

## بهینه سازی قابلیت اطمینان سیستم سری - موازی با زیر سیستم های $k$ -out- $n$ با در نظر گرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار

مانی شریفی،\* قاسم چراغ\*\*

تاریخ دریافت (۹۲/۹/۲۳) تاریخ پذیرش (۹۳/۳/۵)

### چکیده

در این مقاله مدل جدیدی برای مسئله تخصیص افزونگی با ساختار سری- موازی و زیر سیستم های  $k$ -out- $n$  با در نظر گرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار ارائه می شود. در این نوع نرخ خرابی با افزایش تعداد اجزای در حال کار به دلیل تقسیم بار (فشار) بر روی تعداد بیشتری از اجزاء، نرخ خرابی هر یک از اجزاء کاهش می یابد. همچنین برای هر یک از زیر سیستم ها دو نوع سیاست افزونگی آماده به کار سرد و فعال در نظر گرفته شده است. هدف از حل مدل ارائه شده تعیین استراتژی افزونگی، نوع و تعداد جزء مازاد تخصیص یافته برای هر زیر سیستم به منظور حداکثر کردن قابلیت اطمینان کل سیستم تحت محدودیت های وزن و هزینه می باشد. به دلیل  $NP$ -hard بودن مسئله تخصیص افزونگی از الگوریتم فرا ابتکاری ژنتیک (GA) برای حل مدل و از روش سطح پاسخ (RSM) برای تنظیم پارامترهای موثر بر این الگوریتم استفاده شده است. در انتها با ذکر یک مثال عددی به بررسی نتایج به دست آمده از حل مدل ارائه شده می پردازیم.

**کلمات کلیدی:** مسئله تخصیص افزونگی، زیر سیستم  $k$ -out- $n$ ، سیاست افزونگی، نرخ خرابی وابسته به تعداد اجزای در حال کار، الگوریتم GA.

\* استادیار گروه مهندسی صنایع و مکانیک، دانشگاه آزاد اسلامی، واحد قزوین، ایران (نویسنده مسئول) (m.sharifi@Qiau.ac.ir)

\*\* دانشجوی کارشناسی ارشد، گروه مهندسی صنایع و مکانیک، دانشگاه آزاد قزوین، ایران

## مقدمه

قابلیت اطمینان یکی از مهم‌ترین مسائلی است که امروزه در طراحی سیستم‌ها مورد توجه قرار می‌گیرد؛ از این رو تحقیقات گسترده‌ای در مورد عوامل موثر بر آن انجام شده و راه کارهایی نیز برای افزایش آن مطرح شده است که از جمله می‌توان به تخصیص اجزای مازاد به سیستم اشاره کرد. فیف و همکاران برای اولین بار این موضوع را در قالب مسئله تخصیص افزونگی (RAP) بیان کردند که به عنوان یکی از مشهورترین مدل‌های قابلیت اطمینان مطرح می‌شود [۱]. در این مسئله هدف این است که با افزایش اجزای مازاد به سیستم تحت محدودیت‌هایی مانند وزن، بودجه و غیره قابلیت اطمینان کل سیستم حداکثر شود. این مسئله جزء مسائل برنامه ریزی عدد صحیح غیرخطی به حساب می‌آید. چرن در سال ۱۹۹۲ نشان داد مسئله تخصیص افزونگی با افزایش تعداد زیر سیستم‌ها در رسته مسائل NP-Hard قرار می‌گیرد [۲].

از جمله موثرترین عواملی که به صورت مستقیم بر روی قابلیت اطمینان سیستم تأثیر می‌گذارد، نرخ خرابی اجزاء است. در ادبیات گذشته RAP به طور کلی نرخ خرابی به دو صورت در نظر گرفته شده است:

۱) نرخ خرابی ثابت (CFR) که نرخ خرابی هر یک از اجزاء ثابت و مستقل از یکدیگر است و از تابع توزیع نمایی پیروی می‌کند.

۲) نرخ خرابی وابسته به زمان که با افزایش زمان، نرخ خرابی اجزاء تغییر می‌کند و به دو صورت افزایشی (IFR) و کاهش‌شی (DFR) است مانند توزیع ارلنگ، وایبول و غیره.

