

Advanced Optimization

Final Project

Maedeh Zarvandi – Ali Izadi

810199179 - 810199102

۲	مقدمه
۳	روش Dai-Liao
۶	۲- روش ارائه شده و الگوریتم ها
۱۱	۳- اثبات همگرایی فراگیر
۱۶	پیاده سازی
۱۸	نتایج
۱۹	مقایسه الگوریتم ۱ و ۲
۲۲	ارزیابی با استفاده از performance profiles
۲۵	مراجع

مقدمه

در این مقاله هدف به دست آوردن الگوریتمی برای حل کردن دستگاه معادلات غیرخطی است، به شرطی که تابع $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ یک تابع یکنوا باشد.

$$F(x) = 0$$

تابع یکنوا نیز مطابق با زیر تعریف میشود:

$$(F(x) - F(y))^T (x - y) \geq 0 \quad \forall x, y \in \mathbb{R}^n$$

و هم چنین برای نقطه بهینه x^* داریم:

$$F(y) \cdot (y - x^*) \geq 0 \quad \forall y \in \mathbb{R}^n$$

دستگاه معادلات غیرخطی یکنوا، کاربردهای بسیار زیادی دارد. مانند زیرمسئله در الگوریتمهای پراکسیمال تعمیم یافته با فاصله‌های برگمن [۱]. برخی از مسائل نامساوی تغییرات یکنوا نیز می‌توانند به معادلات یکنوا غیرخطی تبدیل شوند [۲][۳]. کاربرد دیگری که معادلات یکنوا دارند استفاده در بازیابی سیگنال و تصویر است [۴]. به این ترتیب رویکردهای تکرارشونده برای حل معادلات یکنوا بسیار مورد توجه بوده‌است. متدهای مختلفی برای حل دستگاه معادلات غیرخطی به کار گرفته شده است. به عنوان مثال روشهای نیوتن و شبه نیوتن به دلیل نرخ همگرایی محلی بالایی که دارند همواره به این منظور جذاب بوده‌اند. با این حال برای حل کردن معادلات یکنوا مناسب نیستند زیرا نیاز به محاسبه ماتریس ژاکوبین و یا تقریبی از آن و همچنین حل دستگاه معادلات خطی در هر ایتريشن دارند. روش گرادیان طیفی [۵] کلاس دیگری از روشهای حل است که پیاده سازی آن بسیار آسان و برای مسائل بهینه سازی بدون قید در مقیاس بزرگ بسیار کارآمد است. به عنوان مثال لا کروز و رایدن [۶] روش گرادیان طیفی را برای حل دستگاه معادلات غیرخطی در مقیاس بزرگ به کار گرفتند. روشی که استفاده کرده بوند الگوریتم طیفی برای معادلات غیرخطی (SANE) نام داشت. همگرایی سراسری با استفاده از تغییر استراتژی غیریکنوا گریپو و همکاران [۷] تضمین شده است. در [۸]، لا کروز و همکاران. یک الگوریتم طیفی کاملاً بدون مشتق برای معادلات غیرخطی به نام (DF-SANE) ایجاد کرد و مثال‌های عددی ارائه شده نشان می‌دهد که الگوریتم (DF-SANE) برای کلاسی از سیستم معادلات غیرخطی به خوبی کار می‌کند. ژانگ و ژو [۹] یک الگوریتم گرادیان طیفی را با ترکیب روش گرادیان طیفی در [۱۰] با تکنیک تصویر کردن در [۱۱] توسعه دادند. همچنین اخیراً محمد و ابوبکر [۱۲] با استفاده از ترکیب محدب دو ضریب طیفی مختلف، فرم ترکیبی روش گرادیان طیفی و روش تصویر کردن را پیشنهاد کردند.

روش گرادیان مزدوج (CG) همچنین به دلیل سادگی و ذخیره‌سازی کم برای مسائل بهینه‌سازی بدون قید در مقیاس بزرگ کارآمد هستند [۱۳]. به عنوان مثال، در سال ۲۰۰۱، دای و لیائو [۱۴] دو روش گرادیان مزدوج را بر اساس شرایط مزدوج اصلاح شده ارائه شده توسط پری [۱۵] معرفی کردند. این روش بعداً توسط کافکی و قنبری [۵] به منظور دستیابی به روش گرادیان مزدوج نزولی اصلاح شد. به طور کلی، روش‌های گرادیان مزدوج در عمل بهتر از روش گرادیان طیفی عمل می‌کنند. با این حال، روش‌های گرادیان مزدوج برای معادلات غیرخطی در ادبیات نسبتاً نادر هستند زیرا اکثر آنها برای مسائل بهینه‌سازی بدون قید استفاده می‌شوند.

به این ترتیب برخی از محققان به این سمت رفتند که روش گرادیان مزدوج را با روش تصویر ابرصفحه برای معادلات غیرخطی یکنوا ترکیب کنند. به این منظور در [۱۶] چنگ یکی از روشهای معروف گرادیان مزدوج یعنی روش پلاک ریه ره [۱۷][۱۸] و تصویر

ابرفصفحه را ترکیب می‌کند. همچنین، پاپ و راپاجیچ چند جهت جدید از نوع فلچر-ریوز (FR) در [۱۹] چارچوب یک الگوریتم معرفی کردند که ترکیبی از رویکرد گرادیان مزدوج و تکنیک تصویر کردن ابرصفحه برای معادلات یکنوا غیرخطی مقیاس بزرگ است. علاوه بر این، یوان و ژانگ [۲۰] روش گرادیان مزدوج three-term را پیشنهاد کردند که در آن جهت دارای جهت نزول کافی همراه با تکنیک تصویر کردن برای معادلات غیر خطی در مقیاس بزرگ است. اخیراً در [۲۱]، فنگ و همکاران. معرفی یک خانواده از روش گرادیان مزدوج همراه با تکنیک طرح ریزی برای معادلات غیر خطی در مقیاس بزرگ.

در این مقاله یک روش نزولی گرادیان مزدوج برای حل دستگاه معادلات غیرخطی ارائه شده است. این روش یک اکستنشن از روش گرادیان مزدوجی است که در [۵] یعنی مقاله‌ای که توسط آقایان قنبری و کفاکی به عنوان یک خانواده از روشهای نزولی گرادیان مزدوج دی لیائو ارائه شده بود است. به این صورت که این ایده با روش تصویر کردن ابرصفحه، همانند رویکردهایی که پیش تر از آنها نام برده شده بود، ترکیب شده است.

