

بهینه‌سازی پویای پارامترهای الگوریتم هارمونی سرچ در حل مسئله‌ی تخصیص قابلیت اطمینان و اجزای مازاد

محمد رضا ولانی (کارشناس ارشد)

جواد بهنامیان* (استادیار)

گروه مهندسی صنایع، دانشگاه بوعلی سینا

مهندسی صنایع و مدیریت شریف، زمستان ۱۳۹۶ (۱۳-۱)
دوری ۱، شماره ۱/۲، ص. ۳-۱۰

امروزه الگوریتم‌های فراابتکاری نقش بسیار مهمی، در حل مسائل بهینه‌سازی دارند. این الگوریتم‌ها پارامترهای اولیه‌ی دارند که تنظیم بهینه‌ی آنها نقش مؤثری در کیفیت جواب‌های به دست آمده دارد. در بیشتر روش‌های موجود، پارامترهای تنظیم در تمام مراحل، به صورت ثابت در نظر گرفته شده است، اگرچه بهتر است پارامترهای تنظیم با توجه به شرایط مختلف مسئله در طول مراحل بهینه‌سازی تغییرات لازم را داشته باشند. در این مقاله ما روشی را براساس طراحی آزمایشات تاگوچی، برای الگوریتم فراابتکاری هارمونی سرچ پیشنهاد داده‌ایم که پارامترهای اولیه را به صورت پویا تنظیم می‌کند و در بسیاری از الگوریتم‌های فراابتکاری قابل اجراست. کارایی روش تنظیم پارامتر تاگوچی پویا در حل چهار مسئله‌ی بهینه‌سازی تخصیص قابلیت اطمینان - اجزای مازاد بررسی شده که نتایج به دست آمده مؤید استواری این روش نسبت به روش کلاسیک تنظیم پارامتر تاگوچی است.

واژگان کلیدی: الگوریتم‌های فراابتکاری، طراحی آزمایشات، تخصیص قابلیت اطمینان - اجزای مازاد، بهینه‌سازی.

m.valaei91@basu.ac.ir
behnamian@basu.ac.ir

۱. مقدمه

سختی برقراری تعادل بین قابلیت اطمینان و محدودیت‌های منابع، راه حل دشواری خواهد داشت.^[۱]

در طول دو دهه‌ی گذشته بسیاری از روش‌های بهینه‌سازی برای حل مسئله‌ی بهینه‌سازی قابلیت اطمینان - اجزای مازاد پیشنهاد شده است.^[۲] این روش‌ها شامل روش‌های دقیق، روش‌های ابتکاری و فراابتکاری می‌شوند. به دلیل عدم تحذب و ناصافی و چند بعدی بودن مسائل بسیاری از روش‌های کلاسیک ریاضی در به دست آوردن راه حل‌های رضایت بخش شکست خورده‌اند. در سال‌های اخیر روش‌های فراابتکاری در حل مسئله‌ی بهینه‌سازی قابلیت اطمینان پتانسیل بسیار خوبی از خود نشان داده‌اند و توجه زیادی را به خود جلب کرده‌اند.^[۱]

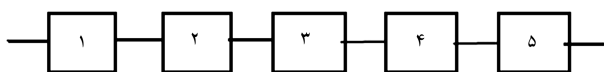
محققین الگوریتم ژنتیک باینری گذشته‌ی برای بهینه‌سازی قابلیت اطمینان پیشنهاد داده‌اند^[۵] و شبیه‌سازی‌هایی بر پایه‌ی سه مسئله‌ی شاخص ارائه داده‌اند که در بین آنها «الگوریتم ژنتیک» در مقایسه با بهترین نتایج قبلی شناخته شده، نتایج بهتری داشته است. همچنین الگوریتم ایمنی مصنوعی برای حل مسائل طراحی قابلیت اطمینان عدد صحیح ترکیبی پیشنهاد شده^[۶] که نتایج رضایت‌بخشی برای چهار سیستم شاخص به دست آورده است. تیلن و گوپال و اگرال^[۷] رویکردهای ابتکاری برای حل مسئله‌ی تخصیص قابلیت اطمینان - اجزای مازاد به کار برده‌اند. همچنین با استفاده از الگوریتم کلونی زنبور عسل^[۹] مسئله‌ی تخصیص قابلیت اطمینان - اجزای مازاد را حل کرده‌اند. در برخی از مطالعات انجام شده NP-hard

بهینه‌سازی قابلیت اطمینان سیستم‌ها در کاربردهای واقعی بسیار مهم است، و در زمینه‌ی آکادمیک و انواع رشته‌های مهندسی توجه زیادی را به خود جلب کرده است.^[۱] دو رویکرد اصلی برای افزایش قابلیت اطمینان سیستم وجود دارد: ۱. افزایش قابلیت اطمینان اجزای سیستم، ۲. استفاده از اجزای مازاد در زیرسیستم‌های مختلف است. «قابلیت اطمینان سیستم» را می‌توان تا درجه خاصی افزایش داد ولی این افزایش برای بسیاری از اجزا قابل توجه نخواهد بود. «استفاده از اجزای مازاد» نیز نیازمند انتخاب بهینه‌ی ترکیب جزء و سطوح اجزای مازاد است. با این عمل قابلیت اطمینان سیستم افزایش می‌یابد و در عین حال بر حجم، هزینه و وزن نیز افزوده می‌شود. در کنار این دو مسیر، ارتباط بین این دو رویکرد و دوباره گمارش اجزای تعویض‌پذیر راه حل شدنی دیگری است که می‌تواند قابلیت اطمینان سیستم را افزایش دهد.^[۲]

مسئله‌ی بهینه‌سازی قابلیت اطمینان - تخصیص اجزای مازاد را می‌توان در بخش برنامه‌ریزی عدد صحیح ترکیبی فرموله کرد که هدف آن دست‌یابی به بالاترین قابلیت اطمینان برای سیستم از طریق انتخاب قابلیت اطمینان اجزا و تخصیص اجزای مازاد با توجه به محدودیت‌های هزینه، وزن و حجم است. این مسئله به دلیل

* نویسنده مسئول

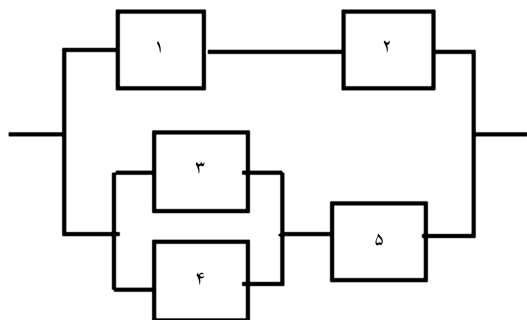
تاریخ دریافت: ۱۳۹۳/۱/۳۰، اصلاحیه ۱۳۹۴/۱/۵، پذیرش ۱۳۹۴/۱/۰۶



شکل ۱. سیستم سری.

