فصل اول

مقدمه‌ای بر بهینه‌سازی غیرخطی

۱-۱مقدمه

امروزه مفهوم «بهینه‌سازی» به‌عنوان یک اصل زیر بنایی در تحلیل بسیاری از مسائل پیچیده تصمیم‌گیری یا تخصیص کاملاً پذیرفته‌شده‌است. برای استفاده از آن ما باید در ابتدا هدف را تعیین کنیم. این هدف می‌تواند سود، زمان، انرژی پتانسیل و به‌طور کلی هر مقدار یا مجموعه‌ای از مقدارها باشدکه می‌تواند توسط تعدادی از متغیرها نمایش داده شود. از اینرو، هدف ما یافتن مقدار متغیرهایی است که هدف را بهینه کند. فرایند تعیین هدف، متغیرها و محدودیت‌ها برای یک مسئله داده شده به «مدل سازی» معروف است. ساخت یک مدل مناسب اولین گام، و گاهی اوقات مهم‌ترین گام، در فرایند بهینه‌سازی است.

پس از اینکه مدل فرمول‌بندی شد، یک الگوریتم بهینه‌سازی می‌تواند برای یافتن جواب آن مورد استفاده قرار گیرد. متاسفانه هیچ الگوریتم بهینه‌سازی جامعی وجود ندارد، بلکه الگوریتم‌های بی‌شماری وجود دارند که هرکدام از آن‌ها برای نوع خاصی از مسئله بهینه‌سازی مناسب هستند. تعیین یک الگوریتم مناسب، انتخاب مهمی است، چرا که تعیین می‌کند مسئله به سرعت یا به کندی حل می‌شود و به‌علاوه آیا مسئله جواب خواهد داشت یا خیر. پس از آنکه یک الگوریتم بهینه‌سازی برای مدل به‌کار گرفته شد، باید بتوانیم تشخیص دهیم که آیا الگوریتم مورد نظر در یافتن جواب بهینه موفق خواهدشد یا خیر. در بسیاری از حالات، برای پاسخ دادن به اینکه آیا مجموعه جواب جاری واقعاً جواب مسئله است یا خیر، شرایطی به‌عنوان شرایط بهینگی وجود دارد. به بیان ساده ریاضی، یک مسئله بهینه‌سازی، یافتن مینیمم یا ماکسیمم مقدار یک تابع به همراه تعدادی محدودیت می‌باشد.

کاربردهای بهینه‌سازی در علوم پایه، مهندسی، اقتصاد و تجارت بسیار فراوان است. به‌طوریکه گفته می‌شود ما در عصر بهینه سازی زندگی می‌کنیم. به‌عنوان مثال می‌توان به طراحی راکتورهای شیمیایی، طراحی مهندسی هوا فضا، تخصیص منابع، زمان بندی، طراحی وساخت پل ها وساختمان ها و... اشاره کرد. بهینه سازی در شاخه آنالیز عددی نیز کاربردهایی نظیر اصول نوسانی در PDE، معادلات غیر خطی در ODE وتوابع جریمه‌ای دارد. در ادامه به بررسی انواع مسائل بهینه سازی می‌پردازیم.

۱-۲ انواع مسائل بهینه‌سازی

فرم عمومی یک مسئله بهینه‌سازی غیرخطی به‌صورت زیر است:

$$\begin{aligned} &\min \quad f(x) \\ &\quad s.t. \\ &h_i(x) = 0\,, \quad i \in I = \{1, \dots, n\} \\ &\mathrm{g}_j(\mathrm{x}) \le 0\,, \quad \mathrm{j} \in \mathrm{J} = \{1, \dots, m\} \\ &\qquad x \in C \end{aligned} \qquad (۱.۲.۱)$$

که در آن $C \subseteq R^n$ یک مجموعه داده‌شده و f و h_1، ...، h_n و g_1،...، g_m توابع تعریف شده روی C (یا یک مجموعه باز شامل C) هستند.

تابع $f(x)$، تابع هدف و معادله‌های $h_i(x)=0$ و نامعادله‌های $g_i(x)\le 0$ محدودیت‌ها یا قیود مسئله نامیده می‌شوند. مجموعه جواب های شدنی مسئله فوق با

$$F = \left\{ x \in C \mid h_i(x) = 0, \quad i = 1,\dots,n, \quad g_j(x) \le 0, \quad j = 1,\dots,m \right\}$$

نمایش داده می‌شود.

- اگر $F = \phi$ مسئله را نشدنی گوییم.
- اگر f روی F از پایین کراندار نباشد،آنگاه مسئله بی‌کران نامیده می‌شود.
- اگر مقدار مینیمم f روی F در $x^* \in F$ به‌دست آورده شود آنگاه x^* یک جواب بهینه و $f(x^*)$ مقدار بهینه نامیده می‌شود.

تعریف ۱-۲-۱: تابع $f : R^n \to R$ را مجذوری گوییم هرگاه

$$\exists Q \in R^{n\times n}, C \in R^n, d \in R \quad s.t. \quad f(x) = \frac{1}{2} x^T Q x + C^T x + d$$

در حالت خاص که $Q = 0$ تابع را آفین و اگر $Q = 0$ و $d = 0$ تابع را خطی گویند.

مسئله بهینه سازی عمومی (۱.۲.۱) ، باتوجه به طبیعت تابع هدف و محدودیت‌های آن (از نظر خطی، غیرخطی و محدب بودن) و همچنین از نظر تعداد متغیرها و هموار بودن توابع موجود (مشتق پذیری یا عدم آن)، می‌تواند به کلاس‌های مختلفی طبقه بندی شود. در حالت کلی مسائل بهینه‌سازی غیرخطی در دو کلاس مهم زیر مورد بررسی قرار می‌گیرند.

۱) بهینه‌سازی نامقید.

۲) بهینه‌سازی مقید.

هدف این پایان نامه، ارائه الگوریتم‌هایی برای حل مسائل بهینه‌سازی غیرخطی نامقید می‌باشد. قابل توجه

اینکه در الگوریتم‌های موجود برای حل مسائل بهینه‌سازی مقید نیز معمولاً نیاز به حل چندین مسئله بهینه‌سازی غیرخطی نامقید می‌باشد که این امر، اهمیت موضوع فوق را بیشتر نمایان می‌سازد.

۱-۳ الگوریتم‌های بهینه‌سازی

اغلب الگوریتم‌هایی که برای حل مسائل بزرگ بهینه‌سازی طراحی شده‌اند، از نوع تکراری هستند. در روند جستجو برای یافتن نقطه‌ای که مسئله بهینه‌سازی را حل کند، معمولا یک نقطه اولیه x_0 انتخاب می‌شود، سپس در یک فرایند تکراری به‌دنبال امکان بهبود جواب در هر مرحله هستیم. با ادامه این روند، دنباله‌ای متوالی از نقاط به‌دست می‌آید که به یک نقطه جواب نزدیک می‌شود. برای مسائل بهینه‌سازی خطی، دنباله تولید شده متناهی است و الگوریتم پس از تعدادی متناهی مرحله (گرچه این تعداد در آغاز و دقیقاً مشخص نیست) به نقطه جواب می‌رسد. برای مسائل بهینه‌سازی غیرخطی، این دنباله در حالت کلی به یک نقطه جواب نمی‌رسد، اما به آن همگرا می‌شود. بنابراین در عمل هنگامی که نقطه‌ای به اندازه کافی نزدیک به نقطه جواب تولید گردد، این فرایند متوقف می‌شود.

