حل عددی مسایل کنترل بهینه غیرخطی

تئوری کنترل بهینه:

$$\dot{\vec{x}}(t) = \vec{a}\left(\vec{x}(t), \vec{u}(t), t\right)$$

$$J(\vec{u}(t)) = h(\vec{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\vec{x}(t), \vec{u}(t), t)\, dt$$

$$\mathcal{H} = g(\vec{x}, \vec{u}, t) + \vec{p}^T \vec{a}(\vec{x}, \vec{u}, t)$$

$$\delta J = \int_{t_0}^{t_f} \left[(\frac{\partial \mathcal{H}}{\partial \vec{x}} + \dot{\vec{p}})^T \delta\vec{x} + (\frac{\partial \mathcal{H}}{\partial \vec{u}})^T \delta\vec{u} + (\frac{\partial \mathcal{H}}{\partial \vec{p}} - \dot{\vec{x}})^T \delta\vec{p} \right] dt$$
$$\left(h_{\vec{x}} - \vec{p}\right)^T_{*,t_f} \delta\vec{x}_f + \left(\mathcal{H} + h_t\right)_{*,t_f} \delta t_f$$

$$\dot{\vec{x}} = \frac{\partial \mathcal{H}}{\partial \vec{p}} = \vec{a}(\vec{x}, \vec{u}, t)$$
$$\dot{\vec{p}} = -\frac{\partial \mathcal{H}}{\partial \vec{x}}$$
$$\vec{0} = \frac{\partial \mathcal{H}}{\partial \vec{u}} \qquad \Rightarrow \qquad \vec{u} = \vec{u}(\vec{x}, \vec{p}, t)$$

مسائل مقدار مرزی دو نقطه‌ای Two Point Boundary Value Problem TPBVP

شروط مرزی $\left(h_{\vec{x}} - \vec{p}\right)^T_{*,t_f} \delta\vec{x}_f + \left(\mathcal{H} + h_t\right)_{*,t_f} \delta t_f = 0$

لزوما حل تحلیلی برای این معادلات پیدا نمی شود.

دو روش کلی حل عددی

- بهینه سازی مستقیم Direct Optimization
  - مستقیما مساله کنترل بهینه به بهینه سازی تبدیل شود
  - Nonlinear programming (unconstrained/constrained)
- حل عددی مساله مقدارمرزی دونقطه ای Indirect Optimization
  - (عملا مساله بهینه شده، فقط باید حل شود)

بهینه سازی مستقیم

مساله:

$$\min f(\mathbf{X})$$

پارامتربندی مساله

روش های مختلف تبدیل مساله کنترل بهینه به بهینه سازی

- **گسسته سازی**

تابع در بازه $[t_0, t_f]$ تبدیل به پارامتر در زمان های

$$T_a = \begin{pmatrix} t_0 \\ t_1 \\ \cdots \\ t_N \end{pmatrix}; \qquad t_i = i\Delta t; \qquad \Delta t = \frac{t_f - t_0}{N}$$

پس متغیرهای بهینه سازی (فعلا)

$$\mathbf{X} = \mathbf{X}_{\text{opt}} = \begin{pmatrix} \vec{u}_0 \\ \vec{u}_1 \\ \vdots \\ \vec{u}_N \end{pmatrix} = \begin{pmatrix} \vec{u}(t_0) \\ \vec{u}(t_1) \\ \vdots \\ \vec{u}(t_N) \end{pmatrix}$$

هرچند $\vec{u}_N$ تاثیری در جواب ندارد (البته بستگی به نوع میانیابی در بازه نهایی دارد)

تنظیم بردار می تواند متفاوت باشد. مثلا اول بردارهای اولین u در زمان های مختلف و بعد دومین u و ...

مساله بهینه سازی تابع $m(N+1)$ مجهول (اگر شرایط مرزی هم مجهول باشند، به پارامترهای بهینه سازی اضافه می شوند)

- **با دانستن شکل جواب**

جواب تقریبی (Approximate solution)

تابع زمان

$$\vec{u}(t) = \vec{a} + \vec{b}t + \vec{c}t^2 + \cdots$$

$$\vec{u}(t) = \vec{a}\sin t + \vec{b}\cos t + \vec{c}\sinh t + \cdots$$

$$\vec{u}(t) = \vec{a}\sin\vec{\omega}_1 t + \vec{b}\cos\vec{\omega}_1 t + \vec{c}\sin\vec{\omega}_2 t + \vec{d}\cos\vec{\omega}_2 t + \cdots$$

این بردارهای مجهول تشکیل بردار پارامترهای بهینه سازی $\mathbf{X}$ را می دهد

تابع متغیرهای حالت یا پارامترهای مساله

$$\vec{u}(t) = \vec{a} + K\vec{x} + \cdots$$

مزیت: تعداد پارامترها بسیار کمتر

ایراد: جواب تقریبی suboptimal solution

شرط بهینگی بهینه سازی نامقید

Unconstrained Optimization

در فرمول های بهینه سازی به جای J از f استفاده می شود (کتاب مرجع)

$$\frac{\partial J}{\partial \mathbf{X}} = \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} = \vec{\nabla} f = \vec{0}$$

روش های حل نیاز به محاسبه تابع هزینه و بعضا گرادیان دارند (حداقل برای شرط توقف)

شرط توقف

در خیلی از موارد (مخصوصا روش های گرادیانی):

$$\left\| \frac{\partial J}{\partial \mathbf{X}_i} \right\| = \left\| \frac{\partial f}{\partial \mathbf{X}_i} \right\| \le \varepsilon$$

که i قدم بهینه سازی است. یا شروط زیر

$$\left\| \mathbf{X}_{i+1} - \mathbf{X}_i \right\| \le \varepsilon$$

$$\left| \frac{f(\mathbf{X}_{i+1}) - f(\mathbf{X}_i)}{f(\mathbf{X}_i)} \right| \le \varepsilon$$

نحوه محاسبه تابع هزینه

گسسته سازی؟

انتگرال مستطیلی:

$$J(\mathbf{X}) \cong h(\vec{x}(t_f), t_f) + \sum_{i=0}^{N-1} g(\vec{x}(t_i), \vec{u}(t_i), t_i)\,\Delta t$$

انتگرال ذوزنقه ای:

$$J(\mathbf{X}) \cong h(\vec{x}(t_f),t_f)+\sum_{i=0}^{N-1}\frac{\Delta t}{2}\Big(g(\vec{x}(t_i),\vec{u}(t_i),t_i)+g(\vec{x}(t_{i+1}),\vec{u}(t_{i+1}),t_{i+1})\Big)$$
$$\cong h(\vec{x}(t_f),t_f)+\frac{\Delta t}{2}g(\vec{x}(t_0),\vec{u}(t_0),t_0)+\sum_{i=1}^{N-1}g(\vec{x}(t_i),\vec{u}(t_i),t_i)\Delta t+\frac{\Delta t}{2}g(\vec{x}(t_N),\vec{u}(t_N),t_N)$$

یا روش های مرتبه بالاتر در کنار حل معادلات دیفرانسیل سیستم:

$$\vec{x}(t_0)=\vec{x}_0$$

$$\dot{\vec{x}}=a(\vec{x}(t),\vec{u}(t),t)$$

تعریف متغیر جدید

$$\dot{x}_{n+1}=g(x(t),u(t),t)$$

$$x_{n+1}(t_0)=0$$

پس انتگرال می شود:

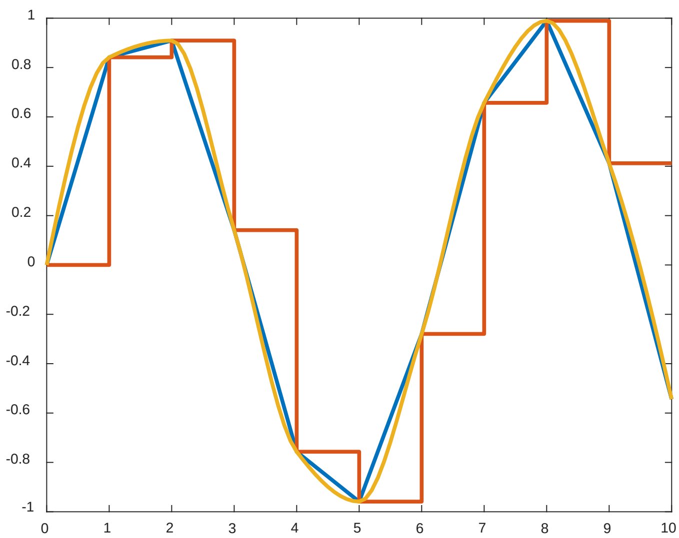
$$x_{n+1}(t_f)=\int_{t_0}^{t_f}g(x(t),u(t),t)\,dt$$

و تابع هزینه:

$$J(\mathbf{X})=h(\vec{x}(t_f),t_f)+x_{n+1}(t_f)$$

نحوه محاسبه u در میان مسیر

میانیابی



گرادیان تابع هزینه نسبت به کنترل

گرادیان:

$$\frac{\partial J}{\partial \mathbf{X}} = \begin{pmatrix} \frac{\partial J}{\partial \vec{u}_0} \\ \frac{\partial J}{\partial \vec{u}_1} \\ \vdots \\ \frac{\partial J}{\partial \vec{u}_N} \end{pmatrix}$$

روش finite difference

$$\left.\frac{\partial J}{\partial u}\right|_i = \frac{\partial f}{\partial \mathrm{x}_i} = \frac{f(\mathbf{X} + \Delta \vec{\mathbf{x}}_i) - f(\mathbf{X})}{\Delta \mathrm{x}_i}$$

منظور از $\frac{\partial f}{\partial \mathrm{x}_i} = \left.\frac{\partial J}{\partial u}\right|_i$ المان iامین بردار گرادیان است (نسبت به یکی از کنترل ها در زمان متناسب)

منظور از $\Delta \vec{\mathbf{x}}_i$ برداری است که فقط مولفه کنترل iام آن مقدار $\Delta \mathrm{x}_i$ دارد و بقیه صفر هستند.

فرمول بهتر (ولی نیاز به محاسبه دو برابر)

$$\frac{\partial f}{\partial \mathrm{x}_i} = \frac{f(\mathbf{X} + \Delta \vec{\mathbf{x}}_i) - f(\mathbf{X} - \Delta \vec{\mathbf{x}}_i)}{2\Delta \mathrm{x}_i}$$

اگر نیاز به مشتق مرتبه دوم باشد (Hessian)

$$H = \frac{\partial^2 J}{\partial \mathbf{X}^2} = \frac{\partial^2 f}{\partial \mathbf{X}^2}$$

اگر گرادیان تحلیلی موجود باشد

$$\vec{g} = \vec{\nabla} f = \frac{\partial f}{\partial \mathbf{X}} \qquad \Rightarrow \qquad H = \frac{\partial \vec{g}}{\partial \mathbf{X}}$$

$$\frac{\partial \vec{g}}{\partial \mathrm{x}_i} = \frac{\vec{g}(\mathbf{X} + \Delta \vec{\mathbf{x}}_i) - \vec{g}(\mathbf{X})}{\Delta \mathrm{x}_i}$$

اگر گرادیان موجود نباشد

$$\frac{\partial^2 f}{\partial \mathrm{x}_i^2} = \frac{f(\mathbf{X} + \Delta \vec{\mathbf{x}}_i) - 2f(\mathbf{X}) + f(\mathbf{X} - \Delta \vec{\mathbf{x}}_i)}{\Delta \mathrm{x}_i^2}$$

