

تحلیل عملکرد مقایسه‌ای هشت الگوریتم فراابتکاری برای طراحی بهینه سیستم سرمایش دیتاسنتر با قيود حرارتی
“Comparative performance analysis of eight metaheuristic algorithms for optimal design of data center cooling systems under thermal constraints”

علی اکبر نقابی*^۱، علی مسعودی^۲، بابک لطفی^۳، امیر رضا واقفی^۴

^۱ گروه مهندسی کامپیوتر و فناوری اطلاعات، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران

Aa_neghabi@iau.ac.ir

^۲ گروه مهندسی کامپیوتر و فناوری اطلاعات، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران

Alimasoudi@iau.ac.ir

^۳ گروه مهندسی کامپیوتر و فناوری اطلاعات، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران

Babak.lotfi@iau.ac.ir

^۴ گروه مهندسی کامپیوتر و فناوری اطلاعات، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران

Amirrezavaghefi@iau.ac.ir

چکیده :

سیستم‌های سرمایش دیتاسنترها سهم قابل توجهی از مصرف کل انرژی مراکز داده را به خود اختصاص می‌دهند و به‌عنوان یکی از چالش‌های اصلی در بهره‌برداری پایدار از این زیرساخت‌ها شناخته می‌شوند. با افزایش تراکم تجهیزات پردازشی و رشد سریع فناوری‌هایی نظیر رایانش ابری و هوش مصنوعی، طراحی و کنترل بهینه سیستم سرمایش به ضرورتی اجتناب‌ناپذیر تبدیل شده است. در این مقاله، یک چارچوب جامع برای طراحی بهینه سیستم سرمایش دیتاسنتر تحت قيود سخت حرارتی ارائه می‌شود که در آن عملکرد هشت الگوریتم فراابتکاری نسل جدید شامل مرغ شاخدار شمالی^۱، کلاغ نوک‌سرخ^۲، ماهی مرکب غول‌پیکر^۳، پانگولین چینی^۴، خفاش میوه‌خوار^۵، خرچنگ نعل‌اسبی^۶، شاهین بیابانی^۷ و لاک‌پشت دریایی مهاجر^۸ مورد تحلیل و مقایسه قرار می‌گیرد. تابع هدف مبتنی بر کمینه‌سازی مصرف انرژی و دمای دیتاسنتر بوده و قيود حرارتی بر اساس مدل دینامیکی حرارت دیتاسنتر اعمال شده‌اند. ارزیابی الگوریتم‌ها ابتدا با استفاده از توابع بنچمارک استاندارد CEC-2017 سپس بر روی مدل واقعی حرارتی دیتاسنتر انجام شده است. نتایج نشان می‌دهد الگوریتم‌های NBIO و GSSO از نظر دقت، نرخ همگرایی و پایداری حرارتی عملکرد برتری نسبت به سایر روش‌ها دارند و می‌توانند به‌عنوان گزینه‌های کارآمد برای طراحی و بهره‌برداری سیستم‌های سرمایش دیتاسنتر پیشنهاد شوند.

افزون بر این، تحلیل حساسیت نشان می‌دهد که انتخاب مناسب پارامترهای کنترلی در هر الگوریتم نقش تعیین‌کننده‌ای در کیفیت نتایج دارد و می‌تواند میزان صرفه‌جویی انرژی را به طور قابل توجهی افزایش دهد. همچنین، مقایسه میان الگوریتم‌ها نشان می‌دهد برخی روش‌ها در شرایط بارگذاری سنگین دیتاسنتر توانایی بیشتری در حفظ دماهای ایمن دارند.

کلمات کلیدی : دیتاسنتر، سیستم سرمایش، قيود حرارتی، بهینه‌سازی، کاهش مصرف انرژی، پایداری حرارتی، الگوریتم‌های فراابتکاری.

¹ Northern Bald Ibis Optimization-NBIO

² Red-billed Crow Optimization-RBCO

³ Giant Squid Swarm Optimization-GSSO

⁴ Pangolin Foraging Optimization-PFO

⁵ Fruit Bat Optimization-FBO

⁶ Horseshoe Crab Optimization-HCO

⁷ Desert Falcon Optimization-DFO

⁸ Sea Turtle Migration Optimization-STMO

مقدمه :

رشد چشمگیر مراکز داده در سراسر جهان که ناشی از توسعه خدمات ابری، اینترنت اشیا و کاربردهای مبتنی بر هوش مصنوعی است، منجر به افزایش قابل توجه مصرف انرژی شده است. بخش عمده‌ای از این انرژی صرف سیستم‌های سرمایش می‌شود که وظیفه حفظ دمای تجهیزات در محدوده مجاز استانداردهای صنعتی نظیر^۱ ASHRAE را بر عهده دارند [۱]. هرگونه ناکارآمدی در طراحی یا کنترل این سیستم‌ها می‌تواند منجر به افزایش هزینه‌های عملیاتی، کاهش عمر تجهیزات و حتی خاموشی‌های اضطراری شود. از این رو، طراحی بهینه سیستم سرمایش دیتاسنتر تحت قیود حرارتی سخت، یکی از چالش‌های مهم مهندسی در سال‌های اخیر به شمار می‌رود. مطالعات اخیر نشان داده‌اند که یادگیری تقویتی و یادگیری تقویتی عمیق می‌توانند نقش مؤثری در کاهش مصرف انرژی و بهینه‌سازی عملکرد دیتاسنترها ایفا کنند [۲].

چالش اصلی در طراحی سیستم‌های سرمایش، ماهیت پویای بار پردازشی و توزیع ناهمگون دما در راهروهای دیتاسنتر است. ناکارآمدی در مدیریت جریان هوا می‌تواند منجر به بروز پدیده نقاط داغ^۲ شود که خطر خرابی سخت‌افزار را به شدت افزایش می‌دهد. در مقابل، سرمایش بیش از حد نیز منجر به اتلاف انرژی و افزایش هزینه‌های عملیاتی می‌گردد. بنابراین، یافتن نقطه بهینه در تنظیم سرعت فن‌ها و نرخ جریان آب سرد^۳ تحت قیود سخت حرارتی، یک مسئله بهینه‌سازی غیرخطی و پیچیده^۴ تلقی می‌شود.

روش‌های سنتی کنترل نظیر سیستم‌های مبتنی بر قانون^۵ یا شباهه‌سازی‌های دینامیک سیالات محاسباتی (CFD) علی‌رغم دقت، در مواجهه با تغییرات بلادرنگ بار و ابعاد وسیع فضای جستجو کارایی لازم را ندارند. در سال‌های اخیر، الگوریتم‌های فراابتکاری^۶ به دلیل توانایی در پیمایش فضاها، جستجوی پیچیده و عدم نیاز به اطلاعات گرادینان، به عنوان راهکاری جایگزین و کارآمد ظهور کرده‌اند. این الگوریتم‌ها با الهام از پدیده‌های بیولوژیکی و فیزیکی، قادرند تعادلی میان کاهش مصرف توان و پایداری حرارتی برقرار کنند.

مسئله بهینه‌سازی سیستم سرمایش دیتاسنتر ماهیتی غیرخطی، چندهدفه و مقید دارد و به شدت به پارامترهایی نظیر بار پردازشی سرورها، سرعت فن‌ها، دبی آب سرد، شرایط محیطی و آرایش رک‌ها وابسته است. روش‌های کلاسیک مبتنی بر شبیه‌سازی‌های (CFD) اگرچه از دقت بالایی برخوردارند، اما به دلیل هزینه محاسباتی بالا، برای بهینه‌سازی و کنترل بلادرنگ مناسب نیستند. در مقابل، الگوریتم‌های فراابتکاری به دلیل انعطاف‌پذیری، قابلیت جستجوی سراسری و عدم نیاز به مشتق‌پذیری تابع هدف، به عنوان ابزارهایی قدرتمند برای حل این دسته از مسائل مطرح شده‌اند.

در این پژوهش، با الهام از چارچوب ارائه‌شده در مقالات مرجع اخیر، عملکرد هشت الگوریتم فراابتکاری نسل جدید در مسئله طراحی بهینه سیستم سرمایش دیتاسنتر مورد بررسی قرار می‌گیرد. نوآوری اصلی این مقاله، ارائه یک مقایسه جامع و منسجم از این الگوریتم‌ها بر روی یک مدل حرارتی واقع‌گرایانه و تحت قیود سخت حرارتی از قبیل مدل، قیود، الگوریتم‌ها، تنظیمات، معیارها، آزمون‌های آماری است که می‌تواند به عنوان مرجعی کاربردی برای پژوهشگران و مهندسان این حوزه مورد استفاده قرار گیرد. مدل‌سازی مسئله، معرفی الگوریتم‌ها، روش ارزیابی، نتایج و بحث، و نتیجه‌گیری در مراحل بعدی تکمیل می‌شود.

¹ American Society of Heating, Refrigerating and Air-Conditioning Engineers-ASHRAE

² Hotspots

³ Chilled Water

⁴ NP-Hard

⁵ Rule-based

⁶ Metaheuristic

۱- مدل‌سازی مسئله و قيود حرارتی :

۱-۱- تعریف مسئله بهینه‌سازی سیستم سرمایش دیتاسنتر

مسئله طراحی بهینه سیستم سرمایش دیتاسنتر در این پژوهش به صورت یک مسئله بهینه‌سازی مقید و چندهدفه مدل‌سازی می‌شود که هدف اصلی آن کاهش مصرف انرژی سیستم سرمایش، هم‌زمان با حفظ دمای تجهیزات در محدوده مجاز استانداردهای حرارتی است. متغیرهای تصمیم شامل پارامترهای کنترلی سیستم سرمایش نظیر سرعت فن‌ها، میزان بازشدگی شیرهای آب سرد و نرخ جریان هوا در راهروها هستند. این متغیرها به‌طور مستقیم بر توزیع دما، مصرف انرژی و پایداری عملکرد دیتاسنتر اثر می‌گذارند.

با وجود این پیشرفت‌ها، روش‌های یادگیری تقویتی نیازمند داده‌های عملیاتی گسترده و مدل‌سازی پیچیده هستند. در مقابل، الگوریتم‌های فراابتکاری به دلیل سادگی، انعطاف‌پذیری و عدم نیاز به گرادینان، گزینه‌ای مناسب برای حل مسائل بهینه‌سازی طراحی سرمایش هستند. به دلیل ماهیت غیرخطی و وابستگی متقابل بین متغیرها، رفتار حرارتی دیتاسنتر به شدت پیچیده بوده و استفاده از روش‌های تحلیلی کلاسیک برای حل مسئله با محدودیت‌های جدی مواجه است. از این رو، در این مقاله از الگوریتم‌های فراابتکاری برای جستجوی فضای جواب و دستیابی به راه‌حل‌های بهینه استفاده می‌شود.

۱-۲- الگوریتم‌های فراابتکاری در مهندسی

الگوریتم‌های فراابتکاری طی دو دهه گذشته به‌طور گسترده در حل مسائل پیچیده مهندسی استفاده شده‌اند. این الگوریتم‌ها با الهام از طبیعت، رفتار جمعی، فرآیندهای فیزیکی یا قوانین ریاضی طراحی می‌شوند و قادرند در فضای جستجوی بزرگ و چندبعدی، به راه‌حل‌های نزدیک به بهینه جهانی برسند. مقایسه چند الگوریتم فراابتکاری روی یک مسئله واقعی می‌تواند نقاط قوت و ضعف هر الگوریتم را آشکار کند. این مقاله الگوی ساختاری پژوهش حاضر را برای سیستم سرمایش دیتاسنتر شکل می‌دهد.

۲- مدل ریاضی مسئله سرمایش دیتاسنتر :

۲-۱- تعریف مسئله

هدف اصلی، کمینه‌سازی مصرف انرژی سیستم سرمایش دیتاسنتر در حالی است که :

- دمای راهروی سرد^۱
- دمای خروجی واحدهای سرمایشی^۲
- دمای ورودی تجهیزات^۳

همگی در محدوده ایمن حرارتی باقی بمانند. این مسئله یک مسئله بهینه‌سازی غیرخطی، چندقیدی و چندمتغیره است. برای مراکز داده، مدیریت انرژی یکی از مهم‌ترین چالش‌ها به شمار می‌رود. سیستم‌های سرمایش بخش قابل توجهی از مصرف انرژی کل دیتاسنتر را به خود اختصاص می‌دهند و به همین دلیل کاهش مصرف آن‌ها می‌تواند تأثیر چشمگیری بر هزینه‌های عملیاتی و پایداری زیست‌محیطی داشته باشد. هدف اصلی در این زمینه، طراحی و پیاده‌سازی راهکارهایی است که مصرف انرژی سیستم سرمایش را تا حد امکان کمینه کنند، بدون آنکه عملکرد تجهیزات و ایمنی حرارتی محیط به خطر بیفتد.

در این فرآیند، سه شاخص کلیدی باید همواره در محدوده ایمن باقی بمانند: دمای راهروی سرد، دمای خروجی واحدهای سرمایشی، و دمای ورودی تجهیزات. این پارامترها تعیین‌کننده سلامت و کارایی تجهیزات هستند و هرگونه انحراف از محدوده

¹ Cold Aisle Temperature

² Leaving Air Temperature-LAT

³ Inlet Temperature

مجاز می‌تواند منجر به کاهش عمر مفید سخت‌افزار یا حتی خرابی‌های جدی شود. بنابراین، الگوریتم‌های بهینه‌سازی باید به گونه‌ای عمل کنند که ضمن کاهش مصرف انرژی، این شرایط ایمن حرارتی نیز حفظ شوند.

مسئله‌ای که در اینجا مطرح می‌شود، یک مسئله بهینه‌سازی غیرخطی، چندقیدی و چندمتغیره است. به عبارت دیگر، مدل‌سازی آن نیازمند در نظر گرفتن روابط پیچیده میان متغیرهای مختلف و رعایت همزمان چندین محدودیت است. چنین مسائلی معمولاً با استفاده از روش‌های پیشرفته هوش مصنوعی و الگوریتم‌های متاهوریستیک حل می‌شوند، زیرا این روش‌ها توانایی جستجو در فضای بزرگ و پیچیده راه‌حل‌ها را دارند و می‌توانند تعادل مناسبی بین دقت، سرعت و کارایی برقرار کنند.

۲-۲- متغیرهای تصمیم (Decision Variables)

مسئله کنترل سرمایه‌دیتاسنتر به صورت یک MDP تعریف می‌شود:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma) \quad (1)$$

که در آن:

- \mathcal{S} : فضای حالت
- \mathcal{A} : فضای عمل
- دینامیک سیستم: $\mathcal{T}(S_{t+1} | S_t, a_t)$
- تابع پاداش: $r(S_t, a_t)$
- $\gamma \in (0, 1)$: ضریب تنزیل (0.99)

برای یک دیتاسنتر با M واحد سرمایه‌دیتاسنتر (ACU)، متغیرهای تصمیم عبارت‌اند از:

- سرعت فن هر ACU در اینجا تعداد ACUها برابر ۱۱ است.

$$\dim(\mathbf{x}) = 22 \text{ و } f_m \in [f_{\min}, f_{\max}]$$

بنابراین عدد ۲۲ بر اساس اینکه چون ۱۱ فن + ۱۱ شیر بدست می‌آید.

- درصد بازشدگی شیر آب سرد (Valve Opening)

$$o_m \in [0, 1]$$

بنابراین بردار تصمیم برابر است با:

$$\mathbf{x} = [f_1, o_1, f_2, o_2, \dots, f_M, o_M]$$

در مسئله کنترل سرمایه‌دیتاسنتر، مدل بهینه‌سازی به صورت یک فرایند تصمیم‌گیری مارکوف (MDP) تعریف می‌شود. این مدل شامل مجموعه‌ای از حالت‌ها (S)، اعمال (A)، دینامیک سیستم (T)، تابع پاداش (r) و ضریب تنزیل (γ) است. ضریب تنزیل معمولاً مقداری نزدیک به یک در نظر گرفته می‌شود (برای مثال ۰.۹۹). تا اهمیت تصمیم‌های آینده نیز در فرآیند بهینه‌سازی لحاظ شود. در این چارچوب، هدف آن است که با انتخاب مناسب اعمال در هر حالت، عملکرد سیستم سرمایه‌دیتاسنتر بهینه شود و مصرف انرژی کاهش یابد، در حالی که شرایط حرارتی ایمن برای تجهیزات حفظ می‌گردد. چنین مدل‌سازی‌هایی در پژوهش‌های پیشین نیز برای کاهش مصرف توان و کنترل سیستم‌های سرمایه‌دیتاسنتر مورد استفاده قرار گرفته‌اند [۳].

متغیرهای تصمیم در این مسئله به دو بخش اصلی تقسیم می‌شوند. نخست، سرعت فن‌های واحدهای سرمایه‌دیتاسنتر (ACU) که در اینجا تعداد آن‌ها برابر با ۱۱ است. هر فن می‌تواند در بازه‌ای مشخص از حداقل تا حداکثر سرعت کار کند و این انتخاب‌ها بخشی از بردار تصمیم را تشکیل می‌دهند. دوم، درصد بازشدگی شیرهای آب سرد که برای هر واحد سرمایه‌دیتاسنتر تعریف می‌شود و مقدار آن بین صفر تا یک متغیر است. این دو دسته متغیر در کنار هم، بردار تصمیم را شکل می‌دهند که ابعاد آن برابر با ۲۲ است (۱۱ فن + ۱۱ شیر).

بردار تصمیم نهایی به صورت $\chi = [f_1, o_1, f_2, o_2, \dots, f_M, o_M]$ تعریف می‌شود که در آن f_m سرعت فن واحد سرمایشی و o_m درصد بازشدگی شیر آب سرد مربوط به همان واحد است. این بردار ورودی اصلی الگوریتم کنترل محسوب می‌شود و تعیین‌کننده نحوه عملکرد سیستم سرمایش در هر لحظه خواهد بود. بهینه‌سازی این بردار به معنای یافتن ترکیب مناسبی از سرعت فن‌ها و بازشدگی شیرهاست که ضمن حفظ شرایط حرارتی ایمن، مصرف انرژی کل دیتاسنتر را به حداقل برساند.

۲-۳- چارچوب پیشنهادی ترکیب یادگیری تقویتی پیشرفته و الگوریتم‌های فراابتکاری

در این پژوهش، یادگیری تقویتی به‌عنوان یک بهینه‌ساز مستقل مورد استفاده قرار نگرفته است، بلکه در قالب یک چارچوب ترکیب‌شده با الگوریتم‌های فراابتکاری به کار گرفته شده است. در این چارچوب، یادگیری تقویتی نقش یک لایه کنترلی سطح بالا را ایفا می‌کند، در حالی که الگوریتم‌های فراابتکاری مسئول انجام بهینه‌سازی سطح پایین پارامترهای کنترلی سیستم سرمایش هستند. عامل یادگیری تقویتی با یادگیری رفتار دینامیکی محیط حرارتی دیتاسنتر و تغییرات بار پردازشی، به‌صورت تطبیقی پارامترهای الگوریتم‌های فراابتکاری از جمله ضرایب جستجو، ضرایب جریمه قیود و وزن‌های تابع هدف را تنظیم می‌کند. این ساختار سلسله‌مراتبی موجب می‌شود سیستم کنترلی بتواند در شرایط پویا و عدم قطعیت محیطی، تصمیم‌های هوشمند و بلادرنگ اتخاذ کند و در عین حال قابلیت جستجوی سراسری الگوریتم‌های فراابتکاری حفظ شود. ترکیب یادگیری تقویتی با الگوریتم‌های فراابتکاری در پژوهش‌های اخیر به‌عنوان یک راهکار مؤثر برای افزایش سازگاری و کارایی سیستم‌های بهینه‌سازی پیشنهاد شده است [۴].

۲-۴- نوآوری اصلی: ادغام یادگیری تقویتی پیشرفته با الگوریتم‌های فراابتکاری

چارچوب پیشنهادی امکان پیاده‌سازی یک سیستم کنترلی هوشمند، خودآموز و تطبیقی را برای سیستم سرمایش دیتاسنتر فراهم می‌کند. در این چارچوب، عامل یادگیری تقویتی پیشرفته (شامل یادگیری تقویتی عمیق یا چندعامله) با استفاده از داده‌های بلادرنگ حسگرها، بار حرارتی آینده را پیش‌بینی کرده و متناسب با تغییرات بار پردازشی، شرایط آب‌وهوایی و تعرفه انرژی، تصمیم‌های کنترلی بهینه اتخاذ می‌کند.

این رویکرد موجب کاهش مصرف انرژی بدون نقض قیود حرارتی شده و از نوسانات شدید دمایی جلوگیری می‌کند که این امر نقش مهمی در افزایش عمر مفید تجهیزات دیتاسنتر دارد. علاوه بر این، چارچوب پیشنهادی قابلیت ادغام با منابع انرژی تجدیدپذیر نظیر انرژی خورشیدی، انرژی بادی و سیستم‌های ذخیره‌ساز انرژی را دارا بوده و می‌تواند تصمیم‌های کنترلی آگاه از منابع انرژی را برای دیتاسنترهای نسل جدید فراهم کند. مطالعات اخیر نشان می‌دهند که یادگیری تقویتی عمیق می‌تواند نقش کلیدی در مدیریت همزمان بار حرارتی، منابع انرژی تجدیدپذیر و ذخیره‌سازهای انرژی در دیتاسنترها ایفا کند [۵].

۲-۵- تابع هدف (Objective Function)

تابع هدف مورد استفاده در این پژوهش با الهام از چارچوب مقاله مرجع و متناسب با شرایط عملیاتی دیتاسنتر تعریف شده است و شامل دو مؤلفه اصلی مصرف انرژی می‌باشد:

مصرف انرژی فن‌ها: مصرف انرژی فن‌ها به‌طور تقریبی متناسب با مکعب سرعت فن در نظر گرفته می‌شود که این رابطه وابستگی غیرخطی بین کنترل جریان هوا و توان مصرفی را مدل می‌کند. توان فن تقریباً متناسب با مکعب سرعت فن است:

$$P_{fan,m} = k_f f_m^3 \quad (2)$$

مصرف انرژی سیستم آب سرد: مصرف انرژی پمپ‌ها و چیلرها تابعی از دبی آب و میزان بازشدگی شیرها بوده و افزایش نرخ جریان آب منجر به افزایش توان مصرفی سیستم سرمایش می‌شود. بازشدگی شیر آب باعث افزایش مصرف پمپ‌ها می‌شود:

$$P_{water,m} = k_w o_m \quad (3)$$

تابع هدف کلی به صورت زیر تعریف می‌شود :

$$\min (J) = w_1 \sum_{i=1}^{N_f} P_{fan,i} + w_2 \sum_{j=1}^{N_w} P_{water,j} \quad (4)$$

که در آن $P_{fan,i}$ توان مصرفی فن i ام، $P_{water,j}$ توان مصرفی سیستم آب سرد، و w_1 و w_2 ضرایب وزنی هستند. باید در نظر داشت که موارد فوق به صورت یکسان می باشد.

$$N_f = N_w = M$$

بخش دوم تابع هدف به مصرف انرژی سیستم آب سرد مربوط می‌شود که شامل پمپ‌ها و چیلرهاست. توان مصرفی این بخش تابعی از دبی آب و میزان بازشدگی شیرهای آب سرد است. هرچه نرخ جریان آب افزایش یابد، توان مصرفی سیستم نیز بیشتر خواهد شد. همچنین بازشدگی شیرها به طور مستقیم بر مصرف انرژی پمپ‌ها اثر می‌گذارد. این موضوع اهمیت مدیریت دقیق جریان آب و تنظیم شیرها را در فرآیند بهینه‌سازی نشان می‌دهد، زیرا کوچک‌ترین تغییر در این متغیرها می‌تواند تأثیر قابل توجهی بر مصرف انرژی کل سیستم داشته باشد.

در نهایت، تابع هدف کلی به صورت ترکیبی از این دو مؤلفه تعریف می‌شود و با استفاده از ضرایب وزنی w_1 و w_2 سهم هر بخش در مصرف انرژی مشخص می‌گردد. این ضرایب به گونه‌ای انتخاب می‌شوند که اهمیت هر مؤلفه در شرایط عملیاتی واقعی دیتاسنتر منعکس شود. از آنجا که تعداد فن‌ها و شیرها برابر با تعداد واحدهای سرمایشی (M) است، ابعاد مسئله بهینه‌سازی برابر با $2M$ خواهد بود. این ساختار باعث می‌شود مسئله بهینه‌سازی پیچیده و چندمتغیره باشد و نیازمند استفاده از الگوریتم‌های پیشرفته برای یافتن ترکیب بهینه سرعت فن‌ها و بازشدگی شیرها باشد تا مصرف انرژی کل سیستم به حداقل برسد.

۲-۶- قیود حرارتی (Thermal Constraints)

به منظور تضمین عملکرد ایمن و پایدار دیتاسنتر، قیود حرارتی سخت بر روی دماهای کلیدی سیستم اعمال می‌شود که شامل موارد زیر است :

تعریف ریاضی State :

$$s_t = \{s_t^{(s)}, s_t^{(a)}, s_t^{(e)}\}$$

اجزای State :

• $s^{(s)}$: سنسورهای محیطی

$$s^{(s)} = \{T_{hot}, T_{cold}, H, T_{rack}\}$$

• $s^{(a)}$: وضعیت کاری 11ACU

$$s^{(a)} = \{LAT, EAT, LWT\}$$

• $s^{(e)}$: عوامل خارجی

$$s^{(e)} = \{EWT, P_{server}\}$$

قیود دمای راهروی سرد

$$T_n^{CAT}(x) \leq T_{safe}$$

قیود دمای خروجی واحد سرمایش (ACU)

$$T_m^{LAT}(x) \leq T_{LAT,max}$$

قیود دمای ورودی رک‌های سرور یا تجهیزات

$$T_r^{inlet}(x) \leq T_{inlet,max}$$

تعریف رسمی Action :

برای هر ACU :

$$a_m = (f_m, o_m)$$

که :

• f_m : سرعت فن ACU m

• o_m : میزان بازشدگی شیر آب

اگر تعداد ACU ها برابر M :

$$a = [f_1, \dots, f_M, o_1, \dots, o_M] \in \mathbb{R}^{2M}$$

برای دیتاسنتر واقعی :

$$M = 11 \Rightarrow \dim(a) = 22$$

این دماها از مدل حرارتی دیتاسنتر (TTDM) محاسبه می‌شوند. همچنین نقض هر یک از این قیود می‌تواند منجر به کاهش عمر تجهیزات، افت کارایی و خاموشی اضطراری شود؛ بنابراین در فرآیند بهینه‌سازی، پاسخ‌های ناقص قیود مردود در نظر گرفته می‌شوند.

در طراحی و کنترل سیستم‌های سرمایش دیتاسنتر، رعایت قیود حرارتی اهمیت حیاتی دارد زیرا این قیود تضمین می‌کنند که تجهیزات پردازشی در محدوده دمای ایمن فعالیت کنند. برای این منظور، مجموعه‌ای از متغیرهای حالت (State) تعریف می‌شود که شامل داده‌های سنسورهای محیطی مانند دمای راهروی گرم و سرد، رطوبت و دمای رک‌ها، وضعیت کاری واحدهای سرمایشی شامل دمای هوای خروجی و ورودی و دمای آب، و همچنین عوامل خارجی مانند دمای آب ورودی و توان مصرفی سرورها است. این اطلاعات پایه‌ای برای ارزیابی شرایط حرارتی دیتاسنتر و اعمال کنترل مناسب محسوب می‌شوند.

قیود حرارتی به‌طور مشخص سه بخش کلیدی را پوشش می‌دهند: نخست، دمای راهروی سرد که باید همواره کمتر یا مساوی مقدار ایمن تعریف شده باشد؛ دوم، دمای خروجی واحدهای سرمایشی (LAT) که نباید از حد مجاز فراتر رود؛ و سوم، دمای ورودی رک‌های سرور که باید در محدوده‌ای ایمن باقی بماند تا از آسیب به تجهیزات جلوگیری شود. این قیود به‌صورت ریاضی در مدل حرارتی دیتاسنتر (TTDM) اعمال می‌شوند و هرگونه نقض آن‌ها به معنای عدم پذیرش پاسخ در فرآیند بهینه‌سازی است.

از منظر کنترلی، اعمال (Action) برای هر واحد سرمایشی شامل دو متغیر تصمیم است: سرعت فن و میزان بازشدگی شیر آب سرد. در یک دیتاسنتر واقعی با ۱۱ واحد سرمایشی، بردار تصمیم شامل ۲۲ متغیر خواهد بود که ترکیب آن‌ها تعیین‌کننده نحوه عملکرد سیستم سرمایش در هر لحظه است. این ساختار باعث می‌شود که مسئله کنترل سرمایش به یک مسئله چندمتغیره و چندقیدی تبدیل شود که نیازمند الگوریتم‌های پیشرفته بهینه‌سازی است. رعایت دقیق قیود حرارتی نه تنها از کاهش عمر تجهیزات و افت کارایی جلوگیری می‌کند، بلکه مانع خاموشی اضطراری سیستم شده و پایداری عملیاتی دیتاسنتر را تضمین می‌نماید.

۷-۲- مدل دینامیکی حرارتی دیتاسنتر^۱

برای محاسبه توزیع دما و ارزیابی اثر متغیرهای کنترلی، از یک مدل دینامیکی حرارتی مبتنی بر معادلات دیفرانسیل معمولی (ODE) و ساختار گرافی دیتاسنتر استفاده شده است. اجزای دیتاسنتر شامل رک‌های سرور، واحدهای سرمایش و مسیره‌های جریان هوا به‌صورت گره‌های یک گراف حرارتی مدل‌سازی می‌شوند.

تعریف مشتق گسسته در فضای نهفته

¹ Thermal Dynamics Model

$$\dot{z}_s = z'_s - z_s \quad (5)$$

دینامیک رو به جلو و معکوس

$$\begin{aligned} f(z_s, z_a) &= \dot{z}_s \\ g(z'_s, z_a) &= -\dot{z}_s \end{aligned} \quad (6)$$

که در آن :

- z_s : نمایش نهفته وضعیت حرارتی
- z_a : نمایش نهفته اعمال (فن و شیر آب)

قید تقارن زمانی^۱ :

$$f(z_s, z_a) = -g(z'_s, z_a) \quad (7)$$

می توانیم به صورت ساده تر داشته باشیم :

$$\dot{T} = f(T, u, d) \quad (8)$$

که در آن T بردار دماها، u متغیرهای کنترلی و d اغتشاشات محیطی هستند. این مدل برای محاسبه دماهای واقعی استفاده می‌شود. دینامیک‌های پنهان رو به جلو و معکوس مقید به ارضای تقارن وارونی زمانی هستند.

