Problemas de Optimización: una Introducción

Programación Matemática a Gran Escala
Ing. Fabio A. González, PhD
Departamento de Ing de Sistemas e Industrial
Universidad Nacional de Colombia
Resolución de Problemas


• Cuatro pasos para resolver un problema:
  – Understanding the problem
  – Devising a plan
  – Carrying out the plan
  – Looking back.
Understanding The Problem

• **First.** You have to *understand* the problem.
• What is the unknown? What are the data? What is the condition?
• Is it possible to satisfy the condition? Is the condition sufficient to determine the unknown? Or is it insufficient? Or redundant? Or contradictory?
• Draw a figure. Introduce suitable notation.
• Separate the various parts of the condition. Can you write them down?
Devising A Plan (1)

• **Second.** Find the connection between the data and the unknown. You may be obliged to consider auxiliary problems if an immediate connection cannot be found. You should obtain eventually a *plan* of the solution.

• Have you seen it before? Or have you seen the same problem in a slightly different form?

• *Do you know a related problem?* Do you know a theorem that could be useful?

• *Look at the unknown!* And try to think of a familiar problem having the same or a similar unknown.
Devising A Plan (2)

• *Here is a problem related to yours and solved before. Could you use it?* Could you use its result? Could you use its method? Should you introduce some auxiliary element in order to make its use possible?

• Could you restate the problem? Could you restate it still differently? Go back to definitions. If you cannot solve the proposed problem try to solve first some related problem.

• Did you use the whole condition? Have you taken into account all essential notions involved in the problem?
Carrying Out The Plan

• **Third.** *Carry out* your plan.
• Carrying out your plan of the solution, *check each step*. Can you see clearly that the step is correct? Can you prove that it is correct?
Looking Back

• **Fourth.** *Examine* the solution obtained.
• Can you *check the result*? Can you check the argument?
• Can you derive the solution differently? Can you see it at a glance?
• Can you use the result, or the method, for some other problem?
Por qué algunos problemas son difíciles de resolver?

- Tamaño del espacio de búsqueda
- Modelado
- Ruido y variabilidad
- Restricciones
- Estrategia adecuada
Problema de Búsqueda

• Dado un conjunto $S$ y un predicado $P(s)$, encontrar $s_0 \in S$ tal que $P(s_0)$.

• Ejemplos:
  – Problema de satisfactibilidad (SAT)
  – Problemas de planeación
  – Camino hamiltoniano
Problemas de Optimización

• Problema de optimización general:
  Dado $f: S \rightarrow \mathbb{R}$, encontrar $x \in S$ tal que $G(x)$ y $y \in S, G(y) \Rightarrow f(x) \leq f(y)$

• Ejemplo:
  – Problema general de programación no lineal:
    minimizar $f(x)$
    sujeto a : $g_i(x) \geq 0 \quad i = 1,\ldots,m$
    $h_j(x) = 0 \quad j = 1,\ldots,p$
Optimo Global y Local

• $x^*$ es un óptimo global si:
  - $\exists y \in G(S), \ f(x^*) \leq f(y)$

  ($G(S)$ corresponde a la región factible y se define como:
  $G(S) = \{ x \in S : G(x) = 1 \}$)

• $x^*$ es un óptimo local si:
  - Existe una vecindad $N(x^*)$ tal que $\exists y \in N(x^*)$, $f(x^*) \leq f(y)$
Planteando un Problema de Optimización (búsqueda)

- Escoger una buena representación
  - $S$: espacio de búsqueda
- Definir el objetivo
  - $f: S \rightarrow \mathbb{R}$
- Definir las restricciones
  - $G: S \rightarrow \{0,1\}$
- Definir una función de evaluación
  - $f': S \rightarrow \mathbb{R}$
- Definir una función de búsqueda local (vecindad)
  - $N: S \rightarrow 2^S$
Clasificación de Problemas de Optimización

• Optimización continua:
  El espacio de búsqueda corresponde a $\mathbb{R}^n$

• Optimización discreta (combinatoria):
  El espacio de búsqueda corresponde a un conjunto finito o posiblemente contable infinito.
  Ejemplo: enteros, conjuntos, permutación, grafo, etc.
Optimización Continua

• No restringida:
  – Una variable
  – Varias variables

• Restringida:
  – Programación lineal
  – Programación no lineal
  – Programación cuadrática
  – Programación convexa
Optimización discreta

• Programación entera
• Optimización en grafos:
  – Minimal spanning tree
  – Camino más corto
  – Problema del agente viajero
  – Matching
  – Flujo máximo
• Programación dinámica
• Scheduling