مجله تحقیق در عملیات در کاربردهای آن، سال هفدهم، شماره اول (پیاپی ۶۴)، بهار ۹۹، ص ص ۱۰۱-۸۵
شاپا ۷۲۸۶-۲۲۵۱

حل مسایل بهینه‌سازی نامقید به وسیله ترکیبی از الگوریتم ناحیه اعتماد نایکنوا و تکنیک فیلتر

زینب سعیدیان طربی[1*]، فرشید ارزانی[2]

۱- استادیار، دانشکده ریاضی، دانشگاه کاشان، اصفهان، ایران

۲- مرکز پژوهشی محاسبات علمی در بهینه‌سازی و مهندسی سامانه‌ها (اسکوپ)، دانشکده ریاضی، دانشگاه صنعتی خواجه‌نصیرالدین طوسی، تهران، ایران

رسید مقاله: ۷ بهمن ۱۳۹۷

پذیرش مقاله: ۳ مهر ۱۳۹۸

چکیده

در این مقاله، یک روش ناحیه اعتماد تطبیقی نایکنوای جدید برای حل مسایل بهینه‌سازی نامقید ارایه می‌کنیم که در آن از تکنیک فیلتر استفاده شده است. در روش پیشنهادی، از تکنیک نایکنوایی خاصی استفاده شده است. با استفاده از این تکنیک، الگوریتم می‌تواند از گام اول، از خواص نایکنوایی بهره ببرد و سرعت رسیدن به جواب را افزایش دهد.

همچنین، فیلتر به کاررفته در این روش، از نوع فیلترهای متناهی است. این مهم، با توجه به ساختار فیلتر اثبات شده است. ساختار الگوریتم، بر اساس دو حلقه خارجی و داخلی استوار است که هر یک با توجه به شرایط موجود در هر تکرار عملیات مشخصی را انجام می‌دهد. در تکرارهای الگوریتم، برای یافتن طول گام، از زیرمساله‌ای با ساختار ساده استفاده می‌کنیم. همچنین، صورت اصلاح شده‌ای از شرط سکانت را برای تقریب هسیان چنان به‌کارمی‌بریم که ماتریس هسیان، همواره معین مثبت باشد. علاوه برآن، همگرایی سراسری الگوریتم تحت برخی شرایط استاندارد اثبات شده است. نتایج عددی حاصل از روش پیشنهادی روی مجموعه مسایل آزمونی، حاکی از کارایی الگوریتم جدید، نسبت به سایر روش‌های مشابه است.

کلمات کلیدی: روش‌های ناحیه اعتماد، روش‌های تطبیقی و نایکنوا، تکنیک فیلتر، همگرایی سراسری.

۱ مقدمه

دستگاه معادلات غیرخطی زیر را در نظر می‌گیریم:

$$f(x) = 0, \qquad (1)$$

که در آن، $f : \mathbb{R}^n \to \mathbb{R}$ ، به طور پیوسته مشتق‌پذیر است. روش‌های بسیاری برای حل این دسته مسایل، پیشنهاد شده‌اند. از آن جمله می‌توان به روش‌های تکراری جستجوی خطی، ناحیه اعتماد و روش‌های نیوتن و شبه نیوتن اشاره کرد [۱، ۲].

* عهده‌دار مکاتبات

آدرس الکترونیکی: saeidian@kashanu.ac.ir

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

در روش‌های جستجوی خطی، همواره در راستای یک جهت (کاهشی) حرکت می‌کنیم تا در تابع هدف، کاهش کافی رخ دهد. در روش‌های ناحیه اعتماد استاندارد، جهت جستجو از مینیمم‌سازی یک مدل درجه دوم از تابع هدف، در نقطه کنونی حول ناحیه‌ای اطراف نقطه که به ناحیه اعتماد معروف است، به دست می‌آید. سپس، بر اساس نسبت ناحیه اعتماد، گام آزمایشی پذیرفته یا رد می‌شود. همچنین، شعاع ناحیه اعتماد، بر اساس نسبت r_k افزایش یا کاهش می‌یابد یا بدون تغییر می‌ماند. برای بهنگام‌سازی شعاع ناحیه اعتماد و نیز انتخاب شعاع اولیه روش‌های مختلفی تاکنون ارایه شده است [۳، ۴]. در روش‌های استاندارد، در هر تکرار شعاع ناحیه اعتماد، مستقل از اطلاعات تابع گرادیان و ماتریس هسیان بهنگام می‌شود. این امر موجب حساسیت مساله به انتخاب شعاع اولیه می‌شود. برای رفع این مشکل، روش‌های ناحیه اعتماد تطبیقی معرفی شدند [۵، ۶]. با استفاده از این دسته روش‌ها، علاوه بر کاهش تعداد تکرارهای حل مساله، از اطلاعات تابع گرادیان و ماتریس هسیان نیز برای بهنگام کردن شعاع بهره می‌گیریم. اغلب، روش‌های ناحیه اعتماد در نقاطی که روش نیوتن عملکرد ضعیفی دارد، دارای عملکرد مناسبی می‌باشند.

در [۴] یک روش ناحیه اعتماد تطبیقی نایکنوا بر اساس یک زیرمساله‌ی ساده برای مسایل بزرگ مقیاس، ارایه شد که از اطلاعات دو تکرار آخر به‌صورت کامل استفاده می‌کند. ایده‌ی زیرمساله‌ی ساده برای آن است که حل زیرمساله‌ی ناحیه اعتماد وقتی ماتریس هسیان، ماتریس چگال بزرگ مقیاس باشد، هزینه‌بر است.

در روش‌های جستجوی خطی، در تکرار k ام نقطه جدید به صورت زیر به‌دست می‌آید:

$$x_{k+1} = x_k + \lambda_k d_k, \qquad (۲)$$

که در آن d_k یک جهت کاهشی و λ_k طول گامی است که از روش‌های جستجوی خطی دقیق یا غیردقیق به‌دست می‌آید. یکی از معروف‌ترین و کارآمدترین طول گام‌های ارایه شده، طول گام برزیلای-بروین است [۶].

فرض کنید

$$g_k = g(x_k) = \nabla f(x_k),\ s_k = x_k - x_{k-1},\ y_k = g_k - g_{k-1}. \qquad (۳)$$

نقطه جدید با استفاده از رابطه برزیلای-بروین به صورت $x_{k+1} = x_k + d_k$ محاسبه می‌شود که در آن

$$d_k = -\frac{1}{\alpha_k} g_k, \qquad (۴)$$

و α_k با یکی از دو رابطه زیر محاسبه می‌شود:

$$\alpha_k = \frac{s_k^T y_k}{s_k^T s_k}, \alpha_k = \frac{y_k^T y_k}{s_k^T y_k}. \qquad (۵)$$

برزیلای و بروین نشان دادند که این طول گام نسبت به سایر روش‌های جستجوی خطی به محاسبات کمتری نیاز دارد و برای مسایل درجه دوم محدب دو بعدی دارای همگرایی زبر خطی است [۷].

از آن‌جا که روش‌های نایکنوا برای حل مسایل بهینه‌سازی سرعت همگرایی بالاتری نسبت به روش‌های یکنوا دارند، روش نایکنوا را برای حل انتخاب می‌کنیم.

