El método de Nelder–Mead para minimización irrestricta sin derivadas

Elvio A. Pilotta

Resumen. En este trabajo presentamos el método de Nelder–Mead, junto a algunos resultados teóricos recientes y ejemplos numéricos.

1 Introducción

En ocasiones se desea encontrar el mínimo de una función a valores reales, definida en todo el espacio n-dimensional, y cuyas derivadas no están disponibles. Esto puede ocurrir, por ejemplo, cuando las derivadas parciales de la función son computacionalmente muy caras de ser calculadas o simplemente porque no se tiene una expresión explícita de las derivadas parciales de la función. Para este tipo de problemas existen métodos de optimización llamados de búsqueda directa, los cuales son muy fáciles de implementar, no requieren calcular derivadas y en general son robustos, aún para funciones no convexas o con discontinuidades.

Una idea intuitiva y natural para problemas de dimensión 2 ó 3 se origina en el conocido método de bisección para ecuaciones no lineales unidimensional, y en la cual se consideran los valores de la función en un conjunto de puntos del plano o del espacio (respectivamente) a fin de localizar el mínimo de la función. Basados en esta idea fue propuesto el método de simplex de Spendley, Hext and Himsworth en 1962 [11] y posteriormente en 1965, el método de Nelder and Mead [8]. A pesar de usar el mismo concepto geométrico no deben confundirse estos métodos con el famoso método simplex para programación lineal de Dantzig [1].

Cada iteración comienza con un simplex en $\mathbb{R}^n$ dado por los $n+1$ vértices y los correspondientes valores funcionales. Luego de realizar algunos test en los valores de f en uno o dos puntos adicionales, la iteración del algoritmo termina con un nuevo simplex, distinto del inicial, de modo que los valores

de la función en los nuevos vértices satisfacen "alguna" condición de descenso comparada con el simplex inicial de esta iteración. Un rasgo interesante de este método a diferencia de otros métodos de búsqueda directa es el bajo número de evaluaciones de función extras que realiza por iteración. Existen métodos de búsqueda directa que realizan n evaluaciones por iteración. En general se conoce que el método funciona bien en dimensiones "pequeñas" y que suele fallar en dimensiones "grandes". El concepto de dimensión pequeña o grande ha variado considerablemente en los últimos años con el uso intensivo de las computadoras para el estudio y desarrollos de algoritmos y nuevos métodos numéricos. Por ejemplo, Torczon y Dennis [3, 12] dieron ejemplos en dimensión ocho donde el método no converge a un minimizador de una función suave.

El método de Nelder–Mead es ampliamente utilizado en química, ingeniería química y medicina. Además, desde su publicación se ha convertido en uno de los métodos más usados en programación no lineal sin restricciones. Para tener una idea de su importancia basta mencionar que es uno de los métodos incluídos en Numerical Recipes [9] y que es el método implementado en el conocido software MATLAB [6] para minimización sin restricciones. Desafortunadamente el método de Nelder–Mead, al igual que el de Spendley, Hext and Himsworth son métodos heurísticos y hasta ahora fueron probados muy pocos resultados de convergencia. Sorprendemente, luego de tres décadas de su publicación se ha despertado nuevamente el interés de científicos y matemáticos del área de optimización en tratar de probar resultados de convergencia para el método de Nelder–Mead. Siendo un método tan utilizado sería muy útil que sus propiedades teóricas sean entendidas tanto como sea posible.

En este trabajo veremos una descripción del método simplex de Nelder–Mead, mencionaremos algunos resultados teóricos recientes, veremos algunos ejemplos numéricos y finalmente presentaremos algunos comentarios finales.

2 Descripción del método simplex de Nelder-Mead

Al comienzo de cada iteración comenzamos con un *simplex no degenerado* en $\mathbb{R}^n$ y termina con otro simplex en $\mathbb{R}^n$ (distinto del anterior). Se define un simplex no degenerado en $\mathbb{R}^n$ como la cápsula convexa de $n+1$ puntos no-coplanares $x_1, x_2, \ldots, x_{n+1} \in \mathbb{R}^n$, es decir, no todos esos puntos están en un mismo hiperplano de $\mathbb{R}^n$. Supongamos que los vértices del simplex inicial están ordenados de modo tal que

$$f_1 \leq f_2 \leq \cdots \leq f_{n+1}, \tag{1}$$

donde $f_i = f(x_i)$.

