Estudio numérico de algoritmos determinísticos para problemas de optimización sin derivadas

Johanna A. Frau
FAMAF - Universidad Nacional de Córdoba
CIEM – CONICET
Ciudad Universitaria 5000, Córdoba
Email: jfrau@famaf.unc.edu.ar

Elvio A. Pilotta
FAMAF - Universidad Nacional de Córdoba
CIEM – CONICET
Ciudad Universitaria 5000, Córdoba
Email: pilotta@famaf.unc.edu.ar

***Abstract*—En este trabajo se presenta un estudio numérico de algoritmos para resolver problemas de optimización sin derivadas basados en los dos principales enfoques determinísticos: métodos basados en búsqueda de patrones y métodos basados en interpolación polinomial junto con estrategias de región de confianza. Se analizan algunas de sus implementaciones más conocidas, incluyendo comparaciones usando un conjunto de problemas test.**

I. INTRODUCTION

Consideramos el problema de optimización

$$\text{Minimizar } f(x) \text{ sujeto a } x \in \Omega$$

donde $f : R^n \to R$ es una función dos veces continuamente diferenciable pero sus derivadas no están disponibles simbólicamente ni numéricamente. El conjunto factible Ω puede estar dado por: i) restricciones de cotas en las variables, es decir, $\Omega = \{x \in R^n | l_i \leq x_i \leq u_i, i = 1, \ldots, n\}$, con $-\infty \leq l_i < u_i \leq \infty$, el cual incluye el caso irrestricto y ii) el caso dado por restricciones lineales, es decir, $\Omega = \{x \in R^n | A_{\mathcal{I}}x \geq b_{\mathcal{I}}, A_{\mathcal{E}}x = b_{\mathcal{E}}\}$, correspondientes a las restricciones lineales de desigualdad e igualdad.

Dado que existen muchos problemas de optimización donde las derivadas de la función objetivo y de las funciones que definen las restricciones no están disponibles, ya que provienen de datos experimentales (con o sin ruido) o son muy costosas de calcular computacionalmente, surge la necesidad de disponer métodos de optimización que no requieran derivadas. Algunos ejemplos de tales problemas son: aplicaciones a la geometría molecular para determinar una configuración de un cluster con energía total mínima, eliminación de armónicos dominantes presentes en los sistemas de alimentación de corriente alterna, determinación de la superficie a escala atómica en nanomateriales, etc.

Actualmente se conocen tres principales enfoques de métodos de optimización sin derivadas. La primera clase combina técnicas de diferencias finitas con métodos Quasi-Newton. Estos métodos son relativamente fáciles de implementar utilizando técnicas conocidas para métodos Quasi-Newton, sin embargo dado que las diferencias finitas son usadas para aproximar derivadas, esta clase de métodos no es siempre robusta, especialmente en la presencia de ruido. La segunda clase, requiere realizar minimizaciones sucesivas de modelos, generalmente cuadráticos o lineales, basados en interpolaciones polinomiales utilizando un conjunto de puntos muestrales combinando con estrategias de región de confianza. Entre las desventajas de estos métodos se puede mencionar la complejidad computacional de los algoritmos y la real dificultad de adecuarse a entornos de cálculo paralelo. Por último, se encuentran los métodos de búsqueda directa que usan sólo valores funcionales explorando el espacio de las variables mediante la generación de puntos de prueba a partir de una clase predefinida de patrones geométricos [3], [8]. Algunos de estos métodos no asumen la suavidad de las funciones involucradas y por lo tanto pueden ser aplicados a una amplia clase de problemas. Otra ventaja importante es la simplicidad de estos métodos, lo cual facilita su implementación resultando atractivos para usuarios de optimización de diferentes disciplinas. Además, un rasgo importante es que varios de estos métodos son potencialmente paralelizables, a diferencia de las clases previamente mencionadas.

En este trabajo se realiza un estudio numérico exhaustivo de algunos métodos basados en búsqueda directa e interpolacin polinomial con regin de confianza utilizando un conjunto grande de problemas test de la literatura y considerando diferentes tipos de restricciones.

