Research Note

مهندسی صنایع و مدیریت شریف، (تابستان ۱۳۹۵)
دوره‌ی ۱ - ۳۲، شماره‌ی ۱/۲، ص. ۱۴۱-۱۴۸، (یادداشت فنی)

بهینه‌سازی همزمان قابلیت اطمینان و هزینه‌ی طراحی در سیستم‌های سری ـ موازی k-out-of-n با در نظرگرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار

مانی شریفی* (استادیار)
کامران دشتی ملجائی (کارشناس ارشد)
قاسم چراغ (کارشناس ارشد)
دانشگاه آزاد اسلامی واحد قزوین، دانشکده‌ی مهندسی صنایع و مکانیک، تهران، ایران

در این نوشتار مدل جدیدی برای مسئله‌ی تخصیص افزونگی با ساختار سری ـ موازی و زیرسیستم‌های k-out-of-n با در نظرگرفتن «نرخ خرابی وابسته به تعداد اجزای در حال کار» ارائه شده که در آن سیاست افزونگی آماده به کار سرد یا فعال برای زیرسیستم‌ها در نظرگرفته می‌شود. هدف، تعیین سیاست افزونگی، نوع و تعداد اجزاء مازاد تخصیص یافته به هر زیرسیستم به‌منظور بیشینه‌سازی قابلیت اطمینان و کمینه‌سازی هزینه‌ی کل سیستم تحت محدودیت وزن است. به‌دلیل NP-hard بودن مسئله از دو الگوریتم فراابتکاری «الگوریتم ژنتیک مرتب‌سازی نامغلوب ۲» (NSGA II) و «الگوریتم ژنتیک رتبه‌بندی نامغلوب» (NRGA) برای حل مدل و از روش سطح پاسخ برای تنظیم پارامترها استفاده شده است. همچنین با استفاده از پنج شاخص عملکردی، عملکرد الگوریتم‌ها مورد بررسی و مقایسه قرارگرفته است. در انتها نیز برای تحلیل نتایج و اثبات درستی روند استفاده از این الگوریتم‌ها از آزمون فرض استفاده شده است.

واژگان کلیدی: مسئله‌ی تخصیص افزونگی، زیرسیستم k-out-of-n، نرخ خرابی وابسته به تعداد اجزای در حال کار، سیاست افزونگی، NSGA II، NRGA.

m.sharifi@qiau.ac.ir
kamidashti@yahoo.com
ghasemcheragh@yahoo.com

۱. مقدمه

با پیشرفت روزافزون تکنولوژی و نیاز به داشتن سیستم‌هایی با قابلیت اطمینان بالا، تحقیقات گسترده‌یی در زمینه‌ی بهینه‌سازی قابلیت اطمینان صورت گرفته است. مسئله‌ی تخصیص افزونگی (RAP)[1] یکی از مهم‌ترین مسائل مطرح‌شده در این زمینه است که هدف آن بیشینه‌سازی قابلیت اطمینان با افزایش اجزای مازاد تحت محدودیت‌های موجود است. این مسئله برای اولین بار در سال ۱۹۶۸ مطرح،[۱] و با استفاده از برنامه‌ریزی پویا حل شد. در سال ۱۹۹۲ محققین نشان دادند که مسئله‌ی تخصیص افزونگی با افزایش تعداد زیرسیستم‌ها در رده‌ی مسائل NP-Hard قرار می‌گیرد.[۲] عوامل زیادی در قابلیت اطمینان یک سیستم تأثیرگذار است که یکی از مهم‌ترین آن‌ها، نرخ خرابی اجزاء است و به دو صورت کلی در مطالعات پیشین مسئله‌ی تخصیص افزونگی (RAP) مطرح شده است.

۱.۱. نرخ خرابی ثابت

نرخ خرابی هریک از اجزاء ثابت و مستقل از یکدیگر است و از تابع توزیع نمایی پیروی می‌کند. در این خصوص، محققین در سال ۱۹۹۱ مدلی برای مسئله‌ی تخصیص افزونگی، با امکان تخصیص اجزای یکسان و سیاست افزونگی فعال به هر زیرسیستم، در نظرگرفتند و آن را با استفاده از برنامه‌ریزی صفر و ۱ حل کردند.[۳] همچنین در سال ۱۹۹۴ برای اولین‌بار الگوریتم ژنتیک (GA)[2] را برای حل مسائل تخصیص افزونگی با در نظرگرفتن چندین حالت خرابی و امکان تخصیص اجزای یکسان به هر زیرسیستم ارائه کردند.[۴] سپس در سال ۲۰۰۰ برای اولین‌بار مدلی با ساختار سری ـ موازی و زیرسیستم‌های k-out-of-n ارائه کردند که در آن به‌طور همزمان دو سیاست افزونگی فعال و آماده به کار سرد به‌صورت از پیش تعیین شده برای هر زیرسیستم در نظرگرفته شده بود؛ این مدل با استفاده از برنامه‌ریزی عدد صحیح حل شد.[۵] در سال ۲۰۱۱ نیز یک روش دومرحله‌یی جدیدِ مبتنی بر مصونیت برای حل مسئله‌ی تخصیص افزونگی تحت محدودیت‌های غیرخطی وزن، هزینه و حجم ارائه شد[۶] و سپس در سال ۲۰۱۲ از یک استراتژی جریمه هدایت‌شده مبتنی بر الگوریتم کلونی زنبور عسل مصنوعی برای حل مسئله‌ی تخصیص افزونگی استفاده شد.[۷]

موارد ذکر شده به‌صورت تک‌هدفه ارائه شده بود. در ادامه به مرور مطالعات انجام‌شده با مشخصه‌ی نرخ خرابی ثابت درخصوص مسئله‌ی تخصیص افزونگی چندهدفه می‌پردازیم. در سال ۲۰۰۱ مدلی با اهداف بیشینه‌سازی قابلیت اطمینان

---

* نویسنده مسئول

تاریخ: دریافت ۱۳۹۳/۲/۷، اصلاحیه ۱۳۹۳/۹/۱۱، پذیرش ۱۳۹۳/۱۰/۷.