جدول ۱. اطلاعات اولیه مسئله‌ی ۱ و ۳.

W	C	V	w_i	$w_i v_i^2$	β_i	$10^5 \alpha_i$	i زیرسیستم
۲۰۰	۱۷۵	۱۱۰	۷	۱	۱٫۵	۲٫۳۳	۱
			۸	۲	۱٫۵	۱٫۴۵	۲
			۸	۳	۱٫۵	۰٫۵۴۱	۳
			۶	۴	۱٫۵	۸٫۰۵	۴
			۹	۲	۱٫۵	۱٫۹۵	۵



شکل ۲. سیستم با ساختار سری - موازی.

جدول ۲. اطلاعات اولیه‌ی مسئله‌ی ۲.

W	C	V	w_i	$w_i v_i^2$	β_i	$10^5 \alpha_i$	i زیرسیستم
۱۰۰	۱۷۵	۱۸۰	۳٫۵	۲	۱٫۵	۲٫۵	۱
			۴	۴	۱٫۵	۱٫۴۵	۲
			۴	۵	۱٫۵	۰٫۵۴۱	۳
			۳٫۵	۸	۱٫۵	۰٫۵۴۱	۴
			۴٫۵	۴	۱٫۵	۲٫۱	۵

$$\text{Maximize } f(r, n) = \prod_{i=1}^m R_i(n_i)$$

subject to :

$$g_1(r, n) = \sum_{i=1}^m w_i v_i n_i^2 \leq V;$$

$$g_2(r, n) = \sum_{i=1}^m \alpha_i (-1000 / \ln r_i)^{\beta_i}$$

$$(n_i + \exp(n_i/4)) \leq C;$$

$$g_3(r, n) = \sum_{i=1}^m w_i n_i \exp(n_i/4) \leq W;$$

$$0 < r_i \leq 1, n_i \in \text{positive integer}, 1 \leq i \leq m; \quad (2)$$

۲.۲. مسئله‌ی ۲. سیستم سری - موازی (شکل ۲)

اطلاعات اولیه و مدل این مسئله در جدول ۲ و رابطه‌ی ۳ موجود است.

$$\text{maximum } f(r, n) = 1 - (1 - R_1 R_2)(1 - (1 - R_2)(1 - R_3) R_5)$$

بودن مسئله‌ی تخصیص اجزاء مازاد ثابت شده است.^[۱۰] از آنجا که فرمول کلی مسئله‌ی تخصیص قابلیت اطمینان - اجزاء مازاد جزء مسائل برنامه‌ریزی عدد صحیح ترکیبی غیرخطی است و آنها سخت‌تر از مسئله‌ی تخصیص اجزاء مازادند، در نتیجه آنها هم جز مسائل NP-hard هستند.^[۱۱] روش‌های بهینه‌سازی کلی همچون برنامه‌ریزی پویا،^[۱۲] رویکرد شاخه و حد،^[۱۳] و شمارش ضمنی^[۱۴] را می‌توان برای حل مسئله‌ی تخصیص قابلیت اطمینان و اجزاء مازاد استفاده کرد، اگرچه این امر، به‌ویژه اگر مسئله بزرگ‌مقیاس باشد، بسیار زمان‌بر است.^[۱۴] محققین رویکردی دومرحله‌یی برای حل مسئله‌ی تخصیص اجزاء مازاد پیشنهاد کرده‌اند^[۱۴] که در مرحله‌ی اول یک الگوریتم مبتنی بر ایمنی بدن را برای به دست آوردن تعداد جزء مازاد توسعه می‌دهد و سپس تخمینی از تخصیص قابلیت اطمینان اجزاء را برای مسئله‌ی تخصیص قابلیت اطمینان - اجزاء مازاد به دست می‌آورد. در مرحله‌ی دوم تعداد اجزاء مازاد را ثابت در نظر می‌گیرد و روشی برای بهبود تخمین تخصیص قابلیت اطمینان اجزاء ارائه می‌دهد.

در این مقاله سعی بر آن است با استفاده از الگوریتم فراابتکاری هارمونی سرچ و تنظیم پویای پارامترهای آن، به بهینه‌سازی مسئله‌ی تخصیص قابلیت اطمینان - اجزاء مازاد پرداخته شود. در بخش ۲ چهار مسئله‌ی بهینه‌سازی ارائه شده، و سپس در بخش ۳ الگوریتم هارمونی سرچ توضیح داده شده است. در بخش ۴ مروری اجمالی بر روش‌های تنظیم پارامتر الگوریتم‌های فراابتکاری شده، و نهایتاً در بخش ۵ روش تنظیم پارامتر تاگوچی پویا را توضیح داده شده است.

۲. مسائل بهینه‌سازی تخصیص قابلیت اطمینان - اجزاء مازاد

مسئله‌ی بهینه‌سازی قابلیت اطمینان - اجزاء مازاد به همراه محدودیت‌های آن به صورت رابطه‌ی ۱ مشخص شده است:

$$\text{Maximize } R_s = f(r, n)$$

$$\text{subject to : } g_j(r, n) \leq l_j ; j = 1, \dots, m \quad (1)$$

R_s : قابلیت اطمینان کل سیستم؛

$r = \{r_1, r_2, r_3, \dots, r_d\}$: قابلیت اطمینان اجزاء که در بازه صفر و ۱ قرار می‌گیرد؛

d : تعداد زیرسیستم‌ها؛

$g_j(r, n)$: j زمین محدودیت؛

$n = \{n_1, n_2, n_3, \dots, n_d\}$: اجزاء مازاد تخصیص یافته به هر زیرسیستم.

هدف یافتن ترکیب بهینه‌ی اجزاء و قابلیت اطمینان آنها تحت محدودیت‌های

موجود، برای دست‌یابی به بیشترین قابلیت اطمینان سیستم است.

تعدادی از مسائل بنچ‌مارک^۲ در موضوع بهینه‌سازی قابلیت اطمینان - اجزاء مازاد که به صورت گسترده‌یی برای آزمون و مقایسه استفاده می‌شود وجود دارد؛^[۱] در این مقاله چهار مسئله‌ی قابلیت اطمینان عدد صحیح ترکیبی به همراه محدودیت‌های غیرخطی ترکیبی در نظر گرفته شده است. سه مسئله‌ی اول شامل سیستم سری، سیستم سری - موازی،^[۱۶,۱۵] سیستم پیچیده (پل)^[۱۸,۱۷,۵,۳] و آخرین مسئله مربوط به سیستم محافظتی سرعت بالای توربین^[۲۰,۱۹] است.