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

روش نایکنوای به‌کار رفته در این مقاله، از آزادسازی روش نایکنوای آهوخوش و امینی [۸] و بهنگام‌سازی شعاع ناحیه اعتماد به کمک روش تطبیقی شی و گو [۹] به‌دست می‌آید. عبارت نایکنوای جدید به صورت $(۱+\phi_k)R_k$ در نظر گرفته می‌شود [۱۰] که در آن داریم:

$$R_k = \epsilon_k f_{\ell(k)} + (۱-\epsilon_k) f_k, \qquad (۶)$$

و $f_k = f(x_k)$ و $\epsilon_k \in [\epsilon_{\min}, \epsilon_{\max}] \subset [۰,۱]$

$$f_{\ell(k)} = \max_{۰\le j\le \min(k,M)} f_{k-j}, \qquad (۷)$$

همچنین،

$$\phi_k = \begin{cases} \eta_k, if\ R_k > ۰, \\ ۰, \quad if\ R_k \le ۰, \end{cases}. \qquad (۸)$$

$\{\eta_k\}$ یک دنباله مثبت است که در رابطه زیر صدق می‌کند:

$$\sum_{k=۱}^{\infty} \eta_k \le \eta < \infty. \qquad (۹)$$

اخیرا، ژو و همکاران، بر اساس اصلاح شرط سکانت وی [۱۱] یک زیرمساله‌ی ساده و یک روش ناحیه اعتماد تطبیقی نایکنوا پیشنهاد دادند [۶]. بعدها، بیگلری و سلیمان‌پور در [۱۲] زیرمساله‌ی ساده‌ی دیگری را با مزایای بیشتر نسبت به [۶] ارایه کردند که در آن، تقریب هسیان در نقطه‌ی کنونی از رابطه زیر محاسبه می‌شود:

$$\hat{\gamma}_k := \gamma(x_k) = \frac{۴(f_{k-۱} - f_k) + ۳g_k^T d_{k-۱} + g_{k-۱}^T d_{k-۱}}{d_{k-۱}^T d_{k-۱}}. \qquad (۱۰)$$

سپس سعیدیان و پیغامی [۱۳]، بر اساس صورت اصلاح شده شرط سکانت [۷] برای تقریب هسیان در نقطه کنونی روش جدیدی ارایه دادند که نسبت به سایر روش‌های پیشین دارای برتری بود. در روش پیشنهادی، از عبارت نایکنوای [۸] و نسخه‌ی اصلاح‌شده‌ی روش تطبیقی [۱۴] در ساختار الگوریتم پیشنهادی استفاده شده است.

تکنیک فیلتر اولین بار توسط فلچر و لیفر در [۱۵] ارایه شد. این ایده بر اساس بهینه‌سازی توابع چندهدفه شکل گرفته است. بر اساس آن، هر نقطه در هر تکرار در مجموعه فیلتر پذیرفته یا رد می‌شود. یک نقطه در فیلتر قابل قبول است، اگر توسط هیچ یک از نقاط داخل فیلتر مغلوب نشود. هنگامی که یک نقطه در فیلتر پذیرفته شد، کلیه نقاطی که توسط نقطه جدید مغلوب می‌شوند از فیلتر حذف می‌شوند.

فاطمی و مهدوی امیری در [۱۶] فیلتری را معرفی کردند که با داشتن شرط پذیرش خاص می‌تواند اندازه فیلتر را کنترل کند. بر اساس فیلتر آن‌ها نقطه x_k نسبت به x_l در فیلتر پذیرفته می‌شود، هرگاه $j \in \{۱,...,n\}$ وجود داشته باشد به طوری که

$$\left|g_j(x_k)\right|^{\rho_۲} + \tau_۲ \left\|g(x_k)\right\|^{\rho_۱} \leqslant \left|g_j(x_l)\right|^{\rho_۲} + \tau_۱ \left\|g(x_l)\right\|^{\rho_۱}, \qquad (۱۱)$$

که در آن $\rho_۱$ و $\rho_۲$ ثابت‌های مثبت هستند و $\tau_۱$ و $\tau_۲$ در رابطه زیر صدق می‌کنند:

$$۰ \leqslant \tau_۱ < \tau_۲ < \frac{۱}{\sqrt{n}}. \qquad (۱۲)$$

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

اگر $g_i(x)$ مولفه iام بردار گرادیان $g(x)$ باشد، آنگاه نقطه x_k در فیلتر $\mathcal{F}$ پذیرفته می‌شود، هرگاه x_k نسبت به هر x_l، با $g(x_l)\in\mathcal{F}$ در فیلتر پذیرفته شود. درنتیجه $g(x_k)$ به فیلتر افزوده می‌شود و هر $g(x_l)\in\mathcal{F}$ با شرط

$$\left|g_j(x_k)\right|^{\rho_2}+\theta_2\left\|g(x_k)\right\|^{\rho_1}\leqslant\left|g_j(x_l)\right|^{\rho_2}+\tau_1\left\|g(x_l)\right\|^{\rho_1},\forall j\in\{1,\ldots,n\}, \qquad (13)$$

از فیلتر حذف می‌شود.

ارزانی و پیغامی در [۱۷]، روشی را برای حل مسایل بهینه‌سازی نامقید مطرح کردند که علاوه بر خواص یکنوایی خاص از تکنیک فیلتر نیز بهره می‌برد. ایشان نشان دادند که با ترکیب مفهوم فیلتر با یک تکنیک یکنوایی خاص (که به الگوریتم اجازه می‌داد از اولین تکرار بتواند از خاصیت یکنوایی بهره ببرد)، می‌توان نتایج بهتری در رسیدن به جواب بهینه برای مسایل بهینه‌سازی نامقید به‌دست آورد. در واقع نسخه‌ای از روش معروف برزیلای-بروین را ارایه دادند که در آن، با استفاده از تکنیک فیلتر قدرت الگوریتم افزایش یافته و می‌توان برای مسایل با اندازه بزرگ نیز از این روش استفاده کرد. همچنین، ایشان در [۱۸] از همین تکنیک برای حل دستگاه معادلات غیرخطی استفاده کرده‌اند که در آن تکنیک فیلتر به یک روش نایکنوای مشتق آزاد افزوده شده بود. تکنیک نایکنوای استفاده شده در این مقاله نیز، همان تکنیک استفاده شده در مقاله [۱۸] می‌باشد.

هدف ما در این مقاله، ارایه یک روش ناحیه اعتماد تطبیقی نایکنوای جدید است. این روش، به منظور حل مسایل بهینه‌سازی نامقید، بر استفاده از تکنیک فیلتر، استوار است.

در ادامه، در بخش دوم این مقاله به معرفی الگوریتم جدید پیشنهادی خواهیم پرداخت. سپس، در بخش سوم به بررسی همگرایی سراسری الگوریتم می‌پردازیم. در ادامه، در بخش چهارم نتایج عددی آورده شده و در بخش پنجم نتیجه‌گیری کلی بیان می‌گردد.

۲ ساختار الگوریتم جدید

ساختار الگوریتم پیشنهادی از دو حلقه خارجی و داخلی تشکیل می‌شود. حلقه داخلی برای یافتن طول گام d_k از روش ناحیه اعتماد پیشنهادی [۱۳] استفاده می‌کند.

برای نقطه‌ی x_k داده شده، گام آزمایشی d_k از حل (تقریبی) زیرمساله‌ی زیر به‌دست می‌آید:

$$\min m_k(d)=g_k^Td+\frac{1}{2}d^T\gamma(x_k)d$$

$$s.t.\ \ \|d\|\leq\Delta_k, \qquad (14)$$

که در آن $\gamma_k:=\gamma(x_k)$، تقریب اسکالر ماتریس هسیان است. تقریب هسیان را به صورت زیر در نظر می‌گیریم:

$$\gamma_k=\frac{4(f_{k-1}-f_k)+(3+\eta_k)g_k^Td_{k-1}+g_{k-1}^Td_{k-1}}{d_{k-1}^Td_{k-1}}, \qquad (15)$$

که در آن

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

$$\eta_k = \begin{cases} \frac{۴(f_k - f_{k-۱}) - ۳g_k^T d_{k-۱} - g_{k-۱}^T d_{k-۱} + \delta}{g_k{}^T d_{k-۱}}, & \hat{\gamma}_k < ۰; \\ ۰, & o.w. \end{cases}.$$

δ یک عدد مثبت کوچک و $\hat{\gamma}_k$ از رابطه‌ی زیر محاسبه می‌شود:

$$\hat{\gamma}_k := \gamma(x_k) = \frac{۴(f_{k-۱} - f_k) + ۳g_k^T d_{k-۱} + g_{k-۱}^T d_{k-۱}}{d_{k-۱}^T d_{k-۱}}. \quad (۱۶)$$

با این تعریف، واضح است که همواره $\gamma_k > ۰$ است.

