



Programmazione Matematica: Introduzione

Alessandro Hill

Basato sul materiale di

Daniele Vigo (D.E.I.) & Marco Boschetti (D.M.).

rev. 1.1(AH) – 2024



Preliminari

Notazione

- \mathbb{R} : insieme dei numeri reali (\mathbb{R}^n : spazio vettoriale a n dimensioni)
- \mathbb{Z} : insieme dei numeri interi (\mathbb{Z}^+ : numeri interi positivi)
- $[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$ (intervallo chiuso)
- $(a, b) = \{x \in \mathbb{R} : a < x < b\}$ (intervallo aperto)
- $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$: norma euclidea; $|z|$: valore assoluto dello scalare $z \in \mathbb{R}$
norma 2 (sommatoria delle componenti di un vettore, con componenti alla seconda)
- $Q = \{q_1, \dots, q_n\}$: insieme degli n elementi q_1, \dots, q_n
- $Q = \{x \in \mathbb{R}^n : P(x)\}$: insieme dei punti di \mathbb{R}^n che soddisfano le condizioni P
- $|Q|$: cardinalità dell'insieme Q
- $\operatorname{argmin}\{f(i) : i \in I\}$: $i^* \in I$ tale che $f(i^*) = \min\{f(i) : i \in I\}$
- $\lfloor z \rfloor = \max\{i \in \mathbb{Z} : i \leq z\}$; $\lceil z \rceil = \min\{i \in \mathbb{Z} : i \geq z\}$
Floor: restituisce il più grande intero minore o uguale a z (cioè più vicino a z).
Ceil (o ceiling): restituisce il più piccolo intero maggiore o uguale a z (cioè più vicino a z).

Notazione

▷ $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$: **vettore colonna n dimensionale** ($x \in \mathbb{R}^n$)

▷ $\mathbf{c}^T = [c_1, \dots, c_n]$: **vettore riga n-dimensionale**

▷ $\mathbf{c}^T \mathbf{x} = \sum_{j=1}^n c_j x_j$: **prodotto scalare** (o anche $c\mathbf{x}$) (vettore riga per vettore colonna)

▷ $\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$: **matrice m × n**

▷ $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix} = \begin{bmatrix} a^1 \\ \vdots \\ a^m \end{bmatrix} = [A_1, \dots, A_n] = [a_1, \dots, a_n]$

riga matrice per vettore colonna

▷ $\mathbf{Ax} = \begin{bmatrix} \sum_{j=1}^n a_{1j} x_j \\ \vdots \\ \sum_{j=1}^n a_{mj} x_j \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^T \mathbf{x} \\ \vdots \\ \mathbf{a}_m^T \mathbf{x} \end{bmatrix} = \begin{bmatrix} a^1 \mathbf{x} \\ \vdots \\ a^m \mathbf{x} \end{bmatrix}$

riga matrice per vettore colonna = b_i

▷ $\mathbf{Ax} = \mathbf{b} \rightarrow \begin{cases} \mathbf{a}_1^T \mathbf{x} = b_1 \\ \vdots \\ \mathbf{a}_m^T \mathbf{x} = b_m \end{cases} \equiv \begin{cases} a^1 \mathbf{x} = b_1 \\ \vdots \\ a^m \mathbf{x} = b_m \end{cases}$

▷ **rango(A): rango di A** \rightarrow Il rango di una matrice è il numero massimo di righe (o colonne) linearmente indipendenti.