روش شبه تحلیلی با کمک معادلات همیلتونی

زمان نهایی ثابت

اگر متغیرهای حالت رو به جلو با معادله زیر حل شود (u تابع زمان معلوم است):

$$\dot{\vec{x}}(t) = \vec{a}\left(\vec{x}(t), \vec{u}(t), t\right)$$

$$\vec{x}(t_0) = \vec{x}_0$$

سپس متغیرهای شبه حالت رو به عقب

$$\dot{\vec{p}} = -\frac{\partial \mathcal{H}}{\partial \vec{x}}$$

$$\vec{p}(t_f) = \frac{\partial h}{\partial \vec{x}}(t_f)$$

پس باتوجه به رابطه تغییرات:

$$\delta J = \int_{t_0}^{t_f} \left[(\frac{\partial \mathcal{H}}{\partial \vec{x}} + \dot{\vec{p}})^T \delta\vec{x} + (\frac{\partial \mathcal{H}}{\partial \vec{u}})^T \delta\vec{u} + (\frac{\partial \mathcal{H}}{\partial \vec{p}} - \dot{\vec{x}})^T \delta\vec{p} \right] dt$$
$$\left(h_{\vec{x}} - \vec{p}\right)^T_{*,t_f} \delta\vec{x}_f + \left(\mathcal{H} + h_t\right)_{*,t_f} \delta t_f$$

تنها چیزی که می ماند:

$$\delta J = \int_{t_0}^{t_f} \left(\frac{\partial \mathcal{H}}{\partial \vec{u}}\right)^T \delta\vec{u}\ dt$$

اگر به فرم جمع نوشته شود (مثلا انتگرال مستطیلی):

$$\delta J \cong \sum_{i=0}^{N-1} \left(\frac{\partial \mathcal{H}}{\partial \vec{u}}\right)^T \left(\vec{x}(t_i), \vec{u}(t_i), \vec{p}(t_i), t_i\right) \delta\vec{u}\ (t_i) \Delta t$$

و درنتیجه:

$$\frac{\partial J}{\partial \vec{u}_i} = \frac{\partial J}{\partial \vec{u}(t_i)} \cong \frac{\delta J}{\delta \vec{u}(t_i)} = \left(\frac{\partial \mathcal{H}}{\partial \vec{u}}\right)\left(\vec{x}(t_i), \vec{u}(t_i), \vec{p}(t_i), t_i\right)\Delta t$$

یعنی:

$$\frac{\partial J}{\partial \mathbf{X}} = \frac{\partial f}{\partial \mathbf{X}} = \begin{pmatrix} \dfrac{\partial J}{\partial \vec{u}_0} \\ \dfrac{\partial J}{\partial \vec{u}_1} \\ \vdots \\ \dfrac{\partial J}{\partial \vec{u}_N} \end{pmatrix} = \Delta t \begin{pmatrix} \left.\dfrac{\partial \mathcal{H}}{\partial \vec{u}}\right|_{t_0} \\ \left.\dfrac{\partial \mathcal{H}}{\partial \vec{u}}\right|_{t_1} \\ \vdots \\ \left.\dfrac{\partial \mathcal{H}}{\partial \vec{u}}\right|_{t_N} \end{pmatrix}$$

گرادیان نسبت به کنترل و پارامتر

به عنوان نمونه، مساله مینیمم زمان

$$J(\vec{u}(t)) = t_f = \int_{t_0}^{t_f} dt \qquad \text{or} \qquad J(\vec{u}(t)) = t_f^2$$

زمان متغیر و مجهول است. مشکلات، اگر به صورت عادی تقسیم بندی زمان شود؟

تغییر متغیر زمان

$$\tau = \frac{t}{t_f} \qquad 0 \le t \le t_f; \quad 0 \le \tau \le 1$$

تغییر معادلات:

$$\frac{d\vec{x}}{d\tau} = \frac{d\vec{x}}{dt}\frac{dt}{d\tau} = t_f \vec{a}\left(\vec{x}(t), \vec{u}(t), t\right) = \vec{a}_N(\vec{x}(t), \vec{u}(t), t_f, t)$$

تغییر تابع هزینه:

$$\begin{aligned} J(\vec{u}(t)) &= h(\vec{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\vec{x}(t), \vec{u}(t), t)\, dt \\ &= h(\vec{x}(\tau_f), t_f) + \int_0^1 t_f\, g(\vec{x}(t), \vec{u}(t), t)\, d\tau \\ &= h(\vec{x}(1), t_f) + \int_0^1 g_N(\vec{x}(t), \vec{u}(t), t_f, t)\, d\tau \end{aligned}$$

درحقیقت تابعیت عوض شد:

$$J(\vec{u}(t), t_f) = h(\vec{x}(1), t_f) + \int_0^1 g_N(\vec{x}(t), \vec{u}(t), t_f, t)\, d\tau$$

$$\mathbf{X}=\mathbf{X}_{\text{opt}}=\begin{pmatrix}\vec{u}(\tau_0=0)\\ \vec{u}(\tau_1)\\ \vdots\\ \vec{u}(\tau_N=1)\\ t_f\end{pmatrix},\qquad \frac{\partial J}{\partial \mathbf{X}}=\frac{\partial f}{\partial \mathbf{X}}=\begin{pmatrix}\frac{\partial J}{\partial \vec{u}_0}\\ \frac{\partial J}{\partial \vec{u}_1}\\ \vdots\\ \frac{\partial J}{\partial \vec{u}_N}\\ \frac{\partial J}{\partial t_f}\end{pmatrix}$$

روش finite difference

فرمول کلی مشابه قبل

$$\frac{\partial f}{\partial \mathrm{x}_i}=\frac{f(\mathbf{X}+\Delta\vec{\mathrm{x}}_i)-f(\mathbf{X})}{\Delta \mathrm{x}_i}$$

الان یکی از x_iها همان زمان نهایی است

روش شبه تحلیلی با کمک معادلات همیلتونی

حال اگر تابع هزینه به فرم کلی زیر داشته باشیم:

$$J(\vec{u}(t),\vec{x}_p)=h(\vec{x}(t_f),t_f,\vec{x}_p)+\int_{t_0}^{t_f} g(\vec{x}(t),\vec{u}(t),\vec{x}_p,t)\,d\tau$$

که $\vec{x}_p$ پارامترهای ثابت هستند (در مثال قبل $\vec{x}_p$ فقط یک متغیر t_f بود) و معادلات هم فرم کلی زیر را داشته باشند:

$$\dot{\vec{x}}=\vec{a}\,(\vec{x}(t),\vec{u}(t),\vec{x}_p,t)$$

مشابه قبل تابع هزینه augmented تشکیل می شود و δJ (ترم تغییرات خطی variation) نسبت به متغیرهای $\underbrace{\vec{x},\vec{u},\vec{x}_p,\vec{p}}_{\vec{x}_{aug}}$ بدست می آید.

با تعریف

$$\mathcal{H}(\vec{x},\vec{u},\vec{x}_p,\vec{p},t)=g(\vec{x},\vec{u},\vec{x}_p,t)+\vec{p}^{\,T}\vec{a}(\vec{x},\vec{u},\vec{x}_p,t)$$

مشابه قبل می شود

$$\delta J=\int_{t_0}^{t_f}\left[(\frac{\partial\mathcal{H}}{\partial\vec{x}}+\dot{\vec{p}})^T\,\delta\vec{x}+(\frac{\partial\mathcal{H}}{\partial\vec{u}})^T\delta\vec{u}+(\frac{\partial\mathcal{H}}{\partial\vec{x}_p})^T\delta\vec{x}_p+(\frac{\partial\mathcal{H}}{\partial\vec{p}}-\dot{\vec{x}})^T\delta\vec{p}\right]dt$$
$$\left(h_{\vec{x}}-\vec{p}\right)_{*,t_f}^T\delta\vec{x}_f+\left(\mathcal{H}+h_t\right)_{*,t_f}\delta t_f+h_{\vec{x}_p}^T\delta\vec{x}_p$$

اگر متغیرهای حالت رو به جلو با معادله زیر حل شود (u تابع زمان معلوم است):

$$\dot{\vec{x}}(t)=\vec{a}\left(\vec{x}(t),\vec{u}(t),t\right)$$

$$\vec{x}(t_0)=\vec{x}_0$$

سپس متغیرهای شبه حالت رو به عقب (زمان نهایی ثابت)

$$\dot{\vec{p}}=-\frac{\partial\mathcal{H}}{\partial\vec{x}}$$

$$\vec{p}(t_f)=\frac{\partial h}{\partial\vec{x}}(t_f)$$

پس باتوجه به رابطه تغییرات تنها چیزی که می ماند:

$$\delta J=\int_{t_0}^{t_f}\left(\frac{\partial\mathcal{H}}{\partial\vec{u}}\right)^T\delta\vec{u}\;dt+\left[\frac{\partial h}{\partial\vec{x}_p}+\int_{t_0}^{t_f}\frac{\partial\mathcal{H}}{\partial\vec{x}_p}dt\right]^T\delta\vec{x}_p$$

مشابه قبل:

$$\frac{\partial J}{\partial\vec{u}_i}=\frac{\partial J}{\partial\vec{u}(t_i)}\cong\frac{\delta J}{\delta\vec{u}(t_i)}=\left(\frac{\partial\mathscr{H}}{\partial\vec{u}}\right)\left(\vec{x}(t_i),\vec{u}(t_i),\vec{p}(t_i),t_i\right)\Delta t$$

و نسبت به پارامترهای ثابت $\vec{x}_p$:

$$\frac{\partial J}{\partial\vec{x}_p}=\frac{\partial h}{\partial\vec{x}_p}+\int_{t_0}^{t_f}\frac{\partial\mathcal{H}}{\partial\vec{x}_p}dt$$

یعنی:

$$\frac{\partial J}{\partial\mathbf{X}}=\frac{\partial f}{\partial\mathbf{X}}=\begin{pmatrix}\dfrac{\partial J}{\partial\vec{u}_0}\\ \dfrac{\partial J}{\partial\vec{u}_1}\\ \vdots\\ \dfrac{\partial J}{\partial\vec{u}_N}\\ \dfrac{\partial J}{\partial\vec{x}_p}\end{pmatrix}=\begin{pmatrix}\left.\dfrac{\partial\mathcal{H}}{\partial\vec{u}}\right|_{t_0}\Delta t\\ \left.\dfrac{\partial\mathcal{H}}{\partial\vec{u}}\right|_{t_1}\Delta t\\ \vdots\\ \left.\dfrac{\partial\mathcal{H}}{\partial\vec{u}}\right|_{t_N}\Delta t\\ \dfrac{\partial h}{\partial\vec{x}_p}+\displaystyle\int_{t_0}^{t_f}\frac{\partial\mathcal{H}}{\partial\vec{x}_p}dt\end{pmatrix}$$

انواع روش حل بهینه سازی نامقید

جستجوی مستقیم	جستجوی کاهشی (گرادیانی)
Direct search methods[a]	Descent methods[b]
Random search method	Steepest descent (Cauchy) method
Grid search method	Fletcher–Reeves method
Univariate method	Newton's method
Pattern search methods	Marquardt method
Powell's method	Quasi-Newton methods
	Davidon–Fletcher–Powell method
	Broyden–Fletcher–Goldfarb–Shanno method
Simplex method	

فرم کلی حل

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i^* \mathbf{S}_i$$

$\mathbf{X}_i$ نقطه شروع این قدم

$\mathbf{S}_i$ راستای جستجوی نقطه بعد

$\mathbf{X}_{i+1}$ نقطه بعدی

λ_i^* طول این قدم در راستای $\mathbf{S}_i$ (چون معمولا بهینه می شود با علامت * نشان داده می شود)

تفاوت روش های بالا در پیدا کردن جهت جستجو (Search Direction)

پیدا کردن طول قدم (جستجوی خطی Line Search)

$$f_{i+1} = f(\mathbf{X}_i + \lambda_i^* \mathbf{S}_i) = \min_{\lambda_i} f(\mathbf{X}_i + \lambda_i \mathbf{S}_i)$$

نرخ تغییرات تابع در راستای بردار جستجو

$$\mathbf{X} = \mathbf{X}_i + \lambda \mathbf{S}_i$$

$$\frac{df}{d\lambda} = \sum_{j=1}^{n} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial \lambda}$$

$$\frac{\partial x_j}{\partial \lambda} = \frac{\partial}{\partial \lambda}(x_{ij} + \lambda s_{ij}) = s_{ij}$$

$$\frac{df}{d\lambda} = \sum_{j=1}^{n} \frac{\partial f}{\partial x_j} s_{ij} = \nabla f^{\mathrm{T}} \mathbf{S}_i$$