با استفاده از d_k، نسبت نایکنوا را به صورت زیر داریم:

$$r_k = \frac{R_k - f(x_k + d_k)}{Pred_k}, \quad (۱۷)$$

که در آن R_k از رابطه‌ی (۶) محاسبه می‌شود و $Pred_k = m_k(۰) - m_k(d_k)$. حال، برای $\mu \in (۰,۱)$ داده شده، اگر $r_k \geq \mu$، آن‌گاه گام آزمایشی پذیرفته می‌شود. در غیر این‌صورت، گام آزمایشی رد می‌شود.

در هر دو حالت، شعاع بر اساس رابطه‌ی زیر بهنگام می‌شود:

$$\Delta_k = \min\left\{\nu_k \frac{\| g_k \|}{\gamma_k}, \Delta_{\max}\right\},$$

که در آن، $\Delta_{\max} > ۰$ کرانی برای شعاع‌ها و $\nu_{k+۱}$ از رابطه‌ی زیر بهنگام می‌شود:

$$\nu_{k+۱} = \begin{cases} \sigma_۰ \nu_k, & r_k < \mu_۱, \\ \nu_k, & \mu_۱ \leq r_k \leq \mu_۲, \\ \min\{\sigma_۱ \nu_k, \nu_{\max}\}, & r_k > \mu_۲, \end{cases} \quad (۱۸)$$

و $۰ < \mu_۱ < \mu_۲ \leq ۱$ ،$۰ < \sigma_۰ < ۱ < \sigma_۱$ ،$\nu_{max} > ۰$ و $۰ < \sigma_۰ < ۱$ ثابت‌های معلوم هستند. حال، اگر $r_k \geq \mu$، نقطه جدید از رابطه‌ی $x_{k+۱} = x_k + d_k$ محاسبه می‌شود. در غیر این‌صورت، قرار می‌دهیم $x_{k+۱} = x_k$.

بنابراین الگوریتم حلقه داخلی به شرح زیر است:

الگوریتم ۲-۱: الگوریتم حلقه داخلی

ورودی‌ها: $x_۰ \in \mathbb{R}^n, ۰ < \mu < \mu_۱ < \mu_۲ \leq ۱, ۰ < \sigma_۰ < ۱ < \sigma_۱, \epsilon, \varepsilon, M, \nu_{\max}, \Delta_{\max} > ۰, ۰ < \theta_۱ < \theta_۲, \delta > ۰$.

گام ۰: قرار ده $k = ۰$، $\nu_۰ = ۱, g_۰ = g(x_۰), \gamma_۰ := \gamma(x_۰) = ۱$ و $\Delta_۰ = \min\left\{\nu_۰ \frac{\| g_۰ \|}{\gamma_۰}, \Delta_{\max}\right\}$.

گام ۱: اگر $\| g_k \| \leq \varepsilon$، آن‌گاه متوقف شو.

گام ۲: d_k را از حل (۱۴) و r_k را از (۱۷) محاسبه کن.

گام ۳: اگر $r_k < \mu$، آن‌گاه قرار ده $\Delta_k = \sigma_۰ \Delta_k$ و به گام ۲ برو.

گام ٤: قرار ده $x_{k+۱} = x_k + d_k$.

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

گام ۵: γ_{k+1} را از رابطه‌ی (۱۵) محاسبه کن.

اگر $\gamma_{k+1} \le \epsilon$ آن‌گاه قرار ده $\gamma_{k+1} = \theta_1$. اگر $\gamma_{k+1} \ge \frac{1}{\epsilon}$ آن‌گاه قرار ده $\gamma_{k+1} = \theta_2$.

گام ۶: ν_{k+1} را از (۱۸) بهنگام کن و قرار ده $\Delta_{k+1} = \min\left\{\nu_{k+1}\frac{\| g_{k+1}\|}{\gamma_{k+1}}, \Delta_{\max}\right\}$. قرار ده $k =: k+1$ و به گام ۱ برو.

در ادامه یک الگوریتم ناحیه اعتماد نایکنوای تطبیقی که با استفاده از تکنیک فیلتر پیاده‌سازی شده است را معرفی می‌کنیم. در این الگوریتم، از الگوریتم حلقه داخلی، برای به‌دست آوردن دنباله نقاط برای هر تکرار از الگوریتم اصلی استفاده می‌کنیم. به عبارت دیگر، به ازای هر تکرار از الگوریتم اصلی یک بار الگوریتم حلقه داخلی از ابتدا تا انتها اجرا می‌شود تا دنباله مورد نظر در آن تکرار را تولید کند.

ملاحظه ۲.۱. در حلقه‌ی خارجی الگوریتم که آن را با نام الگوریتم اصلی در نظر می‌گیریم، در گام kام، جهت کاهشی d_k با شرایط زیر محاسبه می‌شود:

$$\|d_k\| \le c_2\|g_k\|, \quad -\frac{d_k^T g_k}{\|g_k\|} \ge c_1\|g_k\|, \tag{۱۹}$$

که در آن c_1 و c_2 ثابت‌های مثبت هستند. به سادگی می‌توان دید که جهت برزیلای-بروین (۴) و $d_k = -g_k$ در شرایط (۱۹) صدق می‌کنند. برای N داده شده، نقاط آزمایشی z_k^i، برای $i = 0,...,N$، از روابط زیر محاسبه می‌شوند:

$$\begin{cases} z_k^1 = x_k, \\ p_k^1 = d_k, \\ z_k^{i+1} = z_k^i + p_k^i, \qquad i = 1,\dots,m \end{cases} \tag{۲۰}$$

که در آن $m \le N$ و p_k^i به صورت زیر به‌دست می‌آیند:

$$p_k^i = -\frac{1}{\alpha_k}\nabla f(z_k^i), \qquad i = 1,2,\dots,m, \tag{۲۱}$$

که در آن α_k از (۵) با در نظر گرفتن $s_k = z_k^i - z_k^{i-1}$ و $y_k = \nabla f(z_k^i) - \nabla f(z_k^{i-1})$ محاسبه می‌شود.

در ابتدا هر نقطه به‌دست آمده در شرط فیلتر بررسی می‌شود. اگر نقطه مورد نظر در شرط فیلتر پذیرفته نشود، آن‌گاه شرط نایکنوایی روش بررسی می‌شود. در صورتی که نقطه z_k^i در شرط فیلتر پذیرفته شد و یا این که در شرط نایکنوایی روش صدق کرد، آن‌گاه قرار می‌دهیم $x_{k+1} = z_k^i$. در حالت اول، نقطه جدید به فیلتر اضافه شده و سپس فیلتر را بهنگام می‌کنیم. این فرایند حداکثر N بار تکرار می‌شود. در حالتی که هیچ یک از نقاط آزمایشی در فیلتر پذیرفته نشوند و یا در شرط نایکنوایی روش صدق نکنند، یک روش نایکنوای جستجوی خطی در جهت d_k اجرا می‌شود و نقطه جدید $x_{k+1} = x_k + \lambda_k d_k$ با طول گام $\lambda_k \in (0,1]$ به‌دست می‌آید. به عبارت دیگر در بدترین حالت برای هر تکرار، اگر هیچ یک از نقاط به‌دست آمده در فیلتر پذیرفته نشوند یا در

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

شرط نایکنوایی صدق نکنند، در انتها یک روش جستجوی خطی خواهیم داشت و نقطه جدید از اجرای آن به‌دست خواهد آمد.

