Ottimizzazione dei Sistemi Complessi

A.A. 2018-19 – 14 Maggio 2020

prova d'esame

1. (**11 punti**) Dato il problema di PNL

$$\begin{aligned} &\max\ x+y \\ &s.t.\ x+(y-1)^3 \le 0 \\ &\qquad x \ge 0,\ y \ge 0 \end{aligned}$$

   (a) (**3 punti**) Rappresentare sul piano cartesiano la regione ammissibile del problema e stabilire se la medesima è convessa.

   (b) (**4 punti**) Motivare l'esistenza di una soluzione globale e determinarla per via grafica.

   (c) (**4 punti**) Determinare analiticamente tutti i punti di F.J. del problema.

2. (**11 punti**) Si consideri il seguente problema multiobiettivo:

$$\begin{aligned} &\min\ x^2+(y-2)^2,\ x^2+(y+1)^2 \\ &s.t.\ x \ge 0,\ y \ge 0,\ x+y \le 1 \end{aligned} \tag{MOP}$$

   (a) (**4 punti**) scrive le condizioni di ottimalità di KKT;

   (b) (**4 punti**) nel punto $(1,\ 0)^\top$ determinare una direzione che sia di discesa per entrambe le funzioni obiettivo;

   (c) (**3 punti**) determinare almeno un punto di KKT del problema (con i rispettivi moltiplicatori di funzioni obiettivo e vincoli).

3. (**11 punti**) Si consideri il seguente problema non vincolato

$$\min\ (x-3)^2+(y-1)^2$$

   (a) (**3 punti**) Dati il punto $x_0 = (0,0)^\top$ e la direzione $d = (1,1)^\top$, scrivere l'espressione della funzione obiettivo ristretta alla semiretta $x_0 + \alpha d$, $\phi(\alpha)$

   (b) (**4 punti**) Determinare il più piccolo intero nonnegativo $j \in \{0,1,2,3,\dots\}$ tale che

$$\begin{aligned} &\phi(2^j) \le \phi(0) - 0.1(2^j)^2 \\ &\phi(2^{j+1}) > \phi(0) - 0.1(2^{j+1})^2 \end{aligned}$$

   (c) (**4 punti**) A partire dal punto $x_0 = (0,\ 0)^\top$ e con passo iniziale $\Delta_0 = 1$ eseguire una iterazione del metodo compass search "forte" determinando il nuovo punto x_1 ed il nuovo valore del passo Δ_1 al termine dell'iterazione.

```
INPUT: x_0, Δ_0, Δ_min, maxit, D = {±e_i, i = 1,...,n}

k ← 0, x ← x_0, Δ ← Δ_0

while k ≤maxit and Δ ≥ Δ_min do

    k ← k + 1
    Let d̄ ∈ D be s.t. f(x + Δd̄) = min_{d_i∈D} f(x + Δd_i)

    if f(x + Δd̄) < f(x) then
       x ← x + Δd̄
    else
       Δ ← Δ/2
    endif

end while

RETURN: x (miglior punto determinato)
```

Figure 1: Pseudo-codice del metodo compass search "forte"