مهندسی عمران شریف (بهار ۱۳۹۷)
دوره‌ی ۲ - ۳۴، شماره‌ی ۱/۱، ص. ۱۱۵-۱۲۳، (یادداشت فنی)

Research Note

ارزیابی الگوریتم بهینه‌یابی نیروی مرکزی در بهره‌برداری از سیستم‌های چندمخزنه و معرفی الگوریتم بهینه‌یابی نیروی مرکزی تکرارشونده

حسن ترابی‌بوده * **(استادیار)**
برستو همه‌زاده (دانشجوی کارشناسی ارشد)
حجت‌اله یونسی (استادیار)
امیرحمزه حقی آبی (دانشیار)
دانشکده‌ی مهندسی آب، دانشگاه لرستان

در نوشتار حاضر، الگوریتم بهینه‌یابی نیروی مرکزی (CFO) به منظور بهره‌برداری بهینه از سیستم‌های چندمخزنه ارزیابی شده است. به دلیل ضعف الگوریتم مذکور در حل مسائل بهره‌برداری، تغییراتی در آن ایجاد و الگوریتم بهینه‌یابی نیروی مرکزی تکرار شونده (RCFO) پیشنهاد شده است. در ابتدا، یک مسئله‌ی ۴ مخزنه بهینه‌سازی شد. تابع هدف مسئله با استفاده از روش‌های CFO و RCFO به ترتیب ۰/۴۵ و ۰/۱ درصد اختلاف با بهینه‌ی مطلق دارند. پس از موفقیت الگوریتم RCFO در حل سیستم ۴ مخزنه، یک سیستم ۱۰ مخزنه انتخاب شد. تابع هدف به‌دست آمده از الگوریتم‌های CFO و RCFO به ترتیب ۷/۱ و ۰/۱۳ درصد اختلاف با بهینه‌ی مطلق دارند. باتوجه به موفقیت الگوریتم RCFO در حل سیستم ۴ و ۱۰ مخزنه می‌توان نتیجه گرفت که الگوریتم RCFO، پتانسیل مناسبی جهت کاربرد در حل مسائل پیچیده‌ی بهره‌برداری از سیستم‌های چندمخزنه‌ی واقعی دارد.

واژگان کلیدی: الگوریتم بهینه‌یابی نیروی مرکزی، الگوریتم فراکاوشی، بهره‌برداری چندمخزنه، مدل برنامه‌ریزی خطی.

torabi1976@gmail.com
swallow_kaspian@yahoo.com
yonesi.h@lu.ac.ir
haghiabi@yahoo.com

۱. مقدمه

افزایش روزافزون جمعیت و محدودیت منابع آب‌های سطحی، ضرورت مدیریت صحیح از مخازن سدها را ایجاب می‌کند. مطالعات زیادی در زمینه‌ی بهینه‌سازی برداشت از سیستم‌های تک‌مخزنه انجام شده است.[۱و۲] روش‌های بهینه‌سازی مختلفی برای بهره‌برداری از مخازن معرفی شده است که می‌توان آن‌ها را به این صورت دسته‌بندی کرد:[۳] ۱. روش‌های مبتنی بر برنامه‌ریزی خطی (LP)[۱]؛ ۲. برنامه‌ریزی غیرخطی (NLP)[۲]؛ ۳. روش‌های مبتنی بر برنامه‌ریزی پویا (DP)[۳] و ۴. الگوریتم‌های فراکاوشی. روش LP یکی از روش‌های پرکاربرد جهت بهینه‌سازی بهره‌برداری از مخازن است. این روش تاکنون در بهینه‌سازی بهره‌برداری از مخازن با اهداف مختلف استفاده شده است؛[۴] اما استفاده از روش LP با محدودیت‌هایی روبه‌روست و فقط می‌توان آن را برای مسائلی که تابع هدف و قیود خطی باشند، به‌کار برد. یکی دیگر از روش‌های مورد استفاده در بهره‌برداری از مخازن، استفاده از مدل NLP است و زمانی استفاده می‌شود که تابع هدف و قیود، غیرخطی باشند.[۵] در روش DP، مسئله‌ی بهینه‌سازی و متغیرهای آن به چند مسئله‌ی بهینه‌سازی کوچک‌تر تقسیم می‌شود که حل مرحله‌به‌مرحله‌یی آن منجر به حل مسئله‌ی اصلی خواهد شد.[۶] به الگوریتم‌های فراکاوشی نیز در سال‌های اخیر توجه شده است که از میان آنها می‌توان به الگوریتم مورچگان[۴] (ACO)،[۷] الگوریتم ژنتیک[۵] (GA)،[۸] بهینه‌یابی جفت‌گیری زنبورعسل[۶] (HBMO)،[۹] بهینه‌سازی ازدحام ذرات[۷] (PSO)[۱۰] و جست‌وجوی هارمونی(HSA)،[۱۱] اشاره کرد.

علاوه بر روش‌های مذکور، در سال ۲۰۰۷، الگوریتم بهینه‌یابی نیروی مرکزی[۸] (CFO) ارائه شده،[۱۲] و از آن برای حل ۲۳ تابع شناخته شده استفاده شده و نتایج رضایت‌بخشی در مقایسه با سایر الگوریتم‌های فراکاوشی به‌دست آمده است.[۱۳] روش CFO برای بهبود پهنای باند در الکترونیک، طراحی آنتن مایکرواستریپ و بهینه‌سازی شبکه‌ی آب آشامیدنی استفاده شده است،[۱۴-۱۶] و نتایج نشان داده است که روش CFO قادر به حل مسائل پیچیده‌ی بهینه‌سازی است.

بهینه‌سازی سیستم‌های چندمخزنه، پیچیدگی‌های فراوانی دارد و افزایش تعداد

* نویسنده مسئول
تاریخ: دریافت ۱۳۹۴/۱۰/۵، اصلاحیه ۱۳۹۵/۳/۱۲، پذیرش ۱۳۹۵/۵/۱۶.
DOI:10.24200/J30.2018.1331

جدول ۱. جریان ورودی به مخازن ۱ و ۲.

دوره	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
IN_1	۰/۵	۱	۲	۳	۳/۵	۲/۵	۲	۱/۲۵	۱/۲۵	۰/۷۵	۱/۷۵	۱
IN_2	۰/۴	۰/۷	۲	۲	۴	۳/۵	۳	۲/۵	۱/۳	۱/۲	۱	۰/۷

مخازن نیز به پیچیدگی‌های این قبیل مسائل می‌افزاید. در سال‌های اخیر، به بهینه‌سازی سیستم‌های چندمخزنه‌ی منابع آب توجه زیادی شده است. الگوریتم GA توسعه یافته برای سیاست بهره‌برداری از سیستم‌های چندمخزنه استفاده و پس از مقایسه‌ی نتایج آن با روش DDDP، برتری GA مشخص شد.[۱۷] به منظور کمینه‌سازی کمبود آب کشاورزی سدی واقع در مالزی، دو الگوریتم کلونی زنبورعسل[۹] (ABC) و جست‌وجوی گرانشی[۱۰] (GSA) مقایسه شد و نتایج نشان داد که الگوریتم ABC با داشتن ویژگی‌هایی، از جمله: هم‌گرایی سریع‌تر، اطمینان‌پذیری بیشتر و آسیب‌پذیری کمتر برای بهره‌برداری از مخازن مناسب‌تر است.[۱۸] در زمینه‌ی بهینه‌سازی سیستم‌های چندمخزنه (۲۰۰۴) نیز پیشنهاد شد که برای مدل‌سازی بهتر منابع آب از روش‌های رایانه‌یی استفاده شود[۱۹] همچنین الگوریتم ACO (۲۰۰۷) برای بهره‌برداری از یک سیستم ۴ مخزنه در فضای تصمیم گسسته به‌کار گرفته شد که توانست با دقت بیشتر و در زمان محاسبات کمتری نسبت به GA، مسئله را بهینه‌سازی کند.[۲۰]