ساختار این الگوریتم جدید به شرح زیر است:

الگوریتم ۲-۲ : (الگوریتم اصلی) الگوریتم ناحیه اعتماد نایکنوای تطبیقی با استفاده از روش فیلتر

ورودی‌ها: مقادیر $x_0 \in R^n,\ f_{sup} > -\infty$ ، اعداد صحیح $N \geq 1, M \geq 0, \sigma > 0, \rho_1 > 0, \rho_2 > 0$ و

$$0 < \mu < \mu_1 < \mu_2 \leq 1, 0 < \sigma_0 < 1 < \sigma_1, 0 < \epsilon_{\min} < \epsilon_{\max} < 1, \epsilon, \varepsilon, M, \nu_{\max}, \Delta_{\max} > 0, 0 < \theta_1 < \theta_2, \delta > 0, 0 \leq \tau_1 < \tau_2 < \frac{1}{\sqrt{n}}$$

داده شده‌اند.

قرار ده $\mathcal{F}_0 = \varnothing, k = 0, i = 0$.

گام ۱: اگر $\|g_k\| \leq \epsilon$ آن گاه متوقف شو.

گام۲: اگر $i \leq N$، آن‌گاه با استفاده از الگوریتم حلقه داخلی، نقطه z_k^i را محاسبه کن. در غیر این صورت به گام ۴ برو.

گام۳: اگر z_k^i توسط فیلتر پذیرفته شد و $f(z_k^i) \leq f_{sup}$ ، آن‌گاه قرار ده $x_{k+1} = z_k^i$ و z_k^i را به فیلتر $\mathcal{F}_k$ اضافه کن. سپس فیلتر را به صورت $\mathcal{F}_{k+1}$ بهنگام کن و قرار ده $k = k+1$ و به گام ۱ برو.

در غیر این صورت، اگر

$$f(z_k^i) \leq (1+\phi_k) R_k - \max_{1 \leq h \leq i} \{\sigma \|z_k^h - x_k\|\}, \qquad (۲۲)$$

آن‌گاه قرار ده $x_{k+1} = z_k^i, \mathcal{F}_{k+1} = \mathcal{F}_k, k := k+1$ و به گام ۱ برو.

در غیر این صورت قرار ده $i := i+1$ و به گام ۲ برو.

گام ٤: یک روش جستجوی خطی در جهت d_k اجرا کن و طول گام λ_k را به‌دست آور. قرار ده

$$x_{k+1} = x_k + \lambda_k d_k, k = k+1,$$

و به گام ۱ برو.

---

۳ همگرایی سراسری

ابتدا مجموعه‌های A و D را به صورت زیر تعریف می‌کنیم:

$$A = \{k \mid x_k \in \mathcal{F}_k\}.$$

مجموعه D، مجموعه تمام تکرارهای x_k است که در رابطه (۲۲) صدق می‌کنند و یا از طریق گام ۴ الگوریتم ناحیه اعتماد نایکنوای تطبیقی با استفاده از روش فیلتر به‌دست آمده‌اند.

با توجه به ساختار الگوریتم به سادگی مشخص است که $A \cap D = \varnothing$.

هم‌چنین، فرضیات زیر را روی مساله در نظر می‌گیریم:

$A1$. تابع f روی $\mathbb{R}^n$ دوبار به طور پیوسته مشتق‌پذیر است.

$A2$. برای η داده شده در (۹)، مجموعه تراز $L = \{x \in \mathbb{R}^n \mid f(x) \leq e^{\eta} f(x_0)\}$ بسته و کران‌دار است.

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

$A3$. جملات دنباله $\{x_k\}$ تولید شده از الگوریتم اصلی در مجموعه بسته و کران‌دار $\Omega \subset \mathbb{R}^n$ قرار دارد.

با در نظر گرفتن دو مجموعه‌ی A و D، برای اثبات همگرایی سراسری الگوریتم می‌بایست رفتار نقاط $\{x_k\}$ تولید شده از تکرارهای اصلی الگوریتم و همچنین نقاط $\{z_k^i\}$ تولید شده توسط الگوریتم حلقه داخلی را در نظر گرفت. دنباله $\{x_k\}$ که توسط الگوریتم اصلی تولید شده است را در نظر بگیرید. اگر الگوریتم در تکرار k متوقف شود، در این صورت به یک نقطه ایستا می‌رسیم. در غیراینصورت، الگوریتم یک دنباله نامتناهی از نقاط را تولید می‌کند. در حالت دوم، نشان می‌دهیم که دنباله نامتناهی به‌دست آمده حداقل دارای یک زیردنباله همگرا به نقطه ایستای تابع f می‌باشد. بنابراین، برای عدد اصلی مجموعه‌های A و D دو حالت ممکن داریم: $|A| = \infty$ یا $|D| = \infty$.

در ادامه این دو حالت را مورد بررسی قرار می‌دهیم.

در تکرار kام از الگوریتم، مجموعه فیلتر $\mathcal{F}_k$ و شرط پذیرش آن را به صورت (۱۱) در نظر بگیرید. می‌توان نشان داد که اندازه فیلتر $\mathcal{F}_k$ متناهی است [۱۶]. با این وجود فیلتر می‌تواند تعداد نامتناهی تکرار را بپذیرد و همچنان دارای اندازه متناهی باشد. در لم زیر همگرایی را در حالتی بررسی می‌کنیم که فیلتر تعداد نامتناهی تکرار را در خود می‌پذیرد.

لم ۳.۱. فرضیات $A1-A3$ را در نظر بگیرید. اگر داشته باشیم $|A| = \infty$، آن گاه [۱۷، ۱۳]:

$$\lim_{k \in A} \|g_{k+1}\| = 0. \qquad (۲۳)$$

اثبات. با استفاده از اثبات بیان شده در [۱۶]، به سادگی می‌توان این لم را اثبات کرد.

لم ۳.۲. اگر $|D| = \infty$، آنگاه برای دنباله نامتناهی $\{x_k\}_{k \in D}$، یک نقطه حدی وجود دارد که در شرط لازم مرتبه اول همگرایی صدق می‌کند. به بیان دیگر داریم:

$$\liminf_{k \to \infty} \|g_k\| = 0.$$

اثبات. مجموعه D را به صورت $D = D_1 \cup D_2$ افراز می‌کنیم که در آن مجموعه D_1، مجموعه تمام تکرارهای x_k است که رابطه (۲۲) را ارضا می‌کند. همچنین، D_2، مجموعه تکرارهایی است که از طریق گام ۴ الگوریتم اصلی به‌دست آمده‌اند.

اگر $|D_2| = \infty$، آن‌گاه جهت جستجوی d_k که از (۴) به‌دست آمده است در شرط زوتندیک [۲] صدق می‌کند. بنابراین داریم:

$$\lim_{k \to \infty} \frac{g_k^T d_k}{\|d_k\|} = 0. \qquad (۲۴)$$

در نتیجه در این حالت همگرایی الگوریتم ثابت شده است. برای اطلاعات بیشتر می‌توانید به [۲] رجوع کنید.

فرض کنید $|D_1| = \infty$. بدون از دست دادن کلیت مساله می‌توان فرض کرد که مجموعه‌های A و D_2 متناهی هستند و فقط مجموعه D_1 نامتناهی است. این بدان معناست که عدد صحیح و مثبتی مانند M_0 وجود دارد

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

به طوری که برای هر $k \geq M_0$، داریم: $k \in D_1$. علاوه بر این، بدون از دست دادن کلیت مساله و برای راحتی اثبات، می‌توانیم فرض کنیم $M_0 = 0$. بنابراین

$$D_1 = \{0, 1, 2, \ldots\}. \quad (25)$$

در ادامه برای اثبات همگرایی در این حالت چند لم مطرح می‌کنیم.