روش Dai-Liao

در این مقاله هدف انتخاب یک اکستنشن از الگوریتمهای گرادیان مزدوج است به این صورت که جهت جستجوی d فرم ساده زیر را داشته باشد:

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

$$d_k = \begin{cases} -g_k, & \text{for } k = 1, \\ -g_k + \beta_k d_{k-1}, & \text{for } k \geq 2, \end{cases}$$

الگوریتمهای زیادی وجود دارند که هر کدام از آنها

فرمهای مختلفی برای پارامتر β پیشنهاد داده‌اند. برای مثال در روش (HS) زمانی که تابع فرم محدب کوادراتیک را داشته باشد به این صورت می‌شود:

$$f(x) = g^T x + \frac{1}{2} x^T H x,$$

$$\alpha_k = \arg \min_{\alpha > 0} f(x_k + \alpha d_k),$$

که با توجه به اینکه در روش گرادیان مزدوج شرایط مزدوج بودن برقرار خواهد بود داریم:

$$d_i^T H d_j = 0, \quad \forall i \neq j.$$

حال اگر داشته باشیم

$$y_{k-1} = g_k - g_{k-1}$$

با استفاده از نظریه مقدار میانگین خواهیم داشت

$$\alpha_{k-1}^{-1} d_k^T y_{k-1} = d_k^T \nabla^2 f(x_{k-1} + t \alpha_{k-1} d_{k-1}) d_{k-1},$$

می‌بینیم که سمت راست معادله به فرم رابطه بالای صفحه درآمده است. به این ترتیب می‌توانیم آن را با سمت چپ معادله جایگزین کرده و در رابطه اصلی گرادیان مزدوج، برای به دست آوردن جهت قرار دهیم. به این ترتیب برای β خواهیم داشت:

$$d_k^T y_{k-1} = 0.$$

$$\beta_k^{\text{HS}} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}.$$

این الگوریتم درواقع الگوریتم PRP را شبیه سازی می‌کند. به هر حال هر دوی این الگوریتمها همیشه کارآمد نیستند. زیرا شرایط مزدوجی که ارائه شد، به این وابسته هستند که جستجوی خط ما حتما دقیق باشد. به این ترتیب زمانیکه داشته باشیم:

$$g_{k+1}^T d_k \neq 0,$$

شرایط مزدوج بودن معایبی را به بار می‌آورد. برای مثال زمانیکه تابع ما فرم محدب و درجه دو دارد و می‌خواهیم آن را روی یک زیرفضایی که از span کردن مجموعه‌ای از بردارهایی که دو به دو نسبت به هم مزدوج هستند به دست آمده است، بهینه کنیم. فرض می‌کنیم که جستجوی خط در راستای d دقیق نباشد به این معنا که طول قدم بهینه نیست، به این ترتیب فارغ از اینکه چه جستجوی خطی انتخاب کرده باشیم در ایتريشن های آینده، همواره خطایی که در ایتريشن کنونی وجود دارد تا زمانیکه جهت‌های جستجوی آینده نسبت به جهت جستجوی کنونی مزدوج باشند، باقی خواهد ماند:

$$(x_{k+1} - x^*)^T H(x_{k+1} - x^*) \geq (\alpha_1 - \alpha_1^*)^2 d_1^T H d_1,$$

$$x^* = -H^{-1}g$$

به این ترتیب برای حل کردن این مشکل، محققان زیادی به سمت پیدا کردن راهی برای تغییر دادن جهت جستجو رفتند و جهت‌های متفاوتی را توسعه دادند. با این حل این مشکل در هیچ کدام از روشهای پیشنهادی به صورت کامل برطرف نشده است و در خیلی مواقع نیز نتیجه بهتری از روش PRP ندارد. بنابراین همانطور که گفته شد این مقاله، برخلاف پژوهش های دیگر سعی دارد بدون تغییر دادن فرم ساده جهت جستجو در روشهای گرادیان مزدوج، شرایط مزدوجی جدیدی ارائه دهد که با استفاده از

جستجوی دقیق و غیر دقیق کارا باشد. به علاوه براساس شرایط جدید مزدوج بودن، دو روش گرادیان مزدوج غیرخطی که با هر دو روش جستجوی خطی و غیرخطی کار می‌کند پیشنهاد می‌دهند که اثباتهای همگرایی نیز برای آنها وجود دارد.

به این ترتیب داریم

$$d_k = -B_k g_k,$$

$$B_k y_{k-1} = s_{k-1}, s_{k-1} = \alpha_{k-1} d_{k-1}$$

$$d_k^T y_{k-1} = -(B_k g_k)^T y_{k-1} = -g_k^T (B_k y_{k-1}) = -g_k^T s_{k-1}.$$

که نشان می‌دهد شرایط مزدوجی اولیه برای زمانیکه جستجوی خط دقیق باشد، برقرار است. اگرچه بیشتر مواقع شرایط مزدوجی برقرار نیست و به این ترتیب منطقی تر است که شرط زیر را به عنوان شرط مزدوجی در نظر بگیریم:

$$d_k^T y_{k-1} = -t g_k^T s_{k-1},$$

که در آن t یک اسکالر نامنفی است.

به این ترتیب برای اطمینان از اینکه جهت جستجو که در الگوریتم گرادیان مزدوج وجود دارد

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

$$d_k = \begin{cases} -g_k, & \text{for } k = 1, \\ -g_k + \beta_k d_{k-1}, & \text{for } k \geq 2, \end{cases}$$

شرط جدید را ارضا می‌کند، کافی است که شرط قبلی را در y_{k-1} ضرب نماییم و با استفاده از رابطه شرط مزدوجی جدید، برای β خواهیم داشت:

$$\beta_k = \frac{g_k^T (y_{k-1} - t s_{k-1})}{d_{k-1}^T y_{k-1}}.$$

که برابر است با

$$\beta_k = \beta_k^{\text{HS}} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}},$$

و نشان می‌دهند که یک انتخاب مناسب برای پارامتر t برابر است با:

$$t = \frac{g_k^T \bar{d}_k}{\tau_k},$$

$$\tau_k := \bar{\alpha}_k g_k^T \bar{d}_k + \frac{(g_k^T s_{k-1})^2}{s_{k-1}^T y_{k-1}} < 0,$$

به این ترتیب این یکی از روشهای گرادیان مزدوجی غیرخطی است که این مقاله ارائه کرده است.

در ادامه بیان می‌کند که با توجه به اینکه متد ارائه شده زمانی که داشته باشیم $g_k^T d_{k-1} = 0$ برابر خواهد با روش PRP، متد ارائه شده نیازی ندارد که برای تابع های جنرال همگرا شود. به این ترتیب همانطور که گیلبرت و نوسدال [۳] که همگرایی سراسری PRP را با قید $\beta_k^{PRP} \geq 0$ اثبات کردند، می‌توان متد ارائه شده را به صورت زیر درآورد:

$$\beta_k = \max \left\{ \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, 0 \right\} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}},$$

و اثبات می‌شود که این متد برای تابع های جنرال نیز اثبات همگرایی سراسری دارد. این دومین متدی است که این مقاله از روشهای گرادیان مزدوج غیرخطی پیشنهاد می‌کند.

۲- روش ارائه شده و الگوریتم ها

این روش ترکیبی از یکی از الگوریتم های conjugate gradient به نام Dai-liao و با تکنیک تصویر بر روی ابرصفحه (hyperplane projection technique) است.

نقطه‌ی جدید در هر مرحله از الگوریتم مطابق با زیر به دست می‌آید.

$$x_{k+1} = x_k + \alpha_k d_k$$

که در آن α_k طبق line search در رابطه زیر صدق کند و d_k جهت حرکت خواهد بود.

$$-F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha \|F(x_k + \alpha_k d_k)\| \|d_k\|^2$$

با توجه به یکنوا بودن تابع میتوان این فرآیند در جهت بهتری حرکت کرد. برای این کار $z_k = x_k + \alpha_k d_k$ بر روی ابر صفحه H_k تصویر میشود.

ابر صفحه H_k مطابق با زیر تعریف میشود:

$$H_k = \left\{ x_k \in \mathbb{R}^n \mid F(z_k)^T (x - z_k) = 0 \right\}$$

اثبات میشود که این ابر صفحه نقاط x_k را از مجموعه جواب های معادله غیرخطی جدا میکند. بنابراین تصویر نقاط x_k بر روی این صفحه باعث میشود تا فاصله نقاط x_k با مجموعه جواب به صورت یکنوا کاهش پیدا کند.