Como estamos buscando el minimizador de f, decimos que x_1 es el *mejor* vértice y que x_{n+1} es el *peor*.

Llamamos *diámetro del simplex* S a

$$\operatorname{diam}(S) = \max_{1 \leq i,j \leq n+1} \|x_i - x_j\|.$$

Los parámetros ρ, δ, γ y σ son usados en cada iteración y deben satisfacer:

$$\delta > 1, \quad 0 < \rho < \delta, \quad 0 < \gamma < 1, \quad 0 < \sigma < 1.$$

Los valores *defaults* comúnmente usados son:

$$\rho = 1, \quad \delta = 2, \quad \gamma = \frac{1}{2} \quad \text{y} \quad \sigma = \frac{1}{2}.$$

Iteración k del Algoritmo de Nelder–Mead.

PASO 1: Ordenar. Ordenar los $n+1$ vértices del simplex como en (1).

PASO 2: Reflejar. Calcular el *centroide* de los n mejores puntos:

$$\widehat{x} = \sum_{i=1}^{n} \frac{x_i}{n}.$$

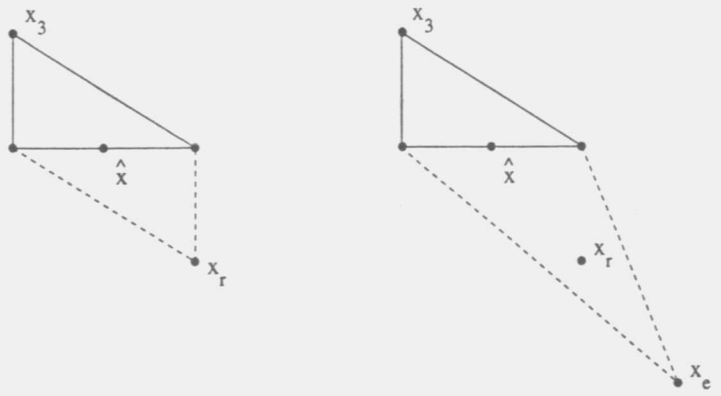

Fig. 1: Reflexión y expansión en el método de Nelder–Mead.

Calcular el *punto reflexión*:

$$x_r = \widehat{x} + \rho(\widehat{x} - x_{n+1}) = (1+\rho)\widehat{x} - \rho x_{n+1} \tag{2}$$

Calcular $f_r = f(x_r)$.

Si $f_1 \leq f_r < f_n$, aceptar x_r como nuevo vértice del simplex, eliminar el peor vértice y terminar la iteración. (Ver Fig. 1).

PASO 3: Expandir. Si $f_r < f_1$ calcular el *punto expansión*

$$x_e = \widehat{x} + \delta(x_r - \widehat{x}) = \widehat{x} + \rho\delta(\widehat{x} - x_{n+1}) = (1+\rho\delta)\widehat{x} - \rho\delta x_{n+1} \tag{3}$$

y evaluar $f_e = f(x_e)$. Si $f_e < f_r$ aceptar x_e, eliminar el peor vértice y terminar la iteración. Si no ($f_e \geq f_r$) aceptar x_r, eliminar el peor vértice y terminar la iteración. (Ver Fig. 1).

PASO 4: Contraer. Si $f_r \geq f_n$ realizar una contracción entre $\widehat{x}$ y el mejor entre x_{n+1} y x_r.

4.a. Contracción externa. Si $f_n \leq f_r < f_{n+1}$, calcular

$$x_{ce} = \widehat{x} + \gamma(x_r - \widehat{x}) = \widehat{x} + \rho\gamma(\widehat{x} - x_{n+1}) = (1+\rho\gamma)\widehat{x} - \rho\gamma x_{n+1} \tag{4}$$

y evaluar $f_{ce} = f(x_{ce})$.

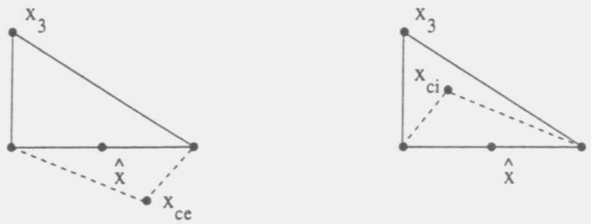

Fig. 2: Contracción interna y externa en el método de Nelder–Mead.

