مجله تحقیق در عملیات در کاربردهای آن، سال سیزدهم، شماره دوم (پیاپی ۴۹)، تابستان ۹۵، ص ص ۹۹-۸۵
شاپا ۷۲۸۶-۲۲۵۱

یک روش نایکنوای فیلتردار برای حل دستگاه معادلات غیرخطی

فرشید ارزانی[۱،۲]، محمدرضا پیغامی[۱،۲*]

۱- دانشجوی دکتری، مرکز پژوهشی محاسبات علمی در بهینه‌سازی و مهندسی سامانه‌ها، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران
۲- دانشیار، دانشگاه صنعتی خواجه نصیر الدین طوسی، گروه ریاضی کاربردی، تهران، ایران

رسید مقاله: ۸ آبان ۱۳۹۴
پذیرش مقاله: ۲۴ فروردین ۱۳۹۵

چکیده

در این مقاله یک روش جدید برای حل دستگاه معادلات غیرخطی معرفی می‌کنیم که در آن از یک استراتژی نایکنوا مشتق آزاد استفاده شده است. هم‌چنین، این روش با استفاده از تکنیک فیلتر پیاده‌سازی شده است. با استفاده از این مفهوم، بسیاری از نقاط آزمایشی به‌دست آمده در تکرارها که توسط الگوریتم‌های جستجوی خطی حذف می‌شدند، ذخیره خواهند شد. الگوریتم جدید از اطلاعات نقاط موجود در فیلتر برای به‌دست آوردن نقاط بعدی بهره می‌برد و باعث افزایش سرعت همگرایی الگوریتم می‌شود. در این روش از یک تکنیک نایکنوای خاص استفاده شده است که به الگوریتم اجازه می‌دهد از خاصیت نایکنوایی از همان اولین تکرار بهره‌مند شود. همگرایی سراسری الگوریتم تحت برخی شرایط استاندارد اثبات می‌شود. نتایج عددی به‌دست آمده از روش جدید روی مجموعه مسایل آزمونی بیانگر کارایی روش پیشنهادی نسبت به برخی روش‌های موجود در ادبیات موضوع است.

کلمات کلیدی: دستگاه معادلات غیرخطی، تکنیک فیلتر، روش جستجوی خطی نایکنوا، همگرایی سراسری.

۱ مقدمه

دستگاه معادلات غیرخطی زیر را در نظر می‌گیریم:

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

که در آن $F:\mathbb{R}^n\to\mathbb{R}^n$ یک نگاشت به طور پیوسته مشتق پذیر است. فرض می‌کنیم دستگاه (۱) یک دستگاه متقارن است، یعنی ماتریس ژاکوبی آن متقارن است. برای دستگاه (۱)، تابع نرم زیر را در نظر می‌گیریم:

$$f(x)=\frac{1}{2}\|F(x)\|^2. \qquad (2)$$

برای حل دستگاه معادلات (۱) روش‌های بسیاری پیشنهاد شده است که می‌توان به روش‌های نیوتن، شبه نیوتن [۱و۲]، گاوس-نیوتن، ناحیه‌ی اعتماد، گرادیان مزدوج و گرادیان طیفی اشاره کرد. نکته‌ی مهم در هر یک از این

* عهده‌دار مکاتبات
آدرس الکترونیکی: peyghami@kntu.ac.ir

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]
[DOR: 20.1001.1.22517286.2016.13.2.6.6]

روش‌ها، نحوه‌ی برخورد با مسایل مقیاس بزرگ است. به عنوان نمونه، روش‌های گرادیان طیفی و گرادیان مزدوج عملکرد مناسبی در حل دستگاه معادلات مقیاس بزرگ دارند [۳و۴].

با مروری بر ادبیات موضوع، نتایج عددی گواه این حقیقت است که روش‌های تکراری مجهز شده به یک استراتژی مناسب نایکنوا، رفتار همگرایی بهتری نسبت به روش‌های مشابه دارند [۵و۶]. گریپو و همکاران اولین روش نایکنوایی را برای روش نیوتن ارایه کردند و آن را برای حل مسایل مختلف بهینه‌سازی بسط دادند[۵]. در روش آن‌ها، در نقطه‌ی فعلی x_k، برای عدد صحیح مثبت M و $\gamma\in(0,1)$ داده شده، مقدار $\lambda\in(0,1]$ طوری تعیین می‌شود که

$$f(x_k+\lambda d_k)\le \max_{0\le j\le \min\{k,M\}} f(x_{k-j})+\gamma\lambda\nabla f(x_k)^T d_k. \qquad (۳)$$

با این وجود، تکنیک نایکنوای گریپو دارای مشکلاتی است که برخی از آن‌ها در [۷] آمده‌اند. برای رفع این مشکلات، آهوخوش و همکاران در [۸] جمله‌ی نایکنوا به صورت زیر ارایه دادند:

$$R_k=\epsilon_k f_{\ell(k)}+(1-\epsilon_k)f_k, \qquad (۴)$$

که در آن

$$f_{\ell(k)}=\max_{0\le j\le \min(k,M)} f_{k-j}, \qquad (۵)$$

$$f_k=f(x_k),\ \epsilon_k\in[\epsilon_{\min},\epsilon_{\max}]\subset[0,1].$$

در [۱۶] ثابت شده است که نتایج همگرایی بهتر برای یک الگوریتم زمانی حاصل می‌شود که در نقاط دور از جواب، تکنیک‌های نایکنوا نقش بیشتری نسبت به یکنوا داشته باشند (نایکنوا قوی) و در نزدیکی جواب این رفتار برعکس شود (نایکنوا ضعیف). رابطه (۴) با انتخاب مناسب ϵ_kها از استراتژی‌های نایکنوا قوی و ضعیف به خوبی بهره می‌برد. در واقع، برای ϵ_k نزدیک یک خاصیت نایکنوا قوی را داریم و برای ϵ_k نزدیک صفر خاصیت نایکنوا ضعیف برقرار است.

به دلیل آن‌که ماتریس ژاکوبی دستگاه (۱) ممکن است موجود نباشد و یا محاسبه‌ی آن هزینه‌بر و نیاز به حافظه‌ی بالایی برای ذخیره‌سازی باشد، روش‌های مشتق آزاد معرفی شده‌اند. کروز و همکاران در [۹] برای دستگاه‌های مربعی با استفاده از ترکیب روش‌های جستجوی خطی لی و فوکوشیما [۱۰] و نایکنوای گریپو، روش باقیمانده‌ی طیفی مشتق آزاد با نام DFSANE ارایه دادند. در روش آن‌ها، طول گام α_k فقط و فقط از یکی از شرایط زیر به دست می‌آید

$$f(x_k-\alpha_k d_k)\le f_\ell(k)+\eta_k-\gamma\alpha_k\|F(x_k)\|^2, \qquad (۶)$$

$$f(x_k+\alpha_k d_k)\le f_\ell(k)+\eta_k-\gamma\alpha_k\|F(x_k)\|^2, \qquad (۷)$$

که در آن $d_k:=\sigma_k F(x_k)$ و σ_k ضریب طیفی برزیلای-بروین به صورت زیر است [۱۱]:

$$\sigma_k=\frac{s_k^T s_k}{y_k^T s_k}, \qquad (۸)$$

که در آن $s_k=x_{k+1}-x_k$ و $y_k=F(x_{k+1})-F(x_k)$.