رابطه تصویر $z_k = x_k + \alpha_k d_k$ بر روی این صفحه مطابق با زیر به دست می آید:

$$z_k = x_k + \alpha_k d_k$$

$$x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k) F(z_k)}{\|F(z_k)\|^2}$$

• اثبات از مقاله [۲۱]

طبق line search گفته شده داریم:

$$F(z_k)^T (x_k - z_k) = F(z_k)^T (-\alpha_k d_k)$$

$$= -\alpha_k F(z_k)^T d_k$$

$$line\ search : \geq \alpha_k^2 \sigma \|d_k\|^2 = \sigma \|x_k - z_k\|^2$$

$$> 0$$

و هم چنین طبق خاصیت گفته شده در معرفی برای توابع یکنوا داریم:

$$F(z_k)^T (x^* - z_k) \leq 0$$

بنابراین طبق دو عبارت بالا نتیجه میگیریم که

$$[1] F(z_k)^T (x_k - z_k) > 0$$

$$[2] F(z_k)^T (x^* - z_k) \leq 0$$

بنابراین ابر صفحه $H_k = \left\{ x_k \in \mathbb{R}^n \mid F(z_k)^T (x - z_k) = 0 \right\}$ نقاط x_k را از مجموعه جواب های معادله غیرخطی جدا میکند.

الگوریتم ۱

در این الگوریتم جهت d به صورت زیر معرفی شده است که با کمی دقت متوجه می‌شویم که همان رابطه‌ای است که پیشتر در مقدمه در مورد جهت جستجو در روشهای گرادیان مزدوج معرفی شده بود:

$$d_k = \begin{cases} -F(x_k), & \text{if } k = 0, \\ -F(x_k) + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases}$$

با توجه به اینکه روش ارائه شده اکستنشنی از روش Dai-Liao می‌باشد، β در این روش همان پارامتر گرادیان مزدوج نزولی Dai-Liao است. بنابراین داریم:

$$\beta_k = \frac{F(x_k)^T y_{k-1}}{y_{k-1}^T d_{k-1}} - t_k \frac{F(x_k)^T s_{k-1}}{y_{k-1}^T d_{k-1}}, \quad t_k = p \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} - q \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2},$$

$$p \geq \frac{1}{4}, \quad q \leq 0, \quad y_k = F(x_{k+1}) - F(x_k), \quad s_k = z_k - x_k = \alpha_k d_k.$$

نکته‌ای که وجود دارد این است که در این رابطه پارامتر گرادیان مزدوج یک تعمیم از دو پارامتر گرادیان مزدوج معروف، برای حل معادلات غیرخطی است. به این ترتیب که اگر $p = 2, q = 0$ باشد، β_k به صورت پارامتری که در [۲۲] توسط هگر و ژانگ ارائه شده است تبدیل می‌شود:

$$\bar{\beta}_k^N = \max \{ \beta_k^N, \eta_k \}, \quad \text{where}$$

$$\eta_k = \frac{-1}{\|d_k\| \min \{ \eta, \|g_k\| \}},$$

$$\beta_k^N = \frac{1}{d_k^T y_k} \left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right)^T g_{k+1}.$$

علاوه بر این زمانی که $p = 1, q = 0$ باشد، پارامتر β تبدیل می‌شود به پارامتری که در [۲۳] توسط دای و کو ارائه شده است که هدفشان پیدا کردن جهت گرادیان مزدوجی است که تا جایی که امکان دارد به جهت BFGS بدون حافظه مقیاس داده شده نزدیک شود:

$$\beta_k(\tau_k) = \frac{g_{k+1}^T y_k}{d_k^T y_k} - \left(\tau_k + \frac{\|y_k\|^2}{s_k^T y_k} - \frac{s_k^T y_k}{\|s_k\|^2} \right) \frac{g_{k+1}^T s_k}{d_k^T y_k},$$

که با توجه به مقادیر مختلف τ به این نتیجه می‌رسند:

$$\beta_k^+(\tau_k) = \max \left\{ \beta_k(\tau_k), \eta \frac{g_{k+1}^T d_k}{\|d_k\|^2} \right\},$$

که بهینه ترین حالت آن برابر است با:

$$\tau_k = \frac{s_k^T y_k}{\|s_k\|^2},$$

در ادامه الگوریتم Dai-Liao Projection Method را به صورت شبه کد مشاهده می‌کنیم. ابتدا از یک مقدار اولیه x_0 شروع می‌کنیم و پارامترهای tolerance و سیگما را نیز متعاقباً انتخاب می‌کنیم. در گام اول مقدار تابع را در نقطه کنونی محاسبه می‌کنیم. اگر نرم مقدار تابع از ϵ کمتر باشد، متوقف می‌شویم و در غیر اینصورت به گام بعد می‌رویم. در این گام جهت را با استفاده از رابطه داده شده برای جهت در روش گرادینان مزدوج محاسبه می‌کنیم و اگر $d_k = 0$ باشد، متوقف می‌شویم. در گام بعدی با استفاده از یک الگوریتم جستجوی خط، طول قدم یا $\alpha = r^{m_k}$ را به دست می‌آوریم که در آن m_k کوچکترین عدد صحیح نامنفی است که در شرط زیر صدق می‌کند:

$$-F(x_k + r^m d_k)^T d_k \geq \sigma r^m \|F(x_k + r^m d_k)\| \|d_k\|^2.$$

در گام بعد، با استفاده از رابطه $z_k = x_k + \alpha_k d_k$ به دست می‌آوریم. به این ترتیب اگر $\|F(z_k)\| = 0$ باشد متوقف می‌شویم و در غیر اینصورت به گام بعدی می‌رویم. در این گام نقطه بعدی را با استفاده از رابطه projection در [۱۱] محاسبه می‌کنیم به این صورت:

$$x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k) F(z_k)}{\|F(z_k)\|^2}.$$

و به همین ترتیب الگوریتم وارد ایتريشن بعدی می‌شود.

Algorithm 1 Descent Dai-Liao projection method (DLPM)

Step 0. Given $x_0 \in \mathbb{R}^n$, $r, \sigma \in (0, 1)$, stopping tolerance $\epsilon > 0$, Set $k = 0$.

Step 1. Compute $F(x_k)$. If $\|F(x_k)\| \leq \epsilon$ stop, else go to **Step 2**.

Step 2. Compute d_k by (8). Stop if $d_k = 0$.

Step 3. Determine $\alpha_k = r^{m_k}$ with m_k being the smallest nonnegative integer m such that

$$-F(x_k + r^m d_k)^T d_k \geq \sigma r^m \|F(x_k + r^m d_k)\| \|d_k\|^2. \quad (9)$$

Step 4. Compute $z_k = x_k + \alpha_k d_k$. If $\|F(z_k)\| = 0$ stop, else go to **Step 5**.

Step 5. Compute x_{k+1} using (4).

Step 6. Let $k = k + 1$ and go to **Step 1**.

تنها تفاوتی که الگوریتم ۲ دارد این است که در این الگوریتم، روش به دست آوردن جهت جستجو d و پارامتر β و همچنین روش جستجوی خط ارائه شده برای به دست آوردن α متفاوت است که در شبه کدی که در ادامه آورده شده است، این مطلب دیده می شود.

