Mathematical Researches
Print ISSN: 2588-2546
Online ISSN: 2588-2554
Kharazmi University
Homepage: https://mmr.khu.ac.ir/

A new trust-region algorithm based on radial basis function interpolation

Mohammad Ahmadvand [1] ✉, Mohsen Esmaeilbeigi [2], Ahmad Kamandi [3]
Farajollah Mohammadi Yaghoobi [4]

1. Department of Mathematics, Malayer Branch, Islamic Azad University, Malayer, Iran.

✉ E-mail: ahmadvand@malayeriau.ac.ir

2. Department of Mathematics, Malayer University, Malayer, Iran.

E-mail: m.esmaeilbeigi@malayeru.ac.ir

3. Department of Mathematics, University of Science and Technology of Mazandaran, Behshahr, Iran.

E-mail: ahmadkamandi@mazust.ac.ir

4. Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran.

E-mail: yaghoobi@iauh.ac.ir

Article Info

Article type:

Research Article

Article history:

Received:
6 December 2018
Received in revisedform:
3 August 2020
Accepted:
13 July 2020
Published online:
14 May 2022

Keywords:

Trust-region algorithm;
Radial basis function;
Unconstrained optimization;
Derivative-free algorithm;
Center changing;
Convergence test.

ABSTRACT

Introduction

In each iteration of the derivative-free trust-region algorithms, an approximate model of the optimization function is constructed. Radial basis functions are a convenient tool for building the approximate models. Some trust-region algorithms such as ORBIT use radial basis functions as an interpolation tool. Also, this algorithm stores the interpolation points and function values in each iteration. We found that these stored points could be used optimally. In this paper, we propose an improved version of ORBIT. Our new trust-region algorithm sorts the interpolation points and selects a point as the trust-region center in which the objective function reaches its minimum value. Numerical results indicate the efficiency of the improved version compared with the original version. In addition, to estimate high-accuracy solutions we equip our new algorithm with a gradient-free convergence test.

Material and methods

In this scheme, we used a point sorting strategy. In this way, we were sure that in each iteration the best possible point was the center of the trust region. Therefore, in each iteration, the approximate model is provided by better points. To compare performance of the old algorithm and the new algorithm that equipped with this strategy, we utilize Moré–Wild performance and data profiles.

Results and discussion

First, we solve a specific problem with ORBIT and the new version of ORBIT presented in this paper. The number of function evaluations, the number of iterations and the algorithm

execution time are reported separately in a table. Then for a more accurate comparison, we solve 100 unconstrained optimization problems by two algorithms. Performance profiles presented for the number of function evaluations, the number of iterations required to solve problems, and the time required to execute the algorithms.

Conclusion

The following conclusions were drawn from this research.

• Paying attention to the arrangement of interpolation points is very effective in trust-region algorithms such as ORBIT.

• Selecting the appropriate center in the trust-region algorithms leads to increasing the efficiency of the algorithms.

• In the trust-region algorithms, selecting the optimal points to build the model in each iteration increases the efficiency.

How to cite: Ahmadvand, M., Esmaeilbeigi, M., Kamandi, A., & Mohammadi Yaghoobi, F. (2022) A new trust-region algorithm based on radial basis function interpolation. *Mathematical Researches*, 8 (1), 1-20

Publisher: Kharazmi University

پژوهش‌های ریاضی

شاپا چاپی: ۲۵۸۸-۲۵٤٦
شاپا الکترونیکی: ۲۵۸۸-۲۵٥٤

Homepage: https://mmr.khu.ac.ir/

یک الگوریتم جدید ناحیهٔ اطمینان مبتنی بر درونیاب تابع پایهٔ شعاعی

محمد احمدوند[۱]✉، محسن اسماعیل بیگی[۲]، احمد کمندی[۳]، فرج الله محمدی یعقوبی[۴]

۱. نویسندهٔ مسئول، گروه ریاضیات، دانشگاه آزاد اسلامی، واحد ملایر ، ملایر، ایران.
پست الکترونیکی: ahmadvand@malayeriau.ac.ir

۲.گروه ریاضیات، دانشگاه ملایر، ملایر، ایران.
پست الکترونیکی: m.esmaeilbeigi@malayeru.ac.ir

۳.گروه ریاضیات، دانشگاه علم و فناوری مازندران، بهشهر، ایران.
پست الکترونیکی: ahmadkamandi@mazust.ac.ir

۴.گروه ریاضیات، دانشگاه آزاد اسلامی، واحد همدان، همدان، ایران.
پست الکترونیکی: ahmadkamandi@mazust.ac.ir

اطلاعات مقاله

نوع مقاله: مقاله پژوهشی

تاریخ دریافت: ۱۳۹۷/۰۹/۱۵
تاریخ بازنگری: ۱۳۹۹/۰۵/۱۳
تاریخ پذیرش: ۱۳۹۹/۰۴/۲۳
تاریخ انتشار: ۱۴۰۱/۰۲/۲۴

واژه‌های کلیدی:
الگوریتم ناحیهٔ اطمینان،
تابع پایهٔ شعاعی،
بهینه‌سازی نامقید،
الگوریتم بی‌نیاز از مشتق،
تغییر مرکز،
آزمون همگرایی.

چکیده

ORBIT یک الگوریتم بهینه‌سازی دارای ساختار ناحیه اطمینان بی‌نیاز از مشتق است. در این ساختار به جای استفاده از مدل‌های جایگزین چندجمله‌ای از مدل‌های جایگزین مبتنی بر درونیاب تابع پایه شعاعی استفاده می‌شود. بنابراین با تعداد کمتری از ارزیابی‌های تابع هدف، قادر خواهیم بود مساله بهینه‌سازی را حل نماییم. در این الگوریتم در هر تکرار، نقاط درونیاب و مقادیر تابع در آنها ذخیره شده و در تکرارهای بعدی مورد استفاده قرار می‌گیرد. با این حال این الگوریتم توجهی به مرتب کردن نقاط درونیاب نمی‌کند. در این مقاله بر اساس دو ایده، یکی مرتب کردن نقاط درونیاب بر حسب مقادیر تابع و دیگری انتخاب نقطه‌ای به عنوان مرکز ناحیه اطمینان که کمترین مقدار تابع را دارد، یک الگوریتم جدید به نام SORT-ORBIT ارائه می‌کنیم. با استفاده از این رویکرد، تعداد دفعات ارزیابی تابع و تعداد تکرارهای الگوریتم ORBIT کاهش می‌یابد. نتایج عددی حاکی از آن است که کارایی الگوریتم جدید به طور مشهودی افزایش می‌یابد. برای بررسی عملکرد الگوریتم ارائه شده در این مقاله در مقایسه با الگوریتم اصلی از شاخص کارایی دولان- موری و شاخص داده موری- ویلد استفاده شده است..

استناد: احمدوند، محمد؛ اسماعیل بیگی، محسن؛ کمندی، احمد؛ محمدی یعقوبی، فرج الله؛ (۱۴۰۱). یک الگوریتم جدید ناحیهٔ اطمینان مبتنی بر درونیاب تابع پایهٔ شعاعی. *پژوهش‌های ریاضی*، ۸ (۱)، ۲۰-۱.

