Métodos de región de confianza adaptativos para problemas de optimización sin restricciones

Emil Mubarqui
emil.mubarqui@cimat.mx
Cipriano C. Hdz.
cipriano.callejas@cimat.mx

***Resumen*—Se estudian dos modificaciones del método de región de confianza convencional, donde a diferencia de este, se permite extender la región de confianza cuando la iteración es muy buena (en algún sentido, posteriormente definido). Comparamos el desempeño de los algoritmos con el método tradicional, observamos que obtenemos una mejora para problemas de gran escala tanto en tiempo como en número de iteraciones.**

Métodos de Región de Cofianza

Deseamos resolver el problema de optimización sin restricciones

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1}$$

donde $f : \mathbb{R}^n \to \mathbb{R}$ es una función suave. Existen al menos dos métodos para generar una sucesión $(x_k)_{k\geq 0}$ que aproxime un punto de equilibrio (o crítico) de f, el de búsqueda en línea y el de región de confianza. En cualquiera de estos, se busca una nueva iteración x_{k+1} tal que $f(x_{k+1}) < f(x_k)$. Ver Figura 1.

Los métodos de región de confianza buscan en una vecindad de x_k, para obtener la nueva iteración x_{k+1}. De hecho, estos métodos se consideran confiables y robustos porque pueden ser usados en problemas donde f no es convexa o incluso mal condicionados.

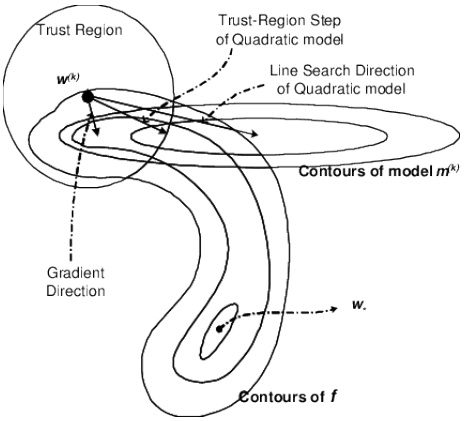

Figura 1. Imagen tomada de [1]

En cada iteración, estos métodos hacen uso de un modelo cuadrático m_k,

$$m_k(x_k + p) = f(x_k) + g_k^T p + \frac{1}{2} p^T B_k p.$$

donde B_k es una matriz simétrica, y usualmente representa una aproximación de la matriz Hessiana. Por tanto, la nueva dirección de búsqueda está dada por

$$\begin{cases} p_k^* = \arg\min_p m_k(p), \\ \text{s.a. } \|p\| \leq \Delta_k. \end{cases} \tag{2}$$

Cuando B_k coincide con ser la matriz hessiana de f, entonces $p_k^* = -B_k^{-1} g_k$. Los métodos de región de

confianza se pueden describir de manera general como sigue, ver [2].

1. **Inicialización**: Un punto inicial x_0, y un radio inicial Δ_0. Así como los parámetros, $0 \leq \eta_1 \leq \eta_2 \leq 1$ y $0 < \gamma_1 < \gamma_2 \leq 1 \leq \gamma_3$ y $\epsilon > 0$.
2. **Modelo cuadrático** Definir el modelo cuadrático m_k y resolver (2) para determinar la dirección p_k.
3. **Aceptación del punto.** Usamos la medida de paso definida por

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)}. \tag{3}$$

   Si $\rho_k \geq \eta_1$, definimos $x_{k+1} = x_k + p_k$, en otro caso, $x_{k+1} = x_k$.
4. **Actualizamos la región de confianza**

$$\begin{cases} \Delta_{k+1} = \gamma_3 \Delta_k & \text{si } \rho_k \geq \eta_2, \\ \Delta_{k+1} = \gamma_2 \Delta_k & \text{si } \eta_1 \leq \rho_k < \eta_2, \\ \Delta_{k+1} = \gamma_1 \Delta_k & \text{si } \rho_k < \eta_1. \end{cases}$$

En este tipo de algoritmos Δ_k juega un papel importante, pues recordemos que *confiamos* en que el óptimo estará dentro de la región de confianza $\|p\| \leq \Delta_k$, si la región es muy grande el mínimo de (2) puede estar muy alejado del mínimo de f, y en caso contrario, puede no estar el óptimo en la región [1]. Por tanto, estos algoritmos dependen fuertemente en cómo se determina el radio de la región de confianza, Δ_k.

Notemos que en (3), el denominador siempre es positivo, entonces el valor de ρ_k determina la relación entre $f(x_k)$ y $f(x_k + p_k)$, por ejemplo, si ρ_k es negativo, entonces no hubo una disminución en el valor de la función. De aquí que esto determina qué tan bueno es el nuevo punto.

Otro punto importante en este tipo de algoritmos es la estrategia para resolver (2), para ello existen distintos métodos, entre los cuales están el punto de Cauchy **Punto de Cauchy** y **Método de Dogleg** (Algoritmo 1), para más detalles consultar [1]. No obstante, este subproblema crea una relación entre el costo computacional y el tiempo de convergencia, por ejemplo, si se usa el paso de Cauchy, se requiere del uso del Hessiano con lo que aumenta el costo computacional, pero la convergencia es casi lineal [3].Por otro lado, en [4], se propone usar el método de Gradiente Conjugado truncado de Steihaug-Toint, ver [2, pág. 205].