لم ۳.۳. فرض کنید دنباله $\{x_k\}_{k \in D_1}$ توسط الگوریتم اصلی تولید شده است. در این صورت داریم:

$$f_{k+1} \leq |f_0| \prod_{i=0}^{k} (1+\phi_i) - \omega_k, \quad (26)$$

که در آن برای برخی $\sigma > 0$، $\omega_k = \max\limits_{1 \leq h \leq i_k} \left\{\sigma \left\| z_k^h - x_k \right\|\right\}$ و i_k بزرگ‌ترین عدد صحیحی است که

$$x_{k+1} = z_k^{i_k}.$$

اثبات. با توجه به (۲۲)، برای تمامی $k \in D_1$ داریم :

$$(1+\phi_k) R_k - f(x_{k+1}) \geq \omega_k \geq 0. \quad (27)$$

این لم را با استفاده از استقرا اثبات می‌کنیم.

برای $k = 0 \in D_1$ با استفاده از (۶) داریم:

$$f_1 \leq (1+\phi_0) R_0 - \omega_0 = (1+\phi_0) f_0 - \omega_0 \leq (1+\phi_0) |f_0| - \omega_0.$$

حال فرض کنید (فرض استقرا) که رابطه (۲۶) برای $1 \leq k \in D_1$ برقرار باشد. یعنی:

$$f_{k+1} \leq |f_0| \prod_{i=0}^{k} (1+\phi_i) - \omega_k \leq |f_0| \prod_{i=0}^{k} (1+\phi_i). \quad (28)$$

نشان می‌دهیم (۲۶) برای $k+1$ برقرار است. یعنی:

$$f_{k+2} \leq |f_0| \prod_{i=0}^{k+1} (1+\phi_i) - \omega_{k+1}.$$

برای این منظور با استفاده از (۶) داریم:

$$\begin{aligned}
f_{k+2} &\leq (1+\phi_{k+1}) R_{k+1} - \omega_{k+1} \\
&= (1+\phi_{k+1})[\epsilon_{k+1} f_{\ell(k+1)} + (1-\epsilon_{k+1}) f_{k+1}] - \omega_{k+1} \\
&\leq (1+\phi_{k+1})[\epsilon_{k+1} f_{\ell(k+1)} + (1-\epsilon_{k+1}) f_{\ell(k+1)}] - \omega_{k+1} \\
&\leq (1+\phi_{k+1}) f_{\ell(k+1)} - \omega_{k+1} \\
&\leq (1+\phi_{k+1}) |f_0| \prod_{i=0}^{\ell(k+1)-1} (1+\phi_i) - \omega_{k+1},
\end{aligned}$$

که در آن آخرین نامساوی از (۲۸) و این حقیقت که $\ell(k+1) \leq k+1$ به‌دست آمده است. بنابراین

$$f_{k+2} \leq (1+\phi_{k+1}) |f_0| \prod_{i=0}^{k} (1+\phi_i) - \omega_{k+1} \leq |f_0| \prod_{i=0}^{k+1} (1+\phi_i) - \omega_{k+1}.$$

که این رابطه اثبات را تکمیل می‌کند.

لم ۳.٤. برای دنباله $\{x_k\}_{k \in D_1}$ تولید شده توسط الگوریتم اصلی داریم:

$$\{x_k\} \subseteq L.$$

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

اثبات. از استقرا کمک می‌گیریم. با توجه به این که $R_0 = f_0$، در نتیجه رابطه برای $k = 0$ برقرار است. حال فرض کنید $x_k \in L$. نشان می‌دهیم $x_{k+1} \in L$. برای این منظور با استفاده از لم ۳.۳، رابطه (۸) و همچنین رابطه بین مقدار میانگین حسابی و میانگین هندسی داریم:

$$\begin{aligned} f_{k+1} &\le | f_0 | \prod_{i=0}^{k} (1+\phi_i) \\ &\le | f_0 | \left(\frac{1}{k+1} \sum_{i=0}^{k} (1+\phi_i) \right)^{k+1} n \\ &\le | f_0 | \left(1+\frac{1}{k+1} \sum_{i=0}^{k} \eta_i \right)^{k+1} \\ &\le | f_0 | \left(1+\frac{\eta}{k+1} \right)^{k+1} \le e^{\eta} | f_0 |, \end{aligned}$$

که در آن دو نامساوی آخر از (۹) و این حقیقت که حد عبارت $\left(1+\frac{\eta}{k+1}\right)^{k+1}$ به e^{η} میل می‌کند، به‌دست آمده است. بنابراین $x_{k+1} \in L$.

لم ۳.۵. فرض کنید دنباله $\{x_k\}_{k \in D_1}$ که توسط الگوریتم اصلی تولید شده است، نامتناهی باشد و $M > 0$ ثابت تعریف شده در (۷) باشد. آن گاه برای هر $k \in D_1$، عدد صحیح و نامنفی L و $0 \le r \le M-1$ وجود دارند، به طوری که $k = LM + r$ و

$$f_{k+1} = f_{LM+r+1} \le | f_0 | \prod_{i=0}^{LM+r} (1+\phi_i) - \sum_{i=0}^{L} \tilde{\omega}_i, \quad L = 0,1,2,\dots, \tag{۲۹}$$

که در آن $\tilde{\omega}_i = \min_{iM \le j \le (i+1)M-1} \omega_j$ و ω_j در لم ۳.۳ تعریف شده است.

اثبات. از استقرا استفاده می‌کنیم.

با توجه به لم ۳.۳ برای $L = 0$ داریم:

$$f_{r+1} \le | f_0 | \prod_{i=0}^{r} (1+\phi_i) - \omega_r \le | f_0 | \prod_{i=0}^{r} (1+\phi_i) - \tilde{\omega}_0, \quad r = 0,1,\dots,M-1,$$

که در آن $\tilde{\omega}_0 = \min_{0 \le l \le M-1} \omega_l$. حال فرض کنید (۲۹) برای $L = t$ برقرار است. یعنی داریم:

$$f_{tM+r+1} \le | f_0 | \prod_{i=0}^{tM+r} (1+\phi_i) - \sum_{i=0}^{t} \tilde{\omega}_i, \quad r = 0,1,\dots,M-1. \tag{۳۰}$$

هدف ما برقراری رابطه برای $L = t+1$ می‌باشد. یعنی

$$f_{(t+1)M+r+1} \le | f_0 | \prod_{i=0}^{(t+1)M+r} (1+\phi_i) - \sum_{i=0}^{t+1} \tilde{\omega}_i, \quad r = 0,1,\dots,M-1. \tag{۳۱}$$

برای این منظور از استقرا روی r استفاده می‌کنیم. از (۲۷) برای $r = 0$ داریم:

$$\begin{aligned} f_{(t+1)M+1} &\le (1+\phi_{(t+1)M}) R_{(t+1)M} - \omega_{(t+1)M} \\ &\le (1+\phi_{(t+1)M}) f_{\ell((t+1)M)} - \omega_{(t+1)M}. \end{aligned} \tag{۳۲}$$

حال چون

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

$$\ell((t+1)M) \le (t+1)M, \qquad (۳۳)$$

از (۳۰) و (۳۲) داریم:

$$f_{(t+1)M+1} \le (1+\phi_{(t+1)M})\left\{|f_0| \prod_{i=0}^{\ell((t+1)M)-1}(1+\phi_i) - \sum_{i=0}^{t}\tilde{\omega}_i\right\} - \omega_{(t+1)M}$$

$$\le (1+\phi_{(t+1)M})\left\{|f_0| \prod_{i=0}^{(t+1)M-1}(1+\phi_i) - \sum_{i=0}^{t}\tilde{\omega}_i\right\} - \omega_{(t+1)M}$$

$$= |f_0| \prod_{i=0}^{(t+1)M}(1+\phi_i) - (1+\phi_{(t+1)M})\sum_{i=0}^{k}\tilde{\omega}_i - \omega_{(t+1)M}$$

$$\le |f_0| \prod_{i=0}^{(t+1)M}(1+\phi_i) - \sum_{i=0}^{t}\tilde{\omega}_i - \omega_{(t+1)M}$$

$$\le |f_0| \prod_{i=0}^{(t+1)M}(1+\phi_i) - \sum_{i=0}^{t}\tilde{\omega}_i - \tilde{\omega}_{t+1}.$$