۱.۲. مسئله‌ی ۱. سیستم سری (شکل ۱)

اطلاعات اولیه و مدل این مسئله در جدول ۱ و رابطه‌ی ۲ آمده است.

جدول ۳. اطلاعات اولیه مسئله ۴.

زیرسیستم i	α_i	β_i	$w_i v_i^2$	w_i	V	C	W
۱	۱	۱٫۵	۱	۶	۲۵۰	۴۰۰	۵۰۰
۲	۲٫۳	۱٫۵	۲	۶			
۳	۰٫۳	۱٫۵	۳	۸			
۴	۲٫۳	۱٫۵	۲	۷			

$$h_1(r, n) = \sum_{i=1}^m v_i n_i^i \leq V;$$

$$h_2(r, n) = \sum_{i=1}^m \alpha_i (-1000 / \ln r_i)^{\beta_i}$$

$$(n_i + \exp(n_i/4)) \leq C;$$

$$h_3(r, n) = \sum_{i=1}^m w_i n_i \exp(n_i/4) \leq W;$$

$$0.5 < r_i \leq 1 - 10^{-6}; r_i \in \text{real number};$$

$$1 \leq i \leq m; 1 \leq n_i \leq 10; n_i \in \text{positive integer} \quad (5)$$

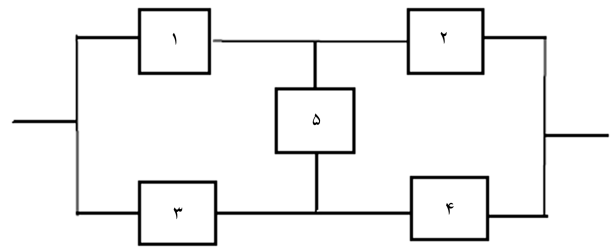
۳. الگوریتم هارمونی سرچ

هارمونی سرچ یک الگوریتم تکاملی مبتنی بر جمعیت است که کاربردهای موفقیت آمیزی در انواع مسائل بهینه‌سازی همچون تابع بهینه‌سازی، طراحی مکانیکی و مهندسی برق داشته است.^[۲۱] از این رو تعداد مقالات تحقیقی درخصوص این روش برای مسائل بهینه‌سازی قابلیت اطمینان وجود دارد. این الگوریتم اولین بار توسط ژانگ وو جم در سال ۲۰۰۱ پیشنهاد شد.^[۲۲] در یکی از مطالعات انجام شده^[۲۳] هارمونی سرچ نو (novel harmony search) و نیز دو اپراتور جدید که برگرفته از هوش جمعیتی توده ذرات است پیشنهاد شده است. همچنین هارمونی سرچ برای بهینه‌سازی قابلیت اطمینان سیستم‌های سری - موازی تحت محدودیت‌های عملکردی و هزینه به کار برده شده است.^[۲۴] الگوریتم هارمونی سرچ برای سیستم سری - موازی باینری و سیستم‌های سری - موازی چندحالتی نیز ارائه شده است.^[۲۵] الگوریتم هارمونی سرچ توانایی خوبی در جست‌وجو دارد و می‌تواند به سرعت ناحیه‌ی دارای بتانسیل را پیدا کند. از این رو باید توانایی تبعیت^۳ این الگوریتم افزایش یابد. برای بهبود این عملکرد، الگوریتم هارمونی سرچ بهبود یافته‌یی که از تنظیم پارامتر پویا استفاده می‌کند پیشنهاد شده است.^[۲۶] ولی هنوز نتایج آن در مقایسه با الگوریتم تکامل^۴ خیلی رضایت‌بخش نیست. همچنین با استفاده از الگوریتم کلونی زنبورهای مصنوعی نسبت به بهینه‌سازی مسئله تخصیص قابلیت اطمینان و اجزاء مازاد اقدام شده است.^[۲۷] در یکی از مطالعات انجام شده از ترکیب هارمونی سرچ و الگوریتم تکامل، الگوریتم تکاملی خاصی ارائه شده که برای حل مسئله بهینه‌سازی قابلیت اطمینان - اجزاء مازاد قابل استفاده است.^[۱] الگوریتم هارمونی سرچ دارای چهار پارامتر تنظیم است:

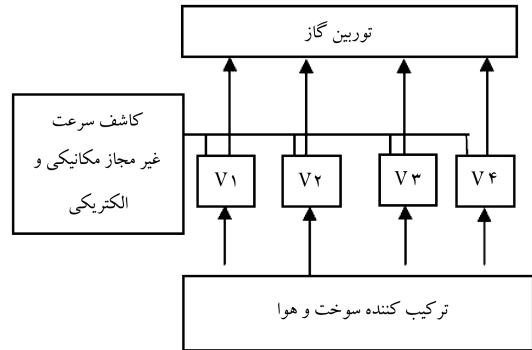
HMS^۵ سایز حافظه‌ی هارمونی

HMCR^۶ نرخ در نظرگیری حافظه‌ی هارمونی

PAR^۷ نرخ تنظیم گام



شکل ۳. سیستم با ساختار پیچیده (پل).



شکل ۴. سیستم محافظتی سرعت بالای توربین.

subject to : $g_1(r, n), g_2(r, n), g_3(r, n);$

$$0 < r_i \leq 1, n_i \in \text{positive integer}, 1 \leq i \leq m; \quad (3)$$

۳.۲. مسئله ۳. سیستم پیچیده یا پل (شکل ۳)

اطلاعات اولیه و مدل این مسئله در جدول ۱ و رابطه ۴ موجود است.

$$\begin{aligned} \text{maximum } f(r, n) = & R_1 R_2 + R_2 R_3 + R_2 R_4 R_5 - R_1 R_2 R_3 R_4 \\ & - R_1 R_2 R_3 R_5 - R_1 R_2 R_4 R_5 - R_1 R_3 R_4 R_5 \\ & - R_2 R_3 R_4 R_5 - 2 R_1 R_2 R_3 R_4 R_5 \end{aligned}$$

subject to : $g_1(r, n), g_2(r, n), g_3(r, n);$

$$0 < r_i \leq 1, n_i \in \text{positive integer}, 1 \leq i \leq m; \quad (4)$$

۴.۲. مسئله ۴. سیستم محافظتی سرعت بالای توربین (شکل ۴)

توربین‌های گازی و توربین‌های بخار نیازمند محافظت از سرعت‌های غیرمجازند. به همین منظور در این قسمت مسئله‌ی محافظت از سرعت غیرمجاز توربین بخار بررسی شده است. یافتن سرعت غیرمجاز توسط سیستم‌های الکترونیکی و مکانیکی به طور پیوسته صورت می‌گیرد و هنگامی که رخ بدهد سیستم نیازمند قطع جریان سوخت است. برای این هدف ۴ دریچه‌ی کنترل‌کننده ($V_1 - V_4$) نصب شده است (شکل ۴). سیستم کنترلی این مسئله با چهار زیرسیستم که به صورت سری با یکدیگر در ارتباطند مدل شده است. اطلاعات اولیه و مدل این مسئله در جدول ۳ و رابطه ۵ موجود است.

$$\text{Maximize } f(r, n) = \prod_{i=1}^m [1 - (1 - r_i)^{n_i}]$$

subject to :

BW^۱ پهنای باند که مقداری دلخواه است.