برای پیدا کردن طول بهینه (تحلیلی)

$$\left.\frac{df}{d\lambda}\right|_{\lambda=\lambda^*} = \nabla f|_{\lambda^*}^{\mathrm{T}} \mathbf{S}_i = 0$$

نکته: در نزدیکی جواب تقریب زیر برقرار است:

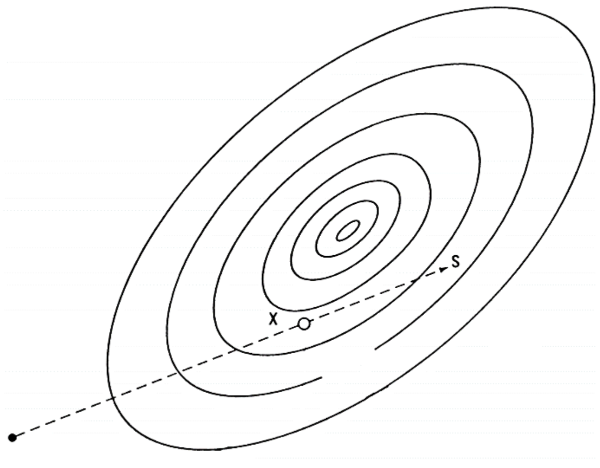
$$f(\mathbf{X}) = c + \mathbf{B}^{\mathrm{T}}\mathbf{X} + \tfrac{1}{2}\mathbf{X}^{\mathrm{T}}[A]\mathbf{X}$$

که

$$c = f(\mathbf{X}_i)$$

$$B = \begin{Bmatrix} \left.\dfrac{\partial f}{\partial x_1}\right|_{\mathbf{X}_i} \\ \vdots \\ \left.\dfrac{\partial f}{\partial x_n}\right|_{\mathbf{X}_i} \end{Bmatrix}$$

$$[A] = \begin{bmatrix} \left.\dfrac{\partial^2 f}{\partial x_1^2}\right|_{\mathbf{X}_i} & \cdots & \left.\dfrac{\partial^2 f}{\partial x_1 \partial x_n}\right|_{\mathbf{X}_i} \\ \vdots & & \vdots \\ \left.\dfrac{\partial^2 f}{\partial x_n \partial x_1}\right|_{\mathbf{X}_i} & \cdots & \left.\dfrac{\partial^2 f}{\partial x_n^2}\right|_{\mathbf{X}_i} \end{bmatrix}$$



روش های جستجوی مستقیم Direct Search Methods

جستجوی تصادفی Random search methods

روش پرش تصادفی Random Jumping

$$l_i \le x_i \le u_i, \quad i = 1, 2, \ldots, n$$

اعداد تصادفی r_i در بازه $[0,1]$ تولید می شوند:

$$\mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} l_1 + r_1(u_1 - l_1) \\ l_2 + r_2(u_2 - l_2) \\ \vdots \\ l_n + r_n(u_n - l_n) \end{Bmatrix}$$

تعداد زیادی عدد تصادفی تولید شود و بهترین نقطه جواب است

همگرایی تضمینی ندارد.

روش قدم تصادفی Random Walk

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i^* \mathbf{u}_i$$

بردار $\mathbf{u}_i$ بردار یکه با جهت تصادفی

$$\mathbf{u} = \frac{1}{(r_1^2 + r_2^2 + \cdots + r_n^2)^{1/2}} \begin{Bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{Bmatrix}$$

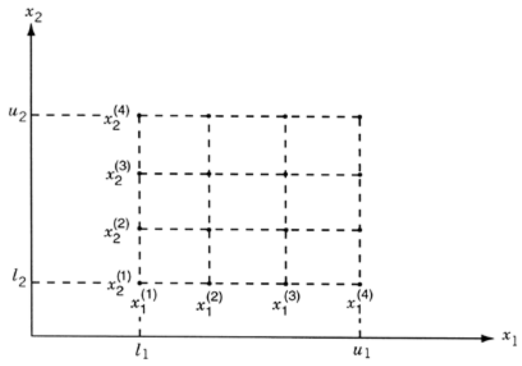
بهتر است طول قدم بهینه سازی شود

$$f_{i+1} = f(\mathbf{X}_i + \lambda_i^* \mathbf{u}_i) = \min_{\lambda_i} f(\mathbf{X}_i + \lambda_i \mathbf{u}_i)$$

همگرایی تضمینی ندارد.

جستجوی یک بازه Grid Search

هزینه محاسباتی زیاد



جستجو در راستاهای یکه Univariate Search Method

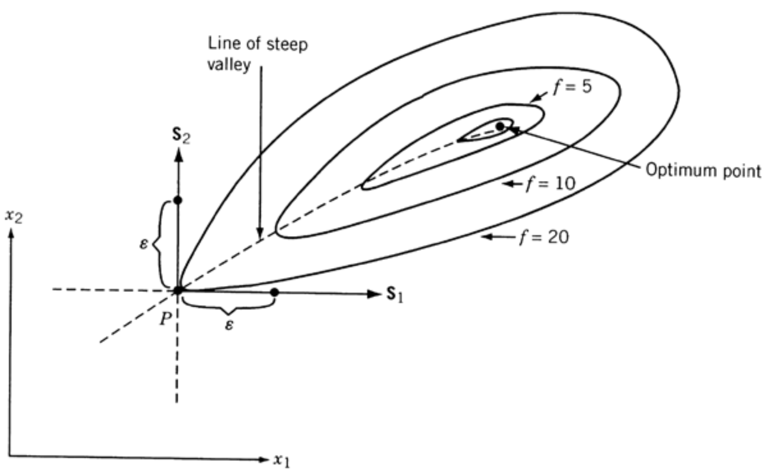
در هر قدم فقط نسبت به یکی از پارامترها بهینه سازی شود

$$\mathbf{S}_i^{\mathrm{T}} = \begin{cases} (1,0,0,\ldots,0) & \text{for} \quad i = 1, n+1, 2n+1, \ldots \\ (1,0,0,\ldots,0) & \text{for} \quad i = 2, n+2, 2n+2, \ldots \\ (0,0,1,\ldots,0) & \text{for} \quad i = 3, n+3, 2n+3, \ldots \\ \quad\vdots & \\ (0,0,0,\ldots,1) & \text{for} \quad i = n, 2n, 3n, \ldots \end{cases}$$

جهت می تواند مثبت یا منفی باشد.

بهینه سازی طول قدم

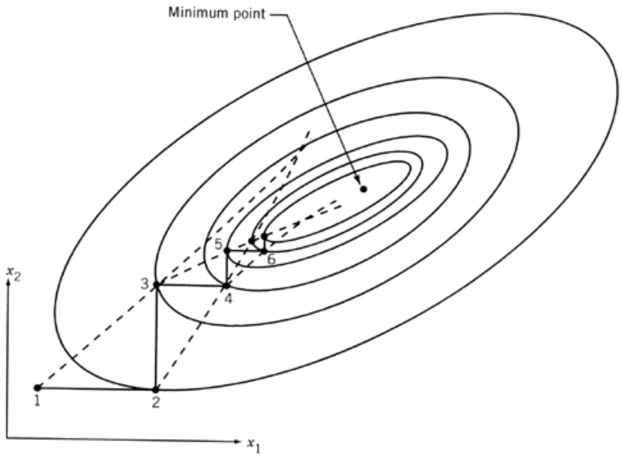
$$f(\mathbf{X}_i \pm \lambda_i^* \mathbf{S}_i) = \min_{\lambda_i}(\mathbf{X}_i \pm \lambda_i \mathbf{S}_i)$$



روش های جستجوی الگو Pattern Search

به عنوان نمونه

$$\mathbf{S}_i = \mathbf{X}_i - \mathbf{X}_{i-n}$$



روش Powell

بر مبنای راستاهای Conjugate

$$\mathbf{S}_i^{\mathrm{T}}\mathbf{A}\mathbf{S}_j = 0 \quad \text{for} \quad \text{all } i \neq j, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n$$

برای یک تابع هزینه درجه ۲، اگر از دو نقطه مختلف در راستای S جستجو شود، تفاضل دو بردار جواب با راستای بردار جستجو Conjugate می شوند

$$Q(\mathbf{X}) = \tfrac{1}{2}\mathbf{X}^{\mathrm{T}}\mathbf{A}\mathbf{X} + \mathbf{B}^{\mathrm{T}}\mathbf{X} + C$$

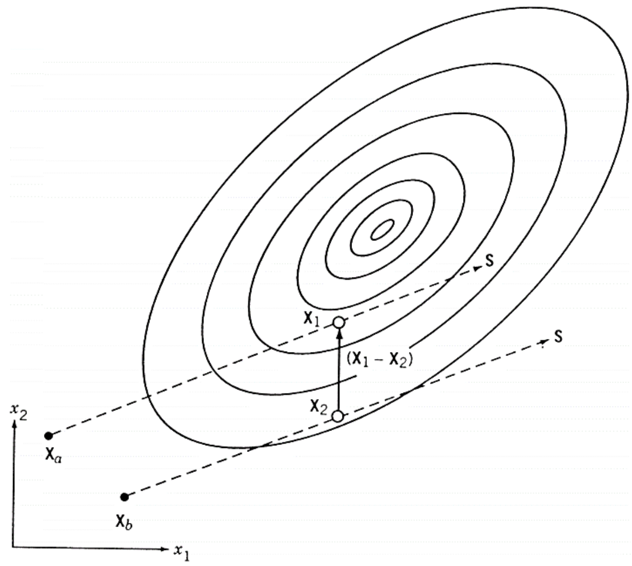
گرادیان تابع

$$\nabla Q(\mathbf{X}) = \mathbf{A}\mathbf{X} + \mathbf{B}$$

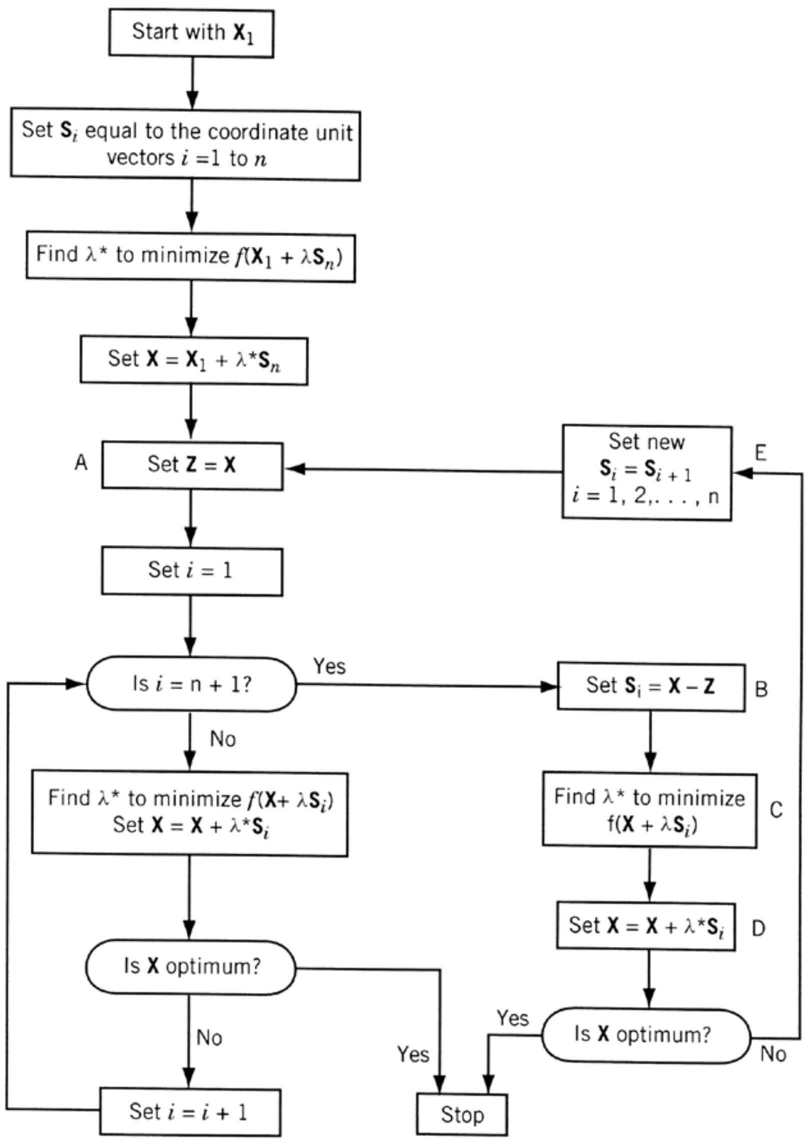
برای پیدا کردن جواب

$$\mathbf{S}^{\mathrm{T}}\nabla Q(\mathbf{X}_1) = \mathbf{S}^{\mathrm{T}}\mathbf{A}\mathbf{X}_1 + \mathbf{S}^{\mathrm{T}}\mathbf{B} = 0$$
$$\mathbf{S}^{\mathrm{T}}\nabla Q(\mathbf{X}_2) = \mathbf{S}^{\mathrm{T}}\mathbf{A}\mathbf{X}_2 + \mathbf{S}^{\mathrm{T}}\mathbf{B} = 0$$

