

روش بهینه سازی نقطه جستجوگر جهت دار داده محور با بازدهی مناسب برای توابع با تعداد متغیر بالا و ناهمواری موضعی

چکیده: در این پژوهش تلاش شده است روشی جدید برای مساله های بهینه یابی غیر خطی با تعداد متغیر بالا و فضای غیر یکنواخت تابع هزینه پیشنهاد شود. غیر یکنواخت بودن توابع و وجود تعداد زیادی اکسترمم محلی، استفاده از روش های گرادیانی را کم فایده می کنند. در اکثر مساله های مهندسی مکانیک اینگونه مساله های بهینه سازی سبب صرف زمان و محاسبات بالا می شود. همچنین نظر به تعداد متغیر بالا در فضای مساله کاملاً غیر یکنواخت، به تعداد حل زیادی در بهینه سازی نیاز است. از سویی هر بار حل یک مساله غیر خطی ممکن از دقایق یا حتی ساعتی به طول انجامد. در این روش تلاش شده است تا مساله های چند بعدی با سرعت بالاتر و تعداد دفعات حل کمتر نسبت به بقیه روش های بهینه سازی هوشمندانه از جمله روش الگوریتم ژنتیک، به پاسخی بهتر برسند. این هدف با انتخاب اتفاقی مبتنی بر تابع چگالی توزیع احتمال و وابستگی نقاط انتخابی به کلاس های سه گانه رفتاری بهینه سازی صورت می پذیرد. در واقع پس از طی چند گام مشخص می شود که هر نقطه اتفاقی جدید به کدام کلاس رفتاری مساله بهینه سازی نزدیک تر است. این کلاس های سه گانه شامل، دسته نقاط با مقدار برتر، دسته نقاط با رشد برتر و دسته نقاط متفرقه هستند.

واژه های راهنما: بهینه سازی تصادفی، آمار و احتمالات مهندسی، الگوریتم ژنتیک، بهینه سازی توده نقاط، تابع چگالی احتمال

آرمان مردانی*

استادیار،
دانشکده مهندسی مکانیک،
دانشگاه صنعتی نوشیروانی بابل،
بابل، مازندران

مقاله علمی پژوهشی

دریافت: ۱۴۰۳/۰۱/۱۵

بازنگری: ۱۴۰۳/۰۲/۱۵

پذیرش: ۱۴۰۳/۰۲/۳۱

Arman Mardani*

Assistant Professor,
Faculty of Mechanical
Engineering
Babol Noshirvani
University of
Technology, Babol,
Mazandaran

Particle directional search optimization method (PDAM) based on saved data with high efficiency for cost function specified by high order of dimension and local extremum

Abstract: This paper contributes to a new optimization method developed for hyper dimension space of solution. The most of mechanical engineering simulation-optimization problems, requires high level of arithmetic cost only for one iteration of solution. On the other hand, the complexity and non-linear trend of the problem makes the optimization method to iterates the cost function more than a simple linear one. In this paper the efforts are considered to solve optimization problem via less arithmetic cost.

Keywords: Random optimization, Probability and statistics, GA, PSO, PDF

۱- مقدمه

اجازه پرش به نقاطی خارج از مسیر قبلی را می‌دهد. این عملگر به مساله کمک می‌کند تا فضاهاى ناشناخته بیشتری را کشف کند [۱۸، ۱۹].

در روش الگوریتم ژنتیک ادعا شده است که بر مبنای روند طبیعی طبیعت عمل می‌شود. این روش که از نظریه تکامل چارلز داروین الهام گرفته شده است، با حذف افراد ضعیف‌تر از فرایند جفتگیری به صورت احتمالی، به پاسخ بهتر میل می‌کند [۲۰]. با انتقال بهترین فرد هر نسل به نسل بعد تضمین می‌شود که هیچگاه مساله در یافتن پاسخ‌های احتمالی به جوابی بدتر از گام قبل نرسد؛ اما تضمینی وجود ندارد که حتماً بهترین نقطه حاصل شود [۲۱]. به عبارتی در الگوریتم ژنتیک چندین نقطه در فضا در هر گام تولید می‌شود. هر نقطه از ترکیب اتفاقی نقاط موفق‌تر گام قبل حاصل شده است. حتی پس از چند بار شبیه‌سازی یک مساله با شرایط اولیه برابر، به خاطر عملگر جهش، پاسخ‌ها متفاوت خواهد بود [۲۲]. همچنین روش ژنتیک هیچ روندی از تغییرات و پیش‌بینی را در نظر نمی‌گیرد. تمامی مشکلات ذکر شده برای روش‌های گرادیانی و جستجو محور به نکته زیر ختم می‌شود. اول اینکه نمی‌توان با هر کدام از روش‌های گرادیانی از به دام افتادن در اکسترم‌های محلی گریخت، و دوم عدم تضمین بهترین پاسخ در روش‌های جستجو محور هوشمندانه و نیازمندی به تعداد بالای ذرات جستجو در ابعاد بالا است.

اگر ابعاد مساله افزایش یابد، به سختی می‌توان با تعداد جستجوی محدود با هر کدام از روش‌های جستجو محور به راحتی به پاسخ رسید. ذات مساله‌های چندین متغیره با روش‌هایی همچون تقسیم طلایی قابل حل نیست [۲۳]. چون در این روش‌ها فضای مساله به تدریج به دو قسمت تقسیم می‌شود و مشخص خواهد شد که در گام بعد کدام قسمت میزبان نقطه بهینه است. مشکل زمانی دوچندان می‌شود که جستجو در فضای مساله‌ای باشد که هم تعداد متغیر بالایی دارد، هم ناهمواری‌های فراوان دارد و صرفاً یک مساله بهینه‌سازی محدب نیست و نیز هر بار محاسبه تابع هزینه، بار محاسباتی فراوانی به مساله تحمیل می‌کند [۲۴]. در دنیای مهندسی مکانیک از این دست مساله‌های بحرانی فراوان قابل ذکر است. مساله بهینه‌سازی مسیر یک ربات در فضای ناهموار و عدم برخورد با موانع مشخص روی نقشه تنها یکی از مثال‌های ممکن است [۲۵، ۲۶].