سیستم تخمین زده شده و کمینه‌سازی واریانس سیستم تخمین زده شده ارائه شد.[8] سپس در سال ۲۰۰۴ محققین از الگوریتم‌های ژنتیک مرتب‌سازی نامغلوب (NSGA)[3] برای حل مسئله‌ی تخصیص افزونگی چندهدفه استفاده کردند؛ هدف آنان بیشینه‌سازی قابلیت اطمینان سیستم وکمینه‌سازی هزینه، وزن و واریانس قابلیت اطمینان بود به‌طوری که تعداد اجزای تخصیص یافته به هر زیرسیستم نمی‌تواند کم‌تر یا بیشتر از حدود از پیش تعیین شده باشد.[9] پس از آن در سال ۲۰۰۶، سه نوع از مسائل تخصیص افزونگی دوهدفه را به‌کمک الگوریتم فراابتکاری چندهدفه‌ی NSGA II حل کردند. توابع هدف این مسائل بیشینه‌سازی قابلیت اطمینان وکمینه‌سازی هزینه‌ی کل سیستم را شامل می‌شود.[10] در سال ۲۰۰۸، محققین یک الگوریتم تکاملی چندهدفه (MOEA)[4] برای حل مدل تخصیص افزونگی چندهدفه ارائه دادند.[11] تفاوت عمده‌ی الگوریتم پیشنهادی آن‌ها با سایر الگوریتم‌های تکاملی چندهدفه در عملگر تقاطع آن است به‌طوری که تنوع ایجاد جواب به‌وسیله‌ی این عملگر افزایش یافت.[11] در سال ۲۰۱۰ یک الگوریتم چندهدفه‌ی جست‌وجوی متغیر همسایه (VNS)[5] برای حل مسئله‌ی تخصیص افزونگی چندهدفه (MORAP)[6] ارائه، و نتایج حاصل از کاربرد آن برای حل سه مدل مختلف را با نتایج به دست آمده از الگوریتم NSGA II مقایسه شد.[12] در سال ۲۰۱۱، محققین مدلی دوهدفه برای مسئله‌ی تخصیص افزونگی با اهداف بیشینه‌سازی کم‌ترین قابلیت اطمینان زیرسیستم و کمینه‌سازی هزینه‌ی کل سیستم ارائه دادند.[13] آن‌ها جواب‌های پارتو مسئله را از طریق روش ϵ محدودیت برای مثال‌هایی با اندازه کوچک و متوسط به دست آوردند. در سال ۲۰۱۴ نیز مدل جدید ترکیبی برنامه‌ریزی غیرخطی عدد صحیحی برای تحلیل بهینه‌سازی دسترس‌پذیری یک سیستم با ساختار تعیین شده، با در نظرگرفتن اجزاء به دو صورت تعمیرپذیر و تعمیرناپذیر ارائه شد که از الگوریتم ژنتیک برای حل آن استفاده شد.[14]

۱.۲. نرخ خرابی وابسته به زمان

نرخ خرابی اجزاء با افزایش زمان تغییر می‌کند و از یک تابع توزیع مانند ارلنگ، وایبول و غیره پیروی می‌کند. در این خصوص، در سال ۲۰۰۱ مدل جدیدی برای مسئله‌ی تخصیص افزونگی ارائه شد که در آن سیستم شناسایی خرابی و کلید زدن به جزء سالم در نظرگرفته شده بود؛ در این مدل تابع توزیع نرخ خرابی اجزاء ارلنگ[7] فرض شد و مدل مذکور با استفاده از برنامه‌ریزی عدد صحیح حل شد.[15] در سال ۲۰۰۳ با متغیر در نظر گرفتن دو سیاست افزونگی فعال و آماده به کار سرد برای هریک از زیرسیستم‌ها مدل جدیدی توسعه داده شد.[16] این مدل توسط توکلی مقدم و صفری (۲۰۰۸) با استفاده از الگوریتم ژنتیک (GA) حل شد[17] و در ادامه در سال ۲۰۱۰، از الگوریتم ممتیک (MA)[8] برای حل این مدل بهره بردند.[18] در سال ۲۰۱۲ با استفاده از مفاهیم فرایندهای شمارشی، روشی برای ارزیابی قابلیت اطمینان سیستم‌های k-out-of-n با سیاست آماده به کار سرد ارائه شد.[19]

مدل‌هایی که با نرخ خرابی وابسته به زمان مرور شد تک‌هدفه‌اند. اینک در ادامه به مرور مدل‌های چندهدفه با مشخصه‌ی نرخ خرابی وابسته به زمان می‌پردازیم. در سال ۱۹۹۲ ترکیبی از برنامه‌ریزی آرمانی[9] و روش نیل آرمانی[10] برای تولید جواب‌های بهینه‌ی پارتو، در یک سیستم سری چهارمرحله‌یی به‌منظور بیشینه‌سازی قابلیت اطمینان وکمینه‌سازی هزینه، وزن و حجم سیستم مورد استفاده قرارگرفت.[20] سپس در سال ۲۰۱۱ مدل ریاضی جدیدی برای مسئله‌ی تخصیص افزونگی چندهدفه (MORAP) با امکان تخصیص اجزای یکسان و سیاست افزونگی از پیش تعیین شده‌ی فعال و آماده به کار سرد، با اهداف بیشینه‌سازی قابلیت اطمینان و سود خالص ارائه شد.[21] در سال ۲۰۱۲ نیز برای اولین‌بار با متغیر در نظرگرفتن سیاست افزونگی برای یک MORAP مدلی با اهداف بیشینه‌سازی قابلیت اطمینان و کمینه‌سازی هزینه توسعه داده شد[22] و با استفاده از دو روش بهینه‌سازی ازدحام ذرات چندهدفه (MOPSO)[11] و NSGA II حل شد. در سیستمی که تعدادی اجزا با هم کار می‌کنند، با خراب شدن یک جزء سایر اجزای در حال کار باید با شدت بیشتری کار کنند، درنتیجه نرخ خرابی اجزاء افزایش می‌یابد. در نظرگرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار باعث واقعی تر شدن شرایط مسئله می‌شود.[23]

در نوشتار حاضر، برای اولین بار نرخ خرابی ثابت و وابسته به تعداد اجزای در حال کار را برای یک مسئله‌ی تخصیص افزونگی چندهدفه با ساختار سری ـ موازی و زیرسیستم‌های k-out-of-n در نظر گرفته‌ایم. 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 موتورکار می‌کند. وقتی جزئی خراب شود به اجزای دیگر فشار بیشتری وارد می‌شود؛ پس در نظرگرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار باعث واقعی‌تر 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 زیرسیستم برای بیشینه‌سازی قابلیت اطمینان کل سیستم وکمینه‌سازی هزینه‌ی کل سیستم تحت محدودیت وزن است. به‌دلیل NP-Hard بودن مسئله‌ی تخصیص افزونگی ارائه شده از دو الگوریتم ژنتیک‌مرتب‌سازی نامغلوب (NSGA II)، و الگوریتم ژنتیک رتبه‌بندی نامغلوب (NRGA)[12] برای حل آن استفاده می‌کنیم.