نرخ خرابی ثابت: میسرا و شارما در سال ۱۹۹۱ یک مسئله تخصیص افزونگی را برای ساختار سری- موازی با زیر سیستم‌های k-out-of-n در نظر گرفتند؛ در مدل آنها سیاست افزونگی فعال و امکان تخصیص اجزای یکسان به هر زیر سیستم فرض شده بود، آنها برای حل مدل خود از برنامه‌ریزی صفرویک استفاده کردند [۳]. در سال ۱۹۹۵ کویت واسمیت مدل جدیدی برای مسئله تخصیص افزونگی را با ساختار سیستم سری- موازی و زیر سیستم‌های k-out-of-

n با امکان تخصیص اجزای غیریکسان به هر زیر سیستم و سیاست افزونگی فعال ارائه کردند [۴]. کویت ویو در سال ۲۰۰۰ برای اولین بار مدلی با ساختار سری- موازی و زیر سیستم‌های k-out-of-n را ارائه کردند که در آن به طور همزمان دو سیاست افزونگی فعال و آماده به کار سرد به صورت از پیش تعیین شده برای هر زیر سیستم در نظر گرفته شده بود؛ آنها این مدل را با استفاده از برنامه‌ریزی عدد صحیح حل کردند [۵]. هسیه و یودر سال ۲۰۱۱ یک روش دو مرحله‌ای جدید مبتنی بر مصونیت را برای حل مسئله تخصیص افزونگی تحت محدودیت‌های غیر خطی وزن، هزینه و حجم را ارائه دادند بطوری که در مرحله اول یک الگوریتم مبتنی بر مصونیت را برای حل مسئله تخصیص توسعه دادند و در مرحله دوم از یک رویه جدید برای بهبود جواب‌های مرحله‌ی اول استفاده کردند و در نهایت به مقایسه جواب‌های روش خودشان با چهار مسئله محک پرداختند [۶]. هسیه و یهدر سال ۲۰۱۲ از یک استراتژی جریمه هدایت شده مبتنی بر الگوریتم کلونی زنبور عسل مصنوعی برای مسئله تخصیص افزونگی در سیستم‌های با ساختار سیستم سری- موازی استفاده کردند [۷].

نرخ خرابی وابسته به زمان: کویتدر سال ۲۰۰۱ مدل جدیدی برای مسئله تخصیص افزونگی را با ساختار سیستم سری- موازی با امکان تخصیص اجزای یکسان به هر زیر سیستم و سیاست افزونگی آماده به کار سرد ارائه داد. او تابع توزیع نرخ خرابی اجزاء را ارلنگ (k-Erlang) فرض و مدل خود را با استفاده از برنامه‌ریزی عدد صحیح حل کرد [۸]. در ادامه کویت در سال ۲۰۰۳ با متغیر در نظر گرفتن دو سیاست افزونگی فعال و آماده به کار سرد برای هر یک از زیر سیستم‌ها مدل جدیدی را توسعه داد و با استفاده از برنامه‌ریزی عدد صحیح حل کرد [۹]. توکلی مقدم و صفری در سال ۲۰۰۷ مدل ریاضی جدیدی برای مسئله تخصیص افزونگی با ساختار سیستم سری- موازی را ارائه دادند که در آن امکان تخصیص اجزای غیریکسان وجود داشت. آنان همچنین سیاست افزونگی فعال و آماده به کار سرد را به عنوان یک متغیر تصمیم در نظر گرفتند [۱۰]. توکلی مقدم و صفری در سال ۲۰۰۸ مدل ارائه شده در [۹] را با استفاده از الگوریتم ژنتیک حل کردند، شاخصه مهم الگوریتم آنها در کروموزم و عملگر جهش آن بود [۱۱]. آماری و دیل در سال ۲۰۱۰ مسئله تخصیص افزونگی را با ساختار سیستم

سری- موازی و زیر سیستم‌های  $k$ -out-of- $n$  با امکان تخصیص اجزای غیریکسان به هر زیر سیستم را ارائه کردند. آنها نرخ خرابی اجزاء را وایبولدر نظر گرفتند [۱۲]. آماریدر سال ۲۰۱۲ با استفاده از مفاهیم فرآیندهای شمارشی، روشی را برای ارزیابی قابلیت اطمینان سیستم‌های  $k$ -out-of- $n$  با سیاست آماده به کار سرد ارایه داد در این روش توزیع گاما برای عمر اجزا در نظر گرفته شده است [۱۳]. وقتی در سیستمی تعدادی اجزا با هم کار می‌کنند با خراب شدن یک جز سایر اجزای در حال کار باید با شدت بیشتری کار کنند، در نتیجه نرخ خرابی اجزاء افزایش پیدامی‌کند. در نظر گرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار باعث واقعی تر شدن شرایط مسئله می‌شود [۱۴].

ما در این مقاله برای اولین بار نرخ خرابی ثابت و وابسته به تعداد اجزای در حال کار را برای یک مسئله تخصیص افزونگی با ساختار سری- موازی و زیر سیستم‌های  $k$ -out-of- $n$  در نظر می‌گیریم. در دنیای واقعی از مدل ارائه شده میتوان در سیستم‌هایی استفاده کرد که اجزای آن با هم کار می‌کنند مانند ایستگاه پمپاژ آب یا هواپیمایی که با چند موتور کار می‌کند. در اینگونه سیستم‌ها وقتی جزئی خراب شود به اجزای دیگر فشار بیشتری وارد می‌شود، پس در نظر گرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار باعث واقعی تر شدن مسئله می‌شود. هدف از حل مدل ارائه شده تعیین سیاست افزونگی (فعال یا آماده به کار)، نوع و تعداد اجزاء مازاد تخصیص یافته به هر زیر سیستم برای حداکثر کردن قابلیت اطمینان کل سیستم تحت محدودیت هزینه و وزن است. به دلیل NP-Hard بودن مسئله تخصیص افزونگی ارائه شده از الگوریتم فرا ابتکاری GA برای حل آن استفاده می‌کنیم.

