

Branch-and-bound:

Problema: Sea S un conjunto finito (típicamente ENORME) y sea $c: S \longrightarrow \mathbb{R}$ una función. Queremos resolver el problema de **OPTIMIZACIÓN COMBINATORIA**

$$(P) \quad z_p = \text{maximizar } \{c(s) : s \in S\}$$

Branch and bound es un método para resolver problemas de este tipo de manera iterativa.

Típicamente requiere tres ingredientes:

(1) Cotas Superiores eficientes
(upper bound)
 $UB(P) \geq z_p$, $UB(P)$

(2) Reglas de branching ó expansión
que reemplazan (P) por un conjunto de subproblemas P_1, P_2, \dots, P_k
de tal manera que alguna solución óptima de P sea solución óptima de algún P_i .

$$\text{branch}(P) = \{P_1, \dots, P_k\}$$

(3) Cotas inferiores. Típicamente surgen de heurísticas eficientes que intentan construir algún punto factible $s' \in S$
(luego $c(s') \leq z_p$.)

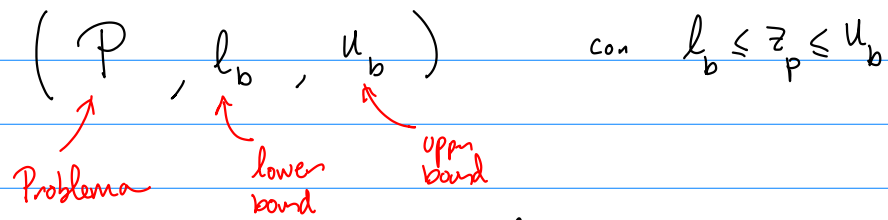
$$LB(P)$$

Queremos explorar un árbol de posibilidades e intentar evitar explorar ramas que sepamos no pueden contener el óptimo

Para problemas de maximización

Método: Mantendremos una lista L de triplas

$$(P, l_b, u_b) \quad \text{con } l_b \leq z_p \leq u_b$$



y un lower bound l_b^* que se actualizará durante la ejecución.

Inicialización:

$$L = \{ (P, LB(P), UB(P)) \}$$

$$l_b^* = LB(P)$$

En cada paso:

while not $L.empty()$:

$Q = L.pop()$

if $UB(Q) \leq l_b^*$:

descartamos Q (by bounds)

else

new-subproblems = branch(Q)

for N in new-subproblems:

if $UB(N) \leq l_b^*$ or N infeasible:

descartar N

else

$$L \leftarrow L \cup \{ (N, LB(N), UB(N)) \}$$

$$l_b^* = \max(l_b^*, LB(N))$$

Obs: Una mejora posible es mantener también un upper bound global de los problemas activos

$$UB := \max \{ l_b^*, \max \{ UB(P) : P \in L \} \}$$

Esto cumple $l_b^* \leq z_p \leq UB$

en todos los insts y permite ESTIMAR EL ERROR en cualquier momento pues $z_p - l_b^* \leq UB - l_b^*$ conocidas

Ejemplo Muy Importante: Programación entera

$$\begin{aligned} Z_p = \max \quad & -x + y \\ \text{s.a.} \quad & 12x + 11y \leq 63 \\ & -22x + 4y \leq -33 \\ & x, y \geq 0 \\ & x, y \in \mathbb{Z} \quad (\text{enteros}) \end{aligned}$$

Puede resolverse mediante branch and bound y optimización lineal como veremos...