



توسعه یک روش برای حل مسأله قابلیت اطمینان مبتنی بر ترکیب تکنیک شبیه‌سازی و الگوریتم بهینه‌سازی ازدحام ذرات در شرایط عدم قطعیت

فریبرز جولای^{۱*}، محمدعلی زارع شوربجه^۲، سبحان جویبار^۳

۱. استاد گروه مهندسی صنایع دانشگاه تهران، تهران

۲. دانشجوی دکترا مدیریت گرایش تحقیق در عملیات دانشگاه تهران، تهران

۳. دانشجوی دکترا مدیریت گرایش تحقیق در عملیات دانشگاه تهران، تهران

خلاصه

امروزه مسأله قابلیت اطمینان از مهم‌ترین مسائل پیش رو در مراحل اولیه برنامه‌ریزی، طراحی، و کنترل سیستم‌های تولیدی است. در این مقاله، نویسندگان در تلاش برای توسعه یک روش ترکیبی شبیه‌سازی-بهینه‌سازی برای حل مسأله قابلیت اطمینان در یک سیستم سری-موازی در شرایط عدم قطعیت با محدودیت‌های تصادفی هستند. ما روشی را پیشنهاد می‌کنیم که در آن ابتدا به کمک تکنیک شبیه‌سازی جمعیتی از جواب‌های نسبتاً خوب تولید، و در ادامه با کمک روش بهینه‌سازی ازدحام ذرات (PSO) این جواب‌ها بهبود داده می‌شوند. برای ارزیابی کارایی الگوریتم پیشنهادی، نتایج این تحقیق با تحقیقات مشابه پیشین مقایسه گردید. یافته‌ها نشان از برتری الگوریتم پیشنهادی دارد. نقطه قوت الگوریتم پیشنهادی بالاتر بودن میانگین جواب‌های بدست آمده و پایین‌تر بودن انحراف معیار بهترین جواب‌ها در تکرارهای متوالی است.

اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۳۹۳/۸/۱۰

پذیرش ۱۳۹۴/۱/۲۳

کلمات کلیدی:

مسأله قابلیت اطمینان

شبیه‌سازی

بهینه‌سازی ازدحام ذرات

عدم قطعیت

۱- مقدمه

کاملی درباره سیستم و رفتار اجزای آن وجود دارد. برای تکمیل اطلاعات احتمالی^۱ دو شرط باید برقرار باشد [۱]:

۱. تمامی توزیع‌های احتمال مشخص یا به شکل دقیقی قابل اندازه‌گیری باشند.
۲. اجزاء سیستم مستقل باشند، یعنی تمامی متغیرهای تصادفی توصیف‌کننده رفتار قابلیت اطمینان اجزاء مستقل‌اند.

در صورتی که حداقل یک شرط برقرار نباشد تنها می‌توان قابلیت اطمینان فاصله‌ای را بدست آورد. در دنیای واقعی در بسیاری از موارد که سیستم جدید یا تنها در قالب یک پروژه است، داده‌های آماری دقیقی وجود ندارد. بدین معنا که اطلاعات ناکاملی درباره اجزای سیستم وجود دارد. بنابراین قابلیت اطمینان یک جزء از سیستم عدد

قابلیت اطمینان را احتمال بقای یک سیستم و عدم توقف آن به شکل رضایت‌بخش حداقل در یک دوره زمانی مفروض تحت شرایط مشخص تعریف می‌کنند. مسأله طراحی یک سیستم با قابلیت اطمینان بالا با در نظر گرفتن محدودیت منابعی چون بودجه، وزن و اندازه موسوم به بهینه‌سازی قابلیت اطمینان است. در بسیاری از تحقیقات انجام شده روی مهندسی قابلیت اطمینان فرض می‌شود که تمام احتمالات به شکل قطعی مشخص هستند. بدین معنی که هر یک از احتمالات به نحو دقیقی قابل اندازه‌گیری و اطلاعات احتمالی

* نویسنده مسئول. فریبرز جولای

تلفن: ۰۲۱-۸۸۰۲۱۰۶۷؛ پست الکترونیکی: fjolai@ut.ac.ir

موازی، سری و پل با هدف حداکثر کردن قابلیت اطمینان و حداقل کردن هزینه با پارامترهای فازی و فاصله‌ای از این الگوریتم کمک گرفتند.

یکی از چالش‌هایی که ترکیب شبیه‌سازی و بهینه‌سازی به دنبال پاسخ آن است، مسأله عدم قطعیت است. مدل‌های متداول غیر قطعی (برنامه‌ریزی تصادفی، برنامه‌ریزی فازی، برنامه‌ریزی با اعداد فاصله‌ای) در مقایسه با شبیه‌سازی دقت و جزئی‌نگری کافی را ندارند. به منظور جبران این مشکل از نقطه قوت شبیه‌سازی (دقت و سرعت) می‌توان بهره گرفت [۳۷]. با توجه به اینکه مسأله بهینه‌سازی قابلیت اطمینان با محدودیت تصادفی و با قابلیت اطمینان اجزاء فاصله‌ای از جنس مسائل عدم قطعیت است، به منظور بالا بردن دقت و سرعت حل از ترکیب شبیه‌سازی و بهینه‌سازی استفاده شد. نویسندگان این مسأله را که پیش از این به وسیله الگوریتم ژنتیک^۶ (GA) حل شده است [۱]، به کمک یک الگوریتم توسعه‌یافته مبتنی بر ترکیب تکنیک شبیه‌سازی و بهینه‌سازی ازدحام ذرات^۷ (PSO) حل نموده‌اند. PSO پیش از این به تنهایی در حل مسأله قابلیت اطمینان [۳۲] استفاده شده بود، اما آنچه نویسندگان در این مقاله به دنبال آن بوده‌اند آزمون کردن ترکیب PSO و شبیه‌سازی بود. هدف از انجام این تحقیق، نشان دادن کارایی بیشتر روش ترکیبی پیشنهادی در بهبود جواب‌های مسأله است.

با توجه به انعطاف و پویایی روزافزون موجود در محیط تولیدی و تشدید عدم اطمینان‌های تحمیلی آن به واحدهای صنعتی، از نقاط قوت این تحقیق، ورود این عدم قطعیت‌های محیطی به مدل به صورت محدودیت‌های تصادفی است. در ادامه ابتدا ریاضیات فاصله‌ای و پس از آن مسأله مورد نظر تشریح می‌شود. سپس روش ترکیبی پیشنهادی برای حل مسأله فوق در شرایط عدم قطعیت ارائه و به کمک دو مثال عددی، نتایج حاصل با نتایج روش‌های GA، PSO و شبیه‌سازی مقایسه می‌شود. در ادامه نیز چهار مسأله با ابعاد بزرگ تر حل می‌شود و در پایان نتیجه‌گیری و پیشنهاداتی برای تحقیقات آتی ارائه می‌گردد.

۲- ریاضیات فاصله‌ای

یک عدد فاصله‌ای A یک بازه بسته است که به صورت $A = \{x: a_L \leq x \leq a_R, x \in R\}$ تعریف می‌شود و در آن به ترتیب a_L و a_R حد چپ و راست بازه است و R مجموعه تمام اعداد حقیقی است. یک فاصله A همچنین می‌تواند به صورت مرکز و شعاع به شکل $A = \langle a_C, a_W \rangle = \{x: a_C - a_W \leq x \leq a_C + a_W, x \in R\}$ تعریف شده که a_C و a_W به ترتیب معرف مرکز و شعاع است که $a_C = (a_L + a_R) / 2$ و $a_W = (a_R - a_L) / 2$. در حقیقت، هر عدد واقعی می‌تواند به عنوان یک عدد فاصله‌ای تعریف گردد، زیرا برای هر

نادقیقی است که می‌تواند به وسیله عددی فاصله‌ای (بازه‌ای) [۲] تعریف و به کمک احتمالات نادقیق محاسبه شود [۱].

در طول دو دهه گذشته تکنیک‌های زیادی برای حل مسائل بهینه‌سازی قابلیت اطمینان^۱ پیشنهاد شده است؛ از جمله روش برنامه‌ریزی پویا، روش شاخه و کران، روش ضریب لاگرانژ [۳-۵]. ژائو و لیو [۶] تکنیک برنامه‌ریزی تصادفی را برای مسائل تخصیص جایگزینی^۲ توسعه دادند. در سال ۱۹۹۰، رامچاندرا و سنکرینیان [۷] از شبیه‌سازی مونت‌کارلو با پارامترهای قطعی^۳ برای حل مسأله بهینه‌سازی یک سیستم سری استفاده کردند. پژوهشگران در تحقیقی [۸] از شبکه عصبی و پارامترهای قطعی برای حل مسأله بهینه‌سازی قطعی یک سیستم موازی n مرحله‌ای بهره جستند. لی و یانگ [۹] با هدف حداقل‌سازی عدم قابلیت اطمینان، یک سیستم پل پیچیده^۴ با پارامترهای قطعی را مدل‌سازی و آن را با روش مینی ماکس پارامتریک تکراری حل نمودند. سونگ و چو [۱۰] در سال ۲۰۰۰ از روش شاخه و کران برای حل مدل بهینه‌سازی سیستم سری با پارامترهای قطعی استفاده کردند.

از سال ۲۰۰۰ به بعد استفاده از روش‌های فرا ابتکاری برای حل مسأله بهینه‌سازی قابلیت اطمینان مورد توجه پژوهشگران قرار گرفت. به عنوان مثال، در مقالات [۱، ۲، ۱۱-۱۵] از الگوریتم ژنتیک، در [۱۶-۲۰] از الگوریتم کلونی مورچگان، در [۲۱-۲۳] از الگوریتم بهینه‌سازی زنبور عسل، در [۲۴] از الگوریتم جستجوی ممنوع، در [۲۵] از شبیه‌سازی تبرید، برای حل مسائل مختلف قابلیت اطمینان در سیستم‌های سری، موازی، سری-موازی با پارامترهای قطعی یا فازی یا فاصله‌ای استفاده شد.

برخی از محققان، علاقه‌مند به استفاده از روش‌های ترکیبی برای حل مسائل فوق بوده‌اند که نمونه‌های نسبتاً زیادی را می‌توان در ادبیات مشاهده نمود. از جمله پژوهش‌های [۲۶، ۲۷] که از ترکیب ژنتیک و شبکه عصبی و [۲۸-۳۰] که به ترتیب از ترکیب ژنتیک با شبیه‌سازی تبرید، جستجوی فاخته^۵ و جستجوی ممنوع کمک گرفته‌اند.