## سیاست افزونگی

سیاست افزونگی به دو دسته سیاست فعال و آماده به کار تقسیم می‌شود. در سیاست افزونگی فعال همه اجزا با هم کار می‌کنند، حتی اگر به همه آنها نیاز نباشد. بنابراین همه آنها در معرض خرابی قرار می‌گیرند. سیاست آماده به کار به سه دسته سرد، گرم و داغ تقسیم می‌شود. در

سیاست آماده به کار سرد، گرم و داغ نرخ خرابی اجزاء مازاد به ترتیب صفر، کمتر و برابر نرخ خرابی اجزاء در حال کار است. در این سیاست حداقل تعداد اجزاء مورد نیاز برای کارکرد سیستم در حال کار بوده و تعدادی اجزاء مازاد در دسترس می‌باشد. هنگامی که یکی از اجزای در حال کار خراب شود، یک جزء مازاد جایگزین آن می‌گردد. در این سیاست افزونگی، از یک سخت افزار (سوئیچ) جهت یافتن اجزاء در حال کار استفاده می‌شود تا بلافاصله جزء مازاد فعال شود. به طور کلی ۲ سناریو برای سوئیچ زدن در سیستم‌هایی با سیاست آماده به کار سرد وجود دارد. در سناریوی ۱ عملکرد سیستم به صورت پیوسته مورد مشاهده قرار می‌گیرد و هنگامی که خرابی یافت شد، سوئیچ جزء مازاد را جایگزین جزء خراب می‌کند. در این مورد خرابی می‌تواند در هر لحظه از زمان اتفاق افتاده و قابلیت اطمینان سوئیچ به صورت  $P$  که یک تابع غیر افزایشی از زمان است نشان داده می‌شود. در سناریوی ۲ احتمال خرابی تنها در لحظه سوئیچ زدن وجود دارد که برابر مقدار ثابت  $P$  است [۹].

در این مقاله دو سیاست افزونگی آماده به کار سرد و فعال برای هر یک از زیرسیستم‌ها در نظر گرفته شده و نیز در سیاست آماده به کار سرد برای سوئیچ زدن از سناریوی دوم استفاده می‌شود.

### فرمول بندی مسئله

مدل ریاضی ارائه شده برای سیستم سری- موازی با زیرسیستم k-out-of-n است که دارای یک محدودیت خطی و یک محدودیت غیر خطی می‌باشد.

#### مفروضات مسئله

- هر زیرسیستم تا زمانی کار می‌کند که حداقل  $k$  جزء در حال کار باشد.
- نرخ خرابی اجزاء وابسته به تعداد اجزای در حال کار می‌باشد.
- قابلیت اطمینان سوئیچ در لحظه تقاضا برای هر زیرسیستم در سیاست افزونگی آماده به کار سرد برابر  $P$  است.
- تنها یک نوع جزء به هر زیرسیستم می‌توان اختصاص داد.

- اجزاء دو وضعیت سالم یا خراب را می‌توانند داشته باشند.
- اجزاء سیستم تعمیر ناپذیرند می‌باشند.
- پارامترهای اجزاء نظیر قابلیت اطمینان، هزینه و وزن معلوم و غیراحتمالی هستند.

نمادها

$s$ : تعداد کل زیر سیستم

$i$ : اندیس زیر سیستم

$k_i$ : حداقل تعداد اجزا در حال کار در زیر سیستم  $i$  ام

$n_i$ : تعداد جزء استفاده شده در زیر سیستم  $i$  ام،  $i = (1, 2, \dots, s)$

$n$ : مجموعه‌ای از  $n_i$ ،  $(n_1, n_2, \dots, n_s)$

$n_{\max, i}$ : حد بالای  $n_i$ ،  $\forall i: n_i \leq n_{\max, i}$

$m_i$ : تعداد نوع در دسترس برای زیر سیستم  $i$  ام،  $i = (1, 2, \dots, s)$

$z_i$ : اندیس نوع جزء استفاده شده برای زیر سیستم  $i$  ام،  $z_i \in (1, 2, \dots, m_i)$

$Z$ : مجموعه‌ای از  $z_i$ ،  $(z_1, z_2, \dots, z_s)$

$t$ : زمان ماموریت

$\lambda_{i, z_i, k_i}$ : نرخ خرابی جزء نوع  $z_i$  ام استفاده شده در زیر سیستم  $i$  زمانی که  $k$  جزء در حال کار می‌باشد.