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

روش‌های زیادی بر اساس ساختار روش DFSANE در ادبیات موضوع طراحی و پیاده‌سازی شده است که می‌توان برخی از آن‌ها را در [۱۲و۱۳و۱۴] دید. برای نمونه، چن و لی در [۱۵] برای دستگاه‌های مربعی یک روش گرادیان باقیمانده‌ی طیفی مشتق آزاد با نام NDFSANE ارایه دادند. در این روش، جمله‌ی نایکنوای $f_{\ell(k)}$ با جمله تعریف شده در [۱۶] جایگزین شده بود. بعدها روش DFSANE با استفاده از یک تقریب از روش تندترین کاهش در [۱۷] بهبود یافت و با نام SDFSANE معرفی شد.

تکنیک فیلتر اولین بار توسط فلچر و لیفر در [۱۸] ارایه شد. این ایده بر اساس بهینه سازی توابع چند هدفه شکل گرفته است و در آن هر نقطه در مجموعه فیلتر رد یا پذیرش می‌شود. یک نقطه در فیلتر پذیرفته می‌شود هرگاه توسط هیچ یک از نقاط داخل فیلتر مغلوب نشود. هنگامی‌که یک نقطه در فیلتر پذیرفته شد، کلیه نقاطی که توسط نقطه‌ی جدید مغلوب می‌شوند از فیلتر حذف می‌گردند. برای اطلاعات بیش‌تر به [۱۸] مراجعه کنید.

هدف ما در این مقاله ارایه یک روش DFSANE نایکنواست که به فیلتر مجهز شده است. روش ما نسخه‌ی جدیدی از روشDFSANE است که با استفاده از فیلتر و شرط نایکنوا جدید، عملکرد بهتر و سرعت همگرایی مناسب‌تری نسبت به روش‌های مشابه دارد. هم‌چنین، همگرایی سراسری این روش تحت شرایط استاندارد بررسی می‌شود. روش پیشنهادی به همراه برخی روش‌های موجود در ادبیات موضوع روی مجموعه مسایل آزمونی اجرا شده و مورد مقایسه قرار می‌گیرد. نتایج عددی حاصل از این روش، بیانگر عملکرد بهتر آن نسبت به روش‌های در نظر گرفته شده است.

در بخش دوم این مقاله به معرفی فیلتر استفاده شده و الگوریتم پیشنهادی خواهیم پرداخت. همگرایی سراسری الگوریتم در بخش سوم مورد بررسی قرار می‌گیرد. در ادامه، در بخش چهارم نتایج عددی و در بخش پنجم نتیجه‌گیری کلی بیان می‌شود.

۲ ساختار الگوریتم جدید

در این بخش، الگوریتمی جدید بر اساس روش DFSANE را که به یک فیلتر متناهی مجهز شده است، بیان می‌کنیم. عمده تمرکز در طراحی این الگوریتم، حل دستگاه‌های مربعی با بعدهای بزرگ است. برای این منظور، ابتدا فیلتر استفاده شده را به همراه یک گام کامل از الگوریتم معرفی می‌کنیم.

در [۱۹] فیلتری معرفی شده است که با داشتن شرط پذیرش خاص، می‌تواند اندازه‌ی فیلتر را کنترل کند. در این فیلتر، نقطه‌ی x_k نسبت به x_l در فیلتر پذیرفته می‌شود، هرگاه اندیس $j \in \{1,\dots,n\}$ وجود داشته باشد به طوری‌که

$$\left|g_j(x_k)\right|^{\mu_2} + \theta_2 \left\|g(x_k)\right\|^{\mu_1} \le \left|g_j(x_l)\right|^{\mu_2} + \theta_1 \left\|g(x_l)\right\|^{\mu_1}, \qquad (۹)$$

که در آن $g(x) = \nabla f(x)$، μ_1 و μ_2 ثابت‌های مثبت و θ_1 و θ_2 در رابطه‌ی زیر صدق می‌کنند:

$$0 \le \theta_1 < \theta_2 < \frac{1}{\sqrt{n}}. \qquad (۱۰)$$

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

بنابراین نقطه‌ی x_k در فیلتر $\mathcal{F}$ پذیرفته می‌شود، هرگاه x_k نسبت به هر x_l که $g(x_l)\in\mathcal{F}$ در فیلتر پذیرفته شود. در نتیجه $g(x_k)$ به فیلتر افزوده می‌شود و هر $g(x_l)\in\mathcal{F}$ با شرط زیر از فیلتر حذف می‌شود:

$$\left|g_j(x_k)\right|^{\mu_2}+\theta_2\left\|g(x_k)\right\|^{\mu_1}\le\left|g_j(x_l)\right|^{\mu_2}+\theta_1\left\|g(x_l)\right\|^{\mu_1},\qquad\forall\, j\in\{1,\dots,n\}. \tag{11}$$

حال یک مرحله کامل از الگوریتم را شرح می‌دهیم: فرض کنید در گام kام، نقطه‌ی x_k به دست آمده با استفاده از جهت کاهشی $d_k=-\sigma_k F_k$ در شرط خروج صدق نکند. مقادیر $\alpha_+=\alpha_-=1$ را در نظر می‌گیریم. نقاط آزمایشی $x^+_{k+1}=x_k+\alpha_+ d_k$ و $x^-_{k+1}=x_k-\alpha_- d_k$ با اولویت x^+_{k+1} در شرط پذیرش فیلتر بررسی می‌شود. در صورتی‌که یکی از نقاط در فیلتر پذیرفته شود، نقطه‌ی جدید به دست آمده است. در غیر این صورت، اگر هیچ یک از نقاط تولید شده در فیلتر پذیرفته نشوند، نقاط آزمایشی با همان اولویت در شرط جستجوی خطی نایکنوا چک می‌شود. در صورتی‌که یکی از نقاط شرط جستجوی خطی را ارضا کند، نقطه‌ی جدید به دست آمده است. حال فرض کنید نقاط آزمایشی تولید شده نه در فیلتر پذیرفته شود و نه در شرط جستجوی خطی صدق کند، در این حالت عملیات پیمایش معکوس بر روی α_+ و α_- انجام می‌شود و عملیات فوق دوباره تکرار می‌شود. این روند تا زمانی که یکی از نقاط آزمایشی در فیلتر پذیرفته و یا در شرط جستجوی خطی صدق کند، ادامه می‌یابد. در الگوریتم ۱، ساختار الگوریتم با فرآیند فوق با جزییات بیش‌تری ارایه شده است.

الگوریتم ۱. الگوریتم جستجوی خطی مشتق آزاد فیلتردار نایکنوا برای حل دستگاه معادلات غیرخطی

گام ۰ : مقادیر $x_0\in\mathbb{R}^n$، $f_{sup}>0$، $M\ge 0$، $0<\sigma_{\min}<\sigma_{\max}<\infty$، $\epsilon>0$، $0\le\theta_1<\theta_2<\frac{1}{\sqrt{n}}$، $0\le\tau_{\min}<\tau_{\max}<1$ و $0<\gamma<1$ داده شده اند. قرار ده $k=0$، $\mathcal{F}_0=\varnothing$.

گام ۱ : اگر $\|F(x_k)\|\le\epsilon$ توقف کن.

گام ۲ : ضریب طیفی σ_k را با $|\sigma_k|\in[\sigma_{\min},\sigma_{\max}]$ تعیین کن و مقدار $f_{\ell(k)}$ را با استفاده از (۵) محاسبه کن. قرار ده $d_k=-\sigma_k F(x_k)$ و $\alpha_+=\alpha_-=1$.

گام ۳ : بردارهای آزمایشی $x^+_{k+1}=x_k+\alpha_+ d_k$ و $x^-_{k+1}=x_k-\alpha_- d_k$ را محاسبه کن.