استفاده از الگوریتم بهینه‌سازی ازدحام ذرات نیز در حل مسائل مختلف قابلیت اطمینان نسبتاً مورد علاقه محققان بوده است. به عنوان مثال، یه [۳۱، ۳۲] از این الگوریتم در حل بهینه‌سازی انواع سیستم‌های سری، بجی و همکاران [۳۳] و صفایی و همکاران [۳۴] از این الگوریتم برای حداکثر کردن قابلیت اطمینان سیستم سری-موازی استفاده کردند. در سال ۲۰۱۲ چنبری و همکاران [۳۵] هم از این الگوریتم برای حل مسأله دو هدفه ماکزیمم کردن قابلیت اطمینان و مینیمم کردن هزینه بهره جستند. گرگ و همکاران [۳۶] و به ترتیب در سال‌های ۲۰۱۳ و ۲۰۱۴ در حل مسائل سری-

1. Reliability
2. Redundancy Allocation Problems
3. Deterministic
4. Complex bridge system
5. cuckoo

6. Genetic Algorithm

7. Particle Swarm Optimization

در منبع [۱] تعاریف گوناگونی برای روابط میان اعداد فاصله‌ای مطرح شده است. به طور کلی دو نوع تصمیم‌گیری وجود دارد. تصمیم‌گیری خوش‌بینانه و بدبینانه. در نوع خوش‌بینانه، تصمیم‌گیرنده بهترین گزینه را بدون در نظر گرفتن عدم اطمینان لحاظ می‌کند در حالی که در نوع بدبینانه، تصمیم‌گیرنده بهترین گزینه را با کمی عدم اطمینان انتخاب می‌نماید.

تعریف ۱. رابطه $\leq_{o\min}$ برای دو عدد فاصله‌ای $A = [a_L, a_R] = \langle a_C, a_W \rangle$ و $B = [b_L, b_R] = \langle b_C, b_W \rangle$ در حالت خوش‌بینانه برای مسئله مینیمم‌سازی به صورت زیر تعریف می‌گردد:

$$A \leq_{o\min} B \Leftrightarrow a_L \leq b_L, A <_{o\min} B \wedge A \neq B$$

بر اساس این تعریف، تصمیم‌گیرنده A را می‌پذیرد. رابطه $\leq_{o\min}$ متقارن^۱ نیست.

تعریف ۲. رابطه $\geq_{o\max}$ بین دو عدد فاصله‌ای $A = [a_L, a_R] = \langle a_C, a_W \rangle$ و $B = [b_L, b_R] = \langle b_C, b_W \rangle$ در حالت خوش‌بینانه برای مسئله ماکزیمم‌سازی به صورت زیر تعریف می‌گردد:

$$A \geq_{o\max} B \Leftrightarrow a_R \geq b_R, A >_{o\max} B \wedge A \neq B$$

بر اساس این تعریف، تصمیم‌گیرنده خوش‌بین A را انتخاب می‌کند. رابطه $\geq_{o\max}$ انعکاسی^۲ و تعدی^۳ است اما متقارن نیست.

تعریف ۳. رابطه $<_{p\min}$ میان دو عدد فاصله‌ای $A = [a_L, a_R] = \langle a_C, a_W \rangle$ و $B = [b_L, b_R] = \langle b_C, b_W \rangle$ در حالت بدبینانه برای مسئله مینیمم‌سازی به صورت زیر خواهد بود:

- I. برای اعداد فاصله- $A <_{p\min} B \Leftrightarrow a_C < b_C$

ای نوع ۱ و ۲

II. برای اعداد $A <_{p\min} B \Leftrightarrow a_C \leq b_C \wedge a_W < b_W$

فاصله‌ای نوع ۳

اما برای عدد فاصله‌ای نوع سوم تصمیم بدبینانه نمی‌تواند اتخاذ شود زمانی که $a_C < b_C \wedge a_W > b_W$.

تعریف ۴. رابطه $>_{p\max}$ بین دو عدد فاصله‌ای $A = [a_L, a_R] = \langle a_C, a_W \rangle$ و $B = [b_L, b_R] = \langle b_C, b_W \rangle$ در حالت بدبینانه برای مسئله ماکزیمم‌سازی به صورت زیر تعریف می‌شود:

I. برای اعداد $A >_{p\max} B \Leftrightarrow a_C > b_C$

فاصله‌ای نوع ۱ و ۲

II. برای اعداد $A >_{p\max} B \Leftrightarrow a_C \geq b_C \wedge a_W < b_W$

فاصله‌ای نوع ۳

اما، برای عدد فاصله‌ای نوع ۳ تصمیم بدبینانه زمانی که $a_C < b_C \wedge a_W < b_W$ است اتخاذ نمی‌شود.

۳- فرموله‌بندی مسئله

$x, x \in R$ را می‌توان به عنوان یک فاصله $[x, x]$ با شعاع ۰ تعریف نمود.

در اینجا عملگرهای ریاضی چون جمع، تفریق، ضرب اسکالر، ضرب، تقسیم و توان اعداد فاصله‌ای [۱] تشریح می‌گردد. این عملگرها حساب فاصله‌ای نامیده می‌شوند. اگر $A = [a_L, a_R] = \langle a_C, a_W \rangle$ و $B = [b_L, b_R] = \langle b_C, b_W \rangle$ مفروض باشد داریم:

$$I. A + B = [a_L + b_L, a_R + b_R], A + B = \langle a_C + b_C \rangle$$

$$II. A - B = [a_L - b_R, a_R - b_L]$$

$$III. \lambda A = \begin{cases} [\lambda a_L, \lambda a_R] & \text{if } \lambda \geq 0 \\ [\lambda a_R, \lambda a_L] & \text{if } \lambda < 0 \end{cases} \lambda A = \lambda \langle a_C, a_W \rangle = \langle \lambda a_C, |\lambda| a_W \rangle$$

$$IV. A \times B = [\min(a_L b_L, a_L b_R, a_R b_L, a_R b_R), \max(a_L b_L, a_L b_R, a_R b_L, a_R b_R)]$$

$$V. \frac{B}{A} = B \times \frac{1}{A} = [b_L, b_R] \times \left[\frac{1}{a_R}, \frac{1}{a_L} \right], 0 \notin [a_L, a_R]$$

$$VI. A^n = \begin{cases} [1, 1] & \text{if } n=0 \\ [a_L^n, a_R^n] & \text{if } a_L \geq 0 \text{ or } n \text{ فرد} \\ [a_R^n, a_L^n] & \text{if } a_R \leq 0, n \text{ زوج} \\ [0, \max(a_L^n, a_R^n)] & \text{if } a_L \leq 0 \leq a_R, n > 0 \text{ زوج} \end{cases}$$

$$VII. (A)^{\frac{1}{n}} = [a_L, a_R]^{\frac{1}{n}} = \sqrt[n]{[a_L, a_R]} = \begin{cases} [\sqrt[n]{a_L}, \sqrt[n]{a_R}] & \text{if } a_L \geq 0 \text{ or } n \text{ فرد} \\ [0, \sqrt[n]{a_R}] & \text{if } a_L \leq 0, a_R \geq 0, n \text{ زوج} \\ [\emptyset] & \text{if } a_R < 0, n \text{ زوج} \end{cases}$$

۱-۲- میانگین و واریانس

اگر $X_i = [X_{iL}, X_{iR}], i = 1, 2, \dots, n$ امین عدد فاصله‌ای باشد، میانگین، واریانس، و انحراف استاندارد استاندارد $X_i = [X_{iL}, X_{iR}], i = 1, 2, \dots, n$ برابر است با:

$$\bar{X} = \left[\frac{1}{n} \sum_{i=1}^n x_{iL}, \frac{1}{n} \sum_{i=1}^n x_{iR} \right]$$

$$\text{var}(X) = \frac{1}{n} \sum_{i=1}^n \left(\left[x_{iL} - \frac{1}{n} \sum_{i=1}^n x_{iR}, x_{iR} - \frac{1}{n} \sum_{i=1}^n x_{iL} \right] \right)^2$$

$$\sigma_x = \sqrt{\text{var}(X)} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\left[x_{iL} - \frac{1}{n} \sum_{i=1}^n x_{iR}, x_{iR} - \frac{1}{n} \sum_{i=1}^n x_{iL} \right] \right)^2}$$

۲-۲- روابط (کوچکتر، بزرگتر و مساوی) اعداد فاصله‌ای

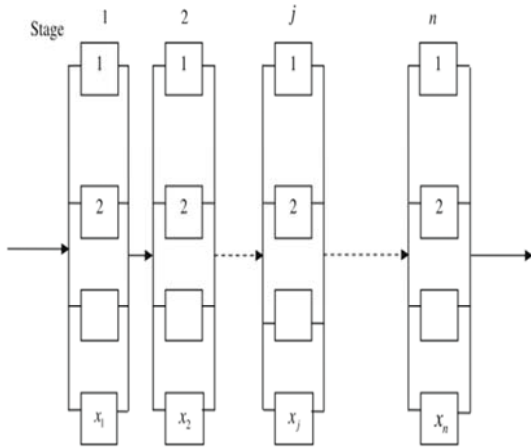
اگر A و B دو عدد فاصله‌ای باشند. این دو عدد یکی از سه حالت زیر را خواهند داشت:

نوع ۱: A و B جدا از هم‌اند.

نوع ۲: A و B با یکدیگر اشتراک دارند.

نوع ۳: یا $A \subset B$ است و یا $B \subset A$.

1. Symmetric
2. Reflexive
3. Transitive



شکل(۱): سیستم سری n مرحله‌ای

مسأله بهینه‌سازی با محدودیت تصادفی برای سیستم سری-موازی با m محدودیت تصادفی می‌تواند به صورت زیر فرموله گردد.

$$\text{Maximize } R_S(x) = \prod_{j=1}^n [R_{SL}(x), R_{SR}(x)]$$

St.

$$\text{Prob}[g_i(x) \leq c_i] \geq 1 - \alpha_i, \quad i = 1, 2, \dots, m$$

$$lo_j \leq x_j \leq up_j, \quad j = 1, 2, \dots, n$$

از آنجا که مجموعه محدودیت مربوط به مدل بالا احتمالی است، در ادامه روش تبدیل این محدودیت‌ها به محدودیت‌های قطعی برای دو توزیع یکنواخت و نرمال تشریح می‌گردد.

