differential evolution approach", *Chaos, Solitons and Fractal*, **41**(2), pp. 594- 602 (2009).
5. Zou, D., Gao, L., Li, S. and et al. "An effective global harmony search algorithm for reliability problems", *Expert Systems with Applications*, **38**(4), pp. 4642-4648 (2011).
6. Ramirez-Marquez, J.E. and Coit, D.W. "A heuristic for solving the redundancy allocation problem for multi-state series-parallel system", *Reliability Engineering and System Safety*, **83**(3), pp. 314-349 (2004).
7. Nahas, N., Nourelfath, M. and Ait-Kadi, D. "Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-parallel systems", *Reliability Engineering and System Safety*, **92**(2), pp. 211-222 (2007).
8. Safaei, N., Tavakkoli-Moghaddam, R. and Kiassat, C. "Annealing-based particle swarm optimization to solve the redundant reliability problem with multiple component choices", *Applied Soft Computing*, **12**(11), pp. 3462-3471 (2012).
9. Liang, Y.C. and Chen, Y.C. "Redundancy allocation of series parallel systems using a variable neighborhood search algorithm", *Reliability Engineering and System Safety*, **9**(3), pp. 323-331 (2007).
10. Ha, C. and Kuo, W. "Reliability redundancy allocation: An improved realization for non-convex nonlinear programming problems", *European Journal of Operation Research*, **171**(1), pp. 24-38 (2006).
11. Soltani, R., Sadjadi, S.J. and Tofigh, A.A. "A model to enhance the reliability of the serial parallel systems with component mixing", *Applied Mathematical Modelling*, **38**, pp. 1064-1076 (2013).
12. Liang, Y.C., Lo, M.H. and Chen, Y.C. "Variable neighborhood search for redundancy allocation problems",



- IMA Journal of Management Mathematics*, **18**(2), pp. 135-155 (2007).
13. Juang, Y.S., Lin, S.S. and Kao, H.P. "A knowledge management system for series-parallel availability optimization and design", *Expert System with Applications*, **34**(1), pp. 181-193 (2008).
  14. Zhang, E., Wu, Y. and Chen, Q. "A practical approach for solving multi-objective reliability redundancy allocation problems using extended bare-bones particle swarm optimization", *Reliability Engineering & System Safety*, **127**, pp. 65-76 (2014).
  15. Chambari, A., Najafi, A.A., Rahmati, S.H.A. and et al. "An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies", *Reliability Engineering & System Safety*, **119**, pp. 158-164 (2013).
  16. Kuo, W. and wan, R. "Recent advances in optimal reliability allocation", *IEEE Transaction on System, Man and Cybernetics-part a: System and Humans*, **39**, pp. 143-156 (2007).
  17. Guedenko, B., Ushakov, I. *Probabilistic Reliability Engineering. A Wiley-Interscience publication*, New York (1995).
  18. Najafi, A.A., Karimi, H., Chambari, A. "Two meta-heuristics for solving the Reliability redundancy allocation problem to Maximize mean time to failure of a series-parallel System", *Scientia Iranica*, **20**(3), pp. 832-838 (2013).
  19. Kim, H. and Kim, P. "Reliability-redudancy allocation problem considering optimal redundancy strategy using parallel genetic algorithm", *Reliability Engineering & System Safety*, **159**, pp. 153-160 (2017).
  20. Feizabadi, M. and Jahromi, A.E. "A new model for reliability of series-parallel system with non-homogeneous components", *Reliability Engineering and System Safety*, **157**, pp. 101-112 (2017).
  21. Mousavi, S.M. Alikar, N., Akhavan Niaki, S.T. and et al. "Two tuned multiobjective meta-heuristic algorithms for solving a fuzzy multi-state redundancy allocation problem under discount strategies", *Applied Mathematical Modelling*, **39**(22), pp. 6968-6989 (2015).