$w_{i, z_i}$ : وزن هر جزء نوع  $z_i$  ام در زیر سیستم  $i$  ام

$c_{i, z_i}$ : هزینه هر جزء نوع  $z_i$  ام در زیر سیستم  $i$  ام

$\theta_{i, z_i}$ : پارامتر مرتبط با هزینه اتصال داخلی جزء نوع  $z_i$  ام در زیر سیستم  $i$  ام

$W$ : وزن کل سیستم

$B$ : بودجه کل سیستم

$A$ : نشان دهنده مجموعه زیر سیستم‌هایی که سیاست افزونگی فعال استفاده می‌کنند.

$S$ : نشان دهنده مجموعه زیر سیستم‌هایی که سیاست افزونگی آماده به کار سرد استفاده می‌کنند.

- $R_{q_1}$ : قابلیت اطمینان زیر سیستمی که از سیاست افزونگی فعال استفاده می‌کنند.
- $R_{q_2}$ : قابلیت اطمینان زیر سیستمیکه از سیاست آماده به کار سرد استفاده می‌کنند.
- $R_q$ : قابلیت اطمینان کل سیستم
- $P$ : قابلیت اطمینان سویچ در لحظه سوئیچ زدن

### مدل ریاضی

$$\text{Max } R_q(t) = \prod_{i \in A} R_{q_1}(t) \times \prod_{i \in S} R_{q_2}(t) \quad (1)$$

S.t :

$$\sum_{i=1}^s (c_{iz_i} (n_i + e^{\theta_{z_i} n_i})) \leq B \quad (2)$$

$$\sum_{i=1}^s w_{iz_i} n_i \leq W \quad (3)$$

$$n_i \in (k_i, 2, \dots, n_{\max}); i = (1, 2, \dots, s) \quad (4)$$

$$z_i \in (1, 2, \dots, m_i); i = (1, 2, \dots, s) \quad (5)$$

در این مدل (۱) قابلیت اطمینان سیستم را نشان می‌دهد که باید حد اکثر شود. (۲) نشان دهنده هزینه در دسترس است که در آن هزینه‌ای جهت اتصال داخلی بین اجزاء در نظر گرفته شده است [۱۵]. (۳) نیز وزن در دسترس را نشان می‌دهد. (۴) و (۵) نیز به ترتیب حداکثر تعداد اجزاء و نوع در دسترس را برای تخصیص به زیر سیستم نام نشان می‌دهند.

$R_{q_1}$  و  $R_{q_2}$  به ترتیب نشان دهنده قابلیت اطمینان زیر سیستم‌هایی می‌باشد که از سیاست آماده به کار فعال و سرد تبعیتی کند. قابلیت اطمینان یک سیستم با در نظر گرفتن نرخ خرابی ثابت که از سیاست آماده به کار فعال تبعیتی کند و از (۶) محاسبه می‌شود [۱۴].

$$R(t) = \left( \prod_{j=k}^n \lambda_j \right) \times \sum_{i=k}^n \left[ \frac{n!}{i(k-1)!} \left( \prod_{\substack{\theta=k \\ \theta \neq i}}^n \frac{1}{\theta \times \lambda_\theta - i \times \lambda_i} \right) \times \frac{e^{-i \times \lambda_i \times t}}{\lambda_i} \right] \quad (6)$$

در عمل وقتی یک جز در حال کار خراب می شود سایر اجزا با شدت بیشتری کار می کنند در نتیجه نرخ خرابی اجزای در حال کار افزایش می یابد. شریفی و همکاران رابطه ای را به منظور واقعی تر شدن ارتباط بین نرخ خرابی اجزای در حال کار ارائه دادند. این نوع نرخ خرابی را می توان از (۷) محاسبه کرد [۱۴].

$$\lambda_k = \frac{n}{n - \gamma(n - k)} \lambda_n \quad (7)$$

که در آن  $0 \leq \gamma \leq 1$  است بطوریکه اگر  $\gamma = 0$  باشد نرخ خرابی اجزاء مساوی و ثابت و اگر  $\gamma = 1$  برابر  $\lambda_k = \frac{n}{k} \lambda_n$  می شود. به دلیل اینکه اجزای تخصیص یافته به هر زیر سیستم از یک نوع است پس نرخ خرابی این اجزا یکسان می باشد در نتیجه می توان نرخ خرابی اجزای تخصیص یافته را بر حسب نرخ خرابی جزء اول آن به دست آورد. با در نظر گرفتن این فرض و به دلیل اینکه سیستم ما دارای ساختار سری- موازی با زیر سیستم های  $k$ -out-of- $n$  و مقادیر متفاوت  $k$  برای هر زیر سیستم می باشد و انواع مختلفی اجزا برای تخصیص آن در نظر گرفته شده است، (۷) به صورت (۸) نوشته می شود.

$$\lambda_{i_z, k_i} = \frac{(k_i - \gamma(k_i - 1)) \lambda_{i_z, 1}}{k_i} \quad (8)$$