می‌دانیم که متغیر تصادفی X دارای توزیع یکنواخت است به شرطی که تابع احتمال آن دارای رابطه زیر باشد:

$$f(x) = \begin{cases} \frac{1}{u-l}; & l < x < u \\ 0; & \text{otherwise} \end{cases}$$

زمانی که یک متغیر تصادفی X توزیع یکنواخت دارد، آن را به صورت $X \square U(l, u)$ نشان می‌دهیم. وقتی c_i دارای توزیع یکنواخت باشد یعنی $c_i \square U(l_i, u_i)$ ، محدودیت احتمالی $\text{Prob}[g_i(x) \leq c_i] \geq 1 - \alpha_i$ می‌تواند به صورت محدودیت قطعی

$$g_i(x) \leq \gamma_i \quad \text{یا} \quad \int_{\gamma_i}^{u_i} \frac{1}{u_i - l_i} dx = 1 - \alpha_i$$

که در آن $\gamma_i = l_i \beta_i + u_i \alpha_i$ است. در این حالت به جای دسته محدودیت‌های اول مسأله (رابطه ۲) محدودیت $g_i(x) \leq l_i \beta_i + u_i \alpha_i, i = 1, 2, \dots, m$ جایگزین می‌گردد.

شایان ذکر است در صورتی که c_i دارای توزیع نرمال باشد یعنی $c_i \square N(\mu_{c_i}, \sigma_{c_i}^2)$ (که در آن μ_{c_i} معرف میانگین از $c_i = E(c_i)$ و $\sigma_{c_i}^2$ بیانگر واریانس از $c_i = \text{Var}(c_i)$) محدودیت $\text{Prob}[g_i(x) \leq c_i] \geq 1 - \alpha_i$ می‌تواند به صورت

$$g_i(x) \leq \mu_{c_i} + e_i \sigma_{c_i}$$

که از رابطه $\phi(e_i) = \alpha_i$ بدست می‌آید. در اینجا

جدول (۱): نمادهای به کار رفته در مدل ریاضی

نماد	تعریف
n	تعداد زیرسیستم
x_j	متغیر تصمیم (تعداد اجزاء j امین زیرسیستم)
R_S	قابلیت اطمینان کل سیستم
R_{SL}	حد پایین قابلیت اطمینان کل سیستم
R_{SR}	حد بالای قابلیت اطمینان کل سیستم
r_{jL}	حد پایین قابلیت اطمینان هر جزء از j امین زیرسیستم
r_{jR}	حد بالای قابلیت اطمینان هر جزء از j امین زیرسیستم
α_i	سطح اطمینان محدودیت تصادفی i ام
m	تعداد محدودیت تصادفی
lo_j	حد پایین j امین متغیر تصمیم
up_j	حد بالای j امین متغیر تصمیم
l	حد پایین تابع توزیع یکنواخت
u	حد بالای تابع توزیع یکنواخت

پیش از بیان مدل ریاضی مسأله لازم است که مفروضات مدل مطرح گردد. این مفروضات عبارتند از:

۱. قابلیت اطمینان هر جزء نادقیق است و دارای مقدار فاصله‌ای است.

۲. اگر یک جزء از هر زیرسیستم خراب شود، کل سیستم آسیب ندیده و از کار نخواهد افتاد.

۳. تمام اجزاء فعالند و بدون تعمیر کار می‌کنند.

۴. حالت اجزاء شامل دو وضعیت عملیاتی و خراب است.

۵. محدودیت‌های منابع ماهیت تصادفی دارند و از توابع توزیعی با پارامترهای احتمالی تبعیت می‌کنند.

فرض می‌شود که سیستم تحت بررسی دارای n زیرسیستم به شکل سری است که در آن j امین زیرسیستم ($1 \leq j \leq n$) دارای x_j جزء به شکل موازی است. چنین سیستمی سیستم سری-موازی یا سیستم n مرحله‌ای نامیده می‌شود (شکل (۱) را ببینید). با فرض اینکه تمامی اجزاء درون یک زیرسیستم یکسان هستند، قابلیت اطمینان سیستم R_S برابر با رابطه زیر خواهد بود:

$$R_S(x) = \prod_{j=1}^n [R_{SL}(x), R_{SR}(x)]$$

که

$$R_{SL}(x) = [1 - (1 - r_{jL})^{x_j}], R_{SR}(x) = [1 - (1 - r_{jR})^{x_j}] \quad (۱)$$

دیگر باشد را ذره^۵ گویند [۴۲]. در مدل پایه‌ای PSO، یک ازدحام شامل N ذره می‌شود که در یک فضای D بعدی در حال پروازند. هر ذره i یک جواب کاندید برای مسأله در تکرار^۶ t ام الگوریتم است که با بردار $x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{id}^t)$ در فضای جواب نشان داده می‌شود. هر ذره موقعیت و سرعت خاصی دارد که جهت حرکت و گام‌های آن را مشخص می‌کند. هر ذره موقعیت خود (x_i^t) را بر اساس دو عامل، به سوی بهینه کلی تغییر می‌دهد: (۱) بهترین موقعیت خود که تا کنون تا تکرار فعلی به دست آورده است که با $p_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{id}^t)$ نشان می‌دهند و (۲) بهترین موقعیتی که تا کنون به وسیله یکی از ذرات کل ازدحام به دست آمده است که با $p_g^t = (p_{g1}^t, p_{g2}^t, \dots, p_{gd}^t)$ نشان می‌دهند. $pbest_i$ و $gbest$ نشان‌دهنده مقادیر تابع هدف (یا همان برازش^۷) بردارهای p_i^t و p_g^t هستند. همچنین بردار $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{id}^t)$ نشان‌دهنده بردار سرعت ذره i در تکرار t ام الگوریتم است [۴۲-۴۵]. در تکرار t ام موقعیت ذره i ام با اضافه شدن موقعیت قبلی آن با سرعت جدید، طبق معادله زیر به‌روزرسانی می‌شود:

$$X_{id}^t = X_{id}^{t-1} + v_{id}^t \quad i=1,2,\dots,N \quad d=1,2,\dots,D \quad (3)$$

همچنین بردار v_{id}^t طبق رابطه زیر به‌روزرسانی شده و در معادله فوق قرار می‌گیرد:

$$v_{id}^t = w.v_{id}^{t-1} + c_1 r_1 (p_{id}^{t-1} - x_{id}^{t-1}) + c_2 r_2 (p_{gd} - x_{id}^{t-1}) \quad i=1,2,\dots,N \quad \text{و} \quad d=1,2,\dots,D \quad (4)$$

در معادله فوق که رابطه به‌روزرسانی سرعت نامیده می‌شود، c_1 و c_2 دو عدد مثبت حقیقی‌اند که ثابت‌های شتاب^۸ یا فاکتورهای یادگیری^۹ نامیده می‌شوند. پارامترهای r_1 و r_2 تضمین می‌کنند که الگوریتم تصادفی است، زیرا این دو پارامتر اعدادی مستقل و تصادفی در بازه $[0,1]$ هستند [۴۲-۴۵]. وزن اینرسی^{۱۰} نامیده می‌شود که نشان‌دهنده تأثیر سرعت پیشین ذره بر سرعت فعلی آن است. معمولاً پارامتر w در فاصله صفر و یک است [۴۲]. هر چه وزن w بزرگ‌تر باشد گستره جستجوی کلی بیشتر (یعنی جستجوی پراکنده‌تر) و هر چه مقدار آن کوچک‌تر باشد جستجو محلی و عمیق‌تر می‌شود [۴۲، ۴۳] (شکل (۲) را ببینید).

محدودیت‌های اول مسأله (رابطه ۲) محدودیت $g_i(x) \leq \mu_i + e_i \sigma_i, i=1,2,\dots,m$ جایگزین می‌گردد.

۴- تکنیک شبیه‌سازی

شبیه‌سازی تقلیدی از عملکرد فرایند یا سیستم واقعی با گذشت زمان است [۳۸]. شبیه‌سازی آزمودن مدل‌ها است [۳۹]. شبیه‌سازی نوعی رونوشت برداری از یک سیستم است [۴۰]. شبیه‌سازی علاوه بر نشان دادن میزان تغییرپذیری، روابط متقابل و پیچیدگی یک سیستم، امکان پیش‌بینی رفتار سیستم را نیز فراهم می‌کند [۴۰]. برای اثربخشی هر چه بیشتر شبیه‌سازی آن را با توجه به نوع مسأله با تکنیک‌های متنوع بهینه‌سازی ترکیب می‌کنند. قابلیت‌های ترکیب شبیه‌سازی و بهینه‌سازی گسترده بوده و طراحی مناسب آن به میزان زیادی به خصوصیات مسأله مورد نظر بستگی دارد. جهش‌های عظیم در قدرت محاسبه رایانه‌های نسل جدید باعث گسترش روز افزون روش‌های ترکیبی شده است. ایده اصلی در ترکیب شبیه‌سازی و بهینه‌سازی استفاده همزمان از «مزیت دقت و سرعت بسیار زیاد» شبیه‌سازی و مزیت «توانایی یافتن جوابهای خوب یا بهینه» توسط بهینه‌سازی است. یکی از مزایای اصلی شبیه‌سازی توانایی آن در رویارویی با فرایندهای پیچیده (قطعی یا تصادفی) بدون درگیر شدن در پیچیدگی‌های ریاضی^۱ است [۳۷].

در یک دسته‌بندی ارائه شده در [۳۷] روش‌های ترکیبی شبیه‌سازی- بهینه‌سازی به دو دسته کلی «مدل‌های ترکیبی^۲» و «مدل‌سازی ترکیبی^۳» تقسیم‌بندی شده‌اند. در دسته نخست دو مدل تحلیلی و شبیه‌سازی با هم در قالب یک مدل ترکیب می‌شوند. در حالیکه در دسته دوم هر مدل قادر به تولید یک جواب کامل است و جواب نهایی از تبادل اطلاعات بین دو مدل منتج می‌شود. در مقاله حاضر، هم تکنیک شبیه‌سازی و هم مدل PSO ارائه شده به تنهایی قادر به تولید جواب موجه می‌باشند از این رو، مدل ما در این مقاله از نوع دسته دوم است.