در پژوهش پیش‌رو تلاش شده است تا با نقد روش الگوریتم ژنتیک و عدم تطابق آن با دنیای واقعی به روشی دست یافت که در تعداد گام مشابه در مقایسه با روش بهینه‌سازی با توده نقاط

زمانی که مساله بهینه‌سازی دارای ابعاد بالای متغیرهای بهینه‌سازی باشد؛ همچنین تابع هزینه بهینه‌سازی یک مساله بهینه‌یابی-شبیه‌سازی باشد، بهینه‌ترین روش بهینه‌سازی، روشی خواهد بود که در کمترین تعداد فراخوانی تابع، به اکسترم مد نظر برسد [۱]. روش‌های گرادیانی به عنوان ابزاری مفید در یافتن اکسترم‌ها تضمین می‌کنند که پاسخی مناسب حاصل شود [۲]؛ اما مشکل اصلی این روش‌ها به دام افتادن در اکسترم‌های محلی است [۳-۵]. یعنی اینکه هر لحظه ممکن است که حل مساله در یک مینیمم یا ماکزیمم محلی به دام بیفتد. همچنین در مورد توابع گسسته این روش‌ها با خطا مواجه می‌شوند [۶]. دلیل این امر مشتق پذیر نبودن توابع شرطی یا گسسته است در حالی که ذات روش‌های گرادیانی بر مبنای مشتق و تغییرات بنا نهاده شده است. به این منظور روش‌های غیرگرادیانی مبتنی بر جستجوی الهام گرفته شده از طبیعت خلق شده اند [۷، ۸]. روش بهینه‌سازی با توده نقاط^۱ یا PSO با ره‌سازی چندین ذره جستجوگر در فضای مساله و بر مبنای بهترین مقدار به دست آمده در هر گام برای هر عضو و عضو برتر به تشکیل بردار حرکت جدید برای هر کدام از جستجو کننده‌ها می‌پردازد [۹، ۱۰]. به عبارتی مشخص می‌کند بنا بر حرکت قبلی ذرات، هر ذره در گام جدید در کدام جهت حرکت کند تا به پاسخی بهتر برسد. در این روش عملاً ممکن است بخش‌هایی از فضای مساله جستجو نگردد [۱۱].

دلیل این امر عدم وجود انتخاب ذرات اتفاقی است. در روش کلونی مورچگان، که البته بیشتر برای یافتن مسیرهای بهینه‌تر یا ارتباطات مناسب‌تر گراف‌ها کارساز است، همین ذرات در یک گام رها می‌شوند و بنا بر ردی که از گام قبل در فضای مساله باقی مانده است، به جست و جو می‌پردازند [۱۳-۱۵]. به عبارتی یک مورچه یک مسیر را طی می‌کند و بنا بر ارزش آن مسیر در نهایت مقداری نمره به همه نقاط آن مسیر می‌افزاید. لازم به ذکر است که هیچکدام از روش‌های جستجوی هوشمندانه تضمین نمی‌کنند که الزاماً به بهترین پاسخ دست یابند [۱۶، ۱۷]. در روش‌های تکاملی، گروهی از افراد که هر کدام یک بردار متغیر هستند به عنوان یک نسل انتخاب می‌شوند. در ادامه، این نسل از تابع هدف عبور می‌کند و متناسب با شایستگی کسب شده، اجازه پیدا می‌کنند تا به صورت احتمالی نسل بعد را ایجاد کنند. میان نسل بعد از دو پارامتر اصلی سود می‌برد، ترکیب و جهش. جهش در الگوریتم ژنتیک عملگری است که به ذرات

¹ Particle swarm optimization method (PSO)

یا جستجوگر گروه ذرات (PSO) و الگوریتم ژنتیک در ساختار مساله‌هایی با ابعاد زیاد و ناهم‌واری فراوان، به پاسخی بهتر دست یابد. در واقع الگوریتم ژنتیک در تولید و تداوم شایسته سالارانه نسل به خوبی عمل می‌کند، اما یک پارامتر مهم از قلم افتاده است. در الگوریتم ژنتیک به یک فرد اجازه داده نمی‌شود که با توجه به مشخصات ثانویه پس از تولد، کمی در فضای مساله پیمایش داشته باشد [۲۷]. همچنین الگوریتم ژنتیک هیچ دیدی از روند بهبود را در شایستگی افراد در نظر نمی‌گیرد. مشکل دیگر الگوریتم ژنتیک تعداد بالای مورد نیاز در هر نسل است تا اینکه بتواند فضایی با ابعاد بالا را به خوبی پوشش دهد.

در روش پیشنهادی، در هر دور از بررسی تنها یک فرد زاده خواهد شد. از آنجایی که بار اصلی مساله بهینه‌سازی در مواجه شدن با توابع هزینه با بار محاسباتی بالا، تعداد فراخوانی تابع است، هیچ تفاوتی نمی‌کند که در ۳۰ گام شبیه‌سازی، در هر نسل ۳۰ نفر وجود داشته باشند، آنکه ۹۰۰ بار تابع فراخوانی شود. در نتیجه در روش پیشنهادی در هر گام فقط یک نفر تولید خواهد شد و به این فرد اجازه داده می‌شود که یک گام پیشروی بر مبنای مشخصات ثانویه (یا همان سوگیری ذاتی علاوه بر مقادیر ذاتی) داشته باشد. سوگیری ذاتی تنها با یک گام حرکت با اندازه اتفاقی روی یکی از پارامترها به صورت اتفاقی انجام خواهد شد. احتمال انتخاب پارامتر مورد نظر بر مبنای سابقه آماری تغییرات تابع متناسب با تغییرات پارامترها، در هر گام یک تابع چگالی احتمال گسسته را رقم خواهد زد. همچنین فرد نیز به طور اتفاقی و احتمالی انتخاب خواهد شد. این احتمال بر مبنای امید ریاضی انتخاب از نواحی مختلف بر مبنای نقاط بهتر و یا کشف نشده ماندن ناحیه، صورت می‌پذیرد [۲۸].

$$X = \arg \max f(X), X \in R^n \quad (1)$$

این معادله بیان‌کننده یک مساله بهینه‌یابی-بیشینه‌یابی است، که در آن بردار متغیرهای مساله و $f(X)$ تابع هدف مساله هستند. در ابتدا یک بردار اولیه X_0 به صورت کاملاً تصادفی در فضای مساله R^n در نظر گرفته خواهد شد. همانگونه که اشاره شد، فضای مساله هم بعد بردار $X \in R^n$ و دارای n بعد است. همچنین یک بردار اولیه حرکت به صورت تصادفی و با ساختار زیر در نظر گرفته می‌شود.

$$\begin{aligned} V &= \mathbf{0}_{1 \times n}, V[m] = s \\ s &= \text{random} \in 0 < Z < 1 \\ m &= \text{random} \in 0 < N < n + 1 \end{aligned} \quad (2)$$

فرمول بالا بیان می‌کند که بردار اولیه V دارای المان‌های s تماماً تصادفی با آدرس m است. این نحوه انتخاب اولین عضو بیان می‌کند که خود عضو X_0 بر مبنای تصادف انتخاب می‌گردد؛ همچنین مسیر نمو ثانویه لحظه‌ای عضو که با بردار V نمایش داده می‌شود، ممکن است تنها در راستای یکی از متغیرها آن هم به صورت رندوم انتخاب شود. در ادامه تشریح خواهد شد که چگونه این انتخاب بر مبنای ابعاد کاهش یافته مساله استوار می‌گردد.