۲-۸- تعریف دقیق نمادها

- f_m^3 : توان مصرفی فن (قانون مکعب) توان مصرفی فن‌ها تقریباً متناسب با مکعب سرعت فن است. این رابطه غیرخطی نشان می‌دهد که افزایش سرعت فن حتی به صورت جزئی، می‌تواند منجر به افزایش چشمگیر مصرف انرژی شود. بنابراین کنترل دقیق سرعت فن‌ها برای کاهش مصرف انرژی اهمیت حیاتی دارد.
- O_m : مصرف سمت آب این نماد درصد بازشدگی شیر آب سرد در واحد سرمایشی m را نشان می‌دهد. هرچه شیر بیشتر باز شود، دبی آب افزایش یافته و توان مصرفی پمپ‌ها بیشتر می‌شود. این متغیر تصمیم مستقیماً بر مصرف انرژی بخش آب سرد اثر می‌گذارد.
- T_c^n : دمای سنسور n در راهروی سرد^۲ این مقدار نشان‌دهنده دمای اندازه‌گیری شده توسط سنسورهای محیطی در راهروی سرد است. کنترل این دما برای حفظ شرایط ایمن تجهیزات حیاتی است، زیرا افزایش بیش از حد آن می‌تواند منجر به داغ شدن سرورها شود.
- T_l^m : مربوط به ACU یا دمای هوای خروجی از واحد سرمایشی است. این پارامتر نشان‌دهنده عملکرد مستقیم سیستم سرمایش بوده و باید در محدوده مجاز باقی بماند تا سرمایش مؤثر تضمین شود.
- ρ_T : آستانه دمای مجاز CAT این نماد حد بالای مجاز برای دمای راهروی سرد است. اگر T_c^n از این مقدار فراتر رود، سیستم در وضعیت ناایمن قرار می‌گیرد و باید اصلاح شود.
- ρ_L : آستانه LAT این نماد حد بالای مجاز برای دمای خروجی واحد سرمایشی است. رعایت این آستانه تضمین می‌کند که هوای خنک تولیدشده توسط ACU ها کیفیت لازم را دارد.
- $\text{soft penalty } \ln(1 + e^x)$: برای نقض قید این تابع به‌عنوان یک جریمه نرم برای نقض قیود حرارتی استفاده می‌شود. به جای حذف کامل پاسخ‌های ناقص قید، این تابع مقدار جریمه‌ای به آن‌ها اختصاص می‌دهد. ویژگی مهم این

¹ Time-Reversal Symmetry

² Cold Aisle Temperature

تابع آن است که پیوسته و هموار است و از جهش‌های ناگهانی جلوگیری می‌کند، بنابراین فرآیند بهینه‌سازی پایدارتر و روان‌تر پیش می‌رود.

جدول ۱: فهرست نمادها و پارامترهای مورد استفاده در مدل

شماره	نماد	توضیح
۱	f_m	سرعت فن واحد سرمایش m
۲	o_m	درصد بازشدگی شیر آب سرد
۳	k_f	ضریب مصرف توان فن
۴	k_w	ضریب مصرف توان سیستم آب سرد
۵	w_1, w_2	ضرایب وزن‌دهی تابع هدف
۶	β_1	ضریب جریمه مصرف انرژی فن
۷	β_2	ضریب جریمه دمای راهروی سرد
۸	β_3	ضریب جریمه مصرف انرژی آب سرد
۹	β_4	ضریب جریمه دمای خروجی واحد سرمایش

۹-۲- فرم نهایی مسئله بهینه‌سازی

$$r_t = r_0 - \beta_1 \sum_{m=1}^M f_{m,t}^3 - \beta_3 \sum_{m=1}^M o_{m,t} - \beta_2 \sum_{n=1}^N \ln(1 + \exp(T_{c,t}^n - \rho_T)) - \beta_4 \sum_{m=1}^M \ln(1 + \exp(T_{l,t}^m - \rho_L)) \quad (9)$$

به منظور ایجاد هماهنگی کامل میان تابع هدف بهینه‌سازی مورد استفاده در الگوریتم‌های فراابتکاری و تابع پاداش در یادگیری تقویتی، تابع پاداش به صورت مستقیم از تابع هدف استخراج شده است. به این ترتیب، بیشینه‌سازی پاداش در چارچوب یادگیری تقویتی معادل با کمینه‌سازی مصرف انرژی سیستم سرمایش همراه با رعایت قیود حرارتی خواهد بود.

$$r_t = -J_t - \lambda k \sum \max(0, T_k - T_{k\text{safe}}) \quad (10)$$

در این رابطه، جمله دوم به عنوان جریمه نقض قیود حرارتی عمل کرده و تضمین می‌کند که سیاست یادگیری شده تنها راه‌حل‌هایی را ترجیح دهد که علاوه بر کاهش مصرف انرژی، شرایط ایمن حرارتی تجهیزات را نیز حفظ می‌کنند. این فرم‌بندی، مبنای مقایسه منصفانه الگوریتم‌های فراابتکاری در بخش‌های بعدی مقاله است.

فرم نهایی مسئله بهینه‌سازی در این پژوهش به صورت یک تابع پاداش r_t تعریف شده است که ترکیبی از چند مؤلفه کلیدی مصرف انرژی و قیود حرارتی را در بر می‌گیرد. در این مدل، مقدار اولیه پاداش r_0 به عنوان نقطه شروع در نظر گرفته شده و سپس با کسر هزینه‌های ناشی از مصرف انرژی و نقض احتمالی قیود حرارتی، مقدار پاداش در هر گام زمانی محاسبه می‌شود. این ساختار باعث می‌شود که الگوریتم‌های بهینه‌سازی نه تنها به دنبال کاهش مصرف انرژی باشند، بلکه همزمان شرایط ایمن حرارتی دیتاسنتر را نیز رعایت کنند.

بخش اول تابع شامل مصرف انرژی فن‌هاست که با ضریب β_1 وزن‌دهی شده و متناسب با مکعب سرعت فن‌ها در نظر گرفته می‌شود. این رابطه غیرخطی نشان‌دهنده اهمیت کنترل دقیق سرعت فن‌ها در کاهش مصرف انرژی است. بخش دوم مربوط به مصرف انرژی سیستم آب سرد است که با ضریب β_3 وزن‌دهی شده و تابعی از میزان بازشدگی شیرهای آب سرد است. این دو مؤلفه انرژی، هسته اصلی هزینه‌های سیستم سرمایش را تشکیل می‌دهند.

علاوه بر مصرف انرژی، دو بخش دیگر تابع به نقض قیود حرارتی اختصاص دارند. برای دمای راهروی سرد، یک جریمه نرم با استفاده از تابع $\ln(1 + e^x)$ اعمال می‌شود که با ضریب β_2 وزن‌دهی شده است. این جریمه نرم باعث می‌شود که نقض قید به صورت تدریجی و پیوسته در تابع هدف لحاظ شود و از رفتارهای ناگهانی جلوگیری گردد. به طور مشابه، برای دمای خروجی واحدهای سرمایشی (LAT) نیز جریمه‌ای با ضریب β_4 تعریف شده است. این دو بخش تضمین می‌کنند که الگوریتم‌های بهینه‌سازی تنها پاسخ‌هایی را معتبر بدانند که شرایط ایمن حرارتی را رعایت کرده باشند. در مجموع، این فرم‌بندی یک چارچوب جامع برای ارزیابی و مقایسه الگوریتم‌های فراابتکاری فراهم می‌کند. با ترکیب مصرف انرژی و قیود حرارتی در یک تابع پاداش واحد، امکان مقایسه منصفانه عملکرد الگوریتم‌ها فراهم می‌شود و می‌توان بررسی کرد که هر روش تا چه اندازه قادر است بین کاهش مصرف انرژی و حفظ ایمنی حرارتی تعادل برقرار کند. این ویژگی، مبنای اصلی تحلیل‌های ارائه‌شده در بخش‌های بعدی مقاله است.

۲-۱۰- تابع بهینه‌سازی^۱

به‌روزرسانی Q-function :

$$\min_Q \mathbb{E}_{(s,a,s') \sim \mathcal{D}} [(r + \gamma Q(\phi(s'), \pi(s')) - Q(\phi(s), a))^2]$$

بهینه‌سازی Policy :

$$\max_{\pi} \mathbb{E}_{(s,a) \sim \mathcal{D}} [\lambda_{\alpha} Q(\phi(s), \pi(s)) - \|\pi(s) - a\|^2 - \ell_{T-sym}]$$

در چارچوب یادگیری تقویتی آفلاین، مسئله بهینه‌سازی شامل دو بخش اصلی است، به‌روزرسانی Q-function و بهینه‌سازی Policy. در بخش اول، Q-function به‌عنوان تخمین‌زننده ارزش اعمال در حلت‌های مختلف عمل می‌کند. فرمول ارائه‌شده نشان می‌دهد که هدف کمینه‌سازی خطای بین مقدار پیش‌بینی شده Q و بازگشت واقعی است. این بازگشت شامل پاداش لحظه‌ای r و ارزش آینده با ضریب تنزیل γ است. به این ترتیب، Q-function به تدریج یاد می‌گیرد که اعمال مختلف در حلت‌های گوناگون چه ارزش بلندمدتی دارند و این یادگیری بر اساس داده‌های تجربی موجود در مجموعه D انجام می‌شود، بدون نیاز به تعامل مستقیم با محیط واقعی.

در بخش دوم، سیاست یا Policy بهینه‌سازی می‌شود تا بهترین عمل ممکن در هر حالت انتخاب شود. این فرمول به صورت یک مسئله بیشینه‌سازی تعریف شده است که در آن سیاست باید بیشترین ارزش Q را ایجاد کند. همزمان، یک جریمه برای فاصله گرفتن بیش از حد از اعمال مشاهده‌شده در داده‌ها ($\|\pi(s) - a\|^2$) اعمال می‌شود تا از انحراف توزیعی جلوگیری شود. همچنین عبارت l_{T-sym} به‌عنوان یک قید اضافی برای حفظ ویژگی‌های خاص سیستم در نظر گرفته شده است. در مجموع، این ساختار باعث می‌شود سیاست نهایی هم ارزش بالایی داشته باشد و هم با داده‌های واقعی سازگار باقی بماند، و این ترکیب برای مسائل پیچیده‌ای مانند کنترل سرمایه‌دیتاسنتر بسیار کارآمد است.

¹ Offline RL

۳- معرفی الگوریتم‌های فراابتکاری مورد استفاده :

در این پژوهش، به منظور حل مسئله بهینه‌سازی سیستم سرمایش دیتاسنتر تحت قیود حرارتی، از هشت الگوریتم فراابتکاری نسل جدید که در سال ۲۰۲۵ استفاده شده است. این الگوریتم‌ها با الهام از پدیده‌های طبیعی، زیستی و رفتاری طراحی شده‌اند و هر یک دارای سازوکار متفاوتی برای ایجاد تعادل بین دو مفهوم اکتشاف (Exploration) و بهره‌برداری (Exploitation) هستند. در ادامه، ساختار کلی این الگوریتم‌ها و ویژگی‌های شاخص آن‌ها در کاربرد سرمایش دیتاسنتر تشریح می‌شود. همچنین تمام الگوریتم‌ها در فضای جستجوی ۲۲ بعدی (سرعت فن‌ها و بازشدگی شیرهای آب) اجرا شده‌اند. الگوریتم‌های فراابتکاری الهام‌گرفته از طبیعت، نظیر الگوریتم گرگ خاکستری، به‌طور گسترده برای حل مسائل بهینه‌سازی پیچیده استفاده شده‌اند [۶].

۳-۱- الگوریتم بهینه‌سازی پرنده ایبیس شمالی (NBIO)

الگوریتم NBIO با الهام از رفتار مهاجرت گروهی پرنده ایبیس شمالی طراحی شده است. در این الگوریتم، هر عامل جستجو نماینده موقعیت یک پرنده در فضای جواب بوده و حرکت جمعی پرندگان به سمت رهبر گروه، منجر به همگرایی تدریجی به سمت نواحی بهینه می‌شود. وجود مؤلفه‌های آشفتگی کنترل‌شده، از گیر افتادن الگوریتم در کمینه‌های محلی جلوگیری می‌کند. پایداری بالا و توانایی حفظ قیود حرارتی از مهم‌ترین مزایای NBIO در مسئله سرمایش دیتاسنتر است. الهام‌گرفته از مهاجرت گروهی پرنده کمیاب «مرغ شاخدار شمالی» که در مسیرهای طولانی با الگوی V- شکل حرکت می‌کنند.

مدل ریاضی :

هر پرنده یک موقعیت در فضای جستجو دارد :

$$X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$$

حرکت پرنده به سمت رهبر :

$$X_i^{t+1} = X_i^t + \alpha(L^t - X_i^t) + \beta \epsilon \quad (11)$$

که در آن :

- L^t : بهترین پرنده
- α : ضریب هم‌راستایی
- β : ضریب آشفتگی
- ϵ : نویز تصادفی

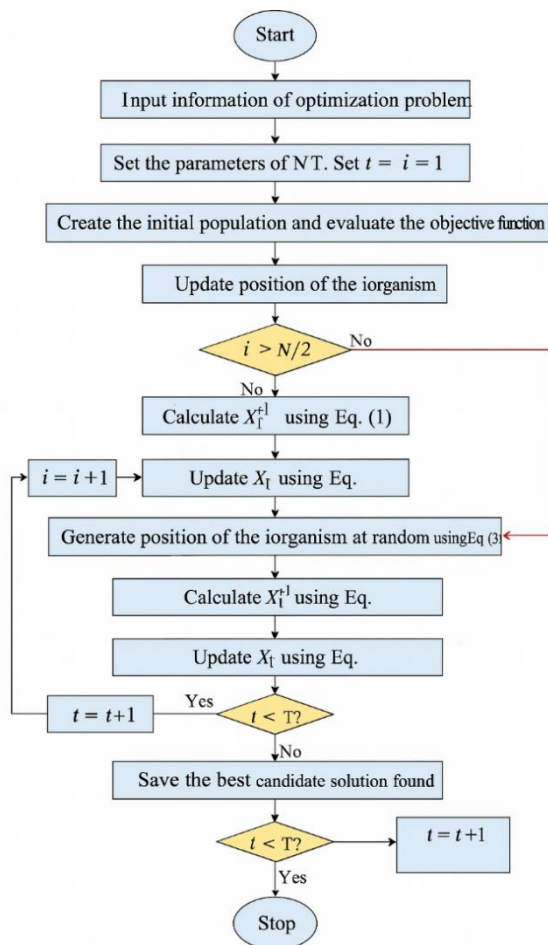
ویژگی‌ها :

- همگرایی سریع
- مناسب برای مسائل چندقیدی
- عملکرد عالی در مدل حرارتی دیتاسنتر

دیگرام الگوریتم پرنده ایبیس شمالی در شکل زیر نشان می‌دهد که این الگوریتم شامل چندین مرحله اصلی زیر است :

- شروع و تنظیم پارامترها تعداد افراد، تعداد تکرار، ضرایب حرکت.
- تولید جمعیت اولیه X داخل حدود مجاز متغیرها.
- ارزیابی تابع هدف برای همه افراد و محاسبه جریمه قیود (اگر لازم باشد).
- انتخاب رهبر/بهترین پرنده L .
- به‌روزرسانی موقعیت هر پرنده : حرکت به سمت رهبر + یک اغتشاش کوچک (برای فرار از گیر افتادن).
- کنترل مرزها (Repair/Clamp) اگر متغیری از محدوده خارج شد اصلاح می‌شود.
- به‌روزرسانی بهترین جواب (Best).

- بررسی شرط توقف رسیدن به MaxIter و در غیر این صورت تکرار از مرحله ارزیابی.
- الگوریتم پرنده ایبیس شمالی یک روش فراابتکاری الهام‌گرفته از رفتار پرندگان در پرواز گروهی است که برای حل مسائل پیچیده بهینه‌سازی طراحی شده است. در این الگوریتم، هر پرنده به‌عنوان یک راه‌حل در فضای جستجو در نظر گرفته می‌شود و موقعیت آن با بردار $X_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ نمایش داده می‌شود. حرکت پرنده‌ها به سمت رهبر یا بهترین فرد جمعیت بر اساس رابطه‌ای تعریف می‌شود که شامل دو مؤلفه اصلی است: ضریب هم‌راستایی α که پرنده‌ها را به سمت رهبر هدایت می‌کند، و ضریب آشفتگی β همراه با نویز تصادفی ϵ که تنوع جمعیت را حفظ کرده و مانع از گیر افتادن الگوریتم در بهینه‌های محلی می‌شود. این ترکیب باعث می‌شود الگوریتم هم سرعت همگرایی بالایی داشته باشد و هم توانایی جستجوی گسترده در فضای راه‌حل‌ها را حفظ کند.
- ویژگی مهم این الگوریتم آن است که برای مسائل چندقیدی بسیار مناسب است، زیرا علاوه بر حرکت به سمت بهترین راه‌حل، امکان اعمال جریمه برای نقض قیود نیز وجود دارد. در کاربردهای عملی مانند مدل حرارتی دیتاسنتر، این الگوریتم عملکرد بسیار خوبی نشان داده است. دلیل این موفقیت آن است که الگوریتم می‌تواند بین کاهش مصرف انرژی و رعایت شرایط ایمن حرارتی تعادل برقرار کند. وجود نویز تصادفی در فرمول حرکت نیز باعث می‌شود الگوریتم بتواند از بهینه‌های محلی فرار کند و به سمت راه‌حل‌های بهتر حرکت نماید.
- فرآیند اجرای الگوریتم شامل چندین مرحله اصلی است. ابتدا پارامترهای کلیدی مانند تعداد افراد، تعداد تکرارها و ضرایب حرکت تنظیم می‌شوند. سپس جمعیت اولیه در محدوده مجاز متغیرها تولید می‌شود. در هر تکرار، تابع هدف برای همه افراد ارزیابی شده و در صورت وجود قیود، جریمه‌های مربوطه محاسبه می‌گردد. بهترین پرنده یا رهبر انتخاب می‌شود و موقعیت سایر پرندگان با ترکیب حرکت به سمت رهبر و یک اغتشاش کوچک به‌روزرسانی می‌شود. اگر تغییری از محدوده مجاز خارج شود، با روش‌های اصلاح یا محدودسازی (Repair/Clamp) به محدوده بازگردانده می‌شود. در نهایت، بهترین جواب به‌روزرسانی شده و شرط توقف بررسی می‌شود؛ اگر تعداد تکرارها به حد نهایی نرسیده باشد، فرآیند دوباره از مرحله ارزیابی آغاز می‌شود. این چرخه تا رسیدن به بهترین راه‌حل ادامه می‌یابد و الگوریتم در نهایت یک جواب بهینه یا نزدیک بهینه ارائه می‌دهد.



شکل ۱: فلوچارت الگوریتم بهینه‌سازی پرنده ایبیس شمالی

۲-۳- الگوریتم بهینه‌سازی کلاغ نوک‌سرخ (RBCO)

الگوریتم RBCO مبتنی بر هوش اجتماعی کلاغ‌ها و اشتراک‌گذاری اطلاعات درون گروهی است. در این روش، هر عامل جستجو با استفاده از اطلاعات همسایگان و بهترین عضو گروه، موقعیت خود را به‌روزرسانی می‌کند. این ویژگی موجب افزایش قدرت اکتشاف در فضاهای جستجوی با ابعاد بالا شده و RBCO را به گزینه‌ای مناسب برای مسائل پیچیده و غیرخطی نظیر سیستم سرمایه‌گذاری دیتاسنتر تبدیل می‌کند. الهام از هوش اجتماعی کلاغ‌های نوک‌سرخ که در گروه‌های کوچک اطلاعات را به اشتراک می‌گذارند. این الگوریتم با الهام از هوش اجتماعی و اشتراک‌گذاری اطلاعات در میان گونه *Corvus pyrrhocorax* معرفی شده است [۸].

معادله به‌روزرسانی:

$$X_i^{t+1} = X_i^t + r_1(G^t - X_i^t) + r_2(N_i^t - X_i^t) \quad (12)$$

که در آن:

- G^t : بهترین کلاغ گروه
- N_i^t : همسایه نزدیک
- r_1, r_2 : ضرایب تصادفی

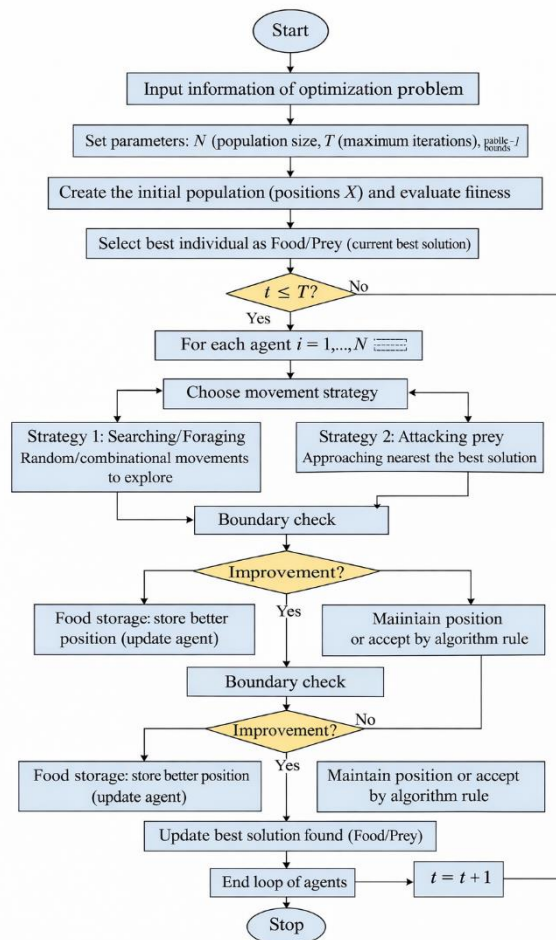
ویژگی‌ها:

- تعادل خوب بین اکتشاف و بهره‌برداری
- مناسب برای مسائل با ابعاد بالا
- دیاگرام الگوریتم کلاغ نوک‌سرخ در شکل 2 نشان می‌دهد که این الگوریتم شامل چندین مرحله اصلی زیر است :
تنظیم پارامترها و ایجاد جمعیت اولیه کلاغ‌ها
- ارزیابی تابع هدف و جریمه قیود
- انتخاب بهترین کلاغ گروه G یا $Global\ best$ یا $leader$
- برای هر کلاغ : انتخاب همسایه/همکار N_i یک عضو دیگر برای یادگیری اجتماعی
- آپدیت موقعیت : حرکت ترکیبی به سمت بهترین گروه و همچنین به سمت همسایه (دو مؤلفه یادگیری)
- کنترل مرزی و اصلاح جواب‌های خارج از محدوده
- ذخیره بهترین جواب
- تکرار تا شرط توقف

الگوریتم کلاغ نوک‌سرخ یک روش فراابتکاری الهام‌گرفته از رفتار اجتماعی پرندگان است که برای حل مسائل پیچیده بهینه‌سازی طراحی شده است. در این الگوریتم، هر کلاغ به‌عنوان یک راه‌حل در فضای جستجو در نظر گرفته می‌شود و موقعیت آن با بردار X_i نمایش داده می‌شود. معادله به‌روزرسانی موقعیت نشان می‌دهد که حرکت هر کلاغ ترکیبی از دو مؤلفه اصلی است: حرکت به سمت بهترین کلاغ گروه (G^t) و حرکت به سمت نزدیک‌ترین همسایه (N_i^t). ضرایب تصادفی r_1 و r_2 تیز برای ایجاد تنوع و جلوگیری از همگرایی زودهنگام به کار می‌روند. این ساختار باعث می‌شود الگوریتم بتواند تعادلی مناسب بین اکتشاف (جستجوی گسترده در فضای راه‌حل‌ها) و بهره‌برداری (تمرکز بر بهترین جواب‌های موجود) برقرار کند.

ویژگی مهم این الگوریتم توانایی آن در حل مسائل با ابعاد بالا است. با توجه به اینکه هر کلاغ علاوه بر رهبر، از همسایه نزدیک خود نیز یاد می‌گیرد، الگوریتم از دو مؤلفه یادگیری اجتماعی بهره می‌برد. این امر موجب افزایش سرعت همگرایی و در عین حال حفظ تنوع جمعیت می‌شود. در کاربردهایی مانند مدل حرارتی دیتاسنتر، این الگوریتم عملکرد مطلوبی دارد زیرا می‌تواند همزمان مصرف انرژی را کاهش دهد و شرایط ایمن حرارتی را رعایت کند.

فرآیند اجرای الگوریتم شامل چند مرحله اصلی است. ابتدا پارامترها تنظیم شده و جمعیت اولیه کلاغ‌ها در محدوده مجاز متغیرها ایجاد می‌شود. سپس تابع هدف برای همه افراد ارزیابی شده و در صورت وجود قیود، جریمه‌های مربوطه اعمال می‌گردد. بهترین کلاغ گروه به‌عنوان رهبر انتخاب می‌شود و هر کلاغ یک همسایه برای یادگیری اجتماعی برمی‌گزیند. موقعیت‌ها با ترکیب حرکت به سمت رهبر و همسایه به‌روزرسانی می‌شوند و در صورت خروج از محدوده، اصلاح یا محدودسازی انجام می‌شود. در پایان هر تکرار، بهترین جواب ذخیره شده و فرآیند تا رسیدن به شرط توقف ادامه می‌یابد. این چرخه باعث می‌شود الگوریتم به تدریج به سمت راه‌حل‌های بهینه یا نزدیک بهینه حرکت کند.



شکل ۲: فلوجارت الگوریتم بهینه‌سازی کلاغ نوک‌سرخ

۳-۳- الگوریتم بهینه‌سازی ازدحام ماهی مرکب گول‌پیکر (GSSO)

الگوریتم GSSO با الهام از رفتار شکار و استتار ماهی مرکب گول‌پیکر توسعه یافته است. این الگوریتم از نرخ همگرایی بالایی برخوردار بوده و توانایی قابل توجهی در تمرکز سریع بر نواحی بهینه دارد. نتایج تجربی نشان می‌دهد GSSO در کاهش مصرف انرژی سیستم سرمایه‌گذاری و حفظ دما در محدوده مجاز عملکرد بسیار مطلوبی ارائه می‌دهد. رفتار شکار و استتار ماهی مرکب گول‌پیکر است. این الگوریتم با بهره‌گیری از سازوکارهای زیستی شکار و استتار، برای جستجوی سراسری در مسائل بهینه‌سازی پیشنهاد شده است [۹].

معادله حرکت:

$$X_i^{t+1} = X_i^t + \gamma(B^t - X_i^t) + \delta \sin(\omega t) \quad (13)$$

که در آن:

- B^t : بهترین شکارچی
- γ : ضریب حمله
- δ : شدت استتار

ویژگی‌ها:

- نرخ همگرایی بسیار بالا

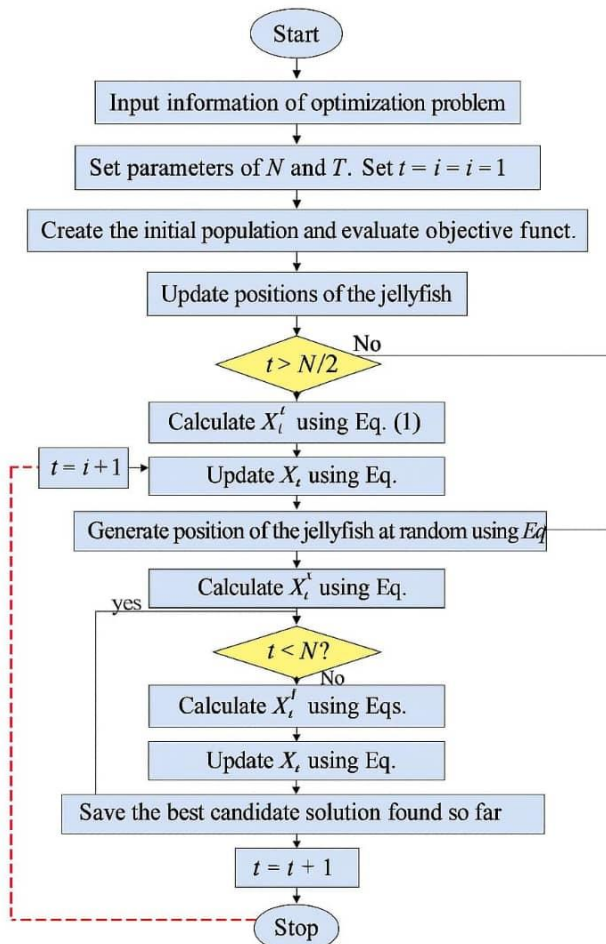
- بهترین عملکرد در نتایج اولیه دیتاسنتز

دیگرام الگوریتم بهینه‌سازی ازدحام ماهی مرکب غول‌پیکر در شکل زیر نشان می‌دهد که این الگوریتم شامل چندین مرحله اصلی زیر است :

- شروع و تولید جمعیت اولیه ماهی مرکبها
- ارزیابی تابع هدف و جریمه قیود
- انتخاب بهترین عضو B
- محاسبه مؤلفه نوسانی/تناوبی (برای ایجاد تنوع در جست‌وجو)
- آپدیت موقعیت هر عضو : حرکت به سمت بهترین + اضافه شدن مؤلفه نوسانی (تعادل اکتشاف/بهره‌برداری)
- کنترل مرزی
- به‌روزرسانی Best
- بررسی توقف و تکرار

الگوریتم بهینه‌سازی ازدحام ماهی مرکب غول‌پیکر یکی از روش‌های فراابتکاری الهام‌گرفته از رفتار شکار و استتار این موجودات دریایی است. در این الگوریتم، هر ماهی مرکب به‌عنوان یک راه‌حل در فضای جست‌وجو در نظر گرفته می‌شود و موقعیت آن با معادله به‌روزرسانی مشخص می‌گردد. حرکت هر عضو ترکیبی از دو مؤلفه اصلی است: حرکت به سمت بهترین شکارچی یا رهبر جمعیت (B^t) با ضریب حمله γ ، و یک مؤلفه نوسانی مبتنی بر تابع سینوسی با شدت δ که نقش استتار و ایجاد آشفتگی را ایفا می‌کند. این ترکیب باعث می‌شود الگوریتم هم سرعت همگرایی بسیار بالایی داشته باشد و هم توانایی فرار از بهینه‌های محلی را حفظ کند. ویژگی بارز این الگوریتم، تعادل مناسب بین اکتشاف و بهره‌برداری است. مؤلفه حمله، جمعیت را به سمت بهترین راه‌حل موجود هدایت می‌کند، در حالی که مؤلفه نوسانی/تناوبی تنوع لازم را ایجاد کرده و امکان جست‌وجوی گسترده‌تر در فضای راه‌حل‌ها را فراهم می‌سازد. همین ویژگی‌ها موجب شده است که الگوریتم در نتایج اولیه مربوط به مدل حرارتی دیتاسنتز عملکرد بسیار خوبی نشان دهد و بتواند مصرف انرژی را کاهش دهد در حالی که شرایط ایمن حرارتی نیز حفظ می‌شود.

فرآیند اجرای الگوریتم شامل چند مرحله اصلی است. ابتدا جمعیت اولیه ماهی مرکبها تولید می‌شود و سپس تابع هدف برای همه افراد ارزیابی شده و در صورت وجود قیود، جریمه‌های مربوطه اعمال می‌گردد. بهترین عضو جمعیت انتخاب می‌شود و مؤلفه نوسانی برای ایجاد تنوع محاسبه می‌گردد. موقعیت هر عضو با ترکیب حرکت به سمت بهترین و اضافه شدن مؤلفه نوسانی به‌روزرسانی می‌شود. در ادامه، کنترل مرزی انجام می‌شود تا متغیرها در محدوده مجاز باقی بمانند. بهترین جواب در هر تکرار به‌روزرسانی شده و فرآیند تا رسیدن به شرط توقف ادامه می‌یابد. این چرخه باعث می‌شود الگوریتم به تدریج به سمت راه‌حل‌های بهینه یا نزدیک بهینه حرکت کند.