الگوریتم HSA (۲۰۱۳) نیز برای بهره‌برداری از سیستم چندمخزنه در محیط گسسته استفاده شد و پس از اجرای آن در دو محیط ویژوال بیسیک و متلب، نتایج مشابهی به‌دست آمد و نشان از بالا بردن سرعت هم‌گرایی و اطمینان پذیری بالای الگوریتم مذکور داشت.[۲۱] در مطالعه‌یی دیگر (۲۰۱۳)، بهینه‌سازی دو سیستم ۴ و ۱۰ مخزنه با استفاده از ۳ روش مختلف بهینه‌سازی ازدحام ذرات مقید بررسی شد و روش‌های پیشنهادی در مقایسه با الگوریتم PSO نتایج بهتری را ارائه کردند و بهترین نتیجه برای حالتی که رهاسازی به‌عنوان متغیر تصمیم انتخاب شد، به‌دست آمد. همچنین روش‌های مذکور نسبت به جمعیت اولیه و اندازه‌ی آن حساس نبودند و سرعت هم‌گرایی بالایی داشتند.[۲۲] مطالعه‌ی دیگری در همان سال، با استفاده از ACOدر سه حالت مقید انجام شد و نتایج نشان از برتری روش‌های مقید داشت.[۲۳]

تاکنون گزارشی ازکاربرد الگوریتم CFO در بهره‌برداری از سیستم‌های چندمخزنه ارائه نشده است. در نوشتار حاضر، پس از تشریح الگوریتم CFO از آن به منظور بهره‌برداری بهینه از یک سیستم ۴ مخزنه استفاده شده است. در ادامه، تغییراتی در آن ایجاد شده و الگوریتم بهینه‌یابی نیروی مرکزی تکرار شونده[۱۱] (RCFO) معرفی و پس از آن یک سیستم ۱۰ مخزنه با استفاده از هر دو روش، بهینه‌سازی شده است.

۲. مواد و روش‌ها

در نوشتار حاضر، ابتدا مسئله‌ی ۴ مخزنه و سپس ۱۰ مخزنه به همراه ورودی‌ها و روابط موردنیاز ارائه شده است. در هر دو مسئله، رهاسازی از مخازن به‌عنوان متغیر تصمیم در نظر گرفته شده است؛ پس از آن الگوریتم CFO به همراه تمامی روابط موردنیاز، مراحل جست‌وجو و نمودار الگوریتم ارائه شده است. در ادامه، پس از بررسی CFO و پیدا کردن نقاط ضعف آن، تغییراتی در آن ایجاد و الگوریتم RCFO معرفی شده است. برای پیدا کردن پارامترهای بهینه از آنالیز حساسیت استفاده شده است. نتایج بهینه‌ی مطلق تابع هدف برای هر دو مسئله که ماهیتی خطی داشتند، با استفاده از لینگو که یک مدل برنامه‌ریزی خطی است، به‌دست آمده است.

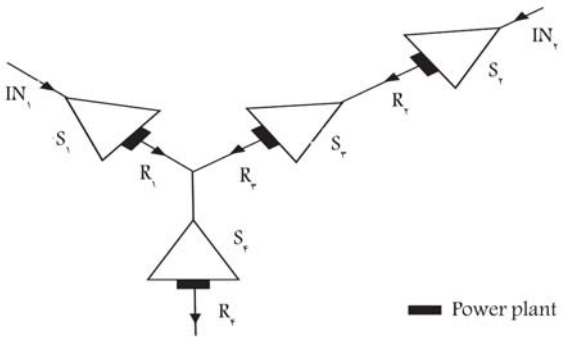
۱.۲. شرح مسئله‌ی ۴ مخزنه

مسئله‌ی ۴ مخزنه، اولین بار در سال ۱۹۷۴ مطرح شد؛[۲۴] که ایده‌ی اصلی آن از سیستم ۴ مخزنه‌ی معرفی شده در پژوهشی در سال ۱۹۷۱ اخذ شده است (شکل ۱).[۲۵] جریان ورودی IN_1 و IN_2 فقط به دو مخزن وارد می‌شود و مقادیر آن در جدول ۱ نشان داده شده است. حجم اولیه و نهایی برای مخازن ۱ تا ۳ برابر ۶ و برای مخزن ۴، برابر با ۸ بوده است. کمینه‌ی رهاسازی برای همه‌ی مخازن، ۰/۰۰۵ و بیشینه‌ی آن برای مخزن ۱ برابر با ۴، برای مخزن‌های ۲ و ۳ برابر با ۴/۵ و برای مخزن ۴، ۸ بوده است (جدول ۲). کمینه‌ی حجم مخزن برای تمامی مخازن در همه‌ی ۱۲ دوره برابر با ۱ بوده است. بیشینه‌ی حجم مخازن نیز در جدول ۲ نشان داده شده است.

رابطه‌ی پیوستگی در هر مخزن (i) و در هر دوره (j) به‌صورت رابطه‌ی ۱ تعریف شده است:

$$S_{(i,j+1)} = S_{(i,j)} + I_{(i,j)} + M.R_{(i,j)} \quad i = 1,...,4; j = 1,...,12 \quad (1)$$

که در آن، S حجم مخزن، I جریان ورودی، R رهاسازی ماهانه و M ماتریس



شکل ۱. سیستم ۴ مخزنه معرفی شده در سال ۱۹۷۴.[۲۴]

جدول ۲. بیشینه‌ی حجم مخازن.

دوره	مخزن ۱	مخزن ۲	مخزن ۳	مخزن ۴
۲	۱۲	۱۷	۸	۱۵
۳	۱۲	۱۵	۸	۱۵
۴	۱۰	۱۵	۸	۱۵
۵	۹	۱۵	۸	۱۵
۶	۸	۱۲	۸	۱۵
۷	۸	۱۲	۸	۱۵
۸	۹	۱۵	۸	۱۵
۹	۱۰	۱۷	۸	۱۷
۱۰	۱۰	۱۸	۸	۱۵
۱۱	۱۲	۱۸	۸	۱۵
۱۲	۱۲	۱۸	۸	۱۵

روابط مخازن به‌صورت رابطه‌ی ۲ است:

$$M = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix} \quad (2)$$

تابع هدف در رابطه‌ی ۳ نشان داده شده است که هدف از حل آن بیشینه‌سازی مقدار تابع براساس رهاسازی بهینه است.

$$\text{Max} \quad I = \sum_{i=1}^{4} \sum_{j=1}^{12} b_{(i,j)}.R_{(i,j)} \quad (3)$$

که در آن، مقادیر ضرایب سود $b_{(i,j)}$ در سال ۱۹۷۴ ارائه شده است.[۲۴] برای هر یک از مخازن، قیود مربوط به رهاسازی، حجم مخزن (روابط ۴ و ۵) و حجم نهایی تعریف شده است:

$$S^{\min}_{(i,j)} \le S_{(i,j)} \le S^{\max}_{(i,j)}; \quad j = 2, 3, ..., 12 \quad (4)$$

$$S^{\min}_{(i,j)} \le S_{(i,j)} \le S^{\max}_{(i,j)}; \quad j = 1, 2, ..., 12 \quad (5)$$

در آنها، $S^{\min}$ و $S^{\max}$ به ترتیب حجم مخزن کمینه و بیشینه؛ و $R^{\min}$ و $R^{\max}$ به ترتیب رهاسازی کمینه و بیشینه از مخزن هستند.