اگر x^+_{k+1} توسط فیلتر پذیرفته شد، قرار ده $x_{k+1}=x^+_{k+1}$ و نقطه‌ی x_{k+1} را به فیلتر $\mathcal{F}_k$ اضافه کن. فیلتر را به هنگام کن تا $\mathcal{F}_{k+1}$ تشکیل شود. قرار ده $\alpha_k=\alpha_+$، $k:=k+1$ و به گام ۱ برو.

در غیر این صورت، اگر x^-_{k+1} توسط فیلتر پذیرفته شد، قرار ده $x_{k+1}=x^-_{k+1}$ و نقطه‌ی x_{k+1} را به فیلتر $\mathcal{F}_k$ اضافه کن. فیلتر را به هنگام کن و $\mathcal{F}_{k+1}$ را بساز. قرار ده $\alpha_k=\alpha_-$، $k:=k+1$ و به گام ۱ برو.

در غیر این صورت، اگر

$$f(x^+_{k+1})\le R_k-\gamma\alpha_+^2 f(x_k), \tag{12}$$

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

قرار ده $\alpha_k = \alpha_+$، $k := k+1$، $x_{k+1} = x^+_{k+1}$ و به گام ۱ برو.

در غیر این صورت، اگر

$$f(x^-_{k+1}) \le R_k - \gamma\alpha_-^2 f(x_k), \qquad (۱۳)$$

قرار ده $\alpha_k = \alpha_-$، $k := k+1$، $x_{k+1} = x^-_{k+1}$ و به گام ۱ برو.

گام ٤ : مقادیر $\alpha_+ \in [\tau_{\min}\alpha_+, \tau_{\max}\alpha_+]$ و $\alpha_- \in [\tau_{\min}\alpha_-, \tau_{\max}\alpha_-]$ را انتخاب کن و به گام ۳ برو.

نکته ۱ با استدلالی مشابه لم‌های ۲.۲ و ۳.۲ در [۸]، به سادگی می‌توان نتیجه گرفت که پس از تعداد متناهی کاهش روی α_+ یا α_- در گام ۴ الگوریتم ۱، یکی از شرایط (۱۲) و (۱۳) بالاخره رخ می‌دهد. این نشان می‌دهد که الگوریتم ۱ خوش‌تعریف است .

۳- همگرایی سراسری

در این بخش، همگرایی سراسری الگوریتم ۱ را مورد بررسی قرار می‌دهیم. ابتدا مجموعه‌های $P۱$ و $P۲$ را به صورت زیر تعریف می‌کنیم:

$$P۱ = \{k \mid x_k \in \mathcal{F}_k\} \qquad (۱۴)$$

$$P۲ = \{k \mid x_k \ satisfies \ (۱۲) \ or \ (۱۳)\}. \qquad (۱۵)$$

با توجه به ساختار الگوریتم، به سادگی مشخص است که $P۱ \cap P۲ = \varnothing$. هم‌چنین، فرضیات زیر را روی مساله در نظر می‌گیریم:

$A۱$: تابع F روی $\mathbb{R}^n$ به طوری پیوسته مشتق‌پذیر است.

$A۲$: مجموعه‌ی تراز $L^\circ = \{x \in \mathbb{R}^n \mid f(x) \le f(x_\circ)\}$ بسته و کراندار است.

$A۳$: جملات دنباله‌ی $\{x_k\}$ تولید شده توسط الگوریتم ۱ در مجموعه‌ی بسته و کران‌دار $\Omega \subset \mathbb{R}^n$ قرار دارند.

بر اساس ساختار الگوریتم و هم چنین فرضیات موجود، برای اثبات همگرایی سراسری الگوریتم باید رفتار نقاط $\{x_k\}$ تولید شده از تکرارهای اصلی الگوریتم را در نظر گرفت. فرض کنید دنباله‌ی $\{x_k\}$ توسط الگوریتم ۱ تولید شده است. این الگوریتم یا در یک تکرار مشخص $k_\circ$ متوقف می‌شود که در این صورت به یک نقطه‌ی ایستا می‌رسیم، و یا یک دنباله‌ی نامتناهی از نقاط تولید می‌کند. در حالت دوم، نشان خواهیم داد که دنباله‌ی نامتناهی به دست آمده حداقل دارای یک زیردنباله‌ی همگرا به نقطه‌ی ایستا تابع f است. بنابراین، فرض می‌کنیم $|P۱ \cup P۲| = \infty$. حال دو حالت ممکن داریم: $|P۱| = \infty$ یا $|P۲| = \infty$.

در ادامه این دو حالت را مورد بررسی قرار می‌دهیم.

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

۳-۱ بررسی همگرایی در حالت $|P1|=\infty$

در تکرار kام از الگوریتم، مجموعه‌ی فیلتر $\mathcal{F}_k$ و شرط پذیرش آن را به صورت (۹) در نظر بگیرید. بر اساس لم ۴ در [۱۹]، می‌دانیم که اندازه‌ی فیلتر $\mathcal{F}_k$ متناهی است. با این وجود، این فیلتر می‌تواند تعداد نامتناهی تکرار را بپذیرد و هم‌چنان دارای اندازه‌ی متناهی باشد. در لم زیر همگرایی را در حالتی بررسی می‌کنیم که فیلتر تعداد نامتناهی تکرار را در خود می‌پذیرد.

لم ۱. فرضیات $A1-A3$ را در نظر بگیرید. اگر $|P1|=\infty$، آن‌گاه

$$\lim_{k\to\infty,k\in P1}\|F(x_{k+1})\|=0. \qquad (16)$$

برهان. اثبات این لم شبیه اثبات لم ۵.۴ در [۲۰] با جایگذاری $\theta(x)=F(x)$ است. لازم به ذکر است که با توجه به مقادیر اولیه برای $\alpha_+=\alpha_-=1$، در گام ۲ الگوریتم ۱، شرط $0\leqslant\theta_1<\theta_2<\frac{1}{\sqrt{n}}$ برقرار است. ■

لم ۱ بیان می کند که در حالت $|P1|=\infty$، دنباله $\{x_k\}$ حداقل دارای یک نقطه‌ی حدی است که نقطه‌ی ایستای تابع f است، یعنی

$$\lim_{k\to\infty,k\in P1}\inf\|F(x_k)\|=0. \qquad (17)$$

۳-۲ بررسی همگرایی در حالت $|P2|=\infty$

فرض کنید $|P2|=\infty$. بدون از دست دادن کلیت مساله، می‌توان فرض کرد که مجموعه‌ی $P1$ متناهی است. در این حالت نشان می‌دهیم برای دنباله‌ی نامتناهی $\{x_k\}_{k\in P2}$، یک نقطه‌ی حدی وجود دارد که نقطه‌ی ایستای تابع f است؛ یعنی:

$$\lim_{k\to\infty,k\in P2}\inf\|F(x_k)\|=0. \qquad (18)$$

با توجه به اینکه $P1$ متناهی است عدد صحیح مثبتی مانند M_0 وجود دارد به طوری‌که، برای هر $k\geq M_0$ داریم: $k\in P2$. بنابراین، بدون از دست دادن کلیت مساله و سادگی تحلیل های بعدی، فرض می‌کنیم $M_0=0$. در نتیجه:

$$P2=\{0,1,2,\ldots\}. \qquad (19)$$

لم ۲. فرض کنید دنباله‌ی $\{x_k\}_{k\in P2}$ توسط الگوریتم ۱ تولید شده باشد. در این صورت داریم:

$$f_{k+1}\leq|f_0|-\omega_k, \qquad (20)$$

که در آن $\omega_k=\gamma\alpha_k^2 f(x_k)$.