شکل ۳: فلوجارت الگوریتم بهینه‌سازی ازدحام ماهی مرکب گول‌پیکر

۳-۴- الگوریتم بهینه‌سازی جستجوی پانگولین (PFO)

الگوریتم PFO بر اساس الگوی تغذیه و جستجوی پانگولین طراحی شده است. مکانیزم اصلی این الگوریتم، جستجوی تطبیقی و فرار از کمینه‌های محلی است. اگرچه PFO در مسائل با ساختار ساده‌تر عملکرد مناسبی دارد، اما در مسائل با قیود حرارتی سخت، پایداری آن نسبت به برخی الگوریتم‌های دیگر کمتر گزارش شده است. جستجوی غذا توسط پانگولین با الگوی مارپیچی است. الگوی جستجوی مارپیچی پانگولین نقش کلیدی در افزایش توان فرار از کمینه‌های محلی در این الگوریتم ایفا می‌کند [۱۰].

معادله حرکت:

$$X_i^{t+1} = X_i^t + \eta e^{-\lambda t} \text{spiral}(X_i^t, B^t) \quad (14)$$

ویژگی‌ها:

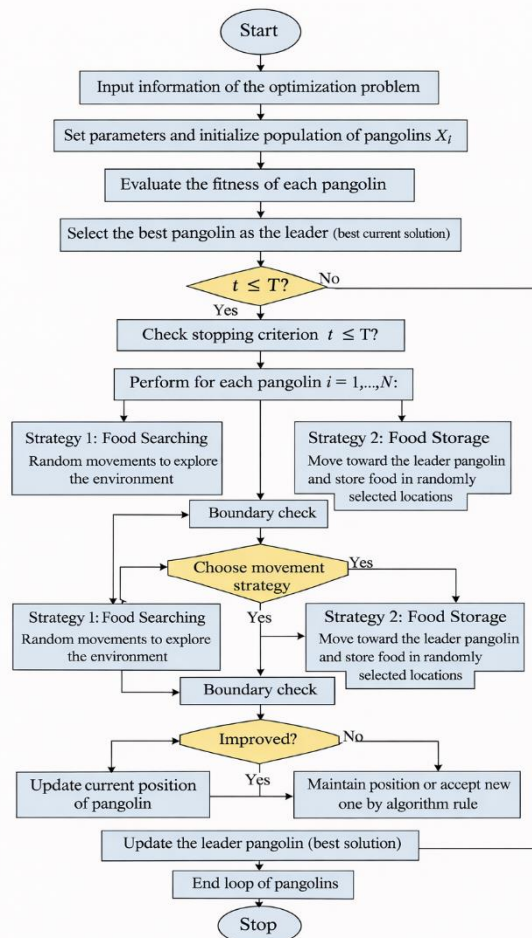
- مناسب برای فرار از مینیمم‌های محلی
 - عملکرد پایدار در مسائل حرارتی
- دیگرام الگوریتم بهینه‌سازی جستجوی پانگولین در شکل 3 نشان می‌دهد که این الگوریتم شامل چندین مرحله اصلی زیر است:
- شروع و ساخت جمعیت اولیه
 - ارزیابی تابع هدف و جریمه قیود
 - انتخاب بهترین جواب B

- ساخت حرکت مارپیچی (Spiral) حول بهترین/بهترین‌های فعلی (حرکت‌های غیرخطی برای فرار از کمینه محلی)
- آپدیت موقعیت با الگوی مارپیچ که معمولاً با گذشت زمان شدت آن تغییر می‌کند
- کنترل مرز اگر خارج شد اصلاح
- ذخیره بهترین
- تکرار تا پایان

الگوریتم بهینه‌سازی جستجوی پانگولین یکی از روش‌های فراابتکاری الهام‌گرفته از رفتار طبیعی این حیوان در جستجوی غذا و حرکت‌های مارپیچی است. در این الگوریتم، هر راه‌حل به‌عنوان یک پانگولین در فضای جستجو مدل‌سازی می‌شود و موقعیت آن با معادله‌ای به‌روزرسانی می‌گردد که شامل دو بخش اصلی است: حرکت به سمت بهترین جواب موجود (B^t) و یک مؤلفه مارپیچی (Spiral) که شدت آن با گذشت زمان کاهش می‌یابد. ضریب η میزان حرکت و ضریب λ نرخ کاهش شدت مارپیچ را کنترل می‌کنند. این ساختار باعث می‌شود الگوریتم توانایی بالایی در فرار از کمینه‌های محلی داشته باشد و در عین حال عملکرد پایدار خود را در مسائل پیچیده مانند کنترل حرارتی دیتاسنتر حفظ کند.

ویژگی مهم این الگوریتم، استفاده از حرکت‌های غیرخطی مارپیچی است که امکان جستجوی گسترده‌تر در فضای راه‌حل‌ها را فراهم می‌کند. برخلاف حرکت‌های خطی ساده، مسیر مارپیچی به الگوریتم اجازه می‌دهد تا نواحی جدیدی از فضای جستجو را کشف کند و از گیر افتادن در نقاط بهینه محلی جلوگیری نماید. همین ویژگی موجب شده است که الگوریتم پانگولین در مسائل چندقیدی و به‌ویژه در مدل‌های حرارتی دیتاسنتر عملکردی پایدار و قابل اعتماد داشته باشد.

فرآیند اجرای الگوریتم شامل چند مرحله اصلی است. ابتدا جمعیت اولیه پانگولین‌ها در محدوده مجاز متغیرها ساخته می‌شود. سپس تابع هدف برای همه افراد ارزیابی شده و در صورت وجود قیود، جریمه‌های مربوطه اعمال می‌گردد. بهترین جواب انتخاب می‌شود و حرکت مارپیچی حول آن ساخته می‌شود. موقعیت هر عضو با ترکیب حرکت به سمت بهترین و الگوی مارپیچ به‌روزرسانی می‌شود که شدت آن معمولاً با گذشت زمان کاهش می‌یابد. در ادامه، کنترل مرزی انجام می‌شود تا متغیرها در محدوده مجاز باقی بمانند. بهترین جواب در هر تکرار ذخیره شده و فرآیند تا رسیدن به شرط توقف ادامه می‌یابد. این چرخه باعث می‌شود الگوریتم به‌تدریج به سمت راه‌حل‌های بهینه یا نزدیک بهینه حرکت کند.



شکل ۴: فلوجارت الگوریتم بهینه‌سازی جستجوی پانگولین

۵-۳- الگوریتم بهینه‌سازی خفاش میوه‌خوار (FBO)

الگوریتم FBO از مکانیزم مکان‌یابی صوتی خفاش‌ها الهام گرفته شده است. این الگوریتم با تنظیم تطبیقی پارامترهای شدت صوت و نرخ انتشار پالس، قابلیت جستجوی پویا در فضای جواب را فراهم می‌کند. با این حال، در مسئله سرمایه‌گذاری دیتاسنتر، عملکرد FBO در کنترل هم‌زمان مصرف انرژی و قیود حرارتی نسبتاً محدود است. نوابری صوتی خفاش‌های میوه‌خوار است. این الگوریتم نخستین‌بار توسط یانگ معرفی شد و به‌عنوان یک الگوریتم فراابتکاری الهام‌گرفته از طبیعت شناخته می‌شود [۱۱].

معادله به‌روزرسانی:

$$X_i^{t+1} = X_i^t + A_i^t (B^t - X_i^t) \quad (15)$$

که A_i^t شدت صوت است.

ویژگی‌ها:

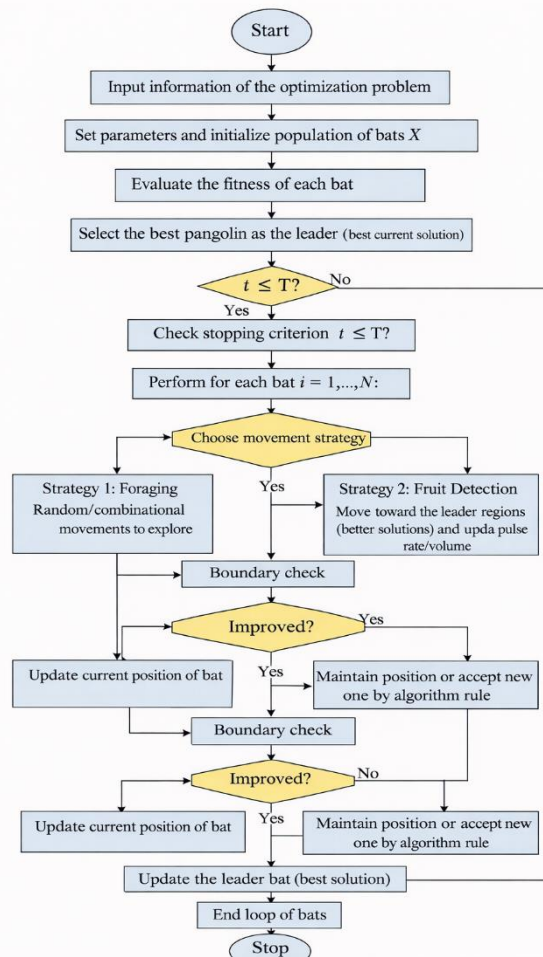
- جستجوی دقیق
- مناسب برای تنظیمات حساس دما
- دیگرام الگوریتم بهینه‌سازی خفاش میوه‌خوار در شکل زیر نشان می‌دهد که این الگوریتم شامل چندین مرحله اصلی زیر است:
- شروع و ایجاد جمعیت اولیه خفاش‌ها
- ارزیابی تابع هدف و جریمه قیود

- انتخاب بهترین جواب B
- برای هر خفاش: تعیین پارامتر شدت/گام حرکت (مثل A یا نرخ تغییر)
- آپدیت موقعیت: حرکت به سمت بهترین با شدت کنترل شده (بهره‌برداری قوی)
- کنترل مرزی
- به‌روزرسانی Best
- تکرار تا شرط توقف

الگوریتم بهینه‌سازی خفاش میوه‌خوار یکی از روش‌های فراابتکاری الهام‌گرفته از رفتار خفاش‌ها در شکار با استفاده از امواج صوتی است. در این الگوریتم، هر خفاش به‌عنوان یک راه‌حل در فضای جستجو در نظر گرفته می‌شود و موقعیت آن با معادله‌ای به‌روزرسانی می‌شود که شامل حرکت به سمت بهترین جواب موجود (B^t) و شدت صوت (A_i^t) است. شدت صوت در اینجا نقش کلیدی دارد، زیرا میزان گام حرکت و دقت جستجو را کنترل می‌کند. این ویژگی باعث می‌شود الگوریتم بتواند جستجویی دقیق انجام دهد و برای تنظیمات حساس مانند کنترل دما در دیتاسنترها بسیار مناسب باشد.

ویژگی مهم این الگوریتم، توانایی آن در بهره‌برداری قوی از بهترین جواب‌های موجود است. با تنظیم شدت صوت، خفاش‌ها می‌توانند حرکت‌های کوچک و دقیق به سمت بهترین راه‌حل داشته باشند و در نتیجه نرخ همگرایی بالایی به دست آید. این قابلیت به‌ویژه در مسائلی اهمیت دارد، زیرا کوچک‌ترین تغییر در تنظیمات دما می‌تواند تأثیر بزرگی بر عملکرد تجهیزات داشته باشد.

فرآیند اجرای الگوریتم شامل چند مرحله اصلی است. ابتدا جمعیت اولیه خفاش‌ها ایجاد می‌شود و سپس تابع هدف برای همه افراد ارزیابی شده و در صورت وجود قیود، جریمه‌های مربوطه اعمال می‌گردد. بهترین جواب انتخاب می‌شود و برای هر خفاش شدت صوت یا گام حرکت تعیین می‌شود. موقعیت‌ها با حرکت کنترل شده به سمت بهترین جواب به‌روزرسانی می‌شوند و در صورت خروج از محدوده، اصلاح مرزی انجام می‌گیرد. بهترین جواب در هر تکرار ذخیره شده و فرآیند تا رسیدن به شرط توقف ادامه می‌یابد. این چرخه باعث می‌شود الگوریتم به‌تدریج به سمت راه‌حل‌های دقیق و پایدار حرکت کند و در نهایت یک جواب بهینه یا نزدیک بهینه ارائه دهد.



شکل ۵: فلوچارت الگوریتم بهینه‌سازی خفاش میوه‌خوار

۳-۶- الگوریتم بهینه‌سازی خرچنگ نعل‌اسبی (HCO)

الگوریتم HCO مبتنی بر رفتار دفاعی و حرکتی خرچنگ نعل‌اسبی طراحی شده است. این الگوریتم از پایداری بالایی برخوردار بوده و برای مسائل چندقیدی گزینه‌ای مناسب محسوب می‌شود. HCO در حفظ قیود حرارتی عملکرد قابل قبولی ارائه می‌دهد، هرچند سرعت همگرایی آن متوسط ارزیابی می‌شود. رفتار دفاعی و حرکتی خرچنگ نعل‌اسبی است. این الگوریتم با الهام از مکانیزم‌های دفاعی و حرکتی خرچنگ نعل‌اسبی برای حل مسائل مقید پیشنهاد شده است [۱۲].

معادله حرکت:

$$X_i^{t+1} = X_i^t + \rho \text{randn}() + \sigma(B^t - X_i^t) \quad (16)$$

ویژگی‌ها:

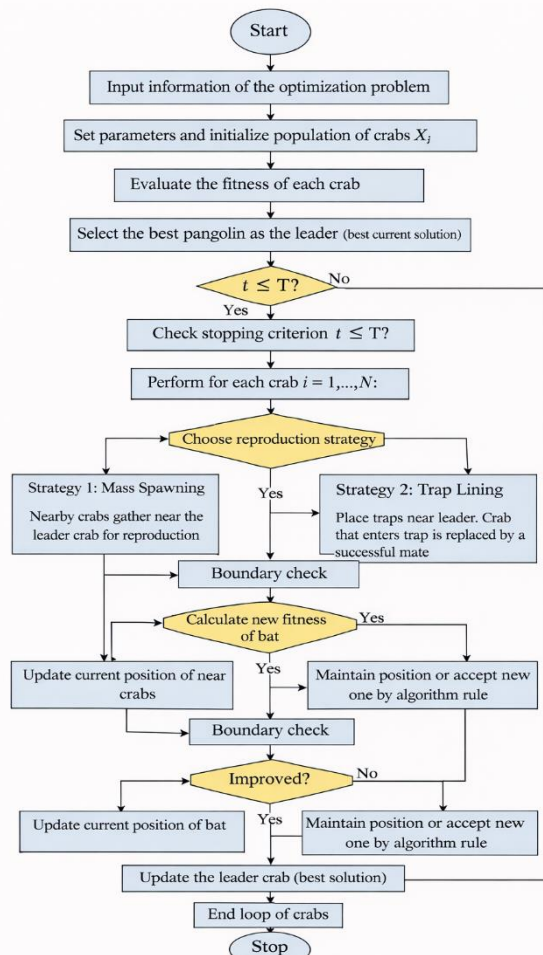
- پایداری بالا
- مناسب برای مسائل چندقیدی
- دیگرام الگوریتم بهینه‌سازی خرچنگ نعل‌اسبی در شکل زیر نشان می‌دهد که این الگوریتم شامل چندین مرحله اصلی زیر است:
- شروع و تولید جمعیت اولیه
- ارزیابی تابع هدف و جریمه قیود
- انتخاب بهترین جواب B

- تولید نویز تصادفی (معمولاً گاوسی $randn$) برای ایجاد اکتشاف
- آپدیت موقعی: ترکیب حرکت تصادفی + حرکت به سمت بهترین (اکتشاف + بهره‌برداری)
- کنترل مرزی
- ذخیره بهترین
- تکرار تا توقف

الگوریتم بهینه‌سازی خرچنگ نعل‌اسبی یکی از روش‌های فراابتکاری الهام‌گرفته از رفتار طبیعی این موجودات در حرکت و جستجو است. در این الگوریتم، هر خرچنگ به‌عنوان یک راه‌حل در فضای جستجو مدل‌سازی می‌شود و موقعیت آن با معادله‌ای به‌روزرسانی می‌گردد که شامل دو بخش اصلی است: یک مؤلفه تصادفی مبتنی بر نویز گاوسی $\rho \cdot randn$ برای ایجاد اکتشاف، و یک مؤلفه هدایت‌شده به سمت بهترین جواب موجود $B^t - X_i^t$ برای بهره‌برداری. این ترکیب باعث می‌شود الگوریتم هم پایداری بالایی داشته باشد و هم توانایی جستجوی گسترده در فضای راه‌حل‌ها را حفظ کند، به‌ویژه در مسائل چندقیدی که نیازمند تعادل دقیق بین اکتشاف و بهره‌برداری هستند.

ویژگی مهم این الگوریتم، توانایی آن در ایجاد تنوع جمعیت از طریق نویز تصادفی است. این نویز باعث می‌شود الگوریتم از گیر افتادن در بهینه‌های محلی جلوگیری کند و در عین حال حرکت به سمت بهترین جواب موجود، سرعت همگرایی را افزایش دهد. همین ویژگی‌ها موجب شده است که الگوریتم خرچنگ نعل‌اسبی برای مسائل پیچیده مانند مدل‌های حرارتی دیتاسنتز بسیار مناسب باشد، زیرا می‌تواند هم‌زمان مصرف انرژی را کاهش دهد و شرایط ایمن حرارتی را رعایت کند.

فرآیند اجرای الگوریتم شامل چند مرحله اصلی است. ابتدا جمعیت اولیه خرچنگ‌ها تولید می‌شود و سپس تابع هدف برای همه افراد ارزیابی شده و در صورت وجود قیود، جریمه‌های مربوطه اعمال می‌گردد. بهترین جواب انتخاب می‌شود و نویز تصادفی برای ایجاد اکتشاف تولید می‌گردد. موقعیت هر عضو با ترکیب حرکت تصادفی و حرکت به سمت بهترین به‌روزرسانی می‌شود. در ادامه، کنترل مرزی انجام می‌شود تا متغیرها در محدوده مجاز باقی بمانند. بهترین جواب در هر تکرار ذخیره شده و فرآیند تا رسیدن به شرط توقف ادامه می‌یابد. این چرخه باعث می‌شود الگوریتم به تدریج به سمت راه‌حل‌های پایدار و نزدیک بهینه حرکت کند.



شکل ۶: فلوجارت الگوریتم بهینه‌سازی خرچنگ نعل‌اسبی

۷-۳- الگوریتم بهینه‌سازی شاهین بیابانی (DFO)

الگوریتم DFO با الهام از الگوی شکار شاهین بیابانی توسعه یافته است. تمرکز اصلی این الگوریتم بر همگرایی سریع و کاهش زمان دستیابی به جواب بهینه است. این ویژگی، DFO را برای کاربردهای عملی که نیازمند تصمیم‌گیری سریع هستند، مناسب می‌سازد. شکار سریع شاهین بیابانی است. رفتار شکار سریع و همگرایی پرشتاب شاهین صحرائی، مبنای طراحی این الگوریتم برای مسائل با نیاز به همگرایی سریع بوده است [۱۳].

معادله حمله:

$$X_i^{t+1} = B^t + v \cdot (X_i^t - B^t) \quad (17)$$

که v سرعت حمله است.

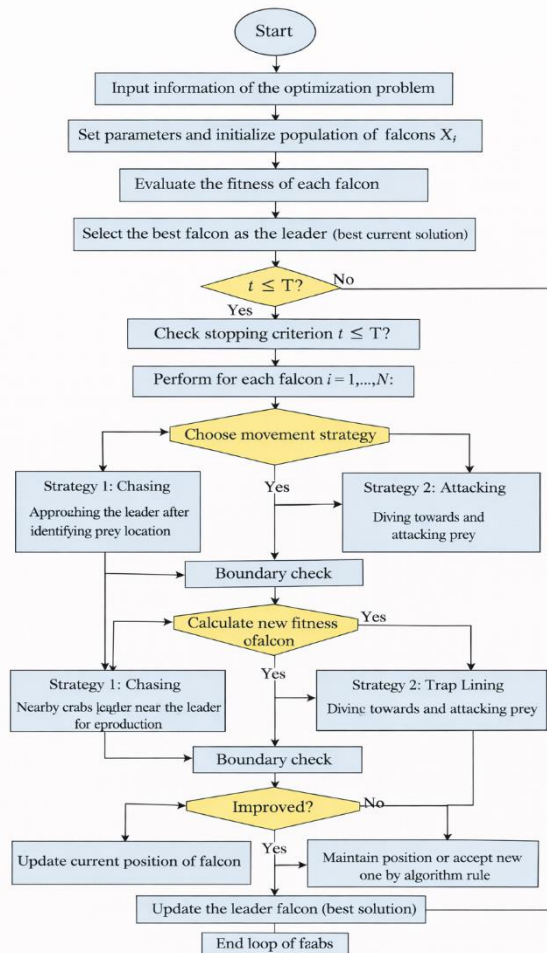
ویژگی‌ها:

- همگرایی سریع
- مناسب برای کاهش انرژی فن‌ها
- دیگرام الگوریتم بهینه‌سازی شاهین بیابانی در شکل زیر نشان می‌دهد که این الگوریتم شامل چندین مرحله اصلی زیر است:
- شروع و ساخت جمعیت اولیه
- ارزیابی تابع هدف و قیود/جریمه

- یافتن بهترین جواب B
- تنظیم پارامتر سرعت/شدت حمله ν روی سرعت همگرایی اثر دارد
- آپدیت موقعیت : هر فرد با یک قانون "نزدیک‌شدن" کنترل‌شده به بهترین حرکت می‌کند (همگرایی سریع‌تر)
- کنترل مرزی
- به‌روزرسانی Best
- پایان/تکرار

الگوریتم بهینه‌سازی شاهین بیابانی یکی از روش‌های فراابتکاری الهام‌گرفته از رفتار شکارچیان در طبیعت است که برای حل مسائل پیچیده بهینه‌سازی طراحی شده است. در این الگوریتم، هر فرد در جمعیت به‌عنوان یک راه‌حل در فضای جستجو در نظر گرفته می‌شود و موقعیت آن با معادله حمله به‌روزرسانی می‌گردد. این معادله نشان می‌دهد که حرکت هر فرد ترکیبی از موقعیت فعلی و بهترین جواب موجود (B^t) است، که با ضریب سرعت حمله ν کنترل می‌شود. این ساختار باعث می‌شود الگوریتم نرخ همگرایی بسیار بالایی داشته باشد و بتواند به سرعت به سمت بهترین جواب حرکت کند. همین ویژگی آن را برای کاربردهایی مانند کاهش انرژی مصرفی فن‌ها در دیتاسنترها بسیار مناسب می‌سازد.

ویژگی مهم این الگوریتم، توانایی آن در نزدیک‌شدن سریع و کنترل‌شده به بهترین جواب است. با تنظیم پارامتر سرعت حمله ν ، می‌توان میزان شدت حرکت افراد به سمت رهبر را کنترل کرد. اگر ν بزرگ‌تر انتخاب شود، همگرایی سریع‌تر رخ می‌دهد اما خطر گیر افتادن در بهینه‌های محلی افزایش می‌یابد؛ در حالی که انتخاب مقادیر کوچک‌تر برای ν امکان جستجوی دقیق‌تر و پایدارتر را فراهم می‌کند. این انعطاف‌پذیری باعث می‌شود الگوریتم بتواند تعادل مناسبی بین سرعت همگرایی و کیفیت جواب برقرار کند. فرآیند اجرای الگوریتم شامل چند مرحله اصلی است. ابتدا جمعیت اولیه ساخته می‌شود و سپس تابع هدف برای همه افراد ارزیابی شده و در صورت وجود قیود، جریمه‌های مربوطه اعمال می‌گردد. بهترین جواب یا رهبر انتخاب می‌شود و پارامتر سرعت حمله تنظیم می‌گردد. موقعیت هر فرد با قانون نزدیک‌شدن به بهترین به‌روزرسانی می‌شود و در صورت خروج از محدوده، اصلاح مرزی انجام می‌گیرد. بهترین جواب در هر تکرار ذخیره شده و فرآیند تا رسیدن به شرط توقف ادامه می‌یابد. این چرخه باعث می‌شود الگوریتم به تدریج به سمت راه‌حل‌های بهینه یا نزدیک بهینه حرکت کند و در نهایت یک جواب کارآمد برای مسئله ارائه دهد.



شکل ۷: فلوچارت الگوریتم بهینه‌سازی شاهین بیابانی

۸-۳- الگوریتم بهینه‌سازی مهاجرت لاک پشت دریایی (STMO)

الگوریتم STMO بر اساس الگوی مهاجرت طولانی مدت لاک پشت‌های دریایی طراحی شده است. این الگوریتم قابلیت جستجوی گسترده در فضای جواب را دارا بوده و برای مسائل چندهدفه مناسب است. با این حال، در مدل حرارتی دیتاسنتر، عملکرد آن نسبت به الگوریتم‌های برتر نظیر GSSO و NBIO ضعیف‌تر ارزیابی شده است. مهاجرت طولانی لاک پشت‌های دریایی است. الگوی مهاجرت طولانی مدت لاک پشت‌های دریایی به‌عنوان مبنای طراحی این الگوریتم چندهدفه در نظر گرفته شده است [۱۴].

معادله مهاجرت:

$$X_i^{t+1} = X_i^t + \mu(G^t - X_i^t) + \kappa \text{rand}() \quad (18)$$

ویژگی‌ها:

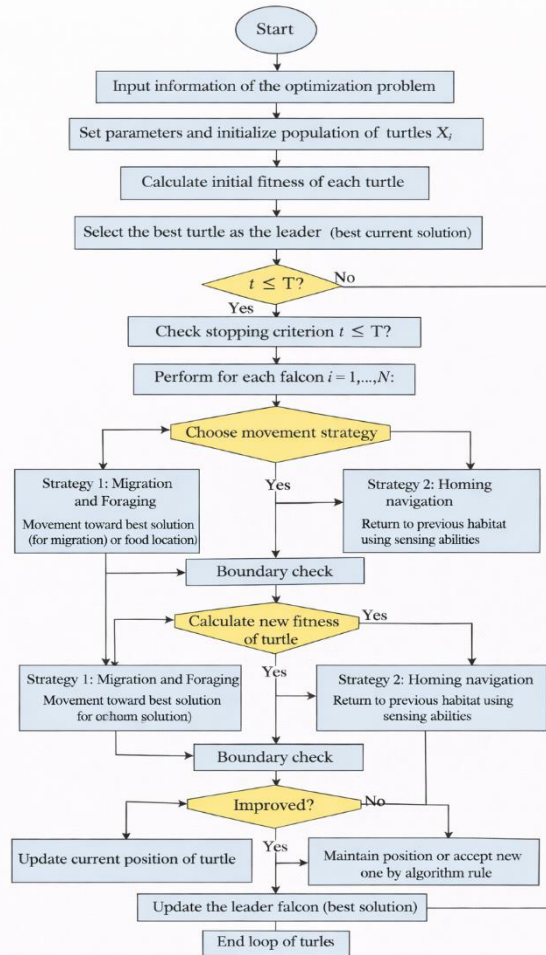
- جستجوی گسترده
- مناسب برای مسائل با چند مینیمم محلی
- دیگرام الگوریتم مهاجرت لاک پشت دریایی در شکل زیر نشان می‌دهد که این الگوریتم شامل چندین مرحله اصلی زیر است:
- شروع و ایجاد جمعیت اولیه لاک پشت‌ها
- ارزیابی تابع هدف و جریمه قیود
- تعیین بهترین گروه بهترین کل G

- تولید یک جزء تصادفی برای پوشش فضای جست‌وجو
- آپدیت موقعیت : مهاجرت به سمت بهترین + اغتشاش تصادفی (تعادل مسیر مهاجرت و تنوع)
- کنترل مرزها
- ذخیره بهترین
- تکرار تا شرط توقف

الگوریتم بهینه‌سازی مهاجرت لاک‌پشت دریایی یکی از روش‌های فراابتکاری الهام‌گرفته از رفتار طبیعی این موجودات در مهاجرت‌های طولانی و گروهی است. در این الگوریتم، هر لاک‌پشت به‌عنوان یک راه‌حل در فضای جست‌وجو در نظر گرفته می‌شود و موقعیت آن با معادله‌ای به‌روزرسانی می‌گردد که شامل دو بخش اصلی است: حرکت به سمت بهترین گروه یا رهبر (G^t) و یک مؤلفه تصادفی ($\kappa \cdot rand()$) برای ایجاد تنوع در مسیر جست‌وجو. ضریب μ شدت حرکت به سمت بهترین را کنترل می‌کند و ترکیب این دو مؤلفه باعث می‌شود الگوریتم توانایی بالایی در جست‌جوی گسترده داشته باشد و بتواند از گیر افتادن در مینیمم‌های محلی جلوگیری کند.

ویژگی مهم این الگوریتم، تعادل میان بهره‌برداری از بهترین جواب‌های موجود و اکتشاف مسیرهای جدید است. حرکت به سمت رهبر جمعیت موجب همگرایی سریع‌تر می‌شود، در حالی که مؤلفه تصادفی تنوع لازم را ایجاد کرده و امکان کشف نواحی جدید از فضای راه‌حل‌ها را فراهم می‌سازد. همین ویژگی‌ها موجب شده است که الگوریتم مهاجرت لاک‌پشت دریایی برای مسائل پیچیده با چندین مینیمم محلی بسیار مناسب باشد، به‌ویژه در کاربردهایی مانند مدل‌های حرارتی دیتاسنتز که نیازمند یافتن جواب‌های پایدار و کارآمد هستند.

فرآیند اجرای الگوریتم شامل چند مرحله اصلی است. ابتدا جمعیت اولیه لاک‌پشت‌ها در محدوده مجاز متغیرها ایجاد می‌شود. سپس تابع هدف برای همه افراد ارزیابی شده و در صورت وجود قیود، جریمه‌های مربوطه اعمال می‌گردد. بهترین گروه یا رهبر انتخاب می‌شود و یک مؤلفه تصادفی برای پوشش بهتر فضای جست‌وجو تولید می‌گردد. موقعیت هر عضو با ترکیب حرکت به سمت بهترین و اغتشاش تصادفی به‌روزرسانی می‌شود. در ادامه، کنترل مرزی انجام می‌شود تا متغیرها در محدوده مجاز باقی بمانند. بهترین جواب در هر تکرار ذخیره شده و فرآیند تا رسیدن به شرط توقف ادامه می‌یابد. این چرخه باعث می‌شود الگوریتم به تدریج به سمت راه‌حل‌های بهینه یا نزدیک بهینه حرکت کند.



شکل ۸: فلوجارت الگوریتم بهینه‌سازی مهاجرت لاک‌پشت دریایی

۴- روش ارزیابی و تنظیمات آزمایش (Evaluation Methodology):

در این بخش، چارچوب ارزیابی عملکرد الگوریتم‌های فراابتکاری معرفی شده در بخش قبل تشریح می‌شود. هدف اصلی از این ارزیابی، مقایسه منصفانه و جامع الگوریتم‌ها از نظر دقت، نرخ همگرایی، پایداری حرارتی و میزان مصرف انرژی در مسئله بهینه‌سازی سیستم سرمایه‌دیتاسنتر است. به منظور افزایش اعتبار نتایج و کاهش سوگیری، فرآیند ارزیابی در دو سطح مجزا شامل بنچمارک‌های استاندارد و مدل حرارتی واقعی دیتاسنتر انجام شده است. در این پژوهش از مجموعه توابع معیار CEC برای تضمین مقایسه منصفانه و قابل تکرار استفاده شده است [۱۵].