یک الگوریتم مناسب باید حداقل دارای شرایط زیر باشد.

۱. حتی با وجود تولید خطا در حین اجرا، توانایی محاسبه جواب را داشته‌باشد.
۲. الگوریتم بیش از اندازه به زمان محاسبه یا ذخیره‌سازی نیاز نداشته‌باشد.
۳. الگوریتم تا حد امکان بدون تاثیر پذیری از خطای داده‌ها و خطای گرد کردن، توانایی یافتن جواب دقیق را داشته‌باشد.

این اهداف می‌توانند ناسازگار باشند. یک الگوریتم ممکن است برای کاهش خطا به مقدار قابل قبول، نیاز به زمان محاسبه بسیار زیادی داشته‌باشد. موضوعاتی مانند مرتبه همگرایی، نیازهای ذخیره‌سازی، سرعت و عوامل دیگر، موضوعات اساسی در بهینه‌سازی هستند، که ارزیابی کمی و مقایسه الگوریتم‌های مختلف را امکان‌پذیر می‌سازند.

عموماً الگوریتم‌هایی که برای حل مسائل بهینه‌سازی نامقید ارائه می‌شوند، الگوریتم‌های تکراری می‌باشند که یک دنباله $\{x_k\}$ همگرا به جواب را می‌سازند. بنابراین در ادامه بحث به تعریف مرتبه همگرایی دنباله که رابطه مستقیمی با سرعت همگرایی دارد می‌پردازیم.

تعریف ۱-۳-۱: [۱] فرض کنید $\{x_k\}_{k=1}^{\infty}$ دنباله‌ای در R^n و همگرا به x^* باشد. مرتبه همگرایی این دنباله را p^* گوییم هرگاه:

$$p^* = sup\left\{p: \text{limsup}\ \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^p} < \infty\right\}.$$

هرچه p^* بزرگ‌تر باشد، همگرایی سریع‌تر است. با معرفی $\beta = \text{limsup}\ \frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|^{p^*}}$ می‌توانیم تعاریف زیر را در مورد مرتبه همگرایی دنباله $\{x_k\}_{k=1}^{\infty}$ ارائه دهیم.

a) هرگاه $p^* = 1$ و $0 < \beta < 1$ آنگاه همگرایی را خطی گوییم.

b) هرگاه $p^* = 1$ و $\beta = 0$ آنگاه همگرایی را زبرخطی[1] گوییم.

c) هرگاه $p^* = 1$ و $\beta = 1$ آنگاه همگرایی را زیرخطی[2] گوییم.

d) هرگاه $p^* = 2$ آنگاه همگرایی را مجذوری گوییم.

اولین سوالی که در مسائل بهینه سازی مطرح می‌شود در مورد وجود جواب مسئله (۱.۴.۱) است. حکم مهمی که برای پاسخ دادن به این پرسش می‌توان از آن استفاده کرد قضیه وایراشتراس است. در قرن نوزدهم،کارل وایراشتراس(۱۸۹۷-۱۸۱۵) نتیجه مشهوری را درتحلیل ثابت کردکه یک تابع پیوسته حقیقی مقدار اینفیمم یا سوپریمم خود را روی یک مجموعه فشرده اختیار می‌کند. یعنی هر مسئله تصمیم‌گیری NLO[3] همراه با شرط پیوستگی f و فشرده بودن ناحیه شدنی دارای جواب بهینه است. در ادامه به تعیین شرایط نقطه بهینه می‌پردازیم.

۱-٤ شرایط بهینگی برای مسئله بهینه‌سازی غیرخطی نامقید

مسئله زیر را در نظر بگیرید

$$min\, f(x) \qquad (۱.۴.۱)$$
$$x \in R^n,$$

که در آن $x \in R^n$ یک بردار حقیقی با $n \geq 1$ مؤلفه می‌باشد و f یک تابع با مشتقات مرتبه دوم پیوسته است. برای یافتن جواب بهینه مسئله فوق ابتدا باید با خصوصیات جواب آن آشنا شویم. در این جهت به تعاریف زیر نیاز داریم. [۲]

تعریف ۱-٤-۱(گرادیان تابع): فرض کنید تابع $f: R^n \to R$ ، به‌طورپیوسته مشتق‌پذیر باشد. در هر نقطه $x \in R^n$ یک بردار شامل مشتقات جزئی مرتبه اول f وجود دارد که بردار گرادیان نامیده می‌شود و به

1 -Superlinear Convergence
2-Sublinear Convergence
3-Nonlinear programming optimization

صورت زیر تعریف می‌گردد.

$$g(x) = \nabla f(x) = (\frac{\partial f}{\partial x_1}(x), ..., \frac{\partial f}{\partial x_n}(x))^T. \qquad (۲.۴.۱)$$

تعریف۱-٤-۲ (ماتریس هسی تابع): فرض کنید تابع $f : R^n \to R$ در هر $x \in R^n$ دو بار به‌طورپیوسته مشتق‌پذیر باشد آنگاه یک ماتریس از مشتقات جزئی مرتبه دوم تابع f موجود است که ماتریس هسی نامیده می‌شود و آنرا به‌صورت $\nabla^2 f(x)$ نمایش می‌دهیم. این ماتریس یک ماتریس مربعی متقارن است که درایه (i, j) آن به‌صورت $\frac{\partial^2 f}{\partial x_i \partial x_j}(x)$ تعریف می‌شود. از آنجا که j امین ستون این ماتریس عبارت است از $\nabla(\frac{\partial f}{\partial x_j})(x)$ از اینرو این ماتریس می‌تواند به‌صورت $\nabla(\nabla f(x))^T$ نوشته شود.

در ابتدا مینیمم موضعی و سراسری تابع هدف مسئله (۱.۴.۱) را تعریف می‌کنیم.

تعریف ۱-٤-۳: [۲] فرض کنید تابع $f : R^n \to R$ داده شده باشد.

۱) نقطه $x^* \in R^n$ را مینیمم موضعی تابع f گوییم، هرگاه

$$\exists \varepsilon > 0 \; s.t. \; \forall x \in R^n : if \; \|x^* - x\| \le \varepsilon \Rightarrow f(x^*) \le f(x).$$

نقطه $x^* \in R^n$ را مینیمم سراسری تابع f گوییم، هرگاه به ازای هر $x \in R^n$

$$f(x^*) \le f(x).$$

۲) نقطه $x^* \in R^n$ را مینیمم موضعی اکید تابع f گوییم، هرگاه

$$\exists \varepsilon > 0 \; s.t. \; \forall x \in R^n : if \; \|x^* - x\| \le \varepsilon \Rightarrow f(x^*) < f(x).$$

نقطه $x^* \in R^n$ را مینیمم سراسری اکیدتابع f گوییم، هرگاه به ازای هر $x \in R^n$

$$f(x^*) < f(x).$$

درحل مسئله (۱.۴.۱)، به‌دنبال یافتن یک نقطه مینیمم سراسری تابع f هستیم. اما معمولاً تعیین این نقطه و حتی مشخص‌کردن محل آن بسیار دشوار است. این واقعیت چه از لحاظ محاسباتی و چه از لحاظ نظری، باعث می‌شود که ما در بسیاری مواقع به نقطه مینیمم موضعی رضایت دهیم. اما به‌منظور توسعه نظریه‌ای که مستقیماً به خصوصیات نقاط مینیمم سراسری و نه مینیمم موضعی بپردازد، ضروری است که نوعی از فرض‌های مربوط به تحدب معرفی شود. مفهوم تحدب در بهینه سازی یک مفهوم اساسی است و یک حالت خاص و مهم، بهینه سازی محدب است که همه جواب‌های موضعی آن، جواب‌های سراسری نیز می‌باشند. اصطلاح تحدب هم برای مجموعه‌ها به‌کار می‌رود و هم برای توابع، لذا به تعاریف زیر توجه می‌کنیم.