برهان. با جایگذاری $\phi_i=0$ در اثبات لم ۳ در [۲۱]، به راحتی حکم این لم به طور مشابه نتیجه می شود. ■

لم ۳. برای دنباله‌ی $\{x_k\}_{k\in P2}$ تولید شده توسط الگوریتم ۱ داریم: $\{x_k\}\subseteq L^0$.

برهان. اثبات این لم شبیه اثبات لم ۴ در [۲۱] است و بنابراین در اینجا بیان نمی‌شود. ■

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

نکته ٤. دنباله‌ی $\{x_k\}_{k\in P۲}$ تولید شده توسط الگوریتم ۱ را در نظر بگیرید. با توجه به (۱۹)، برای عدد صحیح و مثبت M، مجموعه‌ی $P۲$ را می‌توان به صورت زیر بازنویسی کرد:

$$P۲=\{LM+r \mid L\in \mathbb{N}\bigcup\{۰\}, ۰\le r\le M-۱\}. \quad (۲۱)$$

لم ۵. فرض کنید $\{x_k\}_{k\in P۲}$ دنباله‌ی نامتناهی تولید شده توسط الگوریتم ۱ بوده و M ثابت مثبتی باشد که در (۵) تعریف شده است. در این صورت، برای هر $k\in P۲$، عدد صحیح نامنفی L و $۰\le r\le M-۱$ وجود دارند، به طوری‌که $k=LM+r$ و

$$f_{k+۱}=f_{LM+r+۱}\le |f_۰|-\sum_{i=۰}^{L}\omega_{s(i)}, \quad L=۰,۱,۲,\ldots, \quad (۲۲)$$

که در آن $\omega_{s(i)}=\min_{iM\le j\le (i+۱)M-۱}\omega_j$ و ω_j در لم ۲ تعریف شده است.

برهان. اثبات این لم شبیه اثبات لم ۶ در [۲۱] است که در آن برای هر i، $\phi_i=۰$ است. ■

لم ٦. تحت فرضیات $A۱$ و $A۲$، برای دنباله‌ی نامتناهی $\{x_k\}_{k\in P۲}$ تولید شده توسط الگوریتم ۱، داریم:

$$\lim_{i\to\infty}\omega_{s(i)}=۰, \quad (۲۳)$$

که در آن $\omega_{s(i)}$ در لم ۵ تعریف شده است.

برهان. با استفاده از لم‌های ۳ و ۵، برای $۱\le r\le M$ و $LM+r\in P۲$ داریم:

$$\sum_{i=۰}^{L}\omega_{s(i)}\le |f_۰|-f_{LM+r}\le |f_۰|-f_{LM+r}. \quad (۲۴)$$

با حدگیری از طرفین این نامساوی، وقتی $L\to\infty$، و هم‌چنین با استفاده از فرض $A۱$، نتیجه می‌گیریم $\sum_{i=۰}^{\infty}\omega_{s(i)}<\infty$. بنابراین:

$$\lim_{i\to\infty}\omega_{s(i)}=۰, \quad (۲۵)$$

که این اثبات را تمام می‌کند. ■

حال، مجموعه اندیس K را به صورت زیر تعریف می‌کنیم:

$$K=\{s(i)\mid i=۱,۲,\ldots\}.$$

قضیه‌ی زیر همگرایی سراسری الگوریتم ۱ را نشان می‌دهد.

قضیه‌ی ۷. فرض کنید $\{x_k\}$ دنباله‌ی تولید شده توسط الگوریتم ۱ بوده و فرضیات $A۱-A۳$ برقرار باشند. در این صورت، برای هر نقطه‌ی حدی x^* از دنباله $\{x_k\}_{k\in K}$، داریم:

$$\langle J^T(x^*)F(x^*),F(x^*)\rangle=۰, \quad (۲۶)$$

که در آن، $\langle .,.\rangle$ ضرب داخلی دو بردار در $\mathbb{R}^n$ است.

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

برهان. از لم۶ داریم:

$$\lim_{k\in K,k\to\infty}\omega_k=\lim_{k\in K,k\to\infty}\min_{kM\le j\le(k+1)M-1}\gamma\alpha_k^2\|F(x_k)\|^2=0.$$

در نتیجه، با توجه به این که $0<\gamma<1$ است، پس

$$\lim_{k\in K,k\to\infty}\alpha_k^2\|F(x_k)\|^2=0. \qquad (۲۷)$$

چون x^* یک نقطه‌ی حدی $\{x_k\}_{k\in K}$ است، پس مجموعه اندیس $K_1\subset K$ وجود دارد به طوری‌که $\lim_{k\in K_1,k\to\infty}x_k=x^*$.

از طرفی، از (۲۷) داریم:

$$\lim_{k\in K_1,k\to\infty}\alpha_k^2\|F(x_k)\|^2=0. \qquad (۲۸)$$

بنابراین، دو حالت ممکن زیر را داریم:

حالت ۱: $\lim_{k\to\infty,k\in K_1}\alpha_k\neq 0$

در این حالت، مجموعه‌ی نامتناهی $K_2\subset K_1$ و ثابت c وجود دارند به طوری‌که، برای هر $k\in K_2$، $\alpha_k\ge c>0$. بنابراین، (۲۷) نتیجه می‌دهد که

$$\lim_{k\in K_2}\|F(x_k)\|^2=0. \qquad (۲۹)$$

حال با استفاده از (۲۹) و پیوستگی F داریم: $F(x^*)=0$.

حالت ۲: $\lim_{k\in K_1,k\to\infty}x_k=x^*$

در این حالت $k_0\in K_1$ وجود دارد به طوری‌که، برای هر $k_0\le k\in K_1$، $\alpha_k<1$ است. بنابراین، برای تعداد نامتناهی تکرار، جستجوی خطی در اولین مرحله‌ی اجرا موفق نخواهد بود و حداقل یک‌بار مقادیر α_+ و α_- در گام ۴ الگوریتم ۱ به هنگام می‌شوند. فرض کنید در گام $k\in K_1$، مقادیر α_+ و α_- به تعداد m_k بار در گام های ۳ و ۴ الگوریتم ۱ به هنگام شوند. فرض کنید α_k^+ و α_k^- به ترتیب مقادیری باشند که متغیرهای α_+ و α_- در آخرین به هنگام ناموفق می‌پذیرند. به عبارت دیگر، نامساوی‌های (۱۲) و (۱۳) به ازای این مقادیر برقرار نمی‌باشند. حال با توجه به گام ۴ الگوریتم ۱، برای هر $k_0\le k\in K_1$، داریم:

$$\alpha_k\ge\tau_{\min}^{m_k}. \qquad (۳۰)$$

در نتیجه، با توجه به این‌که $\lim_{k\in K_1}\alpha_k=0$ و $\tau_{\min}<1$، خواهیم داشت:

$$\lim_{k\to\infty,k\in K_1}m_k=\infty.$$

به بیان دیگر، گام ۴ الگوریتم ۱ نتیجه می‌دهد که $\alpha_k^+\le\tau_{\max}^{m_k-1}$ و $\alpha_k^-\le\tau_{\max}^{m_k-1}$. بنابراین، با استفاده از این حقیقت که $\tau_{\max}<1$، نتیجه می‌گیریم:

$$\lim_{k\in K_1}\alpha_k^+=\lim_{k\in K_1}\alpha_k^-=0.$$

حال، با استفاده از تعریف α_k^+ و α_k^-، برای هر $k_0\le k\in K_1$، داریم:

$$f(x_k-\alpha_k^+\sigma_kF(x_k))>R_k-\gamma(\alpha_k^+)^2f(x_k), \qquad (۳۱)$$

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

$$f(x_k + \alpha_k^- \sigma_k F(x_k)) > R_k - \gamma(\alpha_k^-)^2 f(x_k). \quad (۳۲)$$