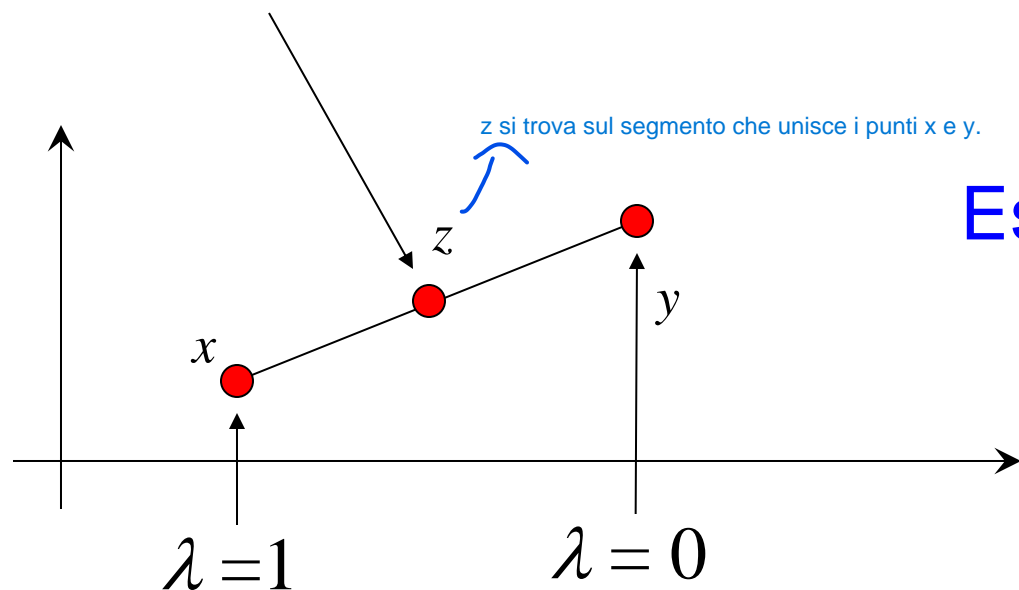
▷ **det(A): determinante di A**

▷ \mathbf{A}^{-1} : **matrice inversa di A**

Combinazione Convessa

Def.: z , è **combinazione convessa** di x, y se

$\exists \lambda \in [0, 1]$ tale che $z = \lambda x + (1 - \lambda) y$



Es. $x, y \in \mathbb{R}^2$

Combinazione Convessa (2)

Def.: Combinazione convessa di K punti

$$p_1, p_2, \dots, p_K \in \mathbb{R}^n$$

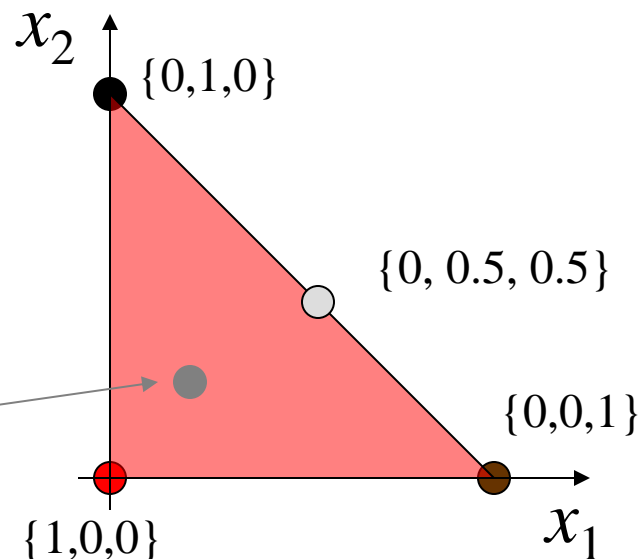
$$z = \sum_{i=1, K} \lambda_i p_i \quad \text{con } \lambda_i \geq 0 \quad \text{e} \quad \sum_{i=1, K} \lambda_i = 1$$

Es. $z = \lambda x + (1 - \lambda) y, \quad \lambda_1 = \lambda > 0, \lambda_2 = 1 - \lambda, \lambda_1 + \lambda_2 = 1$

In questo caso, z appartiene all'involuppo convesso dei vettori x_1, \dots, x_n ; cioè alla regione delimitata dai punti x_1, \dots, x_n .

$$p_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad p_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad p_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\lambda_i = \{0.5, 0.2, 0.3\} \quad z = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix}$$