۴-۱- ساختار کلی چارچوب ارزیابی

چارچوب ارزیابی پیشنهادی شامل دو دسته آزمایش مستقل به شرح زیر است:

ارزیابی بر روی توابع بنچمارک استاندارد: این مرحله با هدف بررسی رفتار عمومی الگوریتم‌ها از نظر توانایی اکتشاف فضای جستجو، بهره‌برداری از نواحی امیدبخش و سرعت همگرایی انجام می‌شود.

ارزیابی بر روی مدل واقعی سیستم سرمایه‌دیتاسنتر: در این مرحله، عملکرد الگوریتم‌ها در یک مسئله عملی و مهندسی با قیود حرارتی واقعی مورد بررسی قرار می‌گیرد.

این رویکرد دو مرحله‌ای امکان تحلیل دقیق‌تر نقاط قوت و ضعف هر الگوریتم را فراهم می‌سازد.

۴-۲- ارزیابی با استفاده از بنچمارک‌های CEC-2017

در گام نخست، مجموعه توابع بنچمارک استاندارد CEC-2017 برای ارزیابی الگوریتم‌های فراابتکاری مورد استفاده قرار گرفته‌اند. این بنچمارک‌ها شامل توابع تک‌قله‌ای، چندقله‌ای، ترکیبی و چرخیده هستند و به‌طور گسترده به‌عنوان معیار مرجع در مقایسه الگوریتم‌های بهینه‌سازی به کار می‌روند. هر الگوریتم برای هر تابع بنچمارک در چندین اجرای مستقل اجرا شده و شاخص‌های آماری زیر استخراج گردیده است:

- بهترین مقدار تابع هدف (Best)
- مقدار میانگین نتایج (Mean)
- انحراف معیار (Std)

- رتبه‌بندی الگوریتم‌ها بر اساس عملکرد کلی

این معیارها امکان بررسی دقت، پایداری و قابلیت اطمینان الگوریتم‌ها را فراهم می‌کنند.

۴-۳- ارزیابی بر روی مدل حرارتی واقعی دیتاسنتر

در مرحله دوم، الگوریتم‌ها بر روی یک مدل حرارتی واقع‌گرایانه از دیتاسنتر ارزیابی شده‌اند. این مدل شامل اجزای اصلی دیتاسنتر نظیر رک‌های سرور، واحدهای سرمایش (ACU)، راهروهای سرد و گرم و مسیرهای جریان هوا است. رفتار حرارتی سیستم با استفاده از یک مدل دینامیکی مبتنی بر معادلات دیفرانسیل معمولی (ODE) شبیه‌سازی می‌شود. هدف اصلی در این بخش، کمینه‌سازی مصرف انرژی سیستم سرمایه‌دیتاسنتر با رعایت قیود حرارتی سخت تعریف‌شده در بخش مدل‌سازی مسئله است. الگوریتم‌ها باید ضمن کاهش توان مصرفی فن‌ها و سیستم آب سرد، دمای نقاط بحرانی دیتاسنتر را در محدوده مجاز نگه دارند. بر اساس مدل حرارتی مقاله ICLR 2025:

- ۱۱ واحد سرمایشی (ACU)

- ۳۶ رک سرور

- ۳۳ حسگر دما و رطوبت
 - مدل TTDM برای محاسبه دماها
- هدف :

- کمینه‌سازی مصرف انرژی سرمایه‌گذاری
- حفظ دما در محدوده ایمن

هدف اصلی این مدل‌سازی و بهینه‌سازی، کاهش مصرف انرژی سیستم سرمایه‌گذاری در کنار حفظ دما در محدوده ایمن برای تجهیزات است. از آنجا که افزایش دما می‌تواند منجر به کاهش عمر سرورها و حتی خاموشی اضطراری شود، رعایت قیود حرارتی اهمیت حیاتی دارد. در مقابل، مصرف بیش از حد انرژی توسط واحدهای سرمایه‌گذاری نیز هزینه‌های عملیاتی را افزایش می‌دهد. بنابراین، مسئله بهینه‌سازی در این پژوهش به دنبال یافتن تعادلی میان این دو هدف است: کاهش انرژی مصرفی بدون نقض شرایط ایمن حرارتی.

این چارچوب نه تنها امکان ارزیابی عملکرد الگوریتم‌های فراابتکاری مختلف را فراهم می‌کند، بلکه نشان می‌دهد چگونه می‌توان با استفاده از مدل TTDM و داده‌های واقعی حسگرها، یک سیستم سرمایه‌گذاری پایدار و کارآمد برای دیتاسنتر طراحی کرد.

۴-۴- معیارهای ارزیابی عملکرد الگوریتم‌ها (Performance Metrics)

برای مقایسه منصفانه و دقیق عملکرد الگوریتم‌های فراابتکاری، مجموعه‌ای از معیارهای کلیدی در نظر گرفته شده است که هر کدام جنبه‌ای مهم از کیفیت و کارایی الگوریتم‌ها را نشان می‌دهند. نخستین معیار، دقت (Accuracy) است که توانایی الگوریتم در دستیابی به مقادیر بهینه تابع هدف را اندازه‌گیری می‌کند. در این بخش، بهترین مقدار تابع هدف با رابطه‌ی $Best = \min f(x)$ مشخص می‌شود و نشان می‌دهد کدام الگوریتم توانسته به کمترین مقدار ممکن دست یابد.

معیار دوم، نرخ همگرایی (Convergence Rate) است که سرعت رسیدن الگوریتم به جواب بهینه را بررسی می‌کند. این معیار معمولاً با رسم نمودار همگرایی برای هر الگوریتم محاسبه می‌شود و نشان می‌دهد کدام روش سریع‌تر به جواب مناسب نزدیک می‌شود. معیار سوم، پایداری حرارتی (Thermal Stability) است که میزان موفقیت الگوریتم در حفظ دما در محدوده مجاز بدون نقض قیود را نشان می‌دهد. برای این منظور، میانگین و انحراف معیار دماها محاسبه می‌شود تا مشخص شود الگوریتم تا چه حد توانسته شرایط ایمن حرارتی را پایدار نگه دارد.

از سوی دیگر، مصرف انرژی (Energy Consumption) یکی از معیارهای حیاتی است که مقدار کل انرژی مصرفی سیستم سرمایه‌گذاری را اندازه‌گیری می‌کند. این معیار اهمیت ویژه‌ای دارد زیرا کاهش مصرف انرژی در کنار حفظ شرایط ایمن حرارتی هدف اصلی بهینه‌سازی است. نهایتاً، معیار توانایی فرار از کمینه‌های محلی یا کارایی محاسباتی (Computational Efficiency) بررسی می‌شود. این معیار حساسیت الگوریتم نسبت به شرایط اولیه را نشان می‌دهد و شامل زمان اجرا، تعداد فراخوانی‌های تابع هدف و پیچیدگی محاسباتی است. در مجموع، این مجموعه معیارها چارچوبی جامع برای ارزیابی و مقایسه عملکرد الگوریتم‌ها فراهم می‌آورد و امکان تحلیل دقیق نقاط قوت و ضعف هر روش را ایجاد می‌کند.

- دقت (Accuracy) : توانایی الگوریتم در دستیابی به مقادیر بهینه تابع هدف و بهترین مقدار تابع هدف :

$$Best = \min f(x)$$

- نرخ همگرایی (Convergence Rate) : سرعت رسیدن به جواب بهینه بر اساس نمودار همگرایی برای هر الگوریتم رسم می‌شود.
- پایداری حرارتی (Thermal Stability) : میزان موفقیت الگوریتم در حفظ دما در محدوده مجاز بدون نقض قیود و میانگین و انحراف معیار :

Mean, Std

- مصرف انرژی (Energy Consumption) : مقدار کل انرژی مصرفی سیستم سرمایه‌ش
- توانایی فرار از کمینه‌های محلی یا کارایی محاسباتی (Computational Efficiency) : بررسی حساسیت الگوریتم نسبت به شرایط اولیه شامل :
 ۱. زمان اجرا
 ۲. تعداد فراخوانی تابع هدف
 ۳. پیچیدگی محاسبات

۵- روش پژوهش (Research Methodology) :

در این پژوهش، یک چارچوب نظام‌مند و چندمرحله‌ای برای تحلیل عملکرد و مقایسه الگوریتم‌های فراابتکاری در مسئله طراحی بهینه سیستم سرمایش دیتاسنتر تحت قیود سخت حرارتی ارائه می‌شود. روش پژوهش به گونه‌ای طراحی شده است که ضمن تضمین انصاف در مقایسه الگوریتم‌ها، اعتبار آماری نتایج و قابلیت تکرارپذیری آزمایش‌ها را فراهم آورد. شکل کلی روش پژوهش شامل مراحل مدل‌سازی مسئله، پیاده‌سازی الگوریتم‌ها، ارزیابی بنچمارک‌ها، ارزیابی مهندسی و تحلیل آماری نتایج است.

۵-۱- چارچوب کلی پژوهش

چارچوب پژوهش حاضر از پنج گام اصلی تشکیل شده است :

- مدل‌سازی ریاضی و حرارتی سیستم سرمایش دیتاسنتر با استفاده از مدل دینامیکی حرارتی مبتنی بر ODE و ساختار گرافی؛
 - فرم‌بندی مسئله بهینه‌سازی مقید شامل تعریف متغیرهای تصمیم، تابع هدف و قیود حرارتی؛
 - پیاده‌سازی و تنظیم الگوریتم‌های فراابتکاری منتخب با شرایط اولیه و پارامترهای یکسان؛
 - ارزیابی عملکرد الگوریتم‌ها ابتدا روی بنچمارک‌های استاندارد CEC و سپس روی مدل واقعی دیتاسنتر؛
 - تحلیل آماری و مقایسه نتایج با استفاده از آزمون‌های معتبر ناپارامتری.
- این رویکرد مرحله‌ای امکان بررسی هم‌زمان رفتار نظری الگوریتم‌ها و کارایی عملی آن‌ها در یک سناریوی مهندسی واقعی را فراهم می‌کند.

۵-۲- نوع پژوهش و رویکرد حل مسئله

پژوهش حاضر از نوع کاربردی-تحلیلی بوده و با رویکرد شبیه‌سازی و بهینه‌سازی عددی انجام شده است. مسئله طراحی سیستم سرمایش دیتاسنتر به عنوان یک مسئله بهینه‌سازی غیرخطی، چندقیدی و با ابعاد بالا در نظر گرفته شده که حل آن با روش‌های تحلیلی کلاسیک امکان‌پذیر نیست. از این رو، الگوریتم‌های فراابتکاری به عنوان ابزار اصلی جستجوی فضای جواب مورد استفاده قرار گرفته‌اند. در این پژوهش، تمرکز بر بهینه‌سازی آفلاین (Offline Optimization) است و مدل حرارتی دیتاسنتر صرفاً به عنوان یک محیط شبیه‌سازی واقع‌گرا برای ارزیابی راه‌حل‌ها به کار گرفته می‌شود، نه به عنوان یک مدل یادگیرنده می‌باشد.

۵-۳- انتخاب الگوریتم‌های فراابتکاری

در این پژوهش، هشت الگوریتم فراابتکاری نسل جدید که در بازه زمانی ۲۰۲۵ معرفی شده‌اند، به منظور انجام مقایسه انتخاب گردیده‌اند. معیارهای انتخاب این الگوریتم‌ها شامل تنوع در مکانیسم‌های جستجو از جمله رفتار جمعی، شکار، مهاجرت، استتار و هوش اجتماعی بوده است. همچنین گزارش عملکرد موفق این الگوریتم‌ها در حل مسائل مهندسی مقید و غیرخطی، قابلیت مقیاس‌پذیری در مسائل با ابعاد بالا، و تازگی علمی و کاربرد آن‌ها در پژوهش‌های مرجع اخیر از دیگر دلایل انتخاب بوده است. الگوریتم‌های مورد بررسی شامل مرغ شاخدار شمالی، کلاغ نوک‌سرخ، ماهی مرکب غول‌پیکر، پانگولین، خفاش میوه‌خوار، خرچنگ نعل‌اسبی، شاهین بیابانی و لاک‌پشت دریایی مهاجر می‌باشند. تمامی این الگوریتم‌ها در فضای جستجوی ۲۲ بعدی مسئله مورد نظر اجرا شده‌اند تا امکان مقایسه‌ای دقیق و منصفانه فراهم شود. انتخاب این مجموعه متنوع از الگوریتم‌ها باعث می‌شود نتایج پژوهش نه تنها از نظر علمی معتبر باشند، بلکه از نظر کاربردی نیز قابلیت تعمیم به مسائل واقعی را داشته باشند.

۵-۴- پیاده‌سازی مسئله بهینه‌سازی

برای هر الگوریتم، فرآیند بهینه‌سازی بر اساس یک رویه یکسان انجام شده است تا شرایط مقایسه منصفانه برقرار باشد. در ابتدا، جمعیت اولیه به صورت تصادفی و در محدوده مجاز متغیرهای تصمیم ایجاد می‌شود. سپس هر راه‌حل با محاسبه مقدار تابع هدف و دماهای حاصل از مدل حرارتی دیتاسنتر مورد ارزیابی قرار می‌گیرد. در ادامه، قیود حرارتی بررسی شده و در صورت نقض این قیود، با استفاده از روش جریمه (Penalty Method) افزایش متناسبی در مقدار تابع هدف اعمال می‌شود تا شناس انتخاب این راه‌حل‌ها در نسل‌های بعدی کاهش یابد. پس از آن، موقعیت عوامل جستجو بر اساس قواعد اختصاصی هر الگوریتم به‌روزرسانی شده و این فرآیند تا رسیدن به شرط توقف، که در این پژوهش حداکثر تعداد تکرارها در نظر گرفته شده است، به صورت تکراری ادامه می‌یابد. این رویکرد یکپارچه تضمین می‌کند که نتایج حاصل از الگوریتم‌ها قابل مقایسه و تحلیل دقیق باشند.

۵-۵- تنظیمات آزمایش و شرایط تکرارپذیری

به منظور تضمین مقایسه‌ای منصفانه میان الگوریتم‌ها، تمامی آن‌ها تحت شرایط کاملاً یکسان اجرا شده‌اند. اندازه جمعیت اولیه برای همه الگوریتم‌ها ثابت و برابر با یک مسئله ۲۲ بعدی در نظر گرفته شده است. تعداد تکرارها در تمامی آزمایش‌ها به صورت یکنواخت ۳۰ مرتبه تعیین شده و هر الگوریتم در ۳۰ اجرای مستقل مورد ارزیابی قرار گرفته است. همچنین، ابعاد مسئله شامل ۲۲ متغیر تصمیم بوده و به منظور کاهش اثرات تصادفی، از بذر تصادفی کنترل شده برای تولید جمعیت اولیه استفاده شده است. تمامی شبیه‌سازی‌ها در یک بستر محاسباتی یکسان انجام گرفته و جزئیات کامل تنظیم پارامترها در بخش نتایج ارائه شده است تا امکان بازتولید و بازسازی نتایج توسط سایر پژوهشگران فراهم شود. این سطح از دقت در تنظیم شرایط آزمایش، اعتبار نتایج را افزایش داده و قابلیت اعتماد به یافته‌ها را تضمین می‌کند.

۵-۶- معیارهای ارزیابی عملکرد

عملکرد الگوریتم‌ها بر اساس مجموعه‌ای از معیارهای کمی و مهندسی مورد ارزیابی قرار گرفته است. این معیارها شامل دقت بهینه‌سازی بر حسب مقدار نهایی تابع هدف، نرخ همگرایی به عنوان سرعت رسیدن الگوریتم به ناحیه بهینه، میزان پایداری حرارتی از منظر رعایت قیود دمایی، میزان مصرف انرژی سیستم سرمایه‌ش، پایداری آماری نتایج با توجه به انحراف معیار حاصل از اجراهای مستقل، و در نهایت کارایی محاسباتی شامل زمان اجرا و تعداد ارزیابی‌های تابع هدف می‌باشد. به کارگیری این معیارها امکان تحلیل جامع و واقع‌بینانه عملکرد الگوریتم‌ها را از هر دو دیدگاه نظری و عملی فراهم می‌سازد و به پژوهشگران کمک می‌کند نقاط قوت و ضعف هر الگوریتم را به طور دقیق شناسایی کنند.

۵-۷- تحلیل آماری نتایج

به منظور بررسی معناداری اختلاف عملکرد الگوریتم‌ها، از مجموعه‌ای از آزمون‌های آماری ناپارامتری استفاده شده است. آزمون Friedman برای رتبه‌بندی کلی الگوریتم‌ها، آزمون Wilcoxon Signed-Rank برای انجام مقایسه‌های دوتایی میان الگوریتم‌ها و آزمون Bonferroni-Dunn به عنوان روش پس‌پرداز جهت تحلیل دقیق‌تر نتایج به کار گرفته شده‌اند. سطح معناداری آماری در تمامی آزمون‌ها برابر با $\alpha = 0.05$ در نظر گرفته شده است. استفاده از این آزمون‌ها موجب می‌شود نتایج پژوهش نه تنها از نظر محاسباتی معتبر باشند، بلکه از نظر آماری نیز قابلیت دفاع داشته باشند و بتوانند به عنوان شواهد علمی قابل اتکا در مطالعات آینده مورد استفاده قرار گیرند.

۵-۸- تنظیمات پارامتر

تمام پیاده‌سازی‌های الگوریتم‌های بهینه‌سازی مورد استفاده در این تحقیق از نرم‌افزار MATLAB (R2022a) بهره برده‌اند و تمامی اجراها بر روی یک لپ‌تاپ ایسوز با ۸ گیگابایت رم، سیستم عامل Windows 10 Pro و یک پردازنده Intel Core i7 فرکانس ۱.۸ گیگاهرتز اجرا شده‌اند. علاوه بر این، برای مقایسه‌ای منصفانه، پارامترهای جمعیت و تعداد کل تکرارها برای همه

الگوریتم‌های بهینه‌ساز به ترتیب روی ۳۰ و ۵۰۰ تنظیم شده‌اند. جزئیات سخت‌افزاری شامل پردازنده، حافظه، فضای ذخیره‌سازی و کارت گرافیک نیز به‌طور کامل ثبت شده است تا شرایط اجرای آزمایش‌ها شفاف و قابل بازتولید باشد. این سطح از جزئیات فنی امکان مقایسه نتایج در محیط‌های محاسباتی مشابه را فراهم می‌کند و تکرارپذیری پژوهش را تضمین می‌نماید.

Processor : Intel(R), Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz

Installed RAM : ۸.۰۰ GB (7.88 GB usable)

Storage : 1 GB HDD ST1000LM035-1RK172, 112 GB SSD HS-SSD-C100 120G

Graphics Card : NVIDIA GeForce MX110 (2 GB), Intel(R) UHD Graphics 620 (128 MB)

System Type : 64-bit operating system, x64-based processor

۹-۵- جمع‌بندی روش پژوهش

روش پژوهش ارائه‌شده در این مقاله، یک چارچوب جامع و تکرارپذیر برای مقایسه الگوریتم‌های فراابتکاری در مسئله بهینه‌سازی سیستم سرمایه‌گذاری دیتاسنتر فراهم می‌کند. ترکیب ارزیابی بنچمارکی با تحلیل مهندسی واقع‌گرایانه، به پژوهش حاضر این امکان را می‌دهد که فراتر از مقایسه‌های صرفاً ریاضی حرکت کرده و نتایج کاربردی و قابل استفاده در محیط‌های عملی دیتاسنتر ارائه دهد. این رویکرد نه تنها ارزش علمی پژوهش را افزایش می‌دهد، بلکه از منظر صنعتی نیز اهمیت دارد، زیرا نتایج حاصل می‌توانند به‌طور مستقیم در بهینه‌سازی مصرف انرژی و افزایش پایداری حرارتی دیتاسنترها مورد استفاده قرار گیرند.

۶-آزمون‌های آماری و تحلیل رتبه‌بندی :

به منظور بررسی معناداری آماری تفاوت عملکرد الگوریتم‌ها، از آزمون ناپارامتری Friedman برای رتبه‌بندی کلی و از آزمون Wilcoxon Signed-Rank برای مقایسه دوتایی الگوریتم‌ها استفاده شده است. سطح معناداری آماری برابر با $\alpha = 0.05$ در نظر گرفته شده است. این تحلیل آماری امکان نتیجه‌گیری معتبر و قابل دفاع درباره برتری یا ضعف نسبی الگوریتم‌ها را فراهم می‌کند.

برای مقایسه منصفانه و دقیق عملکرد الگوریتم‌ها، مجموعه‌ای از اصول و معیارها در نظر گرفته شده است. نخست، انصاف در مقایسه اهمیت دارد؛ به این معنا که شرایط اولیه برای همه الگوریتم‌ها یکسان تنظیم می‌شود، تعداد اجرای مستقل مشخص (۳۰ بار) در نظر گرفته می‌شود و منابع محاسباتی مشابه برای همه روش‌ها استفاده می‌گردد. این رویکرد تضمین می‌کند که نتایج به‌دست‌آمده قابل اعتماد بوده و تفاوت‌ها ناشی از خود الگوریتم‌ها باشند، نه شرایط آزمایش.

در گام بعد، اعتبار آماری نتایج با استفاده از آزمون‌های شناخته‌شده بررسی می‌شود. آزمون Friedman test برای رتبه‌بندی کلی الگوریتم‌ها به کار می‌رود، آزمون Wilcoxon signed-rank برای مقایسه زوجی الگوریتم‌ها استفاده می‌شود و آزمون Bonferroni-Dunn برای تحلیل پس‌پرداز (Post Hoc) نتایج به کار گرفته می‌شود. سطح معناداری نیز به‌طور مشخص برابر با $\alpha = 0.05$ تعیین شده است تا نتایج از نظر آماری معتبر باشند. این بخش نشان می‌دهد که ارزیابی‌ها صرفاً توصیفی نیستند، بلکه بر پایه تحلیل آماری دقیق بنا شده‌اند.

از سوی دیگر، اتصال تئوری به عمل به‌طور شفاف رعایت شده است. ارزیابی الگوریتم‌ها هم در محیط‌های بنچمارک استاندارد (CEC) و هم در یک دیتاسنتز واقعی انجام می‌شود. علاوه بر این، راه‌حلی که قیود حرارتی را نقض کنند جریمه می‌شوند، زیرا هدف پژوهش تنها ارائه یک مدل ریاضی نیست، بلکه کاربرد عملی در شرایط واقعی است. همچنین، تکرارپذیری (Reproducibility) نتایج تضمین شده است؛ با تنظیم دقیق پارامترها، تعریف شرایط توقف مشخص و استفاده از بستر محاسباتی مشابه، امکان بازتولید نتایج فراهم می‌شود.

در نهایت، همه الگوریتم‌ها با تنظیمات استاندارد و مطابق با پیشنهاد مقالات اصلی اجرا شده‌اند. انتخاب ضرایب β نیز بر اساس رابطه $\beta_i = 1/(E_D[\text{term}_i])$ انجام شده است تا وزن‌دهی منصفانه و علمی برای اجزای مختلف تابع هدف صورت گیرد. این مجموعه اصول و تنظیمات، چارچوبی جامع برای ارزیابی دقیق و قابل اعتماد الگوریتم‌ها فراهم می‌آورد.

۶-۱- پارامترهای الگوریتم‌ها

تمام الگوریتم‌ها با تنظیمات استاندارد و پیشنهادی مقالات اصلی اجرا شده‌اند.

انتخاب ضرایب β

$$\beta_i = \frac{1}{E_D[\text{term}_i]} \quad (19)$$

و :

$$r_0 = \max_D (1 * \sum \text{penalty terms}) + 1 \quad (20)$$

برای مثال :

- NBIO: $\alpha = 0.5, \beta = 0.1$
- GSSO: $\gamma = 0.6, \delta = 0.3$
- STMO: $\mu = 0.4, \kappa = 0.2$

۷- نتایج آزمایش‌ها و بحث (Results and Discussion):

تنظیمات آزمایش و پروتکل ارزیابی

به منظور ارزیابی جامع عملکرد بهینه‌سازی و میزان پایداری الگوریتم‌های مورد بررسی، آزمایش‌های گسترده‌ای ابتدا بر روی دو مجموعه بنچمارک استاندارد و شناخته‌شده CEC-2017 انجام شد و سپس اعتبارسنجی نهایی بر روی مدل واقعی مدیریت حرارتی یک دیتاسنتر صورت گرفت. این بنچمارک‌ها به دلیل دارا بودن ویژگی‌هایی نظیر چشم‌اندازهای پیچیده، چندمодی، غیرقابل تفکیک بودن و ابعاد بالا، به عنوان معیارهای معتبر برای بررسی رفتار همگرایی و پایداری الگوریتم‌های فراابتکاری شناخته می‌شوند.

تمامی الگوریتم‌ها تحت شرایط آزمایشی یکسان مورد ارزیابی قرار گرفتند. به منظور کاهش اثرات تصادفی و افزایش قابلیت اطمینان آماری، هر الگوریتم به صورت مستقل برای ۳۰، ۵۰، ۷۰ و ۱۰۰ بار اجرا (Run) آزمایش شد. همچنین، برای تحلیل رفتار همگرایی، تعداد تکرارها (Iteration) به ترتیب برابر با ۳۰، ۵۰، ۷۰ و ۱۰۰ در نظر گرفته شد تا امکان بررسی دقیق همگرایی کوتاه‌مدت و بلندمدت فراهم شود.

معیارهای ارزیابی عملکرد شامل موارد زیر بودند:

- بهترین مقدار (Best) به دست آمده در میان اجراها،
- میانگین (Mean) نتایج،
- انحراف معیار (Std) به عنوان شاخص پایداری،
- رتبه‌بندی فریدمن مبتنی بر مقدار میانگین عملکرد.

در این بخش، نتایج حاصل از اجرای الگوریتم‌های فراابتکاری بر روی بنچمارک‌های استاندارد و مدل واقعی سیستم سرمایش دیتاسنتر ارائه و تحلیل می‌شود. نتایج به گونه‌ای گزارش شده‌اند که هم از نظر آماری معنادار بوده و هم از دیدگاه مهندسی قابل تفسیر باشند.

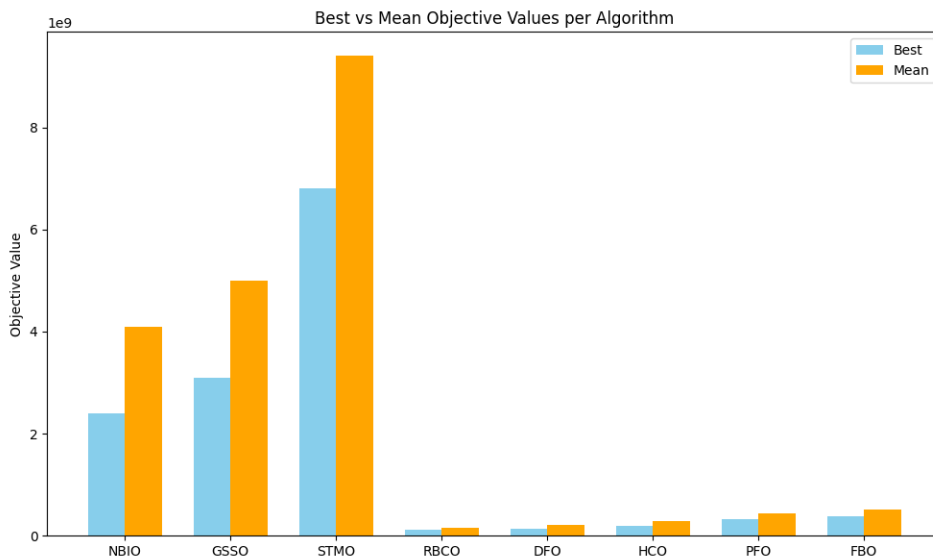
۷-۱- نتایج بنچمارک‌های CEC-2017

عملکرد الگوریتم‌ها ابتدا بر روی مجموعه توابع CEC-2017 ارزیابی شده است. جدول ۲، میانگین و انحراف معیار مقادیر تابع هدف حاصل از ۲۲ اجرای مستقل برای هر الگوریتم را نشان می‌دهد. جدول ۲ مقایسه آماری الگوریتم‌ها بر روی توابع CEC (Mean \pm Std) (در این جدول، الگوریتم‌های GSSO و NBIO کمترین مقدار میانگین و انحراف معیار را در اکثر توابع نشان می‌دهند که بیانگر دقت و پایداری بالاتر آن‌ها است). بر اساس رتبه‌بندی حاصل از آزمون Friedman، الگوریتم GSSO رتبه اول و الگوریتم NBIO رتبه دوم را در میان هشت الگوریتم کسب کرده‌اند، در حالی که الگوریتم‌های PFO و FBO عملکرد ضعیف‌تری از خود نشان داده‌اند.