ناشر: دانشگاه خوارزمی

مقدمه

یک مسأله مینیمم‌سازی سراسری مربوط به یک تابع با محاسبات سنگین به صورت زیر در نظر گرفته می‌شود

$$\min f(x); x \in \mathbb{R}^n \qquad (۱)$$

به طوری که $\mathbb{R}^n$ اشاره به فضای اقلیدسی n بعدی دارد و $f: \mathbb{R}^n \to \mathbb{R}$ یک تابع پیوسته غیرخطی مقدار حقیقی از پایین کراندار است که مشتقاتش یا در دسترس نیستند و یا محاسبهٔ تقریبی آنها بسیار پیچیده و هزینه‌بر است. این دسته از مسائل در شاخه‌های متعددی از جمله ریاضیات، علوم، مهندسی و صنعت دارای کاربرد هستند [۳، ۱۱]. به دست آوردن جواب این مسائل بهینه‌سازی اغلب به صورت مستقیم غیرممکن است، به همین جهت روش‌های عددی برای حل آنها به کار می‌روند. از بین انواع مختلف روش‌های حل مسائل بهینه‌سازی نامقید، روش‌های ناحیه اطمینان[1] یکی از معروفترین‌ها در بین روش‌های تکراری می‌باشند. چارچوب الگوریتم‌های ناحیه اطمینان توسط پاول[2] برای حل مسائل بهینه‌سازی نامقید در سال ۱۹۷۰ بنا نهاده شد[۱۵]. در همه تکرارهای الگوریتم‌های ناحیهٔ اطمینان، تابع هدف در قالب یک مدل جایگزین تقریب زده می‌شود. مدل‌های درجهٔ دوم معمول‌ترین فرم‌های مدل تقریبی در الگوریتم‌های ناحیهٔ اطمینان‌اند. در تکرار kام یک الگوریتم ناحیهٔ اطمینان، با استفاده از یک مدل تقریبی مرتبهٔ دو، s_k به عنوان یک جواب دقیق یا تقریبی برای زیرمسأله زیر تولید می‌شود

$$\begin{aligned} &\min_s m_k(s) = f_k + g_k^T s + \tfrac{1}{2} s^T B_k s \\ &s.t \quad \|s\| \le \Delta_k \end{aligned} \qquad (۲)$$

به طوری که $f_k = f(x_k)$، $g_k = \nabla f(x_k)$ و B_k ماتریس هسیان $\nabla^2 f(x_k)$ و یا تقریبی از آن است. همچنین $\|.\|$ به نرم اقلیدسی اشاره دارد و $\Delta_k > 0$ شعاع ناحیه‌ای حول نقطهٔ x_k است که ناحیهٔ اطمینان نامیده می‌شود. نتایج نظری نشان می‌دهند که مدل‌های تقریبی درجهٔ دوم به این خاطر جذابند که جوابی سراسری و غیرخطی برای زیرمسأله (۲) به دست می‌دهند. بعد از حل زیرمسأله (۲) و پیدا کردن s_k، پارامتر تست نسبت ρ_k به صورت زیر تعریف می‌شود:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m(x_k) - m(x_k + s_k)} \qquad (۳)$$

اگر ρ_k به یک نزدیک باشد، توافق خوبی بین مدل تقریبی و تابع هدف برقرار است. بنابراین، در تکرار بعدی شعاع ناحیهٔ اطمینان را افزایش می‌دهیم. بالعکس، اگر این پارامتر به صفر نزدیک باشد و یا منفی باشد مدل جایگزین نتوانسته است به خوبی تابع هدف را تقریب بزند، بنابراین شعاع ناحیه اطمینان بایستی کاهش یابد. همچنین مرکز ناحیهٔ اطمینان جدید x_{k+1} و شعاع ناحیهٔ اطمینان جدید Δ_{k+1} به ترتیب به صورت زیر به‌روز رسانی می‌شوند:

[1] Trust-region
[2] Powell

$$x_{k+1} = \begin{cases} x_k + s_k, & \rho_k \geq \eta_k \\ x_k, & otherwise \end{cases}$$

$$\Delta_{k+1} = \begin{cases} \Delta_k, & \eta_0 < \rho_k \leq \eta_1 \\ \min\{\gamma_1 \Delta_k, \Delta_{\max}\}, & \rho_k \geq \eta_1 \\ \gamma_0 \Delta_k, & \rho_k < \eta_0 \end{cases} \quad (۴)$$

طوری که $0 < \Delta_k \leq \Delta_{\max}$، $0 < \gamma_0 < 1 < \gamma_1$ و $0 \leq \eta_0 < \eta_1 < 1$.

بر خلاف مزیت‌هایی که مدل‌های تقریبی درجهٔ دو دارند، این نوع مدل‌ها به مشتق مرتبه اول تابع هدف وابسته اند. واضح است که در صورت عدم دسترسی به مشتق مرتبه اول تابع هدف، استفاده از این نوع مدل تقریبی با مشکل مواجه خواهد شد. یک راه برون رفت از این مشکل درونیابی مشتق مرتبه اول است. این ایده توسط وینفیلد[1] [۱۹] ارائه شد و سپس توسط پاول [۱۷] در الگوریتمی به نام UOBYQA و همچنین توسط کان[2] و همکارانش [۶] در الگوریتمی به نام DFO به کار گرفته شد. مشکل اساسی این الگوریتم‌ها این بود که اکیداً به نتایج درونیابی چندمتغیره وابسته بودند، بنابر این تعداد درونیابی‌های لازم در این الگوریتم‌ها تابعی درجه دو از بعد مسأله بود و در نتیجه هزینهٔ محاسباتی این الگوریتم‌ها بسیار بالا بود. مخصوصاً در مسائلی که دارای ابعاد بالا بودند این مشکل به خوبی احساس می‌شد.

راه حل دیگر استفاده از مدل‌های درونیابی بود که ترکیبی از توابع پایه‌ای غیرخطی بودند [۵]. چنین مدل‌های تقریبی را توابع پایهٔ شعاعی[3] می‌نامند. موفقیت درونیاب‌های مبتنی بر توابع پایه شعاعی را می‌توان در قابلیت آنها در درونیابی در ابعاد بالا دانست، به طوری که این نوع درونیاب رفتار عددی مناسبی از خود نشان می‌دهد. الگوریتم‌های بهینه‌سازی متعددی از درونیاب‌های مبتنی بر توابع پایهٔ شعاعی بهره گرفته‌اند. به عنوان نمونه می‌توانید به منابع [۱۲، ۱۴، ۱۸] مراجعه کنید. همچنین ویلد[4] و همکارانش در [۲۰] الگوریتمی در بحث بهینه‌سازی بی‌نیاز از مشتق ارائه کردند که در چاچوب الگوریتم ناحیه اطمینان عمل کرده و از درونیاب‌های توابع پایهٔ شعاعی استفاده می‌کند. این الگوریتم معروف با عنوان ORBIT نامگذاری شده است. در ادامه به بحث و بررسی این الگوریتم می‌پردازیم.