بنابراین (۳۱) برای $r=0$ برقرار است. حال فرض کنید (۳۱) برای $r=j\le M-2$ برقرار است. برای $r=j+1$ از (۲۷) داریم:

$$f_{(t+1)M+j+1} \le (1+\phi_{(t+1)M})R_{(t+1)M+j} - \omega_{(t+1)M+j}$$
$$\le (1+\phi_{(t+1)M+j})f_{\ell((t+1)M)+j} - \omega_{(t+1)M+j}.$$

علاوه بر آن از (۳۳) و فرض استقرا روی r داریم:

$$f_{(t+1)M+j+1} \le (1+\phi_{(t+1)M+j})\left\{|f_0| \prod_{i=0}^{\ell((t+1)M)+j-1}(1+\phi_i) - \sum_{i=0}^{k}\tilde{\omega}_i\right\} - \omega_{(t+1)M+j}$$

$$\le (1+\phi_{(t+1)M+j})\left\{|f_0| \prod_{i=0}^{(t+1)M+j-1}(1+\phi_i) - \sum_{i=0}^{t}\tilde{\omega}_i\right\} - \omega_{(t+1)M+j}$$

$$\le |f_0| \prod_{i=0}^{(t+1)M+j}(1+\phi_i) - (1+\phi_{(t+1)M+j})\sum_{i=0}^{t}\tilde{\omega}_i - \omega_{(t+1)M+j}$$

$$\le |f_0| \prod_{i=0}^{(t+1)M+j}(1+\phi_i) - \sum_{i=0}^{t}\tilde{\omega}_i - \tilde{\omega}_{t+1}$$

$$\le |f_0| \prod_{i=0}^{(t+1)M+j}(1+\phi_i) - \sum_{i=0}^{t+1}\tilde{\omega}_i.$$

که این اثبات را کامل می‌کند.

قضیه ۳.۶. فرضیات $A1-A3$ را در نظر بگیرید، در این صورت الگوریتم اصلی یا در یک نقطه ایستای مساله (۱) متوقف می‌شود یا برای دنباله نامتناهی $\{x_k\}_{k\in D_1}$ خواهیم داشت [۳، ۸]:

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

که در آن مجموعه D_1، مجموعه تمام تکرارهای x_k است که در رابطه (۲۲) صدق می‌کند.

اثبات. فرض کنید الگوریتم اصلی در یک نقطه ایستا متوقف نشود. با توجه به لم‌های ۳.۴ و ۳.۵، برای هر $k = LM + r \in D_1$، داریم:

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

$$\sum_{i=0}^{L}\tilde{\omega}_i \le |f_0| \prod_{i=0}^{LM+r-1}(1+\phi_i) - f_{LM+r} \le e^{\eta}|f_0| - f_{LM+r}.$$

با گرفتن حد از دو طرف این نامساوی وقتی $L \to \infty$ و استفاده از فرضیات $A1$ و $A2$ به این نتیجه می‌رسیم که $\sum_{i=0}^{\infty}\tilde{\omega}_i < \infty$. در نتیجه $\lim_{i\to\infty}\tilde{\omega}_i = 0$.

حال با توجه به تعریف $\tilde{\omega}_i$، عدد $0 \le \tilde{r} \le M-1$ وجود دارد به طوری که $\tilde{\omega}_i = \omega_{iM+\tilde{r}}$. قرار دهید $\beta_i = iM + \tilde{r}$. در نتیجه $\omega_{\beta_i} \to 0$، وقتی $i \to \infty$. بنابراین، با توجه به تعریف ω_i داریم:

$$0 \le \sigma\left\|z_{\beta_i}^h - x_{\beta_i}\right\| \le \omega_{\beta_i}, \qquad \forall 1 \le h \le i_{\beta_i},$$

که در آن i_{β_i} در لم ۳.۳ تعریف شده است. بنابراین با توجه به $i \to \infty$ و $\omega_{\beta_i} \to 0$ و این که $\sigma > 0$ یک ثابت است، خواهیم داشت:

$$\left\|z_{\beta_i}^h - x_{\beta_i}\right\| \to 0, \qquad \forall 1 \le h \le i_{\beta_i}.$$

در حالت خاص، برای $h = 1$ داریم:

$$\|d_{\beta_i}\| = \left\|z_{\beta_i}^1 - x_{\beta_i}\right\| \to 0.$$

با توجه به این که d_{β_i} در (۱۹) صدق می‌کند و $\|d_{\beta_i}\| \to 0$، در نتیجه خواهیم داشت:

$$\left\|g_{\beta_i}\right\| \to 0, \qquad i \to \infty.$$

این نشان می‌دهد که زیر دنباله $\{x_{\beta_i}\}_{\beta_i \in D_1}$ از $\{x_k\}_{k\in D_1}$ به یک نقطه ایستای f همگراست. در نتیجه داریم:

$$\liminf_{k\to\infty, k\in D_1}\|g_k\| = 0.$$

اکنون، با توجه به بررسی همگرایی در سه حالت $|D_1| = \infty, |A| = \infty$ و $|D_2| = \infty$ می‌توانیم قضیه کلی همگرایی را به صورت زیر مطرح کنیم.

قضیه ۳.۷. فرضیات $A1-A3$ را در نظر بگیرید. در این صورت، الگوریتم اصلی یا در یک نقطه ایستا برای مساله (۱) متوقف می‌شود یا دنباله‌ی نامتناهی $\{x_k\}$ را تولید می‌کند به طوری که

$$\liminf_{k\to\infty}\|g_k\| = 0.$$

اثبات. [۳، ۸].

٤ نتایج عددی

نتایج عددی الگوریتم۲-۱، در این بخش گزارش شده است. در این قسمت الگوریتم جدید شرح داده شده در این مقاله با نمادTRFBB، را به همراه الگوریتم‌های زیر مورد بررسی و مقایسه قرار می‌دهیم:

- TR: الگوریتم۱ در [۱۳].
- NNFBB: الگوریتم۱ در [۱۷].

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

تمامی الگوریتم‌ها در محیط MATLAB نسخه ۷.۱۰.۰ برروی کامپیوتری با پردازنده ۶۴۰۰ *Core i5* و حافظه رم ۸ گیگابایت پیاده‌سازی شده‌اند. مسایل آزمونی از مراجع [۱۸، ۱۹] گرفته شده‌اند. ابعاد مسایل آزمونی بین ۱۰۰۰ تا ۱۰۰۰۰ متغیر است.

شرط توقف برای کلیه الگوریتم‌ها $\|g(x_k)\| \le 10^{-6}$ در نظرگرفته شده است. همچنین، حداکثر تعداد تکرارها و محاسبه مقدار تابع را به ترتیب ۱۰۰۰۰ و ۵۰۰۰۰ در نظرگرفتیم. برای مقایسه بهترکارایی الگوریتم‌ها از نمودار دولان-موره [۲۰] استفاده کرده‌ایم. مقادیر زیر برای پارامترهای الگوریتم‌ها درنظرگرفته شده اند:

$$\mu = 0/1, \mu_1 = 0/25, \mu_2 = 0/75, \epsilon_{\min} = 10^{-6}, \epsilon_{\max} = 10^{6}, \Delta_{\max} = 100, M = 20,$$
$$\sigma_0 = c_2 = 0/5, \sigma_1 = c_1 = 4, \nu_{\max} = \sigma_1^4, \nu_0 = 0/25, \varepsilon = \epsilon = 10^{-6}, \delta = 10^{-6},$$
$$\sigma_{\min} = 10^{-6}, \sigma_{\max} = 10^{6}, \gamma = 10^{-4}, \tau_{\min} = 0/1, \tau_{\max} = 0/5, \eta_k = \frac{1}{(1+k)^2}, k \ge 0.$$

نتایج عددی درجدول ۱ گزارش شده است. در این جدول، *Problem*، n، n_i، n_f و n_g به ترتیب مشخص‌کننده، نام مساله، بعدمساله، تعداد تکرارهای الگوریتم، تعداد محاسبه مقدار تابع و تعداد محاسبه مقدار مشتق تابع هستند. بر اساس اطلاعات این جدول، نمودارهای دولان موره، عملکرد بر اساس n_i، n_f و n_g به ترتیب درشکل‌های ۱، ۲ و ۳ ترسیم شده‌اند.