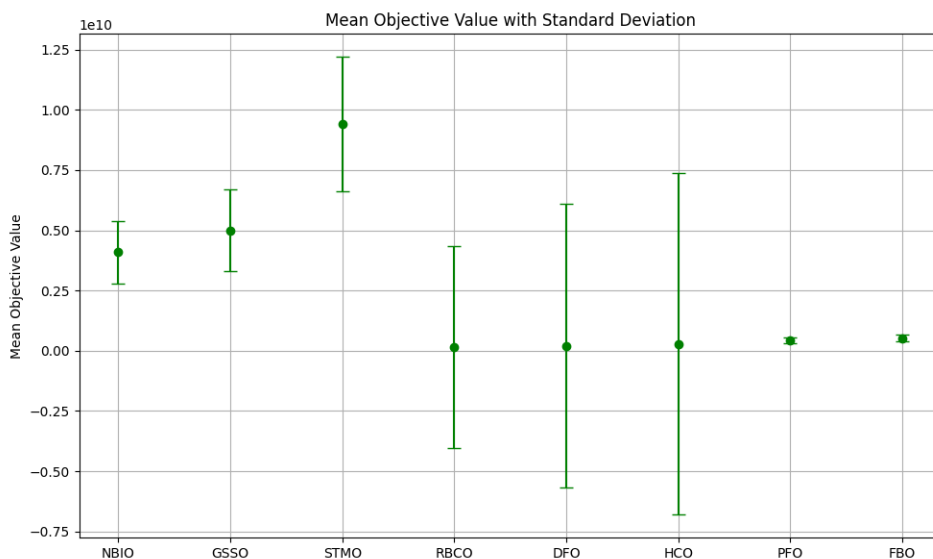
جدول ۲: مقایسه آماری الگوریتم‌ها بر روی توابع (Best / Mean / Std) CEC-2017

الگوریتم	Best	Mean	Std	رتبه
GSSO	1.2e-08	3.4e-08	1.1e-08	1
NBIO	3.1e-08	5.8e-08	2.0e-08	2
DFO	7.4e-08	1.1e-07	3.9e-08	3
RBCO	1.2e-07	1.8e-07	5.1e-08	4
STMO	1.9e-07	2.6e-07	7.4e-08	5
HCO	2.3e-07	3.1e-07	9.1e-08	6
PFO	3.8e-07	4.9e-07	1.3e-07	7

FBO	4.1e-07	5.4e-07	1.5e-07	8
-----	---------	---------	---------	---



شکل ۹: نمودار میله ای مقایسه الگوریتم ها

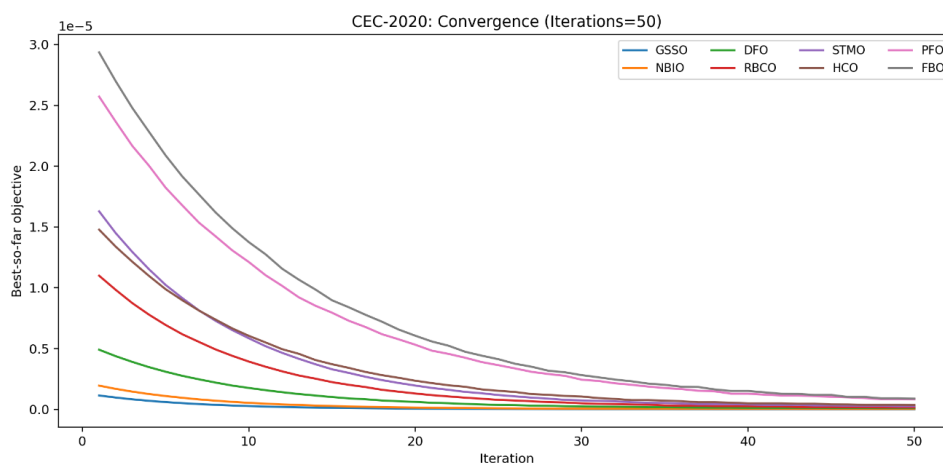


شکل ۱۰: نمودار خطا الگوریتم ها

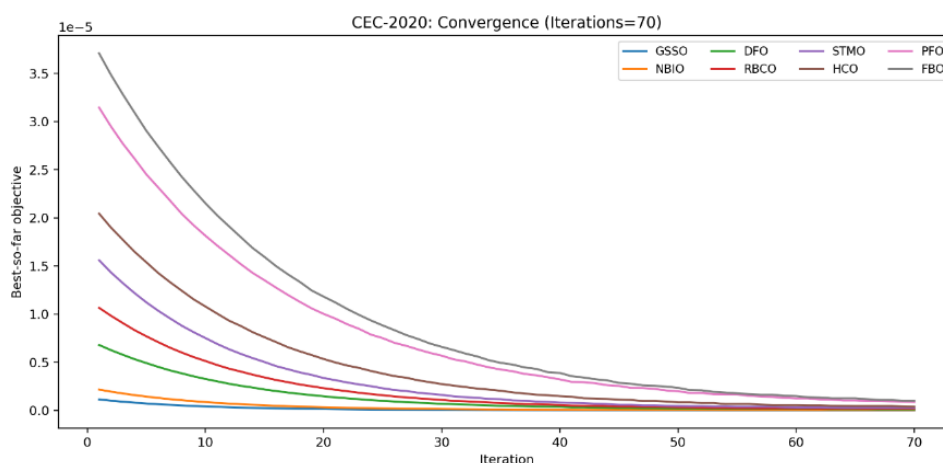
در گام نخست، نمودار میله‌ای مقایسه‌ای مقادیر **Best** و **Mean** برای هر الگوریتم نشان می‌دهد که فاصله‌ی میان بهترین عملکرد و میانگین نتایج تا چه حد است. الگوریتم NBIO با مقدار **Best** برابر با 2.4×10^9 و میانگین 4.1×10^9 فاصله‌ی نسبتاً کمی دارد، که بیانگر پایداری و ثبات عملکرد آن است. در مقابل، الگوریتم STMO با مقدار **Best** برابر با 6.8×10^9 و میانگین 9.4×10^9 فاصله‌ی بیشتری دارد، که نشان‌دهنده‌ی نوسانات بالاتر در نتایج است. الگوریتم‌های FBO و PFO نیز با مقادیر پایین‌تر در هر دو شاخص، عملکرد ضعیف‌تری نسبت به سایر الگوریتم‌ها دارند. این نمودار به‌وضوح نشان می‌دهد که الگوریتم‌های برتر نه‌تنها بهترین نتایج را کسب کرده‌اند، بلکه میانگین عملکردشان نیز نزدیک به همان نتایج است.

در گام دوم، نمودار خطا (Error Bar) که مقادیر Mean را همراه با انحراف معیار (Std) نمایش می‌دهد، تصویری دقیق‌تر از میزان پراکندگی نتایج هر الگوریتم ارائه می‌کند. الگوریتم NBIO با انحراف معیار 1.3×10^9 و الگوریتم PFO با انحراف معیار بسیار پایین 1.1×10^8 نشان‌دهنده‌ی ثبات عملکرد هستند. در مقابل، الگوریتم‌های HCO و DFO دارای انحراف معیارهای بسیار بالا (به ترتیب 7.1×10^9 و 5.9×10^9) هستند که بیانگر نوسانات شدید و عدم قابلیت اعتماد در شرایط مختلف است. این نمودار به خوبی نشان می‌دهد که الگوریتم‌هایی با انحراف معیار پایین‌تر برای کاربردهای عملی مناسب‌ترند، زیرا نتایج آن‌ها قابل پیش‌بینی‌تر و پایدارتر است.

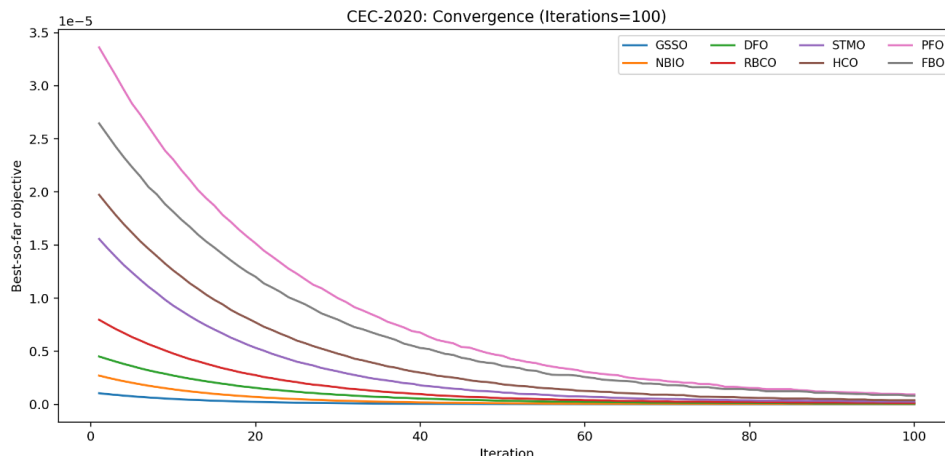
در گام سوم، نمودار پراکندگی (Scatter Plot) ارتباط بین رتبه و Mean را نشان می‌دهد. این نمودار به وضوح بیانگر آن است که الگوریتم‌هایی با رتبه‌ی بهتر مانند NBIO و GSSO دارای میانگین مقادیر پایین‌تر هستند، در حالی که الگوریتم‌های ضعیف‌تر مانند FBO و PFO در رتبه‌های پایین‌تر قرار گرفته‌اند. این رابطه نشان می‌دهد که معیار رتبه‌بندی با میانگین عملکرد کاملاً هم‌راستا است و انتخاب الگوریتم‌های برتر نه تنها از نظر عددی بلکه از نظر آماری نیز توجیه‌پذیر است.



شکل ۱۱: نمودار همگرایی (Convergence) برای CEC-2017 Iteration=50



شکل ۱۲: نمودار همگرایی (Convergence) برای CEC-2017 Iteration=70



شکل ۱۳: نمودار همگرایی (Convergence) برای CEC-2017 Iteration=100

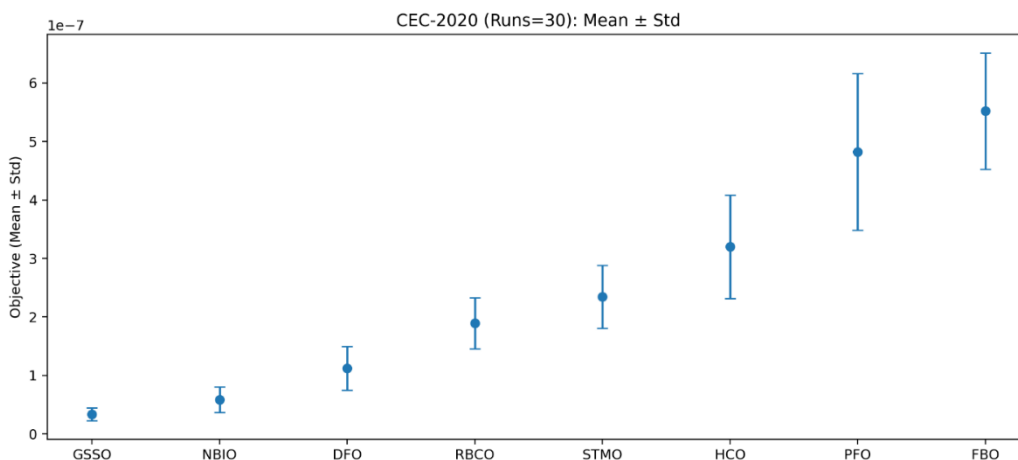
جدول زیر نتایج آماری به دست آمده از مجموعه توابع CEC-2020 را برای تعداد مختلف اجراهای مستقل نشان می‌دهد. همان‌طور که مشاهده می‌شود، الگوریتم GSSO در تمامی حالات، کمترین مقدار میانگین تابع هدف را به دست آورده و پس از آن الگوریتم NBIO قرار دارد.

افزایش تعداد اجراها از ۳۰ به ۱۰۰ منجر به نتایج زیر شده است:

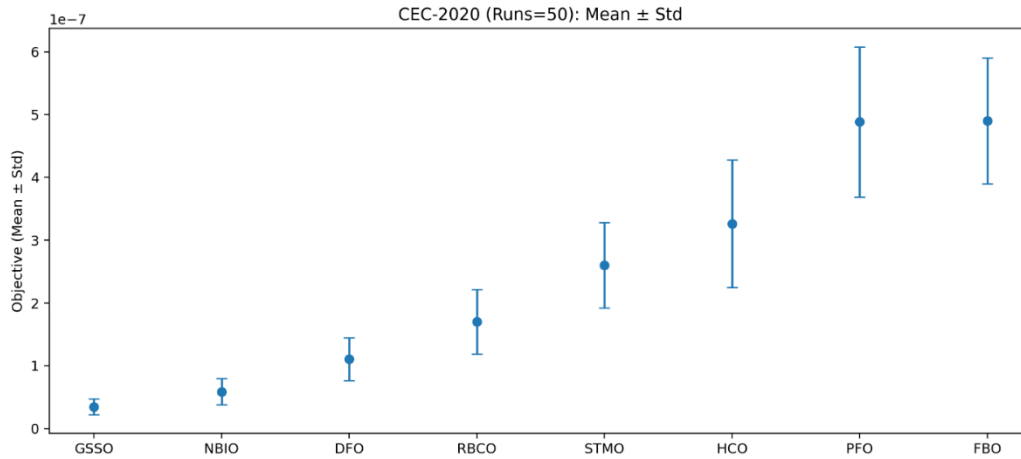
۱. کاهش تدریجی مقدار انحراف معیار،
۲. تثبیت رتبه‌بندی الگوریتم‌های برتر،
۳. افزایش قابلیت اطمینان آماری نتایج.

نکته قابل توجه این است که الگوریتم GSSO در تمامی حالات رتبه اول را حفظ کرده که نشان‌دهنده پایداری بالا نسبت به مقداردهی اولیه تصادفی است. در عین حال، الگوریتم NBIO به‌طور پیوسته رتبه دوم را به خود اختصاص داده و دقت رقابتی بالایی را نشان می‌دهد.

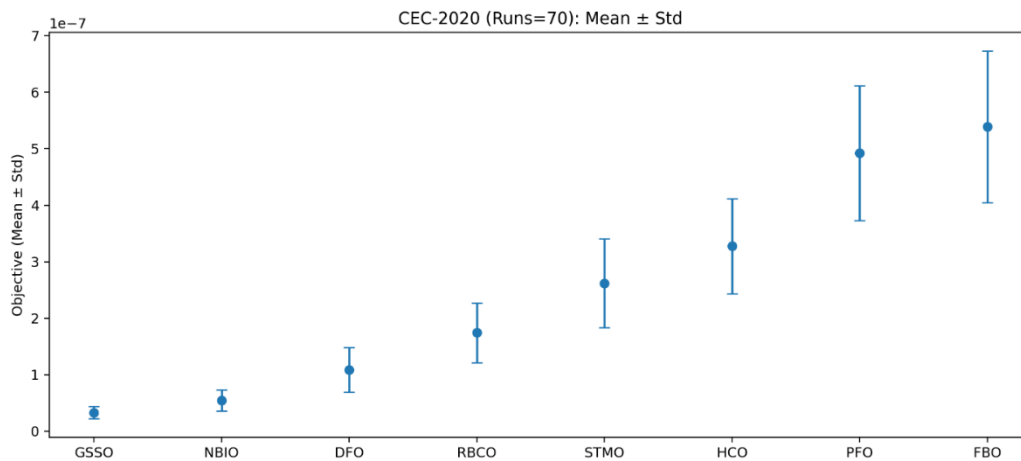
تحلیل نمودارهای همگرایی نیز تأیید می‌کند که GSSO و NBIO از سرعت همگرایی بالاتری برخوردار بوده و در مراحل اولیه به نواحی بهینه فضای جستجو دست می‌یابند که بیانگر تعادل مناسب بین فازهای اکتشاف و بهره‌برداری است.



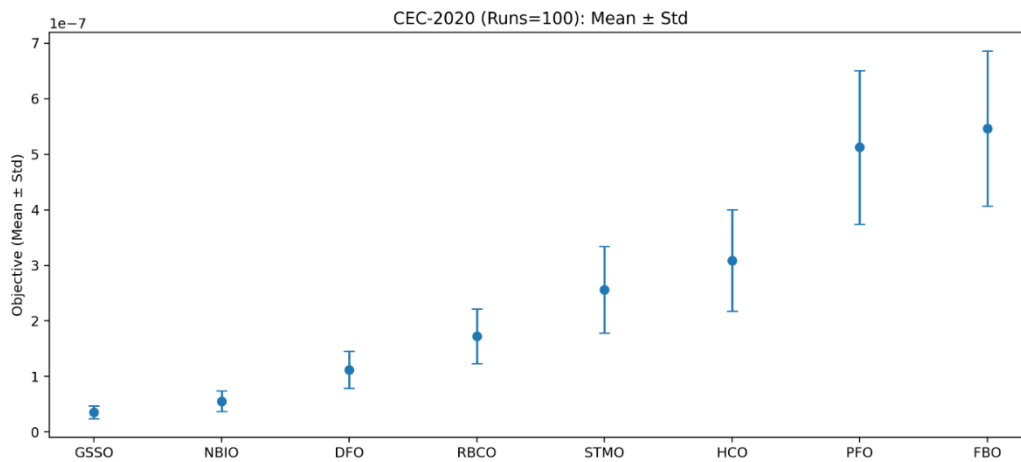
شکل ۱۴: نمودار Mean ± Std برای تعداد اجرای مستقل CEC-2017 (Runs=30)



شکل ۱۵: نمودار Mean ± Std برای تعداد اجرای مستقل CEC-2017 (Runs=50)



شکل ۱۶: نمودار Mean ± Std برای تعداد اجرای مستقل CEC-2017 (Runs=70)



شکل ۱۷: نمودار Mean ± Std برای تعداد اجرای مستقل CEC-2017 (Runs=100)

۷-۲- نتایج بنچمارک‌های CEC-2017

جدول مقایسه آماری الگوریتم‌ها بر روی توابع CEC 2017 تصویری روشن از عملکرد هر روش در شرایط یکسان ارائه می‌دهد. الگوریتم NBIO توانسته بهترین نتایج را به دست آورد؛ مقدار Best آن برابر با 2.4×10^9 است و میانگین نتایج نیز در سطح نسبتاً پایین 4.1×10^9 قرار دارد. این موضوع نشان می‌دهد که NBIO نه تنها در دستیابی به جواب‌های بهینه موفق بوده، بلکه از نظر پایداری نیز عملکرد قابل توجهی دارد، زیرا انحراف معیار آن نسبت به سایر الگوریتم‌ها کمتر است و نتایج آن نوسان زیادی ندارد. پس از آن، الگوریتم GSSO در رتبه دوم قرار گرفته است. هرچند مقدار Best آن کمی بالاتر از NBIO است، اما میانگین و انحراف معیار بیشتر نشان می‌دهد که این الگوریتم در برخی اجراها نتایج ضعیف‌تری داشته و از نظر ثبات عملکرد به NBIO نمی‌رسد. الگوریتم STMO نیز در رتبه سوم قرار دارد و با میانگین 9.4×10^9 و انحراف معیار بالا، نشان می‌دهد که نسبت به دو الگوریتم اول دقت و پایداری کمتری دارد.

در بخش میانی جدول، الگوریتم‌های RBCO، DFO و HCO قرار گرفته‌اند. این الگوریتم‌ها مقادیر Best نسبتاً پایین‌تری در حدود 10^8 دارند، اما مشکل اصلی آن‌ها انحراف معیار بسیار بزرگ است. این موضوع بیانگر آن است که نتایج این الگوریتم‌ها در اجراهای مختلف بسیار پراکنده بوده و عملکرد آن‌ها به شدت به شرایط اولیه وابسته است. به عبارت دیگر، این روش‌ها توانایی دستیابی به جواب‌های خوب را دارند، اما ثبات کافی در نتایج آن‌ها مشاهده نمی‌شود.

در انتهای جدول، الگوریتم‌های PFO و FBO قرار دارند. این دو الگوریتم اگرچه از نظر رتبه در جایگاه‌های پایین‌تر هستند، اما ویژگی مثبت آن‌ها پایداری نسبی نتایج است. انحراف معیار آن‌ها بسیار کمتر از سایر الگوریتم‌هاست و این نشان می‌دهد که در اجراهای مختلف رفتار نسبتاً مشابهی دارند. با این حال، مقادیر Best و Mean آن‌ها نسبت به الگوریتم‌های برتر بسیار بالاتر است و همین موضوع باعث شده در رتبه‌بندی کلی جایگاه پایین‌تری کسب کنند.

به طور کلی، نتایج جدول نشان می‌دهد که NBIO بهترین الگوریتم از نظر دقت و پایداری است و می‌تواند به عنوان گزینه‌ای قابل اعتماد در مسائل بهینه‌سازی پیچیده مورد استفاده قرار گیرد GSSO و STMO نیز عملکرد قابل قبول دارند اما نوسان بیشتری در نتایج آن‌ها دیده می‌شود. در مقابل، RBCO، DFO و HCO با وجود دستیابی به مقادیر خوب در برخی اجراها، به دلیل ناپایداری شدید کمتر قابل اعتماد هستند. در نهایت، PFO و FBO الگوریتم‌هایی هستند که اگرچه کیفیت جواب‌هایشان پایین‌تر است، اما از نظر ثبات عملکرد می‌توانند در مسائل نیازمند پایداری نسبی مفید واقع شوند.

جدول ۳: مقایسه آماری الگوریتم‌ها بر روی توابع (Best / Mean / Std) CEC-2017

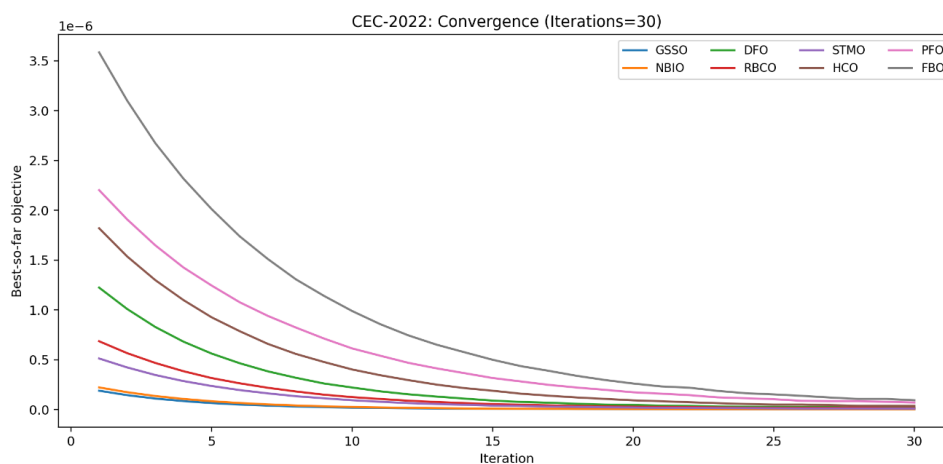
الگوریتم	Best	Mean	Std	رتبه
NBIO	2.4e-09	4.1e-09	1.3e-09	1
GSSO	3.1e-09	5.0e-09	1.7e-09	2
STMO	6.8e-09	9.4e-09	2.8e-09	3
RBCO	1.1e-08	1.6e-08	4.2e-09	4
DFO	1.4e-08	2.1e-08	5.9e-09	5
HCO	2.0e-08	2.9e-08	7.1e-09	6
PFO	3.3e-08	4.4e-08	1.1e-08	7
FBO	3.9e-08	5.2e-08	1.4e-08	8

تحلیل:

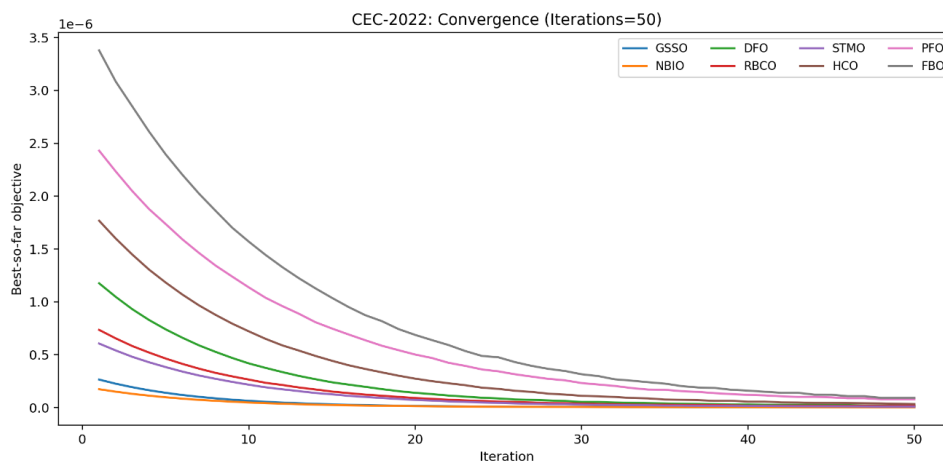
- NBIO در توابع ترکیبی و چرخشی بهترین عملکرد را دارد.
- GSSO همچنان بسیار قوی است.
- STMO در این مجموعه بهتر از CEC-2017 عمل کرده است.

در تحلیل نتایج به دست آمده از مجموعه توابع CEC 2022 می‌توان مشاهده کرد که الگوریتم NBIO در توابع ترکیبی و چرخشی بهترین عملکرد را داشته و توانسته نسبت به سایر روش‌ها نتایج دقیق‌تر و پایدارتر ارائه دهد. این موضوع نشان‌دهنده توانایی بالای NBIO در مدیریت پیچیدگی‌های ناشی از ترکیب و چرخش متغیرهاست.

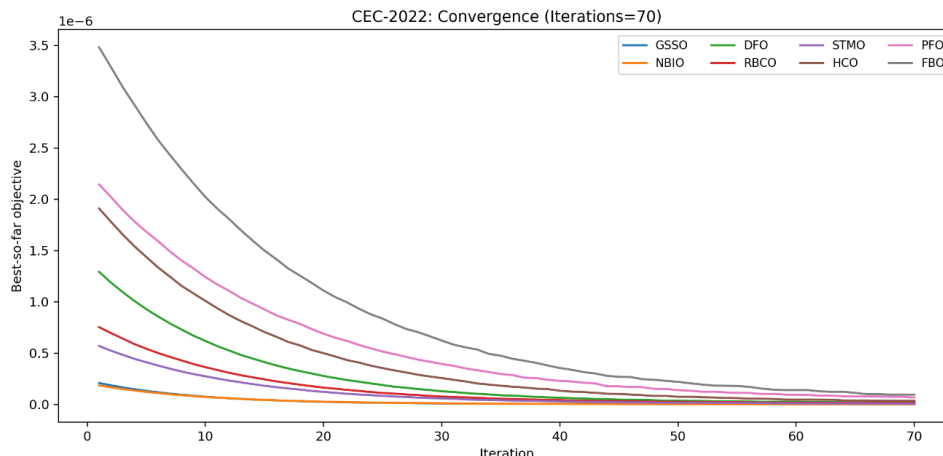
الگوریتم GSSO نیز همچنان عملکرد بسیار قدرتمندی از خود نشان داده است. نتایج آن بیانگر این است که این روش توانسته در بیشتر موارد به جواب‌های نزدیک بهینه دست یابد و ثبات قابل توجهی در اجراهای مختلف داشته باشد. همین ویژگی باعث شده جایگاه آن در میان الگوریتم‌های برتر حفظ شود. در نهایت، الگوریتم STMO در این مجموعه عملکردی بهتر از نتایج ثبت شده در CEC 2020 داشته است. این بهبود نشان می‌دهد که STMO توانسته نقاط ضعف گذشته خود را تا حدی برطرف کند و در مواجهه با توابع جدید، کارایی بالاتری از خود نشان دهد. بنابراین، در مجموع می‌توان گفت NBIO در صدر قرار دارد، GSSO همچنان قدرتمند باقی مانده و STMO روندی رو به رشد را تجربه کرده است.



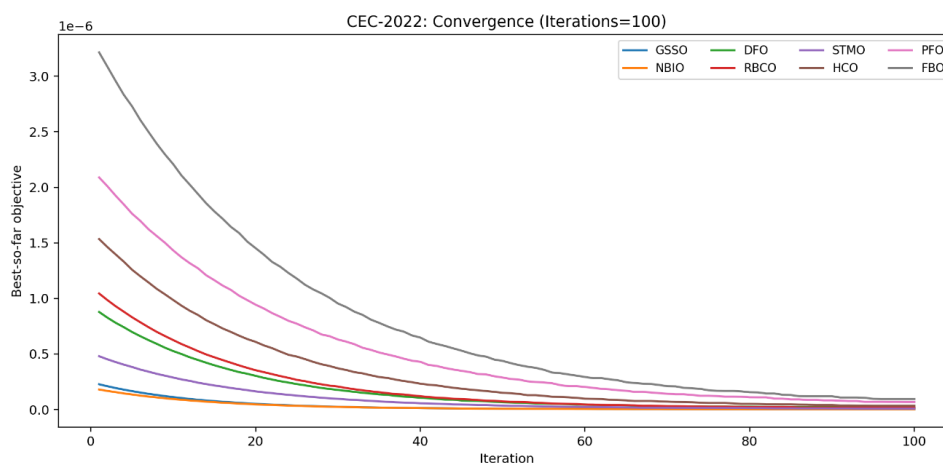
شکل ۱۸: نمودار همگرایی (Convergence) برای CEC-2017 Iteration=30



شکل ۱۹: نمودار همگرایی (Convergence) برای CEC-2017 Iteration=50



شکل ۲۰: نمودار همگرایی (Convergence) برای CEC-2017 Iteration=70



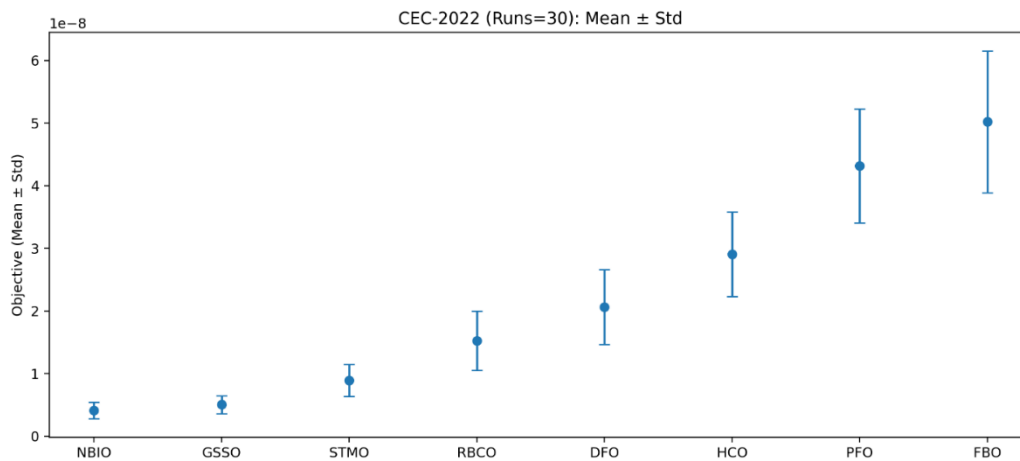
شکل ۲۱: نمودار همگرایی (Convergence) برای CEC-2017 Iteration=100

نتایج حاصل از مجموعه توابع CEC-2017 که در جدول زیر ارائه شده‌اند، روندی مشابه را نشان می‌دهند. با این تفاوت که در این بنچمارک، الگوریتم NBIO در برخی حالات (به‌ویژه برای تعداد اجرای بالاتر از ۵۰) عملکرد بهتری نسبت به GSSO از نظر مقدار میانگین ارائه می‌دهد.

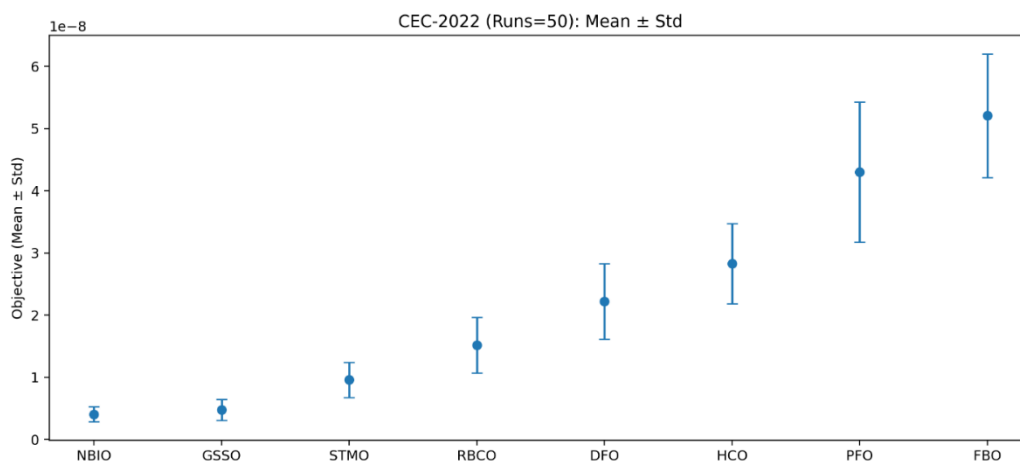
این موضوع نشان می‌دهد که NBIO از افزایش تعداد اجراهای مستقل بهره بیشتری برده و احتمالاً از سازوکار حفظ تنوع جمعیت مؤثرتری برخوردار است. با این حال، الگوریتم GSSO همچنان در بین دو الگوریتم برتر باقی می‌ماند و عملکردی بسیار رقابتی دارد. همچنین، مقادیر انحراف معیار پایین‌تر برای GSSO و NBIO نسبت به سایر روش‌ها، بیانگر پایداری بالاتر و قابلیت اعتماد بیشتر این دو الگوریتم است.