برای تضمین این که مدل تقریبی ساخته شده با تابع هدف و مشتق آن با مشتق تابع هدف به اندازهٔ کافی مطابقت داشته باشد، الگوریتم ORBIT هم مانند الگوریتم‌های ارائه شده توسط کان و همکارانش در[۷] از مدل‌های با شرط کاملاً خطی استفاده می‌کند. بر خلاف الگوریتم‌های ارائه شده در [۱۴، ۱۶] که تعداد نقاط درونیاب همواره عددی ثابت است، در الگوریتم ORBIT تعداد نقاط درونیاب از هر تکرار به تکرار بعدی به راحتی تغییر می‌کند. همچنین در این الگوریتم در هر تکرار یک سری نقاط اضافی به مجموعه نقاط درونیاب افزوده می‌شود. این کار بر اساس طرحی از بیجرکمن[5] [۴] در

[1] Winfield
[2] Conn
[3] Radial Basis Functions (RBFs)
[4] Wild
[5] Bjorkman

رابطه با بهینه‌سازی سراسری با استفاده از درونیاب‌های پایهٔ شعاعی انجام می‌شود. مزیت اصلی الگوریتم ORBIT نسبت به الگوریتم‌های مهم قبلی مانند [۱۴، ۱۵، ۱۶] این است که در این الگوریتم‌ها هنگامی که یک نقطه از مجموعه نقاط درونیاب حذف می‌شود در تکرارهای بعدی این نقطه ممکن است هرگز به مجموعه نقاط درونیاب بر نگردد. این درحالی‌ست که در الگوریتمORBIT وقتی یک نقطه به عنوان نقطه درونیاب مورد استفاده قرار می‌گیرد اطلاعات آن از جمله مقدار تابع هدف در این نقطه ذخیره شده و در تکرارهای بعدی باز هم از این نقطه استفاده می‌شود و چون اطلاعات این نقطه از قبل ذخیره شده است بنابراین با هزینهٔ محاسباتی کمتری مورد استفاده قرار می‌گیرد.

در این مقاله، به بررسی یکی از معایب الگوریتم ORBIT خواهیم پرداخت و با ارائهٔ یک راهکار مناسب جهت برطرف کردن این نقیصه، الگوریتم جدیدی با نام SORT-ORBIT را معرفی خواهیم کرد. با توجه به ساختار الگوریتمORBIT، حداکثر استفاده ممکن از اطلاعات ذخیره شده صورت نمی‌گیرد. یعنی از اطلاعات موجود در تکرارهای گذشته می‌توان استفاده بیشتری کرد، به این صورت که در هر تکرار مقایسه‌ای بین مقادیر به دست آمده برای تابع (اطلاعات موجود) انجام گیرد و سپس نقاط را بر حسب مقادیر تابع به ترتیب از کوچک به بزرگ مرتب کرده و هر نقطه‌ای که شامل کمترین مقدار به دست آمده برای تابع باشد را به عنوان مرکز ناحیه اطمینان انتخاب کنیم. در این صورت علاوه بر این که از بهترین نقطهٔ ممکن به عنوان مرکز ناحیه اطمینان استفاده می‌شود، مدل تقریبی جایگزین نیز بر حسب بهترین نقاط ممکن ساخته می‌شود. به این ترتیب با استفاده از مدل‌های تقریبی مناسب‌تر بهبود چشمگیری در کارایی الگوریتم حاصل خواهد شد. نتایج عددی به دست آمده پس از این تغییرات نشان می‌دهد که استفاده از این ایده‌ها در الگوریتم جدید SORT-ORBIT باعث کاهش هزینهٔ محاسباتی (تعداد فراخوانی تابع، تعداد تکرارهای الگوریتم و زمان اجرای الگوریتم) شده و بنابراین می‌توان نتیجه گرفت که الگوریتم جدید دارای کارایی بیشتری نسبت به الگوریتم ORBIT است.

ادامهٔ مقاله به این صورت تنظیم شده است: ابتدا مروری گذرا بر درونیابی مبتنی بر توابه پایهٔ شعاعی خواهیم داشت. در ادامه با الهام گرفتن از الگوریتم ORBIT چارچوب کلی الگوریتم SORT-ORBIT را معرفی می‌کنیم. سپس به ارائه نتایج عددی جهت مقایسه دو الگوریتم می‌پردازیم. در این بخش از دو شاخص کارایی دولان- موری[1] و شاخص دادهٔ موری-ویلد[2] جهت مقایسه دقیق الگوریتم‌ها استفاده می‌کنیم. در پایان به جمع‌بندی و ارائه یک نتیجه کلی از این مقاله خواهیم پرداخت.

مروری بر درونیابی توابع پایهٔ شعاعی

روش توابع پایه‌ای شعاعی یک روش بدون شبکه برای حل بسیاری از مسائل و معادلات ریاضی است. به کارگیری این روش در ابعاد بالا به دلیل خاصیت شعاعی با دشواری‌های کمتری همراه است و دقت طیفی برای انواع معینی از آنها دست‌یافتنی است. ضمناً روی دامنه‌های نامنظم به خوبی قابل پیاده‌سازی است [۵].

[1] Dolan-More'
[2] More'-Wild

با فرض این که $\Omega \subseteq \mathbb{R}^d$ باشد و $X = \{x_1,...,x_N\}$ یک مجموعه از نقاط داده‌ای متمایز در Ω باشد، درونیابی داده‌های پراکنده با استفاده از توابع پایه‌ای شعاعی برای تابع $u(x)$ بر روی X، به صورت یک ترکیب خطی از توابع پایه شعاعی به شکل زیر نوشته می‌شود:

$$s_{u,X}(x) = \sum_{j=1}^{N} \alpha_j \phi_\epsilon(\|x - x_j\|), \qquad (۵)$$

که $\|.\|$، نرم اقلیدسی است و $\phi_\epsilon: \mathbb{R}_{\geq 0} \longrightarrow \mathbb{R}$ یک تابع شعاعی می‌باشد که به پارامتر شکل[1] ϵ وابسته است به طوری که

$$\phi_\epsilon(\|x - z\|) = \phi(\epsilon\|x - z\|), \qquad x,z \in \Omega.$$

در جدول ۱، برخی از توابع پایه شعاعی پرکاربرد به همراه مرتبه همواری آنها ذکر شده است [۱۰].

جدول ۱: برخی از توابع پایهٔ شعاعی معروف

نام تابع پایه شعاعی	مرتبه همواری	$\phi_\epsilon(r)$
گاوسین	C^∞	$e^{-\epsilon^2 r^2}$
چندربعی معکوس	C^∞	$(1+\epsilon^2 r^2)^{-\frac{1}{2}}$
مترن (M4)	C^4	$e^{-\epsilon r}(\epsilon^2 r^2 + 3\epsilon r + 3)$
وندلند (W2)	C^2	$(1-\epsilon r)_+^4(4\epsilon r + 1)$

لازم به ذکر است که ضرایب حقیقی مجهول $\alpha_1,\alpha_2,...,\alpha_N$ در رابطهٔ (۵) با استفاده از شرایط درونیابی زیر تعیین می‌شوند:

$$s_{u,X}(x_i) = u(x_i), \qquad i = 1,...,N.$$

با در نظر گرفتن این شرایط، دستگاه معادلات خطی متقارنی به صورت زیر حاصل خواهد شد:

$$A\alpha = u, \qquad (۶)$$

که $A_{ij} = \phi_\epsilon(\|x_i - x_j\|)$، $\alpha = (\alpha_1,...,\alpha_N)^T$ و $u = \big(u(x_1),...,u(x_N)\big)^T$.