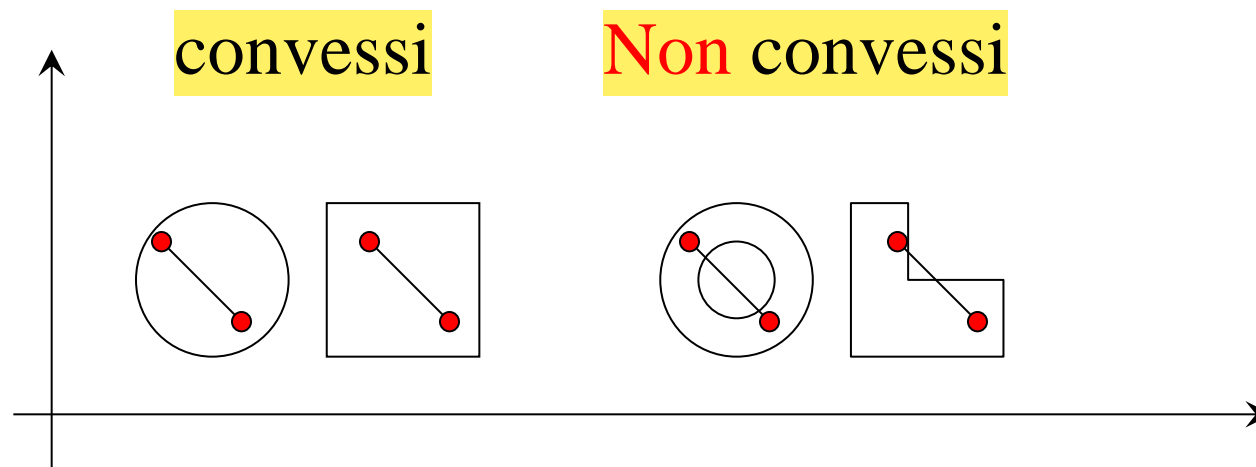
Insiemi Convessi

Def.: $F \subseteq \mathbb{R}^n$ è un **insieme convesso** se

$$\forall x, y \in F \text{ e } \forall \lambda \in [0,1],$$

$$z = \lambda x + (1 - \lambda) y \in F$$

Es. $x, y \in \mathbb{R}^2$



Proprietà degli Insiemi Convessi

Proprietà 0: \mathbb{R}^n è convesso (ovvio)

Proprietà 1: Dati F_i convessi \Rightarrow
 $F = \cap F_i$ è **convesso**

l'intersezione di insiemi convessi è anch'essa un insieme convesso.

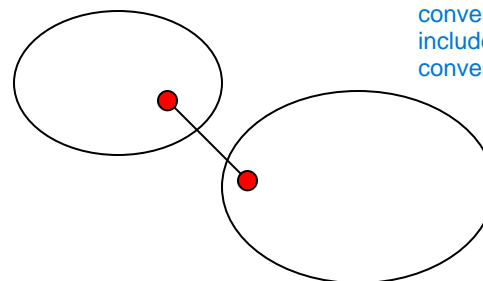
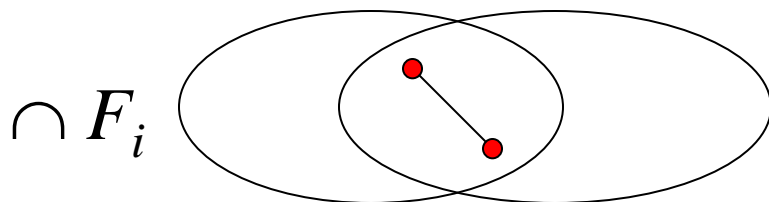
DIM.:

$$x, y \in F \Rightarrow x, y \in F_i \forall i$$

Poiché ogni F_i è convesso, z appartiene a ciascun F_i

$$\Rightarrow z = \lambda x + (1 - \lambda) y \in F_i \forall i, \forall \lambda$$

$$\Rightarrow z \in \cap F_i \quad \bullet \text{ Pertanto, intersezione di } F_i \text{ è convesso}$$

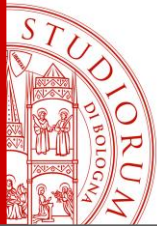


L'unione di insiemi convessi in generale non è convessa. Questo accade perché l'unione può includere punti tra i quali una combinazione convessa non appartiene all'unione stessa.