۲. فرمول‌بندی مسئله

مدل ریاضی ما برای سیستمی سری ـ موازی با s زیرسیستم k-out-of-n ارائه شده که اهداف بیشینه‌سازی قابلیت اطمینان کل سیستم و کمینه‌سازی هزینه‌ی کل سیستم تحت یک محدودیت خطی را دنبال می‌کند.

۱.۲. مفروضات مسئله

- هر زیرسیستم تا زمانی کار می‌کند که حداقل k جزء در حال کار باشد.
- تنها یک نوع جزء به هر زیرسیستم می‌توان اختصاص داد.
- اجزاء فقط می‌توانند دو وضعیت سالم یا خراب داشته باشند.
- نرخ خرابی اجزاء وابسته به تعداد اجزای در حال کار است.
- پارامترهای اجزاء نظیر قابلیت اطمینان، هزینه و وزن معلوم و غیراحتمالی است.
- اجزاء سیستم تعمیرناپذیرند.

۲.۲. نمادها

s: تعداد کل زیرسیستم؛

i: شاخص زیرسیستم؛

k_i: کم‌ترین تعداد اجزا در حال کار در زیرسیستم iام؛

n_i: تعداد جزء استفاده شده در زیرسیستم iام، $i = (۱, ۲, ..., s)$؛

n: مجموعه‌یی از n_i، $(n_۱, n_۲, ..., n_s)$؛

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

m_i: تعداد نوع در دسترس برای زیرسیستم iام، $i = (۱, ۲, ..., s)$؛

z_i: اندیس نوع جزء استفاده شده برای زیرسیستم iام، $z_i \in (۱, ۲, ..., m_i)$ ؛

z: مجموعه‌یی از z_i، $(z_1, z_2, ..., z_s)$؛

t: زمان ماموریت؛

$\lambda_{iz_ik_i}$: نرخ خرابی جزء نوع z_iام استفاده شده در زیرسیستم i زمانی که k جزء در حال کار است؛

w_{iz_i}: وزن هر جزء نوع z_iام در زیرسیستم iام؛

c_{iz_i}: هزینه‌ی هر جزء نوع z_iام در زیرسیستم iام؛

θ_{iz_i}: پارامتر مرتبط با هزینه‌ی اتصال داخلی جزء نوع z_iام در زیرسیستم iام؛

W: وزن کل سیستم؛

A: مجموعه‌ی زیرسیستم‌هایی که از سیاست افزونگی فعال استفاده می‌کنند؛

S: مجموعه‌ی زیرسیستم‌هایی که از سیاست افزونگی آماده به کار سرد استفاده می‌کنند؛

R_{l_1}: قابلیت اطمینان زیرسیستمی که از سیاست افزونگی فعال استفاده می‌کند؛

R_{l_2}: قابلیت اطمینان زیرسیستمی که از سیاست افزونگی آماده به کار سرد استفاده می‌کند؛

R_l: قابلیت اطمینان کل سیستم؛

P: قابلیت اطمینان سوئیچ در لحظه‌ی سوئیچ زدن.

۲.۳. مدل ریاضی

$$\mathrm{Max}\, R_l(t) = \prod_{i\in A} R_{l_1}(t) \times \prod_{i \in S} R_{l_2}(t) \tag{1}$$

$$\mathrm{Min}\, C = \sum_{i=1}^{s} (c_{iz_i}(n_i + e^{\theta_{iz_i} n_i}) \tag{2}$$

$$\mathrm{S.T:}$$

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

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

$$z_i \in (1, 2, ..., m_i); \quad i = (1, 2, ..., s) \tag{5}$$

در این مدل محدودیت ۱ قابلیت اطمینان سیستم را نشان می‌دهد که باید بیشینه شود. محدودیت ۲ نشانگر هزینه‌ی سیستم است که باید کمینه شود و در آن هزینه‌یی برای اتصال داخلی بین اجزاء در نظر گرفته شده است.[۲۴] محدودیت ۳ وزن در دسترس را نشان می‌دهد. محدودیت ۴ و ۵ نیز به‌ترتیب بیشترین تعداد اجزاء و نوع در دسترس را برای تخصیص به زیرسیستم iام نشان می‌دهند.

R_{l_1} و R_{l_2} به‌ترتیب نشان‌گر قابلیت اطمینان زیرسیستم‌هایی هستند که از سیاست آماده به کار فعال و سرد تبعیت می‌کنند. قابلیت اطمینان یک سیستم با در نظرگرفتن نرخ خرابی ثابت که از سیاست آماده به کار فعال تبعیت می‌کند، از رابطه‌ی ۶ محاسبه می‌شود:[۲۳]

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

در عمل وقتی یک جزء در حال کار خراب می‌شود سایر اجزا با شدت بیشتری کار می‌کنند. درنتیجه نرخ خرابی اجزای در حال کار افزایش می‌یابد. محققین به‌منظور واقعی‌تر شدن ارتباط بین نرخ خرابی اجزای در حال کار رابطه‌یی ارائه دادند. این نوع نرخ خرابی را می‌توان از رابطه‌ی ۷ محاسبه کرد:[۲۳]

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

که درآن $1 \leq \gamma \leq 0$ است به‌طوری که اگر $\gamma = 0$ نرخ خرابی اجزاء مساوی و ثابت است، و اگر $\gamma = 1$ آنگاه $\lambda_k = \frac{n}{k}\lambda_n$ می‌شود. از آنجا که اجزای تخصیص یافته به هر زیرسیستم از یک نوع است، پس نرخ خرابی این اجزا یکسان است. درنتیجه می‌توان نرخ خرابی اجزای تخصیص یافته را برحسب نرخ خرابی جزء اول آن به دست آورد. با در نظر گرفتن این فرض و به‌دلیل این که سیستم ما دارای ساختار سری ـ موازی با زیرسیستم‌های k-out-of-n و مقادیر متفاوت k برای هر زیرسیستم است و انواع مختلف اجزا برای تخصیص آن در نظر گرفته شده، رابطه‌ی ۷ تبدیل می‌شود به:

$$\lambda_{iz_ik_i} = \frac{(k_i - \gamma(k_i - 1))\lambda_{iz_i1}}{k_i} \tag{8}$$

با توجه به رابطه‌های ۶ و ۸، قابلیت اطمینان زیرسیستم‌هایی با سیاست افزونگی فعال و نرخ خرابی وابسته به تعداد اجزای در حال کار از رابطه‌ی ۹ به دست می‌آید. قابلیت اطمینان یک سیستم با درنظرگرفتن سناریوی دوم[۱۶] و نرخ خرابی ثابت که از سیاست آماده به کار سرد تبعیت می‌کند از رابطه‌ی ۱۰ محاسبه می‌شود.

