

## بهینه سازی قابلیت اطمینان سیستمی با زیرسیستم‌های k-out-of-n با در نظر گرفتن هزینه‌ای برای کاهش نرخ خرابی شده

مانی شریفی\*، کامران دشتی ملجائی\*\*

(تاریخ دریافت: ۹۲/۲/۲۶ - 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-of-n با در نظر گرفتن هزینه‌ای جهت کاهش نرخ خرابی ارائه شده که در آن دو سیاست افزونگی آماده به کار سرد و فعال به عنوان متغیر تصمیم برای هر یک از زیر سیستم ها در نظر گرفته شده است. هدف از حل مدل ارائه شده، تعیین سیاست افزونگی، نوع و تعداد اجزاء مازاد تخصیص یافته و نیز ضریب کاهش نرخ خرابی هر زیر سیستم برای حداکثر کردن قابلیت اطمینان کل سیستم تحت محدودیت‌های هزینه و وزن می باشد. از آنجایی که مسئله تخصیص افزونگی در رسته مسائل NP-hard قرار می گیرد از یک الگوریتم فرا ابتکاری به نام الگوریتم ژنتیک (GA) برای حل مدل استفاده شده و به منظور تنظیم پارامترهای موثر بر این الگوریتم روش سطح پاسخ (RSM) به کار گرفته شده است. در انتها نتایج ارائه شده و مورد تحلیل قرار گرفته است.

واژگان کلیدی: مسئله تخصیص افزونگی، زیرسیستم k-out-of-n، هزینه کاهش نرخ خرابی، سیاست افزونگی، الگوریتم GA

\* استادیار، دانشگاه آزاد اسلامی، واحد قزوین، دانشکده مهندسی صنایع و مکانیک، قزوین، ایران (نویسنده مسئول)

M.Sharifi@Qiau.ac.ir

\*\* دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی، واحد قزوین، دانشکده مهندسی صنایع و مکانیک، قزوین، ایران

kamidashti@yahoo.com

### مقدمه

با پیشرفت روز افزون تکنولوژی و نیاز به داشتن سیستم‌هایی با قابلیت اطمینان بالا، تحقیقات گسترده‌ای در زمینه بهینه‌سازی قابلیت اطمینان صورت گرفته است. مسئله تخصیص افزونگی (Redundancy Allocation Problem (RAP)) یکی از مهمترین مسائلی است که در این ارتباط مطرح شده است. هدف این مسئله حداکثر کردن قابلیت اطمینان با تعیین اجزای مازاد تحت محدودیت‌های موجود می‌باشد. مسئله تخصیص افزونگی برای اولین بار در سال ۱۹۶۸ توسط فیف و همکاران مطرح و با استفاده از برنامه ریزی پویا حل گردید (Fyffe et al., 1968). در ادامه چرن در سال ۱۹۹۲ نشان داد که مسئله تخصیص افزونگی با افزایش تعداد زیر سیستم‌ها در رسته مسائل NP-Hard قرار می‌گیرد (Chern, 1992).

آیدا و همکاران برای اولین بار الگوریتم ژنتیک (Genetic Algorithm (GA)) را برای حل مسائل تخصیص افزونگی با امکان تخصیص اجزای یکسان به هر زیر سیستم در سیستم‌های سری-موازی با چندین حالت خرابی ارائه کردند (Ida et al., 1994). کویت و اسمیت مدل جدیدی برای مسئله تخصیص افزونگی با امکان تخصیص اجزای غیریکسان به هر زیر سیستم ارائه کردند و مدل خود را با استفاده از روش‌های دقیق برنامه‌ریزی پویا، برنامه‌ریزی عدد صحیح، ترکیب برنامه‌ریزی عدد صحیح غیر خطی حل کرده و به مقایسه جواب‌های بدست آمده توسط این روش‌ها و GA پرداختند (Coit and Smith, 1995). کویت و لیو 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 (Coit and Liu, 2000). کویت در سال ۲۰۰۱ مدل جدیدی برای مسئله تخصیص افزونگی ارائه داد که در آن سیستم شناسایی خرابی و کلید زدن به جزء سالم در نظر گرفته شده بود. او همچنین تابع توزیع نرخ خرابی اجزاء را ارلانگ (k-Erlang) فرض کرد و مدل خود را با استفاده از برنامه‌ریزی عدد صحیح حل نمود (Coit, 2001). 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 (Coit, 2003). صفری و توکلی مقدم مدل ارائه شده توسط کویت را در سال ۲۰۰۸ با الگوریتم فرا ابتکاری GA (Tavakkoli-Moghaddam and Safari, 2008) و در سال ۲۰۱۰ با استفاده از الگوریتم فرا ابتکاری ممتیک (Memetic Algorithm (MA)) حل کردند (Safari and Tavakkoli-Moghaddam, 2010). آماری در 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 ارائه داد (Amari, 2012).