Algorithm 2 A family of conjugate gradient method (FCG)

Step 0. Given an arbitrary initial point $x_0 \in \mathbb{R}^n$, parameters $0 < r < 1$, $\sigma > 0$, $t > 0$, $\rho > 0$, $\epsilon > 0$, and set $k := 0$.

Step 1. If $\|F(x_k)\| \leq \epsilon$, stop, otherwise go to **Step 2**.

Step 2. Compute

$$d_k = \begin{cases} -F(x_k), & \text{if } k = 0, \\ -(1 + \beta_k \frac{F(x_k)^T d_{k-1}}{\|F(x_k)\|^2}) F(x_k) + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases}$$

where β_k is such that

$$|\beta_k| = t \frac{\|F(x_k)\|}{\|d_{k-1}\|},$$

$\forall k \geq 1$ and $t > 0$.

Step 3. Find the trial point $y_k = x_k + \alpha_k d_k$, where $\alpha_k = \rho r^{m_k}$ and m_k is the smallest nonnegative integer m such that

$$-F(x_k + \rho r^m d_k)^T d_k \geq \sigma \rho r^m \|d_k\|.$$

Step 4. Compute x_{k+1} using (4).

Step 5. Let $k = k + 1$ and go to **Step 1**.

۳- اثبات همگرایی فراگیر

برای اثبات همگرایی به ۴ لم نیاز داریم.

لم اول:

جهت تعریف شده d_k طبق مقاله [۵] تعریف شده که در آن مقاله اثبات میشود یک جهت کاهشی است یعنی:

$$F(x_k)^T d_k = -\lambda_k \|F(x_k)\|^2 < 0, \quad \lambda_k > 0$$

ایده اثبات به این ترتیب است که جهت تعریف شده برابر است با جهت حرکت طبق یک روش شبه نیوتن.

سپس بر اساس شرط $s_k^T y_k > 0$ اثبات میشود که ماتریس عکس هسیان بر اساس مقادیر یک ماتریس معین مثبت بوده و در

نتیجه بر اساس مقادیر ویژه لامبدای مثبت آن رابطه بالا برقرار بوده و جهت d_k یک جهت کاهشی است.

- لازم به ذکر است که در مقاله ارائه شده در رابطه بالا و هم چنین در روابط محاسبه جهت d بر خلاف مقاله [۵] مقدار تابع یعنی $F(x_k)$ جایگزین گرادیان تابع شده است که به علت فرض یکنوا بودن F در این مقاله است و بنابراین روش به روش های بدون نیاز به محاسبه گرادیان تبدیل می شود.

لم دوم:

اگر x^* در رابطه $F(x^*) = 0$ صدق کند و $\{x_k\}$ توسط الگوریتم ۱ تولید شود آنگاه خواهیم داشت:

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \|x_{k+1} - x_k\|^2$$

ایده اثبات این قضیه با استفاده از مقاله [۱۱] بر اساس یک خاصیت عملگر تصویر (projection) به دست می آید.

این لم بیان میکند که دنباله $\{\|x_k - x^*\|\}$ یک دنباله غیرافزایشی و همگرا بوده و هم چنین دنباله $\{x_k\}$ یک دنباله محدود (bounded) بوده یعنی:

دنباله محدود:

$$\sum_{k=0}^{\infty} \|x_{k+1} - x_k\| < \infty$$

و دنباله همگرا:

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$$

لم سوم:

اگر $\{x_k\}$ دنباله تولید شده توسط الگوریتم ۱ باشد آنگاه خواهیم داشت:

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0$$

اثبات:

برای اثبات نرم تفاضل دو نقطه متوالی یعنی $\|x_{k+1} - x_k\|$ را با توجه به دو رابطه زیر یعنی نحوه محاسبه x_{k+1} و همچنین line search که در قبل آورده شد بسط میدهیم.

$$x_{k+1} = x_k - \frac{F(z_k)^T (x_k - z_k) F(z_k)}{\|F(z_k)\|^2} \quad [۱]$$

$$[۲] -F(x_k + \alpha_k d_k)^T d_k \geq \sigma \alpha \|F(x_k + \alpha_k d_k)\| \|d_k\|^2$$

$\|x_{k+1} - x_k\|$ را بسط می‌دهیم:

$$\begin{aligned}\|x_{k+1} - x_k\| &= \frac{|F(z_k)^T(x_k - z_k)| \|F(z_k)\|}{\|F(z_k)\|^2} \\ &= \frac{|\alpha_k F(z_k)^T d_k|}{\|F(z_k)\|} \\ &\geq \frac{\sigma \alpha_k^2 \|F(z_k)\| \|d_k\|^2}{\|F(z_k)\|} \\ &= \sigma \alpha_k^2 \|d_k\|^2 \geq 0\end{aligned}$$

در نتیجه طبق دو رابطه زیر:

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$$

$$\|x_{k+1} - x_k\| = \sigma \alpha_k^2 \|d_k\|^2 \geq 0$$

خواهیم داشت:

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0$$

لم چهارم:

هم چنین باید نشان داد که $\|d_k\|$ از بالا محدود است یعنی وجود دارد $M > 0$ به طوری که $\|d_k\| < M$

فرض می‌کنیم که تابع F از پیوسته لیپشیتز باشد و دنباله $\{x_k\}$ یک دنباله محدود باشد یعنی:

$$\begin{aligned}\|F(x) - F(y)\| &\leq L \|x - y\| \\ \exists \omega > 0 : \|F(x_k)\| &< \omega\end{aligned}$$

هم چنین طبق یکنوا بودن تابع داریم:

$$(F(x) - F(y))^T (x - y) \geq c \|x - y\|^2$$

و هم چنین طبق روابط داریم:

$$t_k = p \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} - \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}$$

آنگاه خواهیم داشت:

$$|t_k| \leq p \frac{L^2 \|s_{k-1}\|^2}{c \|s_{k-1}\|^2} + |q| \frac{L \|s_{k-1}\|^2}{\|s_{k-1}\|^2} = p \frac{L^2}{c} + L|q|.$$

در نتیجه با توجه به نامساوی به دست آمده در بالا و بسط رابطه جهت d خواهیم داشت:

$$\begin{aligned} \|d_k\| &= \|-F_k + \beta_k d_{k-1}\| \\ &= \|-F_k + \frac{F_k^T y_{k-1}}{s_{k-1}^T y_{k-1}} s_{k-1} - t_k \frac{F_k^T s_{k-1}}{s_{k-1}^T y_{k-1}} s_{k-1}\| \\ &\leq \|F_k\| + \frac{\|F_k\| \|y_{k-1}\|}{s_{k-1}^T y_{k-1}} \|s_{k-1}\| + |t_k| \frac{\|F_k\| \|y_{k-1}\|}{s_{k-1}^T y_{k-1}} \|s_{k-1}\| \\ &\leq \omega + \frac{L\omega \|s_{k-1}\|}{c \|s_{k-1}\|^2} \|s_{k-1}\| + (p \frac{L^2}{c} + L|q|) \frac{\omega \|s_{k-1}\|}{c \|s_{k-1}\|^2} \|s_{k-1}\| \\ &= \omega + \frac{L\omega}{c} + (p \frac{L^2}{c} + L|q|) \frac{\omega}{c}. \end{aligned}$$

که میتوان M را مطابق با زیر در نظر گرفت و لم چهارم اثبات میشود.