۵- الگوریتم بهینه‌سازی ازدحام ذرات (PSO)

الگوریتم PSO یک تکنیک بهینه‌سازی تصادفی الهام گرفته شده از رفتار اجتماعی نظام‌های طبیعی چون گله پرندگان و ماهی‌ها است که نخستین بار توسط کندی و ابرهارت در سال ۱۹۹۵ معرفی شد [۴۱]. در اصطلاح، این اجتماعات جانوری را ازدحام^۴ و هر یک از اعضای این اجتماعات جانوری که می‌تواند ماهی، پرنده یا جاندار

5. Particle
6. Iteration
7. Fitness
8. Acceleration constants
9. Learning factors
10. Inertia weight

1. mathematical sophistication
2. hybrid models
3. hybrid modeling
4. Swarm

$$C_w = c_w = 0.1, \quad C_p = C_w + c_p = 0.35, \quad C_g = C_p + c_g = 0.7,$$

$$C_1 = \frac{1-0.7}{U_i - L_i + 1} = \frac{1-0.7}{9-0+1} = 0.03 = C_2, \quad C_3 = 0.06,$$

$$C_4 = C_5 = 0.05$$

زیرا:

$$C_g + 7(0.03) = 0.91 = \rho_1 < C_g + 8(0.03),$$

$$C_p < \rho_2 = 0.45 < C_g, \quad C_w < \rho_3 = 0.28 < C_p,$$

$$\rho_4 = 0.05 < C_w,$$

$$C_g + 5(0.03) < \rho_5 = 0.87 < C_g + 6(0.03),$$

بنابراین ما داریم $X_4^4 = (7, 3, 3, 4, 5)$.

۶- الگوریتم پیشنهادی

همان طور که پیش از این بیان شد در این تحقیق از روش ترکیب شبیه‌سازی با مدل گسسته بهینه‌سازی ازدحام ذرات استفاده می‌شود. بنابراین قبل از هر چیز بایستی پارامترهای مورد استفاده در DPSO و جمعیت اولیه^۲ برای شروع الگوریتم و نیز حد پایین و حد بالای هر یک از متغیرهای تصمیم را با توجه به ماهیت مسأله و نظر تصمیم‌گیرندگان مشخص نمود. با توجه به مشابهت مسأله این تحقیق و مسأله بحث شده در [۳۲]، پارامترهای الگوریتم مستقیماً از این مقاله استخراج و $c_w = 0.10$ ، $c_p = 0.30$ و $c_g = 0.50$ قرار داده شد.

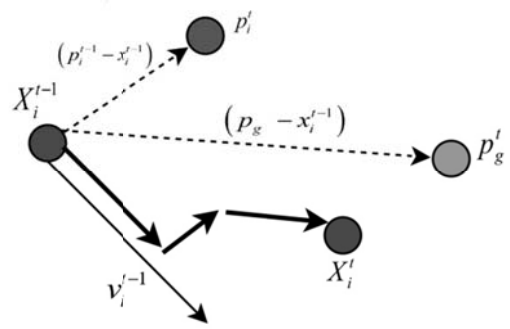
برای انتخاب جمعیت اولیه از شبیه‌سازی^۳ ۱۰۰۰ جواب موجه تصادفی غیرتکراری در نرم‌افزار MATLAB بهره برده می‌شود و از این بین N جواب برتر (دارای مقدار تابع هدف بالاتر) به عنوان ذرات اولیه (جواب‌های اولیه) انتخاب می‌شوند. N تعداد ذرات موجود در جمعیت اولیه برای شروع الگوریتم است. انتخاب این ۱۰۰۰ جواب از تمام گستره منطقه موجه و به صورت کاملاً تصادفی بر مبنای توزیع یکنواخت است. برای اطمینان از قرار گرفتن جواب‌های تصادفی تولید شده در منطقه موجه، کلیه جواب‌های ناموجه با آزمون در محدودیت‌ها حذف می‌شوند.

با توجه به شکل (۱) موجودیت سیستم n مرحله‌ای مذکور منوط به وجود حداقل ۱ جزء در هر مرحله از سیستم است؛ یعنی تمام متغیرهای تصمیم دارای حد پایین ۱ هستند. برای بدست آوردن حد بالای هر متغیر تصمیم نیز، با اختصاص مقدار ۱ به سایر متغیرهای تصمیم در همه محدودیت‌ها، دستگاه نامعادله تک مجهول‌ای بدست خواهد آمد که با حل این دستگاه حد بالای متغیر تصمیم مورد نظر حاصل می‌شود. به عنوان مثال با فرض داشتن سه محدودیت زیر حدود متغیرهای تصمیم به شرح زیر بدست می‌آید:

$$2x_1 + x_2 + x_3 \leq 10$$

$$x_1 + 2x_2 + 3x_3 \leq 12$$

$$2x_1 + 2x_2 + x_3 \leq 8$$



شکل (۲): به روز رسانی موقعیت ذره در یک تکرار PSO

همچنین در PSO برای ممانعت از رشد بسیار سریع سرعت و نیز عدم خروج ذرات از فضای جستجو، هر کدام از مؤلفه‌های بردار سرعت محدود به بازه $[v_{\min}, v_{\max}]$ می‌شوند [۴۶، ۴۲].

۵-۱- مدل بهینه‌سازی ازدحام ذرات گسسته

مدل PSO ارائه شده توسط کندی و ابرهات برای مسائل پیوسته استفاده می‌شود. تا کنون نسخه‌های متعددی از PSO توسط محققان مختلف برای استفاده در فضای گسسته ارائه گردیده است. در این مقاله از مدل بهینه‌سازی ازدحام ذرات گسسته^۱ پیشنهادی در [۳۲] استفاده شده است. در ادامه به اختصار این مدل تشریح می‌شود:

اگر c_w ، c_p و c_g اعداد ثابت مفروض باشند، در این صورت برای به روز رسانی مؤلفه j ام ذره i ام در تکرار t ام از رابطه زیر استفاده می‌شود:

$$x_{ij}^t = \begin{cases} x_{ij}^t & \text{if } \rho_{ij}^t \in [0, C_w) \\ p_{ij}^{t-1} & \text{if } \rho_{ij}^t \in [C_w, C_p) \\ g_i & \text{if } \rho_{ij}^t \in [C_p, C_g) \\ x & \text{if } \rho_{ij}^t \in [C_g + x.\Delta_j, C_g + (x+1).\Delta_j) \end{cases} \quad (5)$$

که ρ_{ij}^t عدد تصادفی بین ۰ و ۱ است و $C_w = c_w$ ، $C_p = C_w + c_p$ ، $C_g = C_p + c_g$ ، $C_g = C_p + c_g$ ، $C_p = C_w + c_p$ ، $C_w = c_w$ که $\Delta_j = \frac{1-C_g}{U(x_j) - L(x_j) + 1}$ ، در آن $U(x_j)$ و $L(x_j)$ به ترتیب حد بالا و حد پایین x_j هستند. برای مثال، اگر داشته باشیم $X_4^3 = (1, 2, 3, 4, 5)$ ، $G_3 = (4, 3, 5, 3, 2)$ ، $P_4^3 = (6, 7, 3, 2, 4)$ ، $U(X) = (1, 10, 6, 6, 5)$ و $\rho = (0.91, 0.45, 0.25, 0.05, 0.87)$ که به ترتیب نشان‌دهنده مختصات ذره ۴ ام در تکرار سوم، مختصات بهترین موقعیتی که تا تکرار سوم توسط ذره ۴ ام کسب شده است، بهترین موقعیتی که توسط یکی از ذرات جمعیت تا تکرار سوم کسب شده است، حد پایین و حد بالای متغیرهای تصمیم و بردار اعداد تصادفی هستند. همچنین اگر $c_w = 0.1$ ، $c_p = 0.25$ و $c_g = 0.35$ باشد، داریم:

گام (۷): بهترین جواب بدست آمده تا کنون را نمایش بده (شکل (۳)).

۷- مثال عددی

برای حل مدل (رابطه ۲) با استفاده از الگوریتم پیشنهادی فوق، از شش مثال استفاده می‌شود. داده‌های دو مثال نخست عیناً از منبع [۱] اقتباس شده است تا امکان مقایسه نتایج روش ترکیبی پیشنهادی با الگوریتم DPSO، تکنیک شبیه‌سازی و الگوریتم ژنتیک پیشنهادی منبع مذکور فراهم گردد. بقیه مثال‌ها توسط نویسندگان با اندازه‌های بزرگتر (با تعداد متغیرهای تصمیم و تعداد محدودیت بیشتر) از دو مثال اول طراحی شده است.

داده‌های مثال ۱ و ۲ که یک مجموعه با چهار زیرسیستم و دو محدودیت تصادفی می‌باشند، به ترتیب در جداول (۲) و (۳) نشان داده شده‌اند. تفاوت مثال ۱ و ۲ در این است که در مثال دوم داده‌های مربوط به قابلیت اطمینان اجزاء با اقتباس از [۴۷] به صورت اعداد قطعی و سایر داده‌ها، تصادفی و مشابه مثال ۱ در نظر گرفته شده است. لازم به توضیح است که اعداد سمت راست محدودیت‌ها، تابع توزیع یکنواخت هستند.

برای حل این مثال‌ها به وسیله الگوریتم DPSO و روش ترکیبی پیشنهادی تعداد ذرات جمعیت اولیه، تعداد اجرای الگوریتم و تعداد تکرار در هر اجرا به ترتیب ۱۲۰، ۱۰ و ۱۲۰ در نظر گرفته شد. همچنین برای مقایسه ارزش اعداد فاصله‌ای نسبت به یکدیگر از تعریف ۴ بخش ۲ استفاده شده است. نتایج حاصل از حل دو مثال نخست به وسیله الگوریتم ژنتیک [۱۱]، تکنیک شبیه‌سازی، الگوریتم DPSO و روش ترکیبی پیشنهادی در جدول (۴) نمایش داده شده و مقایسه گردیده است.

با توجه به آنچه گذشت برای بدست آوردن حد بالای x_1 ، با قرار دادن مقدار ۱ به جای x_2 و x_3 در سه محدودیت بالا دستگاه زیر حاصل می‌شود:

$$\begin{cases} 2x_1 + 1 + 1 \leq 10 \\ x_1 + 2(1) + 3(1) \leq 12 \\ 2x_1 + 2(1) + 1 \leq 8 \end{cases} \Rightarrow \begin{cases} 2x_1 \leq 8 \\ x_1 \leq 7 \\ 2x_1 \leq 5 \end{cases}$$

جواب این دستگاه $x_1 \leq 2.5$ است. از آنجا که متغیر تصمیم عددی صحیح است بنابراین حد بالای نهایی $x_1 \leq 2$ می‌باشد. حد بالای سایر متغیرهای تصمیم نیز با همین روش $x_2 \leq 2$ و $x_3 \leq 3$ بدست می‌آید.

همانند مرحله شبیه‌سازی در هر تکرار از الگوریتم DPSO نیز با لحاظ کردن محدودیت‌های مسأله از تولید جواب‌های ناموجه جلوگیری می‌شود. برای اولویت دادن به جواب‌های تولید شده در هر تکرار و انتخاب بهترین جواب به دست آمده از مفهوم مرکز عدد فاصله‌ای و رابطه موجود در تعریف ۴ استفاده شده است.