ما در این مقاله برای اولین بار هزینه‌ایرا برای کاهش نرخ خرابی اجزاء در یک مسئله تخصیص افزونگی تک هدفه با ساختار سری- موازی و زیر سیستم‌های  $k$ -out-of- $n$  در نظر گرفتیم. در واقع با صرف هزینه‌ای می‌توان نرخ خرابی اجزاء را کاهش داد که این کار موجب افزایش قابلیت اطمینان سیستم می‌شود. کاربرد این موضوع در عمل به عنوان هزینه‌ی نگهداری و تعمیرات پیشگیرانه سیستم مطرح می‌شود. این هزینه شامل دو بخش است. اولین بخش مربوط به هزینه بازدید و سرویس‌های دوره‌ای شامل روان کاری، آچارکشی تجهیزات می‌باشد. دومین بخش عموماً در بردارنده هزینه برنامه‌ریزی نگهداری و تعمیرات، نیروی انسانی و توقف تولید به منظور انجام بازرسی‌های پیشگیرانه است که موجب کاهش توقفات تولید ناشی از خرابی تجهیزات و افزایش عمر و قابلیت اطمینان تجهیزات می‌گردد. این مدل در شرکت‌های تولیدی، نیروگاه‌ها و سایر سیستم‌های که دارای ساختار سری- موازی هستند، قابل استفاده می‌باشد.

در این مقاله هدف از حل مدل ارائه شده، تعیین سیاست افزونگی (فعال یا آماده به کار)، نوع اجزاء، تعداد اجزاء مازاد تخصیص یافته و ضریب کاهش نرخ خرابی هر زیر سیستم جهت حداکثر کردن قابلیت اطمینان کل سیستم تحت محدودیت‌های وزن و هزینه‌هاست. به منظور حل این مسئله و به دلیل NP-Hard بودن آن از الگوریتم فرا ابتکاری GA استفاده می‌شود.

## بیان مسئله

در این مقاله مدلی تک هدفه برای مسئله تخصیص افزونگی با امکان تخصیص اجزای همسان و دو سیاست افزونگی آماده به کار سرد و فعال برای سیستم سری-موازی با زیر سیستم‌های  $k$ -out-of- $n$  در نظر گرفته شده است. در سیاست افزونگی آماده به کار سرد، از یک سخت افزار (سوئیچ) جهت یافتن اجزاء در حال کار استفاده می‌شود تا بلافاصله جزء مازاد فعال شود. به طور کلی دو سناریو برای سوئیچ زدن در سیستم‌هایی با سیاست آماده به کار سرد وجود دارد. در سناریوی یک 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 بیفتد و قابلیت اطمینان سوئیچ به صورت یک تابع غیرافزایشی از زمان در نظر گرفته می‌شود.

در سناریوی دو احتمال خرابی تنها در لحظه سوئیچ زدن وجود دارد که برابر مقدار ثابت است (Coit, 2003). در این مقاله در سیاست آماده به کار سرد از سناریوی دوم برای سوئیچ زدن استفاده می‌شود. در مدل ارائه شده نرخ خرابی اجزاء ثابت است. همچنین به منظور افزایش قابلیت اطمینان کل سیستم هزینه‌ای برای کاهش نرخ خرابی اجزاء در نظر گرفته شده است. برای تعیین این مقدار کاهش پارامتری به نام ضریب کاهش نرخ خرابی تعریف می‌شود که با توجه به این ضریب می‌توان مقدار کاهش نرخ خرابی اجزا را تعیین کرد.

## مفروضات مدل

- ✓ هر زیر سیستم تا زمانی کار می‌کند که حداقل  $k$  جزء در حال کار باشد.
- ✓ تنها یک نوع جزء به هر زیر سیستم می‌توان اختصاص داد.
- ✓ اجزاء دو وضعیت سالم یا خراب را می‌توانند داشته باشند.
- ✓ قابلیت اطمینان سوئیچ در لحظه تقاضا برای هر زیر سیستم در سیاست افزونگی آماده به کار سرد برابر  $P$  می‌باشد.
- ✓ پارامترهای اجزاء نظیر قابلیت اطمینان، هزینه و وزن معلوم و غیر احتمالی هستند.

✓ اجزاء سیستم تعمیر ناپذیر می‌باشند.