با استفاده از لم ۲.۲ در [۶] داریم: $R_k > f(x_k)$. بنابراین، از (۳۱) خواهیم داشت:

$$f(x_k - \alpha_k^+ \sigma_k F(x_k)) - f(x_k) > -\gamma(\alpha_k^+)^2 f(x_k).$$

از طرفی، لم ۳ و فرضیات $A1$ و $A2$ ایجاب می‌کنند که دنباله $\{f(x_k)\}$ از بالا کراندار است. بنابراین،

$$f(x_k - \alpha_k^+ \sigma_k F(x_k)) - f(x_k) > -\gamma(\alpha_k^+)^2 f(x_0),$$

نتیجه می‌دهد که:

$$\left\|F(x_k - \alpha_k^+ \sigma_k F(x_k))\right\|^2 - \left\|F(x_k)\right\|^2 > -\gamma(\alpha_k^+)^2 f(x_0),$$

و در نتیجه:

$$\frac{\left\|F(x_k - \alpha_k^+ \sigma_k F(x_k))\right\|^2 - \left\|F(x_k)\right\|^2}{\alpha_k^+} > -\gamma\alpha_k^+ f(x_0).$$

حال با استفاده از قضیه مقدار میانگین، $\xi_k \in [0,1]$ وجود دارد به طوری‌که

$$\sigma_k \langle g(x_k - \xi_k \alpha_k^+ F(x_k)), -F(x_k))\rangle \geq -f(x_0)\gamma\alpha_k^+. \quad (۳۳)$$

با توجه به الگوریتم ۱، دو حالت برای σ_k رخ خواهد داد: اگر برای تعداد نامتناهی k، $\sigma_k > 0$ باشد، آن‌گاه (۳۳) نتیجه می‌دهد که:

$$\langle g(x_k - \xi_k \alpha_k^+ F(x_k)), F(x_k))\rangle \leq f(x_0)\frac{\gamma\alpha_k^+}{\sigma_k} \leq f(x_0)\frac{\gamma\alpha_k^+}{\sigma_{\min}}. \quad (۳۴)$$

با یک استدلال مشابه برای (۱۳)، $\xi_k' \in [0,1]$ وجود دارد به طوری‌که:

$$\langle g(x_k + \xi_k' \alpha_k^- F(x_k)), F(x_k))\rangle \geq -f(x_0)\frac{\gamma\alpha_k^-}{\sigma_k} \geq -f(x_0)\frac{\gamma\alpha_k^-}{\sigma_{\min}}. \quad (۳۵)$$

حال، با توجه به این که $\alpha_k^+ \to 0$ ، $\alpha_k^- \to 0$ و $\left\|\sigma_k F(x_k)\right\|$ کران‌دار است، با گرفتن حد از دو طرف (۳۴) و (۳۵) خواهیم داشت:

$$\langle J^T(x^*)F(x^*), F(x^*)\rangle = 0. \quad (۳۶)$$

برای حالتی‌که برای نامتناهی k، داشته باشیم $\sigma_k < 0$، با استدلالی مشابه می‌توان به (۳۶) رسید. ■

یک نتیجه‌ی بلافصل از قضیه‌ی ۷ به صورت زیر است:

نتیجه‌ی ۸. فرض کنید x^* یک نقطه‌ی حدی دنباله‌ی $\{x_k\}_{k\in K}$ تولید شده توسط الگوریتم ۱ باشد و برای هر $0 \neq v \in \mathbb{R}^n$، $\langle J(x^*)v, v\rangle \neq 0$. در این صورت، $F(x^*) = 0$.

نکته‌ی ۹. برای $k \in \mathbb{N}$، فرض کنید:

$$L_k := f(x_{\ell(kM)}) = f_{\ell(kM)},$$

که در آن $\ell(kM) \in \{(k-1)M+1, \ldots, kM\}$ و $f_{\ell(j)}$ به صورت (۵) تعریف شده است. در این صورت داریم:

$$L_{k+1} := f(x_{\ell((k+1)M)}) \leq L_k.$$

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

نکته‌ی ۱۰. مجموعه‌ی اندیس K_+ را به صورت زیر در نظر بگیرید:

$$K_+ = \{\ell(1), \ell(2), ...\}.$$

به سادگی می‌توان نشان داد که:

$$\ell(j+1) \le \ell(j) + 2M - 1, \quad j = 1, 2, ... \tag{۳۷}$$

قضیه‌ی زیر نشان می‌دهد، در صورتی‌که یک نقطه‌ی حدی برای دنباله‌ی تولید شده توسط الگوریتم ۱ وجود داشته باشد، آن‌گاه همه‌ی نقاط حدی دنباله جواب دستگاه (۱) هستند.

قضیه‌ی ۱۱. فرض کنید x^* یک نقطه‌ی حدی دنباله‌ی $\{x_k\}$ تولید شده توسط الگوریتم ۱ با $F(x^*) = 0$ باشد و فرضیات $A1-A3$ برقرار باشند. در این صورت داریم:

$$\lim_{k\to\infty} F(x_k) = 0.$$

برهان. فرض کنید K_1 زیرمجموعه‌ای نامتناهی از $\mathbb{N}$ باشد به طوری‌که

$$\lim_{k\in K_1, k\to\infty} x_k = x^*.$$

این رابطه با در نظر گرفتن فرض $A1$ نتیجه می‌دهد:

$$\lim_{k\in K_1, k\to\infty} F(x_k) = 0. \tag{۳۸}$$

حال با استفاده از (۳۸) و این‌که $x_{k+1} = x_k \pm \alpha_k \sigma_k F(x_k)$ و برای هر $k \in \mathbb{N}$، $|\alpha_k \sigma_k| < \sigma_{\max}$، داریم:

$$\lim_{k\in K_1, k\to\infty} \|x_{k+1} - x_k\| = 0,$$

و در نتیجه :

$$\lim_{k\in K_1, k\to\infty} x_{k+1} = x^*.$$

با فرآیندی مشابه و به استقرا می‌توان دید که برای مقدار ثابت $j \in \{0, 1, ..., 2M-1\}$، داریم:

$$\lim_{k\in K_1, k\to\infty} x_{k+j} = x^*. \tag{۳۹}$$

حال با استفاده از (۳۷)، برای $k \in K_1$ داده شده، $\mu(k) \in \{0, 1, ..., 2M-1\}$ وجود دارد به طوری‌که $k + \mu(k) \in K_+$. هم‌چنین، چون مجموعه‌ی K_+ نامتناهی است، $\bar{i} \in \{0, 1, ..., 2M-1\}$ وجود دارد به طوری‌که برای تعداد نامتناهی $k \in K_1$ داریم: $\mu(k) = \bar{i}$.