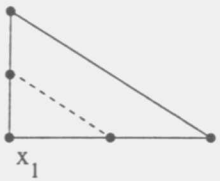

Fig. 3: Encogimiento en el método de Nelder–Mead.

Si $f_{ce} \leq f_r$, aceptar x_{ce}, eliminar el peor vértice y terminar la iteración. Sino, ir al Paso 5. (Ver Fig. 2).

4.b. Contracción interna. Si $f_r \geq f_{n+1}$, calcular

$$x_{ci} = \widehat{x} - \gamma(\widehat{x} - x_{n+1}) = (1-\gamma)\widehat{x} + \gamma x_{n+1} \tag{5}$$

y evaluar $f_{ci} = f(x_{ci})$. Si $f_{ci} < f_{n+1}$, aceptar x_{ci}, eliminar el peor vértice y terminar la iteración. Sino, ir al Paso 5. (Ver Fig. 2).

PASO 5: Encoger. Evaluar f en los n puntos $y_i = x_1 + \sigma(x_i - x_1)$, $i = 2, \ldots, n+1$. Los nuevos vértices del simplex en la próxima iteración serán $x_1, v_2, \ldots, v_{n+1}$. Ver Fig. 3).

3 Algunos resultados teóricos

Recordemos que estamos interesados en resolver el siguiente problema

$$\begin{aligned} &\text{Minimizar } f(x) \\ &x \in \mathbb{R}^n \end{aligned} \tag{6}$$

donde $f : \mathbb{R}^n \to \mathbb{R}$ no es necesariamente continua.

Es claro que el método de Nelder–Mead es iterativo. Luego, como es habitual al considerar algoritmos de minimización surgen algunos interrogantes naturales: ¿el algoritmo está bien definido?, ¿el algoritmo es convergente?, si converge, ¿converge a un minimizador de f?, es decir, ¿converge a una solución del problema (6)? A continuación analizaremos brevemente estos aspectos.

3.1 Buena definición del algoritmo.

El algoritmo estará bien definido si dado un simplex inicial, al comenzar cada iteración, siempre es posible obtener un (único) simplex final como resultado de seguir los pasos indicados en tal iteración. Es fácil ver que esto es cierto. El único inconveniente que podría surgir está relacionado al Paso 1: ¿cómo ordenar los vértices en el caso de tener valores funcionales iguales?. Esta consideración no fue tenida en cuenta en el artículo original de Nelder y Mead [8]. Un criterio simple fue propuesto por Lagarias, Reeds, M. Wright y P. Wright [5] el cual, básicamente, asigna al nuevo vértice el índice más alto posible consistente con la relación de orden (1).

3.2 Convergencia del algoritmo.

Los resultados de convergencia que se disponen actualmente, según el reciente trabajo de Lagarias, Reeds, M. Wright y P. Wright [5] se restringen sólo a funciones estrictamente convexas en dimensiones 1 y 2.

Definición 1 *Una función f es* estrictamente convexa *sobre $\mathbb{R}^n$ si para todo par de puntos y, z con $y \neq z$ y λ tal que $0 < \lambda < 1$, se tiene que*

$$f(\lambda y + (1-\lambda)z) < \lambda f(y) + (1-\lambda)f(z).$$

Si f es estrictamente convexa sobre $\mathbb{R}^n$ y

$$c = \sum_{i=1}^{m} \lambda_i z_i \quad \text{con } 0 < \lambda_i < 1 \quad \text{y} \sum_{i=1}^{m} \lambda_i = 1,$$

entonces

$$f(c) < \sum_{i=1}^{m} \lambda_i f(z_i) \quad \text{y por lo tanto } f(c) < \max\{f(z_1), \ldots, f(z_m)\}.$$

El siguiente resultado indica que si la función es estrictamente convexa entonces habrá descenso en los valores funcionales de los vértices del simplex o reducción en la cardinalidad de vértices con el mismo valor funcional que el peor. Una consecuencia que se deduce de la demostración es que si la función es estrictamente convexa nunca se alcanza el Paso 5 (encogimiento).