تعریف ۱-٤-٤ [۲]

الف) مجموعه‌محدب: مجموعه $S \subseteq R^n$ یک مجموعه محدب ناتهی است اگر برای هر دو نقطه دلخواه $x_1, x_2 \in S$ و هر α که $0 \leq \alpha \leq 1$ داشته‌باشیم

$$\alpha x_1 + (1-\alpha)x_2 \in S.$$

ب)تابع محدب: تابع f، تعریف شده روی مجموعه محدب ناتهی S، محدب گفته می‌شود اگر به ازای همه مقادیر $x_1, x_2 \in S$ و هر α که $0 < \alpha < 1$ داشته‌باشیم

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2).$$

اگر به ازای هر α که $0 < \alpha < 1$ و $x_1 \neq x_2$ داشته‌باشیم

$$f(\alpha x_1 + (1-\alpha)x_2) < \alpha f(x_1) + (1-\alpha) f(x_2).$$

آنگاه f را اکیداً محدب گوییم.

ج)تابع به‌طور یکنواخت محدب[1]: تابع f تعریف شده روی مجموعه محدب ناتهی S، به‌طور یکنواخت محدب گفته می‌شود اگر یک ثابت $\mu > 0$ وجود داشته‌باشد به‌طوریکه به ازای همه مقادیر $x_1, x_2 \in S$ و هر α که $0 < \alpha < 1$ داشته باشیم

$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2) - \frac{1}{2}\mu\alpha(1-\alpha)\|x_1 - x_2\|^2,$$

همچنین اگر f به‌طورپیوسته مشتق‌پذیر باشد، آنگاه به‌سادگی می‌توان نشان داد که این رابطه معادل با روابط زیر است

$$\left(\nabla f(x_1) - \nabla f(x_2)\right)^T (x_1 - x_2) \geq \mu\|x_1 - x_2\|^2. \qquad (۳.۴.۱)$$

$$f(x_1) \geq f(x_2) + \nabla f(x_2)^T (x_1 - x_2) + \frac{1}{2}\mu\|x_1 - x_2\|^2. \qquad (۴.۴.۱)$$

در ادامه با بیان تعدادی لم به بیان خواص توابع محدب می‌پردازیم.

لم ۱-٤-۱: [۲] تابع به‌طورپیوسته مشتق‌پذیر f روی مجموعه محدب ناتهی S، محدب است اگروتنها اگربه ازای هر $x, y \in S$ داشته‌باشیم

$$f(y) \geq f(x) + \nabla f(x)^T (y - x).$$

1-Uniformly Convex Function

لم ۱-٤-۲: [۱] فرض کنید تابع f در هر نقطه $x \in R^n$ دو بار به‌طورپیوسته مشتق‌پذیر باشد، در این صورت f روی مجموعه محدب ناتهی S مشتمل بر نقطه درونی، محدب است اگروتنها اگر ماتریس هسی Gدر سراسر S نیمه معین‌مثبت باشد.

یکی از ابزارهای مهم در بهینه سازی، قضیه تیلور می‌باشد که مبنای بسیاری از روش‌های عددی محسوب می‌شود . در ادامه به بیان این قضیه می‌پردازیم.

قضیه ۱-٤-۱(قضیه تیلور): [۳،۴] فرض‌کنیم f یک تابع حقیقی بر $[a,b]$ بوده و n یک عددصحیح مثبت باشد. اگر $f^{(n)}$ موجود و بر $[a,b]$ پیوسته باشد و $f^{(n+1)}$ بر روی (a,b) وجود داشته باشد، آنگاه برای هر دو نقطه‌ی x و c در $[a,b]$ دایم:

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(c)}{k!}\ (x-c)^{\mathrm{k}} + E_n(x),$$

که در آن، به ازای نقطه‌ای مانند ξ بین x و c داریم:

$$E_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x-c)^{n+1}\ .$$

براساس این قضیه، اگر مشتق مرتبه $n+1$ ام تابع حقیقی f وجود داشته‌باشد، آنگاه می‌توان تابع f را با یک چند جمله‌ای از درجه n تقریب زد. حالت خاص $n=0$ قضیه تیلور، اغلب در استدلال‌ها به قضیه مقدار میانگین معروف است.

قضیه ۱-٤-۲(قضیه تیلورمرتبه دوم برای توابع n متغیره): [۱،۵] فرض کنید $f: R^n \to R$ یک تابع n متغیره و دو بار به‌طورپیوسته مشتق‌پذیر و $p \in R^n$ باشد. آنگاه برای برخی از مقادیر $t \in (0,1)$ داریم:

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)p\ dt.$$

به‌علاوه برای برخی از مقادیر $t \in (0,1)$ داریم:

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp)p.$$

هدف این پایان‌نامه ارائه الگوریتم‌های مناسب برای حل مسئله بهینه‌سازی است. اما برای محاسبه جواب بهینه قبل از هر چیز ابتدا باید با خواص نقطه بهینه آشنا شویم. قضایای زیر می‌تواند ما را در شناسایی نقاط مینیمم تابع f یاری کند.

شرایط لازم مرتبه اول

قضیه ۱-٤-۳: [۵] فرض کنید f روی R^n به‌طورپیوسته مشتق‌پذیرباشد. اگرنقطه $x^* \in R^n$یک مینیمم موضعی تابع f باشد، آنگاه $\nabla f(x^*) = 0$.

اثبات: برای مشاهده اثبات به [۵] مراجعه کنید. ■

مشاهده می‌شود که شرط $\nabla f(x^*) = 0$ یک شرط لازم برای مینیمم موضعی بودن x^* است، اما برقراری این رابطه مینیمم بودن x^* را تضمین نمی‌نماید، نقاط صادق در شرایط فوق نقاط ایستا نامیده می‌شوند. به‌سادگی ثابت می‌گردد که شرط $\nabla f(x^*) = 0$در مورد توابع محدب، یک شرط کافی نیز می‌باشد.

لم ۱-٤-۳: [۵] فرض کنید f یک تابع محدب به‌طورپیوسته مشتق‌پذیر باشد. نقطه $x^* \in R^n$ یک مینیمم موضعی تابع f می‌باشد، اگروتنها اگر $\nabla f(x^*) = 0$.