با توجه به (۶) و (۸)، قابلیت اطمینان زیر سیستم هایی با سیاست افزونگی فعال و نرخ خرابی وابسته به تعداد اجزای در حال کار از (۹) به دست می آید.

$$R_{q_i}(t) = \sum_{i=k_i}^{n_i} P_i(t) = \left( \prod_{j=k_i}^{n_i} \frac{(j - \gamma(j - 1)) \lambda_{i_z, 1}}{j} \right) \times$$



$$\sum_{i=k_i}^{n_i} \frac{n_i!}{i(k_i-1)!} \left( \prod_{\substack{\omega=k_i \\ \omega \neq i}}^{n_i} \frac{1}{((\omega - \gamma(\omega - 1)) - (i - \gamma(i - 1)))\lambda_{i z_i 1}} \right) \times \frac{e^{-(i-\gamma(i-1))\lambda_{i z_i 1} t}}{(i - \gamma(i - 1))\lambda_{i z_i 1}} \quad (9)$$

قابلیت اطمینان یک سیستم با در نظر گرفتن سناریوی دوم [۹] و نرخ خرابی ثابت که از سیاست آماده به کار سرد تبعیت می کند از (۱۰) محاسبه می شود.

$$R_{d_2}(t) = \sum_{j=0}^{n_i-k_i-1} ((1-p)p^j \sum_{m=0}^j \left( \frac{e^{-k_i \lambda_{i z_i k_i} t} \cdot (k_i \lambda_{i z_i k_i} t)^m}{m!} \right)) \quad (10)$$

$$+ p^{n_i-k_i} \sum_{m=0}^{n_i-k_i} \frac{e^{-k_i \lambda_{i z_i k_i} t} \cdot (k_i \lambda_{i z_i k_i} t)^m}{m!}$$

### الگوریتم ژنتیک

الگوریتم ژنتیک برای اولین بار در دهه ۷۰ توسط جان هلند [۱۶] مطرح شد که ایده اساسی آن انتقال خصوصیات موروثی توسط ژن هاست این الگوریتم در کلاس الگوریتم های بهینه سازی تصادفی قرار دارد و به طور خاص برای بهینه سازی مسائل پیچیده با فضای جستجوی نا شناخته مناسب است.

### نحوه نمایش کروموزوم

هر جواب موجه این مسئله به صورت یک ماتریس  $3 \times S$  نشان داده میشود که سطرهای هر ماتریس به ترتیب نشان دهنده استراتژی افزونگی، نوع و تعداد اجزای تخصیص یافته در هر زیر سیستم است. شکل (۱) نحوه نمایش یک جواب مسئله با  $S=14$  را نشان می دهد. به عنوان مثال سیاست افزونگی، نوع و تعداد اجزای تخصیص یافته و در زیر سیستم ۲ به ترتیب S، ۱ و ۲ است.

		زیرسیستم													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
استراتژی	}	A	S	S	A	A	S	A	S	A	A	S	S	A	S
تعداد جزء تخصیص یافته		2	1	3	2	3	4	3	1	4	2	3	2	1	2
نوع جزء تخصیص		4	2	3	4	1	1	3	5	3	2	6	4	2	1

شکل (۱): نحوه نمایش کروموزوم

### جمعیت اولیه

در این مقاله از روش ایجاد جمعیت اولیه به صورت تصادفی استفاده شده است.

### تابع برازش

تابع برازش مورد استفاده در این مقاله همان تابع هدف مسئله برای حداکثر کردن قابلیت اطمینانمی باشد. همچنین به منظور جستجو در مناطق غیر موجه فضای حل مسئله از یک تابع جریمه پویا استفاده شده است [۱۷].

### عملگر تقاطع

عملگر تقاطع مورد استفاده در این الگوریتم از نوع یکنواخت می باشد. در این نوع تقاطع یک کروموزوم تصادفی به نام ماسک هم طول با کروموزوم های موجود ایجاد می شود. کروموزوم ماسک تعیین می کند که کدام ژن از والد اول و کدام ژن از والد دوم به فرزند منتقل می - شود [۱۱].

### عملگر جهش

در این مقاله از دو عملگر جهش استفاده شده است:

(۱) جهش عمومی

در این نوع جهش ماتریسی با ابعاد کروموزوم ایجاد و اعداد داخل این ماتریس به صورت تصادفی بین ۰ و ۱ انتخاب می‌شود. عمل جهش بر روی عناصری از این ماتریس صورت می‌گیرد که کمتر از Mutation rate باشند [۱۱].

## ۲) جهش حداکثر- حداقل

در این نوع جهش در هر جواب بیشترین و کمترین قابلیت اطمینان در زیر سیستم‌ها انتخاب شده و عمل جهش با نرخ جهشی کمتر از یک مقدار از پیش تعیین شده انجام می‌شود [۱۱].