برای جلوگیری از ورود جواب‌های نشدنی به مسئله، مطالعات گسترده‌یی انجام شده و یکی از روش‌های مذکور، استفاده از توابع جریمه است.[۲۶] توابع جریمه به‌صورت رابطه‌ی ۶ تعریف می‌شود:

$$P = C.\Delta^n \quad (6)$$

که در آن، P مقدار جریمه، Δ میزان تخطی از قید و C و n انتخابی هستند. در پژوهش حاضر، برای هر دو مسئله، ضریب C برابر با ۴۰ در نظر گرفته شده است.[۱۷] توان نیز به منظور جلوگیری از کوچک شدن جریمه‌های کمتر از ۱، همان مقدار ۱ انتخاب شد. رابطه‌ی ۷، تابع جریمه را برای حجم مخزن نشان می‌دهد. قیود رهاسازی نیز به همین شیوه تعریف شده‌اند. با درنظرگرفتن ذخیره‌ی هدف f برای دوره‌ی آخر در هر مخزن i، تابع جریمه مربوط به حجم نهایی به‌صورت رابطه‌ی ۸ بیان می‌شود:

$$P = \begin{cases} c_0(1 - S_i/S^{\min})^n & if \;\; S_i < S^{\min} \\ c_0(S_i/S^{\max} - 1)^n & if \;\; S_i > S^{\max} \\ 0 & if \;\; S_i \ge S^{\min} \;\&\; S_i \le S^{\max} \end{cases} \quad (7)$$

$$P_{13} = \begin{cases} c_0(1 - S_{(i,13)}/f_{(i)})^n & if \;\; S_{(i,13)} < f_{(i)} \\ c_0(S_{(i,13)}/f_i - 1)^n & if \;\; S_{(i,13)} > f_{(i)} \\ 0 & else \end{cases} \quad (8)$$

که در آنها، f حجم نهایی برای هر یک از مخازن در انتهای دوره است که این مقدار برای مخازن ۱ تا ۳ برابر با ۶ و برای مخزن ۴ برابر ۸ در نظر گرفته شده است.

۲.۲. شرح مسئله‌ی ۱۰ مخزنه

مسئله‌ی ۱۰ مخزنه که از ۱۰ مخزن تشکیل شده است (شکل ۲)، اولین بار در سال ۱۹۷۹ مطرح شد.[۲۷] کل دوره‌ی زمانی در مسئله‌ی حاضر، ۱۲ دوره‌ی ۲ ساعته و متغیرهای تصمیم معادل ۱۲۰ بوده است. هدف از بهره‌برداری، بیشینه‌سازی تولید انرژی در کل دوره است. جریان رودخانه فقط به مخازن ۳، ۲، ۱، ۵، ۶ و ۸ وارد می‌شود. قیود مسئله، شامل: حجم و رهاسازی کمینه و بیشینه برای هر یک از مخازن است. یک قید دیگر مربوط به حجم نهایی وجود دارد، که باید در آن حجم نهایی مخزن با حجم اولیه برابر باشد.

رابطه‌ی پیوستگی در هر مخزن (i) و در هر دوره (j) به‌صورت رابطه‌ی ۹ تعریف شده است:

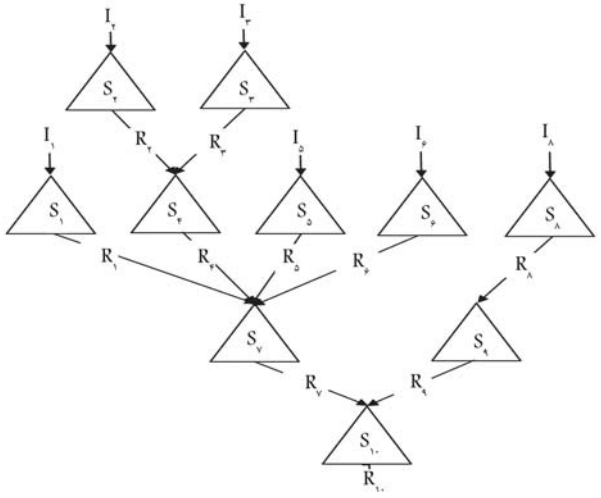
$$S_{(i,j+1)} = S_{(i,j)} + I_{(i,j)} + M.R_{(i,j)} \;\; i = 1, ..., 10; j = 1, ..., 12 \quad (9)$$

که در آن، S حجم مخزن، I جریان ورودی، R رهاسازی ماهانه و M ماتریس روابط مخازن است که از رابطه‌های ۱۰ و ۱۱ به‌دست می‌آیند:

$$M = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix} \quad (10)$$

$$\text{Max } I = \sum_{i=1}^{10} \sum_{j=1}^{12} b_{(i,j)}.R_{(i,j)} \quad (11)$$

که در آن، تابع هدف b ضریب سود حاصل از هر یک از مخازن است. جزئیات بیشتر در منبع اصلی ارائه شده است.[۲۷] در مسئله‌ی حاضر، نیز توابع جریمه و ضرایب آن مشابه سیستم ۴ مخزنه تعریف شده‌اند.



شکل ۲. سیستم ۱۰ مخزنه‌ی طراحی شده در سال ۱۹۷۹.[۲۷]

۳.۲. الگوریتم بهینه‌یابی نیروی مرکزی

الگوریتم بهینه‌یابی نیروی مرکزی اولین بار در سال ۲۰۰۷ مطرح شد.[۱۲] فضای تصمیم در CFO به‌صورت $x_i^{\min} \le x_i \le x_i^{\max}$ تعریف شده است که در آن، x_i متغیرهای تصمیم و $i = ۲, ۱, ..., N_d$ تعداد بُعد است. الگوریتم CFO با پرواز گروهی از پروب‌ها که مسیرهایشان توسط دو معادله‌ی جبری حرکت (موقعیت و شتاب) تعیین می‌شود، جست‌وجو را انجام می‌دهد. بردار موقعیت یک پروب به‌صورت رابطه‌ی ۱۲ است:

$$\vec{R}_j^p = \sum_{m=۱}^{N_d} X_m^{p,j} \hat{e}_m \qquad (۱۲)$$

که در آن $X_m^{p,j}$ ، مختصات mاُمین بُعد از پروب درگام زمانی j و $\hat{e}_m$ بردار یکه در طول محور x_m است. در فضای تصمیم، بردارهای موقعیت پروب تغییر می‌کنند، تا زمانی که تمام پروب‌ها در اطراف بزرگ‌ترین جرم (پروب) مستقر شوند. هر پروب با بردار موقعیت $\vec{R}_{j-۱}^p \in R^{N_d}$ درگام زمانی $j-۱$ بردار شتاب $\vec{a}_{j-۱}^p$ را تحت تأثیر نیروهای گرانشی ایجاد شده به وسیله سایر پروب‌ها تجربه می‌کند. $\vec{a}_{j-۱}^p$ به‌صورت رابطه‌ی ۱۳ است:

$$\vec{a}_{j-۱}^p = G \sum_{\substack{k=۱ \\ k \neq p}}^{N_p} U(M_{j-۱}^k - M_{j-۱}^p).(M_{j-۱}^k - M_{j-۱}^p)^\alpha \frac{(\vec{R}_{j-۱}^k - \vec{R}_{j-۱}^p)}{||\vec{R}_{j-۱}^k - \vec{R}_{j-۱}^p||^\beta} \qquad (۱۳)$$