II. MÉTODOS DE BÚSQUEDA DIRECTA

Estos métodos fueron inicialmente propuestos en la década del 60, sin embargo una década más tarde dejaron de ser interesantes para la comunidad de optimización pues carecían de resultados de convergencia o en ocasiones la convergencia era demasiado lenta. A principios de los 90, los trabajos de V. Torczon dieron un marco teórico con claros resultados de convergencia, lo cual atrajo nuevamente el interés de los investigadores [3], [4], [5], [8].

Se pueden encontrar dos clases particulares de métodos de búsqueda directa globalmente convergentes:

- *métodos de búsqueda de patrones*, los cuales evalúan la función objetivo en un patrón geométrico especificado.
- *métodos de búsqueda lineal*, los cuales utilizan técnicas de métodos basados en gradientes y realizan minimiza-

ciones unidimensionales a largo de direcciones adecuadas.

Particularmente, este trabajo se encuadra en la primera clase. El ejemplo más sencillo de un algoritmo de búsqueda directa para el caso irrestricto está dado por los siguientes pasos:

Algoritmo modelo de búsqueda de patrones para minimización irrestricta
Inicialización:
Sean $f : R^n \to R$ dada y $x_0 \in R^n$ una aproximación inicial. Sean $\Delta_{tol} > 0$, la tolerancia utilizada para probar la convergencia y $\Delta_0 > \Delta_{tol}$, el valor inicial del parámetro de control de longitud de paso.
Para cada iteración $k = 1, 2, \ldots$

Paso 1: Sea $D = \{\pm e_i \,|\, i = 1, \ldots, n\}$ el conjunto de direcciones coordenadas, donde e_i es el i-ésimo vector canónico unitario en R^n.
Paso 2: Si existe $d_k \in D$ tal que $f(x_k + \Delta_k d_k) < f(x_k)$, hacer

$x_{k+1} = x_k + \Delta_k d_k$ (cambiar la aproximación),
$\Delta_{k+1} = \Delta_k$ (mantener el parámetro de control de longitud de paso),
en caso contrario,
$x_{k+1} = x_k$ (mantener la aproximación),
$\Delta_{k+1} = \frac{1}{2}\Delta_k$ (reducir el parámetro de control de longitud de paso),
si $\Delta_{k+1} < \Delta_{tol}$, terminar.

Los principales resultados de convergencia para este algoritmo son:

Teorema 1: *Asumiendo que* $L(x_0) = \{x \in \Omega : f(x) \leq f(x_0)\}$ *es compacto y que* f *es continuamente diferenciable en un conjunto abierto que contiene a* $L(x_0)$*, entonces para la sucesión* $\{x_k\}$ *generada por el algoritmo anterior se tiene que:*

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Para obtener un resultado más fuerte se requieren las siguientes definiciones.

Se define la *matriz patrón* $P_k = BC_k$, donde B y C_k son llamadas *matriz base* y *matriz generadora* respectivamente. La matriz base $B \in R^{n\times n}$ puede ser cualquier matriz no singular, por ejemplo la identidad, mientras que la matriz generadora $C_k \in Z^{n\times p}$ con $p > 2n$ está definida por

$$C_k = [\Gamma_k \quad L_k],$$

donde a su vez Γ_k está formada por

$$\Gamma_k = [M_k \quad -M_k].$$

El método de búsqueda de patrones requiere que $M_k \in M \subset Z^{n\times n}$, donde M es un conjunto finito de matrices no singulares cuyas columnas están asociadas a las direcciones de búsqueda, y que $L_k \in Z^{n\times(p-2n)}$ contenga una (y sólo una) columna con coeficientes todos nulos. Notar que la matriz B no depende del índice k, mientras que C_k cambia en cada iteración.

Además, se requiere definir la siguiente *hipótesis fuerte para movimientos exploratorios*, la cual se utiliza para producir un paso s_k. Estos movimientos pueden ser vistos como un muestreo de la función alrededor del iterado actual x_k.