Teorema 1 *Sea f una función estrictamente convexa en $\mathbb{R}^n$ y sea S_I el simplex inicial en la iteración k. Entonces al finalizar la iteración k, el valor máximo de f en los vértices del simplex final $f_{\max}$ decrece estrictamente ó en el simplex final hay un vértice menos tal que $f(x_i) = f_{\max}$.*

Demostración:

En el Paso 1, ordenamos los vértices del simplex S_I. Así $f_{\max} = f_{n+1}$. Sea $A_{\max} = \{x_i \in S_I | f(x_i) = f_{\max}\}$. Claramente $A_{\max}$ es no vacío.

En el Paso 2 del Algoritmo calculamos x_r y $f_r = f(x_r)$.

Si $f_r < f_n$, termina la iteración y el valor máximo en el simplex final decrece.

Si no, esto es $f_r \geq f_n$, debemos ir al Paso 4 para realizar una contracción.

Si la card$(A_{\max}) = 1$ tenemos dos casos posibles:

Caso 1: si $f_n \leq f_r < f_{n+1}$ se realiza una contracción externa:

$$x_{ce} = \widehat{x} + \gamma(x_r - \widehat{x}) = \gamma x_r + (1-\gamma)\widehat{x}, \qquad 0 < \gamma < 1.$$

Como f es estrictamente convexa

$$f(x_{ce}) < \gamma f_r + (1-\gamma) f(\widehat{x}) < \max\{f(x_r), f(\widehat{x})\}$$

y

$$f(\widehat{x}) = f\left(\sum_{i=1}^{n} \frac{1}{n} x_i\right) < \sum_{i=1}^{n} \frac{1}{n} f(x_i) \leq f_n \leq f_r.$$

Luego $f(\widehat{x}) < f_r$ y entonces $f_{ce} < f_r$. Por lo tanto la iteración termina aquí con descenso en el valor máximo en los vértices del nuevo simplex.

Caso 2: si $f_r \geq f_{n+1}$ se realiza una contracción interna:

$$x_{ci} = \widehat{x} + \gamma(\widehat{x} - x_{n+1}) = \gamma x_{n+1} + (1-\gamma)\widehat{x}, \qquad 0 < \gamma < 1.$$

Como f es estrictamente convexa

$$f(x_{ci}) < \gamma f_{n+1} + (1-\gamma) f(\widehat{x}) < \max\{f(x_{n+1}), f(\widehat{x})\}$$

y $f(\widehat{x}) < f_n \leq f_{n+1}$ entonces $f(x_{ci}) < f_{n+1}$ y por lo tanto la iteración termina aquí con descenso en el valor máximo en los vértices del nuevo simplex.

Si $\text{card}(A_{\max}) > 1$, solo el Caso 2 puede ocurrir (pues $f_n = f_{n+1}$) y en este caso no habrá descenso en el máximo valor de los valores de f en los vértices del nuevo simplex, pero al eliminar x_{n+1} $\text{card}(A_{\max})$ se reduce en una unidad. ∎

En el caso unidimensional se probó en [5] que el método de Nelder–Mead converge a un minimizador. Observar que el simplex en dimensión 1 consiste de un intervalo cerrado.

Por otro lado, para dimensión 2 y usando $\rho = 1, \delta = 2$ y $\gamma = 1/2$, se demostró en [5] que los valores de la función en los 3 vértices del simplex convergen a un mismo valor. Mas aún, el diámetro de los simplex generados por el algoritmo convergen a cero. Notar sin embargo que este último resultado no afirma que la sucesión de simplex converja a un único punto.

Aunque este resultado parezca débil, Mckinnon [7] presentó contraejemplos de funciones estrictamente convexas donde el método de Nelder–Mead converge a puntos no estacionarios. Hasta el presente no hay ninguna demostración en dimensión 2 (ni en dimensiones mayores) acerca de la convergencia del método a minimizadores, aún para funciones tan simples como el paraboloide $z = x^2 + y^2$.

4 Experimentos numéricos

Veremos ahora algunos experimentos numéricos usando el método de Nelder–Mead en dimensión 2.

Problema 1.