درشکل ۱، می‌توان دیدکه الگوریتم TRFBB حدود ۹۳ درصد مسایل را با موفقیت حل می‌کند در صورتی که این درصد برای الگوریتم‌های NNFBB و TR نزدیک به ۹۰ درصد است. همچنین، الگوریتم TRFBB حدود ۶۰ درصد مسایل را درکمترین مقدار n_i حل می‌کند، در صورتی که این درصد برای الگوریتم‌های NNFBB وTR به ترتیب ۴۶ و۳۸ درصد است. شکل ۲ نشان می‌دهدکه وقتی هر سه الگوریتم روی مجموعه مسایل آزمایشی اجرا می‌شوند، الگوریتم TRFBB حدود ۵۸ درصد مسایل را در کم‌ترین میزان محاسبه مقدار تابع حل می‌کند، در صورتی که این درصد برای الگوریتم‌های NNFBB و TR به ترتیب ۴۵ و ۲۱ درصد است. علاوه بر آن برای میزان محاسبه مشتق تابع، شکل۳ نشان می‌دهدکه، الگوریتم TRFBB حدود ۴۷ درصد مسایل را درکم‌ترین میزان محاسبه مقدارمشتق تابع حل می‌کند، در صورتی که این درصد برای الگوریتم‌هایNNFBB وTRبه ترتیب ۳۶ و۲۶ درصد است.

جدول ۱. نتایج عددی اجرای الگوریتم‌ها بر روی مسایل آزمون

Problem	n	TR $n_i / n_f / n_g$	NNFBB $n_i / n_f / n_g$	TRFBB $n_i / n_f / n_g$
BDQRTIC	1000	825/20252/14555	519/22632/13871	501/22010/13105
BDQRTIC	5000	960/22150/16336	375/17681/10275	354/16253/9962
CRAGGLVY	1000	290/6623/3231	203/3512/1170	192/3145/956
CRAGGLVY	5000	Failed	206/2429/1716	206/2432/1753

[Downloaded from mail.jorlu.ir on 2026-11-30]

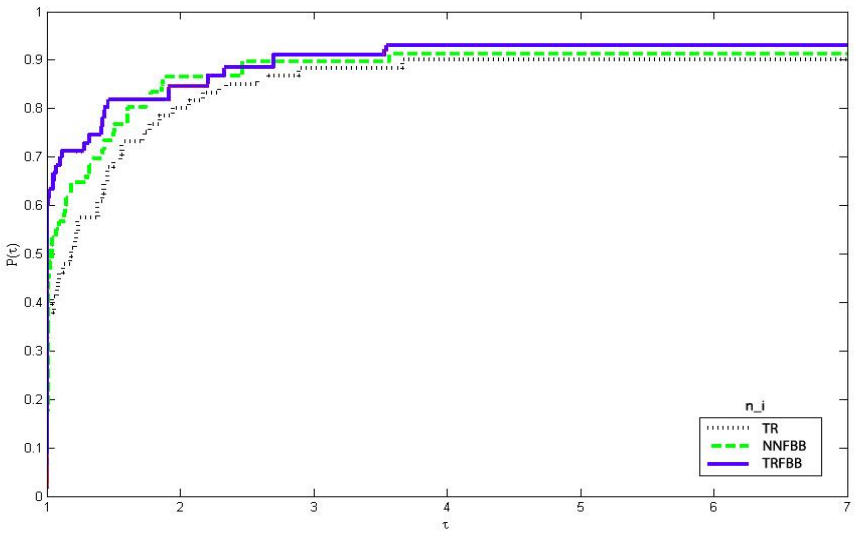
[DOR: 20.1001.1.22517286.2020.17.1.7.3]

FMINSURF	1024	180/978/631	310/1480/1480	395/1620/1521
FREUROTH	1000	Failed	966/29424/12196	1005/30225/14523
FREUROTH	5000	Failed	314/1770/1324	286/1125/452
LIARWHD	1000	33/451/341	81/311/160	80/335/185
LIARWHD	5000	189/1215/884	177/856/478	123/456/425
MOREBV	1000	1338/1914/1452	120/1024/452	125/1163/665
MOREBV	5000	1922/2911/1971	170/2011/1110	236/1523/875
NCB20	1000	942/2356/3122	1133/6970/2970	952/5201/2563
NCB20B	1000	1010/4668/3501	303/1462/653	274/1325/745
NCB20B	2000	1652/7005/4922	396/5635/2237	312/4563/2102
NONCVXUN	1000	7078/12525/7676	2762/14934/2875	2626/16256/6253
NONDIA	1000	448/1998/899	218/1155/668	201/1002/421
NONDQUAR	1000	791/1890/888	624/1728/754	423/1524/653
POWELLSG	1000	158/1200/601	102/450/123	85/321/152
POWELLSG	5000	226/1703/852	82/475/150	90/562/256
POWELLSG	10000	299/1999/1225	35/490/157	28/125/117
POWER	1000	3988/21501/12501	3225/19857/11225	3101/20540/15602
DIXMAANA	3000	11/24/24	7/55/18	6/32/17
DIXMAANB	3000	13/23/20	21/244/140	20/124/101
DIXMAANC	3000	10/12/10	8/41/27	8/40/22
DIXMAAND	3000	18/51/33	19/145/84	12/101/63
DIXMAANE	3000	130/165/110	343/779/203	331/761/178
DIXMAANF	3000	335/462/199	239/643/189	201/546/177
DIXMAANG	3000	332/683/290	174/876/184	166/712/102
DIXMAANH	3000	301/989/245	173/629/213	170/605/195
DIXMAANI	3000	522/570/570	705/921/852	456/854/552
DIXMAANJ	3000	405/1520/787	195/387/105	145/312/109
DIXMAANK	3000	652/1905/656	165/2031/705	179/2020/889
DIXMAANL	3000	562/1804/670	459/1299/94	405/1025/101

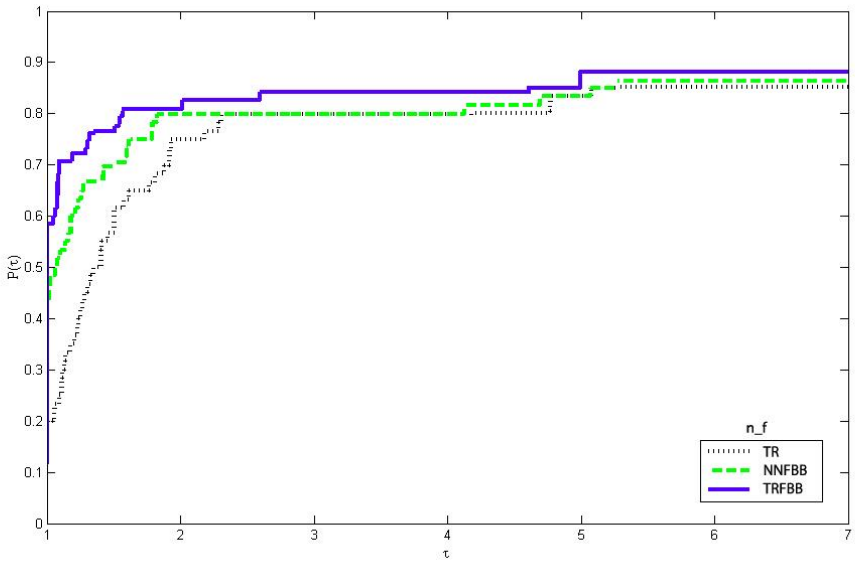
[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