$$\cup F_i$$


$$\varphi(\lambda x + (1 - \lambda)y) \leq \lambda \varphi(x) + (1 - \lambda)\varphi(y)$$

punto sul segmento tra x e y



Problemi di ottimizzazione

Problemi di Ottimizzazione

- $x = (x_1, \dots, x_n) \in \mathbb{R}^n$: vettore di **variabili decisionali**
 - prodotti da realizzare, istanti in cui produrli ...
 - merci o materie prime da stoccare: quanto, quando ...
 - luogo in cui realizzare una infrastruttura ...
 - tratti di strada da scegliere in un percorso ...
- $F \subseteq \mathbb{R}^n$: insieme delle **soluzioni ammissibili**
(regione ammissibile)
- $\varphi : F \rightarrow \mathbb{R}$: **funzione obiettivo** (f. costo)

$$(P) \quad \min_{x \in F} \varphi(x)$$

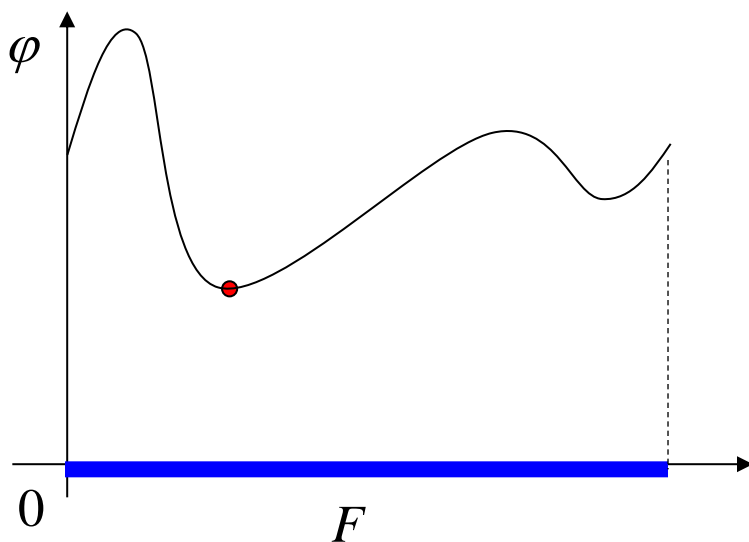


Problemi di Ottimizzazione (2)

$$(P) \quad \min_{x \in F} \varphi(x)$$

ovvero determinare $x^* \in F$ (ottimo globale) tale che:

$$\varphi(x^*) \leq \varphi(x) \quad \forall x \in F$$



In generale φ ed F
sono qualsiasi

storicamente detti
problemi di
programmazione

Regione Ammissibile

La regione ammissibile F può essere definita:

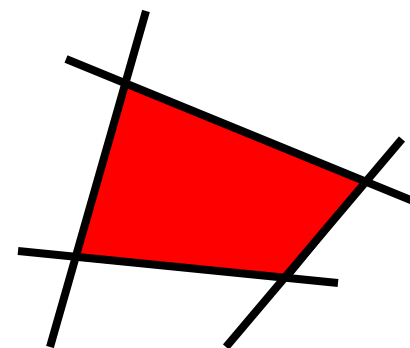
- **esplicitamente**: specificando le proprietà di $x \in F$