بر این اساس با توجه به آنچه گذشت به طور کلی گام‌های الگوریتم پیشنهادی عبارتند از:

گام (۱): شروع الگوریتم با تولید ذرات جمعیت آغازین با استفاده از شبیه‌سازی

گام (۲): اختصاص پارامترهای c_p ، c_w و c_g ؛

گام (۳): انتخاب $pbest_i$ برای هر ذره و سپس $gbest$ ازدحام؛

گام (۴): به‌روزرسانی مختصات هر ذره (x_{ij}^t) با استفاده از رابطه (۵)؛

گام (۵): محاسبه مقدار تابع هدف به ازای x_{ij}^t جدید و به‌روزرسانی $pbest_i$ و $gbest$ در صورت نیاز؛

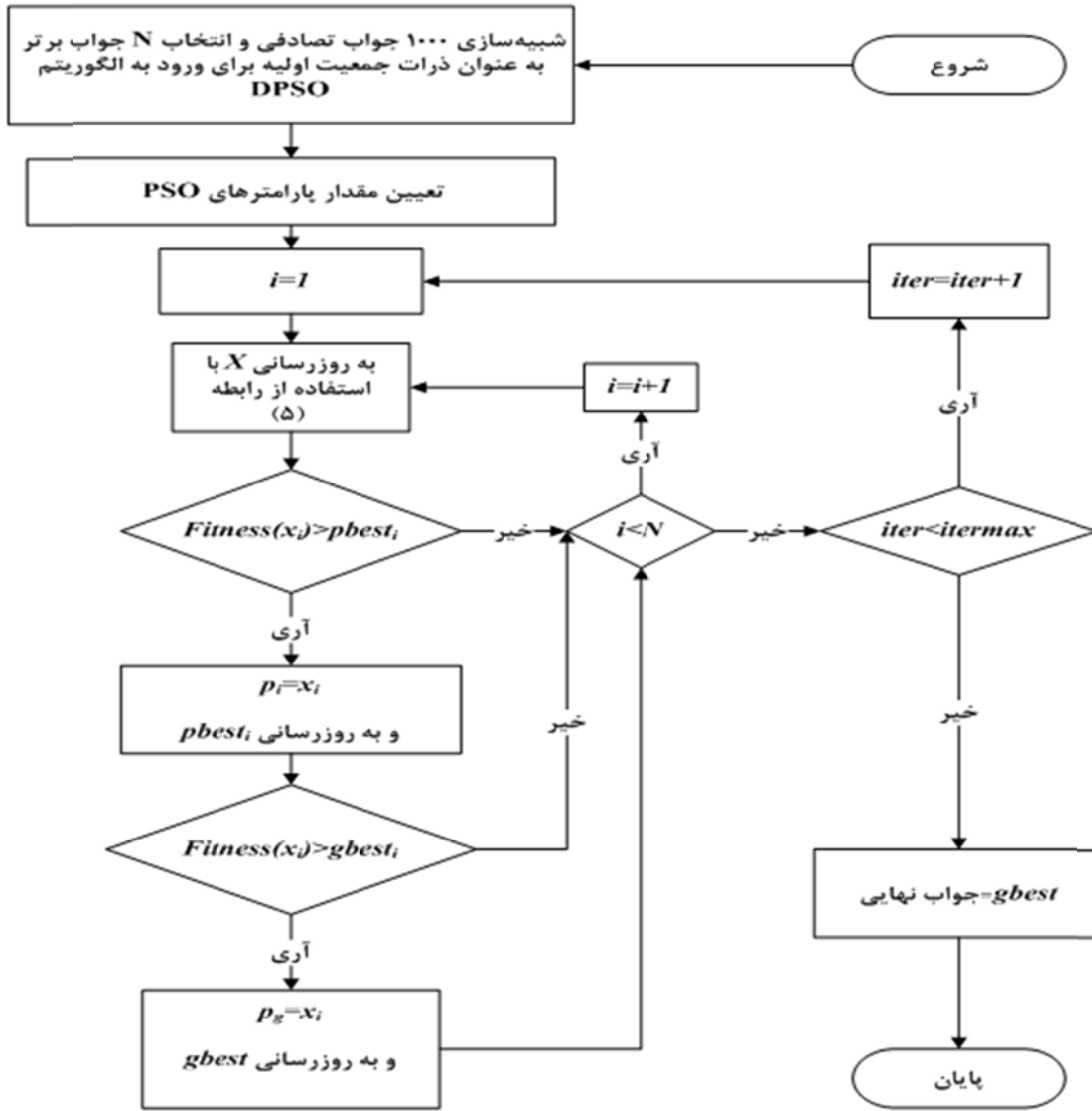
گام (۶): اگر شاخص توقف محقق گردید (رسیدن به سقف تکرار الگوریتم $(Itermax)$)، به گام (۷) برو در غیر این صورت به گام (۴) برو؛

جدول (۲): داده‌های مثال ۱

منابع در دسترس			زیرسیستم j ام					
	l_i	u_i	α_i	۴	۳	۲	۱	r_j
C_1	۵۰	۶۰	۰/۱۰	[۰/۸۳ و ۰/۸۶]	[۰/۷۳ و ۰/۷۸]	[۰/۷۸ و ۰/۸۱]	[۰/۷۴ و ۰/۷۶]	a_{1j}
C_2	۱۱۰	۱۴۰	۰/۱۵	۹/۰	۷/۰	۵/۰	۴/۰	a_{2j}

جدول (۳): داده‌های مثال ۲

منابع در دسترس			زیرسیستم j ام					
	l_i	u_i	α_i	۴	۳	۲	۱	r_j
C_1	۵۰	۶۰	۰/۱۰	[۰/۸۵ و ۰/۸۵]	[۰/۷۵ و ۰/۷۵]	[۰/۸۰ و ۰/۸۰]	[۰/۷۵ و ۰/۷۵]	a_{1j}
C_2	۱۱۰	۱۴۰	۰/۱۵	۹/۰	۷/۰	۵/۰	۴/۰	a_{2j}



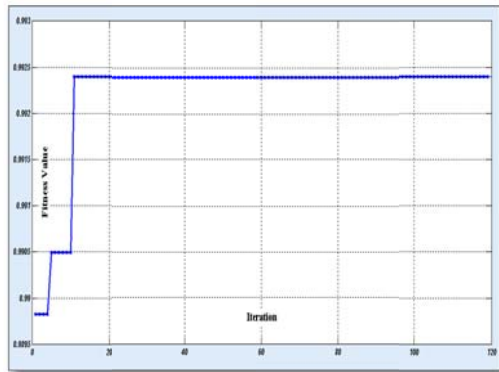
شکل (۳): فلوچارت الگوریتم پیشنهادی

جدول (۴): نتایج حاصل از حل مثال ۱ و ۲

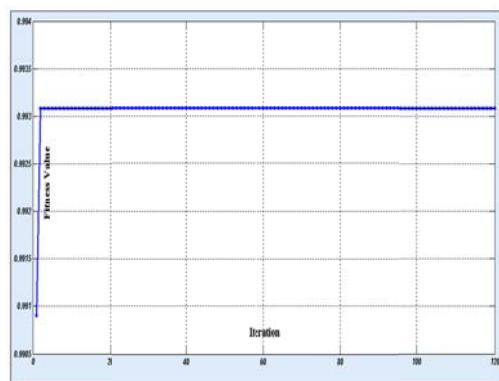
مثال	الگوریتم	مقدار متغیرهای تصمیم	بهترین قابلیت اطمینان یافته شده (R^*)	میانگین جواب‌ها (R^*)	انحراف استاندارد جواب‌ها (R^*)
۱	GA [۱]	(۵ و ۳ و ۴)	[۰/۹۹۰۱۵۴ و ۰/۹۹۴۶۵۰]	[۰/۹۹۰۱۵۴ و ۰/۹۹۴۶۵۰]	[۰/۰۰۰۰۰ و ۰/۰۰۱۴۲۱]
	شبیه‌سازی	(۵ و ۳ و ۴)	[۰/۹۹۰۱۵۴ و ۰/۹۹۴۶۵۰]	[۰/۹۸۸۰۴۲ و ۰/۹۹۳۶۳۴]	[۰/۰۰۰۰۰ و ۰/۰۰۶۲۹۰]
	DPSO	(۵ و ۳ و ۴)	[۰/۹۹۰۱۵۴ و ۰/۹۹۴۶۵۰]	[۰/۹۸۹۳۳۴ و ۰/۹۹۴۱۷۷]	[۰/۰۰۰۰۰ و ۰/۰۰۵۸۳۲]
	روش ترکیبی پیشنهادی	(۵ و ۳ و ۴)	[۰/۹۹۰۱۵۴ و ۰/۹۹۴۶۵۰]	[۰/۹۹۰۱۵۴ و ۰/۹۹۴۶۵۰]	[۰/۰۰۰۰۰ و ۰/۰۰۰۰۰]
۲	GA [۱]	(۵ و ۳ و ۴)	۰/۹۹۳۰۸۸	۰/۹۹۳۰۸۸	۰
	شبیه‌سازی	(۵ و ۳ و ۴)	۰/۹۹۳۰۸۸	۰/۹۹۱۶۱۲	۰/۰۰۱۰۱۴
	DPSO	(۵ و ۳ و ۴)	۰/۹۹۳۰۸۸	۰/۹۹۲۵۹۵	۰/۰۰۰۷۵۳
	روش ترکیبی پیشنهادی	(۵ و ۳ و ۴)	۰/۹۹۳۰۸۸	۰/۹۹۳۰۸۸	۰

۸- تجزیه و تحلیل یافته‌ها

احتمال دستیابی به جواب بهتر در روش پیشنهادی در تعداد تکرار معین نسبت به الگوریتم DPSO بالاتر خواهد بود.