در روش پیشنهادی، در هر دور از بررسی تنها یک فرد زاده خواهد شد. از آنجایی که بار اصلی مساله بهینه‌سازی در مواجه شدن با توابع هزینه با بار محاسباتی بالا، تعداد فراخوانی تابع است، هیچ تفاوتی نمی‌کند که در ۳۰ گام شبیه‌سازی، در هر نسل ۳۰ نفر وجود داشته باشند، آنکه ۹۰۰ بار تابع فراخوانی شود. در نتیجه در روش پیشنهادی در هر گام فقط یک نفر تولید خواهد شد و به این فرد اجازه داده می‌شود که یک گام پیشروی بر مبنای مشخصات ثانویه (یا همان سوگیری ذاتی علاوه بر مقادیر ذاتی) داشته باشد. سوگیری ذاتی تنها با یک گام حرکت با اندازه اتفاقی روی یکی از پارامترها به صورت اتفاقی انجام خواهد شد. احتمال انتخاب پارامتر مورد نظر بر مبنای سابقه آماری تغییرات تابع متناسب با تغییرات پارامترها، در هر گام یک تابع چگالی احتمال گسسته را رقم خواهد زد. همچنین فرد نیز به طور اتفاقی و احتمالی انتخاب خواهد شد. این احتمال بر مبنای امید ریاضی انتخاب از نواحی مختلف بر مبنای نقاط بهتر و یا کشف نشده ماندن ناحیه، صورت می‌پذیرد [۲۸].

انتخاب نقاط اتفاقی از یک احتمال مبتنی بر قاعده بیز پیروی خواهد کرد که مشخص می‌کند نقطه در کلاس نقاط با احتمال بالای بیشینگی قرار گرفته یا خیر [۲۹]. در نهایت اگر فرد با سوگیری اتفاقی به نقطه بهتری دست نیافت، سوی مخالف با همان گام به عنوان نقطه‌ای از نقاط احتمالی برای انتخاب در گام بعد در نظر گرفته خواهد شد. در این روش برای آنکه فرکانس های مختلف قابل بررسی باشند، هر گام حرکت نیز با طول اتفاقی انتخاب خواهد شد.

۲- روش پیشنهادی

در ابتدا به صورت خلاصه صورت مساله در ادامه تشریح می‌شود. فرض می‌شود که مساله بهینه‌سازی-شبیه‌سازی سنگین محاسباتی با تعداد متغیرهای بهینه‌سازی بیش از ۱۰ عدد در

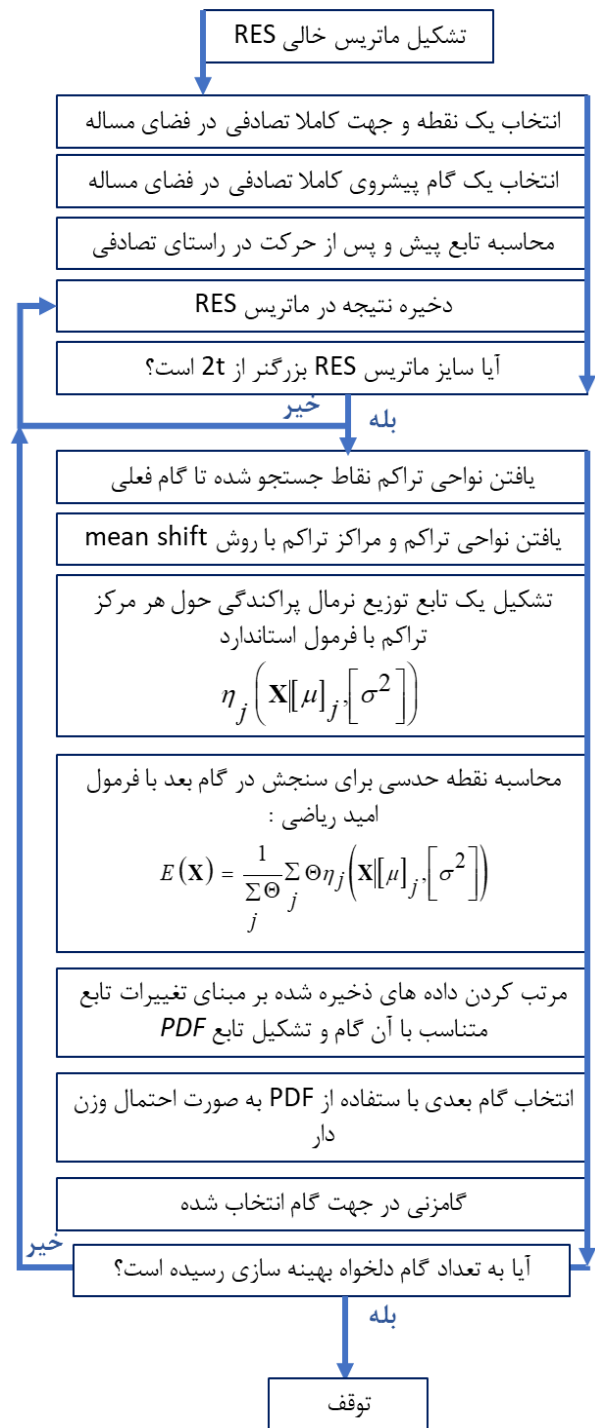
$$\begin{aligned}
 \mathbf{RES} &= [\mathbf{X} \quad \mathbf{V} \quad \mathbf{MX} \quad \mathbf{AMX} \quad \mathit{met}]_{t \times 2n+3} \\
 \mathbf{AMX} &= 1 \text{ if } F(\mathbf{X}) > F(\mathbf{X} + \mathbf{V}) \text{ else } 0 \\
 \mathbf{MX} &= \max(F(\mathbf{X}), F(\mathbf{X} + \mathbf{V})) \\
 \mathit{met} &= \text{method of movement} \in \{1, 2, 3\} \\
 \left\{ \begin{aligned}
 \mathit{met} &= 1: \text{random vector} \\
 \mathit{met} &= 2: \text{average in region} \\
 \mathit{met} &= 3: \text{most important variable} \\
 \mathit{met} &= 4: \text{average of best tries}
 \end{aligned} \right. \quad (3)
 \end{aligned}$$

که در آن اجزای بردار $\mathbf{RES} = [\mathbf{X} \quad \mathbf{V} \quad \mathbf{MX} \quad \mathbf{AMX} \quad \mathit{met}]_{t \times 2n+3}$ به ترتیب در هر گام زمانی موارد ذکر شده در ادامه را در بر خواهد گرفت. این موارد، نقطه رندوم انتخاب شده \mathbf{X} ، بردار جهت حرکت انتخاب شده \mathbf{V} ، بیشینه مقدار به دست آمده قبل \mathbf{MX} و بعد از پیمایش، آرگومان بیشینه مقدار به دست آمده قبل \mathbf{AMX} و بعد از پیمایش و نوع پیمایش met هستند.