قضیه اول: اثبات همگرایی فراگیر

اگر $\{x_k\}$ دنباله تولید شده توسط الگوریتم ۱ باشد آنگاه خواهیم داشت:

$$\lim_{k \rightarrow \infty} \inf \|F(x_k)\| = 0$$

اثبات:

دو حالت را در نظر میگیریم:

$$\lim_{k \rightarrow \infty} \inf \|d_k\| = 0$$

حالت اول: اگر

در این حالت آنگاه بر اساس اثبات مقاله [۲۱] داریم: $\|F(x_k)\| < \|d_k\|$ و در نتیجه:

$$\lim_{k \rightarrow \infty} \inf \|F(x_k)\| = 0$$

با توجه به محدود بودن دنباله $\{x_k\}$ و هم چنین پیوستگی F دنباله $\{x_k\}$ یک نقطه حدی x^* خواهد داشت که $F(x^*) = 0$ و چون طبق لم دوم دنباله $\{\|x_k - x^*\|\}$ همگرا بوده بنابراین $\{x_k\}$ به صورت فراگیر به x^* همگرا میشود.

$$\liminf_{k \rightarrow \infty} \|d_k\| > 0 \quad \text{حالت دوم: اگر}$$

بنابراین مطابق با حالت قبل خواهیم داشت:

$$\liminf_{k \rightarrow \infty} \|F(x_k)\| > 0$$

و طبق [لم سوم](#) خواهیم داشت:

$$\lim_{k \rightarrow \infty} \alpha_k = 0$$

رابطه بالا زمانی درست است که نتوان α_k ی پیدا کرد که در شرط line search صدق کند یعنی

$$-F(x_k + \alpha_k d_k)^T d_k < \sigma \alpha \|F(x_k + \alpha_k d_k)\| \|d_k\|^2$$

و در نتیجه خواهیم داشت:

$$-F(x^*)^T \cdot d^* < 0$$

$$[1] F(x^*)^T \cdot d^* > 0$$

اما طبق [لم اول](#) داریم:

$$-F(x^*)^T \cdot d^* = \|F(x^*)\|^2$$

$$\liminf_{k \rightarrow \infty} \|F(x_k)\| > 0 \quad \text{و طبق نتیجه یعنی}$$

$$-F(x^*)^T \cdot d^* = \|F(x^*)\|^2 > 0$$

$$[2] F(x^*)^T \cdot d^* < 0$$

$$\liminf_{k \rightarrow \infty} \|F(x_k)\| > 0 \quad \text{که [1] و [2] با هم در تناقض بوده و بنابراین}$$

برقرار نیست.

بنابراین تنها حالت اول به درستی برقرار بوده و به معنای همگرایی فراگیر الگوریتم است.

پیاده سازی

برای الگوریتم DLPM پارامترهای زیر در نظر گرفته شده است.

$$\sigma = 0.01, \quad r = 0.6, \quad p = 0.8, \quad q = -0.1, \quad \epsilon = 10e - 6$$

و هم چنین برای الگوریتم FCG پارامترهای زیر در نظر گرفته شده است.

$$\sigma = 0.01, \quad r = 0.5, \quad \rho = 1, \quad t = 1, \quad \epsilon = 10e - 6$$

به منظور پیاده سازی ابتدا کتابخانه‌های مورد استفاده را ایمپورت می‌کنیم. در قسمت بعدی مسائلی که در مقاله آورده شده است و به منظور بهینه کردن انتخاب می‌شود، به صورت تابع های ۱F تا ۸F پیاده سازی شده است. یک تابع به نام feval به منظور به دست آوردن متریک Number of Function Evaluations پیاده سازی شده است.

یک تابع جستجوی خط نیز، به منظور پیاده سازی الگوریتمهای جستجوی خطی که برای هر کدام از الگوریتمهای DLPM و FCG وجود دارد، پیاده سازی شده است که کد آن در تصویر زیر مشخص است.

```

1 def line_search_DLPM(F, x_k, d_k, sigma, r, count):
2     alpha = 1
3     while True:
4         f_x_next = F(x_k + alpha * d_k)
5         count = feval(count)
6         if - f_x_next.T @ d_k >= sigma * alpha * np.linalg.norm(f_x_next) * (np.linalg.norm(d_k) ** 2):
7             break
8         alpha = alpha * r
9     return alpha, count

1 def line_search_FCG(F, x_k, d_k, sigma, r, rho, count):
2     alpha = 1
3     while True:
4         f_x_next = F(x_k + rho * alpha * d_k)
5         count = feval(count)
6         if - f_x_next.T @ d_k >= sigma * rho * alpha * (np.linalg.norm(d_k)**2):
7             break
8         alpha = alpha * r
9     return alpha, count
10

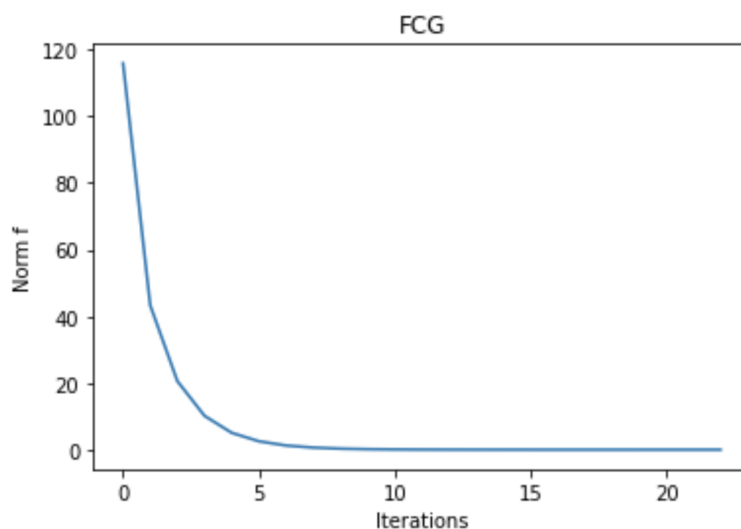
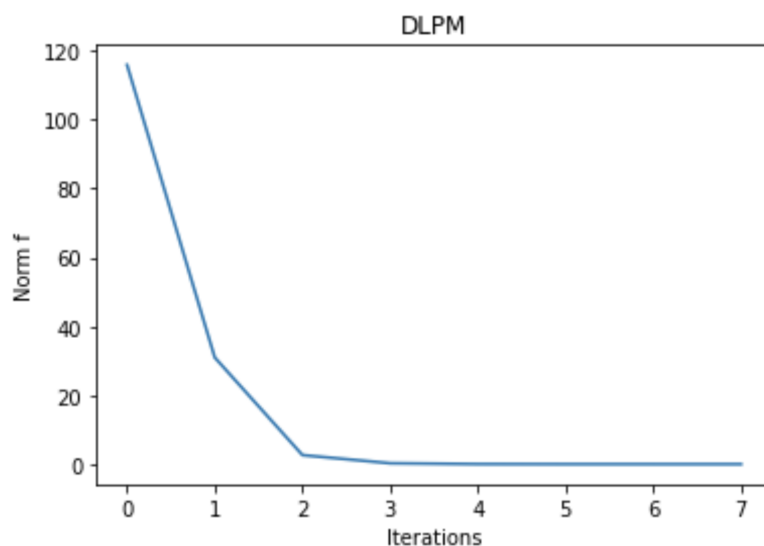
```

به همین ترتیب پیاده‌سازی الگوریتمهای DLMP و FCG نیز به صورت تابع هایی به همین نام وجود دارند که هر دو تابع F را که همان مسئله موردنظر برای بهینه سازی است، ورودی می‌گیرند.