### پارامترهای مدل

$S$ : 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،  $i = (1, 2, \dots, S)$

$n$ : مجموعه ای 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،  $z_i \in (1, 2, \dots, m_i)$

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

$t$ : زمان ماموریت

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

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

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

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

$c_{iz_i}^1$ : هزینه کاهش نرخ خرابی هر جزء نوع  $z_i$  ام در زیر سیستم  $i$  ام

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

$\beta_{iz_i}$ : پارامتر مرتبط با ضریب کاهش نرخ خرابی جزء نوع  $z_i$  ام در زیر سیستم  $i$  ام

$a_i$ : حداکثر مقدار پارامتر مرتبط با ضریب کاهش نرخ خرابی 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.

$R_{I_1}$ : قابلیت اطمینان زیر سیستمی که از سیاست افزونگی فعال استفاده می‌کنند.

$R_{I_2}$ : قابلیت اطمینان زیر سیستمی که از سیاست آماده به کار سرد استفاده می‌کنند.

$R_I$ : قابلیت اطمینان کل سیستم

$P$ : قابلیت اطمینان سوئیچ در لحظه سوئیچ زدن

### مدل ریاضی

$$\text{Max } R_I(t) = \prod_{i \in A} R_{I_1}(t) \times \prod_{i \in S} R_{I_2}(t) \quad (1)$$

S.t :

$$\sum_{i=1}^s (c_{iz_i} (n_i + e^{\theta_{iz_i} n_i}) + c'_{iz_i} f(\beta_i)) \leq B \quad (2)$$

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

$$0 \leq \beta_i \leq a_i \quad (4)$$

$$n_i \in (k_i, 2, \dots, n_{\max}); i = (1, 2, \dots, s) \quad (5)$$

$$z_i \in (1, 2, \dots, m_i); i = (1, 2, \dots, s) \quad (6)$$

در این مدل (۱) نشان دهنده قابلیت اطمینان سیستم است که باید حداکثر شود. (۲) نشان دهنده هزینه در دسترس است که در آن هزینه‌ای نیز برای اتصال داخلی بین اجزاء در نظر گرفته شده است (Wang et al., 2009). یکی از راه‌های افزایش قابلیت اطمینان صرف هزینه‌ای برای کاهش نرخ خرابی است که در اینجا تابعی از  $\beta_i$  است. از آنجایی که افزایش قابلیت اطمینان بیش از یک حد مشخص، مقرون به صرفه نیست و نیازمند مصرف هزینه‌ای بالا است، در نتیجه می‌توان این هزینه را به صورت تابعی افزایشی در نظر گرفت. در این مقاله هزینه‌ی کاهش نرخ خرابی را به صورت نمایی که نوعی خاص از تابع افزایشی است در نظر می‌گیریم که از (۷) محاسبه می‌شود.

$$f(\beta_i) = e^{\beta_i \lambda_{iz_i}} \quad (7)$$

(۳) وزن در دسترس و (۴) حد بالا و پایین ضریب کاهش را نشان می‌دهد. (۵) و (۶) نیز به ترتیب حداکثر تعداد اجزاء و نوع در دسترس را برای تخصیص به زیرسیستم‌ها نشان می‌دهند. به دلیل استفاده از دو سیاست آماده به کار سرد و فعال برای زیرسیستم‌ها قابلیت اطمینان به دو قسمت تقسیم می‌شود.  $R_{I_1}$  و  $R_{I_2}$  به ترتیب نشان دهنده قابلیت اطمینان زیرسیستم‌هایی می‌باشد که از سیاست آماده به کار فعال و سرد تبعیت می‌کند. قابلیت اطمینان یک سیستم با در نظر گرفتن نرخ خرابی ثابت که از سیاست آماده به کار فعال تبعیت می‌کند، از (۸) محاسبه می‌شود (Coit and Liu, 2000).

$$R_{I_1}(t) = \sum_{q=k_i}^{n_i} \binom{n_i}{q} (e^{-\lambda_{iz_i} \times t})^q (1 - e^{-\lambda_{iz_i} \times t})^{n_i - q} \quad (8)$$

که در آن:

$$\lambda_{iz_i} = \lambda'_{iz_i} + \beta_{iz_i} \cdot \lambda_{iz_i} \quad (9)$$

قابلیت اطمینان یک سیستم با در نظر گرفتن سناریوی دوم (Coit, 2003) و نرخ خرابی ثابت که از سیاست آماده به کار سرد تبعیت می کند از (۱۰) محاسبه می شود.

$$R_{l_2}(t) = \sum_{j=0}^{n_i-k_i-1} ((1-p)p^j \sum_{m=0}^j \left( \frac{e^{-k_i \lambda_{iz,k_i} t} \cdot (k_i \lambda_{iz,k_i} t)^m}{m!} \right)) + p^{n_i-k_i} \sum_{m=0}^{n_i-k_i} \frac{e^{-k_i \lambda_{iz,k_i} t} \cdot (k_i \lambda_{iz,k_i} t)^m}{m!} \quad (10)$$

### الگوریتم ژنتیک

الگوریتم ژنتیک برای اولین بار در دهه ۷۰ توسط جان هالند مطرح شد که ایده اساسی آن انتقال خصوصیات موروثی توسط ژن ها است که به طور خاص برای بهینه سازی مسائل پیچیده با فضای جستجوی ناشناخته مناسب است. این الگوریتم در کلاس الگوریتم های بهینه سازی تصادفی قرار دارد (Holland, 1975).