در جدول ۴ پارامترهای الگوریتم هارمونی سرچ و سطوح پیشنهادی آن ارائه شده است.

۱.۳. تابع جریمه

با توجه به این که مسئله‌ی ما دارای محدودیت است برای هدایت جست‌وجو در طول فضاها جست‌وجو نشده در فضای جواب از تابع جریمه استفاده می‌کنیم.^[۹] تابع جریمه‌ی استفاده شده در این مقاله به صورت زیر است:^[۲۸]

$$\min F(x) = -f(x) + \lambda \sum_{j=1}^{ng} \max(0, g_j)$$

که در آن مقدار ضرب جریمه (λ) مورد استفاده در این مقاله برابر 10^5 است.

۴. تنظیم پارامتر الگوریتم‌های فراابتکاری

بر واضح است که یکی از مهم‌ترین عواملی که می‌تواند عملکرد الگوریتم‌های فراابتکاری را تغییر دهد مقادیری است که برای پارامترهای آن تنظیم می‌شود. از این رو ترکیبات مختلفی از پارامترها نتایج متفاوتی به همراه دارد و می‌کشیم پارامترهای الگوریتم را با مناسب‌ترین مقدار تنظیم کنیم.^[۲۹] بدین منظور با استفاده از روش سطح پاسخ اقدام به تنظیم پارامترهای الگوریتم ژنتیک شده است؛ این تنظیم براساس بهینه‌سازی هم‌زمان دقت جواب^۹ و زمان متناظر با آن در الگوریتم به دست آمده است.^[۲۹] این روش متشکل از روش‌های ریاضی و آماری است که برای مدل‌کردن و تحلیل مسائل مفید خواهد بود و در شرایطی رخ می‌دهد که پاسخ مورد نظر ما از چندین متغیر تأثیر می‌پذیرد و هدف ما بهینه‌سازی این پاسخ است.^[۳۰]

چنان که می‌دانید تکنیک تاگوچی، تکنیکی منحصر به فرد و قدرتمند برای بهبود محصول و فرایند است و کاربرد زیادی در طراحی مهندسی دارد و می‌توان در بسیاری از جنبه‌های دیگر از جمله بهینه‌سازی، طراحی آزمایش، تحلیل حساسیت، تخمین پارامتر و پیش‌بینی مدل و... استفاده کرد. روش تاگوچی ساختارمندتر و تأثیرگذارتر از روش طراحی آزمایشات کلاسیک است که البته به نسبت ساده‌تر هم خواهد بود. روش طراحی آزمایشات کلاسیک گاهی پیچیده و زمان‌بر است و کاربرد آن ساده نخواهد بود. طراحی آزمایشات تاگوچی این اجازه را به ما می‌دهد که اولاً کم‌ترین تعداد آزمایش را انجام داده و ثانیاً آزمایشات را به راحتی تنظیم کنیم. با استفاده از آزمایشات تاگوچی نسبت به تنظیم سطوح پارامترهای اولیه‌ی الگوریتم هارمونی سرچ بهبود یافته اقدام شده است.^[۳۱] در متد تاگوچی از آرایه‌های متعامد برای تحقیق در مورد تعداد زیادی از پارامترها به همراه تعداد کمی آزمایش استفاده می‌شود. تاگوچی فاکتورها را به دو دسته‌ی اصلی کنترل‌پذیر و عوامل اغتشاش

جدول ۴. جدول سطوح مورد آزمایش الگوریتم هارمونی سرچ.

	سطوح مورد آزمایش			
	سطح ۱	سطح ۲	سطح ۳	سطح ۴
HMS	۵	۱۰	۱۵	۲۰
HMCR	۰٫۷	۰٫۹	۰٫۹۵	۰٫۹۹
PAR	۰٫۱	۰٫۳	۰٫۴	۰٫۵
BW	۰٫۱	۰٫۳	۰٫۵	۰٫۷

تقسیم کرده است. عوامل کنترل‌پذیر در آرایه‌ی متعامد درونی و عوامل اغتشاش در آرایه‌های متعامد بیرونی قرار می‌گیرند. مراحل فرایند طراحی آزمایش عبارت است از:

۱. تأثیرات عوامل کنترل‌پذیر تحت نرخ S/N و میانگین پاسخ ارزیابی می‌شود. در حقیقت طراحی آزمایش مناسب تحت نرخ S/N و میانگین مشخصات در نظر گرفته شده اجرا می‌شود.

۲. برای هر عامل که تأثیر قابل توجهی بر نرخ S/N دارد سطحی که نرخ S/N را افزایش می‌دهد انتخاب می‌شود.

۳. هر عاملی که هیچ‌گونه تأثیر قابل توجهی بر نرخ S/N نداشته باشد، اما تأثیر قابل توجهی بر میانگین پاسخ‌های y داشته باشد، به عنوان «عامل تنظیم» در نظر گرفته می‌شود و سطحی که میانگین y نزدیک به نقطه‌ی هدف داشته باشد انتخاب می‌شود.

۴. عواملی که تأثیر قابل توجهی بر نرخ S/N و میانگین y نداشته باشد به عنوان عامل اقتصادی در نظر گرفته می‌شود، و باید سطحی را انتخاب کرد که موجب کاهش هزینه‌ی محصول شود.^[۳۲]

در نوشتار حاضر، انجام آزمایش فاکتوریل کامل برای انتخاب سطوح پارامتر مناسب نیازمند $256 = 4^4$ تعداد آزمایش بدون در نظرگیری تکرار است؛ این تعداد آزمایش با توجه به هزینه و زمان مصرفی به صرفه نخواهد بود. بنابراین ما نیاز به طراحی تکراری کسری داریم. برای انتخاب آرایه‌ی متعامد مناسب باید درجه آزادی را به دست بیاوریم.^[۳۳] در این آزمایش ۳ درجه آزادی برای هر عاملی که در چهار سطح دسته‌بندی شده، و ۱ درجه آزادی برای میانگین کلی در نظر گرفته شده است. در نتیجه آرایه‌ی مناسب دست کم باید تعداد $13 = 1 + (4 - 1) \times 4$ ردیف داشته باشد. برای این عمل از طراحی آرایه‌های متعامد که با نام $L_{16}(4 * * * 4)$ از نرم‌افزار minitab استخراج شده، ۱۶ آزمایش به همراه ۵ تکرار انجام می‌دهیم. پس از انجام آزمایشات، میانگین و نرخ S/N را برای هر ردیف به دست آورده^[۳۳] و بهترین سطوح و بدترین سطوح برای پارامترهای اولیه‌ی الگوریتم را به دست می‌آوریم.