که در آن، N_p تعداد پروب‌ها، p شماره‌ی پروب، j گام زمانی محاسبات؛ α، β و G ثابت‌های CFO هستند. $M_{j-۱}^p = f(x_۱^{p,j-۱}, x_۲^{p,j-۱}, ..., x_{N_d}^{p,j-۱})$ مقدار تابع هدف بر حسب پروب p درگام زمانی $j-۱$ است و U تابع پله‌یی یکه است که به‌صورت رابطه‌ی ۱۴ تعریف می‌شود:

$$U(x) = \begin{cases} ۱ & if \;\; x \ge ۰ \\ ۰ & otherwise \end{cases} \qquad (۱۴)$$

و $||\vec{R}_{j-۱}^k - \vec{R}_{j-۱}^p||^\beta$ درواقع اندازه‌ی فاصله‌ی بین موقعیت‌های پروب p و k است که به صورت رابطه‌ی ۱۵ محاسبه می‌شود:

$$||\vec{R}_{j-۱}^k - \vec{R}_{j-۱}^p|| = \sqrt{\sum_{m=۱}^{N_d} (R_{j-۱}^{k,m} - R_{j-۱}^{p,m})^۲} \qquad (۱۵)$$

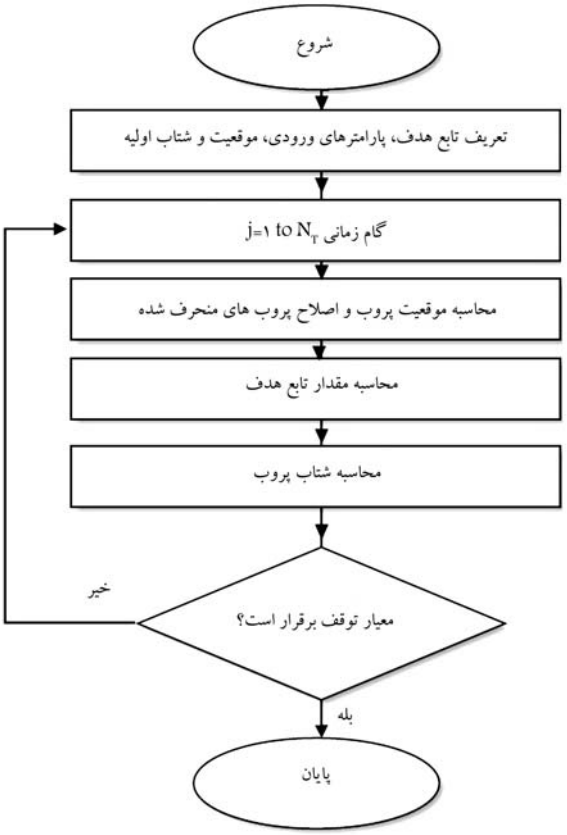
اگر تعداد کل گام‌های زمانی برابر با Nt باشد موقعیت جدید پروب به‌صورت رابطه‌ی ۱۶ تعریف می‌شود:

$$\vec{R}_j^p = \vec{R}_{j-۱}^p + \frac{۱}{۲} \vec{a}_{j-۱}^p \Delta t^۲, \qquad j \ge ۱ \qquad (۱۶)$$

که در آن، Δt فاصله‌ی زمانی گام‌هاست و مقدار آن ۱ درنظرگرفته شده است. پروب‌ها ممکن است به موقعیتی خارج از فضای تصمیم جابه‌جا شوند. رابطه‌های ۱۷ و ۱۸ برای بازسازی پروب‌ها به‌کار می‌روند:[۱۳]

$$if \; \vec{R}_{j,i}^p < x_i^{\min} \; then \vec{R}_{j,i}^p = \max(x_i^{\min} + F_{rep}(\vec{R}_{j-۱,i}^p - x_i^{\min}), x_i^{\min}) \qquad (۱۷)$$

$$if \; \vec{R}_{j,i}^p > x_i^{\max} \; then \vec{R}_{j,i}^p = \min(x_i^{\max} - F_{rep}(x_i^{\max} - \vec{R}_{j-۱,i}^p) \; x_i^{\max}) \qquad (۱۸)$$



شکل ۳. الگوریتم CFO.

که در آنها، $x_i^{\min}$ و $x_i^{\max}$ به ترتیب کران پایین و کران بالای متغیرهای تصمیم و F_{rep} عامل تغییر موقعیت پروب هستند. عامل مهم دیگر در CFO، توزیع مکان اولیه‌ی پروب‌هاست که می‌تواند به‌صورت قطری یا سطری در امتداد بعدها چیده شود، با توجه به حساسیت الگوریتم CFO به پروب‌های اولیه، بهتر است بخشی از پروب‌ها به‌صورت قطری و بخشی دیگر به‌صورت سطری انتخاب شوند.[۲۸] شکل ۳، فلوچارت الگوریتم CFO را نشان می‌دهد.

۴.۲. الگوریتم بهینه‌یابی نیروی مرکزی تکرارشونده

CFO در توابع مختلف برتری خود را در مقایسه با سایر الگوریتم‌ها مانند GA یا PSO نشان داده است.[۱۳] یکی از ویژگی‌های CFO، بالا بودن سرعت همگرایی آن در شروع جست‌وجو است؛ اما با افزایش تکرارها، سرعت همگرایی کاهش پیدا می‌کند. به همین منظور در برخی از مراحل آن تغییراتی به این شرح اعمال شده است:

۱.۴.۲. نرمال‌سازی شتاب

شتاب عامل اصلی حرکت در CFO، از پارامترهای α، β و G تشکیل شده است. در پژوهشی در سال ۲۰۱۰، نشان داده شده است که مقادیر α، β و G برابر با ۲ و F_{rep} برابر با ۰٫۵ بهترین عملکرد را در اغلب موارد دارند.[۱۳] در مسائل مقید (مانند مسئله‌ی حاضر) وجود تابع جریمه باعث افزایش تفاوت وزن بین پروب‌ها و در نتیجه ایجاد شتاب‌های متغیر خواهد شد. شتاب‌های بزرگ باعث خروج پروب‌ها از فضای تصمیم می‌شوند که می‌توانند به کمک روابط ۱۷ و ۱۸ بازگردانده شوند؛ اما فرایند مذکور در حل مسائل پیچیده و مقید بی‌نتیجه خواهد بود، زیرا الگوریتم همواره در حال بازگرداندن پروب‌ها، بدون یافتن نقطه‌ی بهینه است. پیشنهاد می‌شود