اگر ϕ_ϵ یک تابع معین مثبت باشد، آن‌گاه ماتریس ضرایب A وارون‌پذیر است و مسئله درونیابی توابع پایه شعاعی خوش‌وضع خواهد بود و در نتیجه، جواب مسئله موجود و یکتا می‌باشد [۱۰]. بنابراین، هنگامی که بردار α محاسبه شود، درونیاب تابع پایهٔ شعاعی را می‌توان در هر نقطه‌ای مانند x به صورت زیر ارزیابی کرد:

$$s_{u,X}(x) = \phi_\epsilon{}^T(x)\alpha, \qquad (۷)$$

که $\phi_\epsilon{}^T(x) = \Big(\phi_\epsilon(\|x - x_1\|),..., \phi_\epsilon(\|x - x_N\|)\Big)$ است.

[1] Shape parameter

در الگوریتی که در ادامه ارائه خواهیم کرد به دلیل مرتبهٔ همواری و قابلیت‌های دیگر از تابع پایهٔ شعاعی گاوسین استفاده می کنیم.

الگوریتم جدیدSORT-ORBIT

در این بخش بر اساس چارچوب کلی الگوریتم ORBIT، الگوریتم جدید SORT-ORBIT را ارائه می دهیم. به این منظور ابتدا برخی تعاریف و فرضیات لازم را مطرح خواهیم کرد.

تعریف ۱: به ازای هر زوج مرتب $(x_k, \Delta_k > 0)$ ناحیهٔ اطمینان به صورت زیر تعریف می‌شود:

$$B_k = \{x : \|x - x_k\| \le \Delta_k\} \quad (۸)$$

به طوری که $\|.\|_k$ در تکرار k ام می‌تواند نرم استاندارد، نرم دو یا هر نرم دیگری باشد.

تعریف ۲: در تکرار k ام همه الگوریتم‌های ناحیهٔ اطمینان، زیرمسأله‌ای به شکل زیر:

$$\min_s \{m_k(x_k + s): x_k + s \in B_k\} \quad (۹)$$

حل می‌شود. این زیرمسأله، زیرمسأله ناحیأ اطمینان نامیده می‌شود.

تعریف ۳: فرض کنید که $f \in C^1[B_k]$ باشد. برای مقادیر ثابت $\kappa_f, \kappa_g > 0$، مدل تقریبی m، خطی کامل[1] روی ناحیه اطمینان B_k نامیده می‌شود هرگاه برای هر $x \in B_k$ دو شرط زیر برقرار باشند:

$$|f(x) - m(x)| \le \kappa_f \Delta^2 \quad (۱۰)$$

$$\|\nabla f(x) - \nabla m(x)\| \le \kappa_g \Delta \quad (۱۱)$$

تذکر۱: همان طور که گفته شد در هر تکرار بایستی زیرمسأله ناحیهٔ اطمینان را حل کنیم. برای تضمین همگرایی سراسری برای زیرمسأله در یافتن جواب s_k کافی است مدل تقریبی m_k که $\nabla^2 m_k \ne 0$ و $\|\nabla^2 m_k\| \le \kappa_H$، دارای شرط کاهش کافی[2] باشد یعنی:

$$m_k(x_k) - m_k(x_k + s_k) \ge \frac{\kappa_d}{2} \|\nabla m_k(x_k)\| \min\left\{\frac{\nabla m_k(x_k)}{\kappa_H}, \frac{\|\nabla m_k(x_k)\|}{\|\nabla m_k(x_k)\|_k} \Delta_k\right\} \quad (۱۲)$$

به طوری که $\kappa_d \in (0,1)$ عددی ثابت است.

[1] Fully linear
[2] Sufficient decrease condition

اکنون با توجه به تعاریف بالا آماده‌ایم که تکرار kام الگوریتم جدید یعنی SORT-ORBIT را در الگوریتم ۱ معرفی کنیم. نقطهٔ شروع اولیه $x_0 \in \mathbb{R}^n$ و ورودی‌های $\varepsilon_g, \kappa_g, \kappa_f > 0$ ، $p_{\max} > n+1$، $0 \le \eta_0 < \eta_1 < 1$، $0 < \gamma_0 < 1 < \gamma_1$ و $0 < \Delta_0 \le \Delta_{\max}$ را به عنوان مقادیر مفروض در نظر بگیرید.

الگوریتم ۱. تکرار kام الگوریتم SORT-ORBIT

گام ۱: نقاط درونیاب را بر اساس مقادیر تابع مرتب کنید و از میان آنها نقطه x^* با کمترین مقدار تابع هدف را به عنوان مرکز جدید ناحیهٔ اطمینان B_k قرارداده و آنرا x_k بنامید.

گام ۲: برای به دست آوردن یک مدل کاهش کافی $n+1$ نقطه مستقل آفینی پیدا کنید.

گام ۳: برای بهره‌مندی از مزیت غیرخطی بودن توابع پایهٔ شعاعی $p_{\max} - n - 1$ نقطهٔ اضافی به مجموعه نقاط درونیاب اضافه کنید.

گام ۴: مدل تقریبی مبتنی بر تابع پایه شعاعی را بسازید.

گام ۵: تا زمانی که $\|\nabla m_k(x_k)\| \le \frac{\varepsilon_g}{2}$:

اگر m_k در $B_k^g = \left\{ x \in \mathbb{R}^n : \|x_k - x\| \le \frac{\varepsilon_g}{2\kappa_g} \right\}$ خطی کامل بود، الگوریتم را به پایان برسانید.

در غیر این صورت، m_kای که در B_k^g خطی کامل باشد را به دست آورده و قرار دهید $\Delta_k = \frac{\varepsilon_g}{2\kappa_g}$.

گام ۶: به منظور به دست آوردن s_k زیرمساله ناحیه اطمینان را حل کنید طوری که شرط کاهش کافی برقرار باشد. $f(x_k + s_k)$ را محاسبه کنید.

گام ۷: طبق الگوی زیر پارامترهای ناحیه اطمینان را به روز رسانی کنید.

$$x_{k+1} = \begin{cases} x_k + s_k, & \rho_k \ge \eta_1 \\ x_k + s_k, & \rho_k \ge \eta_0 \text{ and } m_k \text{ is fully linear on } B_k \\ x_k, & \text{else} \end{cases}$$

$$\Delta_{k+1} = \begin{cases} \Delta_k, & \rho_k \le \eta_1 \text{ and } m_k \text{ is not fully linear on } B_k \\ \min\{\gamma_1 \Delta_k, \Delta_{\max}\}, & \rho_k \ge \eta_1 \\ \gamma_0 \Delta_k, & \rho_k < \eta_1 \text{ and } m_k \text{ is fully linear on } B_k \end{cases}$$

گام ۸: اگر $\rho_k \ge \eta_0$ و m_k روی B_k خطی کامل نباشد، تابع را در یک نقطه بهبود-مدل برآورد کنید.