پیش‌تر بیان‌شد که هدف مطلوب ما در حل مسائل مینیمم‌سازی، یافتن یک نقطه مینیمم سراسری تابع f است. اما معمولاً تعیین این نقطه و حتی مشخص کردن محل آن بسیار دشوار است، از اینرو در بسیاری از مواقع تنها به یافتن نقطه مینیمم موضعی رضایت می‌دهیم. اما این امر زمانی که f یک تابع محدب باشد، متفاوت خواهد بود. قضیه زیر بیانگر این ویژگی مطلوب توابع محدب است.

قضیه ۱-٤-٤: [۵] فرض کنیم تابع f روی زیرمجموعه محدب S از R^n محدب باشد، آنگاه هر مینیمم موضعی آن یک مینیمم سراسری است.

شرایط لازم مرتبه دوم که برحسب ماتریس هسی $\nabla^2 f$ مرکب از مشتقات جزئی مرتبه دوم تابع f تعریف می‌شوند، از لحاظ نظری بسیار مهم می‌باشند. زیرا این شرایط علاوه بر شرط لازم $\nabla f(x^*) = 0$، که نقاط دیگری به‌جز مینیمم تابع f نیز در آن صدق می‌کنند، شرط دیگربی را بیان می‌کند که مینیمم تابع f باید در آن صادق باشد.

شرایط لازم مرتبه دوم

قضیه ۱-٤-۵: [۵] فرض کنید تابع f دو بار به‌طورپیوسته مشتق پذیر است. اگر x^* مینیمم موضعی تابع f باشد و $\nabla^2 f$ بر روی یک همسایگی باز x^* پیوسته باشد، آنگاه $\nabla f(x^*) = 0$ و $\nabla^2 f(x^*)$ نیمه معین‌مثبت است.

اثبات:برای مشاهده اثبات به [۵] مراجعه کنید. ■

شرایط کافی مرتبه دوم

قضیه ۱-٤-٦: [۵] فرض کنید تابع f دو بار به‌طورپیوسته مشتق پذیر است. اگر نقطه x^* موجود باشد به‌طوریکه $\nabla^2 f$ بر روی یک همسایگی باز x^* پیوسته بوده و $\nabla f\left(x^*\right)=0$ و $\nabla^2 f\left(x^*\right)$ معین‌مثبت باشد، آنگاه x^* یک مینیمم موضعی اکید برای تابع f است.

اثبات: برای مشاهده اثبات به[۵] مراجعه کنید. ■

از این پس و در سراسر این پایان‌نامه همواره فرض می‌کنیم تابع f دو بار به‌طورپیوسته مشتق پذیر است مگر اینکه خلاف آن ذکر شود.

۱-۵ الگوریتم‌های کاهشی برای بهینه‌سازی نامقید

اکنون به هدف اصلی که ساخت الگوریتم‌های مناسب برای حل مسئله بهینه‌سازی نامقید که دارای فرم کلی زیر است باز می‌گردیم.

$$\begin{aligned} &\min \quad f(x) \\ &x \in R^n. \end{aligned} \qquad (۱.۵.۱)$$

ایده اصلی بیشترالگوریتم‌های موجود برای حل مسئله فوق تولید یک دنباله کاهشی$\{x_k\}$ ، با شروع از یک نقطه ابتدایی مانند x_0 و استفاده از یک طرح تکراری برای تولید دنباله است،که هدف مطلوب آن همگرایی به سمت نقطه ایستا است. زیرا در این صورت احتمالاً شرایط مرتبه دوم برقرار خواهندشد. لازم است قبل از هرگونه بحثی مفهوم الگوریتم کاهشی را توضیح دهیم. در ادامه برای سهولت در نمایش، اگر x_k تکرار اخیر و x^* جواب مینیمم یا نقطه ایستای (۱.۵.۱) باشد آنگاه $f(x_k)$ را با f_k ، $\nabla f(x_k)$را با g_k، $\nabla^2 f(x_k)$ را با G_k و $f(x^*)$ را با f^* نمایش می‌دهیم.

در فرایند تکراری حل مسئله (۱.۵.۱) به‌دنبال یک جهت d_k و یک طول گام $\alpha_k > 0$ برای تعریف x_{k+1} به صورت زیر هستیم

$$x_{k+1} = x_k + \alpha_k d_k . \qquad (۲.۵.۱)$$

به‌طوریکه $f(x_{k+1}) \le f(x_k)$. این فرآیند نیازمند انتخاب مناسب d_k و α_k می‌باشد. در راستای یافتن d_k مناسب بسط تیلور تابع f حول x_k در راستای d_k را به‌صورت زیر درنظر بگیرید:

$$f(x_k + \alpha_k d_k) = f_k + \alpha_k g_k^T d_k + \frac{1}{2}\alpha_k{}^2 d_k^T G_k d_k + O\left(\|d_k\|^3\right).$$

اگر $g_k^T d_k < 0$ و α_k به اندازه کافی کوچک باشد، آنگاه $f(x_k + \alpha_k d_k) \le f_k$. بنابراین شرط $g_k^T d_k < 0$

یک شرط کافی برای این است که f در راستای d_k قابلیت کاهش داشته‌باشد. در این حالت اصطلاحاً گفته می‌شود d_kیک جهت کاهشی است. پس از تعیین جهت کاهشیd_k، طول گام α_k در فرایند تکراری (۲.۵.۱) ، با استفاده از جستجوی خطی که در واقع نوعی بهینه‌سازی یک‌متغیره است، تعیین می‌شود

و بدیهی‌ترین روش انتخاب طول گام α_k، جستجوی خطی دقیق می‌باشد،یعنی

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

توجه داریم که الگوریتم عمومی جریان کاهشی برای مینیمم سازی تابع f را می‌توان به‌صورت زیر بیان کرد.

۱-۵-۱ الگوریتم عمومی:

ورودی الگوریتم: پارامتر دقت $\varepsilon > 0$ و بردار آغازین x_0.

گام(۰): قرارده $k = 0, x = x_0$.

گام(۱): یک جهت جستجوی کاهشی d_k را پیدا کن.

اگر جهتی کاهشی موجود نیست توقف.(x_k یک کاندید برای جواب بهینه است.)

گام(۲): (جستجوی خطی دقیق) α_k را به‌گونه ای محاسبه کنید که

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

گام(۳): قرار ده $x_{k+1} = x_k + \alpha_k d_k$.

گام(۴): در صورت برقراری معیار توقف ، توقف کن، در غیر این صورت $k = k+1$ و به گام (۱) برو.

تذکر ۱-۵-۱: انتخاب مناسب بردار آغازین x_0 در الگوریتم عمومی یکی از عناصر حیاتی در همه الگوریتم‌ها است. هر الگوریتم تکراری با مشخص کردن یک نقطه آغازین شروع می‌شود. اگر برای هر نقطه دلخواه آغازین، الگوریتم همواره دنباله‌ای از نقاط را تولید کند که همگرا به یک جواب شوند،آنگاه الگوریتم را «همگرای سراسری» نامند. قطعاً تمام الگوریتم‌ها این خاصیت مطلوب را ندارند. در واقع بسیاری از مهم‌ترین الگوریتم‌ها برای حل مسائل بهینه‌سازی غیرخطی همگرای سراسری نیستند و در نتیجه گاهی دنباله‌هایی تولید می‌کنند که یا اصلاً همگرا نمی‌شوند و یا همگرایی به سمت نقاطی است که جواب مسئله نیستند.