ضرایب α، β و G به‌گونه‌یی انتخاب شوند که مقادیر شتاب متناسب با محدوده‌ی فضای تصمیم باشند. برای کاهش حساسیت شتاب به ضرایب می‌توان از نرمال‌سازی شتاب استفاده کرد. در مسئله‌ی حاضر به منظور محدود کردن شتاب‌ها، رابطه‌ی نرمال‌سازی به‌صورت رابطه‌ی ۱۹ در نظر گرفته شده است:

$$a_{norm} = \frac{a_{j-1}}{(a_{\max} - a_{\min})} \times w \qquad (19)$$

که در آن، $a_{\max}$ و $a_{\min}$ بیشترین و کمترین شتاب تجربه شده توسط پروب‌هاست. اگر اختلاف بین $a_{\max}$ و $a_{\min}$ زیاد باشد، شتاب‌های نرمال شده‌ی بسیار کوچک به‌دست می‌آیند. به‌همین منظور لازم است که معادله‌ی نرمال‌سازی شتاب در ضریب w ضرب شود که مقدار آن متناسب با نوع مسئله و مقادیر شتاب متفاوت خواهد بود. در تکرارهای مذکور، ممکن است شتاب همه‌ی پروب‌ها صفر شود و همگی در یک بهینه‌ی محلی قرار گیرند؛ اگر اختلاف بین $a_{\max}$ و $a_{\min}$ برابر با صفر شد، می‌توان رابطه‌ی ۲۰ را به‌کار برد:

$$a_{j-1} = a_{j-1} + rand \qquad (20)$$

که در آن، a_{j-1} شتاب جدید است که از شتاب قبلی به اضافه‌ی یک عدد تصادفی برای تولید شتاب‌های متغیر تشکیل شده است.

۲.۴.۲. استفاده از جهش

الگوریتم ژنتیک یکی از بهترین الگوریتم‌ها در سال‌های اخیر است که یکی از دلایل موفقیت آن استفاده از جهش است.[۲۹] یکی دیگر از ضعف‌های الگوریتم CFO، گرفتار شدن در بهینه‌ی موضعی است که برای خروج پروب‌ها از آن می‌توان از جهش استفاده کرد. در رابطه‌ی ارائه شده در نوشتار حاضر، فقط یک بُعد از پروب تغییر می‌کند. ابتدا شماره‌ی پروب (M_y) و شماره‌ی بُعد (M_x) به‌صورت تصادفی انتخاب و سپس یک مقدار به موقعیت قبلی اضافه می‌شود. تعداد جهش از رابطه‌ی ۲۱ قابل محاسبه است که در آن mutation درصد جهش است:

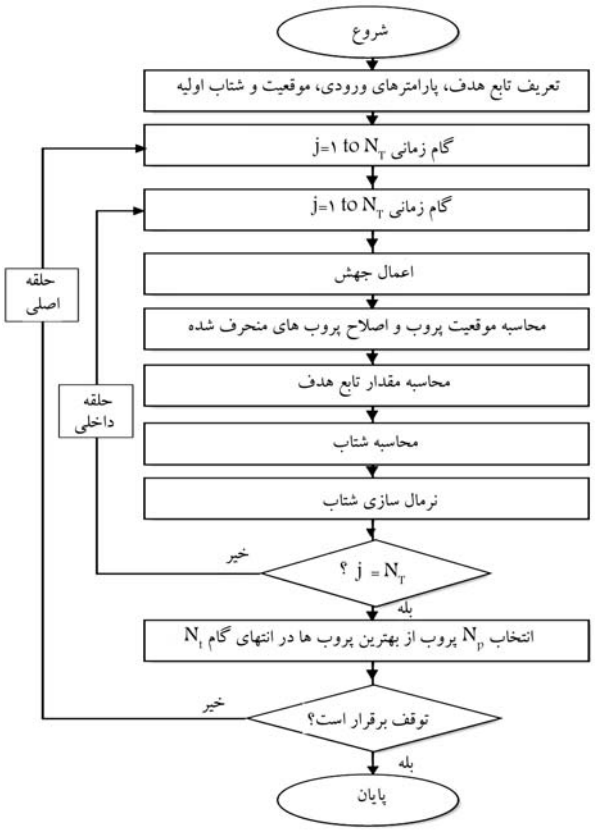
$$N_{mut} = (\frac{mutation}{۱۰۰}) \times N_p \times N_d \qquad (21)$$

$$R - mut^{M_y}_{j-1,M_x} = R^{M_y}_{j-1,M_x} + (r_1(b-a)+a) \times (r_2(x^{\max}_{M_x} - x^{\min}_{M_x}) + x^{\min}_{M_x}) \qquad (22)$$

که در آنها، r_1 و r_2 اعداد تصادفی هستند. پرانتز اول، عددی رندم در محدوده‌ی a تا b ایجاد می‌کند و پرانتز دوم نیز عددی رندم در فضای تصمیم انتخاب می‌کند. پیشنهاد می‌شود $a = ۰/۵$، $b = -۰/۵$ درنظرگرفته شود.

۳.۴.۲. استفاده از تاریخچه‌ی موقعیت پروب‌ها

پیچیدگی‌های الگوریتم CFO باعث افزایش زمان محاسبات می‌شود. الگوریتم CFO برای توابع مختلفی استفاده شده است، اما کمتر مشاهده شده است که از آن برای مسائل با تعداد بُعد بیش از ۳۰ استفاده شده باشد؛ زیرا در این حالت سرعت برنامه به‌شدت کاهش پیدا می‌کند.[۳۰] یکی دیگر از ضعف‌های CFO عدم استفاده از تاریخچه‌ی موقعیت پروب‌هاست. مطالعات زیادی در زمینه‌ی استفاده از تاریخچه‌ی شتاب در الگوریتم CFO انجام شده است.[۳۱] در الگوریتم RCFO نیز از تاریخچه‌ی موقعیت ذرات در تکرارهای قبلی استفاده شده است. اما در انتهای هر گام، تمامی موقعیت‌ها ذخیره نمی‌شود، بلکه فقط بهترین آن‌ها در حافظه باقی می‌ماند. در تکرار پایانی از حلقه‌ی داخلی، یک ماتریس با $N_t - ۱$ سطر ایجاد خواهد شد. از ماتریس به‌دست آمده به تعداد N_p پروب انتخاب می‌شود. بدین ترتیب خروجی حلقه‌ی داخلی تعداد N_p پروب از بهترین پروب‌هاست.



شکل ۴. الگوریتم RCFO.

۴.۴.۲. اضافه کردن حلقه‌ی تکرار اصلی

بعد از اجرای حلقه‌ی داخلی، برنامه به همراه ماتریس تشکیل یافته از بهترین پروب‌ها، وارد حلقه‌ی تکرار اصلی می‌شود. در حلقه‌ی مذکور، اگر شرایط توقف برقرار بود، برنامه متوقف خواهد شد. شکل ۴، مراحل اجرای الگوریتم RCFO را نشان می‌دهد.

۳. نتایج و بحث

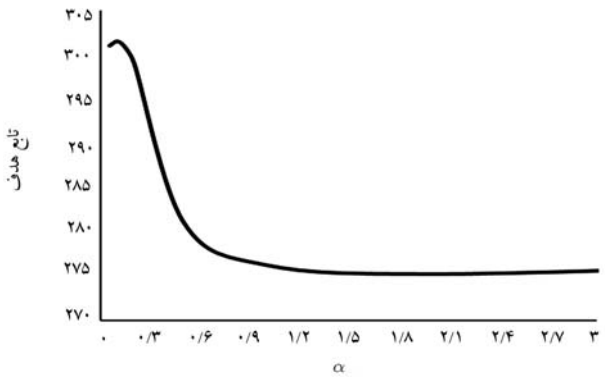
در این قسمت ابتدا مسئله‌ی ۴ مخزنه و سپس مسئله‌ی ۱۰ مخزنه با استفاده از روش های CFO و RCFO حل و نتایج با بهینه‌ی مطلق مقایسه شده است.