$$\begin{aligned} R_{l_1}(t) = \sum_{i=k_i}^{n_i} P_i(t) = &\left(\prod_{j=k_i}^{n_i} \frac{(j-\gamma(j-1))\lambda_{iz_i1}}{j}\right) \\ &\times \sum_{i=k_i}^{n_i} \Bigg[\frac{n_i\,!}{i(k_i-1)\,!} \\ &\left(\prod_{\substack{\omega=k_i \\ \omega\neq i}}^{n_i} \frac{1}{((\omega-\gamma(\omega-1))-(i-\gamma(i-1)))\lambda_{iz_i1}}\right) \\ &\times \frac{e^{-(i-\gamma(i-1))\lambda_{iz_i1}t}}{(i-\gamma(i-1))\lambda_{iz_i1}}\Bigg] \end{aligned} \tag{9}$$

$$\begin{aligned} R_{l_2}(t) = &\sum_{j=0}^{n_i-k_i-1} ((1-p)p^j \sum_{m=0}^{j} (\frac{e^{-k_i\lambda_{iz_ik_i}t}.(k_i\lambda_{iz_ik_i}t)^m}{m\,!})) \\ &+ p^{n_i-k_i} \sum_{m=0}^{n_i-k_i} \frac{e^{-k_i\lambda_{iz_ik_i}t}.(k_i\lambda_{iz_ik_i}t)^m}{m\,!} \end{aligned} \tag{10}$$

۳. روش‌های حل

چنان که اشاره شد مسئله‌ی تخصیص افزونگی در ابعاد بزرگ در رده‌ی مسائل NP-hard قرار می‌گیرد و درنتیجه، روش‌های دقیق ریاضی کارایی خود را برای حل این مسئله از دست می‌دهند. برای حل این مشکل از الگوریتم‌های فراابتکاری استفاده می‌شود. در این نوشتار ما از دو الگوریتم فراابتکاری NSGA IIو NRGA استفاده می‌کنیم و در نهایت به مقایسه‌ی نتایج این دو الگوریتم می‌پردازیم.

زیر سیستم	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴
استراتژی افزونگی	A	S	S	A	A	S	A	S	A	A	S	S	A	S
نوع جزء تخصیص یافته	۲	۱	۳	۲	۳	۴	۳	۱	۴	۲	۳	۲	۱	۲
تعداد جزء تخصیص یافته	۴	۲	۳	۴	۱	۱	۳	۵	۳	۲	۶	۴	۲	۱

شکل ۱. نحوه‌ی نمایش کروموزوم.

۱.۳. نحوه‌ی نمایش کروموزوم

از آنجا که دو الگوریتم فراابتکاری NSGA II و NRGA، الگوریتم‌هایی مبتنی بر جمعیت‌اند، ما از یک نمایش جواب یکسان برای هر دو آن‌ها استفاده می‌کنیم. هر جواب موجه این مسئله به‌صورت یک ماتریس $3 \times S$ نشان داده می‌شود که سطرهای هر ماتریس به‌ترتیب نشان‌دهنده‌ی استراتژی افزونگی، نوع و تعداد اجزای تخصیص یافته در هر زیرسیستم است. در شکل ۱ نحوه‌ی نمایش یک جواب مسئله با $S = 14$ نشان داده شده است. به‌عنوان مثال سیاست افزونگی، نوع و تعداد اجزای تخصیص یافته‌ی خرابی در زیرسیستم ۱ به‌ترتیب A، ۲ و ۴ است.

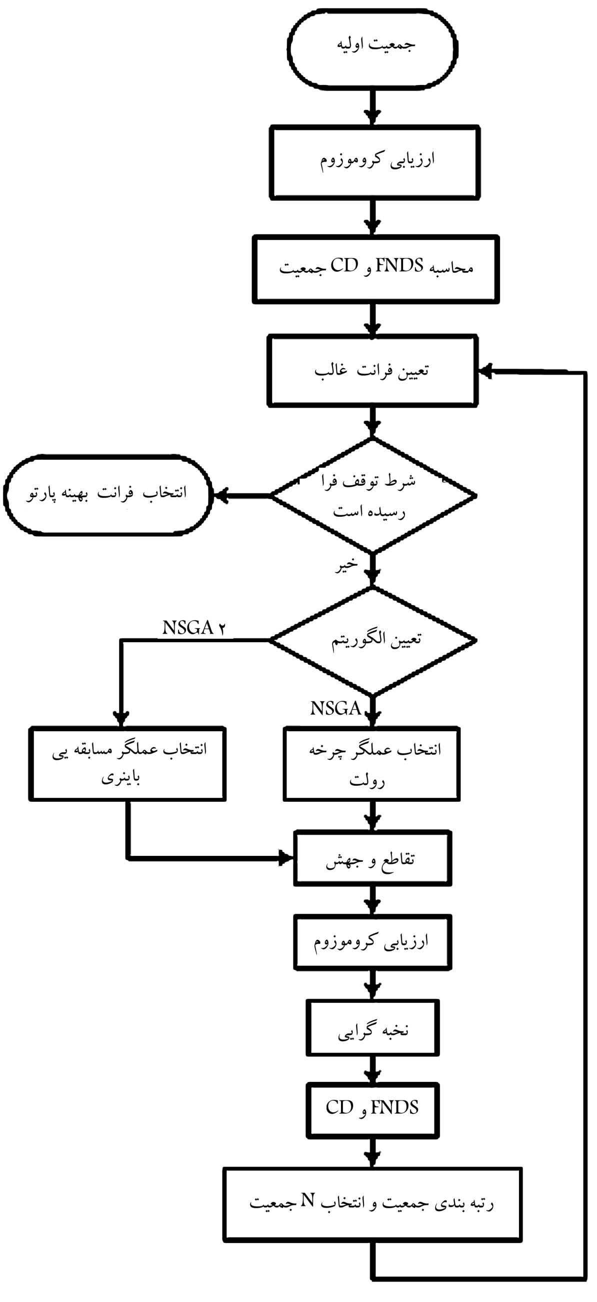
۲.۳. الگوریتم ژنتیک مرتب‌سازی نامغلوب (NSGA)