به عنوان مثال در دو تصویر زیر، نمودار روند کاهش نرم مقدار تابع در مسئله اول با استفاده از هر دو الگوریتم دیده می‌شود. الگوریتم اول در ۷ ایتريشن و حدود ۰.۹۵ ثانیه و تعداد ۲۶ عدد feval و الگوریتم دوم در ۲۲ ایتريشن و حدود ۳.۲۹ ثانیه و تعداد ۹۳ عدد feval همگرا شده است.

در قسمت بعد نقاط اولیه مختلف را مشخص کرده و هر مسئله را با توجه به نقاط اولیه ای که در مقاله برای آن در نظر گرفته شده بود، اجرا کردیم و با استفاده از هر دو متود بهینه سازی را انجام دادیم.

در نهایت متریک های مختلف که در مقاله آورده شده یعنی **time** و تعداد ایتريشن و تعداد **function evaluation** و نرم تابع گزارش شده است. که اینها در قسمت نتایج به طور کامل آورده شده اند.



نتایج

برای تست ومقایسه الگوریتم ها ۸ مسئله زیر در نظر گرفته شده است:

Problem 1:

$$F_i(x) = 2x_i - \sin |x_i|, \quad i = 1, 2, 3, \dots, n.$$

Problem 2:

$$F_1(x) = 2x_1 + \sin(x_1) - 1,$$

$$F_i(x) = -2x_{i-1} + 2x_i + \sin(x_i) - 1, \quad \text{for } i = 2, 3, \dots, n-1,$$

$$F_n(x) = 2x_n + \sin(x_n) - 1.$$

Problem 3:

$$F_1(x) = x_1 - e^{\cos(h(x_1+x_2))}$$

$$F_i(x) = x_i - e^{\cos(h(x_{i-1}+x_i+x_{i+1}))} \quad \text{for } i = 2, 3, \dots, n-1$$

$$F_n(x) = x_n - e^{\cos(h(x_{n-1}+x_n))},$$

$$\text{where } h = \frac{1}{n+1}.$$

Problem 4:

$$F_i(x) = e^{x_i} - 1, \quad \text{for } i = 1, 2, 3, \dots, n.$$

Problem 5:

$$F_1(x) = 2.5x_1 + x_2 - 1$$

$$F_i(x) = x_{i-1} + 2.5x_i + x_{i+1} - 1 \quad \text{for } i = 2, 3, \dots, n-1$$

$$F_n(x) = x_{n-1} + 2.5x_n - 1.$$

Problem 6:

$$F_i(x) = \ln(x_i + 1) - \frac{x_i}{n}, \quad \text{for } i = 2, 3, \dots, n.$$

Problem 7:

$$F_1(x) = x_1(x_1^2 + x_2^2) - 1$$

$$F_i(x) = x_i(x_{i-1}^2 + 2x_i^2 + x_{i+1}^2) - 1 \text{ for } i = 2, 3, \dots, n-1$$

$$F_n(x) = x_n(x_{n-1}^2 + x_n^2).$$

Problem 8:

$$F(x) = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & -1 \\ & -1 & 2 & \end{pmatrix} x + (e^{x_1} - 1, \dots, e^{x_n} - 1)^T.$$

مقایسه الگوریتم ۱ و ۲

در جداول ۱ تا ۸ الگوریتم ارائه شده DLPM با الگوریتم FCG از نظر معیارهای تعداد iteration، زمان و هم چنین تعداد ارزیابی تابع F و هم چنین مقدار نرم نهایی تابع برای هر مسئله ۱ تا ۸ به صورت جداگانه مقایسه شده اند.

Problem 1:

Method	Feval		ITER		Norm		Time	
	DLPM	FCG	DLPM	FCG	DLPM	FCG	DLPM	FCG
x0								
-0.1, n=100000	32	61	5	11	5.766831e-07	0.000006	17.033628	32.557636
-0.5, n=10000	32	61	5	11	9.901152e-07	0.000009	1.131971	2.121534
-1.0, n=100000	47	71	7	13	3.909650e-06	0.000004	24.802898	37.925095
-10.0, n=10000	121	65	16	12	8.814514e-06	0.000005	4.189922	2.233993
0.1, n=10000	23	81	6	19	3.980382e-06	0.000010	0.809523	2.803902
0.5, n=100000	26	97	7	23	5.618589e-06	0.000009	14.186451	51.533784
1.0, n=10000	26	93	7	22	2.622957e-06	0.000010	0.921052	3.395218
10.0, n=100000	405	111	46	26	6.092998e-06	0.000007	215.644637	58.900229

Problem 2:

Problem 3:

Method	Feval		ITER		Norm		Time	
	DLPM	FCG	DLPM	FCG	DLPM	FCG	DLPM	FCG
x0								
-0.1, n=100000	85	109	16	26	0.000008	0.000007	68.946873	80.428400
-0.5, n=10000	37	101	9	24	0.000008	0.000010	2.080523	5.697046
-1.0, n=100000	117	109	20	26	0.000008	0.000009	86.398470	80.698265
-10.0, n=10000	125	109	21	26	0.000008	0.000009	7.064337	6.142025
0.1, n=10000	31	101	8	24	0.000008	0.000008	1.757256	5.723672
0.5, n=100000	70	109	14	26	0.000008	0.000005	52.031798	80.484686
1.0, n=10000	30	101	8	24	0.000002	0.000005	1.881800	5.671424
10.0, n=100000	242	113	34	27	0.000008	0.000009	177.144114	83.013157

Problem 4:

Method	Feval		ITER		Norm		Time	
	DLPM	FCG	DLPM	FCG	DLPM	FCG	DLPM	FCG
x0								
-0.1, n=100000	22	76	6	18	0.000002	0.000006	9.304954	32.201806
-0.5, n=10000	25	71	7	17	0.000002	0.000008	0.622541	1.750117
-1.0, n=100000	38	74	10	18	0.000002	0.000009	15.896683	31.225475
-10.0, n=10000	58	101	18	27	0.000003	0.000009	1.429514	2.457476
0.1, n=10000	23	81	6	19	0.000003	0.000009	0.558157	1.954411
0.5, n=100000	31	97	8	23	0.000002	0.000006	13.114242	41.026883
1.0, n=10000	28	89	7	21	0.000005	0.000006	0.848549	2.186114

Problem 5:

Method	Feval		ITER		Norm		Time	
	DLPM	FCG	DLPM	FCG	DLPM	FCG	DLPM	FCG
x0								
-0.1, n=100000	146	876	20	165	0.000008	0.000008	90.091129	535.523620
-0.5, n=10000	149	867	22	163	0.000007	0.000006	6.452829	37.458624
-1.0, n=100000	121	848	17	159	0.000007	0.000008	73.868643	521.084737
-10.0, n=10000	253	970	31	183	0.000004	0.000009	10.870754	42.166980
0.1, n=10000	190	799	30	150	0.000008	0.000006	8.151117	34.209492
0.5, n=100000	143	851	19	161	0.000004	0.000008	87.144384	517.310199
1.0, n=10000	124	875	18	166	0.000009	0.000007	5.611128	37.886690
10.0, n=100000	577	871	57	163	0.000006	0.000006	355.429512	537.881056

