5.5.1 Globalización por búsqueda lineal

En los métodos con búsqueda lineal los iterantes se generan, en el caso más simple, por la recurrencia siguiente:

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

donde d_k es una dirección de $\mathbb{R}^n$ y α_k es un paso determinado de manera que decrezca una función de mérito.

Vamos a considerar un problema cuadrático mas general que $(\mathcal{PQT})$ de manera que incluya las versiones cuasi-newtonianas de la programación secuencial cuadrática: el hessiano del lagrangiano se reemplaza por una matriz B_k. El problema cuadrático en d se transforma en:

$$\begin{cases} \min \nabla f(x_k)^t d + \frac{1}{2} d^t B_k d \\ c_I(x_k) + A_I(x_k) d = 0 \\ c_D(x_k) + A_D(x_k) d \leq 0, \end{cases} \tag{8}$$

Se supondrá que d_k verifica las condiciones de optimalidad de de primer orden que se escriben para un multiplicador λ_k^{PQ}:

$$B_k d_k + A(x_k)^t \lambda_k^{PQ} = -\nabla f(x_k)$$

$$c_I(x_k) + A_I(x_k) d_k = 0$$

$$c_D(x_k) + A_D(x_k) d_k \leq 0$$

$$(\lambda_k^{PQ})_D \geq 0$$

$$(\lambda_k^{PQ})_D^t (c_D(x_k) + A_D(x_k)) = 0$$

La determinación del paso α_k es compleja y es todavía objeto de investigación. El principio es el siguiente: Se busca α_k de manera que x_{k+1} haga decrecer una función de mérito del problema. Por ejemplo se puede tomar:

$$\theta_\sigma = f(x) + \sigma \| c(x)^{\#} \|$$

y buscar que se cumpla al menos:

$$\theta_\sigma(x_{k+1}) < \theta_\sigma(x_k)$$

Como se verá es necesario ser más exigente que eso. Por otra parte habría que precisar la evolución de las variables λ_k. Ciertos autores hacen una búsqueda lineal en λ utilizando una función de mérito primal-dual. Otros autores proponen simplemente:

$$\lambda_{k+1} = \lambda_k + \alpha_k(\lambda_k^{PQ} - \lambda_k)$$

Para que θ_σ pueda ser utilizada como función de mérito midiendo el progreso alcanzado desplazándose a lo largo de d_k, es necesario que d_k sea una dirección de descenso de θ_σ en x_k:

$$\theta'_\sigma(x_k, d_k) < 0$$

Si B_k es definida positiva y se ajusta σ_k de manera que en cada iteración:

$$\sigma_k \geq \|\lambda_k^{PQ}\|$$

entonces d_k, solución estacionaria del problema cuadrático, es una dirección de descenso de θ_{σ_k} en x_k. Como asintóticamente se debe tener $\sigma > \|\lambda^*\|$ para que θ_σ sea una función de penalización exacta, los algoritmos mantienen la desigualdad anterior en cada iteración mediante reglas adecuadas. Para una constante $\bar{\sigma} > 0$ se adaptará el factor de penalización de manera que:

- $\sigma_k \geq \|\lambda_{PQ}^k\| + \bar{\sigma}$
- existe un índice k_1 tal que $\forall k \geq k_1$

$$\sigma_{k-1} \;\; \geq \|\lambda_{PQ}^k\| + \bar{\sigma} \Rightarrow \sigma_k = \sigma_{k-1}$$

- $\{\sigma_k\}$ acotada $\Rightarrow$ σ_k se modifica un número finito de veces

Vamos a especificar ahora una desigualdad forzando el decrecimiento de θ_{σ_k}. Elegiremos la *condición de Armijo* siguiente: encontrar $\alpha > 0$ tal que para $\omega \in (0, 0.5)$ se cumpla:

$$x_k + \alpha d_k \in \Omega \quad \text{y} \quad \theta_{\sigma_k}(x_k + \alpha_k d_k) \leq \theta_{\sigma_k}(x_k) + \omega\alpha\Delta_k$$

La exigencia $\omega < \dfrac{1}{2}$ se deduce de razones teóricas. El valor de Δ_k debería ser idealmente $\theta'_\sigma(x_k, d_k)$ pero como esta derivada direccional no se puede calcular fácilmente se elige el mayorante:

$$\Delta_k := \nabla f(x_k)^t d_k - \sigma\|c(x_k)^\#\| = -d_k^t B_k d_k - (\lambda_k^{PQ})^t - \sigma\|c(x_k)^\#\|$$

que cumple $\theta'_\sigma(x_k, d_k) \leq \Delta_k < 0$.