شکل (۴): همگرا شدن الگوریتم به سمت جواب نهایی در مثال ۱



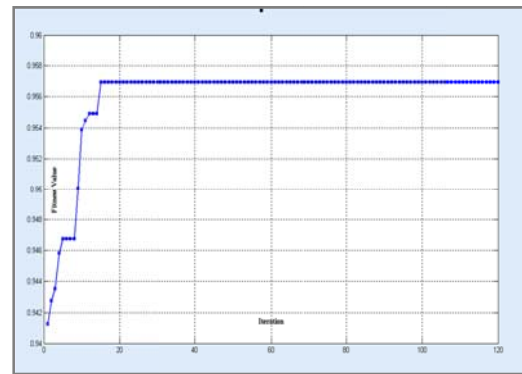
شکل (۵): همگرا شدن الگوریتم به سمت جواب نهایی در مثال ۲

همان طور که در جدول (۴) قابل مشاهده است، بهترین جواب به دست آمده توسط تمامی روش‌ها مقدار یکسانی دارند، اما آنچه روش ترکیبی پیشنهادی را کارتر نشان می‌دهد، پراکندگی کمتر جواب‌های به دست آمده از آن است (ستون آخر جدول (۴) را مشاهده کنید). در حقیقت در این روش پراکندگی جواب‌ها بر خلاف سه روش دیگر برابر با صفر است. شکل‌های (۴) و (۵) چگونگی همگرا شدن جواب‌های الگوریتم به سمت بهترین جواب به دست آمده را نشان می‌دهند. با ثابت شدن کارایی بیشتر الگوریتم پیشنهادی در مقایسه با دیگر روش‌ها، چهار مثال دیگر با اندازه‌های بزرگ‌تر طراحی و اجرا شد که نتایج حاصل از حل این مثال‌ها نیز در جدول (۵) نمایش داده شده است. داده‌های این مثال‌ها به ترتیب در جداول (۶) تا (۹) در ضمیمه الف قابل مشاهده است. شکل (۶) چگونگی همگرا شدن جواب‌های الگوریتم به سمت بهترین جواب به دست آمده در مثال ۶ را نشان می‌دهد. نمایش کاملی از یافته‌های حل مثال ۶ به تفصیل در ضمیمه ب قابل مشاهده است. شایان ذکر است که برای محاسبه میانگین و انحراف استاندارد جواب‌ها از مفاهیم ریاضیات فاصله‌ای بخش ۲ استفاده شده است. برای هر چهار مثال، روش ترکیبی پیشنهادی نسبت به تکنیک شبیه‌سازی جواب بهتری ارائه کرده است. بهترین جواب بدست آمده از روش ترکیبی و الگوریتم DPSO یکی است اما آنچه که روش پیشنهادی را کارتر نشان می‌دهد؛ بدست آوردن بهترین جواب بیشتر (به لحاظ فراوانی) در ۱۰ تکرار متوالی است. که این امر خود را در میانگین بهتر و پراکندگی کمتر نشان می‌دهد. به تعبیری دیگر می‌توان گفت که

جدول (۵): نتایج حاصل از حل مثال ۳ تا ۶

مثال	الگوریتم	مقدار متغیرهای تصمیم	بهترین قابلیت اطمینان یافته شده (R^*)	میانگین جواب‌ها (R^*)	انحراف استاندارد جواب‌ها (R^*)
۳	شبیه‌سازی	(۵و۳و۴و۸و۳و۴و۴)	[۰/۹۶۵۴۳۱و۰/۹۸۲۶۴۶]	[۰/۹۴۸۹۲۷و۰/۹۷۵۱۶۴]	[۰/۰۰۰۰۰۰و۰/۰۳۲۷۲۱]
	DPSO	(۵و۴و۴و۴و۴و۴و۴)	[۰/۹۸۷۸۹۲و۰/۹۹۵۴۳۶]	[۰/۹۸۵۷۸۱و۰/۹۹۴۴۳۶]	[۰/۰۰۰۰۰۰و۰/۰۰۹۵۵۹]
۴	روش ترکیبی پیشنهادی	(۵و۴و۴و۴و۴و۴و۴)	[۰/۹۸۷۸۹۲و۰/۹۹۵۴۳۶]	[۰/۹۸۵۸۷۴و۰/۹۹۴۳۹۲]	[۰/۰۰۰۰۰۰و۰/۰۰۹۴۷۲]
	شبیه‌سازی	(۳و۳و۳و۸و۷و۳و۳)	[۰/۹۵۶۱۲۰و۰/۹۷۷۱۳۸]	[۰/۹۳۸۵۲۴و۰/۹۶۳۵۵۰]	[۰/۰۰۰۰۰۰و۰/۰۳۰۰۲۹]
۵	DPSO	(۴و۴و۴و۴و۴و۴و۴)	[۰/۹۸۵۰۶۵و۰/۹۹۳۴۳۶]	[۰/۹۸۴۰۴۸و۰/۹۹۲۶۱۸]	[۰/۰۰۰۰۰۰و۰/۰۰۹۲۴۹]
	روش ترکیبی پیشنهادی	(۴و۴و۴و۴و۴و۴و۴)	[۰/۹۸۵۰۶۵و۰/۹۹۳۴۳۶]	[۰/۹۸۴۴۲۰و۰/۹۹۲۷۳۶]	[۰/۰۰۰۰۰۰و۰/۰۰۸۸۷۰]
۶	شبیه‌سازی	(۳و۳و۳و۲و۴و۲و۳و۳و۳)	[۰/۰۹۴۱۵و۰/۹۴۱۹۸۴]	[۰/۸۷۷۸۳۹و۰/۹۱۷۹۳۵]	[۰/۰۰۰۰۰۰و۰/۰۵۹۵۳۹]
	DPSO	(۴و۳و۴و۴و۳و۴و۳و۳)	[۰/۹۵۲۳۳۵و۰/۹۷۴۲۴۰]	[۰/۹۴۷۰۹۳و۰/۹۷۰۱۸۵]	[۰/۰۰۰۰۰۰و۰/۰۲۷۷۶۲]
۷	روش ترکیبی پیشنهادی	(۴و۳و۴و۴و۳و۴و۳و۳)	[۰/۹۵۲۳۳۵و۰/۹۷۴۲۴۰]	[۰/۹۵۱۲۱۲۷و۰/۹۷۳۰۲۶]	[۰/۰۰۰۰۰۰و۰/۰۲۳۶۴۱]
	شبیه‌سازی	(۳و۲و۴و۴و۲و۳و۳و۴و۲)	[۰/۸۰۶۱۶۴و۰/۸۷۳۲۷۹]	[۰/۹۸۴۴۲۰و۰/۹۹۲۷۳۶]	[۰/۰۰۴۲۳۶و۰/۰۸۸۰۱۴]
۸	DPSO	(۴و۳و۳و۳و۴و۴و۳و۳)	[۰/۹۴۴۴۹۲و۰/۹۷۱۳۹۳]	[۰/۹۳۵۵۴۸و۰/۹۶۷۵۶۱]	[۰/۰۰۰۰۰۰و۰/۰۳۵۹۹]
	روش ترکیبی پیشنهادی	(۴و۳و۳و۳و۴و۴و۳و۳)	[۰/۹۴۴۴۹۲و۰/۹۷۱۳۹۳]	[۰/۹۳۸۴۰۷و۰/۹۶۹۰۷۴]	[۰/۰۰۰۰۰۰و۰/۰۳۳۹۳۷]

- reliability optimization. *Journal of Global Optimization*, 21: 185–199.
- [6] Zhao, R., Liu, B. (2003). Stochastic programming models for general redundancy-optimization problems. *IEEE Transactions on Reliability R52*, 2: 181–191.
- [7] Ramachandran, V., Sankaranarayanan, V. (1990). Dynamic redundancy allocation using monte-carlo optimization, *microelectro. Rellab*, 30:1131-1136.
- [8] Vinod, V.V., Ghose, S. (1994). Neural network optimization for redundancy allocation *microelectro. Rellab*, 34: 115-12.
- [9] Li, D., Yang, J.-B. (1996). Iterative Parametric Minimax Method for a Class of Composite Optimization Problems, *JOURNAL OF Mathematical analysis and applications*. 198 :64-83.
- [10] Sung, C.S., Cho, Y.K. (2000). Reliability optimization of a series system with multiple-choice and budget constraints. *European Journal of Operational Research*, 127: 159–171.
- [11] Ramirez-Marquez, J.E., Coit, D.W. (2004). A heuristic for solving the redundancy allocation problem for multi-state series-parallel systems. *Reliability Engineering and System Safety*, 83: 341–349.
- [12] Tavakkoli-Moghaddam, R., Safari, J., Sassani, F. (2008). Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. *Reliability Engineering and System Safety*, 93: 550–556.
- [13] Sahoo, L., Bhunia, A.K., Kapur, P.K. (2012). Genetic algorithm based multi-objective reliability optimization in interval environment, *Computers & Industrial Engineering*, 62: 152–160.
- [14] AboueiArdakan, M., ZeinalHamadani, A. (2014). Reliability optimization of series-parallel systems with mixed redundancy strategy in subsystems. *Reliability Engineering and System Safety*, 130: 132-139.
- [15] Roy, P., Mahapatra, B.S., Mahapatra, G.S., Roy, P.K. (2014). Entropy based region reducing genetic algorithm for reliability redundancy allocation in interval environment. *Expert Systems with Applications*, 41: 6147–6160.
- [16] Agarwal, M., Sharma, V.K. (2010). Ant colony approach to constrained redundancy optimization in binary systems. *Applied Mathematical Modelling*, 34: 992–1003.
- [17] Zhao, J.-H., Liu, Z., Dao, M.-T. (2007). Reliability optimization using multiobjective ant colony system approaches. *Reliability Engineering and System Safety*, 92: 109–120.
- [18] Nahas, N., Nourelfath, M., Ait-Kadi, D. (2007). Coupling ant colony and the degraded ceiling algorithm for the redundancy allocation problem of series-parallel systems. *Reliability Engineering and System Safety*, 92: 211–222.



شکل (۶): همگرا شدن الگوریتم به سمت جواب نهایی در مثال ۶

۹- نتیجه‌گیری و پیشنهادها

در این مقاله نویسندگان یک الگوریتم توسعه یافته مبتنی بر ترکیب تکنیک بهینه‌سازی ازدحام ذرات با شبیه‌سازی برای حل مسأله قابلیت اطمینان در شرایط عدم قطعیت با محدودیت‌های تصادفی را پیشنهاد دادند. بدین منظور، مسأله مدل‌سازی شده در [۱] به وسیله الگوریتم پیشنهادی حل و نتایج آن با الگوریتم DPSO، تکنیک شبیه‌سازی و الگوریتم ژنتیک مورد مقایسه قرار گرفت. یافته‌ها نشان از برتری الگوریتم پیشنهادی دارد. نقطه قوت الگوریتم پیشنهادی پایین‌تر بودن انحراف معیار بهترین جواب‌ها در تکرارهای متوالی است. محققان می‌توانند برای تحقیقات آتی مسأله حاضر را به وسیله دیگر روش‌های فرا ابتکاری یا ترکیبی حل کرده و نتایج را با نتایج حاصل از این مقاله مقایسه نمایند.