فرض کنید:

$$K_2 = \{k + \mu(k) \mid k \in K_1 \quad \text{and} \quad \mu(k) = \bar{i}\}.$$

به سادگی می‌توان نشان داد $K_2 \subset K_+$. با استفاده از (۳۹) داریم:

$$\lim_{k\in K_2, k\to\infty} x_{k+1} = x^*.$$

بنابراین:

$$\lim_{k\in K_2, k\to\infty} F(x_k) = 0.$$

چون $K_2 \subset K_+$، زیر دنباله نامتناهی مانند $\{x_{\ell(j)}\}_{j\in J}$ وجود دارد، به طوری‌که:

$$\lim_{j\in J} x_{\ell(j)} = x^*, \qquad \lim_{j\in J} f(x_{\ell(j)}) = \lim_{j\in J} L_j = 0. \tag{۴۰}$$

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

حال، فرض کنید $J=\{j_1,j_2,j_3,\ldots\}$ با $\ldots<j_3<j_2<j_1$. از (۴۰) خواهیم داشت:

$$\lim_{i\to\infty} L_{j_i}=0. \qquad (41)$$

با استفاده از نکته‌ی ۹ داریم:

$$L_p\le L_{j_i}\quad \forall p\in\mathbb{N}, p>j_i.$$

بنابراین :

$$\sup_{p\ge j_i} L_p\le L_{j_i}. \qquad (42)$$

حال، با استفاده از (۴۱) و (۴۲) می‌توان نتیجه گرفت:

$$\lim_{i\to\infty}\sup_{p\ge j_i} L_p=0,$$

که به نوبه خود نتیجه می‌دهد:

$$\lim_{j\to\infty} L_j=0.$$

در نهایت، با استفاده از تعریف L_j داریم:

$$\lim_{k\to\infty} f(x_k)=\lim_{k\to\infty}\frac{1}{2}\|F(x_k)\|=0.$$

این عبارت اثبات را تمام می‌کند. ■

۵ نتایج عددی

در این بخش عملکرد عددی الگوریتم ۱، با نماد DF-DFSANE، را به همراه الگوریتم‌های زیر مورد بررسی و مقایسه قرار می‌دهیم:

- DFSANE: الگوریتم ۱ در [۹].
- NF-DFSANE: الگوریتم ۱ در این مقاله بدون مفهوم فیلتر.

تمامی الگوریتم‌ها در محیط MATLAB نسخه‌ی ۷.۱۰.۰ بر روی کامپیوتری با پردازنده‌ی اینتل ۲.۳۳ GHz Quad و حافظه‌ی رم ۴ گیگابایت پیاده‌سازی و مسایل آزمونی از مراجع [۲۲و۲۳] گرفته شده‌اند. ابعاد مسایل آزمونی بین ۲ تا ۱۰۰۰۰ متغیر است.

شرط خروج برای کلیه الگوریتم‌ها $\|F(x_k)\|\le 10^{-6}$ در نظر گرفته شده است. هم‌چنین، حداکثر تعداد تکرارها و محاسبه‌ی مقدار تابع را به ترتیب ۱۰۰۰۰ و ۵۰۰۰۰ در نظر گرفتیم. برای مقایسه‌ی بهتر کارایی الگوریتم‌ها از نمودار دولان-موره [۲۴] استفاده کرده‌ایم. مقادیر زیر برای پارامترهای الگوریتم‌ها در نظر گرفته شده است:

$$\epsilon=10^{-6},\mu_1=0/25,\mu_2=0/75,\sigma_{\min}=10^{-6},\sigma_{\max}=10^{6},M=20,$$
$$\gamma=10^{-4},\tau_{\min}=0/1,\tau_{\max}=0/5,\eta_k=\frac{1}{(1+k)^2},k\ge 0.$$

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

نتایج عددی در جدول ۱ گزارش شده‌اند. در این جدول، *Problem*، n، n_i و n_f به ترتیب مشخص کننده، نام مساله، بعد مساله، تعداد تکرارها و تعداد محاسبه‌ی مقدار تابع هستند. بر اساس اطلاعات این جدول، نمودارهای عملکرد بر اساس n_i و n_f به ترتیب در شکل‌های ۱ و ۲ ترسیم شده‌اند.

در شکل ۱ می‌توان دید که الگوریتم DF-DFSANE حدود ۹۳ درصد مسایل را با موفقیت حل می‌کند در صورتی‌که این درصد برای الگوریتم‌های NF-DFSANE و DFSANE کم‌تر از ۹۰ درصد است. هم‌چنین، الگوریتم DF-DFSANE حدود ۶۲ درصد مسایل را در کم‌ترین مقدار n_i حل می‌کند، در صورتی‌که این درصد برای الگوریتم‌های NF-DFSANE و DFSANE به ترتیب ۴۰ و ۱۰ درصد است. شکل ۲ نشان می‌دهد که وقتی هر سه الگوریتم روی مجموعه مسایل آزمایشی اجرا می‌شود، الگوریتم DF-DFSANE حدود ۴۹ درصد مسایل را در کم‌ترین میزان محاسبه‌ی مقدار تابع حل می‌کند، در صورتی‌که این درصد برای الگوریتم‌های NF-DFSANE و DFSANE به ترتیب ۳۸ و ۱۰ درصد است. علاوه بر این، الگوریتم DF-DFSANE حدود ۹۰ درصد مسایل را بدون شکست حل می‌کند. لازم به ذکر است که در مسایلی که الگوریتم DF-DFSANE بهترین الگوریتم برای حل نیست، نتایج حاکی از آن است که کارایی آن نزدیک به بهترین الگوریتم است.

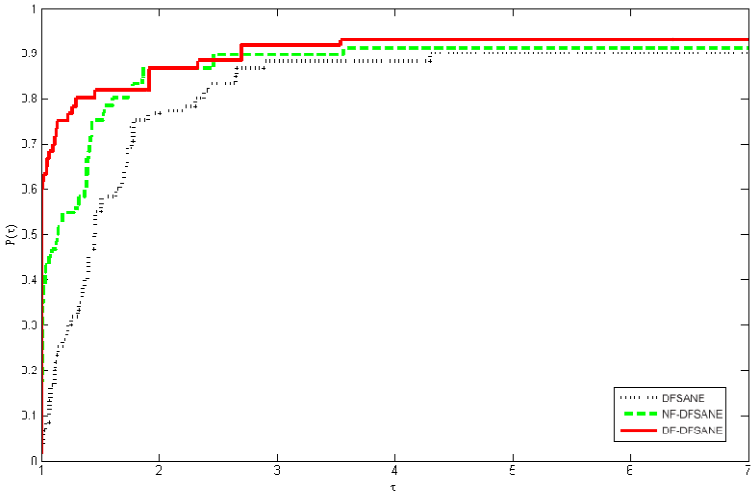
جدول ۱. نتایج عددی اجرای الگوریتم‌ها بر روی مسایل آزمون