یکی دیگر از خواص مطلوب برای الگوریتم‌های بهینه‌سازی نامقید، این است که بتوانیم ثابت کنیم الگوریتم برای مسائل درجه دوم در تعداد متناهی تکرار به جواب x^* همگرا باشد، در این صورت گوییم الگوریتم دارای «خاصیت مدل درجه دوم» است. این ویژگی از این نظر مطلوب است که با توجه به قضیه تیلور داریم

هر تابع در یک همسایگی مناسب x^* بر یک مدل درجه دوم انطباق دارد.

یکی از عناصر تاثیرگذار در الگوریتم‌های مینیمم سازی، چگونگی تولید جهت جستجوی d_k است زیرا با توجه به گام(۱)، الگوریتم عمومی برای ساختن دنباله مورد نظر، به یک جهت جستجو نیاز دارد. اکنون به طور مختصر به بررسی دو نوع از مهم‌ترین الگوریتم‌های تکراری کاهشی در بهینه‌سازی غیرخطی می‌پردازیم.

۱-۵-۱ روش گرادیان

یکی از قدیمی‌ترین و شناخته‌شده‌ترین روش‌ها برای مینیمم سازی یک تابع چند متغیره، روش تندترین کاهش است که به روش گرادیان نیز مشهور می‌باشد. این روش از دیدگاه نظری دارای اهمیت فوق العاده‌ای می‌باشد زیرا یکی از ساده‌ترین روش‌هایی است که برای آن تحلیل قانع کننده‌ای وجود دارد. انگیزه برای طرح الگوریتم‌های پیشرفته‌تر اغلب از تلاش برای اصلاح روش اساسی گرادیان به‌نحوی که الگوریتم‌های جدید دارای خواص همگرایی بهتری باشند، به‌وجود آمده‌اند. بنابراین روش گرادیان نه‌تنها اولین روشی است که برای حل یک مسئله جدید امتحان می‌شود، بلکه مرجعی برای سنجیدن سایر روش‌ها نیز می‌باشد[۱].

فرض می‌کنیم f دارای مشتقات جزیی مرتبه اول پیوسته در R^n باشد. در این روش به بردار گرادیان f نیاز داریم و در رابطه (۲.۵.۱) قرار می‌دهیم

$$d_k = -\nabla f(x_k) \triangleq -\mathrm{g}_k.$$

به‌وضوح مشاهده می‌شود که جهت تولید شده توسط روش گرادیان در شرط کاهشی $g_k^T d_k < 0$ صدق می‌کند.

این روش اگرچه هزینه محاسباتی کمی‌دارد اما سرعت همگرایی آن خطی است و حتی ممکن است دچار پدیده زیگزاگ گردد، عموماً این الگوریتم دور از جواب نتایج مطلوبی تولید می‌نماید. همچنین این الگوریتم دارای خاصیت مدل درجه دوم نیست. در جهت رفع مشکلات روش گرادیان، روش‌های گرادیان مزدوج ارائه شده‌اند که فصل دوم به بررسی این روش‌ها و ویژگی‌های آن‌ها اختصاص یافته است.

۱-۵-۲ روش نیوتن

برای معرفی روش نیوتن ابتدا فرض می‌کنیم که تابع f دو بار به‌طورپیوسته مشتق‌پذیر و محدب اکید است. جهت جستجوی نیوتن در بهینه‌سازی چند بعدی بر پایه مینیمم کردن یک تقریب مرتبه دوم تابع f است. بنابراین با به‌کار بردن سه جمله اول بسط تیلور تابع f ، تقریب مجذوری f را حول نقطه x_k به‌دست آورده

وآنرا مینیمم می‌کنیم.

$$q(x) \cong f_k + g_k^T(x - x_k) + \frac{1}{2}\,(x - x_k)^T \nabla^2 f(x_k)(x - x_k).$$

می‌توان نشان داد $q(x)$ محدب اکید است اگروتنها اگر $\nabla^2 f\,(x_k)$معین‌مثبت باشد و مینیمم مقدار $q(x)$ وقتی که گرادیان آن مساوی بردار صفر باشد، به‌دست می‌آید. به عبارت دیگر در نقطه x_k داریم

$$\nabla q(x) = g_k + \nabla^2 f\,(x_k)(x - x_k) = 0.$$

بنابراین در روش نیوتن که دنباله $\{x_k\}$به‌صورت $x_{k+1} = x_k - G_k^{-1} g_k$ تولید می‌شود، اگر نقطه آغازین به اندازه کافی نزدیک به نقطه مینیمم باشد آنگاه روش‌های نیوتن همگرایی مجذوری با طول گام کامل(بدون هیچ جستجو خطی $\alpha = 1$) را نتیجه می‌دهند .

۱-۵-۲ الگوریتم نیوتن:

گام(۱): از نقطه آغازین x_0 شروع کن .

گام(۲): با محاسبه $G_k = \nabla^2 f(x_k)$و $g_k = \nabla f(x_k)$قرار ده $d_k = -G_k^{-1} g_k$ و $x_{k+1} = x_k + d_k$.

گام(۳): تا رسیدن به دقت مورد نظر فرایند را تکرار کن .

تذکر ۱-۵-۱: عملاً به جای محاسبه d_k از رابطه $d_k = -G_k^{-1} g_k$، d_k را یک جواب دستگاه $G_k d_k = -g_k$ در نظر می‌گیریم. البته توجه داریم در صورتیکه G_k وارون پذیر نباشد نمی‌توان جهت جستجوی d_k را محاسبه نمود.

تذکر ۱-۵-۲: واضح است که جهت نیوتن در صورتی کاهشی است که$g_k^T G_k g_k > 0$، از طرفی در x^*، G^* معین‌مثبت است و لذا بنابر پیوستگی مشتق مرتبه دوم G_kنیز معین‌مثبت است. لذا کاهشی بودن الگوریتم در نزدیکی جواب تضمین می‌گردد، اما در غیر این صورت ممکن است جهت نیوتن کاهشی نباشد، به‌خصوص در حالتی که G_k وارون‌پذیر باشد ولی معین‌مثبت نباشد. در این حالت با اینکه جهت به‌دست آمده از روش نیوتن موجود است اما ممکن است کاهشی نباشد. در این صورت ایده این است که به جای G_k^{-1} یک تقریب معین‌مثبت از آن مانند H_k جایگزین کنیم و بنابراین داریم $d_k = -H_k g_k$.

تذکر ۱-۵-۳: الگوریتم نیوتن از جستجوی خطی استفاده نمی‌کند زیرا طول گام آن کامل است. لذا در بعضی موارد شرط کاهشی بودن نقض می‌گردد. اما بایستی توجه داشت که حتی در صورت معین‌مثبت بودن G_k هیچ تضمینی برای کاهش در تابع هدف وجود ندارد، و معین‌مثبت بودن G_k تنها یک جهت کاهشی تولید می‌کند. در این حالت ممکن است یک نسخه الگوریتم نیوتن به همراه جستجوی خطی، پیشنهادی

مناسب برای تولید الگوریتم کاهشی باشد که این روش نیوتن پرشی[1] نامیده می‌شود، در این صورت $x_{k+1}=x_k+\alpha_k d_k$ که در آن α_k طول گام است. در این حالت $\alpha_k \neq 1$است، زیرا روش نیوتن دارای طول گام یک می‌باشد.