Problem 6:

Method	Feval		ITER		Norm		Time	
	DLPM	FCG	DLPM	FCG	DLPM	FCG	DLPM	FCG
x0								
-0.1, n=100000	23	89	6	21	0.000010	0.000007	14.398838	55.666832
-0.5, n=10000	23	89	6	21	0.000005	0.000006	1.046551	4.017991
0.1, n=10000	19	68	5	16	0.000005	0.000007	0.848756	3.007588
0.5, n=100000	29	79	8	19	0.000002	0.000005	18.142798	49.908320
1.0, n=10000	25	62	7	15	0.000009	0.000009	1.298770	2.770164
10.0, n=100000	283	85	47	22	0.000002	0.000005	176.304551	52.494168

Problem 7:

Method	Feval		ITER		Norm		Time	
	DLPM	FCG	DLPM	FCG	DLPM	FCG	DLPM	FCG
x0								
-0.1, n=100000	144	1995	20	389	0.000004	0.000008	151.476741	2122.650864
-1.0, n=10000	177	3105	23	610	0.000007	0.000009	15.746619	273.837773
1.0, n=10000	159	3102	23	609	0.000005	0.000009	14.016826	271.160795

Problem 8:

Method	Feval		ITER		Norm		Time	
	DLPM	FCG	DLPM	FCG	DLPM	FCG	DLPM	FCG
x0								
-0.1, n=100000	250	1058	29	206	0.000007	0.000010	106.736387	448.745977
0.1, n=10000	227	383	32	74	0.000003	0.000007	5.822476	9.612726

ارزیابی با استفاده از performance profiles

برای مقایسه دو الگوریتم از روش performance profile استفاده میکنیم. در این روش دو الگوریتم از نظر این که چه تعداد مسئله را با چه هزینه ای حل کرده اند مقایسه میشوند.

در این مقایسه هزینه سه مورد تعداد iterationها، زمان و هم چنین تعداد ارزیابی تابع F در نظر گرفته میشود.

- فرض کنید که P مسئله و S الگوریتم مختلف داریم. هزینه حل مسئله p با الگوریتم S را به صورت زیر بیان میکنیم.

$$C_{s,p}$$

- بهترین الگوریتم برای یک مسئله نیز الگوریتمی است که با کمترین هزینه آن مسئله را حل کند یعنی:

$$C_{min,p} = \min_{s \in S}(C_{s,p})$$

- حال هزینه نسبی الگوریتم ها بر اساس این بهترین هزینه مطابق زیر محاسبه میشوند.

$$r_{s,p} = \frac{C_{s,p}}{C_{min,p}}$$

که در نتیجه $r_{s,p} \geq 1$ و $r_{s,p} = 1$ به معنای این است که الگوریتم S جز یکی از بهترین الگوریتم ها از نظر هزینه برای مسئله p است.

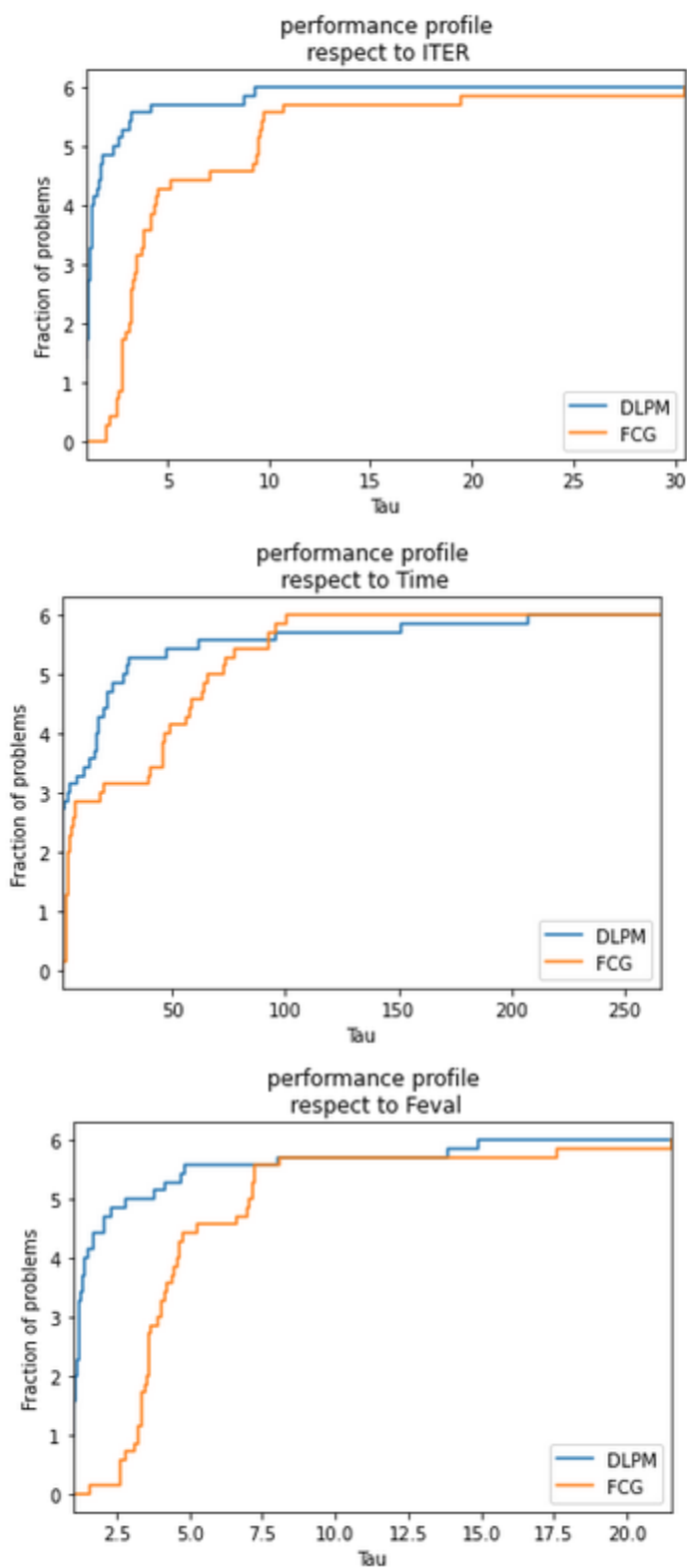
- در نهایت نمودار performance profile بر اساس رابطه زیر محاسبه میشود.

$$P_s(\tau) = \frac{\#\{p \in P \mid r_{s,p} \leq \tau\}}{\#P}$$

که بیان کننده نسبت تعداد مسئله های حل شده توسط الگوریتم S به کل تعداد مسئله هاست که هزینه نسبی آن کمتر مساوی τ باشد.

در نتیجه به ازای مقادیر مختلف τ میتوان مقدار بالا را محاسبه کرد و نموداری مشابه با نمودار زیر برای دو الگوریتم مختلف و هم چنین هزینه های متفاوت شامل تعداد iterationها، زمان و هم چنین تعداد ارزیابی تابع F رسم کرد. در این نمودار هر چقدر سطح زیر نمودار بیشتر باشد به معنای بهتر بودن عملکرد الگوریتم است یعنی الگوریتم با یک هزینه مشخص توانسته است تعداد بیشتری مسئله را حل کند و هم چنین هزینه حل مسائل برای آن الگوریتم کمتر است.