5.5.2 Programación secuencial cuadrática cuasi-newtoniana

Recordemos que, localmente, la programación cuadrática sucesiva calcula un desplazamiento d_k en x resolviendo el problema cuadrático en d siguiente:

$$\begin{cases} \min \nabla f(x_k)^t d + \frac{1}{2} d^t \nabla^2_{xx} \ell(x_k, \lambda_k) d \\ c_I(x_k) + A_I(x_k) d = 0 \\ c_D(x_k) + A_D(x_k) d \leq 0 \end{cases} \tag{9}$$

En la versión cuasi-newtoniana del algoritmo, B_k es una matriz simétrica definida positiva, puesta al díapor la fórmula BFGS utilizando dos vectores γ_k y δ_k a determinar en $\mathbb{R}^n$. Parece razonable que B_k aproxime el hessiano del lagrangiano; por tanto, se puede elegir $\gamma_k = \gamma_k^\ell$ (variación del gradiente del lagrangiano cuando x varía), y tomar, para δ_k el desplazamiento en x:

$$\gamma_k^\ell = \nabla \ell(x_{k+1}, \lambda_k^{PQ}) - \nabla \ell(x_k, \lambda_k^{PQ})$$

En γ_k^ℓ se ha fijado el multiplicador al valor λ_k^{PQ}, que es más próximo a λ^* que λ_k.

Sería necesario poder realizar la condición $(\gamma_k^\ell)^t \delta_k > 0$ (que asegura la conservación del carácter definido positivo de B_k), eligiendo bien x_{k+1}.

Es natural elegir x_{k+1} a lo largo de d_k haciendo una búsqueda lineal para hacer decrecer una función de mérito. Sin embargo, es posible que el lagrangiano tenga una curvatura negativa sobre $\{x_k + \alpha d_k \ / \ \alpha \in \mathbb{R}\}$, es decir, puede ocurrir que $(\gamma_k^\ell)^t \delta_k \leq 0$ con $x_{k+1} = x_k + \alpha_k d_k$ sea cual sea $\alpha_k > 0$.

Para resolver está dificultad Powell propone calcular γ_k modificando γ_k^ℓ cuando el producto $(\gamma_k^\ell)^t \delta_k$ no sea suficientemente positivo. En primer lugar se determina un paso α_k a lo largo de d_k de manera que se haga decrecer una función de mérito (por ejemplo la función de penalización de Han) lo que da el iterante siguiente $x_{k+1} = x_k + \alpha_k d_k$. Se definen después γ_k^ℓ y δ_k como antes y se elige γ_k como una combinación convexa de γ_k^ℓ y $B_k \delta_k$:

$$\gamma_k = \theta \gamma_k^\ell + (1 - \theta) B_k \delta_k$$

donde:

$$\theta = \begin{cases} 1 & \text{si } (\gamma_k^\ell)^t \delta_k \geq 0.2 \delta_k^t B_k \delta_k \\ 0.8 \dfrac{\delta_k^t B_k \delta_k}{\delta_k^t B_k \delta_k - (\gamma_k^\ell)^t \delta_k} & \text{en otro caso} \end{cases}$$

Para modificar γ_k^ℓ lo menos posible (y retener por tanto la mayor información del problema) se elige el parámetro θ tan grande como se pueda en $[0, 1]$ asegurandose de que se cumpla:

$$(\gamma_k)^t \delta_k \geq 0.2 \delta_k^t B_k \delta_k$$

Por último se actualiza B_k con la fórmula BFGS utilizando γ_k. Esta técnica se conoce con el nombre de corrección de Powell.

Se puede dar un algoritmo que resuma todo esto:

DESCRIPCIÓN DE UNA ITERACIÓN DEL ALGORITMO

Al comienzo de la iteración se conoce el iterante (x_k, λ_k) y una matriz B_k, simétrica definida positiva, aproximación del hessiano del lagrangiano. La finalidad de la iteración es dar un nuevo valor a estas tres variables.