۱.۳. حل مسئله‌ی ۴ مخزنه

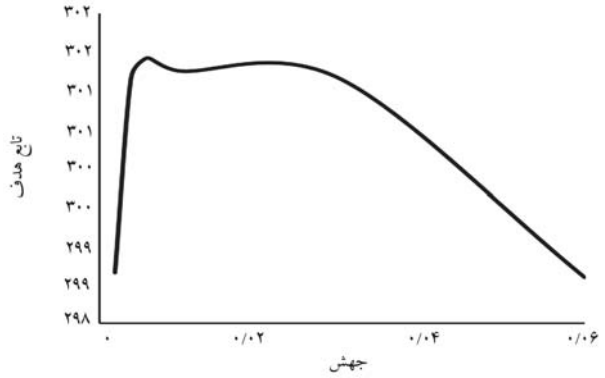
بهینه‌ی مطلق تابع هدف برای مسئله‌ی حاضر که ماهیتی خطی دارد، با استفاده از نرم‌افزار لینگو برابر با ۳۰۸/۲۹ است. با استفاده از الگوریتم CFO در بهترین حالت، نتیجه‌ی ۳۰۶/۸۹ حاصل شده است. در ادامه از الگوریتم RCFO برای حل مسئله‌ی ۴ مخزنه استفاده شده است.

۱.۱.۳. آنالیز حساسیت در روش RCFO

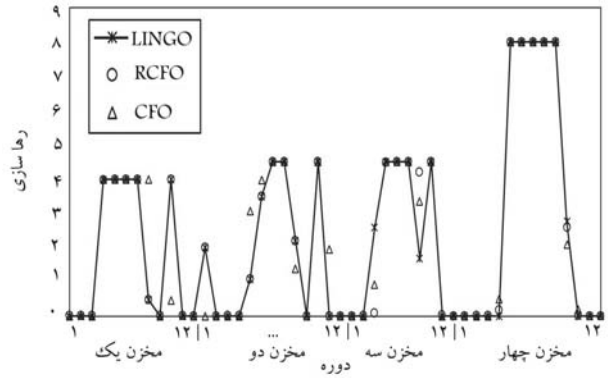
به منظور پیدا کردن مقادیر بهینه‌ی α و درصد جهش از آنالیز حساسیت استفاده شده و پارامترهای $F_{rep} = ۰/۵$ و $\beta = G = ۲$ ثابت درنظرگرفته شده‌اند.[۱۳] شکل ۵، آنالیز حساسیت را برای یافتن α بهینه نشان می‌دهد که با توجه به آن، بهترین α در محدوده‌ی ۰/۰۵-۰/۲ قرار دارد. به ازاء مقادیر بزرگ‌تر از ۱، تغییرات تابع هدف تقریباً یکسان است و برنامه قادر به ارائه‌ی جواب مناسب نیست.



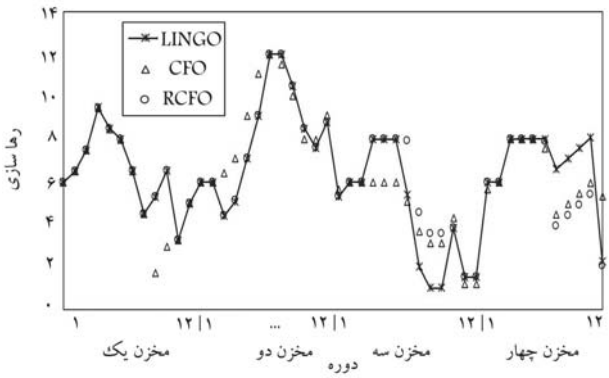
α



RCFO



شکل ۷. مقایسه‌ی رهاسازی به‌دست آمده از **RCFO,CFO** و لینگو.



شکل ۸. مقایسه‌ی حجم مخازن به‌دست آمده از **RCFO,CFO** و لینگو.

نیه

CFO

CFO

CFO

RCFO RCFO CFO

RCFO

۲.۱.۳. مقایسه‌ی نتایج

[۲۷] DDP

DDDP

DDP

HBMO

[۳۲]

RCFO

[۱۷]

ACO

۲.۳. حل مسئله‌ی ۱۰ مخزنه

[۳۳] HSA

[۳۴]

CFO

RCFO

RCFO

جدول ۳. خلاصه نتایج آماری ده بار اجرای RCFO برای ۲۰۰۰ تکرار.

	بهینه‌ی مطلق	بیشینه	متوسط	کمینه	انحراف معیار
تابع هدف	۳۰۸٫۲۹	۳۰۸٫۲۵	۳۰۸٫۲۱	۳۰۸٫۱۶	۰٫۰۳۵

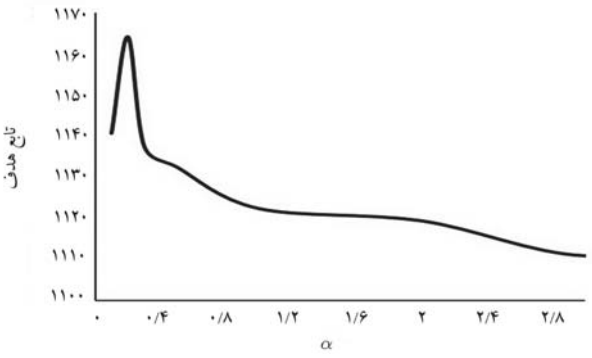
۱.۲.۳. آنالیز حساسیت در روش RCFO

در مسئله‌ی حاضر نیز به منظور پیداکردن پارامترهای بهینه‌ی α و درصد جهش از آنالیز حساسیت استفاده شده است. پارامترهای پارامترهای $F_{rep} = ۰٫۵$ و $\beta = G = ۲$ ثابت در نظر گرفته شده است.[۱۳] از آنجایی که RCFO احتمالاتی است، متوسط نتایج به‌دست آمده از هر پارامتر نمایش داده شده است. همان‌طور که در شکل ۹ مشاهده می‌شود، بهترین α در محدوده‌ی ۰٫۲ـ۰٫۱ است. از ۱ = α به بعد، تغییرات تابع هدف تقریباً یکسان است و تابع هدف روند کاهشی دارد و برنامه قادر به ارائه‌ی جواب مناسب نیست. با مراجعه به شکل ۱۰، بهترین جهش بین ۰٫۰۴۵ـ۰٫۰۳۵ است. در ابتدای نمودار نتایج خوبی به دست نمی‌آید، اما بعد از ۰٫۰۵، تغییرات تابع هدف به‌صورت جزئی مشاهده می‌شود.