این بدان معنی است که $2t$ بار محاسبات تابع انجام شده است و به $2t$ نقطه محاسبه شده دسترسی وجود دارد. در واقع چهار نوع پیمایش در این پژوهش در نظر گرفته شده است، که بنا بر تحلیل آماری مبتنی بر گام‌های قبلی شبیه‌سازی، گام مناسب‌تر با احتمال دستیابی به مقدار بهتر انتخاب خواهد شد. این نوع گام‌زنی‌ها شامل یک بردار کاملاً اتفاقی $\mathit{met} = 1$ ، میانگین در ناحیه تراکم $\mathit{met} = 2$ ، با اهمیت‌ترین متغیر (بنا بر سنجش حساسیت تابع به متغیر در ناحیه بنا بر داده‌های جدول) $\mathit{met} = 3$ و عملگر ترکیب بهترین‌ها به صورت میانگین آنها $\mathit{met} = 4$. برای انتخاب نقطه اتفاقی در گام بعد باید دو بردار احتمالی و یک گام اتفاقی انتخاب شود. گام اتفاقی بر مبنای تابع توزیع احتمال مرتبط با درصد موفقیت کل گام‌ها در یافتن فضایی بهتر نسبت به نقطه اولیه انتخاب خواهد شد که در ادامه تشریح خواهد شد. به عبارتی در ابتدا فرض می‌شود که یک دسته تابع چگالی احتمال گاوسی $\eta_j(\mathbf{X} | [\mu]_j, [\sigma^2])$ برای انتخاب اندازه گام جستجو در هر ناحیه در نظر گرفته خواهد شد. در واقع هر تابع گاوسی حول یک نقطه تراکم شکل خواهد گرفت که σ^2 واریانس پراکنش آن ناحیه تراکم و $[\mu]_j$ مرکز آن ناحیه تراکم هستند. ناحیه تراکم، به دسته‌ای از نقاط گفته می‌شود که در یک دسته هندسی بنا بر فاصله اقلیدسی از هم قرار گرفته‌اند. این نواحی از روش‌های مختلف یادگیری ماشین قابل استخراج و دسته‌بندی هستند.

تعداد این توابع به تعداد نقاط تراکم بستگی دارد. تعداد مراکز تراکم بهترین‌ها با فرض انتخاب از میان بهترین داده‌های ماتریس نتایج ذخیره‌شده صورت می‌گیرد. به عبارتی ممکن است هر تعداد ناحیه تراکم تا لحظه فعلی مشاهده شود. همچنین یک تابع

تمامی اطلاعات عددی مساله در یک ماتریس با عنوان \mathbf{RES} به‌طور مداوم و افزایشی در طول شبیه‌سازی ذخیره خواهد شد. این نام به‌صورت دلخواه انتخاب شده است.



شکل ۱ روند حل مساله

گام این پیشروی هم به اندازه k و عددی اتفاقی خواهد بود. حال فرض کنید که شبیه‌سازی در مرحله t از بهینه‌سازی پیشنهادی قرار گرفته است. در این مرحله سابقه‌ای از بهینه‌سازی به‌صورت زیر در اختیار است:

$$X_{new} = E(X) \text{ if } RANDOM > 0.5$$

$$\text{else random } X$$

$$E(X) = \frac{1}{\sum_j \theta} \sum_j \theta \eta_j(X | [\mu]_j, [\sigma^2]) \quad (6)$$

$$\theta = \max(j \text{ region})$$

در این معادله امید ریاضی با $E(X)$ مشخص شده است. به-عبارتی این معادله مشخص می‌کند که اگر در هر ناحیه تراکم j بیشینه مقدار به دست آمده θ باشد، نقطه بعدی تقریباً با توجه به امید ریاضی چگونه انتخاب شود تا احتمال بیشینه بودن آن بیشتر باشد. همچنین اگر عدد $RANDOM$ کمتر از 0.5 باشد، بردار متغیرها کاملاً اتفاقی انتخاب خواهد شد.

۳- عملگر انتخاب بین روش‌های پیمایش

همانگونه که در ماتریس ذخیره داده‌ها مشخص است، ستون آخر RES مربوط به روش پیمایش است. پس از آنکه نقطه جدید در محلی از فضای مساله انتخاب شد، باید سمت و سوی ذاتی حرکت نیز برای نقطه در نظر گرفته شود؛ تا بتواند یک گام پیشروی داشته باشد. به این منظور، چهار روش در نظر گرفته شده است. یکی از آنها حرکت کاملاً رندوم است که با یک بردار یکه رندوم مشخص می‌شود. در این حالت و پس از حرکت، در ستون آخر روش مرتبط با آن جستجو برابر ۱ ثبت می‌شود. روش پیمایش دوم، همانگونه که در ادامه نشان داده شده است، میانگینی از بردارهای پیمایش در سابقه نتایج برتر است.

$$V_{new} = s \frac{\Lambda_j \Phi_j}{\sum_j \Lambda_j}, \Lambda_j = \frac{1}{\mu_j - X_{new}}$$

$$\Phi_j = V \text{ in } j \text{ region whereby } \frac{df}{dX} - \max\left(\frac{df}{dX}\right) < 2 \min\left(\frac{df}{dX}\right) \quad (7)$$

به عبارت دیگر در ابتدا مشخص خواهد شد که در هر بار شبیه‌سازی، تغییرات تابع df به تغییرات متغیر dX به چه اندازه ثبت شده است. سپس کل داده‌های سابقه در ماتریس RES در ناحیه تراکم j بر مبنای این تغییرات مرتب می‌شوند. سپس آن داده‌هایی که فاصله تغییرات آنها از بیشینه تغییرات ثبت شده در ناحیه از حد خاصی کمتر باشد $-\frac{df}{dX}$ $\max\left(\frac{df}{dX}\right) < 2 \min\left(\frac{df}{dX}\right)$ به عنوان داده‌های موفق در نظر

چگالی توزیع احتمال یکنواخت غیر گاوسی نیز به صورت پیش-فرض برای نقاط دور از نقطه تراکم بهینه در نظر گرفته خواهد شد. در ابتدا فرض می‌شود که محدوده بهینه‌یابی در یک ابر مکعب مستطیل به صورت زیر است:

$$X = [x_1 \dots x_i \dots x_n]$$

$$lx_i \leq x_i \leq hx_i \quad (4)$$

در ادامه در ابتدا در هر گام به تحلیل آماری ماتریس ذخیره داده‌ها RES پرداخته می‌شود. پس از استخراج نواحی تراکم نقاط بهینه و مشخص کردن مرکز آن ناحیه تراکم با روش شیفیت در راستای افزایش تراکم نقاط^۱ که یک روش یادگیری ماشین است انجام می‌شود. که یکی از روش‌های یادگیری ماشین است، به تعداد k محل تراکم، تابع ابر گاوسی چند متغیره حول آن نقطه تراکم را با استفاده از معیار تخمین بیشینه خوش‌شانسی^۲ تشکیل می‌دهیم [۳۰]. البته به دلیل بار محاسباتی نسبتاً بالای این کار در ابعاد بالای مساله، میانگین هر توده تراکم برابر با مرکز تابع گاوسی و پراکندگی آن بر مبنای واریانس در هر بعد از نقاط هر ناحیه تراکم در نظر گرفته خواهد شد.

$$\eta_j(X | [\mu]_j, [\sigma^2]) = \frac{1}{\psi} \exp\left(-\sum_i \frac{(X_j[i] - \mu_j[i])^2}{2\sigma_i^2}\right) \quad (5)$$

که در آن $i, j, \eta_j, \mu_j[i], \sigma_i^2$ و ψ به ترتیب نماینده بعد i در محدوده تراکم j ، تابع گاوسی در آن محدوده تراکم، میانگین نقاط در محدوده، واریانس در راستای بعد i در محدوده تراکم j و عددی که انتگرال کل هر تابع گاوسی را برابر با ۱ قرار خواهد داد. در واقع اگر فرض شود که تعدادی از بردارهای وضعیت متغیرهای مساله X در یک ناحیه تراکم یا به عبارتی یک دسته هندسی نزدیک به هم X_j قرار گرفته اند، با j عدد تابع گاوسی مواجه هستیم که هر کدام نماینده یک ناحیه تراکم است. به عبارتی مشخص می‌شود که تا به حال نقاط به دست آمده در کدام نواحی فضای مساله و در چند گروه متراکم تر شده‌اند. نقطه تصادفی بعدی با توجه به ترکیب امید ریاضی این توابع^۳ $E(X)$ با یک انتخاب کاملاً اتفاقی انجام خواهد شد $randomX$ به گونه‌ای که شانس انتخاب نقطه حاصل از برون‌داد امید ریاضی با مقدار تصادفی برابر شود. این مورد در فرمول بالا نشان داده شده است.