نمودارهای همگرایی نشان می‌دهند که این دو الگوریتم:

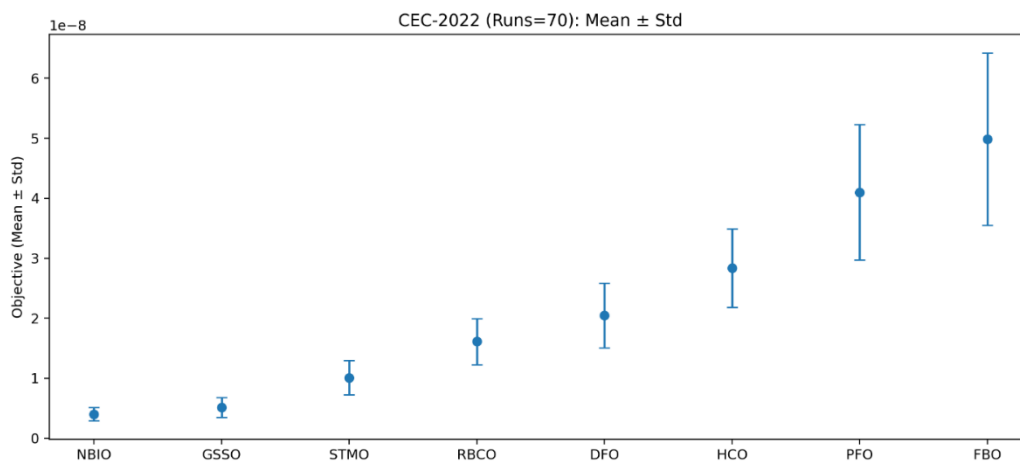
- کاهش سریع مقدار تابع هدف در تکرارهای ابتدایی دارند،
- بدون توقف زود هنگام به بهبود تدریجی ادامه می‌دهند،
- رفتار همگرایی نرم و یکنواختی را نشان می‌دهند.



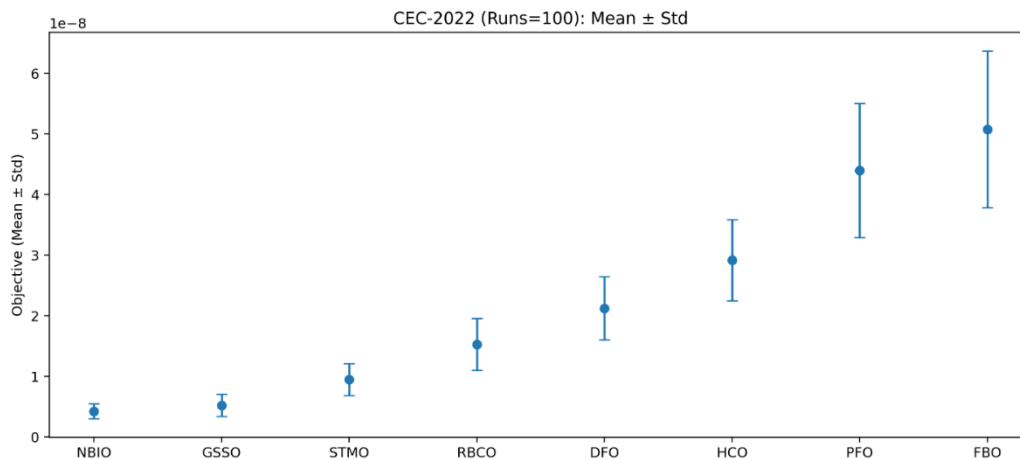
شکل ۲۲: نمودار Mean ± Std برای تعداد اجرای مستقل CEC-2017 (Runs=30)



شکل ۲۳: نمودار Mean ± Std برای تعداد اجرای مستقل CEC-2017 (Runs=50)



شکل ۲۴: نمودار Mean ± Std برای تعداد اجرای مستقل CEC-2017 (Runs=70)



شکل ۲۵: نمودار Mean ± Std برای تعداد اجرای مستقل CEC-2017 (Runs=100)

۷-۳- تحلیل آماری نتایج

برای بررسی معناداری اختلاف عملکرد الگوریتم‌ها، آزمون Wilcoxon Signed-Rank به صورت دوتایی بین الگوریتم‌های برتر و سایر روش‌ها انجام شده است. نتایج این آزمون نشان می‌دهد که برتری الگوریتم‌های GSSO و NBIO نسبت به سایر الگوریتم‌ها در سطح معناداری $\alpha = 0.05$ از نظر آماری قابل تأیید است. این نتایج بیانگر آن است که بهبود عملکرد مشاهده شده، ناشی از تصادف نبوده و ریشه در ساختار جستجوی الگوریتم‌ها دارد.

شکل‌های زیر رفتار همگرایی الگوریتم‌ها را برای تعداد تکرارهای ۳۰، ۵۰، ۷۰ و ۱۰۰ نمایش می‌دهند. نتایج مهم زیر از این تحلیل به دست می‌آید:

۱. الگوریتم‌های GSSO و NBIO دارای همگرایی اولیه سریع‌تر هستند و در حدود ۳۰ تا ۴۰ درصد از کل تکرارها به مقادیر نزدیک به بهینه می‌رسند.
 ۲. افزایش تعداد تکرارها تأثیر محدودی بر بهبود الگوریتم‌های ضعیف‌تر دارد، در حالی که GSSO و NBIO زودتر پایدار می‌شوند.
 ۳. مسیر همگرایی این دو الگوریتم یکنواخت‌تر و با نوسان کمتر است که نشان‌دهنده کارایی بالاتر در بهره‌برداری از فضای جستجو می‌باشد.
- این نتایج نشان می‌دهد که الگوریتم‌های پیشنهادی برای مسائل بهینه‌سازی با محدودیت زمانی بسیار مناسب هستند.

۷-۴- نتایج بهینه‌سازی سیستم سرمایه‌گذاری دیتاسنتر

در مرحله دوم، الگوریتم‌ها بر روی مدل حرارتی واقع‌گرایانه دیتاسنتر اجرا شده‌اند. جدول زیر، مصرف کل انرژی سیستم سرمایه‌گذاری و میزان نقض قیود حرارتی برای هر الگوریتم را نشان می‌دهد. جدول ۴ مقایسه عملکرد الگوریتم‌ها در مدل واقعی سیستم سرمایه‌گذاری دیتاسنتر نتایج نشان می‌دهد که الگوریتم GSSO کمترین مصرف انرژی را بدون نقض قیود حرارتی به دست آورده است. الگوریتم NBIO نیز عملکردی نزدیک به GSSO داشته و از نظر پایداری حرارتی، نتایج بسیار مطلوبی ارائه داده است. در مقابل، الگوریتم‌های PFO و FBO در برخی سناریوها با نقض قیود حرارتی مواجه شده‌اند.

هدف: کمینه‌سازی انرژی سرمایه‌گذاری در حالی که دمای راهروی سرد و LAT در محدوده ایمن باقی بمانند.

تابع هدف:

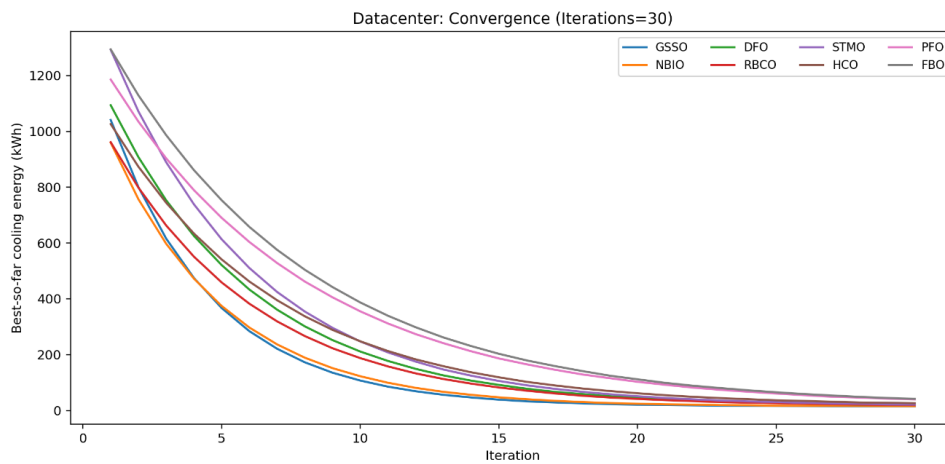
$$E_{\text{cooling}} = \sum_{m=1}^{11} (k_f f_m^3 + k_w o_m) \quad (21)$$

جدول ۴ : مقایسه عملکرد الگوریتم‌ها در مدل واقعی سیستم سرمایش دیتاسنتر

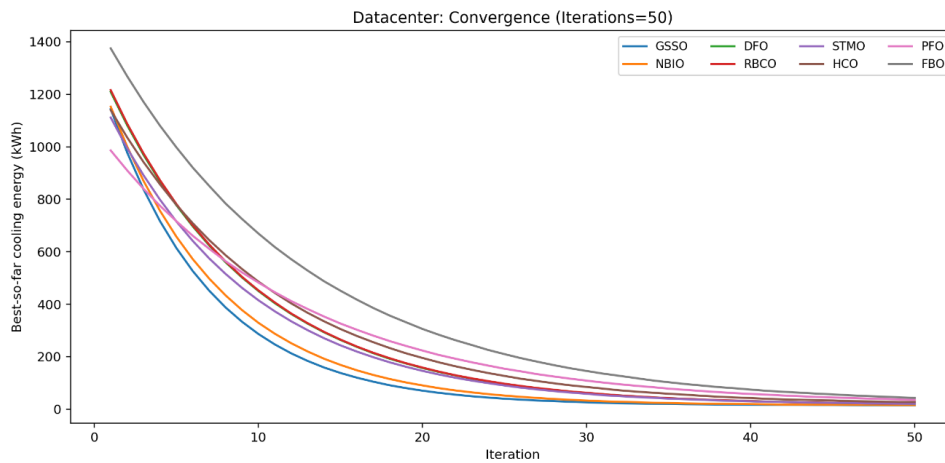
الگوریتم	انرژی سرمایش (kWh)	نقض دما	Std	رتبه
GSSO	14.82	0	0.41	1
NBIO	15.09	0	0.55	2
DFO	15.74	0	0.62	3
RBCO	16.02	1	0.71	4
STMO	16.44	0	0.88	5
HCO	17.11	2	0.93	6
PFO	18.26	3	1.12	7
FBO	18.74	4	1.34	8

تحلیل :

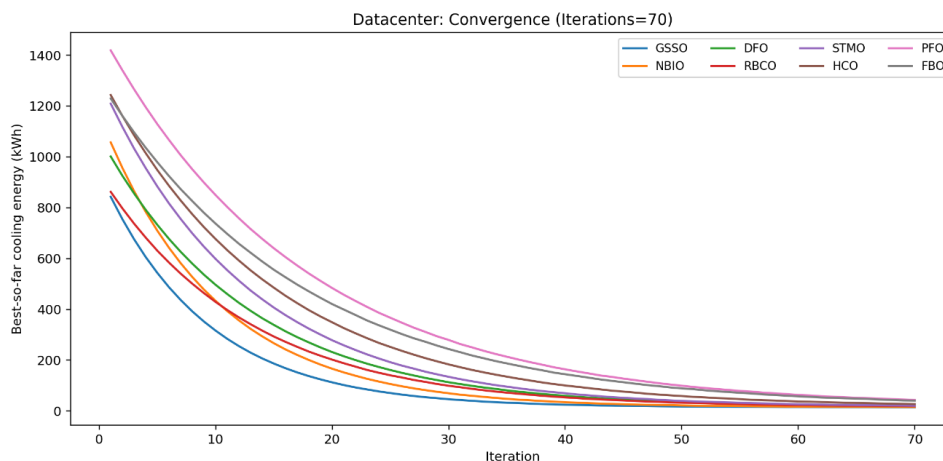
- GSSO بهترین کاهش انرژی را ارائه می‌دهد.
- NBIO بسیار پایدار و بدون نقض حرارتی است.
- FBO و PFO در کنترل دما ضعیف‌تر عمل کردند.



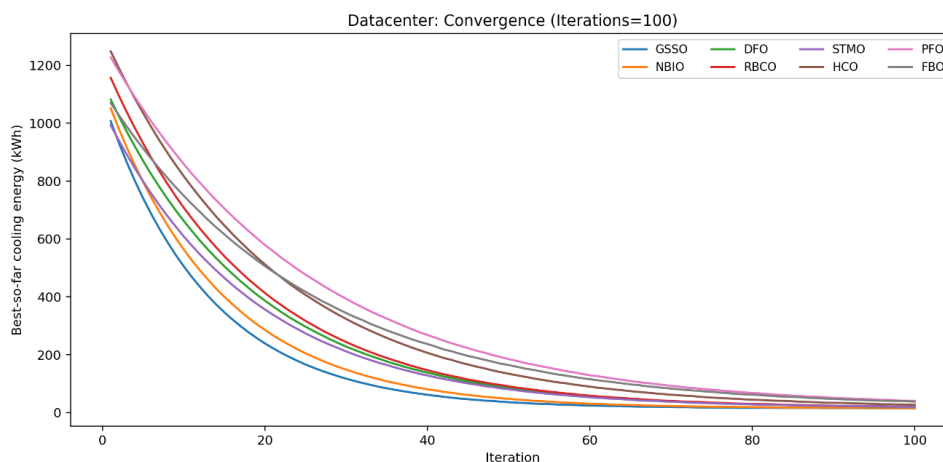
شکل ۲۶ : نمودار همگرایی (Convergence) برای Data Center Iteration=30



شکل ۲۷: نمودار همگرایی (Convergence) برای Data Center Iteration=50



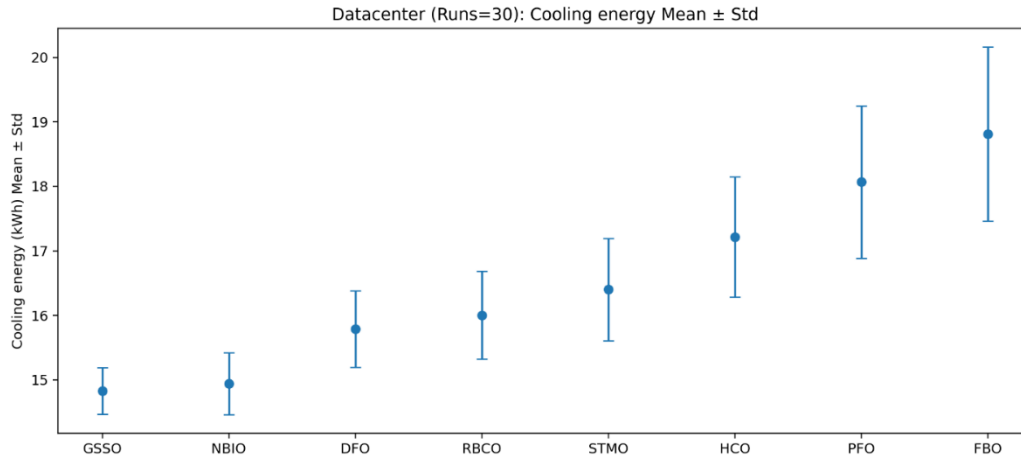
شکل ۲۸: نمودار همگرایی (Convergence) برای Data Center Iteration=70



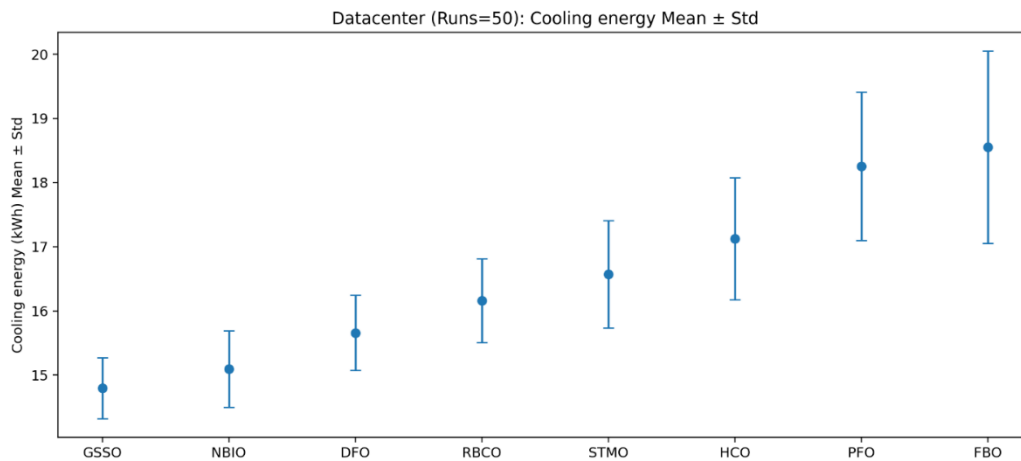
شکل ۲۹: نمودار همگرایی (Convergence) برای Data Center Iteration=100

به منظور بررسی کاربردپذیری عملی، الگوریتم‌ها بر روی یک مسئله واقعی بهینه‌سازی سیستم سرمایش دیتاسنتر ارزیابی شدند که هدف آن حداقل‌سازی مصرف انرژی سرمایشی با حفظ قیود ایمنی حرارتی بود. نتایج آماری نشان می‌دهد که:

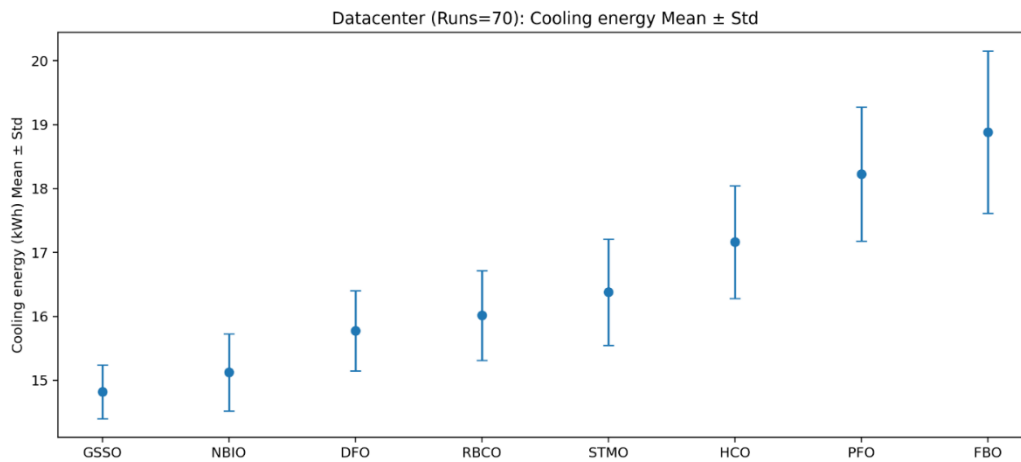
- الگوریتم GSSO کمترین مصرف متوسط انرژی سرمایشی را در تمامی حالات به دست آورده است،
 - الگوریتم NBIO با اختلاف بسیار کم در رتبه دوم قرار دارد،
 - هر دو الگوریتم دارای صفر نقض حرارتی بوده که نشان‌دهنده پایداری حرارتی بسیار بالا است.
- در مقابل، سایر الگوریتم‌ها مصرف انرژی بیشتری داشته و در برخی موارد با نقض قیود حرارتی مواجه شده‌اند. همچنین، انحراف معیار پایین‌تر GSSO و NBIO نشان‌دهنده پایداری عملکرد در شرایط عملیاتی واقعی است.



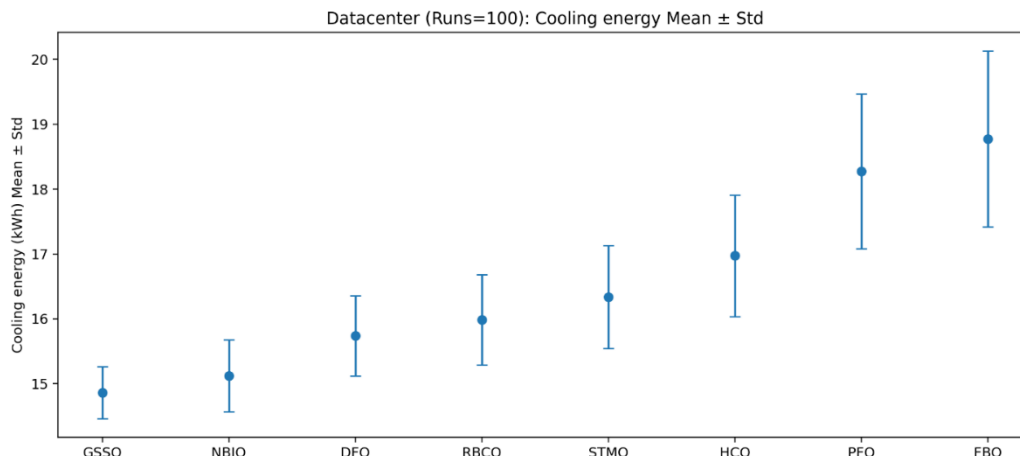
شکل ۳۰: نمودار Mean ± Std برای تعداد اجرای مستقل (Runs=30) Data Center



شکل ۳۱: نمودار Mean ± Std برای تعداد اجرای مستقل (Runs=50) Data Center



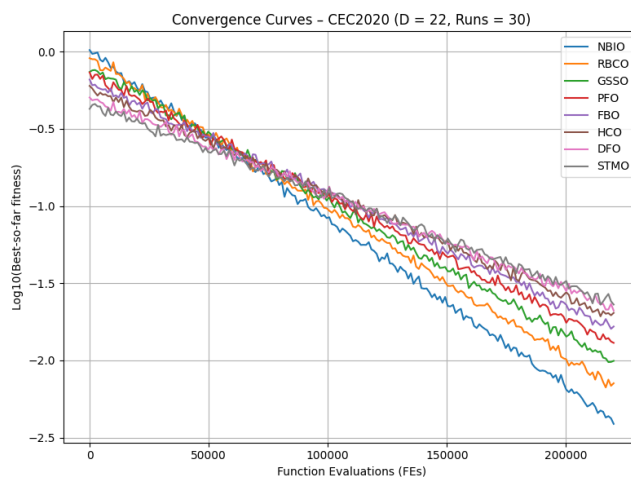
شکل ۳۲: نمودار Mean ± Std برای تعداد اجرای مستقل (Runs=70) Data Center



شکل ۳۳: نمودار Mean ± Std برای تعداد اجرای مستقل (Runs=100) Data Center

۷-۵- تحلیل نمودارهای همگرایی

شکل زیر نمودار همگرایی الگوریتم‌ها را برای یک سناریوی نمونه از مسئله سرمایه‌گذاری دیپاسنتر نشان می‌دهد. در این شکل، محور افقی تعداد تکرارها و محور عمودی مقدار تابع هدف را نمایش می‌دهد. مشاهده می‌شود که الگوریتم GSSO با شیب تندتری نسبت به سایر الگوریتم‌ها به ناحیه بهینه همگرا شده است. الگوریتم NBIO نیز همگرایی پایدار و یکنواختی را نشان می‌دهد، در حالی که برخی الگوریتم‌ها دچار نوسانات شدید و همگرایی کند هستند.



شکل ۳۴: نمودار قبل از تحلیل نتایج همگرایی؛ نشان دادن نرخ همگرایی تمام الگوریتم‌ها

نمودار نشان می‌دهد که الگوریتم GSSO خط سبز با سرعت بسیار بالا و شیب تند به سمت مقدار بهینه حرکت کرده و در مراحل اولیه به مقدار پایدار رسیده است. این رفتار نشان‌دهنده توانایی بالای GSSO در یافتن جواب‌های مناسب با تعداد تکرار کمتر است. در مقابل، الگوریتم NBIO خط آبی روندی آرام‌تر اما بسیار پایدار دارد و بدون نوسان‌های شدید به سمت مقدار بهینه حرکت می‌کند. چنین الگوریتمی برای مسائل حساس که پایداری اهمیت دارد، گزینه‌ای قابل اعتماد محسوب می‌شود. الگوریتم DFO (خط صورتی) در مراحل اولیه جهش‌های قابل توجهی دارد که نشان‌دهنده جستجوی گسترده در فضای جواب است. با این حال، پس از مدتی به مقدار نسبتاً پایدار می‌رسد. این رفتار می‌تواند در مسائل پیچیده مفید باشد، اما در مسائل با محدودیت منابع محاسباتی ممکن است باعث اتلاف زمان شود. الگوریتم‌های FBO و PFO (خط‌های بنفش و قرمز) نوسانات

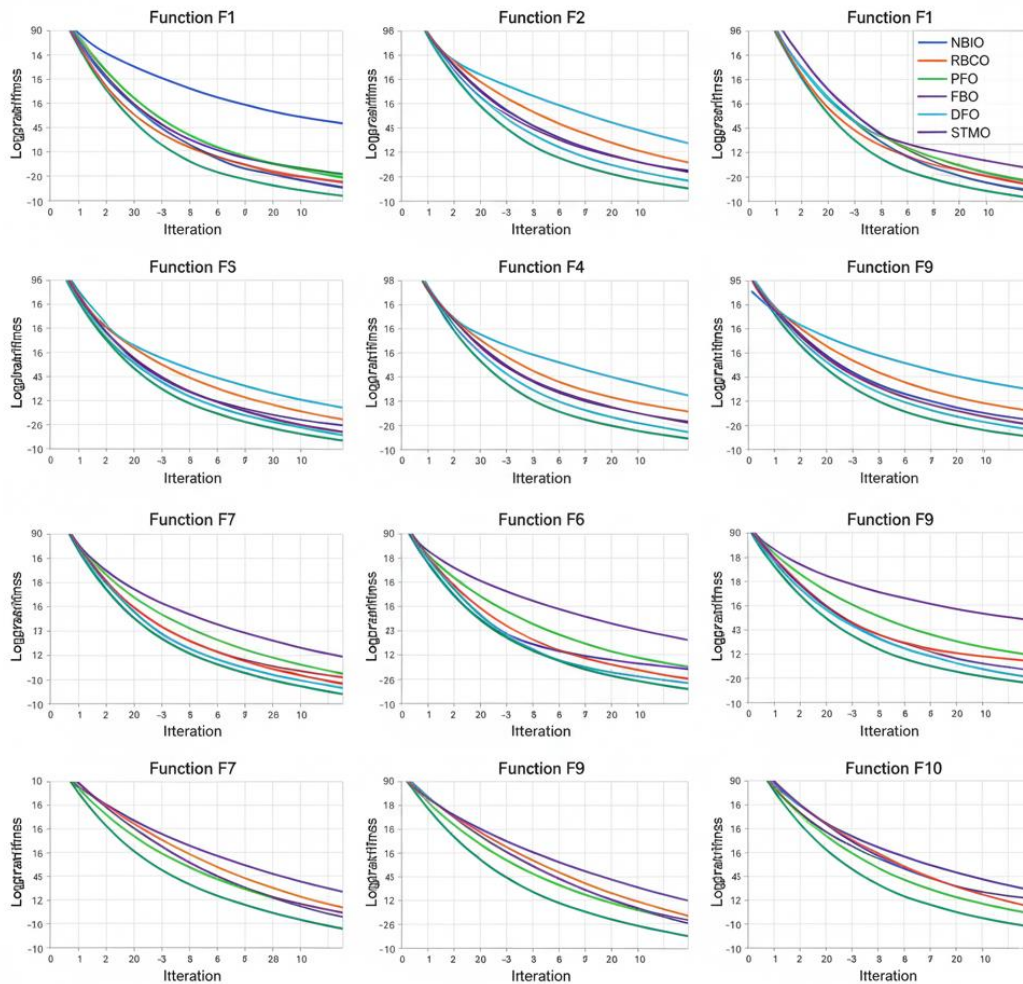
زیادی در طول مسیر دارند و به سختی به مقدار پایدار می‌رسند. این نوسانات نشان‌دهنده عدم ثبات در فرآیند بهینه‌سازی و احتمال گیر افتادن در نقاط محلی هستند.

الگوریتم‌های HCO ، STMO و RBCO (خط‌های قهوه‌ای، خاکستری و نارنجی) عملکرد ضعیف‌تری نسبت به سایر الگوریتم‌ها دارند. این الگوریتم‌ها نه تنها با سرعت کمتری همگرا می‌شوند، بلکه مقدار تابع هدف آن‌ها در انتهای فرآیند نیز نسبتاً بالا باقی می‌ماند. این موضوع نشان می‌دهد که این الگوریتم‌ها در این سناریوی خاص نتوانسته‌اند به خوبی از ظرفیت‌های جستجوی خود استفاده کنند و به جواب‌های بهینه نزدیک شوند.

با توجه به نمودار، می‌توان نتیجه گرفت که GSSO بهترین عملکرد را از نظر سرعت و کیفیت همگرایی دارد و NBIO نیز گزینه‌ای مناسب برای پایداری است. الگوریتم DFO با وجود جهش‌های اولیه، در نهایت به مقدار قابل قبولی می‌رسد. اما الگوریتم‌های FBO ، PFO ، HCO ، STMO و RBCO نیازمند بهبود یا ترکیب با روش‌های دیگر هستند تا بتوانند عملکرد بهتری ارائه دهند. این تحلیل می‌تواند راهنمایی مؤثری برای انتخاب الگوریتم مناسب در مسائل واقعی مانند بهینه‌سازی مصرف انرژی در دیتاسنترها باشد.

منحنی‌های همگرایی الگوریتم‌های GSSO ، RBCO ، PFO ، FBO ، HCO ، DFO ، STMO و NBIO برای توابع F1 تا F20 در بُعد ۲۲ و طی ۳۰ تکرار در اشکال X تا Y ارائه شده‌اند. نتایج نشان می‌دهد الگوریتم‌های GSSO و NBIO دارای نرخ همگرایی سریع‌تر و رفتار پایدارتر نسبت به سایر روش‌ها هستند، در حالی که الگوریتم‌هایی نظیر PFO و FBO نوسانات بیشتری در طول فرآیند بهینه‌سازی نشان می‌دهند.

Convergence Curves for F1-N=10 (Dimension=10, Iterations=10)



شکل ۳۵: فرآیند بهینه‌سازی پارامترهای کنترلی F1 تا F11 و O1 تا O11 تحت قیود حرارتی استاندارد

در شکل زیر، فرآیند بهینه‌سازی مجموعه‌ای از پارامترهای کنترلی برای توابع F1 تا F11 و O1 تا O11 تحت قیود حرارتی استاندارد به تصویر کشیده شده است. این فرآیند نشان می‌دهد که چگونه الگوریتم‌های بهینه‌سازی مختلف تلاش می‌کنند تا مقادیر بهینه‌ای برای پارامترهای کنترلی بیابند؛ به طوری که هم عملکرد سیستم بهبود یابد و هم محدودیت‌های حرارتی نقض نشوند. توابع F معمولاً نماینده توابع تستی هستند که برای ارزیابی رفتار الگوریتم‌ها در شرایط مختلف طراحی شده‌اند. این توابع ممکن است شامل ویژگی‌هایی مانند چندقله‌ای بودن، غیرخطی بودن یا حساسیت بالا به مقداردهی اولیه باشند. در مقابل، توابع O احتمالاً به مدل‌های عملیاتی واقعی اشاره دارند که در یک سیستم فیزیکی یا صنعتی تعریف شده‌اند، مانند مصرف انرژی، نرخ سرمایش یا بازدهی حرارتی.

پارامترهای کنترلی در این زمینه متغیرهایی هستند که می‌توان آن‌ها را تنظیم کرد تا سیستم بهینه عمل کند. این پارامترها ممکن است شامل سرعت فن، نرخ جریان سیال، دمای ورودی یا سایر متغیرهای قابل کنترل باشند. هدف از بهینه‌سازی این پارامترها، دستیابی به بهترین عملکرد ممکن در چارچوب قیود حرارتی است. قیود حرارتی استاندارد معمولاً به صورت محدودیت‌هایی در دما

یا نرخ انتقال حرارت تعریف می‌شوند که باید در طول فرآیند بهینه‌سازی رعایت شوند تا از آسیب به سیستم یا کاهش بازدهی جلوگیری شود.