در زیر سه نمودار performance profile برای تعداد iterationها، زمان و هم چنین تعداد ارزیابی تابع F رسم شده است.



همان طور که مشاهده می‌شود الگوریتم ارائه شده (DLPM) در سه تابع هزینه از الگوریتم FCG بهتر عمل کرده است.

- [١] A. Iusem and M. Solodov, 'Newton-Type Methods With Generalized Distances For Constrained Optimization', *Optimization*, vol. ٤١, pp. ٢٥٧–٢٧٨, Nov. ١٩٩٧, doi: ١٠.١٠٨٠/٠٢٣٣١٩٣٩٧٠٨٨٤٤٣٣٩.
- [٢] M. Fukushima, 'Equivalent Differentiable Optimization Problems and Descent Methods for Asymmetric Variational Inequality', *Math. Program.*, vol. ٥٣, Jan. ١٩٩٢, doi: ١٠.١٠٠٧/BF٠١٥٨٥٤٩٤.
- [٣] Y.-B. Zhao and D. Li, 'Monotonicity of Fixed Point and Normal Mappings Associated with Variational Inequality and Its Application', *SIAM J. Optim.*, vol. ١١, no. ٤, pp. ٩٤٢–٩٧٣, Jan. ٢٠٠١, doi: ١٠.١١٣٧/S١٠٥٢٤٢٣٤٩٩٣٥٧٩٥٧.
- [٤] Y. Xiao, Q. Wang, and Q. Hu, 'Non-smooth equations based method for ℓ_1 -norm problems with applications to compressed sensing', *Fuel Energy Abstr.*, vol. ٧٤, pp. ٣٥٧–٣٥٧٧, Jul. ٢٠١١, doi: ١٠.١٠١٦/j.na.٢٠١١.٠٢.٠٤٠.
- [٥] S. Babaie-Kafaki and R. Ghanbari, 'A descent family of Dai–Liao conjugate gradient methods', *Optim. Methods Softw.*, vol. ٢٩, pp. ٥٨٣–٥٩١, May ٢٠١٤, doi: ١٠.١٠٨٠/١٠٥٥٤٧٨٨.٢٠١٣.٨٣٣١٩٩.
- [٦] '(PDF) Nonmonotone Spectral Methods for Large-Scale Nonlinear Systems'. https://www.researchgate.net/publication/٢٤٠٩٥٤١٦٧_Nonmonotone_Spectral_Methods_for_Large-Scale_Nonlinear_Systems (accessed Feb. ٠٨, ٢٠٢٢).
- [٧] '(PDF) A Nonmonotone Line Search Technique for Newton's Method'. https://www.researchgate.net/publication/٢٤٤٩٦٤٠٠٢_A_Nonmonotone_Line_Search_Technique_for_Newton's_Method (accessed Feb. ٠٨, ٢٠٢٢).
- [٨] W. Cruz, J. Martínez, and M. Raydan, 'Spectral Residual Method without Gradient Information for Solving Large-Scale Nonlinear Systems of Equations', *Math Comput*, vol. ٧٥, pp. ١٤٢٩–١٤٤٨, Jul. ٢٠٠٦, doi: ١٠.١٠٩٠/S٠٠٢٥-٥٧١٨-٠٦-٠١٨٤٠-٠٠.
- [٩] L. Zhang and W. Zhou, 'Spectral gradient projection method for solving nonlinear monotone equations', *J. Comput. Appl. Math.*, vol. ١٩٦, no. ٢, pp. ٤٧٨–٤٨٤, Nov. ٢٠٠٦, doi: ١٠.١٠١٦/j.cam.٢٠٠٥.١٠.٠٠٢.
- [١٠] J. BARZILAI and J. M. BORWEIN, 'Two-Point Step Size Gradient Methods', *IMA J. Numer. Anal.*, vol. ٨, no. ١, pp. ١٤١–١٤٨, Jan. ١٩٨٨, doi: ١٠.١٠٩٣/imanum/٨.١.١٤١.
- [١١] '(PDF) A Globally Convergent Inexact Newton Method for Systems of Monotone Equations'. https://www.researchgate.net/publication/٢٨١٢٣٣٢٣٤_A_Globally_Convergent_Inexact_Newton_Method_for_Systems_of_Monotone_Equations (accessed Feb. ٠٨, ٢٠٢٢).
- [١٢] H. Mohammad and A. Abubakar, 'A positive spectral gradient-like method for large-scale nonlinear monotone equations', *Bull. Comput. Appl. Math.*, vol. ٥, pp. ٩٧–١١٣, May ٢٠١٧.
- [١٣] W. Hager and H. Zhang, 'A survey of nonlinear conjugate gradient method', vol. ٢, Jan. ٢٠٠٦.
- [١٤] Y.-H. Dai and L.-Z. Liao, 'New Conjugacy Conditions and Related Nonlinear Conjugate Gradient Methods', *Appl. Math. Optim.*, vol. ٤٣, pp. ٨٧–١٠١, Dec. ٢٠٠١, doi: ١٠.١٠٠٧/S٠٠٢٤٥٠٠١٠٠١٩.
- [١٥] 'A Modified Conjugate Gradient Algorithm on JSTOR'. <https://www.jstor.org/stable/١٧٠٢٦٦> (accessed Feb. ٠٨, ٢٠٢٢).

- [16] W. Cheng, 'A PRP type method for systems of monotone equations', *Math. Comput. Model.*, vol. 50, pp. 15–20, Jul. 2009, doi: 10.1016/j.mcm.2009.04.007.
- [17] B. Polyak, 'The conjugate gradient method in extreme problem', *USSR Comput. Math. Math. Phys.*, vol. 9, pp. 94–112, Dec. 1969, doi: 10.1016/0041-5543(69)90035-4.
- [18] '[PDF] Note sur la convergence de méthodes de directions conjuguées | Semantic Scholar'. <https://www.semanticscholar.org/paper/Note-sur-la-convergence-de-m%C3%A9thodes-de-directions-Polak-Ribi%C3%A9re/9de5fe4a-d1fccc4-b3cd1726d8f28f3fva3c5d>
- [19] 'FR type methods for systems of large-scale nonlinear monotone equations | Request PDF'. https://www.researchgate.net/publication/281432835_FR_type_methods_for_systems_of_large-scale_nonlinear_monotone_equations (accessed Feb. 08, 2022).
- [20] G. Yuan and M. Zhang, 'A three-terms Polak-Ribière-Polyak conjugate gradient algorithm for large-scale nonlinear equations', *J Comput Appl Math*, 2015, doi: 10.1016/j.cam.2015.03.014.
- [21] '(PDF) A family of conjugate gradient methods for large-scale nonlinear equations'. https://www.researchgate.net/publication/319989678_A_family_of_conjugate_gradient_methods_for_large-scale_nonlinear_equations (accessed Feb. 08, 2022).
- [22] W. W. Hager and H. Zhang, 'Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent', *ACM Trans. Math. Softw.*, vol. 32, no. 1, pp. 113–137, Mar. 2006, doi: 10.1145/1132973.1132979.
- [23] Y.-H. Dai and C. Kou, 'A Nonlinear Conjugate Gradient Algorithm with an Optimal Property and an Improved Wolfe Line Search', *SIAM J. Optim.*, vol. 23, Feb. 2013, doi: 10.1137/100813026.