اسرینیواس و دب (۱۹۹۴) ایده‌ی گلدبرگ را صریحاً درمورد مفاهیم الگوریتم ژنتیک مرتب‌سازی نامغلوب به‌کار بردند.[۲۵] با به‌کارگیری ایده‌ی ارجحیت جواب‌های نامغلوب و اختصاص مقدار برازش بیشتر به آنها، و نیز استفاده از تابع تسهیم طوری که جواب‌های نامغلوب هر دسته به‌طور جداگانه مورد توجه قرار گیرد، الگوریتم NSGA به وجود آمد .روشی کارآمد برای مسائل چندهدفه است، اما به‌دلیل پیچیدگی محاسباتی بالای این الگوریتم و ضعف آن در انتخاب ذره‌های غالب، الگوریتم NSGA II برای رفع این مشکلات مطرح شد.[۲۶] این الگوریتم برای تعیین جواب پارتو از رابطه‌ی غالب و مغلوب، و از عملگرهای جهش مربوط به الگوریتم ژنتیک برای یافتن جواب‌های جدید بهره می‌گیرد و جواب با کیفیت بالا را براساس رابطه‌ی غالب و مغلوب تولید می‌کند.[۲۷]

۳.۳. الگوریتم ژنتیک رتبه‌بندی نامغلوب (NRGA)

یکی از راه‌های افزایش کارایی الگوریتم‌های چندهدفه بهبود عملگرهاست. تمرکز اصلی محققین برای بهبود، بیشتر بر عملگر انتخاب است زیرا بهتر شدن این عملگر باعث همگرایی بیشتر الگوریتم‌های تکاملی چندهدفه می‌شود. از این رو در سال ۲۰۰۸، یک الگوریتم تکاملی چندهدفه‌ی جدید به‌نام الگوریتم ژنتیک رتبه‌بندی نامغلوب (NRGA) با ترکیب الگوریتم انتخاب چرخه‌رولت مبتنی بر رتبه‌بندی[۱۳] و الگوریتم رتبه‌بندی جمعیت براساس پارتو توسعه داده شد.[۲۸] در این ترکیب براساس انتخاب عملگر چرخه‌رولت یک رتبه‌بندی دولایه ارائه می‌شود که به‌طور تصادفی نسل جدید را از نسل والد براساس گزینش بهترین جواب‌ها (براساس برازش و گستردگی) انتخاب می‌کند. NRGA در مقایسه با سایر الگوریتم‌های تکاملی چندهدفه در بیشتر موارد توانایی دست‌یابی به گستره بهتری از جواب‌ها و همچنین هم‌گرایی سریع‌تر به دسته‌ی بهینه پارتو را دارد.

روند کار دو الگوریتم بدین صورت است که جواب‌های موجود درجمعیت براساس میزان نامغلوب بودن‌مرتب می‌شوند؛ به‌طوری که تمام جواب‌های موجود در دسته‌ی اول و آخر به‌ترتیب بهترین و بدترین جواب‌های نامغلوب جمعیت‌اند. درنتیجه به دسته‌ی اول بیشترین برازش، و به دسته‌ی آخرکم‌ترین برازش تعلق می‌گیرد. بعد از رتبه‌بندی دسته‌ها، جواب‌های درون هر دسته نیز براساس فاصله‌ی ازدحامی رتبه‌بندی می‌شود به‌طوری که به جواب با بیشترین و کم‌ترین فاصله‌ی ازدحامی به‌ترتیب بیشترین و کم‌ترین رتبه اختصاص می‌یابد. تفاوت الگوریتم NRGA با NSGA II در بخش استراتژی انتخاب، مرتب‌کردن جمعیت و انتخاب برای نسل بعد است. در الگوریتم NRGA از عملگر چرخه‌ی رولت مبتنی بر رتبه‌بندی به‌جای عملگر مسابقه‌یی ازدحام[۱۴] استفاده می‌شود. در شکل ۲ فلوچارت روند پیاده‌سازی هر دو الگوریتم NRGA و NSGAII نشان داده شده است.



شکل ۲. فلوچارت روند پیاده‌سازی الگوریتم‌های NRGA و NSGAII.

جدول ۱. اطلاعات مثال عددی.

زیرسیستم		جزء نوع ۱			جزء نوع ۲			جزء نوع ۳			جزء نوع ۴		
i	k_i	λ_{iz_i}	c_{iz_i}	w_{iz_i}	λ_{iz_i}	c_{iz_i}	w_{iz_i}	λ_{iz_i}	c_{iz_i}	w_{iz_i}	λ_{iz_i}	c_{iz_i}	w_{iz_i}
۱	۱	۰/۰۰۱۰۵۴	۱	۳	۰/۰۰۰۷۲۶	۱	۴	۰/۰۰۰۹۴۳	۲	۲	۰/۰۰۰۵۱۳	۲	۵
۲	۲	۰/۰۰۰۵۱۳	۲	۸	۰/۰۰۰۶۱۹	۱	۱۰	۰/۰۰۰۷۲۶	۱	۹		—	
۳	۱	۰/۰۰۱۶۲۵	۲	۷	۰/۰۰۱۰۵۴	۳	۵	۰/۰۰۱۳۹۳	۱	۶	۰/۰۰۰۸۳۴	۴	۴
۴	۲	۰/۰۰۱۸۶۳	۳	۵	۰/۰۰۱۳۹۳	۴	۶	۰/۰۰۱۶۲۵	۵	۴		—	
۵	۱	۰/۰۰۰۶۱۹	۲	۴	۰/۰۰۰۷۲۶	۲	۳	۰/۰۰۰۵۱۳	۳	۵		—	
۶	۲	۰/۰۰۰۱۰۱	۳	۵	۰/۰۰۰۲۰۲	۳	۴	۰/۰۰۰۳۰۵	۲	۵	۰/۰۰۰۴۰۸	۲	۴
۷	۱	۰/۰۰۰۹۴۳	۴	۷	۰/۰۰۰۸۳۴	۴	۸	۰/۰۰۰۶۱۹	۵	۹		—	
۸	۲	۰/۰۰۲۱۰۷	۳	۴	۰/۰۰۱۰۵۴	۵	۷	۰/۰۰۰۹۴۳	۶	۶		—	
۹	۳	۰/۰۰۰۳۰۵	۲	۸	۰/۰۰۰۱۰۱	۳	۹	۰/۰۰۰۴۰۸	۴	۷	۰/۰۰۰۹۴۳	۳	۸
۱۰	۳	۰/۰۰۱۸۶۳	۴	۶	۰/۰۰۱۶۲۵	۴	۵	۰/۰۰۱۰۵۴	۵	۶		—	
۱۱	۳	۰/۰۰۰۶۱۹	۳	۵	۰/۰۰۰۵۱۳	۴	۶	۰/۰۰۰۴۰۸	۵	۶		—	
۱۲	۱	۰/۰۰۲۳۵۷	۲	۴	۰/۰۰۱۹۸۵	۳	۵	۰/۰۰۱۶۲۵	۴	۶	۰/۰۰۱۰۵۴	۵	۷
۱۳	۲	۰/۰۰۰۲۰۲	۲	۵	۰/۰۰۰۱۰۱	۳	۵	۰/۰۰۰۳۰۵	۲	۶		—	
۱۴	۳	۰/۰۰۱۰۵۴	۴	۶	۰/۰۰۰۸۳۴	۴	۷	۰/۰۰۰۵۱۳	۵	۶	۰/۰۰۰۱۰۱	۶	۹