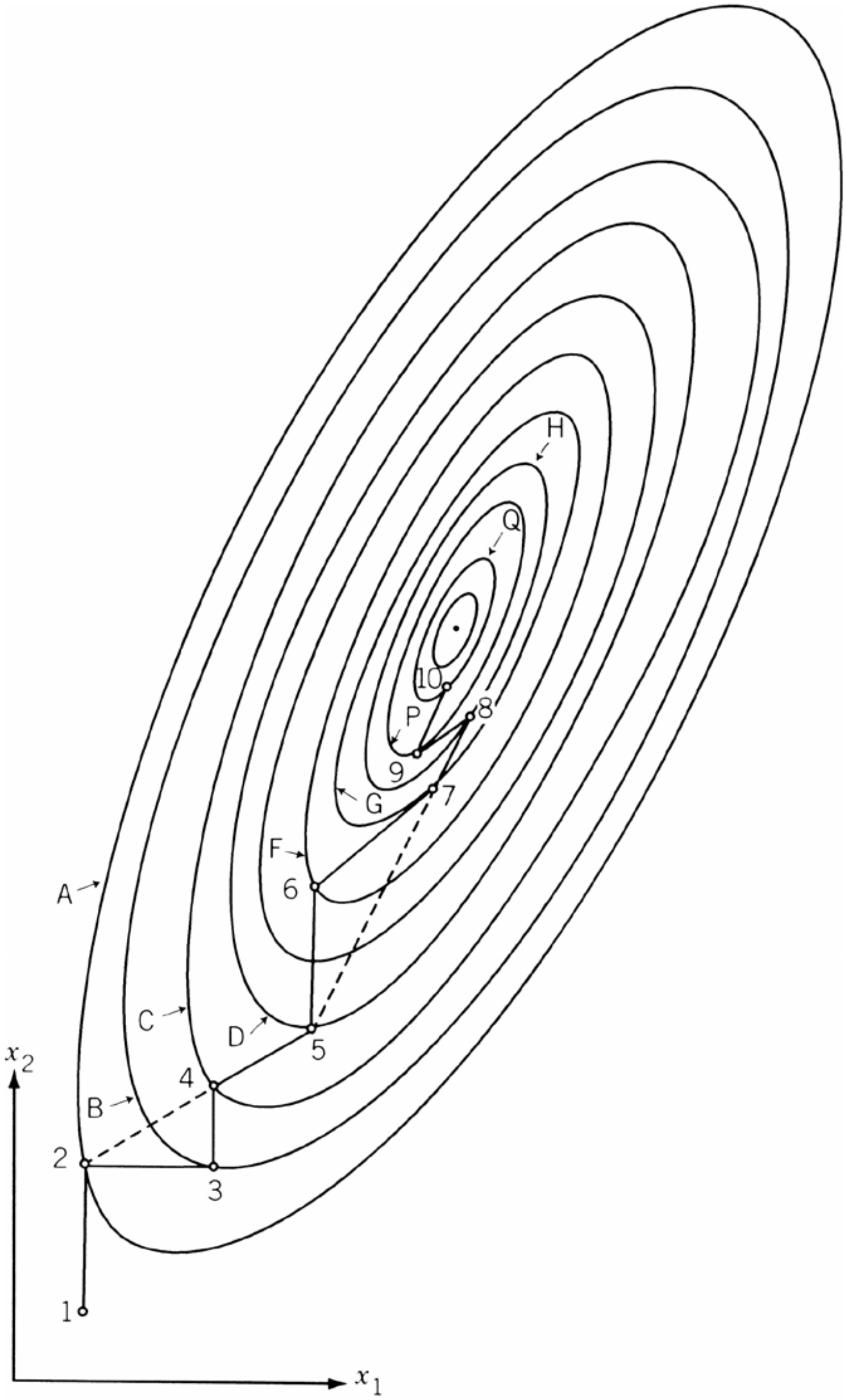
$$\mathbf{S}^{T}\mathbf{A}(\mathbf{X}_1 - \mathbf{X}_2) = \mathbf{0}$$



الگوریتم روش Powell



نمونه مسیر جواب برای روش Powell



روش های گرادیانی

Indirect Search/Descent/Gradient Based Methods

جهت گرادیان بیشترین تغییرات است.

اگر حول نقطه $\mathbf{X}$ با بردار $d\mathbf{X}$ با طول ds بگردیم:

$$d\mathbf{X} = \begin{Bmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_n \end{Bmatrix}$$

$$d\mathbf{X}^T d\mathbf{X} = (ds)^2 = \sum_{i=1}^{n} (dx_i)^2$$

$$d\mathbf{X} = \mathbf{u}\ ds$$

تغییر مقدار تابع هزینه

$$df = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} dx_i = \nabla f^T\ d\mathbf{X}$$

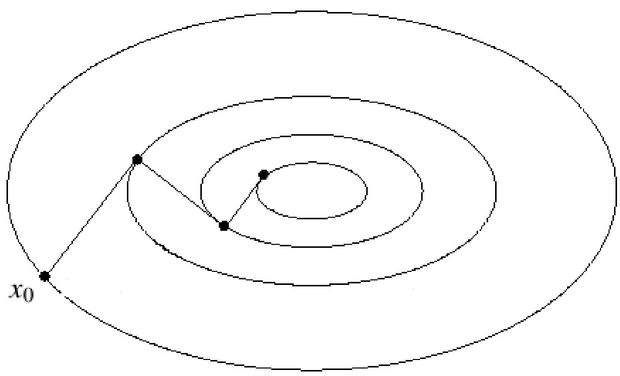
$$\frac{df}{ds} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}\frac{dx_i}{ds} = \nabla f^T \frac{d\mathbf{X}}{ds} = \nabla f^T \mathbf{u}$$

$$\frac{df}{ds} = \|\nabla f\|\ \ \|\mathbf{u}\| \cos\theta$$

روش کوشی/سریعترین شیب Steepest Decent/Cauchy

$$\mathbf{S}_i = -\nabla f_i = -\nabla f(\mathbf{X}_i)$$

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i^* \mathbf{S}_i = \mathbf{X}_i - \lambda_i^* \nabla f_i$$



روش Fletcher-Reeves

در قدم اول:

$$\mathbf{S}_1 = -\nabla f(\mathbf{X}_1) = -\nabla f_1$$

در قدم های بعد

$$\mathbf{S}_i = -\nabla f_i + \frac{|\nabla f_i|^2}{|\nabla f_{i-1}|^2}\mathbf{S}_{i-1}$$

$$\nabla f_i = \nabla f(\mathbf{X}_i)$$

اگر تابع هزینه درجه ۲ باشد

$$f(\mathbf{X}) = \tfrac{1}{2}\mathbf{X}^{\mathrm{T}}[\mathbf{A}]\mathbf{X} + \mathbf{B}^{\mathrm{T}}\mathbf{X} + C$$

و در هر قدم طول قدم بهینه شود:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i^* \mathbf{S}_i$$

راستاهای جستجو نسبت به قبلی ها A-Conjugate می شوند

روش نیوتن Newton's method

اگر بخواهیم معادله $\vec{\nabla} f = \vec{0}$ را حل مستقیم کنیم.

حدس اولیه برای و سپس روش نیوتن.

بسط تیلور جواب:

$$f(\mathbf{X}) = f(\mathbf{X}_i) + \nabla f_i^{\mathrm{T}}(\mathbf{X} - \mathbf{X}_i) + \tfrac{1}{2}(\mathbf{X} - \mathbf{X}_i)^{\mathrm{T}}[J_i](\mathbf{X} - \mathbf{X}_i)$$

هدف این است که در هر قدم کاری کنیم که:

$$\frac{\partial f(\mathbf{X})}{\partial x_j} = 0, \qquad j = 1, 2, \ldots, n$$

با بسط تیلور برای گرادیان:

$$\nabla f = \nabla f_i + [J_i](\mathbf{X} - \mathbf{X}_i) = \mathbf{0}$$

معادله روش نیوتن برای جواب:

$$\mathbf{X}_{i+1} = \mathbf{X}_i - [J_i]^{-1} \quad \nabla f_i$$

اگر تابع هزینه درجه ۲ باشد:

$$f(\mathbf{X}) = \frac{1}{2}\mathbf{X}^{\mathrm{T}}[A]\mathbf{X} + \mathbf{B}^{\mathrm{T}}\mathbf{X} + C$$

جواب تحلیلی:

$$\nabla f = [A]\mathbf{X} + \mathbf{B} = \mathbf{0}$$

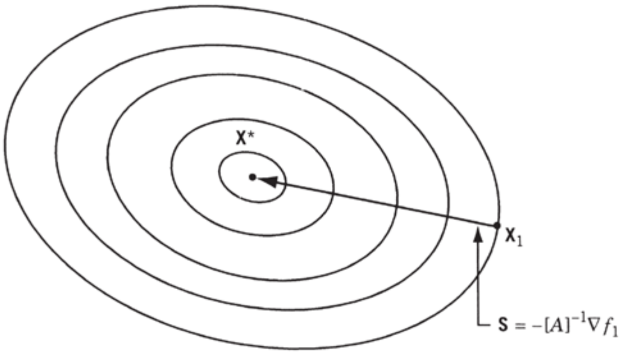
$$\mathbf{X}^* = -[A]^{-1}\mathbf{B}$$

جواب بعد از یک قدم روش نویتن:

$$\mathbf{X}_{i+1} = \mathbf{X}_i - [A]^{-1}([A]\mathbf{X}_i + \mathbf{B})$$

$$\mathbf{X}_{i+1} = \mathbf{X}^* = -[A]^{-1}\mathbf{B}$$

مهم نیست از کجا شروع شود، بعد یک قدم به جواب می رسد



راستای جستجو و فرمول قدم های بعدی در روش نیوتن در حالت عمومی:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i^*\mathbf{S}_i = \mathbf{X}_i - \lambda_i^*[J_i]^{-1}\nabla f_i$$

روش Marquardt

ترکیب روش steepest descent و Newton

$$[\tilde{J}_i] = [J_i] + \alpha_i[I]$$

$$\mathbf{S}_i = -[\tilde{J}_i]^{-1}\nabla f_i$$

α_i جوری انتخاب می شود که $[\tilde{J}_i]$ مثبت معین باشد

اگر α_i خیلی بزرگ انتخاب شود:

$$[\tilde{J}_i]^{-1} = [[J_i] + \alpha_i[I]]^{-1} \approx [\alpha_i[I]]^{-1} = \frac{1}{\alpha_i}[I]$$

در این روش برای شروع از α_i بزرگ شروع می شود.