### نحوه نمایش کروموزوم

کروموزوم ارائه شده برای این مسئله به صورت یک ماتریس  $4 \times S$  نشان داده می شود که سطرها هر ماتریس به ترتیب نشان دهنده استراتژی افزونگی، نوع و تعداد اجزای تخصیص یافته و ضریب کاهش نرخ خرابی در هر زیر سیستم است. در شکل (۱) نحوه نمایش یک جواب مسئله با  $S = 14$  نشان داده شده است. به عنوان مثال سیاست افزونگی، نوع و تعداد اجزای تخصیص یافته و ضریب کاهش نرخ خرابی در زیر سیستم ۱ به ترتیب A، ۲، ۴ و ۰٫۱۸۶۶ می باشد.

زیر سیستم

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
استراتژی افزونگی	A	S	S	A	A	S	A	S	A	A	S	S	A	S
نوع جزء تخصیص یافته	2	1	3	2	3	4	3	1	4	2	3	2	1	2
تعداد جزء تخصیص یافته	4	2	3	4	1	1	3	5	3	2	6	4	2	1
ضریب کاهش نرخ خرابی	0.1866	0.0134	0.1641	0.0527	0.0730	0.1957	0.1921	0.0381	0.1931	0.1312	0.1923	0.1968	0.1205	0.0790

شکل (۱): نحوه نمایش کروموزوم مدل



### جمعیت اولیه

روش‌های متفاوتی برای ایجاد جمعیت اولیه در الگوریتم‌های مبتنی بر جمعیت وجود دارد که در این مقاله از روش ایجاد جمعیت اولیه به صورت تصادفی استفاده شده است.

### تابع برازش

تابع برازش مورد استفاده در این مقاله از مجموع تابع هدف مسئله با تابع جریمه‌ای پویا برای هر یک از جواب‌های غیر موجه تشکیل شده است که مقدار جریمه آن به درجه تخطی آن جواب از محدوده غیر موجه بستگی دارد. در واقع با این کار شانس را نیز برای جستجوی فضاهای غیر موجه برای پیدا کردن جواب‌های بهتر قائل می‌شویم (Coit and Smith, 1996).

### عملگر تقاطع

در این الگوریتم، از عملگر تقاطع یکنواخت برای تولید فرزندان استفاده می‌شود. در این نوع تقاطع کروموزومی به نام کروموزوم ماسک که هم ابعاد با کروموزوم‌های جواب است تعریف می‌شود که تعیین می‌کند کدام ژن از والد اول و کدام ژن از والد دوم به فرزند منتقل شود (Tavakkoli-Moghaddam and Safari, 2008).

### عملگر جهش

در این الگوریتم از دو عملگر جهش می‌شود که عبارتند از:

#### (۱) جهش عمومی

در این نوع جهش کروموزومی به صورت ماتریسی هم ابعاد کروموزوم جواب ایجاد شده و بر روی عناصری از این ماتریس که مقدار آن کمتر از Mutation rate باشند، عمل جهش انجام می‌گیرد (Tavakkoli-Moghaddam and Safari, 2008).

## ۲) جهش حداکثر-حداقل

در این نوع جهش زیر سیستم‌های با بیشترین و کمترین قابلیت اطمینان برای هر جواب انتخاب شده و عمل جهش با نرخ جهشی کمتر از یک مقدار از پیش تعیین شده انجام می‌شود (Tavakkoli-Moghaddam and Safari, 2008).

## انتخاب

در این الگوریتم پس از به کار بردن عملگرهای تقاطع و جهش، از چرخه رولت برای انتخاب نسل بعدی از بین جواب‌های نسل قبلی و تولید شده استفاده می‌کنیم.