Estado del arte

Se han hecho otras modificaciones a los algoritmos de región de confianza. Nocedal et al. [5] plantearon las bases para combinar algoritmos de búsqueda en línea con los de región de confianza, y Gertz [6] lo modificó usando una variante de las condiciones fuertes de Wolfe. En estos, se calcula la dirección p_k del nuevo paso a través de métodos de búsqueda en línea. También se ha trabajado en la actualización del radio de la región de confianza, por ejemplo, Bastin [7] define una nueva medida de aceptación *retrospectiva*:

$$\tilde{\rho}_{k+1} = \frac{f(x_{k+1}) - f(x_{k+1} - p_k)}{m_{k+1}(x_{k+1}) - m_{k+1}(x_{k+1} - p_k)}.$$

La actualización del radio se determina a partir de esta nueva medida, de modo que esto se hace con la información de la iteración corriente, en vez de la anterior. Por otro lado, se han propuesto métodos que no decrecen monótonamente que presentan mayor eficiencia que los métodos tradicionales de región de confianza. Deng et al. [8] propusieron un método no monótono al relajar la condición de aceptación del nuevo paso p_k. Zhang y Zhang [9] plantean un algoritmo de región de confianza adaptativo, en el que disminuyen el radio a través de un parámetro entero y una constante entre cero y uno para iteraciones donde la medida calculada sea menor a un valor dado. Por último, Wenyu [10] propone otro método no monótono al introducir la medida:

$$\bar{r}_k = \frac{f(x_{l(k)}) - f(x_k + p_k)}{m_k(x_k) - m_k(x_k + p_k)},$$

donde

$$f(x_{l(k)}) = \max_{0 \le j \le m(k)} f(x_{k-j}),$$

con $0 = m(0) \le m(k) \le \min\{m(k-1)+1, M\}$. Es decir, se compara el nuevo valor $f(x_k + p_k)$ con el peor valor de las iteraciones anteriores[1] $f(x_{l(k)})$.

Algorithm 1: Dogleg

Given $x_k, \tau', \Delta_k, g_k, B_k$;
$p_k^U = \frac{\|g_k\|^2}{g_k^T B_k g_k} g_k$;
$p_k^U = -B_k^{-1} g_k$;
if $\left\|p_k^B\right\| \le \Delta_k$ **then**
  $p_k = p_k^B$
end
else
  Find (λ);
  $\tau = \lambda + 1$;
  if $\tau \in [0, 1]$ **then**
    $p_k = \tau p_k^U$
  end
  else
    if $\tau \in (1, 2]$ **then**
      $pk = \tau p_k^B + (\tau' - 1) p_k^U - p_k^B$.
    end
  end
end
return p_k

Algoritmos adaptativos de región de confianza

En esta sección se describen dos algoritmos adaptativos de region de confianza donde se busca una mejoría en el paso s_k cuando el modelo se ajusta muy bien a la función f y se encuentra en un punto frontera, $\|s_k\| = \Delta_k$, las iteraciones que cumplan con esta condición serán denominadas como *muy buenas*, y si una iteración antes no se cumplió, se denominarán como *muy exitosas*. El Algoritmo 1 extiende el radio de la región sin actualizar el modelo de la función[2], mientras que el Algoritmo 2 realiza una búsqueda en línea en la dirección de s_k fuera de la región de confianza, de modo que logre disminuir el valor de la función. Ver [4, Algoritmo 1, Algoritmo 2].

Para el análisis de convegencia suponemos ciertas las siguientes condiciones.

1. La función objetivo $f(x)$ está acotada por abajo.
2. Los Hessianos de la función objetivo y del modelo están uniformemente acotados.
3. El valor de la función objetivo y del modelo son iguales en la iteración corriente.
4. El gradiente del modelo es igual al de la función objetivo en la iteración corriente.

Algoritmo 1

En este algoritmo, se busca la extensión del radio de la región de confianza cuando tenemos una muy buena iteración.

El algoritmo puede describirse de la siguiente forma: Para alguna $k \ge 0$, tenemos que el paso $s_{k_0} = s_k$ se calcula minimizando el modelo m_k bajo la condición de que la solución esté dentro de la región de confianza. Entonces, dadas las constantes $0 < \eta_1 < \eta_2 < 1$, el punto $x_k + s_{k_0}$ se acepta si $\eta_1 \le \rho_{k_0} < \eta_2$, donde ρ_{k_j} es la medida de ajuste

$$\rho_{k_j} = \frac{f(x_k) - f(x_k + s_{k_j})}{m_k(x_k) - m_k(x_k + s_{k_j})}.$$

Ahora, si tenemos una muy buena iteración (es decir, $\rho_{k_0} \ge \eta_2$ y $\|s_{k_0}\| = \Delta_{k_0}$), se busca extender el radio de la región de confianza en alguna constante c y se calcula s_{k_1} con el radio $\Delta_{k_1} = c\Delta_{k_0}$. Aquí tenemos tres posibilidades. Primero, si $\rho_{k_1} \le \eta_1$, entonces tomamos como nuevo punto a $x_{k+1} = x_k + s_{k_0}$ y el radio $\Delta_{k+1} = \Delta_{k_0}$. Segundo, si $\rho_{k_1} \ge \eta_1$ y $\|s_{k_1}\| < \Delta_{k_1}$, entonces tomamos $x_{k+1} = x_k + s_{k_1}$ y $\Delta_{k+1} = \Delta_{k_1}$. Por

[1] Donde m es una función que depende de la iteración k.

[2] Este algoritmo es de hecho una modificación del [2, Algoritmo 10.5.1]

último, si $\rho_{k_1} \geq \eta_1$ y $\|s_{k_1}\| = \Delta_{k_1}$, entonces se repite el proceso para extender la región. La constante c se puede tomar fija o como una función del contador j, de modo que aumente cuando j aumenta. Ver Algoritmo 2.

Algorithm 2:

Entrada: Punto inicial x_0, radio inicial $\Delta_0 > 0$, constantes $0 \leq \eta_1 \leq \eta_2 < 1$, $0 < \gamma_1 < \gamma_2 \leq 1 < \gamma_3$ y $\epsilon > 0$.

Paso 0: Tomar $k = j = 0$.

Paso 1: Seleccionar m_k definido en la bola de radio Δ_k centrada en x_k.

Paso 2: Paso 2a. Tomar $\Delta_{k_j} = \Delta_k$.

Paso 2b. Calcular s_{k_j} que reduzca m_k tal que $\|s_{k_j}\| \leq \Delta_k$.

Paso 2c. Calcular la medida de ajuste ρ_{k_j}

Paso 2d. Si $\rho_{k_j} \geq \eta_2$ y $\|s_{k_j}\| = \Delta_{k_j}$, entonces hacer $\Delta_{k_{j+1}} = (1 + \gamma_1/3)^{j+1}\Delta_{k_j}$, $j = j + 1$ e ir al **Paso 2b**.