در هر تکرار الگوریتم ORBIT ممکن است یک سری نقاط درونیابی جدید به مجموعه نقاط اضافه شود. این امر در الگوریتم ORBIT بدون توجه به مقدار تابع در این نقاط صورت می‌گیرد، در حالی که ممکن است مقدار تابع در یکی از این نقاط جدید کمتر از مقدار تابع در مرکز ناحیه اطمینان باشد. بنابر این بهتر است که مجموعه نقاط را قبل از ورود به تکرار بعد، بر اساس مقادیر تابع مرتب کنیم.

در الگوریتم جدیدSORT-ORBIT دو هدف اساسی دنبال می‌گردد، اولاً با مرتب کردن نقاط درونیاب در گام ۱ مدل تقریبی به وسیله نقاط بهتری (از نظر مقادیر تابع) تولید خواهد شد و ثانیا بهترین نقطه ممکن را به عنوان مرکز ناحیه اطمینان جایگزین خواهیم کرد. به این ترتیب امکان انتخاب مدل‌های جایگزین مناسب‌تری فراهم خواهد شد که این امر تأثیر قابل توجهی بر کارایی الگوریتم خواهد داشت.

در گام ۲، الگوریتم به منظور ارائه یک مدل درونیابی خطی کامل در هر تکرار، $n+1$ نقطه مستقل آفینی پیدا می‌کند. استراتژی که این الگوریتم جهت بررسی شرط مستقل آفینی بودن نقاط در پیش می‌گیرد، استفاده از فرایند تجزیه QR است. در واقع الگوریتم با توجه به خاصیت توابع پایه شعاعی، ماتریس ضرایب در دستگاه معادلات (۶) را تجزیه کرده و سعی بر تولید نقاط مستقل آفینی می‌نماید، این فرایند در زیربرنامه‌ای تحت عنوان AffPoints صورت می‌گیرد [۷]. اگر الگوریتم در یک تکرار موفق به تولید این $n+1$ مستقل آفینی نشد، شعاع ناحیه اطمینان را دو برابر می‌کند و دوباره سعی می‌کند که این نقاط مستقل آفینی را در ناحیه اطمینان جدید تولید نماید. اگر باز هم نتوانست موفق به این کار شود باقیمانده نقاط را از فضای متعامد انتخاب خواهد کرد. چرا که این نقاط از فضای متعامد نقاط قبلی انتخاب می‌شوند و لذا شرط مستقل آفینی را بر هم نخواهند زد.

اگر بخواهیم از مزایای مدل‌های درونیابی غیرخطی توسط توابع پایه شعاعی بهره‌مند شویم واضح است که بایستی به فکر اضافه کردن نقاط اضافی به مجموعه $n+1$ نقطه درونیاب خود باشیم. اضافه کردن این نقاط اضافی تضمین خواهد کرد که پارامترهای مدل و مشتقات اول و دوم مدل کراندار بمانند. در گام۳، الگوریتم نقاط اضافی را تا زمانی به مجموعه نقاط درونیابی اضافه می‌کند که تعداد نقاط درونیابی به عدد $p_{\max}$ برسد. الگوریتم جهت اضافه کردن این نقاط اضافی از تکنیکی که در[۴] برای افزودن نقاط، توصیه شده است استفاده می‌کند. البته توجه داشته باشید که افزودن این نقاط اضافی تاثیری روی شرط خطی کامل بودن مدل درونیابی نخواهد داشت. در گام ۸، اگر مدل به دست آمده خطی کامل نباشد به منظور تضمین اینکه مدل یک قدم به خطی کامل بودن نزدیک‌تر شود تابع در یک نقطه اضافی محاسبه می‌شود. به منظور توضیحات کامل‌تر گام‌های الگوریتم می‌توانید به الگوریتم ORBIT در [۲۰] مراجعه کنید.

در ادامه می‌خواهیم راجع به همگرایی الگوریتم جدید صحبت کنیم. می‌دانیم که همگرایی الگوریتم ORBIT در قضیه ۳-۲ در منبع [۲۱] مورد بررسی قرار گرفته است. این قضیه بر پایه یک سری لم‌های میانی در منبع [۷] استوار است. با مرتب کردن نقاط درونیاب و تغییر مرکز ناحیه اطمینان هیچ خللی به اثبات لم‌ها وارد نخواهد شد، بنابراین به راحتی می‌توانیم اثبات همگرایی الگوریتم SORT-ORBIT را به همگرایی الگوریتم ORBIT ارجاع دهیم.

نتایج عددی

در این قسمت به مقایسه عملکرد الگوریتم SORT-ORBIT با الگوریتمORBIT خواهیم پرداخت. به همین منظور در ابتدا مسأله مینیمم‌سازی نامقید خاص زیر را در نظر می‌گیرید.

$$f(x) = \sum_{i=1}^{n/2} [(x_{2i-1} - 2)^2 + (x_{2i-1} - 2)^2 x_{2i}^2 + (x_{2i} + 1)^2].$$

مقدار مینیمم این تابع 0 و نقطه شروع استاندارد برای این مسأله نقطه $x_0 = [1, 1, \ldots, 1]$ است.

در این مسأله قرار می‌دهیم $n = 14$. در جدول ۲ تعداد فراخوانی‌ها، تعداد تکرارها و زمان اجرای دو الگوریتم برای حل این مسأله دیده می‌شود.

جدول۲ .مقایسه دو الگوریتم در حل مساله

نام الگوریتم	تعداد فراخوانی	تعداد تکرار	زمان اجرا
ORBIT	145	65	8
SORT-ORBIT	130	60	5

همان‌طور که در جدول فوق دیده می‌شود الگوریتمSORT-ORBIT دارای موفقیت بیشتری است.

در ادامه به منظور مقایسه دقیق‌تر دو الگوریتم از ۱۰۰ مسأله مینیمم‌سازی نامقید به همراه نقاط شروع استاندارد در مرجع [۱] استفاده خواهیم کرد و تعداد فراخوانی‌های توابع، تعداد تکرارهای دو الگوریتم در حل هر مسأله و همچنین زمان اجرای الگوریتم‌ها را در حل هر مساله با یکدیگر مقایسه خواهیم کرد. در جدول ۳ شما می‌توانید این مسائل به همراه ابعاد آنها را ملاحظه کنید. لازم به ذکر است که کدهای الگوریتمORBIT در آدرس اینترنتی www.mcs.anl.gov/~wild/orbit قابل دسترسی است. ضمنا از تابع پایه شعاعی مکعبی و نرم-دو در پیاده‌سازی الگوریتم استفاده کرده‌ایم. به علاوه حداکثر تعداد نقاط درونیاب در هر تکرار $2n+1$ است. همچنین از پارامترهای زیر در به کارگیری الگوریتم استفاده شده است:

$$\Delta_{max} = 50,\ \epsilon_g = 10^{-5},\ \eta_0 = 0,\ \eta_1 = 0.2,\ \gamma_0 = 0.5,\ \gamma_1 = 2.$$

در ضمن یک الگوریتم را در حل یک مساله بهینه‌سازی زمانی شکست خورده اعلام خواهیم کرد که تعداد تکرارهای آن بیشتر از ۶۰۰ شود. ما برای حل زیرمساله ناحیه اطمینان از روشی به اسم Backtracking procedure که در منبع [۲] به آن اشاره شده است، استفاده می‌کنیم. بقیهٔ پارامترهای لازم در الگوریتم جدید را همانند الگوریتم ORBIT در نظر می‌گیریم.