Estudiaremos en detalle el problema de hallar el minimizador de la función $f(x, y) = x^2 - 4x + y^2 - y - xy$. Es fácil probar que f tiene un minimizador en $x^* = (3, 2)$ y que $f(x^*) = -7$.

Comenzamos con el simplex inicial dado por los puntos: $x_1 = (1, 0)$, $x_2 = (0, 0.25)$ y $x_3 = (0, 0)$, cuyos correspondientes valores funcionales son $f_1 = -3, f_2 = 0.025$ y $f_3 = 0$. A partir de este simplex aplicamos el Algoritmo de Nelder–Mead.

En la Tabla 1 mostramos los resultados obtenidos en las primeras 14 iteraciones. En la primera columna indicamos el número de iteración, y en las columnas restantes están cada uno de los tres vértices del simplex y los respectivos valores funcionales. Como se puede ver en esta tabla, la sucesión generada por el método de Nelder–Mead convergerá al minimizador $x^* = (3, 2)$. En la Fig. 4 se observa la sucesión de simplex (triángulos para problemas bidimensionales) generados por el método de Nelder–Mead.

It.	x_1	$f(x_1)$	x_2	$f(x_2)$	x_3	$f(x_3)$
1	(1,0)	-3	(0, 0.5)	-0.25	(0, 0)	0
2	(1.5, 0.75)	-5.0625	(1, 0)	-3	(0, 0.5)	-0.25
3	(1.5, 0.75)	-5.0625	(2.5, 0.25)	-4.5625	(1, 0)	-3
4	(3,1)	-6	(1.5, 0.75)	-5.0625	(2.5, 0.25)	-4.5625
5	(2, 1.5)	-6.25	(3, 1)	-6	(1.5, 0.75)	-5.0625
6	(3.5, 1.75)	-6.5625	(2, 1.5)	-6.25	(3, 1)	-6
7	(2.5, 2.25)	-6.5625	(3.5, 1.75)	-6.5625	(2,1.5)	-6.25
8	(2.5, 1.75)	-6.8125	(2.5, 2.25)	-6.5625	(3.5, 1.75)	-6.5625
9	(3, 1.875)	-6.9844	(2.5, 1.75)	-6.8125	(2.5, 2.25)	-6.5625
10	(3, 1.875)	-6.9844	(2.875, 1.5938)	-6.88701	(2.5, 1.75)	-6.8125
11	(3, 1.875)	-6.9844	(2.7188, 1.7422)	-6.9269	(2.875, 1.5938)	-6.88701
12	(3, 1.875)	-6.9844	(2.8438, 2.0234)	-6.9714	(2.7188, 1.7422)	-6.9269
13	(3, 1.875)	-6.9844	(3.125, 2.1562)	-6.9795	(2.8438, 2.0234)	-6.9714
14	(2.9531, 2.0195)	-6.9965	(3, 1.875)	-6.9844	(3.125, 2.1562)	-6.9795

Tabla 1. Sucesión de simplex generados por el método de Nelder–Mead.

Problema 2.

Seleccionamos algunos problemas test de la colección de Schittkowski para programación no lineal [10] (ver Apéndice) y usamos el comando `fminsearch` de MATLAB (version 6.0.0.88). Este comando es una implementación del método de Nelder-Mead. Usamos los siguientes parámetros *default* de MATLAB:

- `TolX` $= 10^{-4}$ (tolerancia en la variable x).
- `TolFun` $= 10^{-4}$ (tolerancia para el valor de la función $f(x)$).

En la Tabla 2 mostramos los resultados obtenidos usando `fminsearch`.

Problema	(x_1^*, y_1^*)	$f(y_1^*, y_1^*)$	Iter.	Eval. de función
201	(5.0000, 6.0000)	1.5825E-09	43	83
202	(5.0000, 4.0000)	3.2293E-09	54	105
205	(2.9999, 0.5000)	5.5253E-10	83	161
206	(1.0000, 1.0000)	8.2638E-10	50	98
207	(1.0000, 1.0000)	8.2638E-10	53	98
208	(1.0000, 1.0000)	8.1777E-10	85	159
209	(1.0000, 1.0000)	1.9415E-10	311	579
211	(1.0000, 1.0000)	2.5263E-10	86	166
213	(1.0000, 1.0000)	1.5602E-35	46	89

Tabla 2. El método de Nelder–Mead en algunos problemas tests.