¹ Mean shift method

² Maximum likelihood estimation-MLE

³ Expected value

Flowchart:

RES = results**RESr** = data in the close region to \mathbf{X}_{new}

sort **RESr** based on df/dX

form PDF(k) based on df/dX an linear form (۱۰)

... cumsum $\sigma^2 PDF(k)$

choose a random method based on PDF

لازم به ذکر است در چند گام نخست، تنها از انتخاب‌های کاملاً اتفاقی استفاده خواهد شد. به‌طور دلخواه این تعداد گام ۳ برابر تعداد متغیرها در نظر گرفته شده است. در فرمول بالا PDF همان تابع توزیع چگالی احتمال^۳ است. در واقع در فرمول بالا سابقه ذخیره شده بررسی می‌شود، سپس مشخص می‌شود که کدام نوع پیمایش چه مقدار تاثیر در تغییر اندازه تابع هزینه $\frac{df}{dX}$ را داشته است. متناسب با اندازه این تغییرات یک تابع توزیع احتمال تشکیل می‌شود. با استفاده از انتخاب رندوم وزن دار از تابع توزیع احتمال، نوع گام بعدی مشخص می‌گردد. در کل مشخص خواهد شد که به احتمال بیشتر کدام نوع از گام برداشتن از نقطه فعلی می‌تواند به پاسخی بهتر ختم شود.

۴- اصلاح روش انتخاب نقاط در تعداد داده مناسب

فرض کنید که داده‌های تاریخیچه مساله را در حین حل به سه کلاس تقسیم کنیم، داده‌هایی با مقادیر خوب $\{C_1\}$ ، داده‌هایی با رشد خوب $\{C_2\}$ و داده‌هایی با مقادیر نامناسب و رشد نامناسب $\{C_3\}$. احتمالاً این داده‌ها تداخل نیز خواهند داشت. حال تصور کنید که داده‌های تداخلی را تنها به‌صورت اتفاقی در یک کلاس قرار دهیم. با این کار می‌توان به یک افراز تقریبی رسید. اگر نقطه‌ای جدید \mathbf{x}_{new} انتخاب شود، با استفاده قاعده بیز می‌توان گفت که احتمال تعلق به هر کلاس چقدر خواهد بود:

$$P(C_u | \mathbf{X}_{new}) = \frac{P(\mathbf{X}_{new} | C_u) P(C_u)}{\sum_{u=1}^3 P(\mathbf{X}_{new} | C_u) P(C_u)} \quad (11)$$

به‌عبارتی در به کارگیری قاعده بیز در معادله بالا، هدف به‌عبارتی $P(C_u | \mathbf{X}_{new})$ به دست آوردن احتمال عضویت نقطه جدید انتخاب شده \mathbf{X}_{new} در یکی از سه کلاس رفتاری بهینه‌سازی C_u است. $P(C_u | \mathbf{X}_{new})$ نشان می‌دهد که به شرط انتخاب نقطه

گرفته خواهند شد. این داده‌ها در یک ضریب اطمینان ضرب می‌شوند. این ضریب اطمینان Δ_j برابر با رابطه معکوس فاصله نقطه جدید انتخاب شده \mathbf{X}_{new} تا مرکز دسته تراکم μ_j است. در روش سوم در ابتدا با استفاده از روش آماری بیشترین کوریانس داده^۱، تاثیر گذارترین متغیر انتخاب می‌شود [۳۱].

$$COV(\mathbf{X}, f(\mathbf{X})) = \begin{bmatrix} \sigma_{x_1 x_1}^2 & \cdots & \sigma_{x_1 f(x)}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{f(x) x_1}^2 & \cdots & \sigma_{f(x) f(x)}^2 \end{bmatrix} \quad (8)$$

سپس از ماتریس کوریانس بالا مشخص می‌شود که بیشترین تاثیر بر مقدار تابع از کدام متغیر حاصل شده است. اگر آدرس متغیر در بردار متغیرهای \mathbf{X} با c نمایش یابد در نتیجه:

$$\mathbf{V}_{new} = [\mathbf{O}]_{n \times 1}, \text{update } [\mathbf{V}_{new}]_c = s \quad (9)$$

بردار \mathbf{V}_{new} نماینده گام جدید حرکت با متد بیشینه واریانس در جهت یک متغیر است. سپس آن متغیر به اندازه گام s تغییر اندازه خواهد داد. این روش پیمایش به مساله بهینه‌سازی کمک می‌کند تا در برخی از گام‌ها صرفاً در جهت تغییرات تاثیر گذارترین پارامتر حرکت کند.

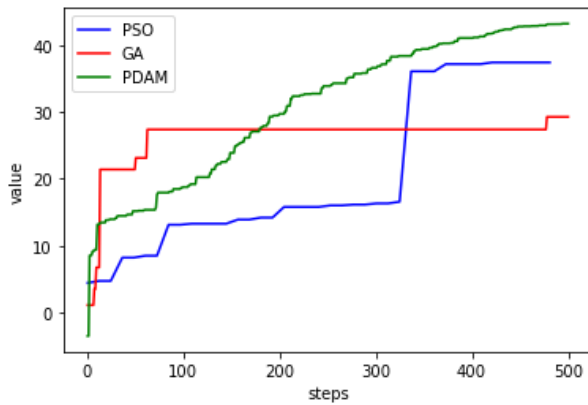
در روش چهارم، میانگین عددی از بردار مختصات نقاط برتر گرفته می‌شود (چیزی مشابه عمل ترکیب در الگوریتم ژنتیک و عمل حرکت به سمت برترین در الگوریتم PSO). به این منظور در هر محدوده نزدیک‌ترین مرکز تراکم، بردار حرکت اتفاقی یکه بر مبنای انتخاب بهترین بردار یکه که در سابقه خود بیشترین تغییر در اندازه تابع هدف را ایجاد کرده است.