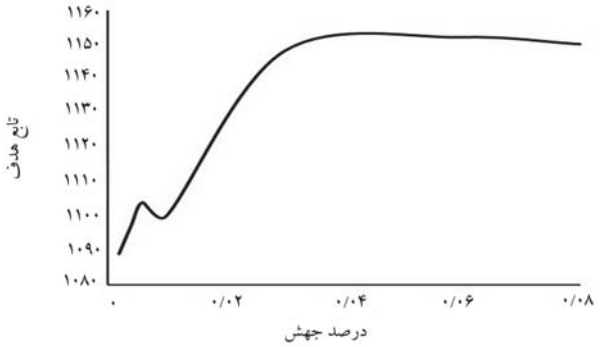
۲.۲.۳. مقایسه‌ی نتایج

مسئله‌ی ۱۰ مخزنه‌ی حاضر، بارها توسط پژوهشگران مختلف ارزیابی شده است. بهترین جواب به‌دست آمده با روش DDP برابر با ۱۱۹۰٫۲۵ و با GA برای ۱٫۲۵۰٫۰۰۰ تکرار برابر با ۱۱۹۰٫۲۵ گزارش شده است.[۱۷] با کاربرد الگوریتم HBMO پس از ۶۵۰۰۰ پرواز جفت گیری و یا به عبارتی ۱۴٫۲۴۲٫۳۴۱ بار ارزیابی، تابع هدف در مدت زمان ۱۵۰۰۰ ثانیه، نتیجه‌ی ۱۱۹۲٫۵۶ حاصل شده است.[۳۲] در پژوهشی در سال ۲۰۰۷، با استفاده از الگوریتم چندجامعه‌یی پیشنهادی ACO و با درنظر گرفتن ۳۰۰ مورچه و ۳۰۰۰ تکرار، مقدار ۱۱۹۲٫۳۹ گزارش شده است.[۲۰] در پژوهش دیگری در سال ۲۰۱۳، با استفاده از الگوریتم های FCPSO، PCOSO۱ و PCPSO۲ به ترتیب جواب‌های ۱۱۹۴٫۰۵، ۱۱۷۸٫۵۵ و ۱۱۹۱٫۷۵ و با تعداد جمعیت ۱۰۰، ۵۰ و ۱۰۰ گزارش شده است.[۲۲] متوسط زمان اجرا در هر سه روش مذکور و برای جمعیت ۵۰ تایی ذرات برابر با ۴۵ ثانیه است. در پژوهش حاضر، با استفاده از روش RCFO در پایان ۲۰۰۰ تکرار و با ۶ پروب، نتیجه‌ی ۱۱۹۲٫۷۴ حاصل شده است، که فقط ۰٫۱۳٪ با بهینه‌ی مطلق اختلاف دارد. متوسط زمان اجرا با استفاده از پردازنده‌ی ۱٫۵ گیگاهرتزی برابر با ۸۰۰۰ ثانیه بوده است. با توجه به این موضوع که برخی از روش‌ها مانند FCPSO نتیجه‌ی بهتری در زمان بسیار کمتری ارائه کرده است، لازم به ذکر است که الگوریتم CFO با توجه به ماهیتی که دارد، زمان طولانی برای جستجو و بهینه‌یابی نیاز دارد؛ اما الگوریتم RCFO توانسته است با برداشتن محدودیت تعداد پروب و فقط با استفاده از ۶ ذره، جست‌وجو را انجام دهد و زمان اجرا را نسبت به CFO تقریباً به نصف کاهش دهد.

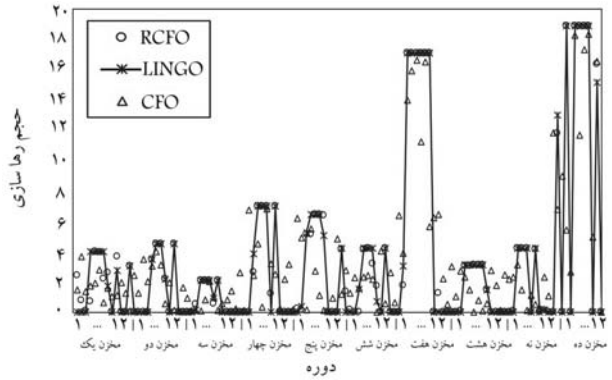
جدول ۴، نتایج آماری ۱۰ بار اجرای RCFO را برای ۲۰۰۰ تکرار نشان می‌دهد. در تمامی ۱۰ اجرا، نتایج قابل قبولی به‌دست آمده است، به طوری که انحراف معیار برابر با ۰٫۵۱۵ است. جواب نهایی پس از ۲۰۰۰ تکرار برابر با ۱۱۹۲٫۷۴ شده است. شکل‌های ۱۱ و ۱۲، به ترتیب رهاسازی و حجم مخزن به‌دست آمده از



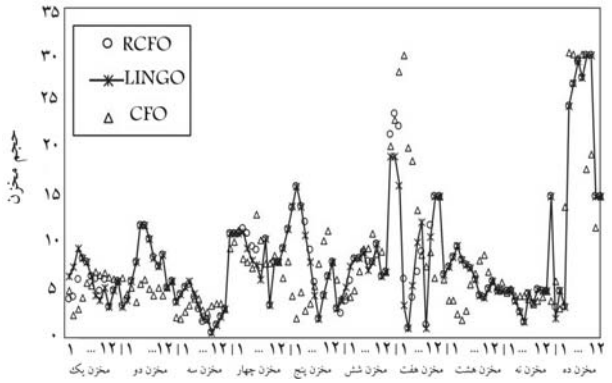
شکل ۹. آنالیز حساسیت α برای ۱۰۰۰ تکرار.



شکل ۱۰. آنالیز حساسیت درصد جهش برای ۱۰۰۰ تکرار.



شکل ۱۱. مقایسه‌ی رهاسازی به‌دست آمده از RCFO,CFO و لینگو.



شکل ۱۲. مقایسه‌ی حجم مخازن به‌دست آمده از RCFO,CFO و لینگو.

جدول ۴. خلاصه‌ی نتایج به‌دست آمده از ۱۰ بار اجرای **RCFO** برای ۲۰۰۰ تکرار.

	بهینه‌ی مطلق	بیشینه	متوسط	کمینه	انحراف معیار
تابع هدف	۱۱۹۴/۴	۱۱۹۲/۷۴	۱۱۹۲/۲۴	۱۱۹۱/۱	۰/۵۱۵

روش لینگو را در مقابل CFO و RCFO نشان می‌دهند که مطابق آنها اختلافات زیادی بین مقادیر به‌دست آمده از روش CFO با بهینه‌ی مطلق در هر دوره مشاهده می‌شود و این در حالی است که نتایج حاصل از RCFO به نتایج بهینه‌ی مطلق در هر دوره نزدیک است و فقط در برخی از دوره‌ها (مانند دوره‌ی ۱ تا ۳ در مخزن اول) اختلافاتی مشاهده می‌شود.

۴. نتیجه‌گیری

در نوشتار حاضر، الگوریتم بهینه‌یابی نیروی مرکزی در حل سیستم‌های چندمخزنه ارزیابی شده است. علی‌رغم توانایی بالای CFO در حل مسائل بهینه‌سازی، این الگوریتم در بهره‌برداری بهینه از سیستم‌های چندمخزنه نتایج خوبی ارائه نداده است.

به همین منظور در ساختار الگوریتم تغییراتی ایجاد و الگوریتم RCFO مطرح و برای بهینه‌سازی سیستم‌های چندمخزنه استفاده شده است. در هر دو مسئله‌ی ۴ و ۱۰ مخزنه، برتری روش RCFO مشخص شده است. الگوریتم CFO با توجه به ماهیتی که دارد، به زمان طولانی برای جستجو و بهینه‌یابی نیاز دارد؛ اما الگوریتم RCFO توانسته است با برداشتن محدودیت، تعداد پروب جست‌وجو را انجام دهد و زمان اجرا را نسبت به CFO تقریباً به نصف کاهش دهد. تغییراتی که در الگوریتم RCFO نسبت به CFO داده شده است منجر به ایجاد الگوریتمی شده است که قادر به حل مسائل پیچیده‌ی بهینه‌سازی است. عمده‌ی این تغییرات به منظور جلوگیری از گرفتار شدن در بهینه‌ی محلی و افزایش سرعت همگرایی است. با توجه به موفقیت اجرای الگوریتم RCFO در حل سیستم ۴ و ۱۰ مخزنه می‌توان نتیجه گرفت که الگوریتم RCFO، پتانسیل مناسبی جهت کاربرد در حل مسائل پیچیده‌ی بهره‌برداری از سیستم‌های چندمخزنه‌ی واقعی دارد.