## انتخاب

بعد از به کار بردن عملگرهای تقاطع و جهش لازم است از بین جواب‌های نسل قبلی و تولید شده نسل بعدی را انتخاب کنیم که در این الگوریتم از چرخه رولت برای انجام این کار استفاده می‌شود.

## شرط توقف

شرط توقف برای الگوریتم ژنتیک، رسیدن به تعداد تکرار از پیش تعیین شده می‌باشد که در اینجا برابر ۱۰۰ تکرار در نظر گرفته شده است.

## مثال عددی

در این بخش قصد داریم برای بررسی کارایی الگوریتم به کار گرفته شده از مثال‌های عددی متنوع استفاده کنیم. داده‌های مورد استفاده در این مثال‌ها بر گرفته شده از [۵] و پنج مسئله اول ارائه شده در [۱۸] است با این تفاوت که ما در این مثال‌ها پنج زیر سیستم اول مسئله [۵] را با امکان تخصیص دو نوع جزء برای هر زیر سیستم در نظر گرفتیم. در جدول (۱) اطلاعات مربوط به هزینه، وزن، حداقل تعداد اجزای مورد نیاز برای کارکرد هر زیر سیستم و نرخ خرابی هر یک از اجزاء آورده شده است حداکثر تعداد جزء تخصیص یافته به هر زیر سیستم برابر  $n_{max} = 4$ ، و هدف مسئله، حداکثر کردن قابلیت اطمینان کل سیستم است که در آن وزن و هزینه سیستم به عنوان محدودیت در نظر گرفته شده است. برای این مثال‌ها  $C = 47$  است. تنها تفاوت این مثال‌ها در وزن تخصیص یافته به هر سیستم است. به طوریکه در مثال

برگرفته شده از [۵]،  $W = 61$  و در مثال‌های به کار گرفته شده از [۱۸] وزن تخصیص یافته به هر سیستم از ۵۰ تا ۵۴ است. که با توجه به انتخاب ۵ زیر سیستم اول [۵] وزن کل سیستم نیز به همان نسبت کاهش می‌یابد. همچنین قابلیت اطمینان سوئیچ برای تمام مثال‌ها برابر ۰,۹۹ در نظر گرفته شده است [۹] به دلیل این که هزینه‌ای نیز جهت اتصال داخلی بین اجزاء در نظر گرفته شده پارامتری مرتبط با این هزینه ( $\theta_{iz_i}$ ) نیز به این مثال اضافه شده است که به ازای تمام زیر سیستم‌ها و انواع اجزا ۰,۲۵ می‌باشد [۱۵]. همچنین  $\gamma = 0.2$  در نظر گرفته شده است.

جدول (۱): اطلاعات مربوط به مثال‌های عددی [۵]

زیرسیستم		جزء نوع ۱			جزء نوع ۲		
$i$	$k_i$	$\lambda_{iz_i}$	$C_{iz_i}$	$W_{iz_i}$	$\lambda_{iz_i}$	$C_{iz_i}$	$W_{iz_i}$
1	1	0.0010 54	1	3	0.000726	1	4
2	2	0.0005 13	2	8	0.000619	1	10
3	1	0.0016 25	2	7	0.001054	3	5
4	2	0.0018 63	3	5	0.001393	4	6
5	1	0.0006 19	2	4	0.000726	2	3

### تنظیم پارامتر

در این قسمت قصد داریم به تنظیم پارامترهای موثر بر الگوریتم‌های GA پردازیم. برای این کار ما از طراحی آزمایشات استفاده کردیم و با استفاده از روش سطح شناسی پاسخ (RSM) مقادیر بهینه هر یک از پارامترها را به دست آوردیم. پارامترهای موثر بر این الگوریتم‌ها عبارتند از: اندازه جمعیت ( $nPop$ )، نرخ عملگر تقاطع ( $P_c$ )، نرخ جهش عمومی ( $P_{m_1}$ )، نرخ جهش حداکثر - حداقل ( $P_{m_2}$ ) و فاکتور جهش یکنواخت ( $mutation-rate$ ). محدوده جستجوی پارامترهای ورودی و مقادیر بهینه آنها در جدول (۲) آورده شده است.

جدول (۲): محدوده جستجو و مقادیر بهینه پارامترهای GA

پارامتر	حد پایین	حد بالا	مقدار بهینه
$nPop$	300	500	500
$P_c$	0.4	0.7	0.4
$P_{m_1}$	0.1	0.3	0.3
$P_{m_2}$	0.1	0.3	0.1
<i>mutation-rate</i>	0.1	0.3	0.3

### اعتبارسنجی الگوریتم GA

به منظور اعتبارسنجی الگوریتم GA، قصد داریم جواب دقیق مثال‌های عددی را با استفاده از فرآیند شمارشی بدست آورده و با جواب به دست آمده از الگوریتم GA مقایسه کنیم. در فرآیند شمارشی تمام جواب‌های مسئله مورد بررسی قرار می‌گیرد. الگوریتم GA نیز به ازاء مقادیر بهینه پارامترها در ۱۰۰ تکرار اجرا شده است. به منظور کد کردن فرآیند شمارشی و الگوریتم GA مطرح شده در این مقاله از نرم‌افزار Matlab 16 استفاده و بر روی لپ‌تاپی با مشخصات RAM=2G و CPU=2 GH اجرا شده است.