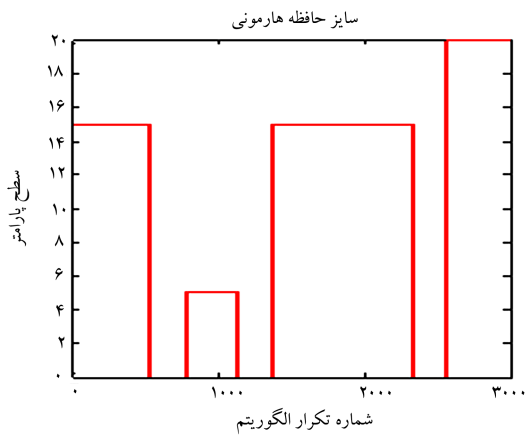
۵. تنظیم پارامتر تاگوچی پویا برای الگوریتم فراابتکاری

هارمونی سرچ

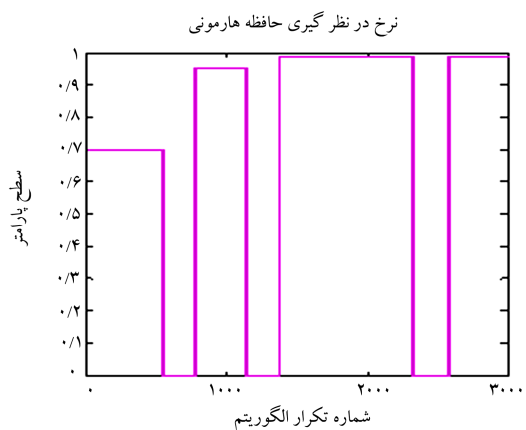
روش تنظیم پارامتر تاگوچی پویا را از دو دیدگاه باید بررسی کرد: کیفیت جواب‌ها و استواری روش.

۱.۵. کیفیت جواب‌های به دست آمده

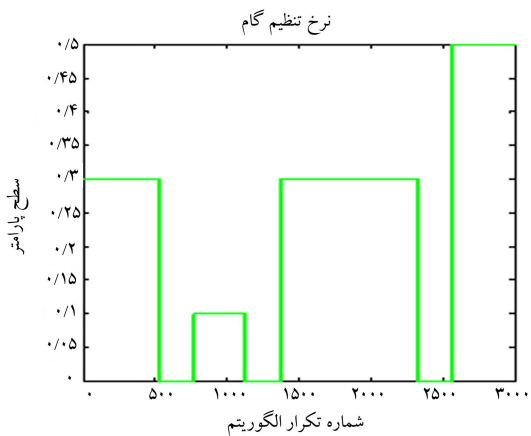
در روش تنظیم پارامتر تاگوچی استاتیک، پارامترها براساس کیفیت جواب نهایی تنظیم می‌شوند، غافل از این که جواب نهایی براساس مقایسه بین کیفیت جواب‌های به دست آمده در طی راه حل‌های جست‌وجو شده‌ی الگوریتم پی‌ریزی شده و زمانی جواب بهینه به دست می‌آید که الگوریتم بتواند از این نقاط عبور کند و نقاطی با کیفیت بهتر را به دست آورد. گاهی الگوریتم در نقاطی از فضای جواب گیر افتاده (که این نقاط می‌توانند نقاط بهینه محلی نیز باشند) و الگوریتم با وجود تعداد جست‌وجوی زیاد، به سستی می‌تواند از آنها عبور کند و جواب بهتری بیابد؛ این نقاط را نقاط گذر سخت نامیده‌ایم. لازم به ذکر است برای خروج الگوریتم از نقاط گذر سخت، می‌توان مسیرهای متنوعی پیدا کرد، که به تبع آن



شکل ۶. سطوح پارامتر تنظیم HMS در طول فرایند جست و جو.



شکل ۷. سطوح پارامتر تنظیم HMCR در طول فرایند جست و جو.

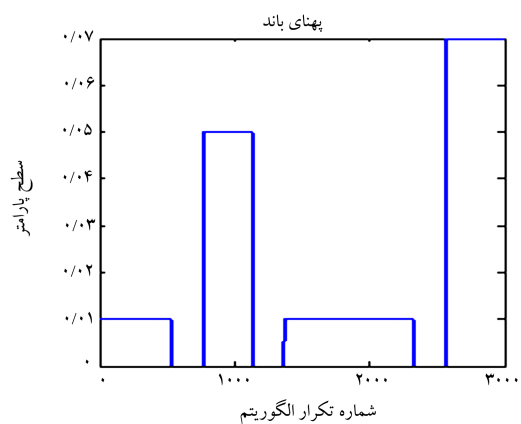


شکل ۸. سطوح پارامتر تنظیم PAR در طول فرایند جست و جو.

و چه در شرایط معمولی، ما آن را در مراحل کلی بهبود مورد استفاده قرار می دهیم. لازم به ذکر است نیازمندی به تعداد کم آزمایش و سادگی کاربرد این روش، دلیل اصلی استفاده از روش طراحی آزمایش تاگوچی برای تنظیم پارامترهای الگوریتم فراابتکاری است. نکته‌ی مهمی که از جمله ویژگی‌های روش پیشنهادی است، توانایی کاربرد این روش در بسیاری از الگوریتم‌های فراابتکاری است که می‌تواند موجب بهبود عملکرد آنها شود.

کیفیت جواب‌های متنوعی هم به دست می‌آید. هدف اصلی این مقاله استفاده از آزمایش‌های تاگوچی (تحت سطوح مختلف پارامترهای تنظیم) برای انتخاب بهترین شرایطی است که الگوریتم می‌تواند از نقاط مذکور عبور کند. لازم به ذکر است که این عمل را می‌توان با روش جست‌وجوی گرادیان^{۱۰} مقایسه کرد. در ادامه، الگوریتم با سطوح جدید پارامترهای خود، مراحل عادی را طی کرده و زمانی که الگوریتم دوباره در نقاط گذر سخت‌گیر بیفتد، الگوریتم مراحل بالا را تکرار می‌کند.