پانوشت‌ها

1. linear programming
2. none linear programming
3. dynamic programming
4. ACO
5. GA
6. HBMO
7. PSO
8. central force optimization
9. artificial Bee colony
10. gravitational search algorithm
11. repetitive central force optimization algorithm

منابع (References)

1. Jalali, M.R., Afshar, A. and Marino, M.A. "Improved ant colony optimization algorithm for reservoir operation", *Scientia Iranica*, **13**(3), pp. 295-302 (2006).
2. Bozorg Haddad, O., Afshar, A. and Marino, M.A. "Honey-bees mating optimization (HBMO) algorithm: A new heuristic approach for water resources optimization", *Water Resources Management*, **20**(5), pp. 661-680 (2006).
3. Rani, D. and Moreira, M.M. "Simulation optimization modeling: A survey and potential application in reservoir systems operation", *Water Resource Management*, **24**(6), pp. 1107-1138 (2010).
4. Barros, M., Tsai, F., Yang, S., Lopes, J. and Yeh, W. "Optimization of large-acale hydropower system operation", *Journal Water Resources Planning Management*, **129**(3), pp. 178-188 (2003).
5. Bazaraa, M.S., Sherali, H.D. and Shetty, C.M., *Nonlinear Programming: Theory and Algorithms*, 3th Edn. John wiley and Sons Inc., Hoboken, New Jersey (2006).
6. Bellman, R., *Dynamic Programming*, Princeton, N.J., Princeton University Press (1957).
7. Dorigo, M. "Ant system: Optimization by a colony of cooperating Agents", *IEEE T SYST*, **26**(1), pp. 29-41 (1996).
8. Holland, J.H., *Adaption in Natural and Artificial Systems*, University of Michigan Press (1975).
9. Abbass, H.A. "MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach", *Conference Evolutionary Computation, Seoul*, **1**, pp. 207-214 (2001).
10. Kennedy, J. and Eberhart, R. "Particle Swarm Optimization", *IEEE International Conference*, **4**, pp. 1942-1948 (Nov.Dec 1995).
11. Geem, Z.W., Kim, J.H. and Loganathan, G.V. "A new heuristic optimization algorithm: Harmony search", *Simulation*, **76**(2), pp. 60-68 (2001).
12. Formato, R.A. "Central force optimization: A new metahereuristic with applications in applied electromagnetics", *Progress in Electromagnetics Research*, **77**(1), pp. 425-491 (2007).
13. Formato, R.A. "Parameter-free deterministic global search with simplified central force optimization", D.-S. Huang, et al. (Eds.), ICIC, Lecture Notes in Computer Science, 6215, pp. 309-318 (2010).

14. Formato, R.A. "Improving Bandwidth of Yagi-Uda Arrays", *Wireless Engineering and Technology*, **3**(1), pp. 18-24 (2012).
15. Mahmoud, K.R. "Central force optimization: Nelder-mead hybrid algorithm for rectangular microstrip antenna design", *Electromagnetics*, **31**(8), pp. 578-592 (2011).
16. Haghighi, A. and Ramos, H.M. "Detection of leakage Freshwater and Friction Factor Calibration in Drinking Networks Using Central Force Optimization", *Water Resource Management*, **26**(8), pp. 2347-2363 (2012).
17. Wardlaw, R. and Sharif, M. "Evaluation of genetic algorithms for optimal reservoir system reservoir system operation", *Water Resources Planning Management*, **125**(1), pp. 25-33 (1999).
18. Ahmad, A., Razali, S.F.M., Mohamed, Z.S. and Elshafie, A. "The application of artificial Bee colony and gravitational search algorithm in reservoir optimization", *Water Resources Management*, **30**(7), pp. 2497-2516 (2016).
19. Labadie J.W. "Optimal operation of multi-reservoir systems: State-of-the-art review", *Water Resources Planning Management*, **130**(2), pp. 93-111 (2004).
20. Jalali, M.R., Afshar, A. and Marino, M.A. "Multi-reservoir operation by adaptive pheromone re-initiated ant colony optimization algorithm", *International Journal of Civil Engineerng*, **5**(4), pp. 284-301 (2007).
21. Kougias, I.P. and Theodossiou, N.P. "Application of the harmony search optimization algorithm for the solution of the multiple dam system scheduling", *Optimization and Engineering*, **14**(2), pp. 331-344 (2013).
22. Afshar, M.H. "Extension of the constrained particle swarm optimization algorithm to optimal operation of multi-reservoirs system", *Electrical Power and Energy Systems*, **51**, pp. 71-81 (2013).
23. Moeini, R. and Afshar, M.H. "Extension of the constrained ant colony optimization algorithms for the optimal operation of multi-reservoir systems", *Hydro Informatics*, **15**(1), pp. 155-173 (2013).
24. Chow, V.T. and Cortes, R. "Applications of DDDP in water resources planning", Research Report 78, University of Illinois, Water Resources Center, Urbana (1974).
25. Heidari, M., chow, V.T., Kokotovic, P.V. and Meredith, D.D. "Discrete differenti a dynamic programing approach to water Resources systems optimization", *Water Resources Research*, **7**(2), pp. 273-282 (1971).
26. Michalewicz, Z. and Schoenauery, M. "Evolutionary algorithms for constrained parameter optimization problems", *Evolutionary Computation*, **4**(1), pp.1-32 (1996).
27. Murray, D.M. and Yakowitz, S.J. "Constrained differential dynamic programming and Its application to Multi-reservoir control", *Water Resources Research*, **15**(5), pp. 1017-1027 (1979).
28. Liua, Y. and Tian, P. "A multi-start central force optimization for global optimization", *Applied Soft Computing*, **27**, pp. 92-98 (2015).
29. Chen, Y., Yu, J., Mei, Y., Wang, Y. and Su, X. "Modified central force optimization (MCFO) algorithm for 3D UAV path planning", *Neurocomputing*, **171**(1), pp. 878-888 (2015).
30. Green, R.C., Wang, L., Alam, M. and Formato, R.A. "Central force optimization on a GPU: A case study in high performance meta-heuristics", *J. Supercomput*, **62**(1), pp. 378-398 (2012).
31. Ding, D., Qi, D., Luo, X., Chen, J., Wang, X. and Du, P. "Convergence analysis and performance of an extended central force optimization algorithm", *Applied Mathematics and Computation*, **219**(4), pp. 2246-2259 (2012).
32. Bozorg Haddad, O., Afshar, A. and Marino, M.A. "Multireservoir optimisation in discrete and continuous domains", *Water Management*, **164**(2), pp. 57-72 (2011).
33. Jalali, M.R. "Optimum design and operation of hydrosystems by Ant Colony Optimization algorithms; A new metaheuristic approach", PHD Dissertation, Civil Engineering, Iran University of Science and Technology (2005).
34. Kougias, I. and Theodossiou, N. "Optimization of multi-reservoir management using harmony search algorithm (HAS)", *3rd Intern. Conf. on Environmental Management Engineering, Planning and Economics*, Greece (2011).