### نتایج

در این بخش به مقایسه نتایج بدست آمده از الگوریتم GA و فرآیند شمارشی می‌پردازیم. در ابتدا به بررسی نتایج به دست آمده از حل مثال عددی [۵] با استفاده از فرآیند شمارشی و الگوریتم GA می‌پردازیم که در جدول (۳) نشان داده شده است. همان‌طور که مشخص است قابلیت اطمینان به دست آمده از الگوریتم GA و فرآیند شمارشی برای این مثال عددی برابر با ۰,۹۹۵۳ می‌باشد، در واقع هر دو روش به جواب دقیق رسیده‌اند. مدت زمانی که به طول انجامید تا فرآیند شمارشی به این

قابلیت اطمینان برسد ۸۴۰۷,۲۲۰ ثانیه است در صورتی که الگوریتم GA در ۸,۵۶۴ ثانیه به جواب رسیده است. شکل (۲) نمودار همگرایی الگوریتم GA را نشان می‌دهد. در ادامه به مقایسه نتایج به دست آمده از حل پنج مسئله اول [۱۸] با استفاده از الگوریتم GA و فرآیند شمارشی می‌پردازیم که نتایج به دست آمده در جدول (۴) نشان داده شده است. با بررسی نتایج مثال‌های ارائه شده می‌توان به سرعت و هوشمندی به بالای الگوریتم GA پی برد.

### نتیجه‌گیری

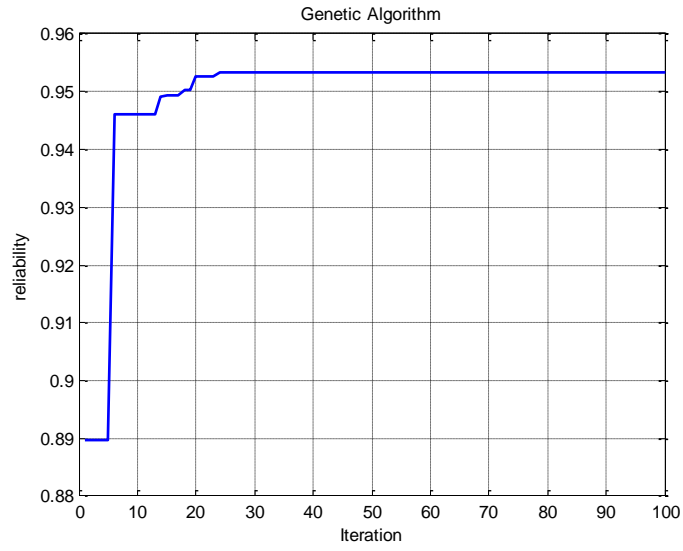
در این مقاله مدلی با در نظر گرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار در سیستمی سری-موازی با زیر سیستم‌های k-Out-of-n ارائه شد که در آن دو سیاست افزونگی فعال و آماده به کار سرد می‌تواند برای هر زیر سیستم انتخاب شود. در نظر گرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار و متغیر در نظر گرفتن سیاست افزونگی موجب واقعی تر شدن شرایط مسئله می‌شود. به دلیل NP-hard بودن مسئله تخصیص افزونگی از الگوریتم فرا ابتکاری GA برای حل مدل و نیز برای تنظیم پارامترهای موثر بر این الگوریتم از روش RSM استفاده شده است. سپس به منظور اعتبار سنجی الگوریتم GA مثال‌های عددی مطرح و جواب‌های به دست آمده فرآیند شمارشی و الگوریتم GA مورد مقایسه و بررسی قرار گرفت که با توجه به نتایج به دست آمده هوشمندی و دقت الگوریتم GA اثبات شد.

جدول (۳): مقایسه جوابهای الگوریتم GA و فرآیند شمارشی مثال عددی برگرفته از [۵]

زیر سیستم	الگوریتم GA			فرآیند شمارشی		
	سیاست افزونگی	تعداد	نوع	سیاست افزونگی	تعداد	نوع
۱	فعال	۲	۱	فعال	۲	۱
۲	فعال	۲	۲	فعال	۲	۲
۳	فعال	۲	۲	فعال	۲	۲
۴	آماده به کار سرد	۳	۲	آماده به کار سرد	۳	۲
۵	فعال	۲	۲	فعال	۲	۲
قابلیت اطمینان	۰,۹۵۳			۰,۹۵۳		
تعداد جوابهای بررسی شده	۵۸۲۱۶			۳۲۰۰۰۰۰		
زمان اجرای الگوریتم (ثانیه)	۸,۵۶۴			۸۴۰۷,۲۲۰		