Hipótesis fuerte para movimientos exploratorios:

a) $s_k \in \Delta_k P_k \equiv \Delta_k BC_k \equiv \Delta_k[B\Gamma_k \quad BL_k]$. [1]
b) Si $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k)$, entonces $f(x_k + s_k) \leq \min\{f(x_k + y), y \in \Delta_k B\Gamma_k\}$.

Esto significa que la elección de los movimientos exploratorios debe asegurar lo siguiente:

1. la dirección de cualquier paso aceptado s_k en la iteración k está definido por el patrón P_k y su longitud está determinada por Δ_k.
2. si un descenso simple en el valor de la función puede ser encontrado entre cualquiera de los $2n$ pasos pruebas definidos por $\Delta_k B\Gamma_k$, entonces los movimientos exploratorios deben producir un paso s_k que también produzca el descenso requerido en b).

De esta manera, es posible obtener el siguiente resultado:
Teorema 2: *Asumiendo que* $L(x_0)$ *es compacto, que* f *es continuamente diferenciable en un conjunto abierto que contiene a* $L(x_0)$*, que las columnas de las matrices generadoras están acotadas en norma, que* $\lim_{k\to+\infty} \Delta_k = 0$ *y que el método de búsqueda de patrones generalizado verifica la hipótesis fuerte en los movimientos exploratorios, entonces para la sucesión* $\{x_k\}$ *generada por este método se tiene que:*

$$\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0.$$

A. Algoritmos e implementaciones

Los aspectos computacionales son una parte esencial de este trabajo, lo cual permite a los autores tener un amplio conocimiento de detalles de implementación de códigos reconocidos para, a partir de aquí desarrollar nuevos códigos y/o variantes. La parte central de este trabajo consiste en el análisis de las performances de códigos bien establecidos: uno de ellos basados en búsqueda de patrones denominado HOPSPACK (Hybrid Optimization Parallel Search Package) [6] y un método basado en interpolación polinomial y región de confianza desarrollado por M. Powell (denominado LINCOA [7]. El primero de ellos resuelve problemas sin restricciones, con restricciones de cotas en las variables, con restricciones lineales y no lineales, el cual es una versión mejorada de su antecesor APPSPACK. El segundo resuelve problemas de optimización irrestrictos, con restricciones de cotas en las variables y con restricciones lineales de desigualdad. Para

[1] La notación $s_k \in \Delta_k P_k$ significa que s_k es alguna columna de la matriz $\Delta_k P_k$.

realizar esta experimentación numérica se utiliza un conjunto amplio de problemas test extraídos de [2].

III. Conclusiones

Se estudian y analizan dos métodos determinísticos para problemas de optimización donde las derivadas no están disponibles. Así como sus metodologías de trabajo son distintas, el comportamiento de ambos difiere al momento de minimizar funciones suaves y con ruido. También se analiza el comportamiento de ambos métodos cuando se utilizan puntos iniciales factibles o infactibles.

References

[1] G. Gray, and T. Kolda, *Algorithm 856: APPSPACK 4.0: Asynchronous Parallel Pattern Search for Derivative-Free Optimization*, ACM Transaction on Mathematical Software, 32, pp.485-507, 2006.

[2] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, v. 187, 1981.

[3] T. Kolda, R. Lewis and V. Torczon, *Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods*, SIAM Review, 45, pp.385-482, 2003.

[4] R. Lewis and V. Torczon, *Pattern Search Algorithms For Bound Constrained Minimization*, SIAM J. OPTIM., 9, pp.1082-1099, 1999.

[5] R. Lewis and V. Torczon, *Pattern Search Methods For Linearly Constrained Minimization*, SIAM J. OPTIM., 10, pp.917-941, 2000.

[6] T. Plantenga, *HOPSPACK 2.0 User Manual (v. 2.0.2)*,2009.

[7] M. Powell, *On fast trust region methods for quadratic models with linear constraints Manual for Fortran Software Package*, DAMTP 2014/NA02, Cambridge, England, 2014.

[8] V. Torczon, *On the Convergence of Pattern Search Algorithms*, SIAM J. OPTIM., 7, pp.1-25, 1997.