تذکر ۱-۵-٤: عموماً در نزدیکی جواب، G_kمعین مثبت است. بنابراین برای حل دستگاه $G_k d_k = -g_k$ می‌توان از روش تجزیه چولسکی استفاده کرد ($G_k = L_k L_k^T$) که نیاز به $O(\frac{1}{6}n^3)$ عملیات محاسباتی دارد. مزیت تجزیه چولسکی این است که برای محاسبه G_{k+1} از G_k ، دیگر $O(\frac{1}{6}n^3)$عملیات لازم نیست بلکه می‌توان $L_{k+1}L_{k+1}^T$ را از $L_k L_k^T$ با $O(n^2)$ عملیات محاسبه نمود[۶].

تذکر ۱-۵-۵: هزینه محاسبات روش نیوتن بالا می‌باشد، زیرا در هر مرحله نیاز به محاسبه n مشتق مرتبه اول برای محاسبه g_k و n^2 مشتق مرتبه دوم برای محاسبه G_k دارد. این مشکل ما را به سمت خانواده روش‌های شبه‌نیوتن سوق می‌دهد. بنابراین از یک تقریب H_k برای G_k^{-1} استفاده می‌کنیم. به‌طوریکه H_kیک ماتریس معین‌مثبت باشد، که هم کاهشی بودن d_k را تضمین می‌کند و هم مشکل هزینه محاسبات G_k را برطرف می‌سازد. در واقع محاسبه ماتریس هسی و معکوس هسی نسبت به محاسبه گرادیان خیلی پرهزینه‌تر است .

در راستای رفع مشکلات روش نیوتن و در عین حال حفظ خواص مطلوب آن، مانند سرعت همگرایی بالای این روش، روش‌های مختلفی ارائه گردیده است که در این راستا می‌توان به روش‌های شبه نیوتن اشاره نمود. در ادامه با توجه به اینکه برخی از الگوریتم‌های بیان شده در فصول آینده با استفاده از خواص روش‌های شبه نیوتنی به‌دست آمده‌اند، لازم به‌نظر می‌رسد که این روش‌ها را با تفصیل بیشتری مورد بررسی قرار دهیم.

۱-۵-۳ روش‌های شبه‌نیوتن

از آنجا که محاسبه و به‌کارگیری ماتریس هسی غیر عملی یا پرهزینه است، ایده زیربنایی روش‌های شبه‌نیوتن بر اساس استفاده از یک تقریب ماتریس وارون هسی به جای وارون حقیقی مورد نیاز در روش نیوتن قرار دارد. شکل این تقریب در روش‌های متفاوت تغییر می‌کند، از ساده ترین شکل که در سرتاسر فرایند تکراری ثابت می‌ماند تا شکل‌های پیشرفته‌تر که تقریب‌های بهتری بر پایه اطلاعات جمع آوری شده در خلال فرایند کاهشی ساخته می‌شود. در حالت ایده‌آل این تقریب‌ها به وارون هسی در نقطه جواب همگرا می‌شوند. در روش‌های شبه نیوتن می‌توان هم ماتریس هسی G_k را به‌وسیله ماتریس B_k تقریب زد و هم ماتریس G_k^{-1} را

1-Damped Newton

به‌وسیله ماتریس H_k تقریب زد. در واقع یک ایده مناسب این است که به جای اینکه G_k را تقریب زده و سپس وارون تقریبی آنرا به‌دست آوریم، همان بهتر است که در ابتدا وارون G_k را به‌وسیله H_k تقریب بزنیم. چون ما به یک الگوریتم کاهشی نیاز داریم، پس بهتر است که H_k معین‌مثبت باشد. از اینرو ایده مناسب این است که یک تقریب معین‌مثبت H_k از G_k^{-1} بیابیم. در نتیجه با توجه به مطالب بیان شده، مهم‌ترین مسئله در روش‌های شبه نیوتن ساخت ماتریس تقریبی H_k است. در حالت کلی می‌توان الگوریتم عمومی شبه نیوتن را به‌صورت زیر بیان کرد:

۱-۵-۳ الگوریتم شبه‌نیوتن:

گام(۰): با بردار آغازین x_0 و ماتریس وارون‌پذیر H_0 شروع کن.

گام(۱): قرارده $d_k = -H_k g_k$.

گام(۲): با یک جستجوی خطی در راستای d_k ، طول گام α_k را محاسبه کن. سپس قرارده $x_{k+1} = x_k + \alpha_k d_k$.

گام(۳): در صورت برقراری شرایط توقف، توقف کن.

گام(۴): H_{k+1} را با استفاده از H_k, g_k, d_k بهنگام کنید.

تذکر ۱-۵-۶:

۱. در گام(۴) علاقه‌مندیم که H_{k+1} معین مثبت بوده و نسبت به H_k برتری داشته‌باشد. توجه داریم که اگر H_k معین‌مثبت باشد، d_k یک جهت کاهشی است. زیرا $d_k^T g_k = -g_k^T H_k g_k < 0$.

۲. در حالت کلی کافی است H_0 را یک ماتریس معین‌مثبت $n \times n$ قرار دهیم. اما معمولاً به‌دلیل اطلاعات کمی که از مسئله در اختیار داریم، بهتر است این هزینه را به خود تحمیل نکرده و قرار دهیم $H_0 = I$ ، در این صورت جهت ابتدایی همان جهت گرادیان است.

۳. ایده مناسب در گام سوم برای تبدیل H_k به H_{k+1} استفاده از رابطه معروف سکانت می‌باشد. ما نیاز داریم که H_{k+1} شبیه G_{k+1}^{-1} رفتار کند، لذا از بسط تیلور بردار گرادیان حول نقطه x_{k+1} داریم

$$g(x) \cong g_{k+1} + G_{k+1}(x - x_{k+1}).$$

اکنون قراردادن $x = x_k$ ، $y_k = g_{k+1} - g_k$ و $s_k = x_{k+1} - x_k$ در رابطه اخیر نتیجه می‌دهد:

$$G_{k+1} s_k \cong y_k.$$

در واقع y_k مقدار $G_{k+1} s_k$ را تقریب می‌زند. از اینرو می‌توان نوشت $G_{k+1}^{-1} y_k \cong s_k$. ما می‌خواهیم H_{k+1} را به‌عنوان تقریب G_{k+1}^{-1} به‌کار ببریم. پس کافی است H_{k+1} را طوری محاسبه کنیم که در رابطه زیر صدق

کند.

$$H_{k+1}y_k = s_k . \qquad (۳.۵.۱)$$

این معادله به معادله سکانت (استاندارد) معروف می‌باشد.