عملگر انتخاب روش پیمایش بین یکی از چهار روش پیمایش، در ابتدا داده‌های ماتریس نتایج در محدوده نقطه انتخابی جدید را جدا می‌کند. سپس داده‌ها را بر مبنای بیشترین مقدار تغییر نسبت به نقطه اصلی مرتب می‌کند. یک تابع انتخاب احتمالی خطی کاهشی را به گونه‌ای شکل می‌دهد که احتمال انتخاب از داده‌های بالای جدول مرتب شده بیشتر باشد. فرض کنید که جدول جدا شده دارای ۲۰ سطر است، در نتیجه سطر اول بیشترین احتمال انتخاب روش و سطر آخر کمترین احتمال را دارد. به‌عبارتی اگر داده‌های جدا شده از جدول اصلی k عدد باشند که به‌صورت کاهشی مرتب شده‌اند، آنگاه انتخاب روش پیمایش در آن نقطه با روش زیر انجام خواهد شد:

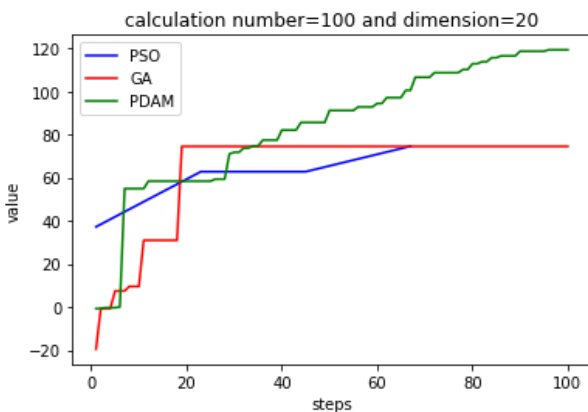
¹ Low variance filter-LVF

² Cumulative summation

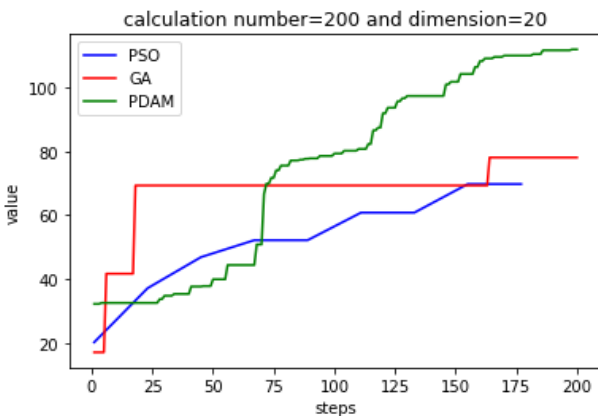
³ Probability density function



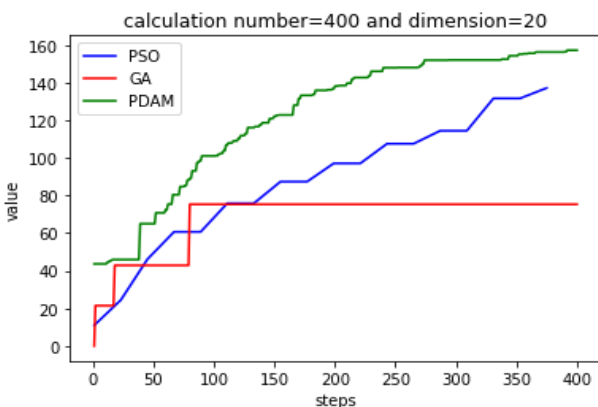
شکل ۲ روند بهینه‌یابی بر مبنای گام در سه روش



(الف)



(ب)



(ج)

چقدر احتمال دارد که این نقطه عوض کلاس رفتاری باشد. این احتمال از بررسی احتمال عضویت هر نقطه مشابه در کلاس رفتاری خاص است که با $P(X_{new}|C_u)$ مشخص می‌شود. اما مشکل اصلی جایی بروز می‌کند که ما به خود نقطه درون مجموعه دسترسی نداریم. در نتیجه فرض می‌کنیم که هر نقطه مشابه و نزدیک (در صورت وجود در سابقه) همان کلاس نقطه ما فرض خواهد شد. به عبارتی قانده بیز به صورت زیر به روز رسانی می‌شود:

$$P(C_u|X_{apx}) = \frac{P(X_{apx}|C_u)P(C_u)}{\sum_{u=1}^3 P(X_{apx}|C_u)P(C_u)} \quad (12)$$

$$X_{apx} = X_{inRES} \text{ifnorm}(X_{new} - X) \leq \lambda$$

در این صورت نقطه انتخاب شده مشخص خواهد شد که احتمالاً به کدام دسته تعلق خواهد داشت. اگر احتمال بودن در دسته اول و دوم از حدی کمتر بود، یک نقطه رندوم جایگزین خواهد شد. پس از این مرحله اصلاحی، مرحله انتخاب بردار جهت یکه انجام خواهد شد.

۵- شبیه‌سازی و مقایسه

در بخش شبیه‌سازی در ابتدا یک تابع کاملاً پیچیده با ابعاد ۱۰ و حالت موجی فراوان (شامل تعداد زیادی اکسترمم نسبی) را در نظر می‌گیریم. این تابع به صورت دلخواه و به شکل زیر تعریف می‌شود.

$$z=0$$

$$\text{for } y \text{ in range}(X \text{ size})$$

$$z=z+y \sin((y+1)X[y]) \cos(X[3]) \quad (13)$$

$$z=z+.1X[4]$$

حال نتایج شبیه‌سازی عددی مساله بهینه‌سازی در شکل (۲) نمایش داده شده است. همانگونه که از شکل مشخص است در این ابعاد خاص از تابع فرض شده با ۴۰۰ گام فراخوانی کل تابع هزینه، روش پیشنهادی نقطه جهت‌دار جستجوگر داده‌محور از دو روش رایج دیگر پیوسته‌تر و بهتر پاسخ داده است.

اما سوال اصلی اینجاست که آیا با افزایش ابعاد مساله به ۲۰ و بررسی تغییرات متناسب با تعداد فراخوانی مختلف، باز این روش برتر خواهد بود؟ در ادامه با تغییر ابعاد مساله و ثابت نگه داشتن گام‌های شبیه‌سازی نتایج نمایش یافته در شکل (۳) به دست خواهند آمد.

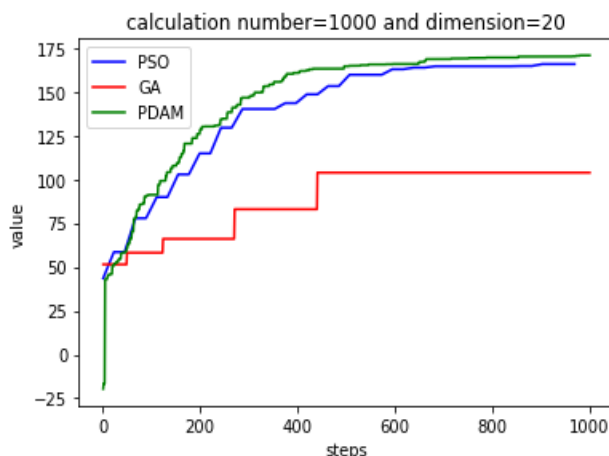
همانگونه که از شکل (۳) مشخص است، هر چقدر تعداد فراخوانی بیشتر باشد، تفاوت موفقیت روش پیشنهادی با دو روش دیگر به خصوص PSO کاهش می‌یابد. دلیل این امر اجازه دادن به دو روش مرسوم برای جستجو در ابعاد بیشتر است. برتری این روش اینجا مشخص می‌شود که با تحلیل آماری و دادن سمت و سوی احتمالی بنا بر توابع توزیع چگالی احتمال، در گام کمتری به نتیجه بهتر خواهد رسید. در واقع اگر هزینه محاسباتی تابع هزینه پایین باشد، یا اینکه ابعاد مساله کم باشد، بهتر است از دو روش مرسوم استفاده شود. چون پیاده‌سازی این روش پیشنهادی دشوارتر از روش‌های مرسوم است.