- اگر جهت جستجو در جهت کاهش تابع هزینه باشد، α_i کوچک می شود (که شبیه نیوتن شود)
- اگر جهت جستجو در جهت افزایش تابع هزینه باشد، α_i بزرگ می شود (که شبیه کوشی شود و درنتیجه حتما تابع هزینه کاهشی باشد)

روش های شبه نیوتن

مبنای روش همان روش نیوتن است

$$\mathbf{X}_{i+1} = \mathbf{X}_i - [J_i]^{-1}\nabla f(\mathbf{X}_i)$$

ولی تخمینی از هشن یا معکوس آن استفاده می شود:

$$\mathbf{S}_i = -[B_i]\nabla f(\mathbf{X}_i)$$

$$\mathbf{X}_{i+1} = \mathbf{X}_i - \lambda_i^*[B_i]\nabla f(\mathbf{X}_i)$$

مبنای تخمین هشن یا معکوس آن (هدف این است که در هر مرحله تخمین بهتری از هشن داشته باشیم):

بسط تیلور:

$$\nabla f(\mathbf{X}) \approx \nabla f(\mathbf{X}_0) + [J_0](\mathbf{X} - \mathbf{X}_0)$$

پس برای دو نقطه مختلف حول نقطه فعلی:

$$\nabla f_{i+1} = \nabla f(\mathbf{X}_0) + [A_i](\mathbf{X}_{i+1} - \mathbf{X}_0)$$
$$\nabla f_i = \nabla f(\mathbf{X}_0) + [A_i](\mathbf{X}_i - \mathbf{X}_0)$$

منظور از $[A_i]$ تخمین هشن است. با تعریف دو متغیر زیر:

$$\mathbf{d}_i = \mathbf{X}_{i+1} - \mathbf{X}_i$$
$$\mathbf{g}_i = \nabla f_{i+1} - \nabla f_i$$

می توان تخمین هشن یا معکوس آن باید در رابطه زیر صدق کند:

$$[A_i]\mathbf{d}_i = \mathbf{g}_i$$

$$\mathbf{d}_i = [B_i]\mathbf{g}_i$$

$$[B_i] = [A_i]^{-1}$$

این مجموعه معادلات بینهایت جواب دارد (برای ماتریس $[A_i]$ یا $[B_i]$)

از یک مقدار شروع می شود (برای اینکه ابتدا مثل کوشی باشد و همچنین متقارن مثبت معین):

$$[A_i] = [B_i] = I$$

بعد با فرمول زیر update می شود:

$$[B_{i+1}] = [B_i] + [\Delta B_i]$$

فرمول Rank 1 update

برای اینکه مطمئن باشیم متقارن و مثبت معین می ماند:

$$[\Delta B_i] = c\mathbf{z}\mathbf{z}^{\mathrm{T}}$$

$$[B_{i+1}] = [B_i] + c\mathbf{z}\mathbf{z}^{\mathrm{T}}$$

برای اینکه تخمین خوبی برای معکوس هشن باشد:

$$\mathbf{d}_i = [B_{i+1}]\mathbf{g}_i$$

$$\mathbf{d}_i = ([B_i] + c\mathbf{z}\mathbf{z}^T)\mathbf{g}_i = [B_i]\mathbf{g}_i + c\mathbf{z}(\mathbf{z}^T\mathbf{g}_i)$$

ساده ترین انتخاب:

$$c\mathbf{z} = \frac{\mathbf{d}_i - [B_i]\mathbf{g}_i}{\mathbf{z}^T\mathbf{g}_i}$$

$$\mathbf{z} = \mathbf{d}_i - [B_i]\mathbf{g}_i$$

$$c = \frac{1}{\mathbf{z}^T\mathbf{g}_i}$$

فرمول update برای معکوس هشن (فرمول Broyden):

$$[B_{i+1}] = [B_i] + [\Delta B_i] \equiv [B_i] + \frac{(\mathbf{d}_i - [B_i]\mathbf{g}_i)(\mathbf{d}_i - [B_i]\mathbf{g}_i)^{\mathrm{T}}}{(\mathbf{d}_i - [B_i]\mathbf{g}_i)^T\mathbf{g}_i}$$

فرمول های Rank 2 update

$$[\Delta B_i] = c_1 \mathbf{z}_1 \mathbf{z}_1^{\mathrm{T}} + c_2 \mathbf{z}_2 \mathbf{z}_2^{\mathrm{T}}$$

$$[B_{i+1}] = [B_i] + c_1 \mathbf{z}_1 \mathbf{z}_1^{\mathrm{T}} + c_2 \mathbf{z}_2 \mathbf{z}_2^{\mathrm{T}}$$

با همان تعریف اصلی

$$\mathbf{d}_i = [B_{i+1}] \mathbf{g}_i$$

$$\mathbf{d}_i = [B_i] \mathbf{g}_i + c_1 \mathbf{z}_1 (\mathbf{z}_1^{\mathrm{T}} \mathbf{g}_i) + c_2 \mathbf{z}_2 (\mathbf{z}_2^{\mathrm{T}} \mathbf{g}_i)$$

یک انتخاب ساده:

$$\mathbf{z}_1 = \mathbf{d}_i$$

$$\mathbf{z}_2 = [B_i] \mathbf{g}_i$$

$$c_1 = \frac{1}{\mathbf{z}_1^{\mathrm{T}} \mathbf{g}_i}$$

$$c_2 = -\frac{1}{\mathbf{z}_2^{\mathrm{T}} \mathbf{g}_i}$$

نهایتا فرمول update می شود (Davidon–Fletcher–Powell (DFP) formula)

$$[B_{i+1}] = [B_i] + [\Delta B_i] \equiv [B_i] + \frac{\mathbf{d}_i \mathbf{d}_i^{\mathrm{T}}}{\mathbf{d}_i^T \mathbf{g}_i} - \frac{([B_i]\mathbf{g}_i)([B_i]\mathbf{g}_i)^{\mathrm{T}}}{([B_i]\mathbf{g}_i)^{\mathrm{T}} \mathbf{g}_i}$$

راه دیگر: همین مسیر برای هشن (به جای معکوس آن)

$$[A_{i+1}] = [A_i] + \frac{\mathbf{g}_i \mathbf{g}_i^{\mathrm{T}}}{\mathbf{g}_i^T \mathbf{d}_i} - \frac{([A_i]\mathbf{d}_i)([A_i]\mathbf{d}_i)^{\mathrm{T}}}{([A_i]\mathbf{d}_i)^T \mathbf{d}_i}$$

Broydon–Fletcher–Goldfarb–Shanno (BFGS) formula

تقریب معکوس آن:

$$[B_{i+1}] = [B_i] + \frac{\mathbf{d}_i \mathbf{d}_i^{\mathrm{T}}}{\mathbf{d}_i^T \mathbf{g}_i} \left(1 + \frac{\mathbf{g}_i^{\mathrm{T}} [B_i] \mathbf{g}_i}{\mathbf{d}_i^T \mathbf{g}_i} \right) - \frac{[B_i] \mathbf{g}_i \mathbf{d}_i^{\mathrm{T}}}{\mathbf{d}_i^T \mathbf{g}_i} - \frac{\mathbf{d}_i \mathbf{g}_i^{\mathrm{T}} [B_i]}{\mathbf{d}_i^T \mathbf{g}_i}$$

فرم کلی (Huang's family of updates)

$$[B_{i+1}] = \rho_i \left([B_i] - \frac{[B_i]\mathbf{g}_i\mathbf{g}_i^{\mathrm{T}}[B_i]}{\mathbf{g}_i^T[B_i]\mathbf{g}_i} + \theta_i \mathbf{y}_i \mathbf{y}_i^T \right) + \frac{\mathbf{d}_i\mathbf{d}_i^{\mathrm{T}}}{\mathbf{d}_i^T\mathbf{g}_i}$$

$$\mathbf{y}_i = (\mathbf{g}_i^{\mathrm{T}}[\mathbf{B}_i]\mathbf{g}_i)^{1/2} \left(\frac{\mathbf{d}_i}{\mathbf{d}_i^T\mathbf{g}_i} - \frac{[B_i]\mathbf{g}_i}{\mathbf{g}_i^T[B_i]\mathbf{g}_i} \right)$$

به ازای $\rho_i = 1$ و $\theta_i = 0$ می شود DFP

به ازای $\rho_i = 1$ و $\theta_i = 1$ می شود BFGS

بهینه سازی مقید

Direct methods	Indirect methods
Random search methods	Transformation of variables technique
Heuristic search methods	Sequential unconstrained minimization techniques
Complex method	Interior penalty function method
Objective and constraint approximation methods	Exterior penalty function method
Sequential linear programming method	Augmented Lagrange multiplier method
Sequential quadratic programming method	
Methods of feasible directions	
Zoutendijk's method	
Rosen's gradient projection method	
Generalized reduced gradient method	

روش های غیرمستقیم Indirect Methods

روش های تغییر متغیر Transformation Techniques

اگر متغیر در یک بازه محدود باشد:

$$l_i \leq x_i \leq u_i$$

$$x_i = l_i + (u_i - l_i)\sin^2 y_i$$

کلا اگر در بازه 0 تا 1 محدود باشد: [0,1]

$$x_i = \sin^2 y_i \qquad x_i = \cos^2 y_i$$

$$x_i = \frac{e^{yi}}{e^{yi} + e^{-yi}} \qquad x_i = \frac{y_i^2}{1 + y_i^2}$$

کلا اگر در بازه 1- تا 1 محدود باشد: [-1,1]

$$x_i = \sin y_i, \quad x_i = \cos y_i, \quad \text{or} \quad x_i = \frac{2y_i}{1 + y_i^2}$$

اگر فقط مقدار مثبت دارد: [0,∞]

$$x_i = \text{abs}(y_i), \quad x_i = y_i^2 \quad \text{or} \quad x_i = e^{yi}$$

روش های تابع جریمه Penalty Function

قیدهای نامساوی

تبدیل مساله:

$$\begin{array}{lll} & \min & f(\mathbf{X}) \\ \text{subject to} & g_j(\mathbf{X}) \leq 0 & j = 1,\ldots,m \end{array}$$

در هرقدم به یک مساله بهینه سازی نامقید:

sequential unconstrained minimization techniques

$$\phi_k = \phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} G_j[g_j(\mathbf{X})]$$

توابع جریمه داخلی Interior Penalty Functions

$$G_j = -\frac{1}{g_j(\mathbf{X})}$$
$$G_j = \log[-g_j(\mathbf{X})]$$

barrier methods

$$\phi(\mathbf{X}, r_k) = f(\mathbf{X}) - r_k \sum_{j=1}^{m} \frac{1}{g_j(\mathbf{X})}$$

$$r_{k+1} = c r_k$$

$$c < 1$$

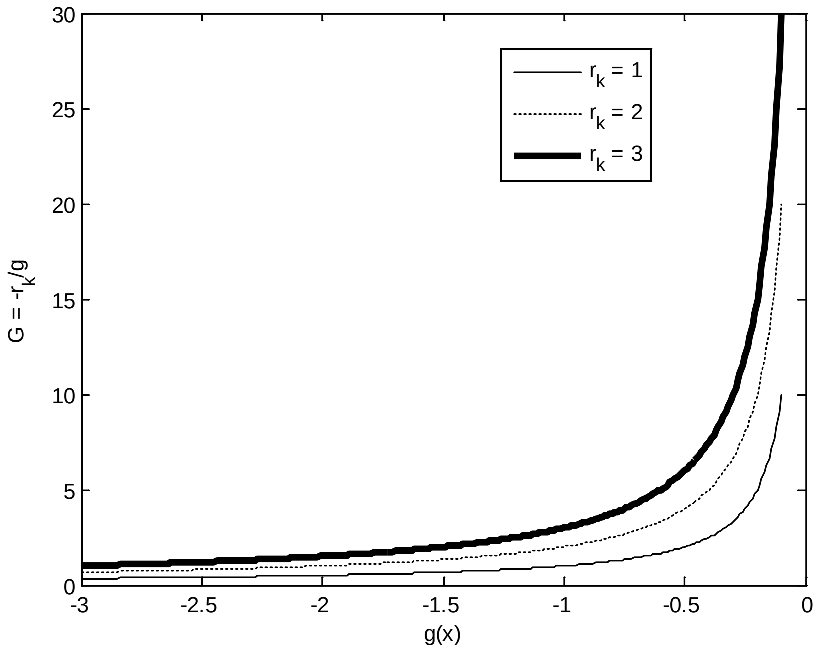
نقطه شروع باید حتما مجاز feasible باشد (قید را ارضا کرده باشد)

مقدار اولیه r_1

$$r_1 \simeq 0.1 \text{ to } 1.0 \frac{f(\mathbf{X}_1)}{-\sum_{j=1}^{m} 1/g_j(\mathbf{X}_1)}$$

مقدار c بستگی به مساله می تواند خیلی کوچک (در حد 0.1) یا بزرگتر (در حد 0.9) باشد.