با توجه به الگوریتم عمومی شبه نیوتن دیده می‌شود که مهم‌ترین گام الگوریتم چگونگی تولید ماتریس H_{k+1} از H_k است که به بهنگام سازی ماتریس هسی معروف است. در زیر سه نمونه از ساده‌ترین طرح‌های پیشنهادی به منظور بهنگام سازی ماتریس H_{k+1} را به‌طور مختصر مرور می‌کنیم. [۷]

۱-۵-۳-۱ تصحیح رتبه یک *SR1*

در این روش در هر مرحله فرض می‌کنیم $H_{k+1}=H_k+E_k$ که در آن E_k را یک ماتریس با مرتبه یک تعریف می‌کنیم که تقریب وارون را بهنگام سازد. به دلیل تقارن مورد نیاز قرار می‌دهیم:

$$H_{k+1} = H_k + E_k = H_k + \alpha uu^T .$$

ما نیاز داریم H_{k+1} در رابطه سکانت صدق کند، یعنی $H_{k+1}y_k = s_k$. با انجام محاسبات لازم، رابطه بهنگام رتبه یک (*SR1*) به‌صورت زیر به‌دست می‌آید:

$$H_{k+1}=H_k+\frac{(s_k-H_k y_k)(s_k-H_k y_k)^T}{(s_k-H_k y_k)^T y_k}. \qquad (۱.۳.۵.۱)$$

که اگر H_k متقارن باشد، H_{k+1} نیز متقارن است.

برای رابطه بهنگام رتبه یک (*SR1*) دو حالت ممکن است رخ دهد:

۱) $(s_k-H_k y_k)^T y_k \neq 0$.در این صورت از روند فوق یک بهنگام رتبه یک منحصر بفرد به‌وجود می‌آید. البته اگر $(s_k-H_k y_k)^T y_k$ ناصفر باشد، اما بسیار کوچک باشد، این امر ممکن است منجر به مشکلات عددی گردد.

۲) $(s_k-H_k y_k)^T y_k = 0$. در این صورت دو حالت را درنظر می‌گیریم: یا $s_k = H_k y_k$ که در این حالت کافیست قراردهیم $H_{k+1}=H_k$، یعنی به بهنگام سازی نیازی نیست و ماتریس H_k خاصیت مطلوب را دارد. در غیر این صورت هیچ فرمول بهنگام رتبه یکی موجود نیست که در معادله سکانت صدق کند و در نتیجه باید از فرمول‌های بهنگام دیگری کمک گرفت.

این بهنگام با وجود سادگی دارای مشکلاتی است، اول اینکه معین‌مثبت بودن را فقط در صورتی حفظ

می‌کندکه $(s_k - H_k y_k)^T y_k > 0$. برقراری این شرط را نمی‌توان تضمین کرد. در واقع حتی تضمینی برای اینکه $(s_k - H_k y_k)^T y_k \neq 0$ موجود نیست، یعنی ممکن است رابطه خوش‌تعریف نباشد. دوماً همان‌طور که بیان شد در صورتیکه $(s_k - H_k y_k)^T y_k \neq 0$ اما بسیار کوچک باشد، این امر موجب رشد درایه‌های ماتریس H_{k+1} شده و می‌تواند سبب مشکلاتی از جمله ناپایداری عددی روش گردد. در عمل مشاهده می‌شود که بهنگام تولید شده از روش (*SR1*) تقریب های خوبی برای ماتریس هسی تولید می‌کند و نتایج آن‌ها در بسیاری از موارد از بهنگام رتبه دو بهتر است. اما چون راهی برای تضمین معین‌مثبت بودن وجود ندارد ، ناچار هستیم از بهنگام رتبه دو استفاده کنیم.

۱-۵-۳-۲ روش دیویدان-فلچر-پاول (*DFP*)

یکی از قدیمی‌ترین وهوشمندانه‌ترین شیوه‌ها برای ساختن وارون هسی یا خود هسی، شیوه ای است که ابتدا *Davidon* در سال ۱۹۵۹آنرا پیشنهاد کرد و سپس *Fletcher* و *Powell* آنرا تکامل بخشیدند. در هر گام وارون هسی با جمع دو ماتریس متقارن رتبه یک بهنگام می‌شود و به این دلیل اغلب به روش تصحیح رتبه دو معروف است. این روش دارای این خاصیت جذاب و مطلوب است که برای یک تابع هدف درجه دوم، هم‌زمان با ساختن ماتریس هسی، جهت‌های روش گرادیان مزدوج[1] را تولید می‌کند. به‌منظور به‌دست آوردن این رابطه قرار می‌دهیم.

$$H_{k+1} = H_k + auu^T + cvv^T. \qquad (۲.۳.۵.۱)$$

باز هم هدف برقراری رابطه سکانت $H_{k+1} y_k = s_k$ است. پس از انجام محاسبات لازم، به رابطه بهنگام *DFP* به صورت زیر می‌رسیم.

$$H_k^{DFP} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}. \qquad (۳.۳.۵.۱)$$

قضیه زیر یکی از خواص مهم بهنگام *DFP* را مطرح می‌نماید.

قضیه ۱-۵-۳: [۷] اگر H_k معین‌مثبت و $s_k^T y_k > 0$، آنگاه H_{k+1} نیز معین‌مثبت است.

تذکر ۱-۵-۷: توجه کنید که شرط $s_k^T y_k > 0$ را می‌توان به‌وسیله شرایط جستجوی خطی به مسئله تحمیل کرد. در ادامه به برخی از خواص مطلوب الگوریتم *DFP* می‌پردازیم.

a) در مورد توابع درجه دوم با هسی معین‌مثبت

1- *Conjugate Gradient*

۱) الگوریتم شبه‌نیوتن DFP با استفاده از جستجوی خطی دقیق در هر تکرار، در حداکثر $n+1$ تکرار به جواب می‌رسد $g_{n+1}=0$ و $H_{n+1}=G^{-1}$.

۲) رابطه سکانت برای قدم های پیشین حفظ می‌گردد یعنی برای $i=1,2,...,n$ و برای $j=1,2,...,i-1$ رابطه $H_i y_j = s_j$ همواره برقرار است.

۳) اگر $H_0=I$ آنگاه مسیرهای تولید شده همان مسیرهای گرادیان مزدوج نسبت به ماتریس هسی G هستند.

b) خواص الگوریتم DFP در مورد توابع عمومی

۱) در هر مرحله $3n^2+O(n)$ عملیات اصلی نیاز داریم.

۲) مرتبه همگرایی زبرخطی است.

۳) با جستجوی خطی دقیق همگرایی سراسری در مورد توابع محدب ثابت شده‌است.

۱-۵-۳-۳ روش برویدن-فلچر-گلدفارب-شانو ($BFGS$)

فرمول کلی تصحیح رتبه دو متقارن، به‌منظور بهنگام سازی ماتریس H_{k+1}، به‌صورت زیر می‌باشد:

$$H_{k+1} = H_k + [u\ \ v]\begin{bmatrix} a & b \\ b & c \end{bmatrix}[u\ \ v]^T.$$

یعنی:

$$H_{k+1} = H_k + auu^T + cvv^T + b(vu^T + uv^T). \qquad (۴.۳.۵.۱)$$

اگر در رابطه (۴.۳.۵.۱)، قرار دهیم $b=0$ ، آنگاه تصحیح رتبه دو متقارن(۲.۳.۵.۱) به‌دست می‌آید. مشابه روش DFP به‌منظور برقراری رابطه سکانت $H_{k+1}y_k = s_k$ ، پس از انجام محاسبات لازم و قراردادن $c=0$، رابطه بهنگام $BFGS$ برای ماتریس H_{k+1} به‌صورت زیر به‌دست می‌آید:

$$H_{k+1}^{BFGS} = H_k + \left(1+\frac{y_k{}^T H_k y_k}{y_k{}^T s_k}\right)\frac{s_k s_k{}^T}{s_k{}^T y_k} - \left(\frac{s_k y_k{}^T H_k + H_k y_k s_k{}^T}{y_k{}^T s_k}\right). \qquad (۵.۳.۵.۱)$$

تذکر ۱-۵-۹: بهنگام $BFGS$ تمامی خواص مطلوب بهنگام DFP را دارا می‌باشد. به‌علاوه در صورت استفاده از جستجوی خطی غیر دقیق، روش $BFGS$ همگرای سراسری است، در حالیکه این مطلب در مورد روش DFP هنوز اثبات نشده است.