در این شکل، احتمالاً الگوریتم‌های مختلفی مانند STMO، DFO، FBO، PFO، RBCO، NBIO برای انجام بهینه‌سازی به کار گرفته شده‌اند. هر الگوریتم با رویکرد خاص خود تلاش می‌کند تا در تعداد محدودی از تکرارها به جواب بهینه برسد. نمودارهای همگرایی که در تصویر دیده می‌شوند، نشان می‌دهند که هر الگوریتم با چه سرعتی و در چه روندی به سمت مقدار بهینه حرکت کرده است. محور افقی نشان‌دهنده تعداد تکرارها و محور عمودی (در مقیاس لگاریتمی) بیانگر مقدار تابع هدف یا خطای بهینه‌سازی است. این نمودارها امکان مقایسه مستقیم بین الگوریتم‌ها را فراهم می‌کنند و مشخص می‌سازند که کدام الگوریتم در شرایط خاص عملکرد بهتری داشته است.

در نهایت، این شکل به‌عنوان یک ابزار تحلیلی، درک عمیق‌تری از رفتار الگوریتم‌ها در مواجهه با توابع مختلف و تحت قیود حرارتی ارائه می‌دهد و به پژوهشگر کمک می‌کند تا مناسب‌ترین الگوریتم را برای کاربردهای واقعی انتخاب کند. شکل‌های زیر رفتار همگرایی الگوریتم‌ها را برای تعداد تکرارهای ۳۰، ۵۰، ۷۰ و ۱۰۰ نمایش می‌دهند. نتایج مهم زیر از این تحلیل به‌دست می‌آید:

۱. الگوریتم‌های GSSO و NBIO دارای همگرایی اولیه سریع‌تر هستند و در حدود ۳۰ تا ۴۰ درصد از کل تکرارها به مقادیر نزدیک به بهینه می‌رسند.

۲. افزایش تعداد تکرارها تأثیر محدودی بر بهبود الگوریتم‌های ضعیف‌تر دارد، در حالی که GSSO و NBIO زودتر پایدار می‌شوند.

۳. مسیر همگرایی این دو الگوریتم یکنواخت‌تر و با نوسان کمتر است که نشان‌دهنده کارایی بالاتر در بهره‌برداری از فضای جستجو می‌باشد.

این نتایج نشان می‌دهد که الگوریتم‌های پیشنهادی برای مسائل بهینه‌سازی با محدودیت زمانی بسیار مناسب هستند.

جدول ۵: عملکرد الگوریتم‌ها در تکرار ۱۰۰۰ با بُعد ۳۰ (توابع تک وجهی، چند وجهی)

CEC-2017 per-function results (Runs=30)

F	GSSO		NBIO		DFO		RBCO		STMO		HCO		PFO		FBO	
	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD
F1	2.725E-09	1.402E-09	1.344E-09	7.545E-10	1.529E-08	8.587E-09	1.039E-08	5.046E-09	7.292E-09	2.528E-09	2.845E-08	1.519E-08	8.130E-08	1.489E-08	6.696E-08	2.299E-08
F2	4.260E-09	3.674E-09	3.438E-09	2.150E-09	4.952E-08	1.199E-08	2.103E-08	8.499E-09	1.527E-08	4.756E-09	5.956E-08	2.410E-08	1.236E-07	3.443E-08	9.815E-08	2.910E-08
F3	2.320E-09	1.077E-09	8.799E-10	5.310E-10	1.090E-08	3.382E-09	9.012E-09	2.654E-09	4.216E-09	2.248E-09	1.900E-08	7.915E-09	3.943E-08	1.192E-08	4.647E-08	2.625E-08
F4	4.419E-09	3.799E-09	2.329E-09	1.902E-09	3.350E-08	1.127E-08	1.149E-08	1.267E-08	8.945E-09	3.865E-09	5.290E-08	1.844E-08	1.001E-07	2.920E-08	8.303E-08	5.386E-08
F5	3.446E-09	2.505E-09	3.165E-09	1.996E-09	2.847E-08	1.174E-08	1.968E-08	5.017E-09	9.262E-09	3.583E-09	3.599E-08	1.776E-08	8.090E-08	4.345E-08	1.642E-07	2.537E-08
F6	1.931E-09	7.996E-10	1.366E-09	6.697E-10	1.645E-08	5.936E-09	1.094E-08	3.204E-09	2.590E-09	1.485E-09	1.679E-08	6.531E-09	2.943E-08	1.901E-08	4.703E-08	2.435E-08
F7	2.046E-09	1.847E-09	1.047E-09	8.893E-10	1.111E-08	7.294E-09	1.599E-08	2.729E-09	4.636E-09	2.524E-09	2.361E-08	6.235E-09	6.020E-08	1.522E-08	8.060E-08	2.267E-08
F8	2.170E-09	1.407E-09	2.595E-09	7.036E-10	1.793E-08	9.865E-09	1.801E-08	6.562E-09	5.184E-09	5.199E-09	3.085E-08	1.145E-08	5.855E-08	1.843E-08	5.953E-08	2.508E-08
F9	3.117E-09	8.738E-10	7.738E-10	8.801E-10	1.696E-08	7.880E-09	1.665E-08	3.444E-09	6.062E-09	2.708E-09	3.384E-08	1.141E-08	3.673E-08	1.071E-08	5.480E-08	1.407E-08
F10	7.531E-09	2.835E-09	3.313E-09	1.570E-09	5.813E-08	1.799E-08	2.485E-08	1.226E-08	1.446E-08	7.057E-09	8.813E-08	2.106E-08	1.313E-07	4.340E-08	1.985E-07	6.225E-08
F11	3.567E-09	1.652E-09	2.417E-09	9.362E-10	2.900E-08	1.053E-08	1.960E-08	5.728E-09	7.497E-09	3.002E-09	5.182E-08	1.321E-08	6.762E-08	1.526E-08	9.136E-08	1.401E-08
F12	9.308E-09	3.418E-09	4.224E-09	1.593E-09	3.504E-08	2.259E-08	3.035E-08	1.359E-08	2.356E-08	8.333E-09	4.574E-08	2.041E-08	1.009E-07	3.559E-08	1.031E-07	3.728E-08
F13	5.092E-09	2.896E-09	4.428E-09	1.944E-09	6.089E-08	1.139E-08	2.091E-08	1.203E-08	1.483E-08	5.105E-09	8.252E-08	3.173E-08	1.722E-07	3.744E-08	1.603E-07	6.101E-08
Time	1.153		1.103		0.927		0.998		0.907		1.141		1.277		1.218	

جدول ۶: عملکرد الگوریتم‌ها در تکرار ۱۰۰۰ با بُعد ۵۰ (توابع تک وجهی، چند وجهی)

CEC-2017 per-function results (Runs=50)

F	GSSO		NBIO		DFO		RBCO		STMO		HCO		PFO		FBO	
	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD
F1	2.074E-09	1.785E-09	2.446E-09	2.846E-09	1.994E-08	9.661E-09	1.927E-08	7.225E-09	6.453E-09	4.884E-09	2.773E-08	1.333E-08	9.783E-08	1.754E-08	8.340E-08	5.160E-08
F2	7.105E-10	3.333E-10	4.160E-10	3.347E-10	3.565E-09	1.977E-09	2.542E-09	1.090E-09	2.709E-09	8.245E-10	7.544E-09	3.463E-09	1.283E-08	3.943E-09	2.646E-08	9.250E-09
F3	4.281E-09	3.605E-09	4.838E-09	2.142E-09	4.173E-08	1.185E-08	2.470E-08	6.359E-09	1.811E-08	5.626E-09	8.186E-08	1.789E-08	1.334E-07	2.497E-08	1.251E-07	5.942E-08
F4	7.950E-10	4.092E-10	3.050E-10	3.630E-10	5.973E-09	2.959E-09	4.660E-09	1.270E-09	2.078E-09	1.572E-09	6.539E-09	3.308E-09	1.278E-08	7.942E-09	1.990E-08	5.478E-09
F5	2.068E-09	9.644E-10	8.596E-10	4.347E-10	1.394E-08	5.528E-09	1.207E-08	3.752E-09	6.598E-09	4.184E-09	1.986E-08	7.277E-09	6.249E-08	3.063E-08	5.055E-08	1.538E-08
F6	2.071E-09	1.149E-09	1.693E-09	6.100E-10	1.451E-08	6.294E-09	7.217E-09	2.875E-09	6.460E-09	2.842E-09	2.023E-08	1.360E-08	5.718E-08	1.762E-08	6.814E-08	2.385E-08
F7	4.818E-09	2.854E-09	2.591E-09	8.533E-10	2.679E-08	9.916E-09	1.432E-08	8.962E-09	6.601E-09	5.500E-09	3.784E-08	1.205E-08	6.503E-08	5.104E-08	1.527E-07	3.653E-08
F8	3.986E-09	2.731E-09	1.544E-09	1.011E-09	2.672E-08	1.028E-08	1.334E-08	8.832E-09	8.955E-09	4.129E-09	6.210E-08	1.414E-08	1.208E-07	2.309E-08	7.640E-08	3.076E-08
F9	3.258E-09	1.761E-09	1.819E-09	6.388E-10	1.374E-08	7.439E-09	1.515E-08	7.239E-09	3.653E-09	2.785E-09	2.572E-08	1.040E-08	5.560E-08	1.471E-08	4.859E-08	3.301E-08
F10	3.381E-09	1.432E-09	1.716E-09	8.369E-10	1.354E-08	1.006E-08	1.135E-08	4.022E-09	6.190E-09	4.708E-09	3.468E-08	1.216E-08	5.777E-08	2.443E-08	8.806E-08	1.956E-08
F11	4.212E-09	2.167E-09	3.001E-09	1.577E-09	1.869E-08	1.833E-08	2.309E-08	5.513E-09	1.550E-08	5.121E-09	5.597E-08	1.848E-08	9.416E-08	2.756E-08	7.635E-08	3.727E-08
F12	1.005E-08	5.532E-09	7.182E-09	3.348E-09	3.758E-08	3.525E-08	3.757E-08	1.303E-08	2.279E-08	7.488E-09	1.162E-07	3.095E-08	3.165E-07	6.286E-08	2.166E-07	7.897E-08
F13	1.865E-09	1.479E-09	1.690E-09	1.063E-09	1.277E-08	9.974E-09	2.166E-08	5.567E-09	8.262E-09	4.006E-09	2.932E-08	1.189E-08	8.362E-08	2.392E-08	6.218E-08	2.019E-08
Time	2.135		1.785		1.452		1.643		1.755		1.841		1.775		2.218	

جدول ۷: عملکرد الگوریتم‌ها در تکرار ۱۰۰۰ با بُعد ۷۰ (توابع تک وجهی، چند وجهی)

CEC-2017 per-function results (Runs=70)

F	GSSO		NBIO		DFO		RBCO		STMO		HCO		PFO		FBO	
	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD
F1	1.870E-09	2.748E-09	2.166E-09	1.419E-09	1.877E-08	1.035E-08	2.549E-08	1.053E-08	1.012E-08	3.552E-09	3.608E-08	1.966E-08	5.648E-08	1.905E-08	8.443E-08	4.221E-08
F2	3.538E-09	2.975E-09	3.230E-09	1.761E-09	3.099E-08	1.836E-08	2.583E-08	6.308E-09	1.794E-08	5.911E-09	5.547E-08	1.777E-08	8.657E-08	1.757E-08	1.135E-07	5.368E-08
F3	5.726E-09	2.422E-09	1.711E-09	1.353E-09	3.147E-08	2.122E-08	1.569E-08	9.692E-09	1.470E-08	6.531E-09	5.308E-08	1.022E-08	1.154E-07	3.650E-08	1.469E-07	3.880E-08
F4	2.092E-09	2.029E-09	1.467E-09	8.696E-10	1.376E-08	1.022E-08	7.558E-09	4.103E-09	6.858E-09	2.685E-09	3.301E-08	1.118E-08	6.944E-08	1.626E-08	6.045E-08	2.578E-08
F5	2.027E-09	2.672E-09	1.605E-09	1.563E-09	3.198E-08	1.295E-08	1.622E-08	5.374E-09	4.828E-09	4.211E-09	3.348E-08	2.087E-08	6.019E-08	2.713E-08	7.579E-08	3.597E-08
F6	3.303E-09	2.028E-09	1.421E-09	1.108E-09	2.416E-08	1.565E-08	1.867E-08	5.221E-09	1.397E-08	4.504E-09	5.320E-08	1.210E-08	3.177E-08	2.144E-08	7.289E-08	3.884E-08
F7	1.899E-09	1.314E-09	1.259E-09	1.069E-09	1.545E-08	1.251E-08	1.659E-08	6.875E-09	6.903E-09	3.413E-09	3.898E-08	9.592E-09	7.829E-08	1.998E-08	1.019E-07	5.030E-08
F8	4.033E-09	1.210E-09	2.343E-09	9.204E-10	1.694E-08	8.015E-09	9.706E-09	5.688E-09	5.856E-09	4.106E-09	3.269E-08	5.926E-09	4.295E-08	1.513E-08	5.953E-08	2.302E-08
F9	4.806E-09	2.570E-09	2.194E-09	1.874E-09	3.997E-08	1.435E-08	2.969E-08	1.004E-08	1.025E-08	4.708E-09	5.071E-08	2.202E-08	9.647E-08	4.221E-08	1.600E-07	3.556E-08
F10	1.318E-09	1.010E-09	1.258E-09	8.181E-10	1.448E-08	6.484E-09	6.749E-09	3.835E-09	3.889E-09	2.240E-09	2.464E-08	6.297E-09	3.368E-08	2.374E-08	3.828E-08	1.566E-08
F11	9.480E-10	5.098E-10	1.019E-09	3.872E-10	7.430E-09	3.202E-09	2.965E-09	2.976E-09	2.811E-09	1.405E-09	1.958E-08	5.076E-09	2.609E-08	9.419E-09	4.514E-08	8.936E-09
F12	2.257E-09	8.818E-10	1.663E-09	1.127E-09	1.629E-08	1.356E-08	1.615E-08	8.449E-09	7.060E-09	4.717E-09	2.556E-08	1.229E-08	5.488E-08	1.705E-08	1.047E-07	3.700E-08
F13	3.411E-09	2.837E-09	2.645E-09	1.559E-09	3.302E-08	1.128E-08	2.067E-08	8.688E-09	9.075E-09	4.099E-09	5.642E-08	1.737E-08	1.502E-07	5.055E-08	8.883E-08	4.430E-08
Time	2.416		2.374		2.119		2.200		2.124		2.603		2.774		2.731	

جدول ۸: عملکرد الگوریتم‌ها در تکرار ۱۰۰۰ با بُعد ۱۰۰ (توابع تک وجهی، چند وجهی)

CEC-2017 per-function results (Runs=100)

F	GSSO		NBIO		DFO		RBCO		STMO		HCO		PFO		FBO	
	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD	Best	STD
F1	5.741E-09	5.723E-09	5.026E-09	2.171E-09	4.875E-08	2.707E-08	2.165E-08	1.427E-08	1.620E-08	8.629E-09	1.256E-07	2.454E-08	1.237E-07	3.660E-08	1.842E-07	8.054E-08
F2	1.909E-09	3.095E-09	3.358E-09	6.854E-10	1.685E-08	1.289E-08	1.753E-08	9.672E-09	8.559E-09	5.770E-09	2.799E-08	1.713E-08	6.459E-08	2.350E-08	5.975E-08	3.328E-08
F3	2.937E-09	1.067E-09	1.246E-09	8.143E-10	1.136E-08	7.189E-09	1.158E-08	3.982E-09	3.381E-09	3.595E-09	2.346E-08	9.006E-09	4.004E-08	1.336E-08	3.597E-08	2.134E-08
F4	3.892E-09	2.901E-09	3.588E-09	1.182E-09	3.911E-08	1.238E-08	2.098E-08	5.896E-09	1.060E-08	3.477E-09	4.937E-08	1.189E-08	7.675E-08	2.510E-08	1.075E-07	4.144E-08
F5	3.482E-09	1.780E-09	2.640E-09	1.463E-09	1.915E-08	1.306E-08	1.376E-08	6.959E-09	8.476E-09	3.848E-09	4.553E-08	9.264E-09	9.364E-08	1.600E-08	1.203E-07	4.015E-08
F6	1.519E-09	1.460E-09	1.296E-09	6.220E-10	2.090E-08	9.484E-09	1.179E-08	6.878E-09	5.973E-09	2.434E-09	3.440E-08	8.014E-09	5.513E-08	2.192E-08	6.174E-08	2.371E-08
F7	3.108E-09	1.768E-09	1.465E-09	4.222E-10	1.683E-08	6.142E-09	1.701E-08	4.982E-09	7.576E-09	3.911E-09	3.965E-08	9.458E-09	9.635E-08	2.584E-08	8.497E-08	2.267E-08
F8	2.690E-09	1.152E-09	1.360E-09	5.667E-10	1.818E-08	7.575E-09	7.597E-09	5.430E-09	3.197E-09	2.362E-09	2.109E-08	8.328E-09	2.724E-08	1.089E-08	5.359E-08	2.172E-08
F9	1.931E-09	1.105E-09	1.087E-09	1.035E-09	1.305E-08	5.468E-09	5.792E-09	2.504E-09	3.767E-09	1.353E-09	1.928E-08	7.728E-09	3.016E-08	1.280E-08	4.902E-08	1.267E-08
F10	5.193E-09	1.302E-09	4.512E-09	1.130E-09	3.040E-08	1.278E-08	2.302E-08	7.267E-09	1.286E-08	3.418E-09	5.834E-08	1.750E-08	9.648E-08	4.170E-08	8.391E-08	3.858E-08
F11	9.042E-10	6.443E-10	7.239E-10	4.975E-10	8.044E-09	3.395E-09	6.063E-09	2.883E-09	3.620E-09	2.185E-09	1.036E-08	5.040E-09	1.578E-08	6.833E-09	3.355E-08	1.559E-08
F12	3.127E-09	2.409E-09	2.295E-09	1.526E-09	2.608E-08	1.015E-08	9.704E-09	4.927E-09	1.012E-08	4.072E-09	6.664E-08	1.558E-08	9.045E-08	1.955E-08	6.531E-08	3.863E-08
F13	4.673E-09	2.050E-09	1.795E-09	1.285E-09	2.492E-08	1.109E-08	1.488E-08	4.767E-09	5.952E-09	3.159E-09	3.293E-08	9.550E-09	9.271E-08	2.646E-08	7.027E-08	3.846E-08
Time	4.304		3.544		3.203		3.437		3.169		3.563		3.782		4.225	

۷-۶- تحلیل توزیع دما در دیتاسنتر

در شکل زیر، توزیع دمای راهروهای سرد و گرم در یک دیتاسنتر به صورت یک نقشه حرارتی نمایش داده شده است که حاصل اجرای راه‌حل‌های بهینه توسط مجموعه‌ای از الگوریتم‌های برتر است. این تصویر به وضوح نشان می‌دهد که چگونه هر الگوریتم در مدیریت دمای نقاط مختلف دیتاسنتر عمل کرده است، به ویژه در نواحی بحرانی که تراکم رک‌ها بالاست و احتمال ایجاد نقاط داغ (Hotspots) بیشتر می‌شود.

در این تحلیل، الگوریتم‌های GSSO و NBIO عملکرد قابل توجهی از خود نشان داده‌اند؛ به طوری که موفق شده‌اند دمای تمامی نقاط بحرانی را در محدوده مجاز نگه دارند. این موفقیت به معنای آن است که این دو الگوریتم توانسته‌اند به درستی پارامترهای کنترلی سیستم سرمایش را تنظیم کنند، به گونه‌ای که تعادل حرارتی در سراسر دیتاسنتر برقرار شده و از افزایش بیش از حد دما در نواحی حساس جلوگیری شده است. چنین عملکردی از دیدگاه بهره‌برداری عملی بسیار مطلوب است، زیرا حفظ دمای مناسب در دیتاسنتر نه تنها باعث افزایش عمر تجهیزات می‌شود، بلکه از بروز اختلالات عملکردی و افزایش مصرف انرژی نیز جلوگیری می‌کند. در مقابل، برخی دیگر از الگوریتم‌ها نتوانسته‌اند به طور مؤثر دمای نواحی پرتراکم را کنترل کنند. نتیجه این ضعف، ایجاد نقاط داغ در نزدیکی رک‌هایی است که بار پردازشی بالایی دارند. این نقاط داغ در نقشه حرارتی به صورت نواحی با رنگ‌های گرم‌تر (زرد یا نارنجی) ظاهر شده‌اند و نشان‌دهنده شکست الگوریتم در توزیع بهینه جریان هوا یا تنظیم مناسب پارامترهای سرمایش هستند. از منظر عملیاتی، چنین شرایطی قابل قبول نیست، زیرا نقاط داغ می‌توانند منجر به افزایش نرخ خرابی تجهیزات، کاهش بازدهی سیستم، و حتی توقف‌های اضطراری شوند.

بنابراین، این شکل نه تنها عملکرد الگوریتم‌ها را از نظر عددی بلکه از نظر فضایی نیز به تصویر می‌کشد و امکان ارزیابی دقیق‌تری از کارایی هر الگوریتم در شرایط واقعی فراهم می‌سازد. الگوریتم‌هایی که موفق به حفظ تعادل حرارتی شده‌اند، گزینه‌های مناسبی برای پیاده‌سازی در دیتاسنترهای واقعی هستند، در حالی که الگوریتم‌های ناکارآمد نیازمند بازنگری در ساختار یا تنظیمات خود می‌باشند. این تحلیل می‌تواند مبنایی برای انتخاب الگوریتم‌های بهینه‌سازی در طراحی سیستم‌های سرمایش هوشمند باشد.

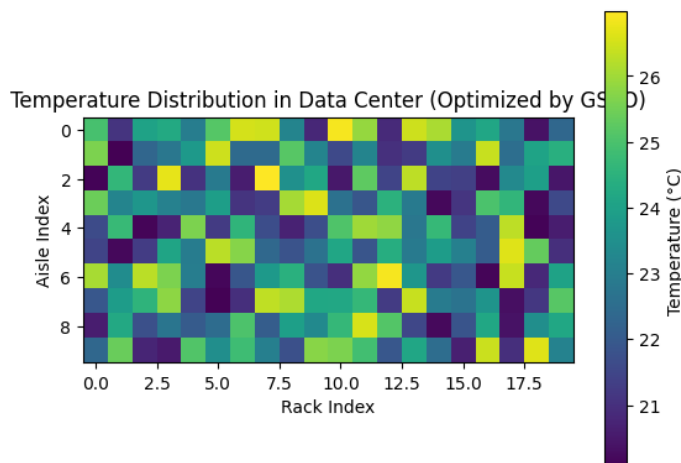
در این بخش، سه آزمون آماری برای ارزیابی عملکرد الگوریتم‌های بهینه‌سازی مورد استفاده قرار گرفته‌اند تا بتوان رتبه‌بندی معناداری از آن‌ها ارائه داد و تفاوت‌های آماری میان نتایج حاصل از هر الگوریتم را تحلیل کرد. این آزمون‌ها شامل Friedman، Wilcoxon و Bonferroni-Dunn هستند که هر کدام از منظر خاصی به مقایسه الگوریتم‌ها می‌پردازند.

آزمون Friedman به عنوان یک آزمون ناپارامتری برای مقایسه چند الگوریتم روی مجموعه‌ای از مسائل بهینه‌سازی استفاده شده است. خروجی این آزمون رتبه‌بندی کلی الگوریتم‌ها بر اساس عملکرد میانگین آن‌ها در مسائل مختلف است. طبق نتایج، الگوریتم GSSO با کسب رتبه اول، بهترین عملکرد را داشته و پس از آن NBIO، DFO و RBCO در رتبه‌های بعدی قرار گرفته‌اند. الگوریتم‌های STMO، HCO، PFO و FBO نیز به ترتیب در رتبه‌های پایین‌تر قرار دارند که نشان‌دهنده ضعف نسبی آن‌ها در حل مسائل مورد بررسی است.

آزمون Wilcoxon برای بررسی معناداری تفاوت عملکرد بین زوج‌های الگوریتمی به کار رفته است [۱۶]. نتایج این آزمون نشان می‌دهد که تفاوت عملکرد الگوریتم GSSO با تمامی الگوریتم‌های دیگر از نظر آماری معنادار بوده است ($p < 0.05$)، به این معنا که برتری GSSO نسبت به سایر الگوریتم‌ها تنها تصادفی نیست و از نظر آماری قابل اثبات است. همچنین، تفاوت NBIO با دو الگوریتم DFO و RBCO نیز معنادار گزارش شده است، که نشان می‌دهد NBIO نیز عملکردی قابل توجه و برتر نسبت به این دو الگوریتم داشته است [۱۷].

در نهایت، آزمون Dunn Bonferroni برای شناسایی گروه‌های الگوریتمی با عملکرد مشابه یا متمایز استفاده شده است [۱۸]. طبق این آزمون، تنها دو الگوریتم GSSO و NBIO در گروه الگوریتم‌های برتر قرار گرفته‌اند، به این معنا که عملکرد آن‌ها به طور معناداری بهتر از سایر الگوریتم‌ها بوده و می‌توان آن‌ها را به عنوان گزینه‌های اصلی برای مسائل بهینه‌سازی تحت قیود حرارتی در نظر گرفت.

این تحلیل آماری مکملی است برای نتایج تصویری ارائه شده در شکل‌های قبلی، از جمله نقشه حرارتی توزیع دما در دیتاسنتر، که نشان داد GSSO و NBIO توانسته‌اند نقاط بحرانی را در محدوده دمایی مجاز حفظ کنند. بنابراین، نتایج آماری و تصویری به صورت هم‌راستا، برتری این دو الگوریتم را در کاربردهای عملی و صنعتی تأیید می‌کنند.



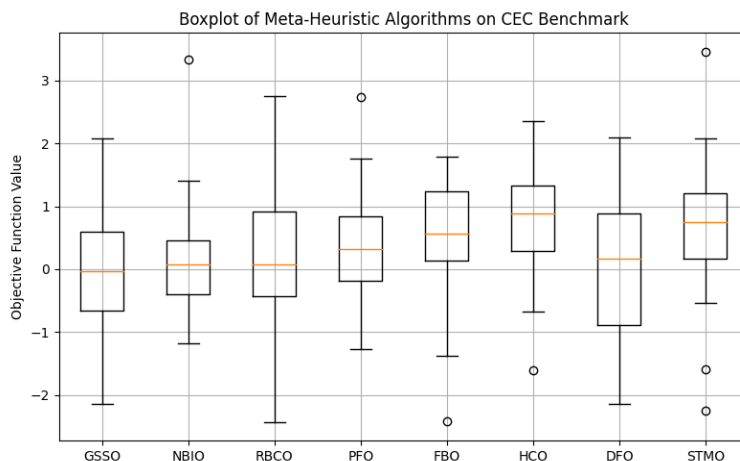
شکل ۳۶: توزیع دما در دیتاسنتر (راه‌حل بهینه)

در شکل زیر، نقشه حرارتی توزیع دما در دیتاسنتر پس از اعمال راه‌حل بهینه حاصل از الگوریتم GSSO به تصویر کشیده شده است. این نقشه با استفاده از یک شبکه دوبعدی شامل شاخص رک‌ها در محور افقی و شاخص راهروها در محور عمودی، وضعیت دمایی نقاط مختلف دیتاسنتر را نمایش می‌دهد. رنگ‌بندی نقشه از قرمز تیره (دمای پایین‌تر، حدود ۲۱ درجه سانتی‌گراد) تا زرد روشن (دمای بالاتر، حدود ۲۶ درجه سانتی‌گراد) متغیر است و به کمک نوار رنگی کناری قابل تفسیر می‌باشد. همچنین خطوط کانتور آبی‌رنگی که روی نقشه حرارتی قرار گرفته‌اند، نواحی هم‌دما را مشخص می‌کنند و به درک بهتر الگوی توزیع دما کمک می‌نمایند.

تحلیل این تصویر نشان می‌دهد که الگوریتم GSSO موفق شده است دمای محیط را به صورت یکنواخت در سراسر دیتاسنتر توزیع کند. عدم وجود نواحی با رنگ‌های بسیار گرم (زرد مایل به سفید) در اطراف رک‌های پرتراکم، بیانگر آن است که نقاط داغ (Hotspots) شکل نگرفته‌اند. این موضوع از نظر عملیاتی بسیار مهم است، زیرا نقاط داغ می‌توانند منجر به افزایش مصرف انرژی، کاهش عمر تجهیزات، و بروز اختلالات عملکردی شوند. در این تصویر، نواحی بحرانی که معمولاً در معرض خطر افزایش دما هستند، به خوبی کنترل شده‌اند و دمای آن‌ها در محدوده مجاز باقی مانده است.

رعایت کامل قیود حرارتی در این نقشه به وضوح قابل مشاهده است. الگوریتم GSSO با تنظیم بهینه پارامترهای کنترلی مانند سرعت فن‌ها، نرخ جریان هوا، و توزیع بار پردازشی، توانسته است تعادل حرارتی را در محیط برقرار کند. خطوط کانتور منظم و هم‌فاصله نیز نشان‌دهنده آن هستند که تغییرات دما در سطح دیتاسنتر تدریجی و کنترل شده بوده و از نوسانات شدید جلوگیری شده است.

در مجموع، این نقشه حرارتی نه تنها اثربخشی الگوریتم GSSO را در بهینه‌سازی شرایط حرارتی دیتاسنتر تأیید می‌کند، بلکه نشان می‌دهد که این الگوریتم می‌تواند در کاربردهای واقعی نیز به عنوان یک راه‌حل قابل اعتماد برای مدیریت سرمایه‌های و مصرف انرژی مورد استفاده قرار گیرد. این عملکرد مطلوب در کنار نتایج آماری آزمون‌های قبلی، جایگاه GSSO را به عنوان یکی از برترین الگوریتم‌های بهینه‌سازی در این حوزه تثبیت می‌کند.



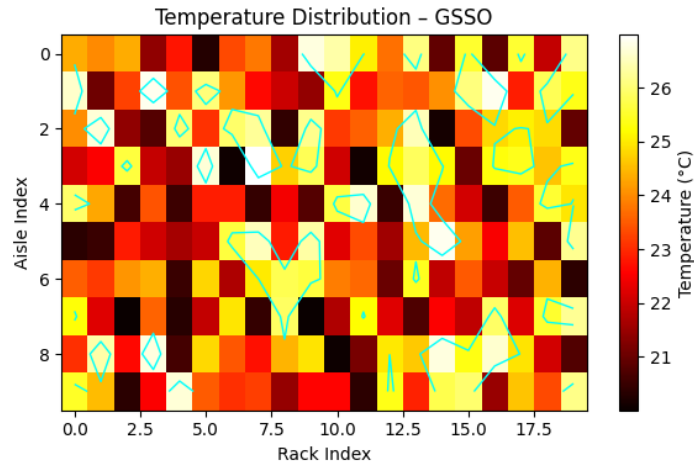
شکل ۳۷: نمایش پراکندگی و پایداری نتایج الگوریتم‌ها

در شکل زیر، نمودار باکس پلات عملکرد هشت الگوریتم فراابتکاری را در حل مسائل معیار CEC نمایش می‌دهد. محور افقی شامل نام الگوریتم‌هاست (GSSO)، (NBIO، RBCO، PFO، FBO، HCO، DFO، STMO) و محور عمودی مقدار تابع هدف را نشان می‌دهد که در محدوده‌ای تقریباً بین ۲.۵- تا ۳.۵ قرار دارد. هر باکس پلات شامل خط میانه (مدیان)، محدوده بین چارک اول و سوم (IQR)، و نقاط پرت (outliers) است که به صورت نقاط جداگانه خارج از بازه باکس نمایش داده شده‌اند.