## 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 مدل ارائه شده را با استفاده از مثال‌های عددی متنوع حل می‌کنیم و کارایی الگوریتم را مورد بررسی قرار می‌دهیم. اطلاعات این مثال‌های عددی بر گرفته شده از (Coit and Liu, 2000) و پنج مسئله اول ارائه شده در (Nakagawa and Miyazaki, 1981) است. در این مثال‌ها سیستمی سری-موازی با زیر سیستم‌های  $k$ -out-of- $n$  و امکان تخصیص ۳ یا ۴ نوع جزء به هر زیر سیستم در نظر گرفته شده است. حداکثر تعداد جزء تخصیص یافته به هر زیر سیستم برابر ۶ ( $n_{\max} = 6$ ) و هدف مسئله، حداکثر کردن قابلیت اطمینان کل سیستم تحت محدودیت‌های وزن و هزینه کل سیستم است. هزینه کل سیستم برای این مثال‌ها برابر ۱۷۰ ( $C = 170$ ) در نظر گرفته شده است. تنها تفاوت این مثال‌ها در وزن تخصیص یافته به سیستم می‌باشد بطوریکه در مثال برگرفته شده از (Coit and Liu, 2000)، وزن کل سیستم ۱۷۰ ( $W = 170$ ) و در مثال‌های بکار گرفته شده از (Nakagawa and Miyazaki, 1981) وزن

تخصیص یافته به سیستم از ۱۵۹ تا ۱۶۳ متغیر است. همچنین قابلیت اطمینان سوئیچ برابر ۰,۹۹، در نظر گرفته شده است (Coit, 2001). به دلیل این که هزینه ای برای اتصال داخلی بین اجزاء در نظر گرفته شده پارامتری مرتبط با این هزینه ( $\theta_{i z_i}$ ) به این مثال اضافه شده که به ازای تمام زیر سیستم ها و انواع اجزا ۰,۲۵ است (Wang et al., 2009). همچنین هزینه کاهش نرخ خرابی اجزاء برابر  $0.75C_{i z_i}$  در نظر گرفته شده است. سایر اطلاعات مربوط به مثال‌ها در جدول (۱) آورده شده است.

جدول (۱): اطلاعات مربوط به مثال‌های عددی (Coit and Liu, 2000)

زیرسیستم	جزء نوع ۱			جزء نوع ۲			جزء نوع ۳			جزء نوع ۴				
	$i$	$k_i$	$\lambda_{i z_i}$	$c_{i z_i}$	$w_{i z_i}$	$\lambda_{i z_i}$	$c_{i z_i}$	$w_{i z_i}$	$\lambda_{i z_i}$	$c_{i z_i}$	$w_{i z_i}$	$\lambda_{i z_i}$	$c_{i z_i}$	$w_{i z_i}$
۱	۱	۱	۰,۰۰۱۰۵۴	۱	۳	۰,۰۰۰۷۲۶	۱	۲	۰,۰۰۰۹۳۳	۲	۲	۰,۰۰۰۵۱۳	۲	۵
۲	۲	۲	۰,۰۰۰۵۱۳	۲	۸	۰,۰۰۰۶۱۹	۱	۱۰	۰,۰۰۰۷۲۶	۱	۹	-	-	-
۳	۱	۱	۰,۰۰۰۶۲۵	۲	۷	۰,۰۰۰۱۰۵۲	۳	۵	۰,۰۰۰۱۳۳۳	۱	۶	۰,۰۰۰۸۳۴	۴	۴
۴	۲	۲	۰,۰۰۰۱۸۶۳	۳	۵	۰,۰۰۰۱۳۳۳	۴	۶	۰,۰۰۰۱۶۲۵	۵	۴	-	-	-
۵	۱	۱	۰,۰۰۰۶۱۹	۲	۴	۰,۰۰۰۷۲۶	۲	۳	۰,۰۰۰۵۱۳	۳	۵	-	-	-
۶	۲	۲	۰,۰۰۰۱۰۱	۳	۵	۰,۰۰۰۲۰۲	۳	۴	۰,۰۰۰۳۰۵	۲	۵	۰,۰۰۰۴۰۸	۳	۴
۷	۱	۱	۰,۰۰۰۹۴۳	۴	۷	۰,۰۰۰۸۳۴	۴	۸	۰,۰۰۰۶۱۹	۵	۹	-	-	-
۸	۲	۲	۰,۰۰۰۱۰۷	۳	۴	۰,۰۰۰۱۰۵۴	۵	۷	۰,۰۰۰۹۳۳	۶	۶	-	-	-
۹	۳	۳	۰,۰۰۰۳۰۵	۴	۸	۰,۰۰۰۱۰۱	۳	۹	۰,۰۰۰۴۰۸	۴	۷	۰,۰۰۰۹۳۳	۳	۸
۱۰	۳	۳	۰,۰۰۰۱۸۶۳	۴	۶	۰,۰۰۰۱۶۲۵	۴	۵	۰,۰۰۰۱۰۵۴	۵	۶	-	-	-
۱۱	۳	۳	۰,۰۰۰۶۱۹	۳	۵	۰,۰۰۰۵۱۳	۴	۶	۰,۰۰۰۴۰۸	۵	۶	-	-	-
۱۲	۱	۱	۰,۰۰۰۲۳۵۷	۲	۴	۰,۰۰۰۱۹۸۵	۳	۵	۰,۰۰۰۱۶۲۵	۴	۶	۰,۰۰۰۱۰۵۴	۵	۷
۱۳	۲	۲	۰,۰۰۰۲۰۲	۳	۵	۰,۰۰۰۱۰۱	۳	۵	۰,۰۰۰۳۰۵	۲	۶	-	-	-
۱۴	۳	۳	۰,۰۰۰۱۰۵۴	۴	۶	۰,۰۰۰۸۳۴	۴	۷	۰,۰۰۰۵۱۳	۵	۶	۰,۰۰۰۱۰۱	۶	۹