Paso 2e. Si $\rho_{k_j} \geq \eta_2$ y $\|s_{k_j}\| < \Delta_{k_j}$, entonces hacer $\Delta_{k+1} = \gamma_3\Delta_{k_j}$ y $x_{k+1} = x_k + s_{k_j}$ e ir al **Paso 3**.

Paso 2f. Si $\eta_1 \leq \rho_{k_j} \leq \eta_2$, hacer $\Delta_{k+1} = \gamma_2\Delta_{k_j}$ y $x_{k+1} = x_k + s_{k_j}$ e ir al paso **Paso 3**.

Paso 2g. Si $\rho_{k_j} < \eta_1$ y $j \geq 1$, hacer $\Delta_{k+1} = \Delta_{k_{j-1}}$ y $x_{k+1} = x_k + k_{j-1}$ e ir al **Paso 3**.

Paso 2h. Si $\rho_{k_j} < \eta_1$ y $j = 0$, hacer $\Delta_{k+1} = \gamma_1\Delta_k$, $x_{k+1} = x_k$ e ir al **Paso 3**.

Paso 3: Si $\|\nabla f(x_{k+1})\| < \epsilon$, entonces **Termina**, en otro caso hacer $k = k + 1$, $j = 0$ e ir al **Paso 1**.

Salida : $\mathbf{x}_k + 1$

Análisis de convergencia

Este algoritmo difiere del tradicional en la extensión de la región de confianza, de esta manera, basta entonces demostrar que tanto el número de muy buenas iteraciones como el número de iteraciones muy exitosas es finito.

Lema 1: El número de muy buenas iteraciones para cada iteración muy exitosa es finito.

Demostración. Sea k el índice de una iteración muy exitosa y θ_k el conjunto de todos los índices de muy buenas iteraciones en la iteración k. Procedamos por contradicción, es decir, supongamos que $|\theta_k| = \infty$. Como estas son muy buenas iteraciones y esperamos que el modelo decrezca por lo menos como lo hace con el paso de Cauchy, tenemos que

$$f(x_k) - f(x_k + s_{k_j}) \geq \eta_2|\theta_k|\kappa\varepsilon \min\left\{\frac{\varepsilon}{\kappa'}, \kappa''\right\},$$

donde $\kappa \in (0,1), \kappa' > 0, \kappa'' > 0, \varepsilon > 0$. Entonces, $f(x_k) - f(x_k + s_{k_j})$ no está acotada, lo cual contradice a la suposición 1. Entonces $|\theta_k| < \infty$ y el lema queda demostrado.

Lema 2: El número de iteraciones muy exitosas es finito.

Demostración. Análogamente al lema anterior, procedemos por contradicción. Construimos Γ_k, el conjunto de todos los índices correspondientes a iteraciones muy exitosas hasta la iteración k. Suponemos que $|\Gamma_k| \xrightarrow{k\to\infty} \infty$, entonces vemos que la diferencia entre las iteraciones muy exitosas r_0 y r_{k+1} no es acotado. Esto es

$$f(x_{r_0}) - f(x_{r_{k+1}}) \geq \eta_1|\Gamma_k|\zeta_{r_0,r_{k+1}}|\kappa_1\varepsilon \min\left\{\frac{\varepsilon}{\kappa_2}, \kappa_3\right\},$$

donde las constantes son como antes y

$$\zeta_{\ell_1,\ell_2} = \{i : \ell_1 \leq i \leq \ell_2, \rho_i \geq \eta_1\}$$

es el conjunto de iteraciones exitosa desde la iteración ℓ_1 hasta ℓ_2. Entonces, como en el límite esto contradice la suposición 1, $|\Gamma_k| < \infty$ y queda probado el lema.

Teorema 1: El Algoritmo 1 termina en un número finito de iteraciones o genera una secuencia $\{x_k\}$ que satisface

$$\lim_{k\to\infty} \text{ínf}\, \|g_k\| = 0$$

Demostración. Basta demostrar el caso cuando no termina en un número finito de iteraciones. Procedemos por contradicción, supongamos que existe $\epsilon > 0$ tal que $\|g_k\| > \epsilon$. Por el Teorema 6.4.3 de [2], el radio de la región está acotado inferiormente y alejado del cero, y en consecuencia deben existir un número infinito de iteraciones exitosas. Más aún, tenemos que

$$f(x_0) - f(x_{k+1}) \geq |\zeta_{0,k}|\kappa_1\epsilon\eta_1 \,\text{mín}\left\{\frac{\epsilon}{\kappa_2}, \kappa_3\right\} > 0.$$

Sin embargo, esto contradice la suposición 1, con lo que demuestra lo deseado.

Algoritmo 2

En este algoritmo, se implementa un método de búsqueda en línea para encontrar un tamaño de paso $\tilde{\alpha} \geq 1$ cuando tenemos una muy buena iteración, esto es, cuando $\rho_k \geq \eta_2$ y $\|s_k\| = \Delta_k$, se busca $\tilde{\alpha}$ tal que maximice la medida de ajuste del modelo

$$\tilde{\alpha} = \arg\max_{\alpha\geq 1}\left\{\psi_k(\alpha) = \frac{f(x_k) - f(x_k + \alpha s_k)}{m_k(x_k) - m_k(x_k + \alpha s_k)}\right\}. \quad (4)$$

Entonces, si $m_k(x_k + \tilde{\alpha}s_k) < m_k(x_k + s_k)$, se acepta $x_k + \tilde{\alpha}s_k$ como el nuevo punto de la iteración.