لازم به ذکر است که مسائل بهینه‌سازی روی یک لپ تاپ SAMSUNG با مشخصات Intel Core i5, 2.60GHz, 4GRAM تحت ویندوز ۷ و با نرم افزار MATLAB 8.2 مورد بررسی قرار گرفته است.

جدول۳. مسائل بهینه‌سازی مربوط به شاخص کارایی

نام مساله	بعد	نام مساله	بعد
ALMOST PQ	16	EXTENDED DF	8
ARGLINB	12	EXTENDED F-R	16
ARGLINC	18	EXTENDED HIM	18

ARWHEAD	15	EXTENDED MAR	14
BDEXP	20	EXTENDED PENALTY	12
BDQRTIC	16	EXTENDED QE1	8
BG2	18	EXTENDED QP1	13
BIGGSB1	15	EXTENDED QP2	9
BROYDEN T	11	EXTENDED T1	14
COSINE	10	EXTENDED T2	12
CRAGGLVY	8	EXTENDED TET	12
CUBE	17	FLETCHCR	15
CURLY20	21	FLETCBV3	10
DIAGONAL1	4	FULL HESSIAN1	16
DIAGONAL2	20	FULL HESSIAN3	19
DIAGONAL3	11	GENERALIZED PSC1	16
DIAGONAL4	18	GENERALIZED Q	6
DIAGONAL5	10	GENERALIZED R	21
DIAGONAL6	15	GENERALIZED T1	15
DIAGONAL7	20	GENERALIZED T2	11
DIAGONAL8	21	GENERALIZED WHITE	10
DIAGONAL9	21	GENHUMPS	12
DIXON3DQ	8	HAGER	16
DIXMAANA	12	HARKERP2	17
DIXMAANB	15	HIMMEL BG	12
DIXMAANC	9	HIMMEL H	16
DIXMAAND	6	INDEF	4
DIXMAANE	18	MCCORMCK	14
DIXMAANF	21	NONDIA	13
DIXMAANG	12	NONDQUAR	11
DIXMAANH	9	NONSCOMP	11
DIXMAANI	9	LIARWHD1	10
DIXMAANJ	15	LIARWHD2	10
DIXMAANK	18	PARTIAL PQ	14
DIXMAANL	18	PERTURBED Q	14
DQDRTIC	19	PERTURBED QD	17
EDENSCH	14	PERTURBED TQ	18
ENGVAL1	8	POWER	9
EXPLIN1	11	QUARTC	12
EXPLIN2	13	QUADRATIC QF1	12
EXTENDED BEALE	6	QUADRATIC QF2	9
EXTENDED BD1	20	RAYDAN1	10
EXTENDED BD2	16	RAYDAN2	14
EXTENDED HIEBERD	18	SINE	17
EXTENDED WHITE	20	SINNCOS	20
EXTENDED WOOD	12	SINQUAD	15
EXTENDED PSC1	10	STAIRCASE1	12
EXTENDED POWELL	16	STAIRCASE2	14
EXTENDED CLIFF	14	TRIDIA	10
EXTENDED DB	14	VARDIM	4

به منظور مقایسه دقیق‌تر الگوریتم‌ها، از شاخص کارایی دولان- موری [8] استفاده کرده‌ایم. در این تکنیک ارزش‌یابی یک آیتم برای مقایسه بین الگوریتم‌ها انتخاب شده و رفتار الگوریتم‌ها را نسبت به این آیتم بررسی و مقایسه می‌کنیم، سپس نتایج را به شکل یک نمودار شاخص کارایی در اختیار خواهیم داشت. اگر P مجموعه مسائل مورد آزمایش و S مجموعه الگوریتم‌های مورد مقایسه باشند، شاخص کارایی یک الگوریتم $s \in S$ به صورت زیر تعریف می‌شود:

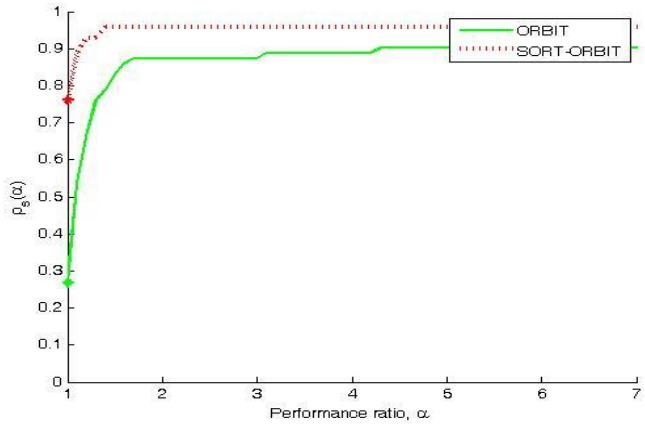
$$\rho_S(\alpha) = \frac{1}{|P|} size\left\{p \in P : r_{p,s} \le \alpha\right\} \qquad (۱۳)$$

به طوریکه $|P|$ تعداد اعضای P است و $r_{p,s}$ به صورت زیر تعریف می‌شود:

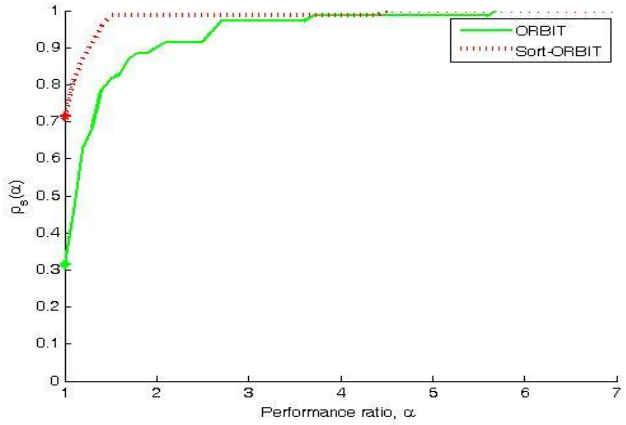
$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}} \qquad (۱۴)$$

$t_{p,s} > 0$ یک اندازه کارایی است که به ازای $p \in P$ و $s \in S$ حاصل می‌گردد. این اندازه می‌تواند شامل آیتم‌هایی مانند تعداد فراخوانی تابع و تعداد تکرارهای لازم برای حل مساله باشد. $\rho_S(1)$ به درصد مسائلی که توسط بهترین الگوریتم $s \in S$ حل می‌شوند اشاره می‌کند و به ازای α به اندازه کافی بزرگ، $\rho_S(\alpha)$ به درصدی از مسائل اشاره می‌کند که توسط $s \in S$ با موفقیت حل می‌شوند. بنابراین میتوان نتیجه گرفت که شاخص کارایی، توزیع احتمال برای نرخ کارایی $r_{p,s}$ است و به طور کلی الگوریتم‌های با مقادیر بزرگتر $\rho_S(\alpha)$ موفق‌تر هستند.