Es. $[0,1]^2$; x intere nell'ipercubo di lato 1

- **implicitamente**: equazioni e disequazioni

Es. $F := \{x \in \mathbb{R}^n : g_i(x) \leq 0, (i=1, \dots, m)$

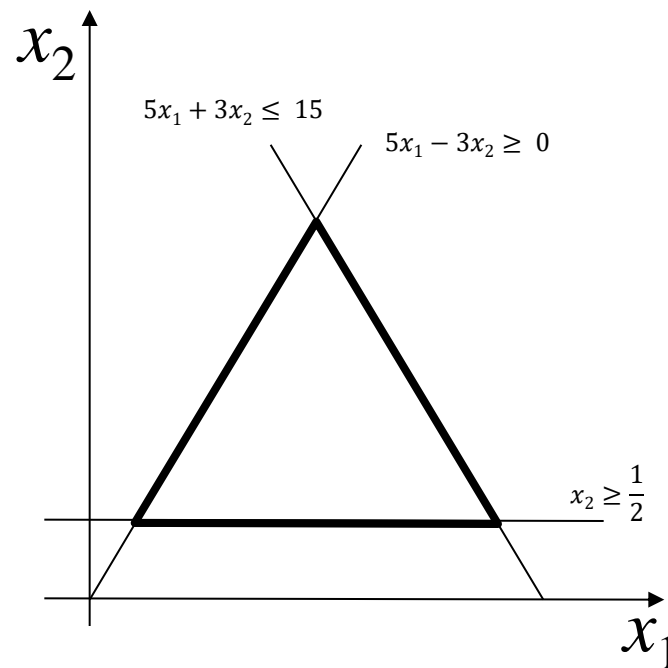
$h_j(x) = 0, (j=1, \dots, p)\}$





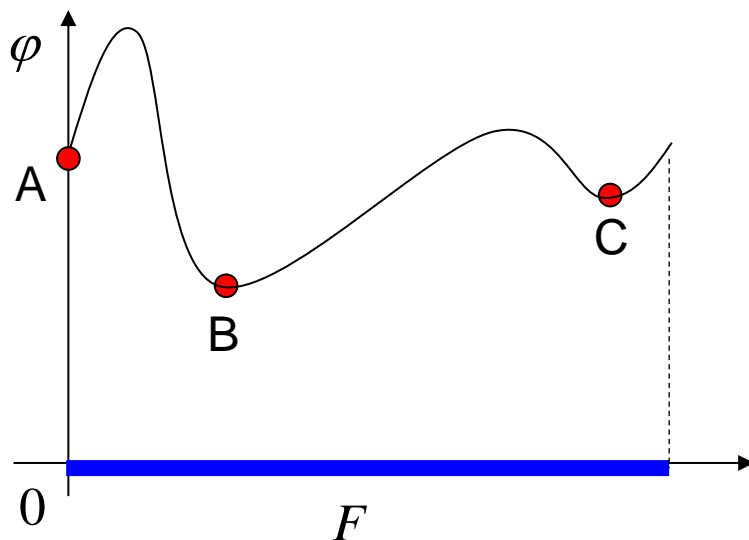
Esempio di regione ammissibile

$$F = \{x \in \mathbb{R}^2 : 5x_1 + 3x_2 \leq 15; \\ 5x_1 - 3x_2 \geq 0; x_2 \geq \frac{1}{2}, x_1, x_2 \geq 0\}$$



Minimi Locali e Globali

- non è detto che x^* esista ($F = \emptyset$) o che sia unica
- possono esistere ottimi (minimi) **locali** e **globali**