Veamos qué hace esta modificación en términos del modelo y la función. Primero notemos que como hablamos de una muy buena iteración, $\psi_k(1) = \rho_k \geq \eta_2$ y $\alpha \geq 1$, por lo tanto, $\psi_k(\tilde{\alpha}) \geq \psi(\alpha) \geq \eta_2$. Más aún, como esperamos que el modelo decrezca más que con el paso de Cauchy

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa\|g_k\| \,\text{mín}\left\{\frac{\|g_k\|}{\beta_k}, \Delta_k\right\}.$$

Si se acepta el nuevo paso de la iteración, entonces $m_k(x_k + \tilde{\alpha}s_k) < m_k(x_k)$ y, por lo tanto, $f(x_k + \tilde{\alpha}s_k) < f(x_k)$. Además,como $\tilde{\alpha}$ maximiza ψ_k, se debe cumplir $f(x_k + \tilde{\alpha}s_k) \leq f(x_k + s_k)$. En consecuencia,

$$f(x_k + \tilde{\alpha}s_k) \leq f(x_k + s_k) < f(x_k),$$

es decir que el algoritmo encuentra un mejor punto sin empeorar la medida de ajuste. Ver Algoritmo 3

Algorithm 3:

Entrada: Punto inicial x_0, radio inicial $\Delta_0 > 0$, constantes $0 \leq \eta_1 \leq \eta_2 < 1$, $0 < \gamma_1 < \gamma_2 \leq 1 < \gamma_3$ y $\epsilon > 0$.

Paso 0: Tomar $k = 0$.

Paso 1: Seleccionar m_k definido en la bola de radio Δ_k centrada en x_k.

Paso 2: Determinar s_k que reduzca m_k tal que $\|s_k\| \leq \Delta_k$. Calcular la medida de ajuste ρ_k

Paso 3: Si $\rho_k \geq \eta_2$ y $\|s_k\| = \Delta_k$, entonces calcular $\tilde{\alpha}$, tomar $x_{k+1} = x_k + \tilde{\alpha}s_k$ y establecer el nuevo radio como $\Delta_{k+1} = \|x_{k+1} - x_k\|$. Ir al **Paso 5**.

Paso 4: Si $\rho_k \geq \eta_2$ y $\|s_k\| < \Delta_k$, entonces hacer $\Delta_{k+1} = \gamma_3\Delta_k$ y $x_{k+1} = x_k + s_k$. En cambio si $\rho_k \geq \eta_1$, hacer $\Delta_{k+1} = \gamma_2\Delta_k$ y $x_{k+1} = x_k + s_k$. En otro caso, hacer $\Delta_{k+1} = \gamma_1\Delta_k$ y $x_{k+1} = x_k$.

Paso 5: Si $\|\nabla f_{k+1}\| < \epsilon$, **Termina**. Si no, hacer $k = k + 1$ e ir al **Paso 1**.

Salida : $\mathbf{x}_k$

Para determinar $\tilde{\alpha}$ usamos el método de forward-tracking, donde se impone la condición $m_k(x_k + \tilde{\alpha}s_k) < m_k(x_k + s_k)$. Ver Algortimo 4.

Análisis de convergencia

De nuevo, este algoritmo es muy parecido al algoritmo convencional de región de confianza, la única diferencia está en que ahora se extiende el radio de la región cuando tenemos una muy buena iteración. Notemos que el algoritmo de forward tracking está bien definido,

Algorithm 4: Método de Forward-tracking para resolver (4).

Entrada: Constantes $K > 0$ y $\rho < 1$.

Tomar $\tilde{\alpha} = 1$, $\alpha = \rho\tilde{\alpha}$ y $k = 1$. **while**

$\psi_k(\alpha) > \psi_k(\tilde{\alpha})$ *y* $m_k(x_k + \alpha s_k) < m_k(x_k + \tilde{\alpha} s_k)$ *y*

$k < K$ **do**

$\tilde{\alpha} = \alpha$,

$\alpha = \rho\alpha$,

$k = k + 1$

end

Salida : $\tilde{\alpha}$

ya que $\tilde{\alpha} < \infty$, pues tiene un criterio de paro con un número máximo de iteraciones.

Por otro lado, por el Teorema 4.1 de [1], tenemos que existe $\lambda \geq 0$ tal que $(B_k + \lambda I)s_k = -g_k$, donde $(B_k + \lambda I) \geq 0$, de modo que

$$m_k(x_k) - m(x_k + \tilde{\alpha} s_k) = -\frac{1}{2}\tilde{\alpha} s_k^T g_k + \frac{1}{2}\tilde{\alpha}^2 \lambda \|s_k\|^2.$$

Por otro lado, por el Lema 2.4 de [5],

$$\tilde{\alpha} s_k^T g_k \leq -\frac{1}{2}\tilde{\alpha}\|g_k\| \text{mín}\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}.$$

Con lo que se sigue el siguiente resultado.

Lema 3: Existe un escalar $\beta \in (0, 1]$ tal que

$$m_k(x_k) - m_k(x_{k+1}) \geq \beta\|g_k\| \text{mín}\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}, k = 0, 1, 2, \ldots$$

Teorema 2: El Algoritmo 2 termina en un número finito de iteraciones o genera una sucesión infinita $\{x_k\}$ que satisface

$$\lim_{k\to\infty} \text{ínf} \|g_k\| = 0.$$

Demostración: Basta probar el caso cuando el algoritmo no termina en un número finito de iteraciones. Procedamos por contradicción, supongamos que existe una constante $\varepsilon > 0$, tal que $\|g_k\| > \varepsilon$, para toda k. Usando el lema anterior se puede demostrar que

$$\sum_{k=1}^{\infty}(f(x_k) - f(x_{k+1})) \geq \beta\eta_1 \sum_{k\in\zeta} \|g_k\| \text{mín}\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\},$$

donde ζ es el conjunto de todas las iteraciones exitosas. Sin embargo, por la suposición 1, se tiene que $\{f(x_k)\}$ está acotada inferiormente, con lo que $f(x_k) - f(x_{k+1})$ debe decrecer y necesariamente

$$\lim_{k\to\infty, k\in\zeta} \|g_k\| \text{mín}\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\} = 0,$$

mas aún, como $\varepsilon > 0$ no depende de k, lo anterior se reduce a

$$\lim_{k\to\infty} \text{mín}\left\{\Delta_k, \frac{\varepsilon}{\|B_k\|}\right\} = 0.$$

Esto tiene dos implicaciones, que $\Delta_k \to 0$ o $\|B_k\| \to \infty$. Sabemos que la primera implicación no puede ser cierta, en efecto, como $\|g_k\| \gg 0$, por el Teorema 6.4.3 de [2], se tiene que $\{\Delta_k\} \gg 0$. Entonces, debe ser que B_k no sea acotada, lo cual contradice la suposición 2. Por lo tanto, $\|g_k\| > \varepsilon$ no puede ser cierta y se demuestra lo deseado.