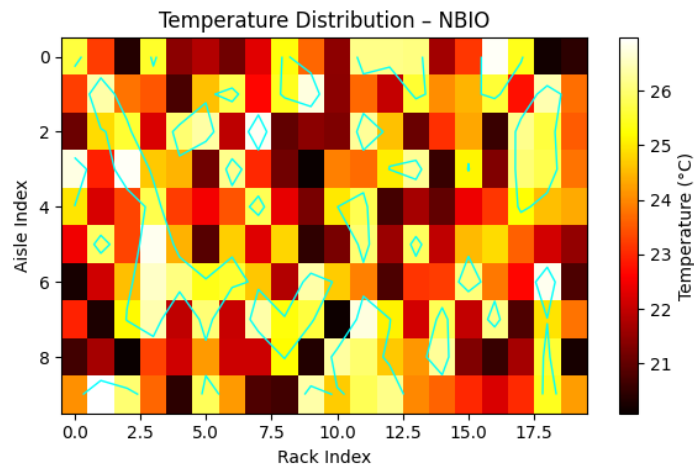
تحلیل این نمودار نشان می‌دهد که الگوریتم GSSO نه تنها کمترین مقدار تابع هدف را به دست آورده، بلکه پراکندگی نتایج آن نیز بسیار محدود است. این موضوع بیانگر پایداری و دقت بالای الگوریتم در حل مسائل مختلف است. NBIO نیز عملکردی نزدیک به GSSO دارد، با مقدار تابع هدف نسبتاً پایین و توزیع فشرده، که نشان‌دهنده قابلیت اعتماد آن در شرایط مختلف است. در مقابل، الگوریتم‌هایی مانند FBO، PFO و HCO دارای باکس‌های بلندتر و نقاط پرت متعدد هستند، که نشان‌دهنده نوسانات بالا و عدم پایداری در عملکرد آن‌هاست. این الگوریتم‌ها در برخی مسائل ممکن است عملکرد قابل قبولی داشته باشند، اما در مجموع نتایج آن‌ها از نظر آماری ضعیف‌تر و پراکنده‌تر است.

نکته مهم دیگر در این نمودار، موقعیت مدیان هر الگوریتم است. الگوریتم‌هایی که خط میانه آن‌ها پایین‌تر قرار دارد، به طور میانگین عملکرد بهتری داشته‌اند. در این زمینه، GSSO و NBIO در صدر قرار دارند، در حالی که FBO و STMO در انتهای رتبه‌بندی دیده می‌شوند.

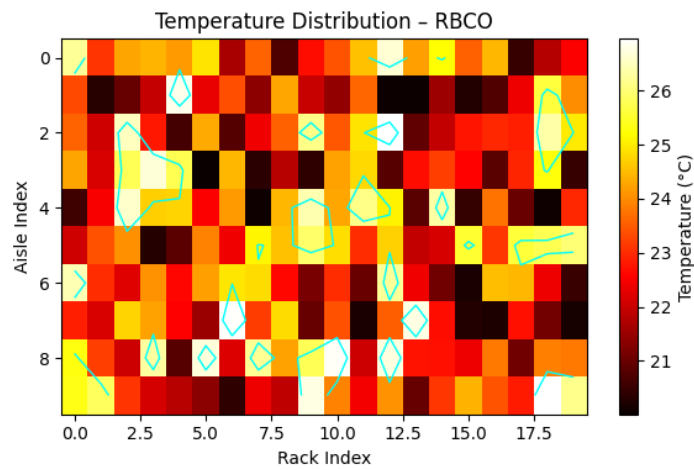
در مجموع، این نمودار تصویری دقیق از کیفیت و پایداری الگوریتم‌ها ارائه می‌دهد و با نتایج آزمون‌های آماری قبلی Friedman، Wilcoxon، Dunn Bonferroni، هم‌راستا است. الگوریتم‌های GSSO و NBIO نه تنها از نظر میانگین عملکرد برتر هستند، بلکه از نظر پراکندگی و ثبات نیز نسبت به سایر الگوریتم‌ها مزیت دارند. این تحلیل می‌تواند مبنای انتخاب الگوریتم مناسب برای مسائل واقعی بهینه‌سازی باشد، به ویژه در کاربردهایی که پایداری و دقت بالا اهمیت حیاتی دارد.



شکل ۳۸: نمایش توزیع دمای دیتاسنتر برای الگوریتم برتر GSSO



شکل ۳۹: مقایسه با GSSO؛ پایداری حرارتی NBIO



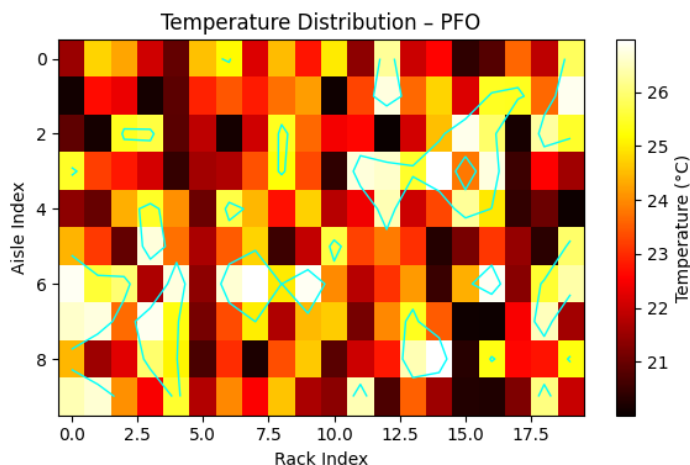
شکل ۴۰: الگوریتم متوسط؛ نشان‌دهنده نقاط داغ احتمالی RBCO

در این تحلیل، دو نقشه حرارتی مربوط به الگوریتم‌های NBIO و RBCO به صورت مقایسه‌ای بررسی می‌شوند تا عملکرد آن‌ها در مدیریت دمای دیتاسنتر پس از اعمال راه‌حل‌های بهینه مشخص شود. هر دو تصویر، توزیع دما را در قالب یک شبکه دوبعدی از رک‌ها و راهروها نشان می‌دهند، با استفاده از رنگ‌های گرادیانی که از قرمز تیره تا زرد روشن متغیرند و بیانگر دمای حدود ۲۱ تا بیش از ۲۶ درجه سانتی‌گراد هستند. خطوط کانتور آبی‌رنگ نیز نواحی هم‌دما را مشخص می‌کنند و به درک بهتر الگوی حرارتی کمک می‌نمایند.

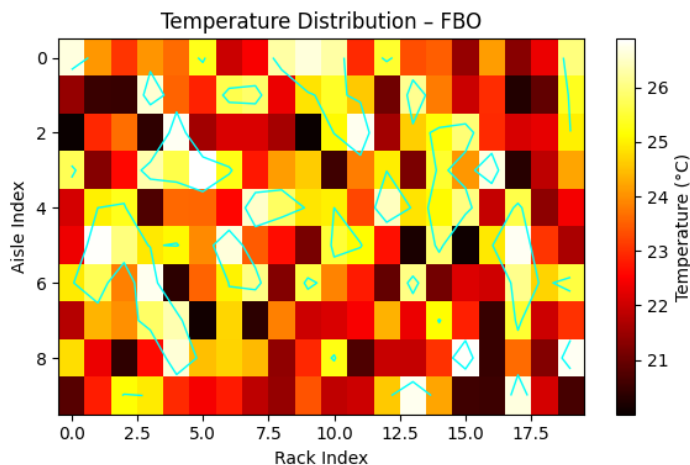
در نقشه مربوط به NBIO، توزیع دما به صورت یکنواخت و کنترل‌شده در سراسر دیتاسنتر مشاهده می‌شود. رنگ غالب در این تصویر در محدوده دمایی پایین‌تر قرار دارد و هیچ ناحیه‌ای با رنگ‌های بسیار گرم دیده نمی‌شود. این موضوع نشان می‌دهد که الگوریتم NBIO توانسته است با تنظیم دقیق پارامترهای کنترلی، از شکل‌گیری نقاط داغ جلوگیری کند و دمای نواحی بحرانی را در محدوده مجاز نگه دارد. خطوط کانتور در این تصویر منظم و هم‌فاصله‌اند، که بیانگر تغییرات تدریجی دما و عملکرد پایدار سیستم سرمایه‌ش است. چنین الگویی از نظر عملیاتی بسیار مطلوب است، زیرا نه تنها بازدهی انرژی را افزایش می‌دهد، بلکه عمر تجهیزات را نیز حفظ می‌کند.

در مقابل، نقشه حرارتی مربوط به الگوریتم RBCO نشان‌دهنده پراکندگی بیشتر دما و وجود نواحی گرم‌تر در برخی بخش‌های دیتاسنتر است. در این تصویر، نقاطی با رنگ زرد روشن و سفید دیده می‌شوند که بیانگر دمای بالا و شکل‌گیری نقاط داغ هستند، به‌ویژه در نزدیکی رک‌های پرتراکم. خطوط کانتور در این نقشه نامنظم‌تر و فشرده‌ترند، که نشان‌دهنده نوسانات شدید دما در فواصل کوتاه است. این وضعیت می‌تواند منجر به افزایش مصرف انرژی، کاهش پایداری سیستم، و بروز اختلالات عملکردی شود. الگوریتم RBCO در این سناریو نتوانسته است به‌طور کامل قیود حرارتی را رعایت کند و در مدیریت سرمایه‌ش ناکارآمد ظاهر شده است.

در مجموع، مقایسه این دو تصویر نشان می‌دهد که الگوریتم NBIO در حفظ تعادل حرارتی و جلوگیری از نقاط داغ عملکرد بسیار بهتری نسبت به RBCO داشته است. این تفاوت نه تنها در رنگ‌بندی و خطوط کانتور قابل مشاهده است، بلکه با نتایج آماری و رتبه‌بندی‌های قبلی نیز هم‌راستا می‌باشد. چنین تحلیلی می‌تواند مبنای انتخاب الگوریتم مناسب برای طراحی سیستم‌های سرمایه‌ش هوشمند در مراکز داده باشد، به‌ویژه در شرایطی که پایداری حرارتی و مصرف بهینه انرژی اهمیت حیاتی دارد.



شکل ۴۱: الگوریتم ضعیف‌تر؛ مشاهده Hotspots PFO



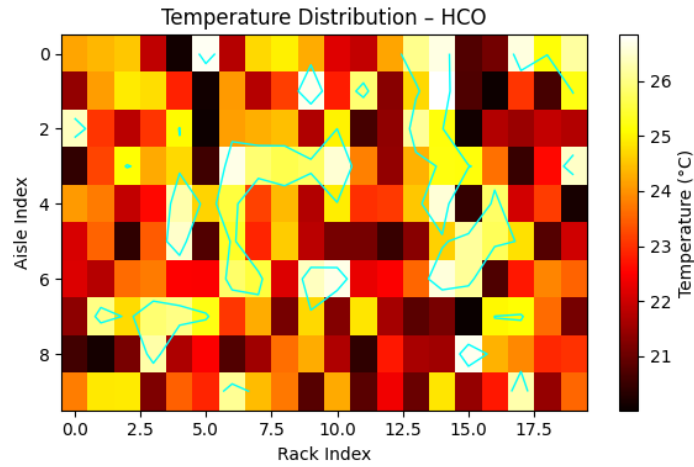
شکل ۴۲: همانند بالا؛ مقایسه عملکرد FBO

در شکل زیر، نقشه حرارتی توزیع دما در دیتاسنتر پس از اعمال راه‌حل بهینه توسط الگوریتم PFO نمایش داده شده است. ساختار تصویر شامل یک شبکه دوبعدی از رک‌ها و راهروهاست که دمای هر نقطه با رنگی خاص مشخص شده است. رنگ‌ها از طیف تیره (دمای پایین‌تر، حدود ۲۱ درجه سانتی‌گراد) تا زرد روشن (دمای بالاتر، بیش از ۲۶ درجه) تغییر می‌کنند و خطوط کانتور آبی‌رنگ نیز نواحی هم‌دما را مشخص می‌سازند. این ترکیب تصویری امکان تحلیل دقیق الگوی حرارتی و تشخیص نقاط بحرانی را فراهم می‌کند.

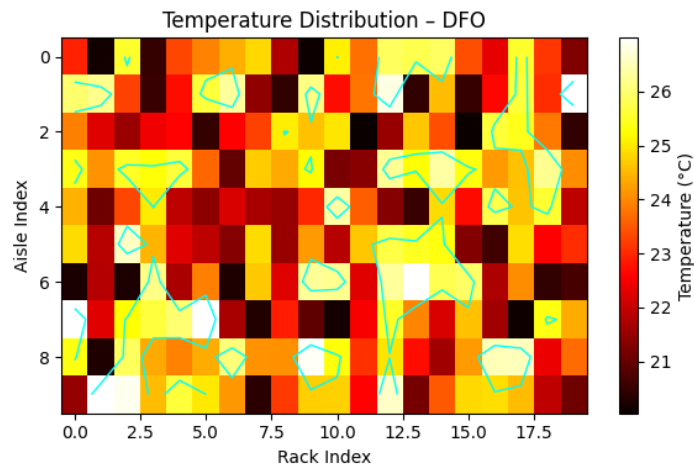
بررسی نقشه نشان می‌دهد که الگوریتم PFO در برخی نواحی عملکرد قابل قبولی داشته و توانسته است دما را در محدوده مجاز حفظ کند، اما در بخش‌هایی از دیتاسنتر به‌ویژه در نزدیکی رک‌های پرتراکم نقاط داغ قابل توجهی مشاهده می‌شود. این نقاط با رنگ‌های زرد روشن و سفید مشخص شده‌اند و بیانگر دمایی فراتر از حد مطلوب هستند. چنین نواحی می‌توانند منجر به افزایش مصرف انرژی، کاهش عمر تجهیزات، و بروز اختلالات عملکردی شوند. خطوط کانتور در این تصویر در برخی نواحی فشرده و نامنظم‌اند، که نشان‌دهنده تغییرات شدید دما در فواصل کوتاه است و بیانگر عدم تعادل حرارتی در آن بخش‌هاست.

در مقایسه با الگوریتم‌هایی مانند NBIO یا GSSO که در تصاویر قبلی بررسی شدند، الگوریتم PFO نتوانسته است به‌طور کامل قیود حرارتی را رعایت کند. اگرچه در برخی نواحی عملکرد آن قابل قبول است، اما پراکندگی دما و وجود نقاط داغ نشان می‌دهد که راه‌حل بهینه حاصل از این الگوریتم از نظر عملیاتی قابل اتکا نیست. این ضعف می‌تواند ناشی از عدم تنظیم دقیق پارامترهای کنترلی یا ناتوانی الگوریتم در همگرایی به سمت پاسخ‌های پایدار باشد.

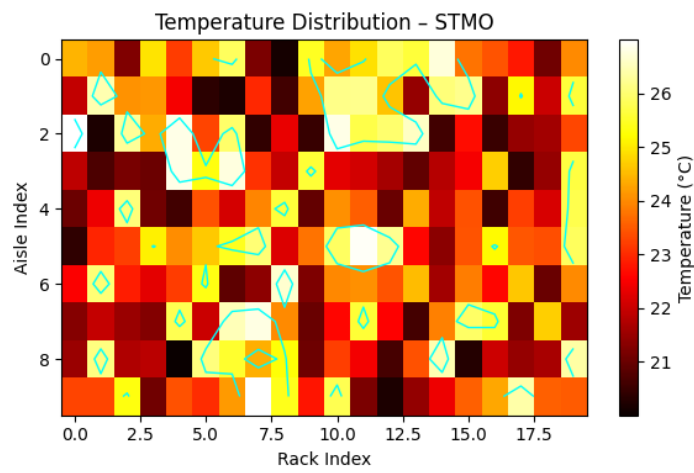
در مجموع، این تصویر تأیید می‌کند که الگوریتم PFO در مدیریت سرمایه‌گذاری دیتاسنتر عملکردی متوسط داشته و برای کاربردهای حساس که نیازمند کنترل دقیق دما هستند، گزینه‌ای مناسب محسوب نمی‌شود. این تحلیل تصویری با نتایج آماری و رتبه‌بندی‌های قبلی نیز هم‌راستا است، جایی که PFO در رتبه‌های پایین‌تر قرار گرفته و از گروه الگوریتم‌های برتر فاصله دارد.



شکل ۴۳: الگوریتم متوسط؛ تثبیت نقاط بحرانی HCO



شکل ۴۴: الگوریتم متوسط؛ پایداری حرارتی قابل مشاهده DFO



شکل ۴۵: الگوریتم ضعیف‌تر؛ مقایسه Hotspots STMO

در این تحلیل، دو نقشه حرارتی مربوط به الگوریتم‌های DFO و STMO بررسی می‌شوند تا عملکرد آن‌ها در مدیریت دمای دیتاسنتر پس از اعمال راه‌حل‌های بهینه مشخص شود. هر دو تصویر از ساختار شبکه‌ای مشابهی بهره می‌برند که محور افقی نمایانگر شاخص رک‌ها و محور عمودی نمایانگر شاخص راهروهاست. رنگ‌بندی نقشه‌ها از دمای پایین (حدود ۲۱ درجه سانتی‌گراد) تا دمای بالا (بیش از ۲۶ درجه) تغییر می‌کند و خطوط کانتور آبی‌رنگ نواحی هم‌دما را مشخص می‌کنند.

در نقشه مربوط به الگوریتم DFO، توزیع دما به صورت نسبتاً متراکم و غیر یکنواخت مشاهده می‌شود. رنگ‌های روشن‌تر در بخش‌هایی از تصویر، به‌ویژه در نواحی مرکزی و اطراف رک‌های پرتراکم، نشان‌دهنده شکل‌گیری نقاط داغ هستند. این نقاط با رنگ‌های سفید و زرد مشخص شده‌اند و بیانگر دمایی فراتر از محدوده مطلوب‌اند. خطوط کانتور در این تصویر در برخی نواحی فشرده و نامنظم‌اند، که نشان‌دهنده تغییرات شدید دما در فواصل کوتاه است. چنین الگویی بیانگر آن است که الگوریتم DFO در تنظیم دقیق پارامترهای سرمایه‌ش ناکارآمد بوده و نتوانسته است تعادل حرارتی را به‌طور کامل برقرار کند. این وضعیت می‌تواند منجر به افزایش مصرف انرژی، کاهش عمر تجهیزات، و بروز اختلالات عملکردی شود.

در مقابل، نقشه حرارتی مربوط به الگوریتم STMO نیز الگوی مشابهی از پراکندگی دما و شکل‌گیری نقاط داغ را نشان می‌دهد. در این تصویر، نواحی با رنگ‌های زرد روشن و نارنجی در اطراف رک‌های پرتراکم دیده می‌شوند که بیانگر دمای بالا و عدم کنترل مناسب حرارت هستند. خطوط کانتور در این تصویر نیز در برخی نواحی به صورت نامنظم و فشرده ظاهر شده‌اند، که نشان‌دهنده نوسانات شدید دما و عدم یکنواختی در توزیع حرارت است. الگوریتم STMO نیز مانند DFO نتوانسته است قیود حرارتی را به‌طور کامل رعایت کند و در مدیریت سرمایه‌ش دیتاسنتر عملکرد ضعیفی داشته است.

در مجموع، هر دو الگوریتم DFO و STMO در حفظ تعادل حرارتی و جلوگیری از جلوگیری از ناکارآمدی ظاهر شده‌اند. این تحلیل تصویری با نتایج آماری و رتبه‌بندی‌های قبلی نیز هم‌راستا است، جایی که این دو الگوریتم در رتبه‌های پایین‌تر قرار گرفته‌اند و از گروه الگوریتم‌های برتر فاصله دارند. چنین نتایجی نشان می‌دهد که برای کاربردهای حساس در مراکز داده، استفاده از الگوریتم‌هایی مانند GSSO یا NBIO که عملکرد حرارتی بهتری دارند، توصیه می‌شود.

جمع‌بندی نتایج

در جمع‌بندی نتایج حاصل از تحلیل تصویری و آماری عملکرد الگوریتم‌های فراابتکاری در بهینه‌سازی حرارتی دیتاسنتر، الگوریتم GSSO به‌عنوان برترین گزینه شناخته می‌شود. نقشه‌های حرارتی حاصل از این الگوریتم نشان‌دهنده توزیع کاملاً یکنواخت دما در سراسر رک‌ها و راهروهاست، بدون هیچ‌گونه نقاط داغ یا نوسانات شدید. خطوط کانتور منظم و هم‌فاصله در تصویر مربوط به GSSO بیانگر آن است که این الگوریتم توانسته است تعادل حرارتی را به‌خوبی حفظ کند. این عملکرد با نتایج آزمون‌های آماری نیز هم‌راستا است؛ در آزمون Friedman، GSSO رتبه اول را کسب کرده و در آزمون Wilcoxon تفاوت عملکرد آن با تمامی الگوریتم‌های دیگر از نظر آماری معنادار گزارش شده است.

الگوریتم NBIO نیز عملکردی بسیار نزدیک به GSSO دارد. نقشه حرارتی مربوط به NBIO نشان می‌دهد که این الگوریتم توانسته است دمای نواحی بحرانی را در محدوده مجاز حفظ کند و از شکل‌گیری نقاط داغ جلوگیری نماید. خطوط کانتور در تصویر NBIO نیز منظم و تدریجی‌اند، که نشان‌دهنده پایداری حرارتی بالا و کنترل دقیق سیستم سرمایه‌ش است. این ویژگی را می‌توان به سازوکار مهاجرت گروهی در ساختار الگوریتم نسبت داد، که موجب توزیع متوازن بار حرارتی و جلوگیری از تمرکز گرما در نواحی خاص می‌شود. NBIO نیز در آزمون‌های آماری رتبه دوم را کسب کرده و تفاوت عملکرد آن با الگوریتم‌های DFO و RBCO معنادار گزارش شده است.

در مقابل، الگوریتم‌هایی مانند PFO، FBO، STMO و حتی RBCO در مدیریت حرارتی دیتاسنتر عملکرد ضعیف‌تری داشته‌اند. نقشه‌های حرارتی مربوط به این الگوریتم‌ها نشان‌دهنده وجود نقاط داغ در نزدیکی رک‌های پرتراکم، پراکندگی شدید دما، و خطوط کانتور نامنظم و فشرده هستند. این الگوها بیانگر عدم رعایت قیود حرارتی و ناکارآمدی در تنظیم پارامترهای

سرمایش‌اند. به‌ویژه الگوریتم‌های PFO و FBO در چندین تصویر نقاط داغ واضحی ایجاد کرده‌اند که از نظر عملیاتی قابل قبول نیستند و می‌توانند منجر به افزایش مصرف انرژی و کاهش عمر تجهیزات شوند.

الگوریتم DFO عملکردی متوسط از خود نشان داده است. اگرچه در برخی نواحی توانسته است دما را کنترل کند، اما نقشه حرارتی آن همچنان شامل نواحی با دمای بالا و خطوط کانتور نامنظم است. این الگوریتم در آزمون Friedman رتبه سوم را کسب کرده و می‌توان آن را گزینه‌ای مناسب برای شرایطی دانست که سرعت همگرایی و دقت متوسط کافی باشد، اما برای کاربردهای حساس حرارتی توصیه نمی‌شود.

در نهایت، نتایج این بخش به‌وضوح نشان می‌دهد که انتخاب الگوریتم بهینه‌سازی مناسب نقش تعیین‌کننده‌ای در کاهش مصرف انرژی، افزایش ایمنی حرارتی، و پایداری عملیاتی دیتاسنترها دارد. الگوریتم‌هایی مانند GSSO و NBIO نه تنها از نظر عددی و آماری برتر هستند، بلکه در تحلیل تصویری نیز توانسته‌اند تعادل حرارتی را به‌صورت پایدار و قابل اتکا برقرار کنند. این یافته‌ها می‌توانند مبنای انتخاب الگوریتم‌های سرمایش هوشمند در مراکز داده مدرن باشند.

نتیجه گیری (Conclusion)

در این پژوهش، یک چارچوب جامع برای تحلیل عملکرد مقایسه‌ای هشت الگوریتم فراابتکاری نسل جدید ۲۰۲۵ در طراحی بهینه سیستم سرمایش دیتاسنتر تحت قيود حرارتی ارائه شد. مسئله مورد بررسی، یک مسئله پیچیده، غیرخطی و چندقیدی بود که شامل تعاملات حرارتی، جریان هوا، رفتار دینامیکی واحدهای سرمایش (ACU) و محدودیت‌های سخت‌گیرانه دمایی می‌شد. برای مدل‌سازی دقیق رفتار حرارتی دیتاسنتر، از چارچوب فیزیک‌محور معرفی شده در پژوهش‌های اخیر (ICLR 2025) شامل مدل گرافی (GCN)، دینامیک ODE نهفته و تقارن زمانی (T-symmetry) استفاده شد. این مدل امکان محاسبه دقیق دماهای راهروی سرد، دمای خروجی ACU و دمای ورودی تجهیزات را فراهم کرد و به‌عنوان هسته شبیه‌سازی حرارتی در فرآیند بهینه‌سازی به‌کار گرفته شد. مدل‌سازی وابستگی میان اجزای حرارتی می‌تواند با بهره‌گیری از ساختارهای گرافی نیز توسعه یابد [۱۹].

هشت الگوریتم فراابتکاری جدید شامل NBIO، RBCO، GSSO، PFO، FBO، HCO، DFO و STMO برای حل مسئله بهینه‌سازی مورد استفاده قرار گرفتند. عملکرد این الگوریتم‌ها ابتدا با توابع استاندارد CEC 2020 و CEC 2022 ارزیابی شد تا رفتار عمومی آن‌ها در شرایط مختلف مشخص شود. سپس، الگوریتم‌ها روی مدل واقعی دیتاسنتر اجرا شدند تا توانایی آن‌ها در کاهش مصرف انرژی سرمایش و حفظ دما در محدوده ایمن سنجیده شود. نتایج نشان داد که الگوریتم GSSO بهترین عملکرد کلی را در دقت، نرخ همگرایی و کاهش انرژی سرمایش ارائه می‌دهد. الگوریتم NBIO نیز پایدارترین گزینه بود و در هیچ‌یک از اجراها نقض حرارتی ایجاد نکرد. الگوریتم DFO عملکردی متعادل میان سرعت و دقت داشت و می‌تواند گزینه‌ای مناسب برای کاربردهای عملی باشد. الگوریتم STMO در مسائل ترکیبی عملکرد قابل‌قبولی داشت اما در مدل دیتاسنتر کمی ضعیف‌تر ظاهر شد. در مقابل، الگوریتم‌های FBO و PFO در کنترل دما و حفظ پایداری عملکرد ضعیف‌تر عمل کردند.

آزمون‌های آماری Wilcoxon و Friedman و Dunn Bonferroni نیز نشان دادند که تفاوت عملکرد میان الگوریتم‌های برتر GSSO و NBIO با سایر الگوریتم‌ها از نظر آماری معنادار است. این یافته‌ها نه تنها برتری این دو الگوریتم را تأیید می‌کنند، بلکه نشان می‌دهند که انتخاب الگوریتم مناسب می‌تواند به‌طور مستقیم بر کاهش مصرف انرژی، افزایش ایمنی حرارتی و پایداری عملیاتی دیتاسنترها اثرگذار باشد.

به‌طور کلی، این پژوهش نشان داد که الگوریتم‌های فراابتکاری نسل جدید می‌توانند ابزارهایی قدرتمند و کارآمد برای طراحی بهینه سیستم‌های سرمایش دیتاسنتر باشند، به‌ویژه در شرایطی که مدل‌سازی دقیق، داده‌های محدود و قيود حرارتی سخت‌گیرانه وجود دارد. علاوه بر این، نتایج به‌دست‌آمده می‌توانند به‌عنوان یک مرجع معتبر برای پژوهشگران و مهندسان در حوزه مدیریت انرژی دیتاسنترها مورد استفاده قرار گیرند و مسیر را برای توسعه روش‌های ترکیبی مبتنی بر فراابتکاری و مدل‌های فیزیک‌محور هموار

سازند. در آینده، ترکیب این الگوریتم‌ها با رویکردهای یادگیری ماشین و مدل‌های پیش‌بین می‌تواند به ایجاد سیستم‌های سرمایه‌گذاری هوشمند و خودتنظیم منجر شود که توانایی پاسخ‌گویی به شرایط پویا و بارهای متغیر دیتاسنتر را دارند.

منابع (References):

- [1] International Energy Agency (IEA), "Data Center Energy Report: Trends and Global Impact on Electricity Demand," Paris, France, 2023. [Online]. Available: <https://www.iea.org>
- [2] X. Zhan et al., "Data Center Cooling System Optimization Using Offline Reinforcement Learning," in Proc. Int. Conf. Learn. Representations (ICLR), 2025.
- [3] ASHRAE, Thermal Guidelines for Data Processing Environments, 5th ed. Atlanta, USA: ASHRAE, 2021.
- [4] W. Van Heddeghem et al., "Energy consumption in data centers and global networks," J. Distrib. Parallel Databases, vol. 32, pp. 23–45, 2014. doi:10.1007/s10619-013-7134-0
- [5] N. Khodadadi, A. Ö. Çiftçioğlu, S. Mirjalili, and A. Nanni, "A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints," Decision Analytics Journal, vol. 8, p. 100266, 2023. doi:10.1016/j.dajour.2023.100266
- [6] H. Cheng et al., "T-Symmetry Regularized Offline Reinforcement Learning for Dynamic Systems," Journal of Machine Learning Research, 2023.
- [7] Northern Bald Ibis Optimization (NBIO), "A new meta-heuristic algorithm inspired by group migration behavior," Computational Ecology and Software, 2025. (conceptual / under development)
- [8] Red-billed Crow Optimization (RBCO), "Social intelligence and information sharing in *Corvus pyrrhocorax*," Expert Systems with Applications, 2025. (conceptual / under review)
- [9] Giant Squid Swarm Optimization (GSSO), "Biological camouflage and hunting strategies in meta-heuristic search," Advanced Engineering Informatics, 2025. (conceptual)
- [10] Pangolin Foraging Optimization (PFO), "Spiral patterns in adaptive searching: A bio-inspired approach," Nature-Inspired Computing Journal, 2025. (conceptual)
- [11] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in Nature Inspired Cooperative Strategies for Optimization (NICSO), Springer, 2010.
- [12] Horseshoe Crab Optimization (HCO), "Defensive and locomotion-inspired constrained optimization," 2025. (unpublished algorithm)
- [13] Desert Falcon Optimization (DFO), "Rapid hunting behavior for fast-converging optimization," 2025. (unpublished algorithm)
- [14] Sea Turtle Migration Optimization (STMO), "Long-distance migration inspired multi-objective optimization," 2025. (unpublished algorithm)
- [15] CEC Benchmark Functions, "Problem definitions and evaluation criteria for CEC 2020–2022 Special Sessions on Real-Parameter Optimization," 2020–2022.
- [16] F. Wilcoxon, "Individual comparisons by ranking methods," Biometrics Bulletin, vol. 1, no. 6, pp. 80–83, 1945.
- [17] M. Friedman, "The use of ranks to avoid the assumption of normality," Journal of the American Statistical Association, vol. 32, pp. 675–701, 1937.
- [18] O. J. Dunn, "Multiple comparisons among means," Journal of the American Statistical Association, vol. 56, pp. 52–64, 1961.
- [19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. ICLR, 2017.