از آنجا که برای ورود به مرحله‌ی تغییر سطوح پارامتر نیازمند یک معیار خاص هستیم، معیار تعداد تکراری که الگوریتم جواب بهتری ایجاد نمی‌کند را برای ورود به مرحله‌ی تغییر سطوح پارامتر الگوریتم انتخاب می‌کنیم و این معیار را NIWI^{۱۱} نامیده‌ایم. لازم به ذکر است پس از تغییر سطوح پارامتر الگوریتم معیار NIWI صفر خواهد شد. نکته‌ی مهمی که در اینجا مطرح است تعداد تکرارهای لازم برای الگوریتم در نقاط گذر سخت است. این موضوع به شدت به دو عامل بستگی دارد: نخستین عامل «سطوح حافظه‌ی هارمونی» است، زیرا اختلاف حداکثر و حداقل سطوح حافظه‌ی هارمونی در نظر گرفته شده در ابتدای کار تأثیر زیادی بر تعداد تکرارهای الگوریتم در نقاط گذر سخت دارد. این موضوع را با این مثال می‌توان بیان کرد که اگر الگوریتم با اندازه حافظه‌ی هارمونی ۵ وارد نقطه‌ی گذر سخت شود و آزمایشی که برای یکی از حالات مختلف سطوح در نظر گرفته شده دارای حافظه‌ی هارمونی ۲۰ باشد، برای این که حافظه‌ی هارمونی را پرکنیم نیازمند به ۱۵ تکرار در آن سطح آزمایش هستیم. دومین عامل تعداد آزمایش مورد نیاز برای تنظیم سطوح پارامتر الگوریتم است. در نوشتار حاضر با توجه به اختلاف حداقل و حداکثری سطوح حافظه‌ی هارمونی و تعداد ۱۶ سطح آزمایش، هنگامی که الگوریتم وارد نقاط گذر سخت می‌شود نیازمند $240 = 16 \times 15$ تکرار الگوریتم هستیم. نمودارهای نشان داده شده در شکل‌های ۵ تا ۸ نشان‌دهنده‌ی نمونه‌یی از تغییر پارامتر پویای الگوریتم هارمونی سرچ هستند که در ۳۰۰۰ تکرار الگوریتم و شرط ۳۰۰ تکرار بدون بهبود الگوریتم برای ورود به تنظیم پارامتر تاگوچی پویا (در مسئله‌ی شماره ۴) به دست آمده است. در این اشکال زمانی که الگوریتم وارد تنظیم پارامتر تاگوچی پویا می‌شود پارامترها را به صورت صفر نمایش می‌دهیم و پس از انجام آزمایشات، دوباره سطوح جدید نمایان می‌شود. لازم به ذکر است که برای نشان دادن فرایند مرحله به مرحله‌ی الگوریتم از فلوجارت نیز استفاده شده است (شکل ۹). باید توجه داشت که در این مقاله تعداد تکراری که در نقاط گذر سخت اجرا می‌شود جزو تعداد کل تکرارهای الگوریتم به حساب می‌آید و هرگاه بهبودی در الگوریتم به وجود می‌آید، چه در شرایط آزمایش

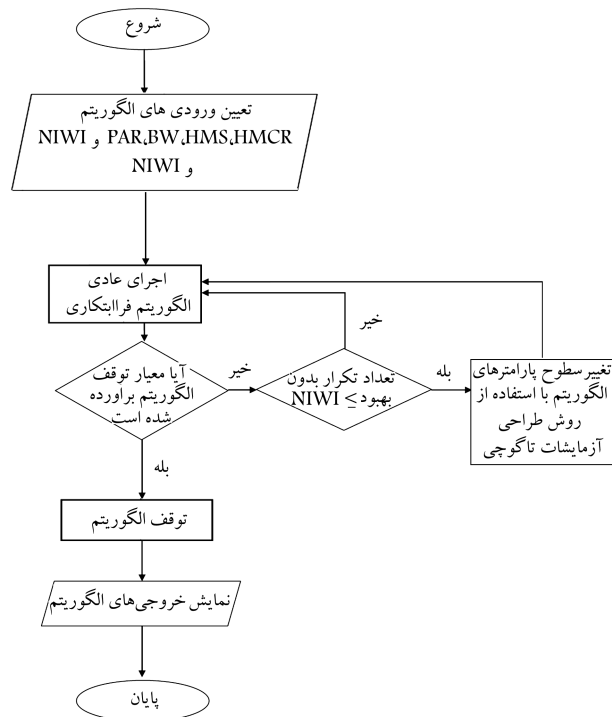


شکل ۵. سطوح پارامتر تنظیم BW در طول فرایند جست و جو.

جدول ۵. مقایسه دو روش تنظیم پارامتر الگوریتم هارمونی سرچ.

	تنظیم پارامتر		
	پویا	کلاسیک	
سری - موازی	mean	۰٫۹۹۹۹۰۹۳۴	۰٫۹۹۹۹۴۵۹۹
	max	۰٫۹۹۹۹۴۵۹۲	۰٫۹۹۹۹۷۵۸۸
	min	۰٫۹۹۹۷۲۸۹۲	۰٫۹۹۹۹۰۶۱۳
	SD	۰٫۰۰۰۰۶۵۲	۰٫۰۰۰۰۲۲۲۲
سری - موازی	mean	۰٫۹۹۶۲۵۵۶۳	۰٫۹۹۸۶۷۴۹۵
	max	۰٫۹۹۹۲۲۸۶۵	۰٫۹۹۹۹۵۰۴۶
	min	۰٫۹۸۸۰۴۰۱۷	۰٫۹۹۴۳۱۲۱۱
	SD	۰٫۰۰۳۴۸۳۰۵	۰٫۰۰۱۷۲۲۸۴
پل	mean	۰٫۹۹۹۷۴۳۷۱	۰٫۹۹۹۸۰۷۸۳
	max	۰٫۹۹۹۸۱۷۵۸	۰٫۹۹۹۸۸۸۹۱
	min	۰٫۹۹۹۵۳۵۹۱	۰٫۹۹۹۶۴۵۴۳
	SD	۰٫۰۰۰۰۹۴۷۶	۰٫۰۰۰۰۰۸۲۱۲
پل	mean	۰٫۹۹۹۳۵۰۹۱	۰٫۹۹۹۳۸۳۵۲
	max	۰٫۹۹۹۷۱۵۵۲	۰٫۹۹۹۷۱۷۲۲
	min	۰٫۹۹۸۳۷۰۶۴	۰٫۹۹۹۰۸۲۷۸
	SD	۰٫۰۰۰۰۴۴۴۸۰	۰٫۰۰۰۰۲۴۹۶۴
سیستم حافظه سرعت بالا	mean	۰٫۹۹۹۹۳۴۹۵	۰٫۹۹۹۹۴۳۸۲
	max	۰٫۹۹۹۹۵۳۸۵	۰٫۹۹۹۹۵۴۳۱
	min	۰٫۹۹۹۸۸۵۱۴	۰٫۹۹۹۹۱۷۸۷
	SD	۰٫۰۰۰۰۰۲۲	۰٫۰۰۰۰۰۱۰۷۱
سری	mean	۰٫۹۹۷۷۸۴۳۲	۰٫۹۹۹۶۳۵۷۸
	max	۰٫۹۹۹۹۲۷۴۴	۰٫۹۹۹۹۲۹۲۰
	min	۰٫۹۸۸۳۰۵۹۴	۰٫۰۹۷۶۹۵۰۰
	SD	۰٫۰۰۳۴۷۱۰۴	۰٫۰۰۰۶۳۲۲۳
سری	mean	۰٫۹۹۹۳۹۸۶	۰٫۹۲۱۴۵۳۰۲
	max	۰٫۹۳۱۱۷۷۶۶	۰٫۹۳۱۵۹۸۶۰
	min	۰٫۸۷۸۲۷۲۶۷	۰٫۸۹۱۴۳۶۷۶۳
	SD	۰٫۰۱۵۸۱۰۴۱	۰٫۰۱۲۱۳۷۹۳۸
سری	mean	۰٫۸۸۹۱۶۶۳۷	۰٫۸۸۹۷۲۳۹۰
	max	۰٫۹۳۱۶۵۲۹۲	۰٫۹۲۳۲۴۶۵۶
	min	۰٫۸۲۷۶۰۲۹۴	۰٫۸۲۹۴۹۷۰۹
	SD	۰٫۰۴۵۳۰۳۵۲	۰٫۰۳۵۴۸۹۱۱