بستگی به سرعت همگرایی روش اصلی بهینه سازی هم دارد.



توابع جریمه خارجی Exterior Penalty Functions

$$G_j = \max[0, g_j(\mathbf{X})]$$

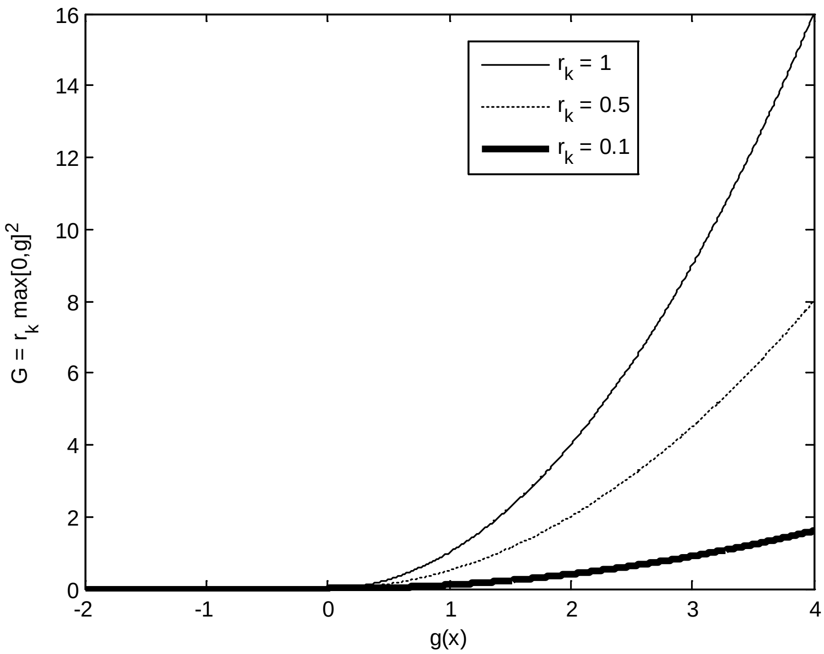
$$G_j = \{\max[0, g_i(\mathbf{X})]\}^2$$

در فرم کلی:

$$\phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} \langle g_j(\mathbf{X}) \rangle^q$$

$$\langle g_j(\mathbf{X}) \rangle = \max\langle g_j(\mathbf{X}), 0 \rangle$$

$$= \begin{cases} g_j(\mathbf{X}) & \text{if } g_j(\mathbf{X}) > 0 \\ & \quad \text{(constraint is violated)} \\ 0 & \text{if } g_j(\mathbf{X}) \le 0 \\ & \quad \text{(constraint is satisfied)} \end{cases}$$

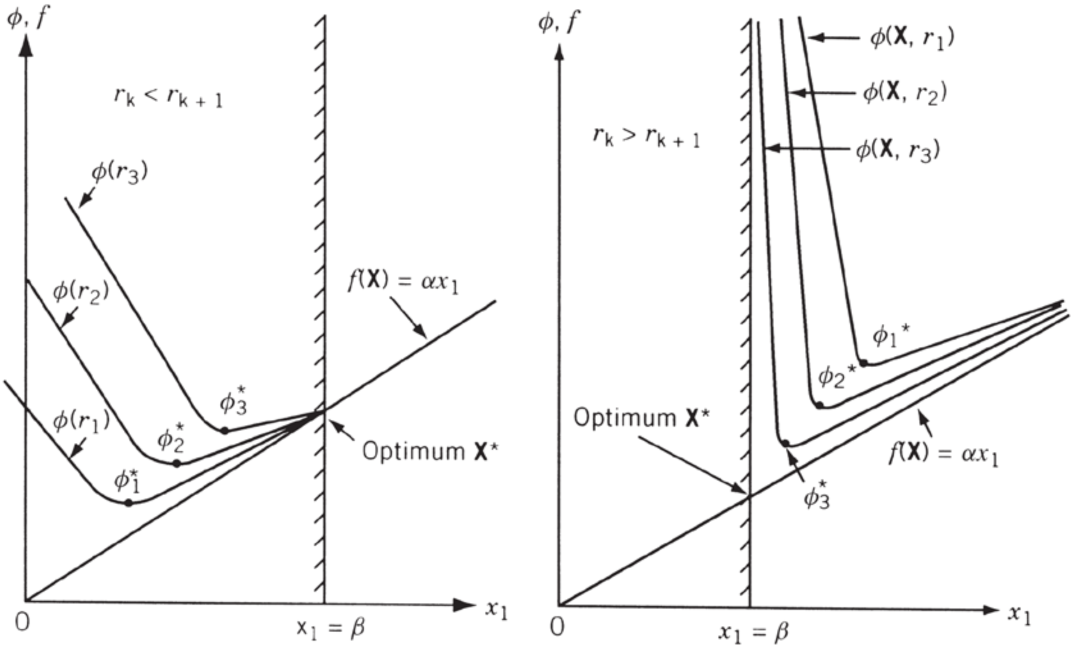


به عنوان مثال:

$$\text{Find } \mathbf{X} = \{x_1\} \text{ which minimizes } f(\mathbf{X}) = \alpha x_1$$

subject to

$$g_1(\mathbf{X}) = \beta - x_1 \le 0$$



$G_j\{g_j(\mathbf{X})\} = \{\max[0, g_j(\mathbf{X})]\}^2$ $G_j\{g_j(\mathbf{X})\} = -1/g_j(\mathbf{X})$

توابع جریمه تعمیم یافته Extended (Interior) Penalty Functions

$$\phi_k = \phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} \tilde{g}_j(\mathbf{X})$$

Linear Extended Penalty Function

$$\tilde{g}_j(\mathbf{X}) = \begin{cases} -\dfrac{1}{g_j(\mathbf{X})} & g_i(\mathbf{X}) \le \varepsilon \\ \dfrac{g_j(\mathbf{X}) - 2\varepsilon}{\varepsilon^2} & g_i(\mathbf{X}) > \varepsilon \end{cases}$$

Quadratic Extended Penalty Function

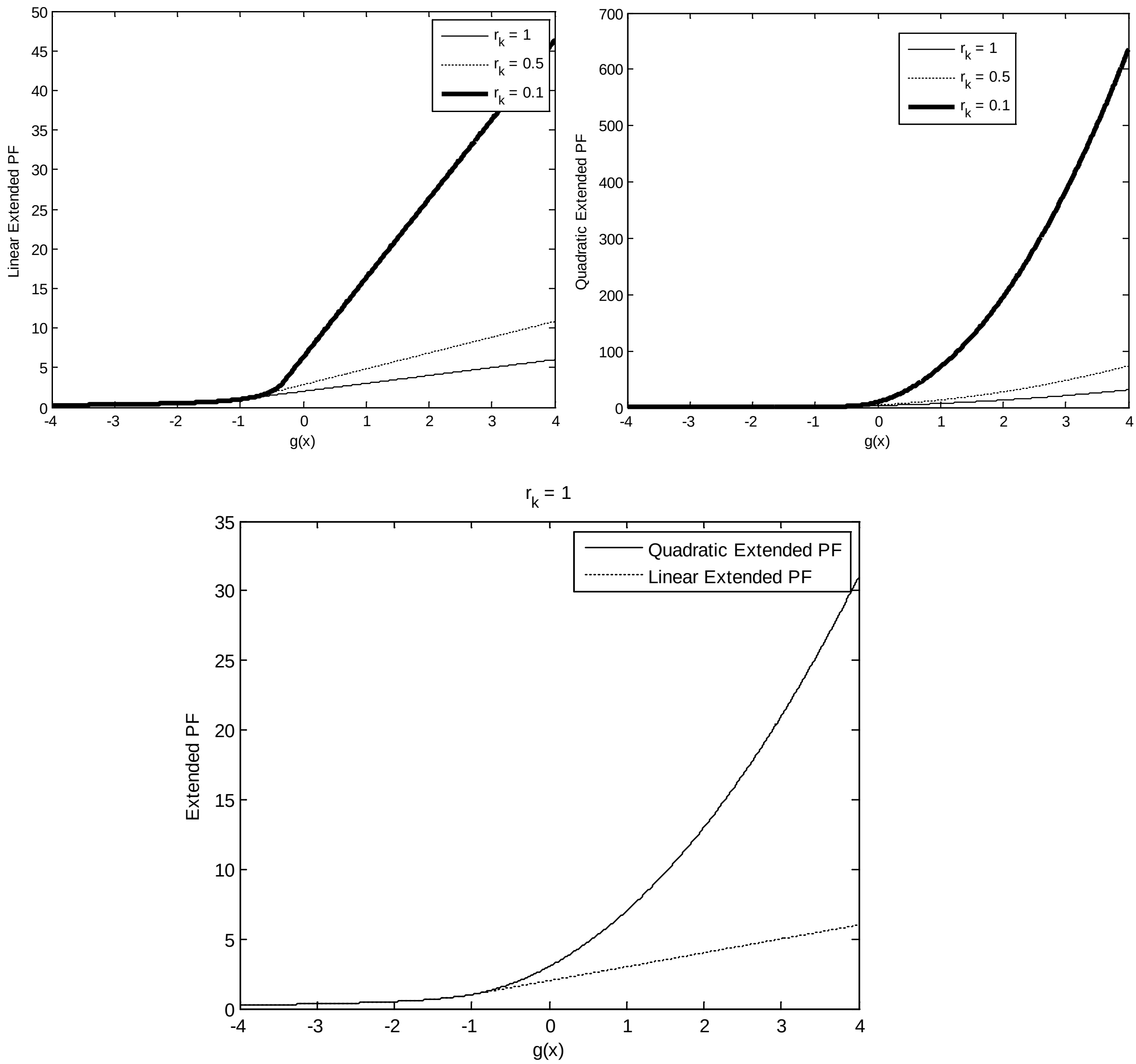
$$\tilde{g}_j(\mathbf{X}) = \begin{cases} -\dfrac{1}{g_i(\mathbf{X})} & g_i(\mathbf{X}) \le \varepsilon \\ -\dfrac{1}{\varepsilon}\left(3 - \dfrac{3g_i(\mathbf{X})}{\varepsilon} + \left(\dfrac{g_i(\mathbf{X})}{\varepsilon}\right)^2\right) & g_i(\mathbf{X}) > \varepsilon \end{cases}$$

که

$$\varepsilon = -c\left(r_k\right)^a \qquad \frac{1}{3} \le a \le \frac{1}{2}$$

$$r_{k+1} = c r_k$$

$$c < 1$$



تابع جریمه قیدهای مساوی Equality Constraint

$$\begin{aligned} &\min \quad f(\mathbf{X}) \\ \text{subject to} \quad & g_j(\mathbf{X}) \le 0 \qquad j = 1,\dots,m \\ & l_j(\mathbf{X}) = 0 \qquad j = 1,\dots,p \end{aligned}$$

$$\phi_k = \phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} G_j[g_j(\mathbf{X})] + H(r_k) \sum_{j=1}^{p} l_j^2(\mathbf{X})$$

$H(r_k)$ باید بزرگ شود

توابع جریمه داخلی و توسعه یافته Interior Penalty Functions

$$\phi_k = \phi(\mathbf{X}, r_k) = f(\mathbf{X}) - r_k \sum_{j=1}^{m} \frac{1}{g_j(\mathbf{X})} + \frac{1}{\sqrt{r_k}} \sum_{j=1}^{p} l_j^2(\mathbf{X})$$

توابع جریمه خارجی Exterior Penalty Functions

$$\phi_k = \phi(\mathbf{X}, r_k) = f(\mathbf{X}) + r_k \sum_{j=1}^{m} \langle g_j(\mathbf{X}) \rangle^2 + r_k \sum_{j=1}^{p} l_j^2(\mathbf{X})$$