در یک تحلیل نهایی به تعداد موفقیت کل روش‌ها در مقایسه با هم پرداخته خواهد شد. در واقع به دلیل ماهیت اتفاقی موجود در بخشی از سه روش، ممکن است به‌طور اتفاقی داده مناسب‌تری از یک روش دریافت شود. در نتیجه در ادامه، ۱۰۰ بار یک مساله بهینه‌سازی با هر سه روش حل شده است و در هر بار حل، روشی که در انتها بهترین نتیجه را داشته، یک امتیاز پیروزی (محور عمودی در شکل (۴)) را دریافت کرده است. در واقع محور عمودی شکل (۴) نشان می‌دهد که در ۱۰۰ بار تکرار بهینه‌سازی، چند بار هر روش پیروز شده است.

همانگونه که مشاهده می‌شود احتمال پیروزی روش پیشنهادی با کاهش فراخوانی تابع هزینه به نسبت افزایش می‌یابد. به‌عنوان نمونه، در ۱۰۰ بار فراخوانی روش PSO تقریباً با اختلاف کمی از روش پیشنهادی شکست می‌خورد. اما در ۲۰۰ بار فراخوانی به‌طور واضح روش پیشنهادی بهتر عمل می‌کند.

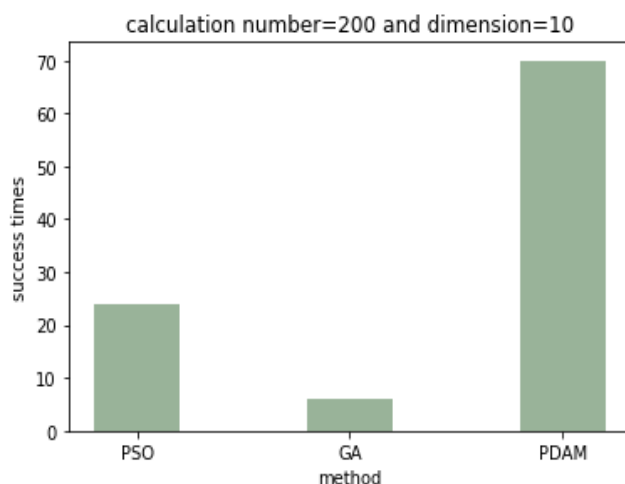
۶- نتیجه‌گیری

در این مقاله تلاش شد تا با استفاده از یک روش جستجوی مبتنی بر آمار و احتمال، به پاسخی بهتر از روش PSO و GA دست یافت. هدف مساله، رسیدن به پاسخ بهتر یک مساله بهینه‌سازی با ابعاد بالا با فراخوانی کمتر تابع هزینه بود. در این روش، نقطه تصادفی بر مبنای توابع گاوسی چگالی احتمال مطرح شده در معادله (۵) و (۷) حاصل شده است. همچنین این نقطه بر مبنای وابستگی به یکی از سه کلاس رفتاری بهینه‌سازی یعنی پاسخ بهتر، رشد بهتر و پاسخ یا رشد کمتر تصحیح شده است. نتایج حاصل شده در شکل (۲) نشان می‌دهد که در بهینه‌سازی یک تابع با ابعاد مساله بالا و تعداد اکستریم نسبی فراوان، روش جدید در مقایسه با دو روش مرسوم به نقاط بهتری از فضای حل دست می‌یابد. همچنین شکل (۳) و (۴) نشان می‌دهد که موفقیت روش در ابعاد بالاتر و تعداد گام فراخوانی

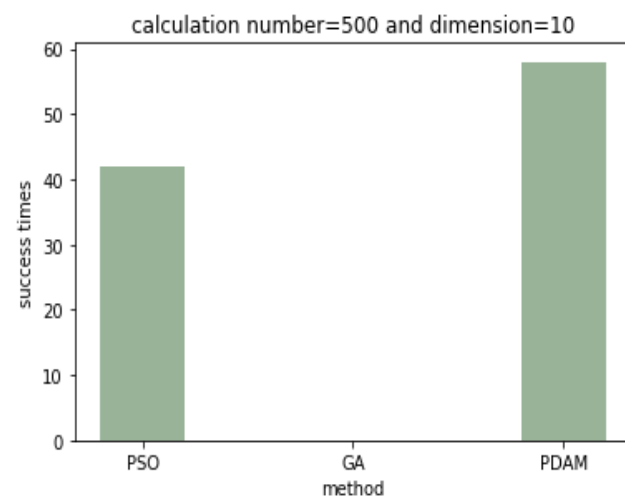


(د)

شکل ۳ روند بهینه‌یابی بر مبنای گام در سه روش با تغییر تعداد کل فراخوانی تابع هزینه. (الف) ۱۰۰ فراخوانی، (ب) ۲۰۰ فراخوانی، (ج) ۴۰۰ فراخوانی و (د) ۱۰۰۰ فراخوانی



(الف)



(ب)

شکل (۴) تعداد پیروزی‌های سه روش مختلف بر مبنای (الف) ۲۰۰ بار تکرار کل بهینه‌سازی از نخست و (ب) ۱۰۰ بار تکرار کل بهینه‌سازی از نخست

بالا، روش بهینه‌سازی پیشنهادی پاسخ بهتری دارد. در کل، اگر با یک مساله بهینه‌سازی پیچیده با تعداد نقاط اکستریم نسبی بالا و ابعاد بالا روبرو باشیم، مطابق شبیه‌سازی نشان داده شده در شکل (۴)، روش پیشنهادی پاسخ بهتری به همراه خواهد داشت.