Resultados

Se implementaron los algoritmos modificados de región de confianza presentados en la sección anterior y se compararon con el Algoritmo 4.1 de [1]. Consideramos que un algoritmo había convergido si $\|g_k\| < 10^{-5}$, y se tomaron los siguientes parámetros $\Delta_0 = 0.1\|g(x_0)\|$, $\eta_1 = 0.1$, $\eta_2 = 0.9$, $\gamma_1 = 0.5$, $\gamma_2 = 1$ y $\gamma_3 = 3$. Se consideraron 60 problemas de la colección CUTEst[3] y se tomó el punto inicial sugerido en la misma, para resolver (2) optamos por el algoritmo de Dogleg.

Para evaluar el Algoritmo 2, Algoritmo 3 y Algoritmo 4.1 de [1], usamos la siguiente estrategia: sean S el conjunto de todos los algoritmos probados y P el conjunto de todos los problemas a resolver con dichos algoritmos, sea $t_{p,s}$ el tiempo que le toma al algoritmo s resolver el problema p, la tasa de desempeño se define como

$$r_{p,s} = \frac{t_{p,s}}{\text{mín}_{s\in S}\{t_{p,s}\}},$$

[3] https://github.com/jfowkes/pycutest

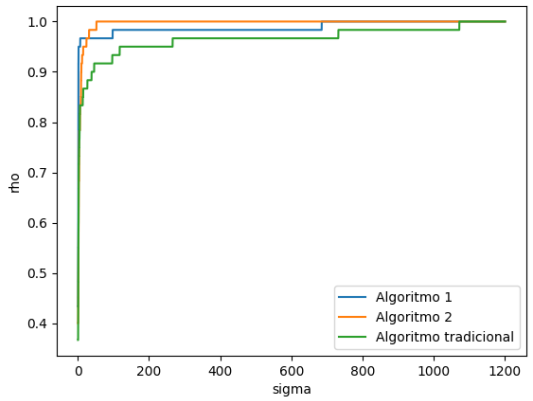

Figura 2. Perfiles de desempeño del número total de evaluaciones ($\#f + n\#g$).

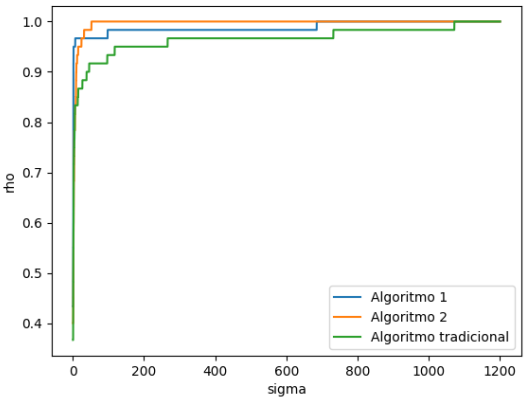

Figura 3. Perfiles de desempeño del número de evaluaciones de la función objetivo ($\#f$).

y el perfil de desempeño como

$$\rho_s(\sigma) = \frac{|\{p \in P : r_{p,s} \leq \sigma\}|}{|P|},$$

donde $\sigma \in \mathbb{R}$. Esta función se anula para $\sigma < 1$ y es creciente. Entonces, el perfil de desempeño de nivel σ, $\rho_s(\sigma)$, es la proporción de problemas que tienen una tasa de desempeño menor que σ. También podemos construir estas medidas, sustituyendo a $t_{s,p}$ por el número de evaluaciones de f ($\#f$), de su gradiente g ($\#g$) y las totales[4] ($\#f + n\#g$), respectivamente. Las gráficas de estos perfiles se encuentran en las figuras 2-5.

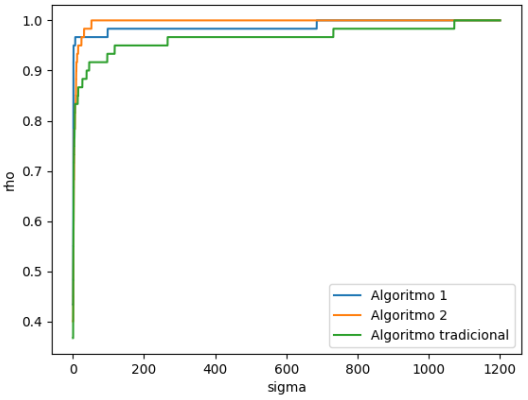

Figura 4. Perfiles de desempeño del número de evaluaciones del gradiente ($\#g$).

Conclusiones

Para los problemas de grandes escalas, como NONDQUAR, con 5000 variables, los algoritmos claramente obtuvieron una gran ventaja, en iteraciones y tiempo, respecto al algoritmo tradicional, de la misma manera para problemas como LIARWHD, ENGVAL1 y POWELLSG, entre otros. En cuanto al rendimiento computacional, el Algoritmo 3 hace menos evaluaciones tanto de la función como del gradiente, comparado

[4]Donde n es el número de variables en cada problema.

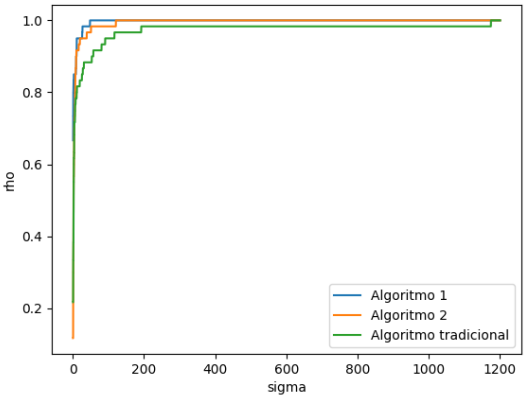

Figura 5. Perfiles de desempeño del tiempo computacional.

a los demás, esto es conveniente sobre todo en problemas de gran escala. Respecto al tiempo, el Algoritmo 2 resuelve la mayoría de los problemas en menos tiempo, incluso en los más grandes como SPARSINE, en [4] reportan un tiempo de 59,838.67 segundos, mientras que en este trabajo obtuvimos un valor de 2,134.36 segundos, ver Tabla III.