1. Calcular la única solución primal-dual (d_k, λ_k^{PQ}) del problema cuadrático tangente:

$$\begin{cases} \min \nabla f(x^k)d + \frac{1}{2}d^t B_k d \\ c_E(x^k) + A_E(x^k)d = 0 \\ c_I(x^k) + A_I(x^k)d \leq 0, \end{cases} \tag{10}$$

2. Preparación de la búsqueda lineal: adaptar el factor de penalización σ_k de manera que se tenga:

$$\sigma_k \geq \|\lambda_{PQ}^k\| + \bar{\sigma}, \tag{11}$$

donde $\bar{\sigma} > 0$ es una constante "pequeña". Las reglas de actualización serán:

<u>Primera iteración</u>:

Se elige

$$\bar{\sigma} = \max(\sqrt{\texttt{eps}}, \|\lambda_1\|/100) \quad \text{y} \quad \sigma_1 = \|\lambda_1\| + \bar{\sigma},$$

para `eps` igual al valor del ϵ del ordenador.

<u>Iteraciones siguientes:</u>

Podemos aumentar el valor de σ para satisfacer (11), pero puede o- currir que sea muy "grande" para hacer la búsqueda lineal. Entonces hacemos:

si $\sigma_{k-1} < \|\lambda_k^{PQ}\| + \bar{\sigma}$,
$\quad \sigma_k = \max(1.5\, \sigma_{k-1}, \|\lambda_k^{PQ}\| + \bar{\sigma})$;
si no, si $\sigma_{k-1} > 1.1\,(\|\lambda_k^{PQ}\| + \bar{\sigma})$,
$\quad \sigma_k = (\sigma_{k-1} + \|\lambda_{PQ}^k\| + \bar{\sigma})/2$;
si no
$\quad \sigma_k = \sigma_{k-1}$.

3. Búsqueda lineal: encontrar un paso α_k de manera que se verifique la condición de Armijo sobre la función de penalización de Han $\Theta_\sigma(x) = f(x) + \sigma\|c(x)^\#\|$:

$$\Theta_\sigma(x + \alpha d_k) \leq \Theta_\sigma(x_k) + \omega\alpha\Delta_k$$

donde $\omega \in (0, 0.5)$ es dado y

$$\Delta_k := -(d_k)^t B_k d_k + (\lambda_k^{PQ})^t c(x_k) - \sigma_k\|c(x_k)^\#\|$$

4. Hacer $x_{k+1} = x_k + \alpha_k d_k$ y $\lambda_{k+1} = \lambda_k + \alpha_k(\lambda_k^{PQ} - \lambda_k)$.

5. Calcular:

$$\begin{aligned} \delta_k &= x_{k+1} - x_k \\ \gamma_k^\ell &= \nabla\ell(x^{k+1}, \lambda_k^{PQ}) - \nabla\ell(x_k, \lambda_k^{PQ}) \\ \gamma_k &= \theta\gamma_k^\ell + (1-\theta)B_k\delta_k \end{aligned}$$

donde

$$\theta = \begin{cases} 1 & \text{si } (\gamma_k^\ell)^t\delta_k \geq 0.2\delta_k^t B_k\delta_k \\ 0.8\,\dfrac{\delta_k^t B_k\delta_k}{\delta_k^t B_k\delta_k - (\gamma_k^\ell)^t\delta_k} & \text{en otro caso} \end{cases}$$

Calcular B_{k+1} por la fórmula de BFGS:

$$B_{k+1} = B_k - \frac{B_k\delta_k\delta_k^t B_k}{\delta_k^t B_k\delta_k} + \frac{\gamma_k\gamma_k^t}{\gamma_k^t\delta_k}$$

Bibliografía básica:

- D. BERTSEKAS : Nonlinear Programming. Athena Scientific. 1995.
- F. BONNANS - J. Ch. GILBERT - C. LEMARECHAL - C. SAGASTIZABAL : Optimisation Numérique: aspects théoriques et pratiques. SMAI-Springer Verlag, 1997.
- R. FLETCHER : Practical Methods of Optimization. John Wiley and Sons, 1987.
- P.E. GILL - W. MURRAY - M. WRIGHT : Practical Optimization. Academic Press, 1981.
- P.E. GILL - W. MURRAY - M. WRIGHT : Numerical linear Algebra and Optimization. Addison-Wesley, 1991.
- D. GOLDFARB - M. J. TODD : Linear Programming. En Handbook in OR & MS, Vol. 1. (G.L. Nemhauser et al., ed.). North-Holland, 1989.
- D.G. LUENBERGER : Programacin lineal y no lineal. Addison-Wesley Iberoamericana, 1989.
- M. MINOUX : Programmation Mathématique. Dunod, 1983.
- J. NOCEDAL - S.J. WRIGHT : Numerical Optimization. Springer, 1999.
- C. POLA : Algoritmos numéricos para la resolución de problemas de optimización con restricciones. Tesis de Doctorado, Universidad de Cantabria, 1992.