جستجوی خطی Line Search

یادآوری: شکل کلی مسیر حل بهینه

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \lambda_i^* \mathbf{S}_i$$

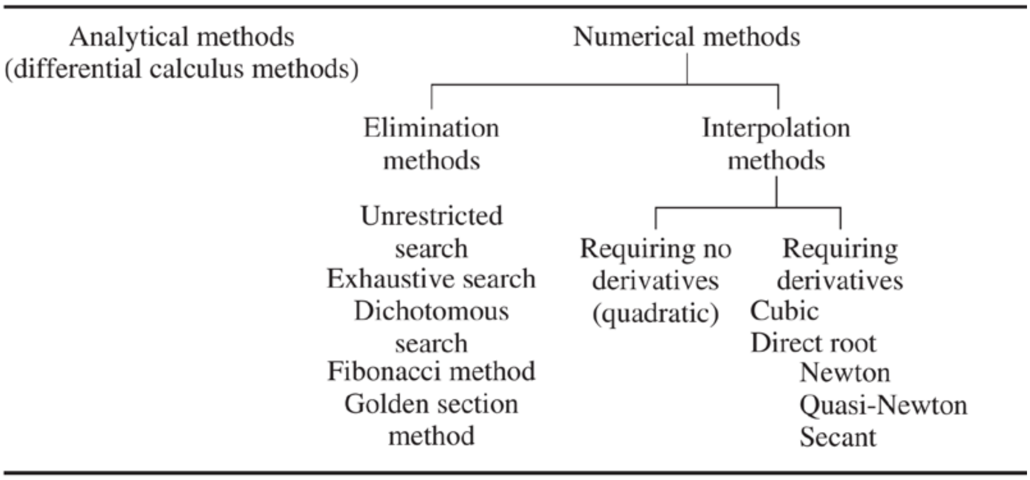
λ_i^* طول قدم بهینه در راستای $\mathbf{S}_i$

$$f_{i+1} = f(\mathbf{X}_i + \lambda_i^* \mathbf{S}_i) = \min_{\lambda_i} f(\mathbf{X}_i + \lambda_i \mathbf{S}_i)$$

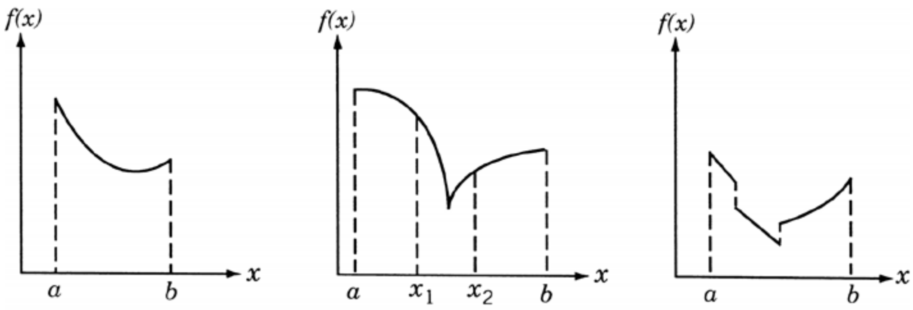
برای پیدا کردن طول بهینه (تحلیلی)

$$\left.\frac{df}{d\lambda}\right|_{\lambda=\lambda^*} = \nabla f|_{\lambda^*}^{\mathrm{T}} \mathbf{S}_i = 0$$

روش های عددی بهینه سازی تک پارامتری

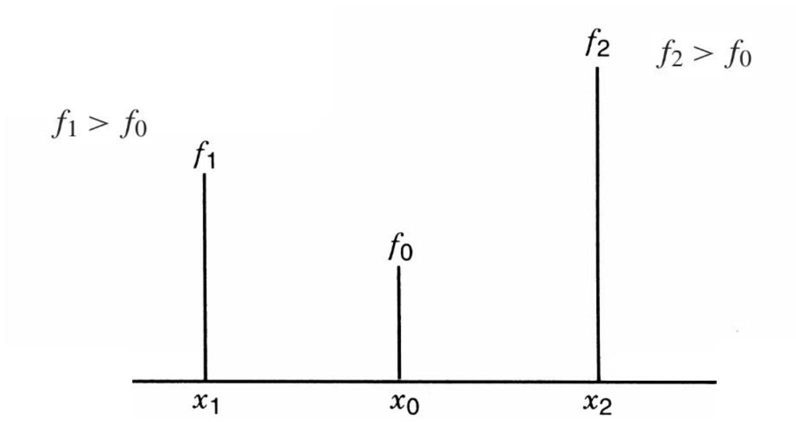


فرض همگی این است که تابع unimodal باشد



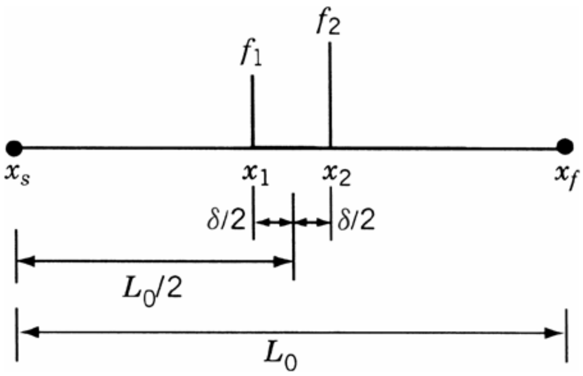
بحث اول پیدا کردن بازه مناسب (آخر سر می گویم)

شرط بازه این است که وسط نقطه بهینه باشد



روش های حذف بازه Elimination methods

روش Dichotomous



$$x_1 = \frac{L_0}{2} - \frac{\delta}{2}$$

$$x_2 = \frac{L_0}{2} + \frac{\delta}{2}$$

اگر $f_1 < f_2$ بازه سمت راست حذف می شود

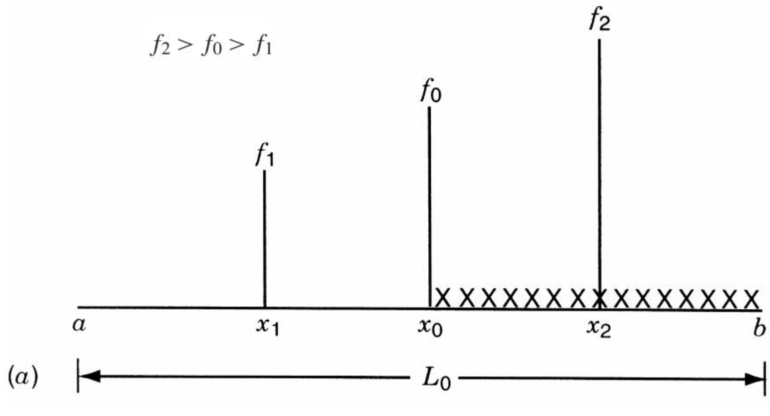
اگر $f_1 > f_2$ بازه سمت چپ حذف می شود

Number of experiments	2	4	6
Final interval of uncertainty	$\frac{1}{2}(L_0+\delta)$	$\frac{1}{2}\left(\frac{L_0+\delta}{2}\right)+\frac{\delta}{2}$	$\frac{1}{2}\left(\frac{L_0+\delta}{4}+\frac{\delta}{2}\right)+\frac{\delta}{2}$

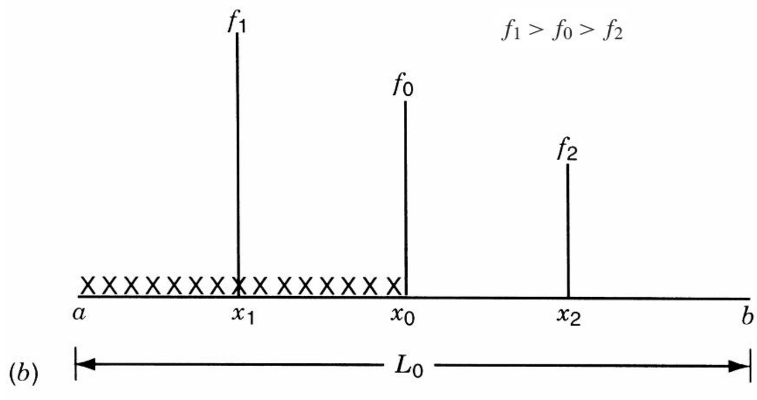
$$L_n = \frac{L_0}{2^{n/2}} + \delta\left(1 - \frac{1}{2^{n/2}}\right)$$

$$\frac{L_n}{L_0} = \frac{1}{2^{n/2}} + \frac{\delta}{L_0}\left(1 - \frac{1}{2^{n/2}}\right)$$

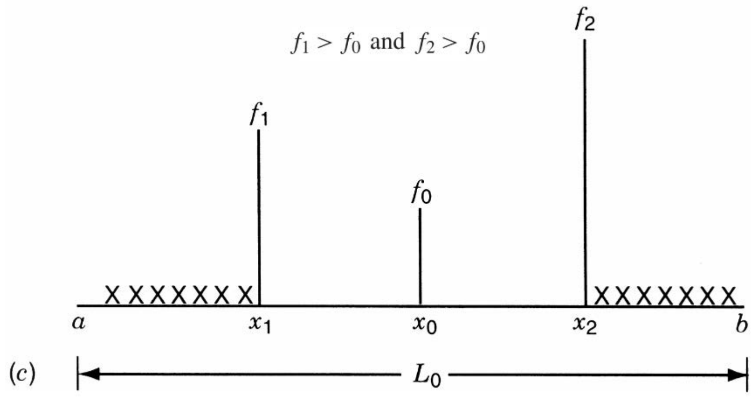
روش نصف کردن بازه Interval Halving



حذف بازه نیمه راست



حذف بازه نیمه چپ



حذف بازه ربع چپ و راست

دفعه اول یک بار تابع بیشتر حساب می شود (دفعه اول ۳ بار، دفعه های بعد ۲ بار)

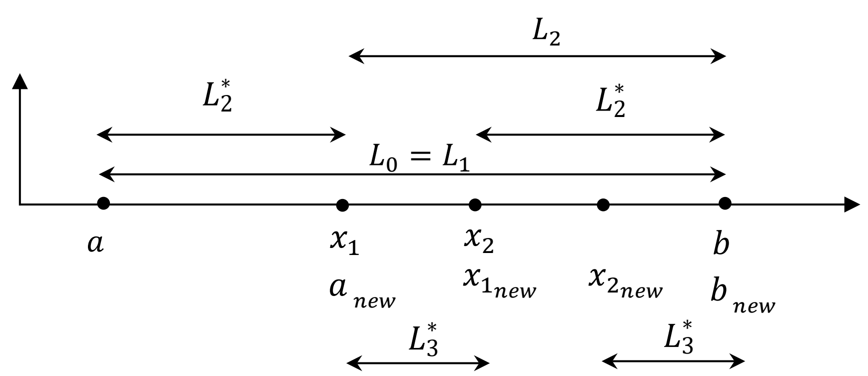
$$L_n = \left(\frac{1}{2}\right)^{(n-1)/2} L_0$$

روش فیبوناچی Fibonacci

دنباله فیبوناچی

$$F_0 = F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2}, \qquad n = 2, 3, 4, \ldots$$
$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

کاربرد در حذف بازه جواب؟



اگر $f_1 < f_2$ بازه سمت راست حذف می شود و اگر $f_1 > f_2$ بازه سمت چپ حذف می شود

بخشی از طول در بازه جدید که قرار است حذف شود

$$L_2^* = \frac{F_{n-2}}{F_n} L_0$$

$$L_2^* = \frac{F_{n-2}}{F_n} L_0 \le \frac{1}{2} L_0 \quad \text{for} \quad n \ge 2$$

برای پیدا کردن نقاط

$$x_1 = a + L_2^* = a + \frac{F_{n-2}}{F_n} L_0$$
$$x_2 = b - L_2^* = b - \frac{F_{n-2}}{F_n} L_0 = a + \frac{F_{n-1}}{F_n} L_0$$

طول جدید:

$$L_2 = L_0 - L_2^* = L_0 \left(1 - \frac{F_{n-2}}{F_n}\right) = \frac{F_{n-1}}{F_n} L_0$$

$$L_2^* = \frac{F_{n-2}}{F_n} L_0 = \frac{F_{n-2}}{F_{n-1}} L_2$$

فاصله یکی از نقاط باقیمانده نسبت به کناره نزدیکش:

$$L_2 - L_2^* = \frac{F_{n-3}}{F_n} L_0 = \frac{F_{n-3}}{F_{n-1}} L_2$$

بخشی از طول در بازه جدید که قرار است حذف شود:

$$L_3^* = \frac{F_{n-3}}{F_n} L_0 = \frac{F_{n-3}}{F_{n-1}} L_2$$

پس عملا یک نقطه قبلا تابع f آن محاسبه شده

طول جدید:

$$L_3 = L_2 - L_3^* = L_2 - \frac{F_{n-3}}{F_{n-1}} L_2 = \frac{F_{n-2}}{F_{n-1}} L_2 = \frac{F_{n-2}}{F_n} L_0$$

به همین ترتیب:

$$L_j^* = \frac{F_{n-j}}{F_{n-(j-2)}} L_{j-1} \qquad L_j = \frac{F_{n-(j-1)}}{F_n} L_0$$

$$\frac{L_j}{L_0} = \frac{F_{n-(j-1)}}{F_n} \qquad \frac{L_n}{L_0} = \frac{F_1}{F_n} = \frac{1}{F_n}$$

می توان بر مبنای طول و دقت موردنیاز، تعداد اعداد فیبوناچی را انتخاب کرد.