### تنظیم پارامتر

در این بخش از روش طراحی آزمایشات (Design Of Experiments (DOE)) به منظور تنظیم پارامترهای الگوریتم ارائه شده و تعیین عوامل موثر بر روی آن استفاده می‌شود سپس روش سطح پاسخ (Response Surface Methodology (RSM)) را برای تعیین مقادیر بهینه این پارامترها به کار می‌بریم. پارامترهایی از الگوریتم که تنظیم می‌شوند 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
$nPop$	۳۰۰	۵۰۰	۵۰۰
$P_c$	۰,۴	۰,۷	۰,۷
$P_{m_1}$	۰,۱	۰,۳	۰,۱
$P_{m_2}$	۰,۱	۰,۳	۰,۱۷۸۷۸۸
mutation – rate	۰,۱	۰,۳	۰,۱

### اعتبارسنجی الگوریتم GA

در این بخش قصد داریم به منظور اعتبارسنجی الگوریتم فراابتکاری از روش جستجوی تصادفی ((Random Search (RS) استفاده کنیم. این روش می‌تواند یک حد پایینی (در مسائل بیشینه سازی) برای روش‌های حل دیگر باشد. در واقع اثبات کارکرد هوشمندانه الگوریتم‌های فراابتکاری را می‌توان با مقایسه آن‌ها با یک RS نشان داد. به طوری که این الگوریتم‌ها همواره می‌باید قدرتمندتر از یک RS عمل کنند. پس به منظور صحت‌گذاری عملکرد الگوریتم GA، یک RS ارائه شده است. شبه کد RS پیشنهادی در شکل (۲) نشان داده شده است (Rastrigin, 1963).

```

Initialize  $x$  with a random position in the search space. Until a
termination criterion (number of iteration) is met, repeat the following:
    Sample a new position  $x$  from the hyperactive
    sphere of a given radius surrounding the
    current position  $x$ 
    If  $(f(y) > f(x))$  then move to the new
    Position by setting  $x = y$ 
    Now  $x$  holds the best-found position.
    
```

شکل (۲): شبه کد RS پیشنهادی

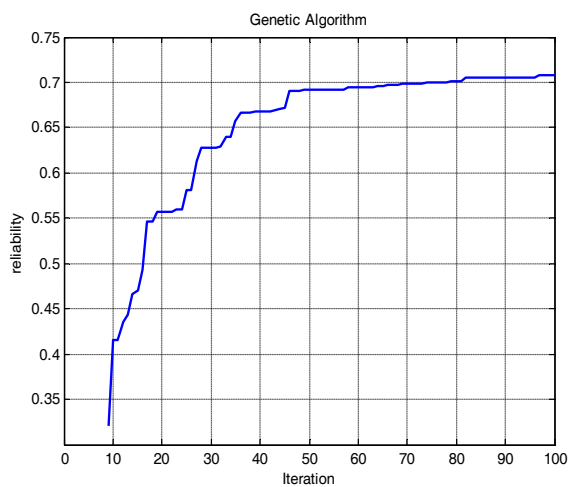
## نتایج

در این بخش به ارائه نتایج بدست آمده از حل مثال‌های عددی بخش قبل می‌پردازیم. در ابتدا به مقایسه نتایج حل مثال عددی ارائه شده در (Coit and Liu, 2000) می‌پردازیم. با توجه به ماهیت تصادفی الگوریتم، این الگوریتم به ازای مقادیر بهینه پارامترها ده بار اجرا و میانگین و بهترین جواب در این اجراها در جدول (۳) نشان داده شده است.