ARWHEAD	5000	667/4015/2155	217/3158/1197	665/3998/2026
BRYBND	5000	20/154/143	26/509/312	20/156/147
BRYBND	10000	216/1103/645	163/729/433	151/656/326
DQRTIC	1000	705/13124/10256	345/12328/6165	295/11015/5878
DQRTIC	5000	1903/22501/15603	Failed	Failed
EDENSCH	2000	12/15/15	11/155/84	11/12/12
ENGVAL1	5000	31/159/105	21/184/69	15/150/123



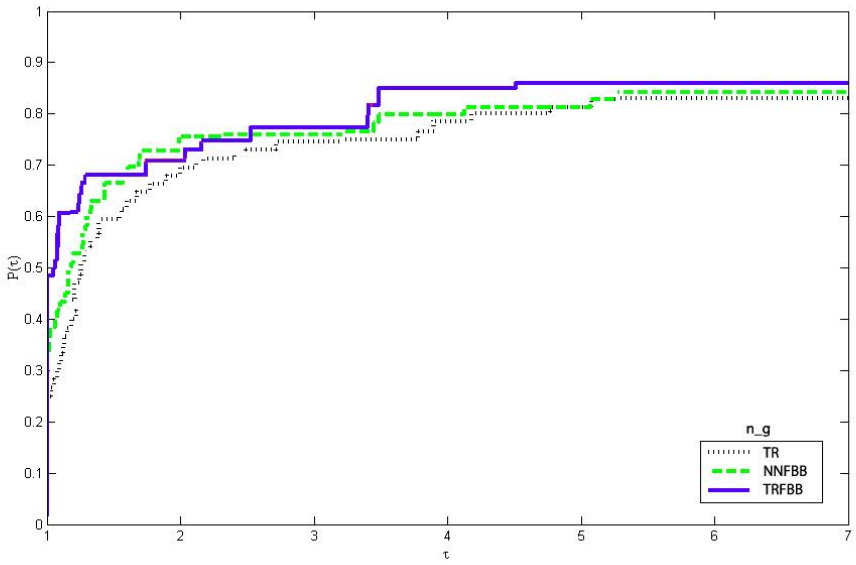
شکل ۱. نمودار دولان-موره برای تعداد تکرارها از جدول ۱



شکل ۲. نمودار دولان-موره برای تعداد محاسبه مقدار تابع از جدول ۱

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]



شکل۳. نمودار دولان–مورہ برای تعداد محاسبه مقدار مشتق تابع از جدول۱

۵ نتیجه‌گیری

در این مقاله، یک الگوریتم جدید نایکنوای فیلتردار برای حل مسایل بهینه‌سازی نامقید (بزرگ مقیاس) معرفی کردیم. این الگوریتم، بر روی یک الگوریتم ناحیه اعتماد نایکنوا مفهوم فیلتر متناهی پیاده‌سازی شده است. با استفاده از این مفهوم، بسیاری از نقاط آزمایشی به‌دست آمده در تکرارها که روش‌های قبلی (بدون فیلتر) آن‌ها را حذف می‌کردند، ذخیره شدند. سپس، در تکرارهای بعدی از اطلاعات آن‌ها برای تسریع رسیدن به جواب استفاده شد. علاوه بر این، الگوریتم جدید مجهز به شرط نایکنوای خاصی شده است که در [۱۶] ارایه شده بود. در ادامه، تحت شرایط استاندارد، همگرایی سراسری الگوریتم پیشنهادی را ثابت کردیم. نتایج عددی حاصل از پیاده‌سازی الگوریتم پیشنهادی به همراه برخی دیگر از الگوریتم‌های موجود در ادبیات موضوع نشان از کارایی این روش روی مسایل آزمونی در مقایسه با سایر الگوریتم‌های در نظر گرفته شده دارد.

سپاس‌گزاری

نویسندگان این مقاله از امور پژوهشی دانشگاه کاشان و مرکز پژوهشی اسکوپ برای حمایت از این پژوهش تقدیر و قدردانی می‌نمایند.

منابع

[۱] عهدی اقدم، ص. رضاپور، ر. قوسی، س.، (۱۳۸۸). روش‌های تکراری شبه نیوتن برای دستگاه‌های معادلات غیرخطی. مجله تحقیق در عملیات درکاربردهای آن، ۶ (۲۳)، ۱۰-۱.

[۱۸] ارزانی،فرشید. پیغامی، محمدرضا.، (۱۳۹۵). یک روش نایکنوای فیلتردار برای حل دستگاه معادلات غیرخطی. تحقیق درعملیات درکاربردهای آن، ۱۳ (۲) ، ۸۵-۹۹.

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]

[2] Nocedal, J., Wright, S.J., (2006). Numerical Optimization. Springer, New York.
[3] Li, G.D. (2006). A trust region method with automatic determination of the trust region radius, Chinese J. Engry. Math., 23, 843–848.
[4] Sartenear, A. (1997). Determination of an initial truet region in nonlinear programming. SIAM J.SCI. Comut., 18(6), 1788-1803.
[5] Zhang, X.S., Zhang, J.L., Liao, L.Z. (2002). An adaptive trust region method and its convergence, Sci. China, 45, 620–631.
[6] Zhou, Q., Zhang, C. (2012). A new nonmonotone adaptive trust region method based on simple quadratic models, J.Appl. Math. Comput., 40 ,111–123.
[7] Barzilai, J., M.Borwein, J.(1988). Two-point step size gradient methods. J.Numer. Anal., 8,141-148.
[8] Ahookhosh, M., Amini, K. , Peyghami, M.R. (2012). Anonmonotone trust-region linesearch method for large-scale unconstrained optimization. Appl. Math. Model, 36, 478-487.
[9] Shi, Z.J., Guo, J.H. (2008). A new trust region methods forunconstrained optimization. J. Comput. Appl. Math., 213, 509-520.
[10] Ataee Tarzanagh, D., Peyghami, M. R., Mesgarani, H. (2014). A new nonmonotone trust region method for unconstrained optimization equipped by an efficient adaptive radius.Optim. Methods Softw., 29(4) , 819–836.
[11] Wei, Z., Li, G., Qi, L. (2006). New quasi-Newton methods for unconstrained optimization problems. Appl. Math.Comput., 175, 1156–1188.
[12] Biglari, F. , Solimanpur, M. (2013). Scaling on the spectral gradient Method. J. Optim. Theory Appl.,158(2), 626-635.
[13] Saeidian, Z., Peyghami, M.R., (2015). An adaptive nonmonotone trust region method for unconstraine optimization problemsbased on a simple subproblem, Iranian J. of numerical analysis and optimization, 2, 95-117.
[14] Sang, Z., Sun, Q. (2009). A self-adaptive trust region method with linesearch based on a simple subproblem model. J.Comput. Appl. Math., 232, 514–522.
[15] Fletcher, R., Leyff er, S. (2002). Nonlinear programming without a penalty function. Math. Program., 91, 239-269.
[16] Fatemi, M. , Mahdavi-Amiri, N. (2012). A non-monotone trust region algorithm for unconstrained optimization with dynamic reference iteration updates using filter.Optimization, 61(6), 733-763.
[17] Arzani, F., Reza Peyghami M.R. (2016). A new nonmonotone filter Barzilai-Borwein method for solving unconstrained optimization problems. Int. J. Comput. Math., 93(3) ,596-608.
[18] Andrei, N. (2008). An unconstrained optimization test functions collection. Advanced Modeling and Optimization, 10, 147-161.
[19] Hock, W., Schittkowski, K. (1981).Test Examples for Nonlinear Programming Codes. Springer Series Lectures Notes in Economics Mathematical Systems.
[20] Dolan, E., More, J.J. (2002). Benchmarking optimization software with performance problems. Mathematical Programming, 91, 201-213.

[Downloaded from mail.jorlu.ir on 2026-11-30]

[DOR: 20.1001.1.22517286.2020.17.1.7.3]