دفعه اول یک بار تابع بیشتر حساب می شود (دفعه اول ۲ بار، دفعه های بعد ۱ بار)

Value of n	Fibonacci number, F_n	Reduction ratio, L_n/L_0
0	1	1.0
1	1	1.0
2	2	0.5
3	3	0.3333
4	5	0.2
5	8	0.1250
6	13	0.07692
7	21	0.04762
8	34	0.02941
9	55	0.01818
10	89	0.01124
11	144	0.006944
12	233	0.004292
13	377	0.002653
14	610	0.001639
15	987	0.001013
16	1,597	0.0006406
17	2,584	0.0003870
18	4,181	0.0002392
19	6,765	0.0001479
20	10,946	0.00009135

روش نسبت طلایی Golden Section

حالت حدی فیبوناچی

Value of N	2	3	4	5	6	7	8	9	10	∞
Ratio $\frac{F_{N-1}}{F_N}$	0.5	0.667	0.6	0.625	0.6156	0.619	0.6177	0.6181	0.6184	0.618

تغییرات طول فیبوناچی:

$$L_2 = \lim_{N\to\infty} \frac{F_{N-1}}{F_N} L_0$$

$$L_3 = \lim_{N\to\infty} \frac{F_{N-2}}{F_N} L_0 = \lim_{N\to\infty} \frac{F_{N-2}}{F_{N-1}} \frac{F_{N-1}}{F_N} L_0$$

$$\simeq \lim_{N\to\infty} \left(\frac{F_{N-1}}{F_N}\right)^2 L_0$$

$$L_k = \lim_{N\to\infty} \left(\frac{F_{N-1}}{F_N}\right)^{k-1} L_0$$

$$F_N = F_{N-1} + F_{N-2} \qquad\qquad \frac{F_N}{F_{N-1}} = 1 + \frac{F_{N-2}}{F_{N-1}}$$

با تعریف (عکس نسبت طول ها):

$$\gamma = \lim_{N\to\infty} \frac{F_N}{F_{N-1}}$$

معادله عکس نسبت طول:

$$\gamma \simeq \frac{1}{\gamma} + 1 \qquad\qquad \gamma^2 - \gamma - 1 = 0$$

$$\gamma = \frac{1 \pm \sqrt{5}}{2} = \begin{cases} 1.618 \\ -0.618 \end{cases} \quad \Rightarrow \quad \gamma = 1.618$$

$$\frac{1}{\gamma} = 0.618$$

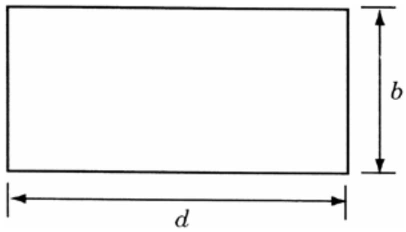
تغییرات طول

$$L_k = \left(\frac{1}{\gamma}\right)^{k-1} L_0 = (0.618)^{k-1} L_0$$

بخشی که باید از کناره ها حرکت کرد:

$$L_2^* = \frac{F_{N-2}}{F_N} L_0 = \frac{F_{N-2}}{F_{N-1}} \frac{F_{N-1}}{F_N} L_0 = \frac{L_0}{\gamma^2} = 0.382 L_0$$

نسبت طلایی



$$\frac{d+b}{d} = \frac{d}{b} = \gamma$$

پیدا کردن بازه (Bracketing)

با همین نسبت طلایی:

$$\gamma = 1.618$$

شروع

$$a = 0, b = b_0 = \varepsilon$$

شرط اینکه جهت جستجو خوب است

$$f(b) < f(a)$$

وگرنه یا ε بزرگ است یا جهت جستجو کاهشی نیست.

یک نقطه بعدی

$$c = b + \gamma(a - b)$$

شرط توقف:

$$f(b) < f(c)$$

درغیراینصورت:

$$\begin{aligned} a &= b_{prev} \qquad & f(a) &= f(b_{prev}) \\ b &= c_{prev} \qquad & f(b) &= f(c_{prev}) \\ c &= b + \gamma(a - b) \end{aligned}$$

مزیت: وقتی توقف کند، یکی از نقاط میانی پیدا شده است.

روش های میانیابی Interpolation methods

میانیابی درجه ۲ (Quadratic interpolation)

منحنی درجه ۲ به نقاط برازش می شود

به سه نقطه نیاز دارد (روش bracketing قبلی سه نقطه می دهد که شرط unimodal را هم دارد)

$$h(\lambda) = a + b\lambda + c\lambda^2$$

$$\frac{dh}{d\lambda} = b + 2c\lambda = 0$$

جواب

$$\tilde{\lambda}^* = -\frac{b}{2c}$$

شرط مینیمم بودن:

$$\left.\frac{d^2h}{d\lambda^2}\right|_{\tilde{\lambda}^*} > 0$$

یا:

$$c > 0$$

برای سه نقطه:

$$\lambda = A, \lambda = B, \text{ and } \lambda = C$$

$$f_A = a + bA + cA^2$$

$$f_B = a + bB + cB^2$$

$$f_C = a + bC + cC^2$$

جواب برای ضرایب منحنی درجه ۲:

$$a = \frac{f_A BC(C-B) + f_B CA(A-C) + f_C AB(B-A)}{(A-B)(B-C)(C-A)}$$

$$b = \frac{f_A(B^2-C^2) + f_B(C^2-A^2) + f_C(A^2-B^2)}{(A-B)(B-C)(C-A)}$$

$$c = -\frac{f_A(B-C) + f_B(C-A) + f_C(A-B)}{(A-B)(B-C)(C-A)}$$

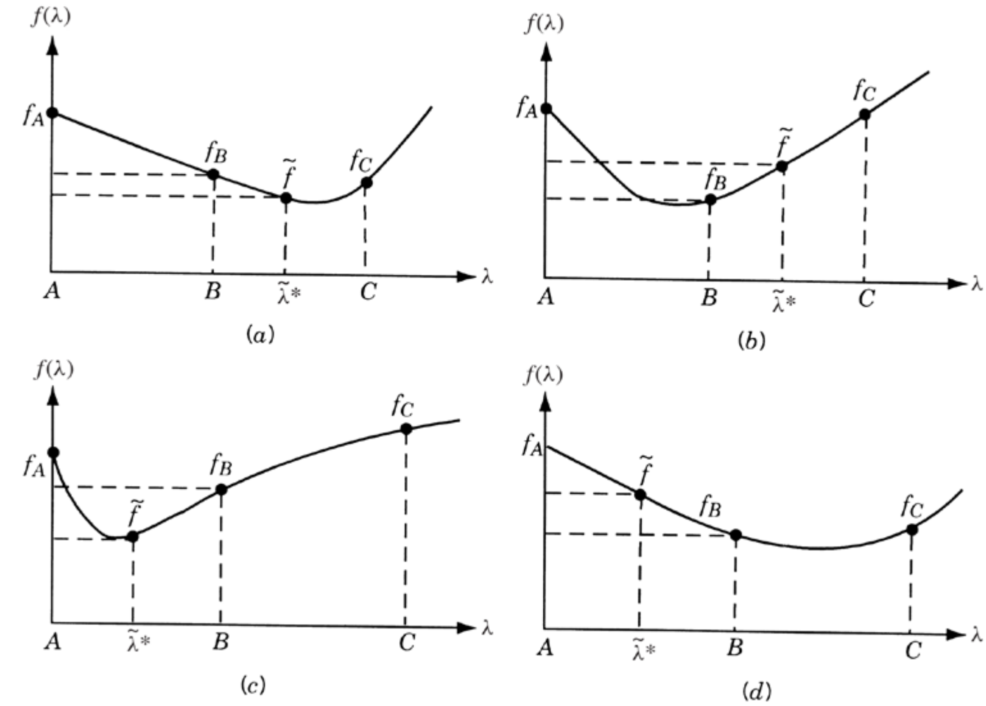
و جواب:

$$\tilde{\lambda}^* = \frac{-b}{2c} = \frac{f_A(B^2 - C^2) + f_B(C^2 - A^2) + f_C(A^2 - B^2)}{2[f_A(B - C) + f_B(C - A) + f_C(A - B)]}$$

شرط توقف:

$$\left| \frac{h(\tilde{\lambda}^*) - f(\tilde{\lambda}^*)}{f(\tilde{\lambda}^*)} \right| \le \varepsilon_1$$

اگر خوب برازش نشده بود، باید ۳ نقطه جدید انتخاب کرد (بین ۴ تا) و دوباره منحنی برازش کرد

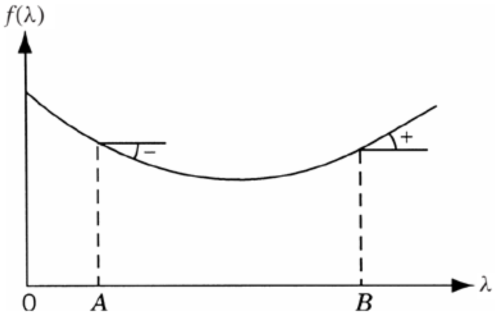


Case	Characteristics	New points for refitting: New	Old
1	$\tilde{\lambda}^* > B$	A	B
	$\tilde{f} < f_B$	B	$\tilde{\lambda}^*$
		C	C
		Neglect old A	
2	$\tilde{\lambda}^* > B$	A	A
	$\tilde{f} > f_B$	B	B
		C	$\tilde{\lambda}^*$
		Neglect old C	
3	$\tilde{\lambda}^* < B$	A	A
	$\tilde{f} < f_B$	B	$\tilde{\lambda}^*$
		C	B
		Neglect old C	
4	$\tilde{\lambda}^* < B$	A	$\tilde{\lambda}^*$
	$\tilde{f} > f_B$	B	B
		C	C
		Neglect old A	

میانیابی درجه ۳ (Cubic interpolation)

منحنی درجه ۳:

$$h(\lambda) = a + b\lambda + c\lambda^2 + \lambda^3$$



$$f_A = a + bA + cA^2 + dA^3$$

$$f_B = a + bB + cB^2 + dB^3$$

$$f'_A = b + 2cA + 3dA^2$$

$$f'_B = b + 2cB + 3dB^2$$

نیاز به شیب تابع دارد که در مساله کنترل بهینه موجود نیست.

روش های حل مستقیم ریشه Direct root method

حل این مساله:

$$f'(\lambda^*) = 0$$

حداقل به محاسبه شیب تابع نیاز دارد.

مثلا روش نیوتن:

$$\lambda_{i+1} = \lambda_i - \frac{f'(\lambda_i)}{f''(\lambda_i)}$$

بعضا به مشتق دوم هم نیاز دارد که موجود نیست.