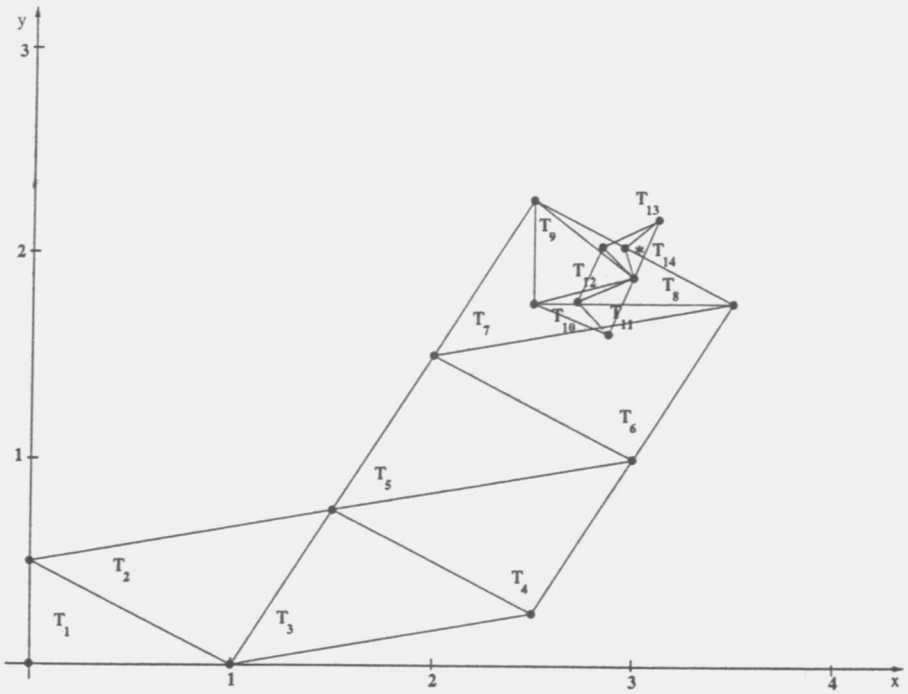

Fig. 4: Sucesión de simplex del método de Nelder–Mead para el Problema 1.

5 Comentarios finales

Como puede apreciarse el método de Nelder–Mead se presenta como un método atractivo para el problema de minimización sin restricciones. Las principales razones por las que se ha hecho tan popular se escriben a continuación: en primer lugar, en varias aplicaciones tales como procesos industriales de control el usuario no desea encontrar el mínimo de alguna función sino está sólo interesado en encontrar valores que mejoren la performance de esa función, esto es, que la función tome valores menores. Esto puede lograrse en pocas iteraciones con este método. En segundo lugar, existen muchas aplicaciones de la "vida real" donde las derivadas de la función a minimizar no están disponibles, porque no pueden ser calculadas o porque son muy "caras" numéricamente de calcular, y como vimos en la descripción del algoritmo no se requieren derivadas de ningún orden. Este método sólo utiliza unas pocas evaluaciones de función por iteración, a diferencia de otros métodos de búsqueda directa o diferencias finitas

que requieren n o más evaluaciones de función por iteración. Por último, la idea del método es simple y fácil de entender así como muy fácil de programar.

Además el método de Nelder–Mead es desafiante tanto desde el punto de vista teórico como práctico. Para obtener resultados en ambas direcciones es fundamental realizar numerosos experimentos numéricos con funciones de diferentes tipos (convexas, no convexas, generales) a fin de lograr una mayor comprensión del método y tratar de mejorar sus puntos débiles, incluyendo nuevas estrategias en el algoritmo. Por ejemplo, recientemente, Kelly [4] propuso una condición de descenso más fuerte que en la versión original la cual evita puntos de "estancamiento" no estacionarios.

Bibliografía

[1] M. Bazaraa, J. Jarvis and H. Sherali (1990), *Linear programming and network flows*, second edition, John Wiley and sons.

[2] D. Bertsekas, (1999), Nonlinear Optimization.

[3] J. E. Dennis and V. Torczon (1991), Direct search methods on parallel machines, *SIAM Journal on Optimization*, Vol. 1, pp. 448–474.

