Optimización numérica

Programa del curso

Otoño, 2015

1. Introducción

El proceso de minimizar/maximizar una función sujeta a restricciones aparece en un gran número de disciplinas y contextos. Los siguientes son algunos casos bastante difundidos: a) control de procesos; b) finanzas computacionales; c) estimación de parámetros de modelos complejos; d) estadística; e) reconstrucción de imágenes; f) logística; g) diseño óptimo. En la gran mayoría de estos casos el problema original requiere de la resolución de uno a varios subproblemas que se pueden formular como:

$$\begin{array}{lll} \text{minimizar} & f(x), & f : \mathbb{R}^n \to \mathbb{R}, \\ \text{sujeta a} & c(x) \geq 0, & c : \mathbb{R}^n \to \mathbb{R}^m \end{array} \tag{1}$$

en donde f y c son funciones 2 veces continuamente diferenciables. La función f es llamada el *objetivo*, mientras que las componentes de la función c representan a las *restricciones*. Finalmente el vector x representa al conjunto de variables de decisión.

En general, debido a las condiciones impuestas por el problema original, los métodos que se usan con mayor frecuencia sólo obtienen una aproximación a una *solución local* de (1).

2. Objetivos del curso

El curso está dedicado al estudio y aplicación de métodos de optimización diferenciable. La finalidad de estos métodos es obtener aproximaciones a una solución local de f mediante el uso de una cantidad de recursos computacionales (memoria y tiempo de procesador) razonable. Los objetivos específicos del curso son los siguientes:

- Estudiar las características de problemas reales que se pueden formular como (1).
- Caracterizar matemáticamente el problema de optimización (1). Establecer y probar las condiciones de optimalidad de (1).
- Utilizar las condiciones de optimalidad para diseñar algoritmos y sus realizaciones computacionales para resolver eficientemente problemas de la forma (1).
- Interpretar y analizar los resultados obtenidos por un algoritmo. Distinguir: a) limitaciones del método; b) limitaciones debidas a la realización computacional.

3. Prerrequisitos

Es indispensable que el alumno haya cursado las siguientes materias:

- Cálculo 3
- Algorítmica y programación
- Algebra lineal
- Cálculo numérico
- Programación lineal
- Análisis aplicado.

4. Contenido del curso

1. Introducción. Optimizar es importante. Ejemplos. Diferentes tipos de optimización: a) discreta vs. continua; b) global vs. local; c) optimización diferenciable vs. optimización no diferenciable.
2. Optimalidad 1. Caracterización de una solución. Mínimos locales. Condiciones necesarias de primer orden. Multiplicadores de Lagrange. Significado y aplicación práctica al diseño de algoritmos.
3. Optimalidad 2. Direcciones limitantes. Condiciones necesarias y suficientes de segundo orden. La Hessiana proyectada. Problemas convexos. Significado y aplicación práctica al diseño de algoritmos.
4. Aspectos algorítmicos. Propiedades deseables de un algoritmo. Optimalidad vs. factibilidad: funciones de mérito.
5. Programación cuadrática. Problemas de igualdad. Métodos basados en: a) el rango; b) el espacio nulo; c) direcciones conjugadas. Problemas de desigualdad. Conjuntos activos.
6. Programación cuadrática sucesiva. Algoritmos locales. Funciones de mérito. Algoritmos globales.
7. Métodos de puntos interiores. Barrera logarítmica.

5. Evaluación del curso

El curso se evaluará de la siguiente manera:

2 exámenes parciales	50 %
proyectos	30 %
examen final	20 %

Los exámenes parciales se realizarán en las horas de clase en las fechas siguientes:

1er examen	martes 29 de septiembre
2o examen	martes 17 de noviembre

Para aprobar el curso es necesario: a) tener promedio aprobatorio en los exámenes parciales; b) aprobar el examen final; c) aprobar el laboratorio. Los estudiantes realizarán *todas* las tareas y/o proyectos en forma *individual*.

Referencias

[1] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer Verlag, New York, second ed., 2006.

[2] D G Luenberger, *Linear and Nonlinear Programming*, 2nd Edition, Addisson-Wesley, Reading, Massachusetts, 1984.

[3] R Fourer, D M Gay, and B W Kernighan, *AMPL, A Modeling Language for Mathematical Programming*, boyd & fraser publishing company, Belmont, 1993.