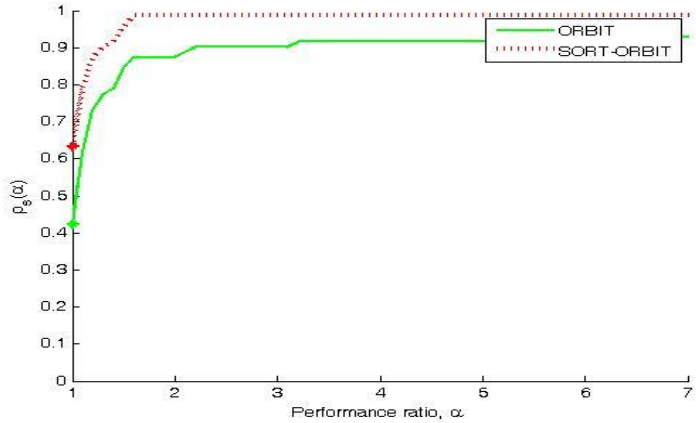
با قرار دادن $\alpha \in [1,7]$ شکل‌های ۱ و۲ و۳ به ترتیب نمودارهای شاخص کارایی برای تعداد فراخوانی توابع، تعداد تکرارهای لازم در حل مسائل و زمان لازم برای اجرای الگوریتم‌ها را نشان می‌دهند. در شکل ۱ می‌بینیم که الگوریتم SORT-ORBIT دارای بیشترین کارایی است به طوری که توانسته است بیش از ۷۵ درصد مسائل را با موفقیت حل کند. به علاوه الگوریتم SORT-ORBIT می‌تواند به ازای $\alpha \geq 2$ صد در صد این دسته مسائل را حل کند. همچنین برتری الگوریتم SORT-ORBIT در مقایسه با الگوریتم ORBIT در شکل‌های ۲ و۳ مشهود است. جایی‌که الگوریتم‌ها از نظر تعداد تکرارهای لازم برای حل مسائل و زمان اجرای آنها مورد مقایسه قرار گرفته‌اند.



شکل۱. نمودار شاخص کارایی برای تعداد فراخوانی توابع



شکل۲. نمودار شاخص کارایی برای تعداد تکرارهای الگوریتم



شکل۳. نمودار شاخص کارایی برای زمان اجرای الگوریتم

از سوی دیگر الگوریتمORBIT از یک آزمون همگرایی وابسته به مشتق به صورت زیر جهت توقف الگوریتم استفاده می‌کند:

$$\|\nabla f(x)\| \le \tau \qquad (۱۵)$$

که در آن τیک ثابت دلخواه مثبت است. به علاوه محدودیت مشخصی روی تعداد فراخوانی تابع در آزمون همگرایی الگوریتم در نظر گرفته شده است.

در چنین الگوریتم‌هایی با تغییر τ کارایی الگوریتم چندان تغییر نمی‌کند. به همین خاطر چنین الگوریتم‌هایی ممکن است هرگز موفق به تولید یک جواب با دقت بالا برای برخی از مسائل بهینه سازی نگردند. به منظور دستیابی به طیفی از جواب‌ها با دقت‌های مختلف، آزمون همگرایی بدون مشتق زیر را روی الگوریتم ORBIT اعمال می‌کنیم:

$$f(x_0) - f(x) \ge (1-\tau)(f(x_0) - f_L) \qquad (۱۶)$$

به طوری که τ یک متغیر و x_0 نقطه شروع و همچنین f_L یک تقریب دقیق از تابع f در نقطه مینیمم سراسری تابع است. آزمون همگرایی فوق اولین بار توسط الستر[1] و نیومایر[2] استفاده ش[۹]. متغیر $\tau \in [0,1]$ به میزان کاهش از مقدار اولیه $f(x_0)$ اشاره می‌کند. با کاهش τ مقدار به دست آمده برای $f(x)$ به f_L نزدیک‌تر می‌شود و بنابراین می‌توانیم امیدوار به یافتن یک جواب با دقت بالاتر باشیم. در ادامه قصد داریم به مقایسه الگوریتم‌های ORBIT و SORT-ORBIT با اعمال آزمون همگرایی جدید بپردازیم. به این منظور مجدداً ۱۶ مسأله بهینه‌سازی نامقید غیرخطی که تقریب دقیقی از مینیمم سراسری آنها در دسترس است را از منبع [۱] انتخاب کرده و تعداد فراخوانی توابع به دست آمده توسط الگوریتم‌ها را محاسبه می‌کنیم. در جدول ۴ شما می‌توانید این مسائل به همراه ابعاد و f_L مربوطه را ملاحظه کنید.

جدول۴. توابع بهینه‌سازی مربوط به شاخص داده

نام مساله	بعد	f_L
BEALE	2	0
BIGGS EXP6	6	5.66E-03
BOX THREE-DIMENSIONAL	3	0
BROWN BADLY SCALED	2	0
BROWN and DENNIS	4	85822.2
EXTENDED ROSENBROCK	8	0
GAUSSIAN	3	1.13E-08
GULF RESEARCH and DEVELOPMENT	3	0
HELICAL VALLEY	3	0
POWELL BADLY SCALED	2	0
PENALTY I	10	7.09E-05
PENALTY II	10	2.94E-04
TRIGONOMETRIC	15	0
VARIABLY DIMENSIONED	5	0
WATSON	20	2.49E-20
WOOD	14	0

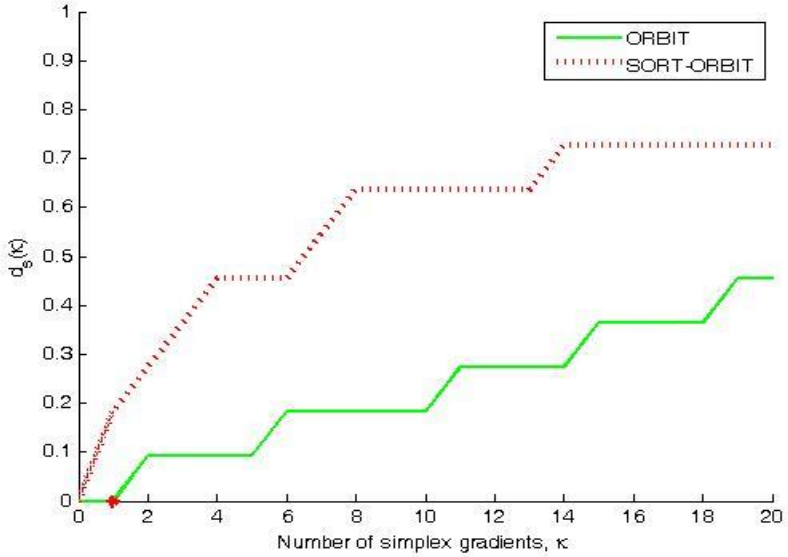
این بار به منظور مقایسه الگوریتم‌ها از شاخص داده موری- ویلد [۱۳] به صورت زیر:

$$d_s(\kappa) = \frac{1}{|P|} size\left\{ p \in P : \frac{t_{p,s}}{n_p + 1} \le \kappa \right\} \qquad (۱۷)$$