Referencias

[1] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[2] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust region methods*. SIAM, 2000.

[3] M. Rezapour, W. S. U. D. of Mathematics, and Statistics, *Trust-Region Methods for Unconstrained Optimization Problems*. Washington State University, 2020. [Online]. Available: https://books.google.com.mx/books?id=9pgxzgEACAAJ

[4] M. Rezapour and T. J. Asaki, "Adaptive trust-region algorithms for unconstrained optimization," *Optimization Methods and Software*, pp. 1–23, 2019.

[5] J. Nocedal, Y. Yuan, and Y. Xiang, "Combining trust-region and line-search techniques," *Optimization Technology Center*, 1998.

[6] E. M. Gertz, "A quasi-newton trust-region method," *Mathematical Programming*, pp. 447—470, 2004.

[7] F. Bastin, V. Malmedy, M. Mouffe, P. L. Toint, and D. Tomanos, "A retrospective trust-region method for unconstrained optimization," *Mathematical Programming*, pp. 395–418, 2010.

[8] N. Deng, Y. Xiao, and F. Zhou, "Nonmonotonic trust region algorithm," *Journal of Optimization Theory and Applications*, pp. 259—285, 1993.

[9] J.-L. Zhang and X.-S. Zhang, "A nonmonotone adaptive trust region method and its convergence," *Computers and Mathematics with Applications*, vol. 45, no. 10, pp. 1469–1477, 2003.

[10] S. Wenyu, "Nonmonotone trust region method for solving optimization problems," *Applied Mathematics and Computation*, vol. 156, no. 1, pp. 159–174, 2004.

Tabla I

RESULTADOS RESOLVER 60 PROBLEMAS DE LA COLECCIÓN CUTEST CON 3 ALGORITMOS DIFERENTES. SE MUESTRAN EL NÚMERO DE VARIABLES DEL PROBLEMA (N), EL TIEMPO DE CÓMPUTO EN SEGUNDOS, EL NÚMERO DE ITERACIONES (#ITER), EL NÚMERO DE EVALUACIONES DE LA FUNCIÓN OBJETIVO ($\#f$), EL NÚMERO DE EVALUACIONES DEL GRADIENTE ($\#g$), EL NÚMERO DE EVALUACIONES TOTALES ($\#f + n\#g$), EL VALOR DE LA FUNCIÓN OBJETIVO (f) Y EL VALOR DE LA NORMA DEL GRADIENTE ($\|g\|$).

		Algoritmo tradicional							Algoritmo 1							Algoritmo 2						
Problema	n	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$
AKIVA	2	1.13594617	17	52	52	156	6.166042212	2.97E-06	1.580695465	6	31	31	93	6.166042212	1.39E-09	1.714781012	38	115	115	345	6.166042212	2.72E-06
ARGLINA	200	39.06829032	5	16	16	3216	200	7.44E-13	14.078792	6	30	30	6030	200	1.52E-13	50.61327064	6	22	22	4422	200	2.82E-13
BARD	3	0.107422072	48	145	145	580	0.00821488	6.85E-06	0.036441703	9	46	46	184	0.008214877	5.13E-06	0.145385934	55	166	166	664	0.008214879	4.60E-06
BDQRTIC	5000	599.9762592	11	34	34	170034	20006.25688	1.00E-06	472.9373374	9	46	46	230046	20006.25688	2.85E-06	562.8714066	9	28	28	140028	20006.25688	2.85E-06
BEALE	2	0.002386326	7	22	22	66	2.25E-18	5.71E-09	0.022058053	7	36	36	108	2.25E-18	5.71E-09	0.003431419	7	22	22	66	2.25E-18	5.71E-09
BOX3	3	0.003132959	7	22	22	88	2.88E-11	3.15E-06	0.082386894	7	36	36	144	2.55E-11	2.95E-06	0.004428061	7	22	22	88	2.55E-11	2.95E-06
BRKMCC	2	0.001967749	2	7	7	21	0.169042679	6.11E-06	0.053478599	2	11	11	33	0.169042679	6.11E-06	0.014157167	2	7	7	21	0.169042679	6.11E-06
BROWNAL	200	155.3953438	25	76	76	15276	4.62E-16	1.35E-07	14.44238116	6	31	31	6231	1.74E-16	8.38E-08	287.2663514	26	282	282	56682	8.31E-22	9.10E-10
BROWNBS	2	3.255043168	10000	30001	30001	90003	6.29813E+11	1587231.888	0.027839846	8	41	41	123	0	0	0.086723286	36	109	109	327	1.97E-31	8.88E-10
CHNROSNB	50	59.99525619	241	724	724	36924	2.91E-12	2.30E-06	2.131658454	34	171	171	8721	3.75E-13	6.46E-06	19.75226314	227	682	682	34782	1.37E-18	2.40E-08
CLIFF	2	0.121935622	27	82	82	246	0.199786614	9.58E-10	0.053334863	27	136	136	408	0.199786614	9.58E-10	0.212905088	27	82	82	246	0.199786614	9.58E-10
CUBE	2	0.64072485	190	571	571	1713	1.43E-21	3.20E-11	0.020573469	8	41	41	123	7.05E-17	5.31E-07	0.1678694	183	1296	1296	3888	2.82E-19	4.41E-10
DECONVU	51	0.961378039	49	148	148	7696	7.25E-08	9.83E-06	0.487534451	28	141	141	7332	3.92E-08	6.55E-06	19.11006688	39	118	118	6136	3.18E-08	4.76E-06
DENSCHNA	2	0.06569743	5	16	16	48	2.21E-12	2.97E-06	0.010676652	5	26	26	78	2.21E-12	2.97E-06	0.018512489	5	16	16	48	2.21E-12	2.97E-06
DENSCHNB	2	0.059104418	23	70	70	210	5.00E-12	4.53E-06	0.016957293	7	35	35	105	4.38E-18	4.58E-09	0.27883013	7	225	225	675	5.88E-17	1.60E-08
DENSCHNC	2	0.048399836	18	55	55	165	8.19E-23	4.56E-11	0.020398105	10	51	51	153	2.18E-20	7.90E-10	0.028525345	24	73	73	219	4.08E-20	1.06E-09
DENSCHNE	3	0.163668562	39	118	118	472	3.22E-12	3.59E-06	0.025208049	12	61	61	244	6.64E-17	1.63E-08	0.055698719	14	43	43	172	4.32E-12	4.16E-06
DENSCHNF	2	0.013595212	6	19	19	57	6.51E-22	6.29E-10	0.01273529	6	31	31	93	6.51E-22	6.29E-10	0.022006584	6	19	19	57	6.51E-22	6.29E-10
DIXMAANA	3000	569.5216892	11	34	34	102034	1	0	234.3601164	9	46	46	138046	1	1.71E-18	250.5543848	7	22	22	66022	1	6.55E-21
DIXMAANB	3000	544.8022571	10	31	31	93031	1	1.04E-08	128.3231897	7	36	36	108036	1	1.18E-12	243.5596442	7	22	22	66022	1	1.18E-12