Problem	n	**DFSANE** n_i / n_f	**NF-DFSANE** n_i / n_f	**DF-DFSANE** n_i / n_f
Extended Beale [22]	۱۰۰۰	۳۸/۵۲	۳۳/۴۶	۲۹/۳۹
Extended Beale [22]	۵۰۰۰	۵۳/۷۹	۴۷/۵۷	۳۸/۵۲
Extended Beale [22]	۱۰۰۰۰	۶۲/۹۰	۴۸/۷۲	۳۸/۶۴
Extended penalty[22]	۱۰۰۰	۱۷/۳۹	۱۳/۳۵	۱۰/۲۴
Extended penalty[22]	۵۰۰۰	۲۱/۴۹	۱۷/۴۵	۲۱/۴۷
Extended penalty[22]	۱۰۰۰۰	۲۳/۴۹	۲۱/۵۱	۲۲/۵۰
Extended Three Exponential [22]	۱۰۰۰	۱۸/۳۲	۱۶/۲۴	۱۵/۱۹
Extended Three Exponential [22]	۵۰۰۰	۲۹/۴۳	۱۸/۳۲	۱۴/۳۰
Extended Three Exponential [22]	۱۰۰۰۰	۳۰/۶۲	۲۰/۳۴	۲۰/۳۲
Generalized Tridiagonal-2 [22]	۱۰۰۰	۵۸/۶۸	۵۸/۷۲	۵۸/۷۲
Generalized Tridiagonal-2 [22]	۵۰۰۰	Failed	۱۹۳/۲۹۶	۱۲۶/۲۵۸
Generalized Tridiagonal-2 [22]	۱۰۰۰۰	Failed	۷۸۹/۳۲۹۹	۴۴۰/۱۰۸۳
Extended PSC1 Function [22]	۱۰۰۰	۱۸/۲۶	۱۴/۲۱	۱۴/۲۰
Extended PSC1 Function [22]	۵۰۰۰	۲۲/۴۲	۱۸/۲۶	۱۷/۲۳
Extended PSC1 Function [22]	۱۰۰۰۰	۳۶/۴۲	۲۲/۳۴	۱۷/۲۳
Extended Block Diagonal BD1 [22]	۱۰۰۰	۱۷/۲۳	۱۷/۲۳	۱۵/۲۳
Extended Block Diagonal BD1 [22]	۵۰۰۰	۱۹/۲۵	۱۵/۲۳	۲۰/۲۴
Extended Block Diagonal BD1 [22]	۱۰۰۰۰	۹۹/۱۳۱	۱۱۴/۲۱۳	۱۰۵/۲۸۵
DQDRTIC (CUTE) [22]	۱۰۰۰	۶۴/۱۱۶	۵۲/۷۴	۵۰/۷۰
DQDRTIC (CUTE) [22]	۵۰۰۰	۸۹/۲۰۹	۶۰/۹۰	۳۵/۶۱
DQDRTIC (CUTE) [22]	۱۰۰۰۰	۱۲۶/۳۰۱	۴۴/۶۶	۳۳/۴۷
LIARWHD function [22]	۱۰۰۰	۱/۱	۱/۱	۱/۱
LIARWHD function [22]	۵۰۰۰	۱۸/۹۰	۱۸/۹۰	۱۶/۷۴
LIARWHD function [22]	۱۰۰۰۰	۸۷/۲۳۳	۱۱۱/۴۰۱	۱۰۵/۳۴۱
Extended DENSCHNF [22]	۱۰۰۰	۱/۱	۱/۱	۱/۱
Extended DENSCHNF [22]	۵۰۰۰	۲۵/۴۱	۲۱/۴۹	۲۱/۴۹
Extended DENSCHNF [22]	۱۰۰۰۰	۵۲/۷۶	۲۵/۴۱	۲۲/۴۰
Generalized Quartic [22]	۱۰۰۰	۱۷/۲۳	۱۰/۱۲	۱۰/۱۰
Generalized Quartic [22]	۵۰۰۰	۲۰/۲۸	۱۰/۱۲	۲/۴
Generalized Quartic [22]	۱۰۰۰۰	۵۰/۹۸	۲۱/۲۹	۱۲/۱۶
Diagonal 8 [22]	۱۰۰۰	۸/۹	۶/۷	۵/۶
Diagonal 8 [22]	۵۰۰۰	۲۰/۳۷	۱۶/۱۷	۷/۹
Diagonal 8 [22]	۱۰۰۰۰	۳۸/۹۵	۵/۳۸	۷/۹

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

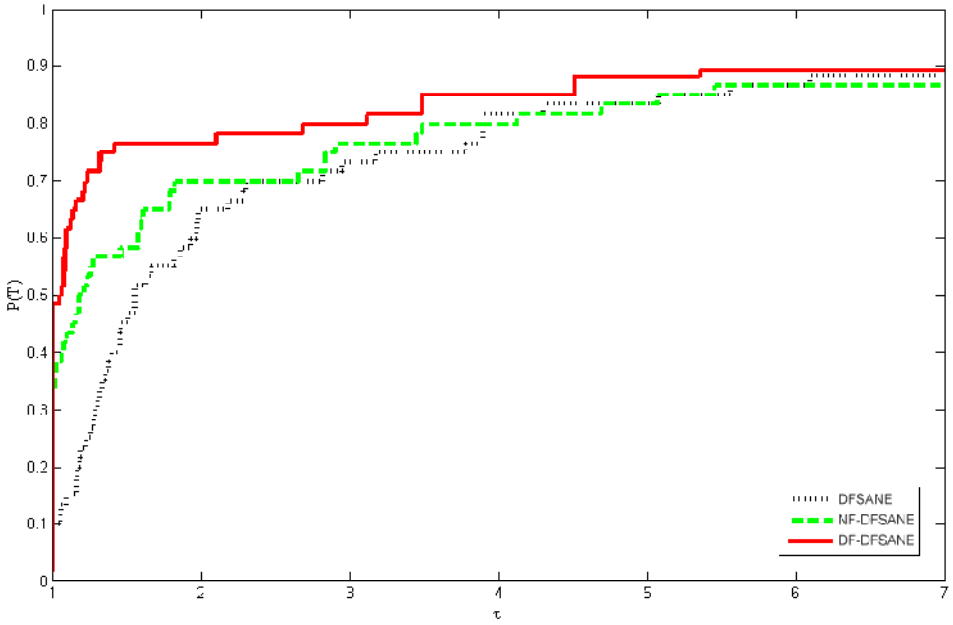
Problem	*n*	**DFSANE** n_i / n_f	**NF-DFSANE** n_i / n_f	**DF-DFSANE** n_i / n_f
Full Hessian FH3 [22]	۱۰۰۰	۴/۲۴	۴/۲۴	۴/۲۲
Full Hessian FH3 [22]	۵۰۰۰	۵/۳۱	۴/۳۰	۴/۲۸
Full Hessian FH3 [22]	۱۰۰۰۰	Failed	Failed	Failed
SINCOS [22]	۱۰۰۰	۱۸/۲۶	۱۴/۲۱	۱۱/۱۶
SINCOS [22]	۵۰۰۰	۲۷/۳۳	۱۸/۲۶	۱۷/۲۳
SINCOS [22]	۱۰۰۰۰	۳۶/۴۲	۳۰/۳۶	۱۷/۲۳
HIMMELH (CUTE) [22]	۱۰۰۰	۱/۱	۱/۱	۱/۱
HIMMELH (CUTE) [22]	۵۰۰۰	۱۴/۱۸	۱۴/۱۸	۸/۱۰
HIMMELH (CUTE) [22]	۱۰۰۰۰	Failed	Failed	Failed
Power [22]	۱۰۰۰	۷۵/۹۱	۷۴/۹۰	۵۴/۶۲
Power [22]	۵۰۰۰	۱۵۲/۱۷۶	۱۲۳/۱۴۲	۱۲۰/۱۳۴
Power [22]	۱۰۰۰۰	۲۲۴/۳۴۵	۲۱۰/۳۱۱	۱۲۹/۱۳۵
FLETCHCR function (CUTE) [22]	۱۰۰۰	۷۵۰/۱۴۰۴	۷۱۲/۱۲۰۱	۴۲۵/۶۸۹
FLETCHCR function (CUTE) [22]	۵۰۰۰	Failed	۱۰۱۲/۴۸۶۵	۹۹۴/۴۷۵۲
FLETCHCR function (CUTE) [22]	۱۰۰۰۰	Failed	Failed	Failed
Problem 2 [23]	۲	۴۶/۴۹	۲۵/۳۱	۲/۶
problem 4 [23]	۵	۱/۲	۱/۲	۱/۲
Problem 6 [32]	۲	۳/۷	۴/۱۵	۳/۶
Problem 7 [23]	۵	۲۶۸/۲۷۱	۶۲/۷۴	۳۳/۴۲
Problem 8 [23]	۱۰	۱۵/۱۸	۱۹/۲۵	۱۴/۱۶
Problem 26 [23]	۵	۱۴۹/۱۵۷	۲۱/۲۱	۲۱/۲۱
Problem 27 [23]	۲	۱/۱	۱/۱	۱/۱
Problem 39 [23]	۵	۱۳۸۳/۲۱۰۰	۲۴/۲۸	۲۰/۳۴
Problem 40 [23]	۵	Failed	Failed	Failed
Problem 42 [23]	۲	۱/۱	۱/۱	۱/۱
Problem 46 [23]	۵	Failed	Failed	Failed
Problem 47 [23]	۱۰	۳۲۴/۸۴۱	۲۶۰/۶۴۲	۲۲۲/۴۲۳
Problem 48 [23]	۵	۴۲۳/۱۱۰۸	۱۱۲/۵۰۴	۵۳/۷۱
Problem 53 [23]	۵	۱/۱	۱/۱	۱/۱
Problem 56 [23]	۱۰	۲۲/۲۴	۳/۵	۳/۳
Problem 61 [23]	۱۰	۳۹۴/۵۷۸۴	۳۵۷/۵۶۵۰	۳۵۷/۵۶۵۰
Problem 63 [23]	۵	۲۲/۲۶	۴۷/۶۶	۲۱/۳۸
Problem 77 [23]	۱۰	۱۴۹/۲۱۹	۵۱/۱۲۶	۱۷/۲۴
Problem 78 [23]	۲	Failed	Failed	Failed
Problem 79 [23]	۵	۹۷۰/۴۱۳۰	۹۵۲/۴۰۲۵	۸۴۴/۳۴۴۱
Problem 81 [23]	۵	Failed	Failed	Failed
Problem 81 [23]	۱۰	Failed	۸۴۳/۴۳۴۴	۴۳۲/۳۴۵۶
problem 111 [23]	۵	۳۴۴/۱۸۱۵	۳۱۰/۱۰۱۲	۷۲/۷۷۰
problem 111 [23]	۱۰	Failed	۵۷۲/۴۵۶۷	۴۲۴/۳۲۴۵