تذکر ۱-۵-۱۰: با توجه به تذکر ۱-۵-۶ در صورتیکه از ماتریس B_{k+1} برای تقریب ماتریس هسی G_{k+1} استفاده کنیم، معادله سکانت به‌صورت زیر خواهد بود:

$$B_{k+1}s_k = y_k. \qquad (۶.۳.۵.۱)$$

راه دیگری که برای تبدیل یک فرمول بهنگام سازی H به فرمولی برای B یا بالعکس وجود دارد، گرفتن

وارون است. در نتیجه اگر از رابطه (۵.۳.۵.۱) وارون بگیریم، فرمول بهنگام سازی زیر را برای ماتریس B_{k+1} داریم:

$$B_{k+1}^{BFGS} = B_k + \frac{y_k y_k{}^T}{s_k{}^T y_k} - \frac{B_k s_k s_k{}^T B_k}{s_k{}^T B_k s_k}. \qquad (۷.۳.۵.۱)$$

فرمول بهنگام سازی اخیر برای ماتریس B_k به فرمول بهنگام سازی $BFGS$ مشهور است. به‌طور مشابه گرفتن وارون از رابطه بهنگام سازی (۳.۳.۵.۱) نتیجه می‌دهد:

$$B_k^{DFP} = B_k + \left(1 + \frac{s_k{}^T B_k s_k}{s_k{}^T y_k}\right)\frac{y_k y_k{}^T}{y_k{}^T s_k} - \left(\frac{y_k s_k{}^T B_k + B_k s_k y_k{}^T}{s_k{}^T y_k}\right). \qquad (۸.۳.۵.۱)$$

لازم به ذکر است که اگر در رابطه (۵.۳.۵.۱) نقش H را با B، و نقش s را با y عوض کنیم، رابطه بهنگام سازی(۸.۳.۵.۱) به‌دست می‌آید، که در اصطلاح به این عمل مکمل گیری گفته می‌شود. به‌طور مشابه مکمل رابطه(۳.۳.۵.۱) نیز رابطه بهنگام سازی (۷.۳.۵.۱) خواهد بود. [۲]

۱-۶ جستجوی خطی

جستجوی خطی در واقع روشی برای تعیین طول گام α_k در فرایند تکراری (۲.۵.۱) می‌باشد. بدیهی‌ترین روش انتخاب طول گام α_k، جستجوی خطی دقیق می‌باشد، یعنی $\phi(\alpha)$

$$\alpha_k = \underset{\alpha>0}{Arg\min}\ f(x_k + \alpha d_k).$$

اما به‌دلیل پرهزینه بودن محاسبه دقیق α_k ، می‌توان α_k را یک تقریب از مینیمم کننده دقیق مسئله فوق در نظر گرفت که منجر به جستجوی خطی غیر دقیق می گردد. البته در جهت برقراری همگرایی موضعی یا سراسری مجبوریم شرایطی را به α_k تحت عنوان شرایط جستجوی خطی اعمال کنیم. جستجوی خطی در حقیقت به معنی بهینه سازی تک بعدی است. از آنجا که تابع $\phi(\alpha) = f(x_k + \alpha d_k)$ یک تابع تک متغیره است بنابراین مسئله ما به مسئله پیدا کردن مینیمم تابع تک متغیره $\phi(\alpha)$ تبدیل می‌شود، و اگر این تابع مشتق‌پذیر باشد باید یک صفر از $\phi'(\alpha)$ را بیابیم. بنابراین یک دسته از روش‌های جستجوی خطی، همان روش‌های حل معادلات غیرخطی هستند.

روش‌های جستجوی خطی را می‌توان به سه دسته کلی زیر تقسیم کرد، که هر یک از دسته ها خود شامل روش‌های مختلفی می‌باشند.

۱.روش‌های بدون استفاده از مشتق: جستجوی یکنواخت، نسبت طلایی، جستجوی فیبوناچی.

۲.روش‌های استفاده کننده از مشتق: تنصیف، نیوتن، سکانت، درونیابی.

۳.روش‌های عملی و ابتکاری: جستجوی خطی غیر دقیق، جستجوی خطی با برازش منحنی.

هر کدام از روش‌های بیان شده دارای ویژگی‌های خاص خود بوده و نیز دارای خواص همگرایی ویژه ای می‌باشند و برای حل دسته ای از مسائل بهینه‌سازی نامقید مناسب هستند، که این دلیل اصلی توسعه روش‌های جستجوی خطی می‌باشد.

شرایط جستجوی خطی

چون جستجوی خطی دقیق از نظر محاسباتی پرهزینه می‌باشد، از اینرو اغلب استراتژی ها بر اساس جستجوی خطی غیر دقیق استواراند، به‌طوریکه برای محاسبه یک تقریب خوب α هزینه کمی نیاز باشد. دسته ای از روش‌های جستجوی خطی موسوم به *Typical line search* یک دنباله از کاندیدها را برای α معرفی می‌کنند و یک روند تکراری را تا جایی ادامه می‌دهند که الگوریتم به یک مقدار مناسب α دست یابد. این جستجوها عموماً به دو مرحله تقسیم می‌شوند:

۱) پیمایش معکوس[1]

۲) درونیابی[2]

در مرحله اول، یک بازه پذیرفتنی شامل α_k پیدا می‌شود و در گام دوم از بازه تولید شده α، طول گام مناسب انتخاب می‌گردد. جستجوهای خطی تقریبی در اکثر موارد کارایی نسبتاً خوبی دارند، اما ممکن است خواص مطلوب جستجوهای خطی دقیق را نداشته باشند و احتمالاً موجب ازدست رفتن ویژگی‌های تئوریک الگوریتم مانند همگرایی سراسری، همگرایی موضعی و نوع همگرایی شوند. هدف ما تقریب مناسب α_k است، اما باید α_k را به‌گونه‌ای تقریب بزنیم که خواص α_k تقریبی، شبیه α_k دقیق باشد یا حداقل شرایط مورد نظر را داشته‌باشد. مهم‌ترین شرطی که روی α_k اعمال می‌شود این است که نیاز داریم یک کاهش مناسب در f اتفاق بیفتد، یعنی $f(x_k + \alpha_k d_k) \le f(x_k)$. اما می‌توان نشان داد که این شرط کافی نیست و ممکن است با اعمال این شرط دنباله از همگرا بودن به مینیمم کننده خود عاجز باشد. برای جلوگیری از این مشکلات نیاز داریم شرایطی اعمال کنیم که کاهش کافی را ایجاد کند. در ادامه به بیان چند مورد از این شرایط می‌پردازیم.

1-*Backtracking*
2-*Interpolation*