استفاده می‌کنیم. به طوری که برای $p \in P$ و $s \in S$ ، $t_{p,s}$ تعداد فراخوانی لازم برای برقراری رابطهٔ (۱۶) به ازای τ داده شده است و n_p نیز تعداد متغیرهای مسأله p است. در این تعریف می‌توانیم κ را به عنوان سیمپلکس گرادیان تعبیر کنیم. در این صورت $d_s(\kappa)$ به درصد مسائلی که توسط الگوریتم s قابل حل هستند اشاره دارد. با فرض $\kappa \in [0,20]$ نمودارهای شاخص داده را به ازای $\tau = 10^{-3}$ برای دو الگوریتم به دست آورده‌ایم. شکل ۴ را ملاحظه کنید. در این شکل به وضوح می‌بینیم که الگوریتم SORT-ORBIT موفق ترعمل کرده است. به عنوان مثال اگر سیمپلکس گرادیانی برابر با ۱۰ داشته باشیم آنگاه نمودارهای شاخص داده در شکل۴ نشان می‌دهند که الگوریتم

[1] Elster
[2] Neumaier

ORBIT قادر به حل ۲۰ درصد مسائل است. این در حالی است که الگوریتم SORT-ORBIT می‌تواند ۶۵ درصد مسائل را حل کند.



شکل۴. نمودار شاخص داده

نتیجه‌گیری

در این مقاله با ارائه ایده‌های جدید، کارایی الگوریتم ORBIT به عنوان یک الگوریتم ناحیه اطمینان مهم در زمینه بهینه‌سازی توابع نامقید به طور قابل توجهی افزایش یافت. در رویکرد ارائه شده در این مقاله، در ابتدا نحوه مدیریت این الگوریتم بر نقاط درونیاب مورد مطالعه قرار گرفت. بر اساس مطالعه انجام شده دریافتیم که این الگوریتم نقاط درونیاب و مقدار تابع در آن نقاط را در هر تکرار ذخیره کرده و در تکرارهای بعد مورد استفاده قرار می‌دهد. اما این اطلاعات به صورت بهینه مورد استفاده قرار نمی‌گیرند. استراتژیی که ما برای ارتقاء این الگوریتم به کار بردیم به این صورت بود که قبل از ورود به هر تکرار ابتدا مجموعه نقاط درونیاب را بر اساس مقادیر توابع از کوچک به بزرگ مرتب کنیم. با این کار به دو هدف دست می یابیم. اولاً برای ساخت مدل جایگزین، از نقاطی استفاده خواهد شد که مقدار تابع در آن نقاط کمتر است و ثانیا بهترین نقطه (نقطه ای که دارای کمترین مقدار تابع است) را به عنوان مرکز ناحیه اطمینان جایگزین می کنیم. بر اساس نتایج عددی، با استفاده از این رویکرد تعداد دفعات ارزیابی تابع و تعداد تکرار های الگوریتم ORBIT به طور قابل توجهی کاهش می‌یابد.

References

1. Andrei, N., "An unconstrained optimization test functions collection", Adv. Model.Optim, vol. 10, no. 1 (2008) 147-161

2. Armijo, L., "Minimization of functions having Lipschitz continuous first partial derivatives", Pacific J. Math.,vol. 16 (1966) 11-30.

3. Bertsekas, D. P., "Nonlinear Programming", Massachusetts Institute of Technology, Athena Scientific, Belmont, Massachusetts, second edition (1999).

4. Björkman, M., Holmström, K., "Global optimization of costly nonconvex functions using radial basis functions", Optim. Eng., vol. 1 (2000) 373-397.

5. Buhmann, M. D., "Radial Basis Functions", Cambridge University Press New York, NY, USA (2003).

6. Conn, A. R., Scheinberg, K., Toint, Ph. L., "Recent progress in unconstrained nonlinear optimization without derivatives", Math. Program., vol. 79 (1997) 345-397.

7. Conn, A. R., Scheinberg, K.,Vicente, L. N.,"Global convergence of general derivative-free trust-region algorithms to first and second order critical points", SIAM J. Optim., vol. 20 (2009) 387-415.

8. Dolan, E., More', J. J., "Benchmarking optimization software with performance profiles", Math. Program., vol. 91 (2002). 201-213.

9. Elster, C., Neumaier, A., "A grid algorithm for bound constrained optimization of noisy function", IMA J. Numer. Anal., vol. 15 (1995) 585- 608.

10. Fasshauer, G.E., Meshfree Approximation Methods with Matlab, World Scientifc, Singapore, 2007.

11. Günther, C.,Tammer, C., "Relationships between constrained and unconstrained multi-objective optimization and application in location theory", Math. Meth. Oper. Res., vol. 84, no. 2 (2016) 359-387.

12. Gutmann, H. M., "A radial basis function method for global optimization", J. Global. Optim., vol. 19 (2001) 201-227.

13. More', J., Wild, S. M., "Benchmarking derivative-free optimization algorithms", SIAM J. Optim., vol. 20, no. 1 (2009) 172-191.

14. Oeuvray, R., Bierlaire, M., "A new derivative-free algorithm for the medical image registration problem", Int. J. Modelling and Simulation., vol. 27 (2007) 115-124.

15. Powell, M. J. D., "A new algorithm for unconstrained optimization", in: Rosen, J. B., Mangassarian, O.L., Ritter, K., editors, Nonlinear Programming, Academic Press, New York, (1970) 31-66.

16. Powell, M. J. D., "The NEWUOA software for unconstrained optimization without derivatives". In Di Pillo, G., Roma, M., editors, Large-Scale Nonlinear Optimization, Springer, (2006) 255-297.

17. Powell, M. J. D., "UOBYQAunconstrained optimization by quadratic approximation", Math. Program., vol. 92 (2002) 555-582.

18. Regis, R.G., Shoemaker, C. A., "A stochastic radial basis function method for the global optimization of expensive functions", INFORMS J. Comput., vol. 19 (2007) 497-509.

19. Winfield, D., "Function minimization by interpolation in a data table, J. Inst. Math. Appl., vol. 12 (1973) 339-347.

20. Wild, S. M., Regis, R. G., Shoemaker, C. A., "Optimization by radial basis function interpolation in trust-region", SIAM J. Sci Comput., vol. 30, no. 6 (2008) 3197-3219.

21. Wild, S. M., Shoemaker, C. A., "Global convergence of radial basis function trust region derivative-free algorithms". SIAM J. Optim, vol. 21, no. 3 (2011) 761-781.