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

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]



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

٦ نتیجه‌گیری

در این مقاله یک الگوریتم DFSANE فیلتردار برای حل دستگاه معادلات غیرخطی با مقیاس بزرگ معرفی کردیم. در این الگوریتم بر توسیعی از روش DFSANE مفهوم فیلتر پیاده‌سازی شده است. با استفاده از این مفهوم بسیاری از نقاط آزمایشی به دست آمده در تکرارها که روش DFSANE آن‌ها را حذف می‌کرد، ذخیره شدند. الگوریتم جدید از اطلاعات نقاط موجود در فیلتر برای به‌دست آوردن نقاط بعدی بهره می‌برد و این سرعت همگرایی الگوریتم را افزایش می‌دهد. علاوه بر این، الگوریتم جدید مجهز به شرط نایکنوای جدیدی است که در [۸] ارایه شده است. 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 حاکی از کارایی این روش روی مسایل آزمونی در مقایسه با بقیه‌ی الگوریتم‌های در نظر گرفته شده است.

سپاسگزاری

نویسندگان این مقاله از امور پژوهشی دانشگاه صنعتی خواجه نصیرالدین طوسی و مرکز پژوهشی اسکوپ برای حمایت این پژوهش تقدیر و قدردانی می‌نمایند.

منابع

[۱] صابری نجفی، ه.، کردرستمی، س.، سهرابی گیلانی، ن، (۱۳۸۵). مقایسه روش‌های تکراری نیوتن، بریدن و شبه‌نیوتن برای حل دستگاه‌های. مجله تحقیق در عملیات درکاربردهای آن،۳(۸)، ۲۳-۳۷.

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]

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

[3] Cheng, W. Y., (2007). A two-term PRP based-descent method. Numerical Functional Analysis and Optimization, 28, 1217-1230.
[4] Nocedal, J., Wright, S. J., (2006). Numerical Optimization, Springer, New York.
[5] Grippo, L., Lampariello, F., Lucidi, S., (1991). A class of nonmonotone stabilization methods in unconstrained optimization. Numerische Mathematik, 59,779-805.
[6] Grippo, L., Sciandrone, M., (2002). Nonmontone globalization techniques for the Barzilai-Borwein gradient method. Computational Optimization and Applications, 23, 143-169.
[7] Ahookhosh, M., Amini, K., (2010). A nonmonotone trust region method with adaptive radius for unconstrained optimization. Computer and Mathematics with Applications, 60, 411-422.
[8] Ahookhosh, M., Amini, K., Peyghami, M. R., (2012). A nonmonotone trust-region line search method for large-scale unconstrained optimization. Applied Mathematical Modelling, 36, 478-487.
[9] La Cruz, W., Martinez, J. M., Raydan, M., (2006). Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. Mathematics of Computation, 75, 1429-1448.
[10] Li, D. H., Fukushima, M., (2000). A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. Optimization Methods and Software, 13, 181-201.
[11] Barzilai, J., Borwein, J.M., (1988). Two-point step size gradient methods. IMA Journal of Numerical Analysis, 8, 141-148.
[12] La Cruz, W., (2014). A projected derivative-free algorithm for nonlinear equations with convex constraints. Optimization Methods and Software, 29(1), 24-41.
[13] La Cruz, W., (2010). A residual method for solving nonlinear operator equations and their application to nonlinear integral equations using symbolic computation. Applied Mathematics and Computation, 17, 11-24.
[14] La Cruz, W., Raydan, M., (2003). Nonmonotone spectral methods for large-scale nonlinear systems. Optimization Methods and Software, 18, 583-599.
[15] Cheng, W., Li, D.H., (2009). A derivative-free nonmonotone line search and its application to the spectral residual method. IMA Journal of Numerical Analysis, 29, 814-825.
[16] Zhang, H., Hager, W. W., (2004). A nonmonotone line search technique and its application to unconstrained optimization. SIAM Journal on Optimization, 14(4), 1043-1056.
[17] Cheng, W., Chen, Z., (2013). Nonmonotone spectral method for large-scale symmetric nonlinear equations. Numerical Algorithms, 62, 149-162.
[18] Fletcher, R., Leyffer, S., Toint, Ph.L., (2006). A brief history of filter methods. SIAG/OPT Views and News, 18, 2-12.
[19] Fatemi, M., Mahdavi-Amiri, N., (2012). A filter trust-region algorithm for unconstrained optimization with strong global convergence properties. Computational Optimization and Applications, 52, 239-266.
[20] 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.
[21] Arzani, F., Peyghami, M. R., (2016). A new nonmonotone filter Barzilai-Borwein method for solving unconstrained optimization problems. International Journal of Computer Mathematics, 93(3), 596-608.
[22] Andrei, N., (2008). An unconstrained optimization test functions collection. Advanced Modeling and Optimization, 10, 147-161.
[23] Hock, W., Schittkowski, K., (1981). Test Examples for Nonlinear Programming Codes. Springer Series Lectures Notes in Economics Mathematical Systems.
[24] Dolan, E., More, J. J., (2002). Benchmarking optimization software with performance problems. Mathematical Programming, 91, 201-213.

[Downloaded from jamlu.lahijan.iau.ir on 2026-10-30]

[DOR: 20.1001.1.22517286.2016.13.2.6.6]