Tabla II

RESULTADOS RESOLVER 60 PROBLEMAS DE LA COLECCIÓN CUTEST CON 3 ALGORITMOS DIFERENTES. SE MUESTRAN EL NÚMERO DE VARIABLES DEL PROBLEMA (N), EL TIEMPO DE CÓMPUTO EN SEGUNDOS, EL NÚMERO DE ITERACIONES (#ITER), EL NÚMERO DE EVALUACIONES DE LA FUNCIÓN OBJETIVO ($\#f$), EL NÚMERO DE EVALUACIONES DEL GRADIENTE ($\#g$), EL NÚMERO DE EVALUACIONES TOTALES ($\#f + n\#g$), EL VALOR DE LA FUNCIÓN OBJETIVO (f) Y EL VALOR DE LA NORMA DEL GRADIENTE ($\|g\|$).

		Algoritmo tradicional							Algoritmo 1							Algoritmo 2						
Problema	n	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$
DIXMAANC	3000	640.8404217	11	34	34	102034	1	3.02E-06	144.6321357	8	41	41	123041	1	3.94E-14	960.738712	33	100	100	300100	1	2.80E-15
DIXMAAND	3000	683.7724958	22	67	67	201067	1	4.02E-18	179.4459041	9	46	46	138046	1	2.68E-12	1011.182518	33	100	100	300100	1	4.53E-14
DIXMAANE	3000	481.1997446	16	49	49	147049	1	8.53E-10	447.9727082	19	96	96	288096	1.000000001	4.01E-06	982.9960545	30	91	91	273091	1.000000001	9.17E-07
DIXMAANF	3000	634.5850847	20	61	61	183061	1	8.19E-09	268.9247992	14	71	71	213071	1	5.22E-07	1007.511692	31	94	94	282094	1.000000001	4.80E-06
DIXMAANG	3000	546.8021972	20	61	61	183061	1	7.13E-08	268.7172431	14	71	71	213071	1.000000001	6.16E-06	521.7211791	14	43	43	129043	1.000000001	6.16E-06
DIXMAANI	3000	678.9762518	28	85	85	255085	1	1.62E-11	223.2056234	10	51	51	153051	1	1.34E-06	494.5862124	15	46	46	138046	1	3.35E-10
DIXMAANK	3000	547.6190428	21	64	64	192064	1.000000007	2.61E-06	398.5814786	22	111	111	333111	1.000000001	1.12E-06	990.8297823	34	103	103	309103	1.000000001	8.64E-07
DIXMAANL	3000	987.5108448	43	130	130	390130	1	3.06E-08	381.5351776	19	96	96	288096	1.000000021	7.06E-06	398.9114597	19	58	58	174058	1.000000021	7.06E-06
DJTL	2	0.280603622	810	2431	2431	7293	-8951.544724	9.18E-08	2.653027712	3333	16666	16666	49998	-8951.544724	2.14E-07	0.948333195	487	5421	5421	16263	-8951.544724	4.15E-07
EDENSCH	2000	590.6602761	32	97	97	194097	12003.28459	1.55E-07	75.00127879	12	61	61	122061	12003.28459	1.63E-10	341.3721765	41	124	124	248124	12003.28459	2.88E-10
EG2	1000	136.5886345	38	115	115	115115	-998.9473933	7.34E-14	6.993946958	4	21	21	21021	-998.9473933	4.92E-09	846.3730684	28	1100	1100	1101100	-998.9473933	1.48E-13
ENGVAL1	5000	570.6866508	10	31	31	155031	5548.668419	2.25E-06	372.4995978	7	36	36	180036	5548.668419	2.69E-06	273.3692771	7	22	22	110022	5548.668419	2.69E-06
GROWTHLS	3	0.720245914	1654	4963	4963	19852	1.004040584	8.36E-06	0.330364391	65	326	326	1304	3542.149032	9.95E-06	0.53760439	1034	3103	3103	12412	1.004040584	2.73E-08
GULF	3	10.74605612	10000	30001	30001	120004	5.472838931	0.138642316	0.097328889	10	49	49	196	32.835	0	0.009151645	8	28	28	112	32.835	0
HATFLDD	3	0.015611064	25	76	76	304	6.62E-08	1.20E-06	0.171665655	33	166	166	664	6.62E-08	2.87E-06	0.046634908	32	300	300	1200	6.62E-08	2.43E-08
HELIX	3	0.043720075	107	322	322	1288	2.21E-11	9.46E-06	0.071930667	13	66	66	264	4.81E-17	5.99E-08	0.091433063	65	602	602	2408	7.45E-23	6.09E-11
HILBERTA	2	0.003706381	9	28	28	84	2.05E-33	1.85E-17	0.010764804	9	46	46	138	2.05E-33	1.85E-17	0.00438515	11	34	34	102	0	0
HILBERTB	10	0.011723011	5	16	16	176	4.08E-31	2.95E-15	0.008075428	3	16	16	176	3.23E-30	8.08E-15	0.011224029	5	16	16	176	4.21E-31	2.91E-15
HIMMELBB	2	0.072151565	24	73	73	219	1.38E-23	5.11E-10	3.485914834	10001	50006	50006	150018	7.19E-21	0.00017836	0.114405196	49	148	148	444	1.46E-25	5.25E-11
HIMMELBF	4	0.91013331	414	1243	1243	6215	318.5717488	8.94E-06	5.48879108	10001	50006	50006	250030	455.6600116	0.002673006	0.634683167	170	511	511	2555	318.5717488	9.86E-06