جدول (۳): میانگین و بهترین جواب بدست آمده از GA

الگوریتم	بهترین قابلیت اطمینان	میانگین قابلیت اطمینان
GA	۰,۷۰۶۳	۰,۶۹۳۷

همچنین شکل (۳) نمودار همگرایی بهترین اجرای GA را نشان می‌دهد. به منظور بررسی عملکرد الگوریتم ارائه شده بهترین جواب بدست آمده از GA با جواب بدست آمده از RS مقایسه و نتایج آن در جدول (۴) آورده شده است. با توجه به نتایج جدول (۴) ملاحظه می‌شود الگوریتم GA از لحاظ قابلیت اطمینان کل، تعداد جواب‌های بررسی شده و نیز زمان اجرای الگوریتم نسبت به الگوریتم RS نتایج بهتری را ارائه کرده است. همچنین الگوریتم GA توانست در مدت زمان ۹,۳۲۴۷ ثانیه و بررسی ۱۱۷۴۷ جواب به جواب نهایی الگوریتم RS دست یابد. در ادامه به مقایسه نتایج بدست آمده از حل پنج مسئله اول (Nakagawa and Miyazaki, 1981) با استفاده از الگوریتم‌های GA و RS می‌پردازیم که نتایج آن در جدول (۵) نشان داده شده است. نتایج بدست آمده از حل این مثال‌ها نشان دهنده قدرت و هوشمندی الگوریتم GA ارائه شده می‌باشد.



شکل (۳): نمودار همگرایی بهترین اجرای GA مثال عددی برگرفته از (Coit and Liu, 2000)

جدول (۴): مقایسه جواب های الگوریتم های RS و GA مثال عددی برگرفته از (Coit and Liu, 2000)

زیر سیستم	الگوریتم GA				الگوریتم RS			
	سیاست افزودگی	تعداد	نوع	ضریب کاهش نرخ خرابی	سیاست افزودگی	تعداد	نوع	ضریب کاهش نرخ خرابی
۱	آماده به کارسرد	۲	۱	۰,۱۰۰۶	آماده به کارسرد	۲	۲	۰,۱۰۶۶
۲	فعال	۱	۲	۰,۱۲۱۸	آماده به کارسرد	۱	۱	۰,۱۰۲۱
۳	آماده به کارسرد	۲	۳	۰,۰۸۴۱	فعال	۲	۳	۰,۰۸۸۵
۴	آماده به کارسرد	۳	۱	۰,۱۵۹۳	فعال	۴	۱	۰,۰۹۶۹۱
۵	آماده به کارسرد	۲	۲	۰,۰۹۵۵	فعال	۱	۱	۰,۰۸۸۵
۶	آماده به کارسرد	۳	۴	۰,۰۹۶۲	آماده به کارسرد	۲	۴	۰,۰۵۲۵
۷	آماده به کارسرد	۱	۲	۰,۱۹۲۰	آماده به کارسرد	۱	۳	۰,۱۰۴۹
۸	آماده به کارسرد	۲	۱	۰,۱۰۸۲	آماده به کارسرد	۲	۱	۰,۱۲۵۲
۹	فعال	۳	۱	۰,۱۲۴۰	فعال	۳	۱	۰,۱۰۲۷
۱۰	آماده به کارسرد	۲	۲	۰,۱۷۴۴	آماده به کارسرد	۲	۲	۰,۱۰۰۳
۱۱	آماده به کارسرد	۴	۱	۰,۱۰۸۵	آماده به کارسرد	۳	۱	۰,۰۷۵۳
۱۲	آماده به کارسرد	۲	۱	۰,۱۱۹۷	آماده به کارسرد	۲	۱	۰,۱۰۷۵
۱۳	آماده به کارسرد	۲	۱	۰,۱۰۹۵	فعال	۲	۱	۰,۰۸۹۲
۱۴	آماده به کارسرد	۳	۲	۰,۱۷۱۴	فعال	۲	۳	۰,۱۱۵۲
قابلیت اطمینان	۰,۷۰۸۳				۰,۵۲۹۴			
تعداد جواب های بررسی شده	۴۸۹۱۱				۱۰۰۰۰۰۰			
زمان اجرای الگوریتم (ثانیه)	۳۳,۲۰۸۹				۲۸۳۷,۲۳۷۶			

جدول (۵): مقایسه جواب های الگوریتم های RS و GA مثال عددی برگرفته از (Nakagawa and Miyazaki, 1981)