به غیر از یک مورد دارای بیشینه جواب بهتر نیز هست. براین اساس می‌توان نتیجه گرفت که تنظیم پارامتر تاگوچی پویا عملکرد بهتری نسبت به تنظیم پارامتر تاگوچی کلاسیک دارد. هدف این مطالعه تأیید عملکرد بهتر تنظیم پارامتر تاگوچی پویا نسبت به تنظیم پارامتر تاگوچی کلاسیک برای الگوریتم هارمونی سرچ، برای بهینه‌سازی مسئله‌ی تخصیص قابلیت اطمینان - اجزاء مازاد است. به‌منظور اطمینان از کارایی کلی این روش، مقایسه‌ی بهترین جواب به دست آمده با بهترین جواب تعدادی از مقالات دیگر در جدول ۶ ارائه شده است.



شکل ۹. فلوچارت الگوریتم هارمونی سرچ با تنظیم پارامتر پویا.

۲.۵. استواری روش

چنان‌که می‌دانید الگوریتم‌های فراابتکاری به شدت به تنظیم اولیه‌ی پارامترهای خود وابسته‌اند و برای مقابله با این موضوع نیز، تنظیم پارامتر پویای تاگوچی بسیار مفید خواهد بود. در ادامه نحوه‌ی مقایسه‌ی دو روش تنظیم پارامتر را توضیح می‌دهیم و استواری آنها را بررسی می‌کنیم.

ابتدا با استفاده از نرم‌افزار mintab و روش ذکر شده، (تنظیم پارامتر تاگوچی کلاسیک) بهترین و بدترین سطوح طراحی پارامتر را برای الگوریتم هارمونی سرچ بدون تنظیم پارامتر پویا در هر یک از چهار مسئله و در ۳۰۰ تکرار الگوریتم هارمونی سرچ به دست می‌آوریم. سپس با استفاده از همین پارامترها، الگوریتم هارمونی سرچ را با پارامترهای تنظیم‌شده‌ی پویا و شرط ۳۰۰ تکرار بدون بهبود الگوریتم برای ورود به تنظیم پارامتر تاگوچی پویا اعمال می‌کنیم. نتایج به دست آمده در ۱۰ تکرار را در نرم‌افزار Matlab ۲۰۱۰ برای هر یک از حالات زیر با یکدیگر مقایسه می‌کنیم.

۱. اجرای الگوریتم در بدترین سطوح پارامترهای الگوریتم هارمونی سرچ با استفاده از تنظیم پارامتر تاگوچی کلاسیک؛
۲. اجرای الگوریتم در بدترین سطوح پارامترهای الگوریتم هارمونی سرچ با استفاده از تنظیم پارامتر تاگوچی کلاسیک؛

۳. اجرای الگوریتم به همراه تنظیم پارامتر تاگوچی پویا، که در ابتدا از سطوح بهینه‌ی پارامترهای حالت اول استفاده کرده باشد.

۴. اجرای الگوریتم به همراه تنظیم پارامتر تاگوچی پویا که در ابتدا از سطوح پارامتر مشخص شده حالت دوم استفاده کرده باشد.

چنان‌که در جدول ۵ ملاحظه می‌کنید میانگین جواب‌های به دست آمده از روش تنظیم پارامتر تاگوچی پویا در تمام شرایط، از روش تنظیم پارامتر تاگوچی کلاسیک بهتر است، و نیز در تمامی موارد دارای انحراف معیار کم‌تر و کمیته‌ی جواب بهتر و

جدول ۶. مقایسه بهترین جواب‌ها.

سیستم حافظت سرعت بالا	پل	سری - موازی	سری	
۰٫۹۹۹۹۵۴۳۱	۰٫۹۹۹۸۸۸۹۱	۰٫۹۹۹۹۷۵۸۸	۰٫۹۳۱۵۹۸۶	الگوریتم هارمونی سرچ با تنظیم پارامتر پویا
۰٫۹۹۹۹۵۳	۰٫۹۹۹۸۸۹۵۷	-	-	کولهو [۳۴]
۰٫۹۹۹۹۴۲	۰٫۹۹۹۸۸۹۲۱	۰٫۹۹۹۹۷۶۵۸	۰٫۹۳۱۶۷۸	چن [۶]
-	۰٫۹۹۹۷۸۹۳۷	۰٫۹۹۹۹۶۸۷۵	۰٫۹۳۱۳۶۳	هیکیتا و همکاران [۳]
-	۰٫۹۹۹۸۷۹۱۶	۰٫۹۹۹۹۷۴۱۸	۰٫۹۳۱۵۷۸	هسیه و همکاران [۵]

بهینه‌سازی تخصیص قابلیت اطمینان - اجزاء مورد بررسی قرار گرفت. نتایج به دست آمده مؤید این مطلب است که روش پیشنهادی جواب‌های بهتر و استواری بیشتری نسبت به تنظیم پارامتر تاگوچی کلاسیک ارائه می‌دهد. نکته‌ی مهم دیگر این است که می‌توان روش پیشنهادی را در بسیاری از الگوریتم‌های فراابتکاری به‌کار برد، که این امر می‌تواند موجب بهبود نتایج الگوریتم‌های دیگر شود.