[4] C. T. Kelley (1998), Detection and remediation of stagnation in the Nelder–Mead algorithm using a sufficient decrease condition, *SIAM Journal on Optimization*, Vol. 10, N. 1, pag. 43–55.

[5] J.C. Lagarias, J. A. Reeds, M. H. Wright y P. E. Wright (1998), Convergence properties of the Nelder-Mead simplex method in low dimensions, *SIAM Journal on Optimization*, Vol. 9, pag. 112–147.

[6] Math Works, MATLAB. The Math Works, Natick, MA, 1994.

[7] K. I. M. Mckinnon (1998), Convergence of the Nelder–Mead simplex method to a nonstationary point, *SIAM Journal on Optimization*, Vol. 9, N. 1, pag. 148–158.

[8] A. Nelder and R. Mead (1965), A simplex method for function minimization, *Computer Journal* 7, pp. 308–313.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling and Flannery (1992), Numerical Recipes in Fortran, Cambridge University Press, Cambridge, UK.

[10] K. Schittkowski (1987), More test examples for nonlinear programming codes, *Lecture Notes in Economics and Mathematical Systems*, 282.

[11] W. Spendley, G. R. Hext and F. R. Himsworth (1962), Sequential applications of simplex designs in optimization and evolutionary operation, *Technometrics* 4, pp. 441–461.

[12] V. Torczon, On the convergence of pattern search algorithms (1997), *SIAM Journal on Optimization*, pp. 1–25.

Apéndice

Problemas de minimización sin restricciones de la colección de problemas test de K. Schittkowski [10].

- **Problema 201:** $f(x) = 4(x_1 - 5)^2 + (x_2 - 6)^2$.

  Punto inicial: $(x_0, y_0) = (8, 9)$. Punto solución: $(x^*, y^*) = (5, 6)$.

- **Problema 202:** $f(x) = (-13 + x_1 - 2x_2 + 5x_2^2 - x_2^3)^2$
  $+ (-29 + x_1 - 14x_2 + x_2^2 + x_2^3)^2$.

  Punto inicial: $(x_0, y_0) = (6, 10)$. Punto solución: $(x^*, y^*) = (5, 4)$.

- **Problema 205:** $f(x) = (1.5 - x_1(1 - x_2))^2$
  $+ (2.25 - x_1(1 - x_2^2))^2 + (2.625 - x_1(1 - x_2^3))^2$.

  Punto inicial: $(x_0, y_0) = (0, 0)$. Punto solución: $(x^*, y^*) = (3, 0.5)$.

- **Problema 206:** $f(x) = (x_2 - x_1^2)^2 + 100(1 - x_1)^2$.

  Punto inicial: $(x_0, y_0) = (-1.2, 1)$. Punto solución: $(x^*, y^*) = (1, 1)$.

- **Problema 207:** $f(x) = (x_2 - x_1^2)^2 + (1 - x_1)^2$, (función Banana de Rosenbrock).

  Punto inicial: $(x_0, y_0) = (-1.2, 1)$. Punto solución: $(x^*, y^*) = (1, 1)$.

- **Problema 208:** $f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$.

  Punto inicial: $(x_0, y_0) = (-1.2, 1)$. Punto solución: $(x^*, y^*) = (1, 1)$.

- **Problema 209:** $f(x) = 10^4(x_2 - x_1^2)^2 + (1 - x_1)^2$.

  Punto inicial: $(x_0, y_0) = (-1.2, 1)$. Punto solución: $(x^*, y^*) = (1, 1)$.

- **Problema 211:** $f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$.

  Punto inicial: $(x_0, y_0) = (-1.2, 1)$. Punto solución: $(x^*, y^*) = (1, 1)$.

- **Problema 213:** $f(x) = (10(x_1 - x_2)^2 + (x_1 - 1)^2)^4$.

  Punto inicial: $(x_0, y_0) = (3, 1)$. Punto solución: $(x^*, y^*) = (1, 1)$.

Facultad de Matemática, Astronomía y Física, FaMAF, CIEM
Universidad Nacional de Córdoba
Ciudad Universitaria (5000) Córdoba, Argentina.
(Email: pilotta@mate.uncor.edu)