Tabla III

RESULTADOS RESOLVER 60 PROBLEMAS DE LA COLECCIÓN CUTEST CON 3 ALGORITMOS DIFERENTES. SE MUESTRAN EL NÚMERO DE VARIABLES DEL PROBLEMA (N), EL TIEMPO DE CÓMPUTO EN SEGUNDOS, EL NÚMERO DE ITERACIONES (#ITER), EL NÚMERO DE EVALUACIONES DE LA FUNCIÓN OBJETIVO ($\#f$), EL NÚMERO DE EVALUACIONES DEL GRADIENTE ($\#g$), EL NÚMERO DE EVALUACIONES TOTALES ($\#f + n\#g$), EL VALOR DE LA FUNCIÓN OBJETIVO (f) Y EL VALOR DE LA NORMA DEL GRADIENTE ($\|g\|$).

		Algoritmo tradicional							Algoritmo 1							Algoritmo 2						
Problema	n	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$	Tiempo (s)	#iter	$\#f$	$\#g$	$\#f + n\#g$	f	$\|g\|$
HIMMELBH	2	0.005298208	7	22	22	66	-1	4.67E-10	0.004500494	7	35	35	105	-1	1.36E-10	0.041601617	6	222	222	666	-1	1.00E-07
LIARWHD	5000	6414.592997	136	409	409	2045409	1.42E-21	1.06E-08	550.4252086	13	66	66	330066	6.38E-22	4.07E-09	2205.027118	54	163	163	815163	0	0
MANCINO	100	69.42869402	44	133	133	13433	1.83E-21	1.20E-07	28.76550226	13	66	66	6666	1.48E-21	1.08E-07	196.3072499	64	396	396	39996	1.82E-21	1.20E-07
MSQRTALS	1024	545.1097121	100	301	301	308525	2.38E-21	1.60E-11	134.7042145	23	116	116	118900	1.60E-11	5.17E-07	562.747129	73	220	220	225500	1.09E-17	2.26E-09
NONDIA	5000	838.2655279	19	58	58	290058	3.04E-14	1.63E-06	212.4561974	5	26	26	130026	5.27E-16	3.64E-08	1498.442676	33	100	100	500100	3.38E-14	1.82E-06
NONDQUAR	5000	727.388863299	18	55	55	275055	1.05E-09	6.20E-06	663.645793411	18	91	91	455091	1.05E-09	6.20E-06	707.346171622	18	55	55	275055	1.05E-09	6.20E-06
PALMER1D	7	0.568825348	516	1549	1549	12392	0.652673985	1.88E-08	0.010681586	3	16	16	128	0.652673985	5.73E-09	0.060599905	22	67	67	536	0.652673985	1.20E-06
PALMER2C	8	0.885839643	206	619	619	5571	0.014368886	8.53E-10	0.009744485	3	16	16	144	0.014368886	3.81E-06	0.498201614	80	241	241	2169	0.014368886	2.46E-08
PALMER3C	8	0.046739531	13	40	40	360	0.019537639	1.12E-09	0.00753173	2	11	11	99	0.019537639	1.68E-07	0.075179793	44	133	133	1197	0.019537639	2.08E-09
PALMER4C	8	0.712531376	245	736	736	6624	0.050310687	1.29E-06	0.012352871	3	16	16	144	0.050310687	8.30E-08	0.078464233	45	136	136	1224	0.050310687	6.24E-09
PALMER5C	6	0.002310461	1	4	4	28	2.128086643	1.83E-13	0.003455441	1	6	6	42	2.128086643	1.83E-13	0.005048339	1	4	4	28	2.128086643	1.83E-13
PALMER6C	8	0.853945797	627	1882	1882	16938	0.016387438	3.94E-08	0.010579911	3	16	16	144	0.016387438	2.60E-09	0.021764834	12	37	37	333	0.016387438	2.22E-09
PALMER7C	8	0.00344848	2	7	7	63	0.601987195	1.75E-09	0.007351358	2	11	11	99	0.601987195	1.75E-09	0.004431787	2	7	7	63	0.601987195	1.75E-09
PALMER8C	8	1.860800539	1420	4261	4261	38349	0.159767833	6.32E-09	0.009682234	3	16	16	144	0.159767833	9.07E-09	0.024540288	15	46	46	414	0.159767833	3.39E-09
POWELLSG	5000	1009.373544	24	73	73	365073	6.05E-08	5.15E-06	715.264745	18	91	91	455091	4.22E-08	4.96E-06	660.2499926	18	55	55	275055	4.22E-08	4.96E-06
ROSENBR	2	1.576129364	233	700	700	2100	1.90E-15	4.41E-08	0.068863607	5	26	26	78	1.85E-11	8.61E-06	0.061471459	75	632	632	1896	2.43E-14	1.56E-07
S308	2	0.006885458	32	97	97	291	0.773199056	3.16E-06	0.027457129	9	46	46	138	0.773199056	2.10E-06	0.002474971	11	34	34	102	0.773199056	3.91E-10
SINEVAL	2	0.070813582	263	790	790	2370	2.67E-15	2.10E-07	0.09132571	28	140	140	420	2.40E-25	4.38E-11	0.088421923	167	880	880	2640	3.68E-13	2.50E-06
SISSER	2	0.003039933	12	37	37	111	1.07E-08	5.52E-06	0.005144186	12	61	61	183	1.07E-08	5.52E-06	0.002855368	12	37	37	111	1.07E-08	5.52E-06
SPARSINE	5000	2567.133794209	44	133	133	665133	2.20E-09	5.76E-06	2134.364003745	32	161	161	805161	1.70E-09	6.27E-06	4945.121866414	50	151	151	755151	1.40E-08	9.04E-06
VAREIGVL	5000	1023.007571	20	61	61	305061	1.14E-09	3.06E-06	900.5948148	16	81	81	405081	4.26E-10	2.99E-06	2059.92723	47	142	142	710142	5.90E-09	9.57E-06