کمتر، بیشتر است. به‌طور خاص، در نمودار شکل (۴- الف)، روش پیشنهادی تعداد موفقیت بیشتری را کسب کرده است. اما هرچه تعداد دفعات فراخوانی تابع تبدیل بیشتر می‌شود، روش‌های موجود پاسخ بهتری می‌دهند. به‌عبارتی از تعداد صد بار شبیه‌سازی مجدد مساله بهینه‌سازی روی یک تابع هزینه با ابعاد

۷- مراجع

- [9] M. Jain, V. Saijpal, N. Singh, and S. B. Singh, "An overview of variants and advancements of PSO algorithm," *Applied Sciences*, vol. 12, no. 17, p. 8392, 2022, doi: <https://doi.org/10.3390/app12178392>.
- [10] A. Tharwat and W. Schenck, "A conceptual and practical comparison of PSO-style optimization algorithms," *Expert Systems with Applications*, vol. 167, p. 114430, 2021, doi: <https://doi.org/10.1016/j.eswa.2020.114430>.
- [11] T. M. Shami, A. A. El-Saleh, M. Alswaiti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *Ieee Access*, vol. 10, pp. 10031-10061, 2022, doi: <https://doi.org/10.1109/ACCESS.2022.3142859>.
- [12] M. A. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 621-630, 2021, doi: <https://doi.org/10.1016/j.asej.2020.07.011>.
- [13] Y. Wang and Z. Han, "Ant colony optimization for traveling salesman problem based on parameters optimization," *Applied Soft Computing*, vol. 107, p. 107439, 2021, doi: <https://doi.org/10.1016/j.asoc.2021.107439>.
- [14] L. Wu, X. Huang, J. Cui, C. Liu, and W. Xiao, "Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot," *Expert Systems with Applications*, vol. 215, p. 119410, 2023, doi: <https://doi.org/10.1016/j.eswa.2022.119410>.
- [15] H. Liang, J. Zou, K. Zuo, and M. J. Khan, "An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system," *Mechanical Systems and Signal Processing*, vol. 142, p. 106708, 2020, doi: <https://doi.org/10.1016/j.ymsp.2020.106708>.
- [16] H. Moayedi, M. Raftari, A. Sharifi, W. A. W. Jusoh, and A. S. A. Rashid, "Optimization of ANFIS with GA and PSO estimating α ratio in driven piles," *Engineering with Computers*, vol. 36, no. 1, pp. 227-237, 2020, doi: <https://doi.org/10.1007/s00366-019-01517-7>.
- [1] T. Hadas and O. Schwartz, "Towards practical fast matrix multiplication based on trilinear aggregation," in *Proceedings of the 2023 International Symposium on Symbolic and Algebraic Computation*, 2023, pp. 289-297, doi: <https://doi.org/10.1145/3597066.3597099>.
- [2] P. D. Khanh, B. S. Mordukhovich, and D. B. Tran, "A new inexact gradient descent method with applications to nonsmooth convex optimization," *Optimization Methods and Software*, pp. 1-29, 2024, doi: <https://doi.org/10.1080/10556788.2024.2322700>.
- [3] M. Lapucci and P. Mansueto, "A limited memory Quasi-Newton approach for multi-objective optimization," *Computational Optimization and Applications*, vol. 85, no. 1, pp. 33-73, 2023, doi: <https://doi.org/10.1007/s10589-023-00454-7>.
- [4] K. Barkalov, I. Lebedev, and E. Kozinov, "Acceleration of global optimization algorithm by detecting local extrema based on machine learning," *Entropy*, vol. 23, no. 10, p. 1272, 2021, doi: <https://doi.org/10.3390/e23101272>.
- [5] R. Jiang and A. Mokhtari, "Accelerated quasi-newton proximal extragradient: Faster rate for smooth convex optimization," *Advances in Neural Information Processing Systems*, vol. 36, 2024, doi: <https://doi.org/10.48550/arXiv.2306.02212>.
- [6] J. R. Martins and A. Ning, *Engineering design optimization*. Cambridge University Press, 2021.
- [7] M. Sánchez, J. M. Cruz-Duarte, J. Carlos Ortíz-Bayliss, H. Ceballos, H. Terashima-Marin, and I. Amaya, "A systematic review of hyper-heuristics on combinatorial optimization problems," *IEEE Access*, vol. 8, pp. 128068-128095, 2020, doi: <https://doi.org/10.1109/ACCESS.2020.3009318>.
- [8] L. Abualigah, "Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications," *Neural Computing and Applications*, vol. 33, no. 7, pp. 2949-2972, 2021, doi: <https://doi.org/10.1007/s00521-020-05107-y>.

- [24] M. Danilova *et al.*, "Recent theoretical advances in non-convex optimization," in *High-Dimensional Optimization and Probability: With a View Towards Data Science*: Springer, 2022, pp. 79-163.
- [25] T. Osa, "Multimodal trajectory optimization for motion planning," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 983-1001, 2020, doi: <https://doi.org/10.1177/0278364920918296>.
- [26] R. Jin, P. Rocco, and Y. Geng, "Cartesian trajectory planning of space robots using a multi-objective optimization," *Aerospace Science and Technology*, vol. 108, p. 106360, 2021, doi: <https://doi.org/10.1016/j.ast.2020.106360>.
- [27] H. Chung and K.-s. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustainability*, vol. 10, no. 10, p. 3765, 2018, doi: <https://doi.org/10.3390/su10103765>.
- [28] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*. John Wiley & sons, 2020.
- [29] M. Baron, *Probability and statistics for computer scientists*. Chapman and Hall/CRC, 2019.
- [30] J. Peng, L. Li, and Y. Y. Tang, "Maximum likelihood estimation-based joint sparse representation for the classification of hyperspectral remote sensing images," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 6, pp. 1790-1802, 2018, doi: <https://doi.org/10.1109/TNNLS.2018.2874432>.
- [31] K. Hopf and S. Reifenrath, "Filter Methods for Feature Selection in Supervised Machine Learning Applications--Review and Benchmark," *arXiv preprint arXiv:2111.12140*, 2021, doi: <https://doi.org/10.48550/arXiv.2111.12140>.
- 238, 2020, doi: <https://doi.org/10.1007/s00366-018-00694-w>.
- [17] H. Chung and K.-s. Shin, "Genetic algorithm-optimized multi-channel convolutional neural network for stock market prediction," *Neural Computing and Applications*, vol. 32, no. 12, pp. 7897-7914, 2020, doi: <https://doi.org/10.1007/s00521-019-04236-3>.
- [18] H. Alibrahim and S. A. Ludwig, "Hyperparameter optimization: Comparing genetic algorithm against grid search and bayesian optimization," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, 2021: IEEE, pp. 1551-1559, doi: <https://doi.org/10.1109/CEC45853.2021.9504761>.
- [19] L. Liu, X. Su, L. Chen, S. Wang, J. Li, and S. Liu, "Elite Genetic Algorithm based self-sufficient Energy Management System for Integrated Energy Station," *IEEE Transactions on Industry Applications*, 2023, doi: <https://doi.org/10.1109/TIA.2023.3292326>.
- [20] G. Papazoglou and P. Biskas, "Review and comparison of genetic algorithm and particle swarm optimization in the optimal power flow problem," *Energies*, vol. 16, no. 3, p. 1152, 2023, doi: <https://doi.org/10.3390/en16031152>.
- [21] A. Nemirovski, *Introduction to linear optimization*. World Scientific, 2024.
- [22] C. Darwin, "Origin of the Species," in *British Politics and the Environment in the Long Nineteenth Century*: Routledge, 2023, pp. 47-55.
- [23] S. Fidanova and S. Fidanova, "Ant colony optimization," *Ant Colony Optimization and Applications*, pp. 3-8, 2021. [Online]. Available: https://books.google.com/books?id=SoogEAAAQB-AJ&lr=&source=gbs_navlinks_s.