۶. نتیجه‌گیری و پیشنهادات آتی

در نوشتار حاضر عملکرد روش پیشنهادی تنظیم پویای پارامترهای اولیه‌ی الگوریتم فراابتکاری هارمونی سرچ، که مبتنی بر روش تاگوچی است، برای حل مسئله‌ی

پانویس‌ها

1. implicit enumeration
2. Benchmark
3. exploitation
4. differential evolution
5. harmony memory size
6. harmony memory considering rate
7. pitch adjusting rate
8. Bandwidth
9. solution accuracy
10. gradinet search procedure
11. number of iterations without improvement

منابع (References)

1. Wang, L. and Li, L.-P. "Coevolutionary differential evolution with harmony search for reliability-redundancy optimization", *Journal Expert Systems with Applications*, **39**, pp. 5271-5278 (2012).
2. Kuo, W. and Prasad, V.R. "An annotated overview of system-reliability optimization", *IEEE Transactions on Reliability*, **49**(2), pp. 176-87 (2000).
3. Hikita, M., Nakagawa, Y. and Harihisa, H. "Reliability optimization of systems by a surrogate constraints algorithm", *IEEE Transactions on Reliability*, **41**(3), pp. 473-80 (1992).
4. Kuo, W. and Wan, R. "Recent advances in optimal reliability allocation", *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, **37**(2), pp. 143-156 (2007).
5. Hsieh, Y.C., Chen, T.C. and Bricker, D.L. "Genetic algorithm for reliability design problems", *Microelectronic Reliability*, **38**, pp. 1599-1605 (1998).
6. Chen, T.C. "IAs based approach for reliability-redundancy allocation problems", *Applied Mathematics and Computation*, **182**(2), pp. 1556-1567 (2006).
7. Tillman, F.A., Hwang, C.L. and Kuo, W. "Determining component reliability and redundancy for optimum system reliability", *IEEE Transactions on Reliability*, **26**(3), pp. 162-5 (1977).
8. Gopal, K. and Aggarwal, K.K. "A new method for solving reliability optimization problem", *IEEE Transactions on Reliability*, **28**, pp. 36-8 (1978).
9. Yeh, W.-C. and Hsieh, T.-J. "Solving reliability redundancy allocation problems using an artificial bee", *Computers & Operations Research*, **38**, pp. 1465-1473 (2011).
10. Chern, M.S. "On the computational complexity of reliability redundancy allocation in a series system", *Oper. Res. Lett.*, **11**, pp. 309-315 (1992).
11. Ha, C. and Kuo, W. "Reliability redundancy allocation: An improved realization for nonconvex nonlinear programming problems", *Eur. J. Oper. Res.*, **171**, pp. 24-38 (2006).
12. Kuo, W., Prasad, V.R., Tillman, F.A. and Hwang, C.L., *Optimal Reliability Design: Fundamentals and Applications*, Cambridge University Press, Cambridge (2001).
13. Prasad, V.R. and Kuo, W. "Reliability optimization of coherent system", *IEEE Trans. Reliab.*, **49**, pp. 323-330 (2000).
14. 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**, pp. 1297-1307 (2011).
15. Liang, Y.C. and Chen, Y.C. "Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm", *Reliability Engineering and System Safety*, **92**, pp. 323-31 (2007).

16. Yun, W.Y., Song, Y.M. and Kim, H.G. "Multiple multi-level redundancy allocation in series systems", *Reliability Engineering & System Safety*, **92**, pp. 308-13 (2007).
17. Kuo, W., Hwang, C.L. and Tillman, F.A. "A note on heuristic methods in optimal system reliability", *IEEE Transactions on Reliability*, **R27**, pp. 320-4 (1978).
18. Xu, Z., Kuo, W. and Lin, H.H. "Optimization limits in improving system reliability", *IEEE Transactions on Reliability*, **R39**, pp. 51-60 (1990).
19. Yokota, T., Gen, M. and Li, Y.X. "A genetic algorithm for interval nonlinear integer programming problem", *Computers and Industrial Engineering*, **30**(4), pp. 905-17 (1996).
20. Dhingra, A.K. "Optimal apportionment of reliability & redundancy in series systems under multiple objectives", *IEEE Transactions on Reliability*, **41**(4), pp. 576-82 (1992).
21. Geem, Z.M., *Music-Inspired Harmony Search Algorithm: Theory and Applications*, 1st ed., Berlin: Springer (2009).
22. Geem, Z.W., Kim, J.H. and Loganathan, G.V. "A new heuristic optimization algorithm: Harmony search", *Simulation*, **76**(2), pp. 60-68 (2001).
23. Zou, D., Gao, L., Wu, J., Li, S. and Li, Y. "A novel global harmony search algorithm for reliability problems", *Computers & Industrial Engineering*, **58**(2), pp. 307-316 (2010).
24. Zeblah, A., Chatelet, E., Yalaoui, F., Samrout, M. and Massim, Y. "Reliability optimization using harmony search algorithm under performance and budget constraints", *International Journal of Reliability and Safety*, **4**(2-3), pp. 299-317 (2010).
25. Nahas, N. and Thien-My, D. "Harmony search algorithm: Application to the redundancy optimization problem", *Engineering Optimization*, **42**(9), pp. 845-861 (2010).
26. Mahdavi, M., Fesanghary, M. and Damangir, E. "An improved harmony search algorithm for solving optimization problems", *Applied Mathematics and Computation*, **188**, pp. 1567-1579 (2007).
27. Yeh, W.-C. and Hsieh, T.-J. "Solving reliability redundancy allocation problems using an artificial bee", *Computers & Operations Research*, **38**, pp. 1465-1473 (2011).
28. Zou, D., Gao, L., Li, S. and Wu, J. "An effective global harmony search algorithm for reliability problems", *Expert Systems with Applications*, **38**, pp. 4642-4648 (2011).
29. Najafi, A., Akhavan Niaki, S.T. and Shahsavara, M. "A parameter-tuned genetic algorithm for the resource investment problem with discounted cash flows and generalized precedence relations", *Computers & Operations Research*, **36**, pp. 2994-3001 (November 2009).
30. Montgomery, D.C., *Design and Analysis of Experiments*, 5th ed. John Wiley (2001).
31. Marković, D., Madić, M. and Petrović, G. "Assessing the performance of improved harmony search algorithm (IHSA) for the optimization of unconstrained", *Scientific Research and Essays*, **7**(12), pp. 1312-1318 (30 March 2012).
32. Phadke, M.S., *Quality Engineering Using Robust Design*, Upper Saddle River, NJ, Prentice-Hall (1989).
33. Sadjadi, S. and Soltani, R. "An efficient heuristic versus a robust hybrid meta-heuristic for general framework of serial-parallel redundancy problem", *Reliability Engineering and System Safety*, **94**, pp. 1703-1710 (2009).
34. Coelho, L.D.S. "An efficient particle swarm approach for mixed-integer programming in reliability-redundancy optimization applications", *Reliability Engineering and System Safety*, **94**(4), pp. 830-837 (2009a).