۴. مثال عددی

در این بخش قصد داریم برای بررسی کارایی الگوریتم‌های مورد استفاده از یک مثال عددی استفاده کنیم. داده‌های مورد استفاده در این مثال برگرفته از مطالعات انجام شده[5] است. در مثال اصلی، سیستمی سری ـ موازی با زیرسیستم‌های k-out-of-n و امکان تخصیص ۳ یا ۴ نوع جزء به هر زیرسیستم در نظرگرفته شده است. در جدول ۱ اطلاعات مربوط به هزینه، وزن، کم‌ترین تعداد اجزای مورد نیاز برای کارکرد هر زیرسیستم و نرخ خرابی هریک از اجزاء آورده شده است. بیشترین تعداد جزء تخصیص‌یافته به هر زیرسیستم برابر ۶ ($n_{\max} = ۶$) و هدف مسئله، بیشینه‌سازی قابلیت اطمینان و هزینه‌ی کل سیستم تحت محدودیت وزن سیستم $W = ۱۷۰$ است. همچنین قابلیت اطمینان سوئیچ برابر ۰/۹۹ در نظرگرفته شده است.[16] به‌دلیل رعایت هزینه‌ی اتصال داخلی بین اجزاء، پارامتری مرتبط با این هزینه (θ_{iz_i}) نیز به این مثال اضافه شده که به‌ازای تمام زیرسیستم‌ها و انواع اجزا ۰/۲۵ است.[24] همچنین $\gamma = ۰/۲$ در نظرگرفته شده است.

۵. تنظیم پارامتر

در اینجا قصد داریم پارامترهای مؤثر بر الگوریتم‌های NRGA و NSGAII ارائه شده در این نوشتار را تنظیم کنیم. برای این کار ما از طراحی آزمایشات (DOE)[15] استفاده کردیم. پارامترهای مؤثر بر این الگوریتم‌ها عبارت‌اند از:

اندازه جمعیت ($nPop$)، نرخ عملگر تقاطع (P_c)، نرخ جهش عمومی ($P_{m_۱}$)، نرخ جهش max-min ($P_{m_۲}$). محدوده‌ی جست‌وجوی متغیرهای ورودی و مقادیر بهینه‌ی آن‌ها در جدول ۲ ارائه شده است. پس از طراحی آزمایشات به‌منظور تعیین مقادیر بهینه‌ی عوامل تأثیرگذار بر الگوریتم‌های NSGAII و NRGA از روش سطح پاسخ (RSM)[16] استفاده می‌کنیم. RSM مجموعه‌یی از تکنیک‌های آماری و ریاضی است که با طرح آزمایشی مناسب، مدلی مناسب روی داده‌ها برازش داده و با استفاده از آن مقادیر بهینه‌ی چند متغیر را تعیین می‌کند.[29]

جدول ۲. محدوده‌ی جست‌وجوی پارامترهای ورودی NSGAII و NRGA.

الگوریتم	پارامتر	حد پایین	حد بالا	مقدار بهینه
NSGAII	$nPop$	۵۰	۱۰۰	۵۰
	P_c	۰/۳	۰/۶	۰/۳
	$P_{m_۱}$	۰/۱	۰/۳	۰/۱
	$P_{m_۲}$	۰/۱	۰/۳	۰/۱
NRGA	$nPop$	۵۰	۱۰۰	۸۳
	P_c	۰/۳	۰/۶	۰/۳
	$P_{m_۱}$	۰/۱	۰/۳	۰/۱۹۶۹۷۰
	$P_{m_۲}$	۰/۱	۰/۳	۰/۳

۶. اندازه گیری عملکرد

در این مقاله از پنج شاخص عملکرد[17] برای ارزیابی بهتر و دقیق‌تر عملکرد دو الگوریتم ارائه شده استفاده می‌کنیم:

الف) **بیشترین گسترش**[18]. این معیار طول قطر مکعب فضایی راکه توسط مقادیر انتهایی اهداف برای مجموعه جواب‌های نامغلوب به‌کار می‌رود، اندازه‌گیری می‌کند. رابطه‌ی ۱۱ رویه‌ی محاسباتی این شاخص را نشان می‌دهد. بنابراین، هرچه این معیار بزرگ‌تر باشد، نشان‌گر گسترش بیشتر جواب‌های آرشیو پارتو است.[29]

$$D = \sqrt{\sum_{j=۱}^{m} \left(\max_i f_i^j - \min_i f_i^j \right)^۲} \qquad (۱۱)$$

ب) **فاصله‌گذاری**[19]. این معیار میزان فاصله‌ی نسبی جواب‌های متوالی را با استفاده از رابطه‌ی ۱۲ محاسبه می‌کند.

$$S = \sqrt{\frac{۱}{|n|} \sum_{i=۱}^{n} (d_i - \bar{d})^۲} \qquad (۱۲)$$

این معیار انحراف معیارهای مقادیر مختلف d_i را اندازه‌گیری می‌کند. زمانی که جواب‌ها به‌طور یکنواخت درکنار هم باشند آنگاه مقدار S نیز کوچک خواهد بود، بنابراین الگوریتمی که جواب‌های نامغلوب نهایی آن دارای مقدار فاصله‌گذاری کوچکی باشند بهتر خواهد بود.[۲۹]

ج) **تعداد جواب‌های پارتو**[۲۰]. مقدار معیار NPS نشان‌دهنده‌ی تعداد جواب‌های بهینه‌ی پارتو است که در هر الگوریتم می‌توان یافت.[۲۹] هرچه مقدار این معیار بیشتر باشد نشان‌دهنده‌ی عملکرد بهتر الگوریتم است.