جدول (۴): مقایسه جوابهای الگوریتم GA و فرآیند شمارشیمثال عددی برگرفته از [۱۸]

مسئله	محدودیت وزن	الگوریتم GA		فرآیند شمارشی	
		قابلیت اطمینان	زمان اجراء	قابلیت اطمینان	زمان اجراء
۱	۵۰	۰,۹۲۸۰	۱۱,۵۸۱۳	۰,۹۲۸۰	۸۳۱۲,۲۲۷
۲	۵۱	۰,۹۲۸۰	۹,۵۲۱۵۷	۰,۹۲۸۰	۸۶۴۵,۹۵۳
۳	۵۲	۰,۹۳۱۴	۹,۶۷۴۶	۰,۹۳۱۴	۸۵۷۳,۳۹۱
۴	۵۳	۰,۹۳۳۹	۹,۴۹۵۹	۰,۹۳۳۹	۸۶۸۰,۴۰۸
۵	۵۴	۰,۹۳۳۹	۸,۷۹۰۵	۰,۹۳۳۹	۸۱۳۵,۴۷۱



شکل (۲): نمودار همگرایی GA مثال برگرفته شده از [۱۵].



## مراجع

Fyffe, D.E., Hines, W.W. and Lee, N.K., "System Reliability Allocation and a Computational Algorithm", IEEE Transactions on Reliability, 1968; Vol. 17, pp. 64-69.

Chern, M. S., "On the Computational Complexity of Reliability Redundancy Allocation in a Series System", Operation Research Letters 1992; Vol. 11, pp. 309-315.

Misra, K.B. and Sharma, U., "Reliability optimization of a system by zero-one programming", Microelectronics and Reliability, 1991 31(2/3), 323-32335.

Coit, D.W. and Smith, A., "Optimization Approaches to the Redundancy Allocation to the Redundancy Allocation Problem for Series-Parallel Systems", Proceedings of the Fourth Industrial Engineering Research Conference, 1995.

Coit D.W. and Liu J. "System reliability optimization with k-out-of-n subsystems", International Journal of Reliability, Quality and Safety Engineering 2000, 7 (2), pp. 129-43.

Hsieh, Y.C. and You, P.S., "An effective immune based two-phase approach for the optimal reliability-redundancy allocation problem", Applied Mathematics and Computation, 2012, 218(4), 1297-1307.

Hsieh, T. J. and Yeh, W. C. "Penalty guided bees search for redundancy allocation problems with a mix of components in series-parallel systems", Computers & Operations Research 2012, 39(11), pp. 2688-2704.

Coit, D.W. "Cold-standby redundancy optimization for non-repairable systems", IIE Transactions 2001, 33(6), pp.471-478.

Coit D.W. "Maximization of system reliability with a choice of redundancy strategies", IIE Transactions 2003, 35(6), pp.535-44.

Tavakkoli-Moghaddam, R. and Safari, J., "A New mathematical model for a redundancy allocation problem with mixing components

redundant and choice of redundancy strategies”. *Applied Mathematical Sciences*, 2007; 45(1), 2221-2230.

Tavakkoli-Moghaddam, R., Safari, J. and Sassani, F., “Reliability Optimization of Series-Parallel Systems with a Choice of Redundancy Strategies Using a Genetic Algorithm”, *Reliability Engineering and System Safety* 2008; Vol. 93, pp. 550–556.

Amari, S.V. and Dill, G. “Redundancy optimization problem with warm-standby redundancy”. *Reliability and Maintainability Symposium (RAMS)*, 2010 Proceedings, 1(6), 25-28.

Amari, S.V., “Reliability of k-out-of-n standby systems with gamma distributions”. *Reliability and Maintainability Symposium (RAMS)*, 2012 Proceedings, 1(6), 23-26.

Sharifi, M., Memariani, A. and Noorossana, R. “Real Time Study of a k-out-of-n System n Identical Elements with Constant Fuzzy Failure Rates”. *World Applied Science Journal*, 2010 8(9),1136-1143.

Wang, Z., Chen, T., Tang, K. and Yao, X. “A multi-objective approach to Redundancy Allocation Problem in parallel-series systems”. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC2009)*, Trondheim, Norway, 582-589.

Holland, J., 1992, “Adaptation Natural and Artificial ststems”, University of Michigan press, An Arbor, MI, (1975), MIT Press, Cambridge, Ma

Coit DW, Smith A., “Penalty guided genetic search for reliabilitydesign optimization”, *Computer and Industrial Engineering* 1996;30(4):895–904.

Nakagawa, Y. and Miyazaki, S., “Surrogate Constraints Algorithm for Reliability Optimization Problems with Two Constraints”, *IEEE Transaction on Reliability* 1981; Vol. 30, pp. 175-180.