۱۰- مراجع

- [1] Bhunia, A.K., Sahoo, L., Roy, D. (2010). Reliability stochastic optimization for a series system with interval component reliability via genetic algorithm. *Applied Mathematics and Computation*, 216: 929–939.
- [2] Gupta, R.K., Bhunia, A.K., Roy, D. (2009). A GA based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliabilities of components. *Journal of Computational and Applied Mathematics*, 232: 275–284.
- [3] Kuo, W., Prasad, V.R., Tillman, F.A., Hwuang, C.L. (2001). *Optimal Reliability Design Fundamentals and Application*. Cambridge University Press.
- [4] Sun, X., Duan, D.L. (2002). Optimal condition and branch and bound algorithm for constrained redundancy optimization in series system. *Optimization and Engineering*, 3: 53–65.
- [5] Sun, X.L., Mckinnon, K.I.M., Li, D. (2001). A convexification method for a class of global optimization problems with application to

- [32] Eh, W.C. (2009). A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems. *Expert Systems with Applications* 36: 9192-9200.
- [33] Beji, N., Jarboui, B., Eddaly, M., Chabchoub, H. (2010). A Hybrid Particle Swarm Optimization Algorithm for the Redundancy Allocation Problem. *Journal of Computational Science*, 1: 159-167.
- [34] Safaei, N., Tavakkoli-Moghaddam, R., Kiassat, C. (2012). Annealing-based particle swarm optimization to solve the redundant reliability problem with multiple component choices. *Applied Soft Computing*, 12: 3462-3471.
- [35] Chambari, A., Rahmati, S.H.A., Najafi, A.A., karimi, A. (2012). A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies. *Computers & Industrial Engineering*, 63: 109-119.
- [36] Garg, H., Rani, M., Sharma, S.P., Vishwakarma, Y. (2014). Intuitionistic fuzzy optimization technique for solving multi-objective reliability optimization problems in interval environment. *Expert Systems with Applications*, 41: 3157-3167.
- [37] Figueira, G., Almada-Lobo, B. (2014). Hybrid simulation-optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory*, 46: 118-134.
- [۳۸] بنکس، جری، کارسن، جان. (۱۹۸۴). شبیه‌سازی سیستم‌های گسسته-پیشامد، ترجمه هاشم محلوجی، موسسه انتشارات علمی دانشگاه صنعتی شریف.
- [39] Korn, G.A. (2007). *Advanced dynamic-system simulation: model-replication techniques and monte carlo simulation*, John Wiley & Sons, Inc.
- [40] Robinson, S. (2004). *Simulation: The practice of model development and use*. John Wiley & Sons, Inc.
- [41] Kennedy, J., Eberhart, R.C. (1995), *Particle swarm optimization*. IEEE Intl. Conf. on Neural Networks Perth, Australia, 1942-1948.
- [42] Talbi, E.G. (2009). *Metaheuristics, from design to implementation*, John Wiley & Sons, Inc.
- [43] Reche-Lopez, P., Ruiz-Reyes, N., Galan, G.S., Jurado, F. (2009). Comparison of metaheuristic techniques to determine optimal placement of biomass power plants. *Energy Conversion and Management*, 50: 2020-2028.
- [44] Olariu, S., Zomaya, A.Y. (2006). *Handbook of bioinspired algorithms and applications*. Chapman & Hall/CRC, Taylor & Francis Group.
- [45] Kennedy, J., Eberhart, R. (1995), A new optimizer using particle swarm theory. Sixth international symposium on micro machine and human science, 39-43.
- [46] Kennedy, j. (1997). The particle swann: social adaptation of knowledge. *International Conference on Evolutionary Computation*, Indianapolis, Indiana, 303-308.
- [۱۹] جلالی‌نائینی سیدغلامرضا، احمدی‌زر، فردین. (۱۳۸۶). بهینه‌سازی قابلیت اطمینان در سیستم‌های سری با انتخاب‌های چندگانه و محدودیت بودجه با استفاده از الگوریتم مورچگان، فصلنامه دانش مدیریت، شماره ۷۶ صفحه ۳-۲۲.
- [20] Ahmadizar, F., Soltanpanah, H. (2011). Reliability optimization of a series system with multiple-choice and budget constraints using an efficient ant colony approach. *Expert Systems with Applications*, 38: 3640-3646.
- [21] Hsieh, T.J., Yeh, W.C. (2012). Penalty guided bees search for redundancy allocation problems with a mix of components in series-parallel systems. *Computers & Operations Research*, 39: 2688-2704.
- [22] Garg, H., Sharma, S.P. (2013). Multi-objective reliability-redundancy allocation problem using particle swarm optimization. *Computers & Industrial Engineering*, 64: 247-255.
- [23] Soltani, R., Sadjadi, S.J., Tofigh, A.A. (2014). A model to enhance the reliability of the serial parallel systems with component mixing. *Applied Mathematical Modelling*, 38: 1064-1076.
- [24] Ouzineb, M., Nourelfath, M., Gendreau, M. (2008). Tabu search for the redundancy allocation problem of homogenous series-parallel multi-state systems. *Reliability Engineering and System Safety*, 93: 1257-1272.
- [25] SeyedEsfahani, M., HajianHeidary, M., Jaber, S. (2014). A Simulated Annealing Algorithm for Fuzzy Reliability Optimization Model in Series-Parallel and Standby Systems. *International Journal of Industrial Engineering & production Management*, 24: 413-422.
- [26] Gen, M., Yun, Y. (2006). Soft computing approach for reliability optimization. *Reliability Engineering and System Safety*, 91: 1008-1026.
- [27] Coit, D.W., Smith, A.E. (1996). Solving the redundancy allocation problem using a combined neural network/genetic algorithm approach. *Computers Ops Res*, 23: 515-526.
- [28] He, P., Wu, K., Xu, J., Wen, J., Jiang, Z. (2013). Multilevel redundancy allocation using two dimensional arrays encoding and hybrid genetic algorithm. *Computers & Industrial Engineering*, 64: 69-83.
- [29] Kanagaraj, G., Ponnambalam, S.G., Jawahar, N. (2013). A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems. *Computers & Industrial Engineering*, 66: 1115-1124.
- [30] Ouzineb, M., Nourelfath, M., Gendreau, M. (2010). An efficient heuristic for reliability design optimization problems. *Computers & Operations Research*, 37: 223-235.
- [31] Yeh, W.C. (2014). Orthogonal simplified swarm optimization for the series-parallel redundancy allocation problem with a mix of components. *Knowledge-Based Systems*, 64: 1-12.

series system with m chance constraints. South African Journal of Science, 103: 502-504.

[47] Yadavalli, V.S.S., Malada ,A., Charles, V. (2007). Reliability stochastic optimization for an n -stage

ضمیمه (الف):

جدول (۶): داده‌های مثال ۳

منابع در دسترس											زیرسیستم j ام	
	۸	۷	۶	۵	۴	۳	۲	۱				
l_i	u_i	α_i	[۰/۷۹و۰/۸۵]	[۰/۷۵و۰/۷۹]	[۰/۷۸و۰/۸۳]	[۰/۸۰و۰/۸۵]	[۰/۸۰و۰/۸۶]	[۰/۷۳و۰/۷۸]	[۰/۷۸و۰/۸۱]	[۰/۷۵و۰/۷۸]	r_j	
C_1	۱۵۰	۱۶۰	۰/۱۰	۱/۵	۵	۴/۳	۴	۴/۴	۳/۲	۲/۱	۴/۱	a_{1j}
C_2	۱۹۰	۲۳۰	۰/۱۵	۶	۵	۷	۹	۲	۱	۳	۴	a_{2j}
C_3	۱۱۰	۲۰۰	۰/۱۰	۴	۱	۱	۳	۵	۳	۱	۴	a_{3j}
C_4	۱۷۰	۲۲۰	۰/۱۵	۹	۶	۴	۴	۸	۱	۴	۵	a_{4j}

جدول (۷): داده‌های مثال ۴

منابع در دسترس											زیرسیستم j ام	
	۸	۷	۶	۵	۴	۳	۲	۱				
l_i	u_i	α_i	[۰/۷۲و۰/۷۷]	[۰/۸۳و۰/۸۶]	[۰/۸۰و۰/۸۴]	[۰/۷۸و۰/۸۳]	[۰/۷۹و۰/۸۱]	[۰/۷۳و۰/۷۶]	[۰/۷۵و۰/۷۹]	[۰/۸۰و۰/۸۴]	r_j	
C_1	۱۵۰	۱۶۰	۰/۱۰	۴/۱	۴/۴	۱/۵	۴/۳	۲/۱	۳/۲	۵	۴	a_{1j}
C_2	۱۹۰	۲۳۰	۰/۱۵	۲	۳	۷	۵	۲	۴	۸	۶	a_{2j}
C_3	۱۱۰	۲۰۰	۰/۱۰	۵	۱	۳	۱	۳	۴	۳	۵	a_{3j}
C_4	۱۷۰	۲۲۰	۰/۱۵	۷	۴	۴	۶	۲	۵	۴	۹	a_{4j}

جدول (۸): داده‌های مثال ۵

منابع در دسترس			۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	زیرسیستم j ام	
l_i	u_i	α_i	[۰/۸۵ و ۰/۸۸]	[۰/۹۰ و ۰/۹۳]	[۰/۸۰ و ۰/۸۲]	[۰/۷۵ و ۰/۷۹]	[۰/۷۸ و ۰/۸۳]	[۰/۸۰ و ۰/۸۵]	[۰/۸۳ و ۰/۸۶]	[۰/۷۳ و ۰/۷۸]	[۰/۷۸ و ۰/۸۱]	[۰/۷۴ و ۰/۷۶]	r_j	
c_1	۱۵۰	۱۶۰	۰/۱۰	۴	۳	۲/۱	۵	۴/۳	۴/۱	۴/۴	۳/۲	۳/۳	۱/۵	a_{1j}
c_2	۱۹۰	۲۴۰	۰/۱۵	۲	۱	۵	۳	۱	۶	۹	۷	۵	۴	a_{2j}
c_3	۱۱۰	۱۹۰	۰/۱۰	۵	۸	۱	۱	۳	۴	۳	۱	۵	۴	a_{3j}
c_4	۱۸۰	۲۲۰	۰/۱۵	۸	۷	۶	۴	۱	۹	۴	۴	۸	۵	a_{4j}
c_5	۲۰۰	۲۵۰	۰/۱۰	۲	۲	۲	۳	۳	۳	۹	۶	۳	۲	a_{5j}