د) **فاصله از جواب ایده‌آل**[۲۱]. این معیار که برای اندازه‌گیری میزان نزدیکی به سطح بهینه‌ی پارتوی واقعی به کار می‌رود. هرچه مقدار این معیار کوچک‌تر باشد، مطلوبیت آن مجموعه بیشتر خواهد بود. این معیار از رابطه‌ی ۱۳ محاسبه می‌شود:

$$MID = \frac{\sum_{i=1}^{n} \sqrt{f_{i1}^2 + f_{i2}^2}}{n} \qquad (۱۳)$$

که در آن f_{i1} و f_{i2} به‌ترتیب نشانگر مقادیر اولین و دومین تابع هدف در جواب iام است.[۲۹]

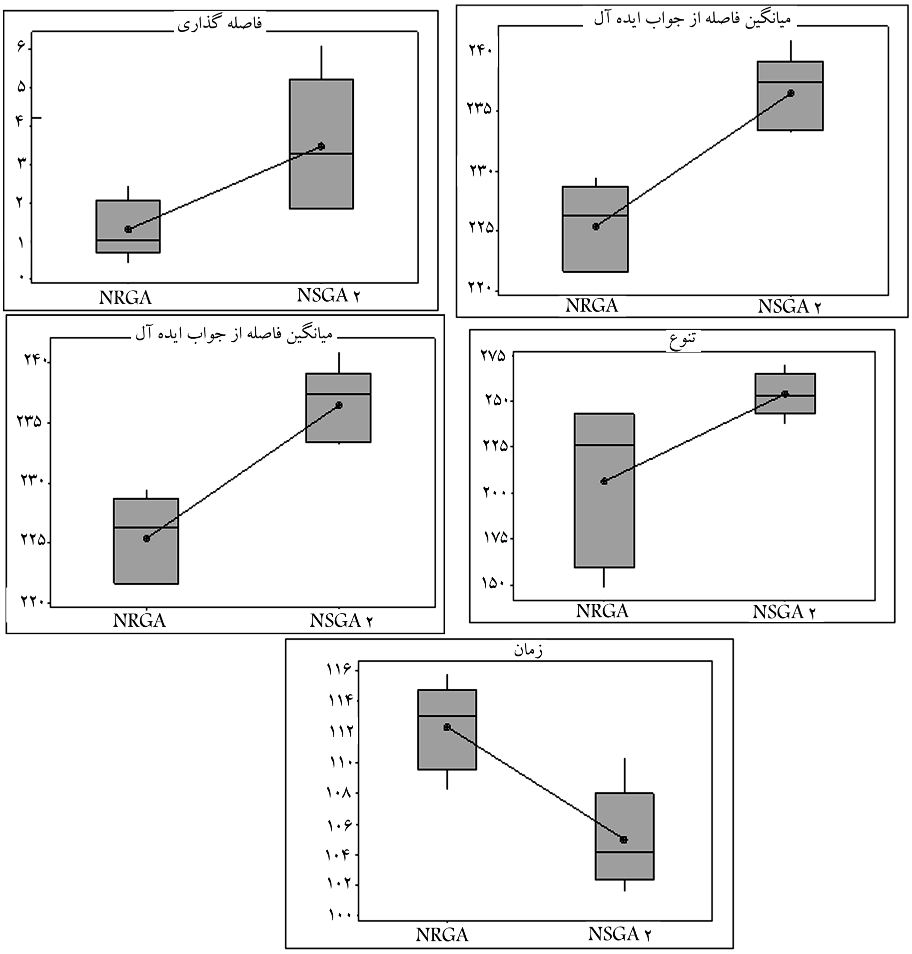
هـ) **زمان اجرای الگوریتم**. زمان اجرای الگوریتم یکی از مهم‌ترین شاخص‌ها در کارایی هر الگوریتم فراابتکاری است.[۲۹]

۷. نتایج

در این بخش نتایج حاصل از الگوریتم‌های مورد استفاده در این مقاله را مقایسه می‌کنیم. در این جا برای کدکردن الگوریتم‌ها از نرم‌افزار Matlab۱۰ و لپ‌تاپی با مشخصات ۶ G = RAM و ۱٫۷۳GH = CPU استفاده شده است. به‌ازای مقادیر بهینه‌ی پارامترهای به دست آمده برای هریک از الگوریتم‌ها و ده بار اجرای آن‌ها، به مقایسه‌ی این الگوریتم‌ها پرداختیم. تعداد تکرار در الگوریتم‌های NSGAII و NRGA برابر ۱۰۰ است. با توجه به شکل ۳، الگوریتم NRGA در شاخص‌های میانگین فاصله از جواب ایده‌آل، تعداد جواب‌های پارتو، فاصله‌گذاری و تنوع و در شاخص زمان NSGAII کارایی بهتری از خود نشان داد.

۱.۷. تحلیل نتایج

برای تشخیص وجود اختلاف معنادار بین نتایج شاخص‌ها، از آزمون فرض در سطح اطمینان ۰٫۹۵ استفاده می‌کنیم. نتایج این آزمون در جدول ۳ ثبت شده است. چون مقدار P-value کم‌تر از مقدار α (سطح ریسک) است، فرض برابری میانگین شاخص‌ها برای هریک از الگوریتم‌ها رد می‌شود. بنابراین اختلاف معناداری بین شاخص‌های الگوریتم‌ها وجود دارد.



شکل ۳. مقایسه شاخص‌های عملکردی دو الگوریتم NRGA و NSGAII.

جدول ۳. مقایسه‌ی مقادیر شاخص‌های عملکردی NSGAII و NRGA.

نتیجه نهایی	نتیجه	P-value	شاخص‌های عملکردی
NRGA	فرض صفر رد شده است	۰٫۰۱۹	میانگین فاصله از جواب ایده آل
NRGA	فرض صفر رد شده است	۰٫۰۰۱	تعداد جواب‌های پارتو
NRGA	فرض صفر رد شده است	۰٫۰۳۳	فاصله‌گذاری
NRGA	فرض صفر رد شده است	۰٫۰۲۷	تنوع
NSGAII	فرض صفر رد شده است	۰٫۰۰۲	زمان

۸. نتیجه‌گیری

در این مطالعه مدلی دوهدفه با در نظر گرفتن نرخ خرابی وابسته به تعداد اجزای در حال کار، در سیستمی سری ـ موازی با زیرسیستم‌های k-out-of-nارائه شد که در آن می‌توان دو سیاست افزونگی فعال و آماده به کار سرد را برای هر زیرسیستم انتخاب کرد. در این نوع نرخ خرابی با افزایش تعداد اجزای در حال کار به‌دلیل تقسیم بر تعداد بیشتری از اجزاء، نرخ خرابی هر یک از اجزاء کاهش می‌یابد. به‌دلیل NP-hard بودن مسئله‌ی تخصیص افزونگی، برای حل آن از الگوریتم‌های فراابتکاری NSGAII و NRGA و نیز برای تنظیم پارامترهای مؤثر بر این الگوریتم‌ها از روش RSM استفاده شده است. سپس به‌منظور مقایسه‌ی عملکرد الگوریتم‌ها از پنج شاخص استفاده شد و از آزمون فرض نیز برای اثبات درستی روند استفاده از این الگوریتم‌ها بهره جستیم. در نهایت الگوریتم NRGA در شاخص‌های میانگین فاصله از جواب ایده‌آل، تعداد جواب‌های پارتو، فاصله‌گذاری و تنوع و نیز در شاخص زمان NSGAII کارایی بهتری از خود نشان داد. در تحقیقات آتی می‌توان فرضیات دیگری مانند تعمیرپذیر بودن اجزاء و استفاده از سایر ساختارهای سیستم را در نظر گرفت.