مسئله	محدودیت وزن	الگوریتم GA		الگوریتم RS	
		قابلیت اطمینان	زمان اجراء	قابلیت اطمینان	زمان اجراء
۱	۱۵۹	۰,۶۶۶۰	۳۳,۲۹۸	۰,۳۹۸۵	۲۷۸۵,۸۸۳
۲	۱۶۰	۰,۶۶۲۹	۳۸,۷۲۳	۰,۴۰۸۵	۲۶۶۹,۶۳۴
۳	۱۶۱	۰,۶۷۰۲	۳۶,۳۲۷	۰,۴۱۷۴	۲۵۲۳,۸۳۲
۴	۱۶۲	۰,۶۷۴۲	۳۵,۳۴۸	۰,۴۵۳۲	۲۹۸۵,۱۲۳
۵	۱۶۳	۰,۶۸۰۱	۳۴,۲۷۹	۰,۴۶۹۱	۲۸۸۵,۶۷۲

### نتیجه گیری

در این مقاله مدلی با در نظر گرفتن هزینه ایبرای کاهش نرخ خرابی و متغیر در نظر گرفتن سیاست افزونگی در سیستمی سری-موازی با زیر سیستم های k-out-of-n ارائه شد. به دلیل NP-hard بودن مسئله تخصیص افزونگی، الگوریتم فرا ابتکاری GA برای حل مدل ارائه شده به کار گرفته شد. همچنین برای تنظیم پارامترهای موثر بر این الگوریتم از تکنیک DOE و از روش RSM برای تعیین مقادیر بهینه آنها استفاده شد. سپس به منظور بررسی عملکرد الگوریتم GA مثال‌های عددی استاندارد حل شد. همچنین به منظور اعتبار سنجی الگوریتم ارائه شده الگوریتم RS به کار گرفته شد و نتایج آن با جواب‌های GA مورد مقایسه قرار گرفت و نشان داده شد که الگوریتم ارائه شده از کارایی مناسبی برخوردار است. در تحقیقات آتی می‌توان فرضیات دیگری مانند تعمیر پذیر بودن اجزاء، فازی در نظر گرفتن پارامترهای اجزاء نظیر قابلیت اطمینان، وزن، هزینه و استفاده از سایر ساختارهای سیستم را به کار گرفت.

## منابع

- Amari, S.V.,(2012)“Reliability of k-out-of-n standby systems with gamma distributions”. Reliability and Maintainability Symposium (RAMS), 2012 Proceedings, 1(6), 23-26.
- Chern, M. S.,(1992) “On the Computational Complexity of Reliability Redundancy Allocation in a Series System”, Operation Research Letters; Vol. 11, pp. 309-315.
- Coit, D.W. and Smith, A.,(1995)“Optimization Approaches to the Redundancy Allocation to the Redundancy Allocation Problem for Series-Parallel Systems”, Proceedings of the Fourth Industrial Engineering Research Conference, pp. 342–349.
- Coit DW, Smith A.,(1996)“Penalty guided genetic search for reliability design optimization”,Computer and Industrial Engineering,;30(4):895–904.
- Coit D.W. and Liu J.(2000) ”System reliability optimization with k-out-of-n subsystems”, International Journal of Reliability, Quality and Safety Engineering, 7 (2), pp. 129–43.
- Coit, D.W. (2001)”Cold-standby redundancy optimization for non-repairable systems”, IIE Transactions, 33(6), pp.471–478.
- Coit D.W.(2003) ”Maximization of system reliability with a choice of redundancy strategies”, IIE Transactions, 35(6), pp.535–44.
- Fyffe, D.E., Hines, W.W. and Lee, N.K.,(1968) “System Reliability Allocation and Computational Algorithm”, IEEE Transactions on Reliability,; Vol. 17, pp. 64-69.
- Holland, J., 1992, “Adaptation Natural and Artificial systems”, University of Michigan press, An Arbor, MI, (1975), MIT Press, Cambridge, Ma
- Ida, K., Gen, M. and Yokota, T.,(1994) “System Reliability Optimization with Several Failure Modes by Genetic Algorithm”, Proceeding of the 16th International Conference on Computers and Industrial Engineering, Ashikaga of Japan, .82.
- Nakagawa, Y. and Miyazaki, S.,(1981)“Surrogate Constraints Algorithm for Reliability Optimization Problems with Two Constraints”, IEEE Transaction on Reliability; Vol. 30, pp. 175-180.
- Rastrigin, L. A. ,(1963)“The convergence of the random search method in the external control of a many parameter system. Automation and Remote Control,; 24, 1337–1342.



Safari J. and Tavakkoli-Moghaddam, R.(2010) "A redundancy allocation problem with the choice of redundancy strategies by a Memetic algorithm", Journal of Industrial Engineering International, 6(11), pp.6–16.

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

Wang, Z., Chen, T., Tang, K. and Yao, X. (2009)"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.





پروپوزیشن گاہ علوم انسانی و مطالعات فرہنگی  
پرتال جامع علوم انسانی