جدول (۹): داده‌های مثال ۶

منابع در دسترس			۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	زیرسیستم j ام	
l_i	u_i	α_i	[۰/۸۳ و ۰/۸۷]	[۰/۷۸ و ۰/۸۲]	[۰/۷۹ و ۰/۸۵]	[۰/۷۳ و ۰/۷۶]	[۰/۷۳ و ۰/۷۹]	[۰/۹۰ و ۰/۹۳]	[۰/۸۱ و ۰/۸۳]	[۰/۷۵ و ۰/۸۰]	[۰/۸۵ و ۰/۸۹]	[۰/۷۸ و ۰/۸۳]	r_j	
c_1	۱۵۰	۱۶۰	۰/۱۰	۴/۴	۳/۳	۴/۱	۱/۵	۳/۲	۳	۲/۱	۵	۴	۴/۳	a_{1j}
c_2	۱۹۰	۲۴۰	۰/۱۵	۸	۳	۵	۴	۵	۱	۵	۳	۲	۴	a_{2j}
c_3	۱۱۰	۱۹۰	۰/۱۰	۳	۵	۴	۴	۳	۸	۲	۱	۵	۳	a_{3j}
c_4	۱۸۰	۲۲۰	۰/۱۵	۴	۸	۹	۵	۴	۷	۶	۴	۸	۱	a_{4j}
c_5	۲۰۰	۲۵۰	۰/۱۰	۷	۳	۳	۲	۶	۴	۲	۳	۲	۴	a_{5j}

ضمیمه ب

جدول (۱۰): نمایش جواب در ۱۰ تکرار برای مثال ۶

الگوریتم	مرکز جواب‌ها	جواب‌ها (اعداد فاصله‌ای)	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	میانگین	انحراف معیار
شبیه‌سازی	۰.۸۰۷۳۲۲۵۷۴	[۰.۷۶۴۸۳۴۲۳۲ -۰.۸۴۹۸۱۰۹۲]	۲	۴	۳	۴	۱	۳	۳	۲	۳	۸	[۰.۸۰۶۱۶۴ -۰.۸۷۳۲۷۹۲]	[۰.۰۱۳۳۹۶ -۰.۱۳۰۵۱۸]
	۰.۸۱۱۰۴۰۳۵۴	[۰.۷۷۷۵۵۴۰۱۹ -۰.۸۴۴۵۲۶۶۹]	۲	۳	۳	۲	۲	۵	۲	۳	۲	۹		
	۰.۸۲۴۱۴۹۱۱۸	[۰.۷۹۳۱۶۱۴۹۲ -۰.۸۵۵۱۳۶۷۴]	۷	۴	۴	۲	۱	۵	۳	۳	۲	۳		
	۰.۸۲۸۲۲۵۳۱۹	[۰.۷۹۳۰۰۰۴۴۵ -۰.۸۶۳۴۵۰۱۹]	۳	۴	۷	۶	۳	۳	۲	۲	۲	۲		
	۰.۸۳۶۵۷۲۶۶۸	[۰.۷۹۹۰۶۹۱۷۷ -۰.۸۷۴۰۷۶۱۶]	۴	۳	۲	۶	۲	۸	۳	۲	۲	۲		
	۰.۸۳۷۲۷۲۸۷	[۰.۷۹۸۹۹۵۶۹۲ -۰.۸۷۵۵۵۰۰۵]	۲	۲	۶	۳	۲	۲	۷	۳	۲	۵		
	۰.۸۴۳۱۶۱۳۶۵	[۰.۸۱۷۲۸۹۲۴۹ -۰.۸۶۹۰۳۳۴۸]	۳	۴	۳	۲	۲	۵	۲	۵	۲	۴		
	۰.۸۴۵۵۵۶۲۳۵	[۰.۸۱۵۲۸۱۸۰۲ -۰.۸۷۵۸۳۰۶۷]	۵	۲	۵	۲	۱	۳	۶	۴	۳	۳		
	۰.۸۵۲۳۳۵۹۴۸	[۰.۸۱۵۷۷۵۹۹ -۰.۸۸۸۶۹۵۹۱]	۶	۲	۸	۳	۲	۳	۳	۲	۲	۲		
	۰.۹۱۱۶۷۸۰۵۷	[۰.۸۸۶۶۷۴۸۰۵ -۰.۹۳۶۶۸۱۳۱]	۳	۲	۴	۴	۲	۳	۳	۴	۵	۲		
DPSO	۰.۹۵۰۴۲۷۱۴۹	[۰.۹۳۵۱۵۱۲۶ -۰.۹۶۵۷۰۳۰۴]	۴	۳	۴	۴	۲	۴	۳	۳	۳	۴	[۰.۹۳۵۵۴۸ -۰.۹۶۷۵۶۱۸۴]	[۰.۰۰۰۰۰۰۰ -۰.۰۳۵۹۹]
	۰.۹۳۹۸۴۳۵۱۷	[۰.۹۲۲۲۶۴۰۳۴ -۰.۹۵۷۴۲۳]	۵	۲	۶	۴	۲	۴	۳	۳	۳	۴		
	۰.۹۵۱۶۱۵۳۸۹	[۰.۹۳۴۷۵۸۷۴۶ -۰.۹۶۸۴۷۲۰۳]	۴	۳	۶	۴	۲	۳	۴	۳	۳	۳		
	۰.۹۵۶۹۴۲۸۰۵	[۰.۹۴۲۴۹۲۲۹ -۰.۹۷۱۳۹۳۳۲]	۴	۳	۵	۳	۲	۴	۴	۳	۳	۳		
	۰.۹۵۰۱۵۳۷۹۳	[۰.۹۳۴۹۶۷۱۵۷ -۰.۹۶۵۳۴۰۴۳]	۴	۳	۴	۴	۲	۵	۳	۳	۳	۳		
	۰.۹۵۰۰۴۵۱۴۱	[۰.۹۳۳۶۹۵۷۳۳ -۰.۹۶۶۳۹۴۵۵]	۴	۲	۴	۴	۲	۴	۴	۴	۳	۳		
	۰.۹۵۶۹۴۲۸۰۵	[۰.۹۴۲۴۹۲۲۹ -۰.۹۷۱۳۹۳۳۲]	۴	۳	۵	۳	۲	۴	۴	۳	۳	۳		
	۰.۹۵۵۶۳۳۹۸۸	[۰.۹۳۹۸۷۴۶۵۷ -۰.۹۷۱۳۹۳۳۲]	۳	۳	۵	۴	۲	۴	۴	۳	۳	۳		
	۰.۹۵۵۶۳۳۹۸۸	[۰.۹۳۹۸۷۴۶۵۷ -۰.۹۷۱۳۹۳۳۲]	۳	۳	۵	۴	۲	۴	۴	۳	۳	۳		
	۰.۹۴۸۳۱۱۳۴۹	[۰.۹۲۹۹۱۰۶۲۱ -۰.۹۶۶۷۱۲۰۸]	۳	۳	۵	۴	۲	۳	۵	۳	۳	۳		
پیشنهادی	۰.۹۵۶۹۴۲۸۰۵	[۰.۹۴۲۴۹۲۲۹ -۰.۹۷۱۳۹۳۳۲]	۴	۳	۵	۳	۲	۴	۴	۳	۳	۳	[۰.۹۳۸۴۰۷ -۰.۹۶۹۰۷۴]	[۰.۰۰۰۰۰۰۰ -۰.۰۳۳۹۳۷]
	۰.۹۵۶۹۴۲۸۰۵	[۰.۹۴۲۴۹۲۲۹ -۰.۹۷۱۳۹۳۳۲]	۴	۳	۵	۳	۲	۴	۴	۳	۳	۳		
	۰.۹۵۶۹۴۲۸۰۵	[۰.۹۴۲۴۹۲۲۹ -۰.۹۷۱۳۹۳۳۲]	۴	۳	۵	۳	۲	۴	۴	۳	۳	۳		
	۰.۹۵۶۹۴۲۸۰۵	[۰.۹۴۲۴۹۲۲۹ -۰.۹۷۱۳۹۳۳۲]	۴	۳	۵	۳	۲	۴	۴	۳	۳	۳		
	۰.۹۵۶۹۴۲۸۰۵	[۰.۹۴۲۴۹۲۲۹ -۰.۹۷۱۳۹۳۳۲]	۴	۳	۵	۳	۲	۴	۴	۳	۳	۳		
	۰.۹۵۳۶۰۴۷۸۶	[۰.۹۳۸۲۹۱۴۴۷ -۰.۹۶۸۹۱۸۱۲]	۳	۳	۵	۳	۲	۵	۴	۳	۳	۳		
	۰.۹۵۰۰۴۵۱۴۱	[۰.۹۳۳۶۹۵۷۳۳ -۰.۹۶۶۳۹۴۵۵]	۴	۲	۴	۴	۲	۴	۴	۴	۳	۳		
	۰.۹۴۷۰۰۵۶۵۱	[۰.۹۲۹۷۱۷۱۳۱ -۰.۹۶۴۲۹۴۱۷]	۳	۲	۴	۴	۲	۴	۴	۴	۳	۴		
	۰.۹۵۱۶۱۵۳۸۹	[۰.۹۳۴۷۵۸۷۴۶ -۰.۹۶۸۴۷۲۰۳]	۴	۳	۶	۴	۲	۳	۴	۳	۳	۳		
	۰.۹۵۰۴۲۷۱۴۹	[۰.۹۳۵۱۵۱۲۶ -۰.۹۶۵۷۰۳۰۴]	۴	۳	۴	۴	۲	۴	۳	۳	۳	۴		



Development of a method based on combination of simulation and Particle Swarm Optimization to solve reliability problem in uncertainty conditions

F. Jolai*, M.A. Zare, S. Jooybar

Department of Industrial Engineering, Faculty of Engineering, University of Tehran, Tehran

ARTICLE INFO

Article history:

Received 1 November 2016

Accepted 12 April 2015

Keywords:

Reliability problem

Simulation

Particle Swarm Optimization

Uncertainty

ABSTRACT

Nowadays, reliability problem is one of the most important issues at early stages of planning, designing and control of production systems. In this paper, the authors try to develop a hybrid simulation-optimization method in order to solve reliability problem for series-parallel system in uncertainty condition with stochastic constraints. We propose a method in which first, a population of fairly good solutions is generated by simulation technique and then, these solutions are improved by Particle Swarm Optimization (PSO). To assess the efficiency of the proposed algorithm, the results of this research were compared with previous similar studies. The results show that the proposed algorithm is superior. The strength of the proposed algorithm is higher average and lower standard deviation of solutions obtained in successive iteration.

* Corresponding author. Fariborz Jolai

Tel.: 021-88021067; E-mail addresses: fjolai@ut.ac.ir