۸. تقدیر و تشکر

این مقاله برگرفته از طرح پژوهشی تحت حمایت مالی دانشگاه آزاد اسلامی واحد قزوین می‌باشد.

پانوشت‌ها

1. redundancy allocation problem
2. genetic algorithm
3. non-dominated sorting genetic algorithm
4. multi-objective evolutionary algorithms
5. variable neighborhood search
6. multi-objective redundancy allocation problem (MORAP)
7. k-Erlang
8. Mometic algotithm
9. goal programming
10. goal attainment
11. multi-objective particle swarm optimization
12. non-dominated ranked genetic algorithm (NRGA)
13. ranked based roulette wheel selection
14. crowded tournament selection operator
15. design of experiment
16. response surface methodology
17. performance metrics
18. maximum spread or diversity
19. spacing
20. number of pareto solution
21. mean ideal distance

منابع (References)

1. Fyffe, D.E., Hines, W.W. and Lee, N.K. "System reliability allocation and a computational algorithm", *IEEE Transactions on Reliability,* **17**, pp. 64-69 (1968).
2. Chern, M.S. "On the computational complexity of reliability redundancy allocation in a series system", *Operations Research Letters,* **11**, pp. 309-315 (1972).
3. Misra, K.B. and Sharma, U. "Reliability optimization of a system by zero-one programming", *Microelectronics and Reliability*, **31**(2/3), pp. 323-335 (1991).
4. Ida, K., Gen, M. and Yokota, T. "System reliability optimization with several failure modes by genetic algorithm", *in Proceedings of the 16th International Conference on Computers and Industrial Engineering*, pp. 349-352 (1994).
5. Coit, D.W. and Liu, J. "System reliability optimization with k-out-of-n subsystems", *International Journal Reliability Quality Safety Engineering*, **7**(2), pp. 129-43 (2000).
6. 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,* **218**(4), pp. 1297-1307 (2011).
7. 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*, **39**(11), pp. 2688-2704 (2012).
8. Coit, D.W. and Jin, T. "Multi-criteria optimization: Maximization of a system reliability estimate and minimization of the estimate variance", *Proceedings of the 2001 European Safety & Reliability International Conf. (ESREL),* Turin, Italy (2001).
9. Baheranwala, F., Konak, S.K. and Coit, D.W. "Solution of stochastic multi-objective system reliability design problems using genetic algorithm", IEEE Transaction on Reliability (2004).
10. Salazar, D., Rocco, C.M. and Galvan, B.J. "Optimization of constrained multiple-objective reliability problems using evolutionary algorithms", *Reliability Engineering and System Safety*, **91**(9), pp. 1057-70 (2006).

11. Taboada, H.A., Baheranwala, F. and Coit, D.W. "Practical solutions for multi objective optimization: An approach to system reliability design problems", *Reliability Engineering and System Safety*, **92**, pp. 314-322 (2007).
12. Liang, Y.C. and Lo, M.H. "Multi-objective redundancy allocation optimization using a variable neighborhood search algorithm", *Journal of Heuristics,* **16**(3), pp. 511-535 (2010).
13. Soylu, B. and Ulusoy, S.K. "A preference ordered classification for a multi-objective max–min redundancy allocation problem", *Computers & Operations Research*, **38**(12), pp. 1855-1866 (2011).
14. Zoulfaghari, H., Zeinal Hamadani, A. and Abouei Ardakan, M., "Bi-objective redundancy allocation problem for a system with mixed repairable and non-repairable components", *ISA Transactions,* **53**(1), pp. 17-24 (2014).
15. Coit, D.W. "Cold-standby redundancy optimization for non-repairable systems", *IIE Transactions 2001*, **33**(6), pp, 471-478 (2001).
16. Coit, D.W. "Maximization of system reliability with a choice of redundancy strategies", *IIE Transactions*, **35**(6), pp. 535-44 (2003).
17. 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,* **93**(4), pp. 550-556 (2008).
18. Safari, J. and Tavakkoli-Moghaddam, R. "A redundancy allocation problem with the choice of redundancy strategies by a memetic algorithm", *Journal of Industrial Engineering International,* **6**(11), pp. 6-16 (2010).
19. Amari, S.V. "Reliability of k-out-of-n standby systems with gamma distributions", *Reliability and Maintainability Symposium (RAMS), 2012 Proceedings,* **1**(6), pp. 23-26 (2012).
20. Dhingra, A.K. "Optimal apportionment of reliability & redundancy in series systems under multiple objectives", *IEEE Transactions on Reliability,* **41**, pp. 576-582 (1992).
21. EbrahimNezhad, M., Maleki Vishkaei, B., Pasandideh, H.R. and Safari, J. "Increasing the reliability and the profit in a redundancy allocation problem", *International Journal of Applied Operational Research,* **1**(2), pp. 57-64 (2011).
22. Chambari, A.H., Rahmati, S.H., Najafi, A.A. and karimi, A. "A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies", *Computers & Industrial Engineering*, **63**(1), pp. 109-119 (2012).
23. 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,* **8**(9), pp. 1136-1143 (2010).
24. 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, pp. 582-589 (2009).
25. Srinivas, N. and Deb, K. "Multi objective optimization using non dominated sorting in genetic algorithms", *MIT Press Journals,* **2**(3), pp. 221-248 (1994).
26. Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, Chichester, Wiley (2001).
27. Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T. "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II", *In Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI) Conference*, pp. 849-858 (2000).
28. Al Jadaan, O., Rao, C.R. and Rajamani, L. "Non-dominated ranked genetic algorithm for solving multi-objective optimization problems", *Journal of Theoretical and Applied Information Technology*, **2**(1), pp. 60-67. (2008).
29. Montgomery, D.C., *Design and Analysis of Experiments,* 6th ed, New York (USA), John Wiley and Sons (2005).