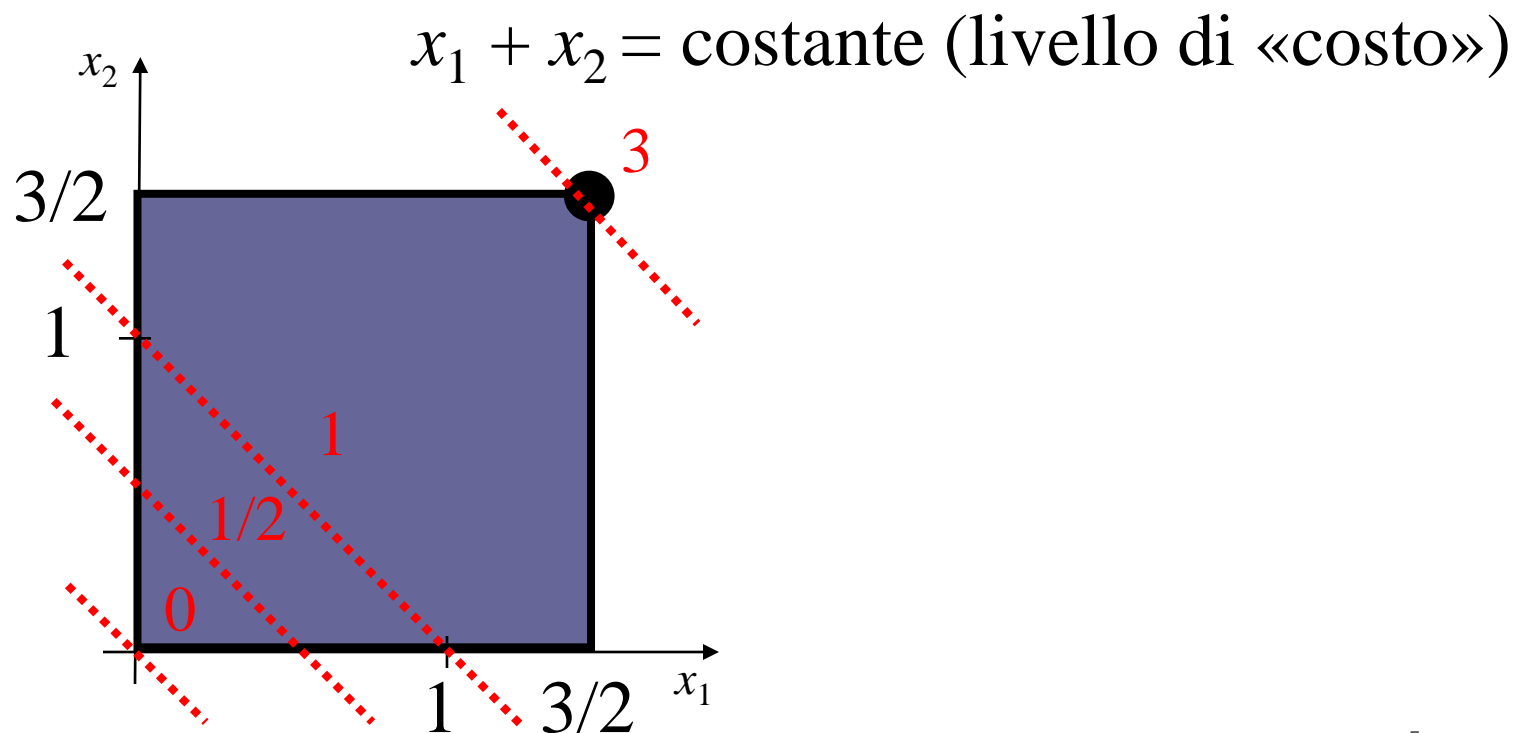


(P) richiede di trovare almeno un ottimo globale



Esempio 1

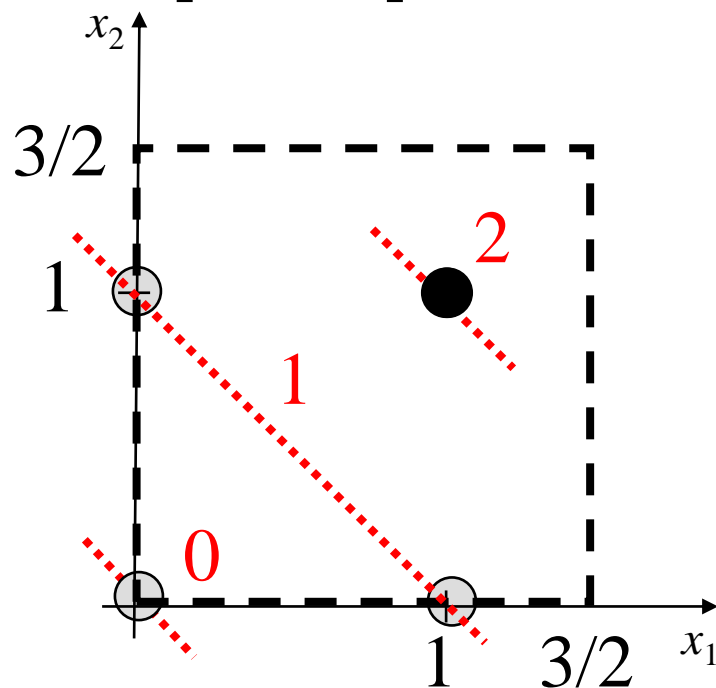
problema continuo: $F = [0, 3/2]^2 \subset \mathbb{R}^2$
 $\max \varphi(x) = x_1 + x_2$



Esempio 2

- problema discreto:

$$F = [0, 3/2]^2 \cap \mathbb{Z}^2$$



$$\max \varphi(x) = x_1 + x_2$$

si può valutare $\varphi(x)$
in ciascun vertice

se $F = [0, 1]^{100} \cap \mathbb{Z}^{100}$

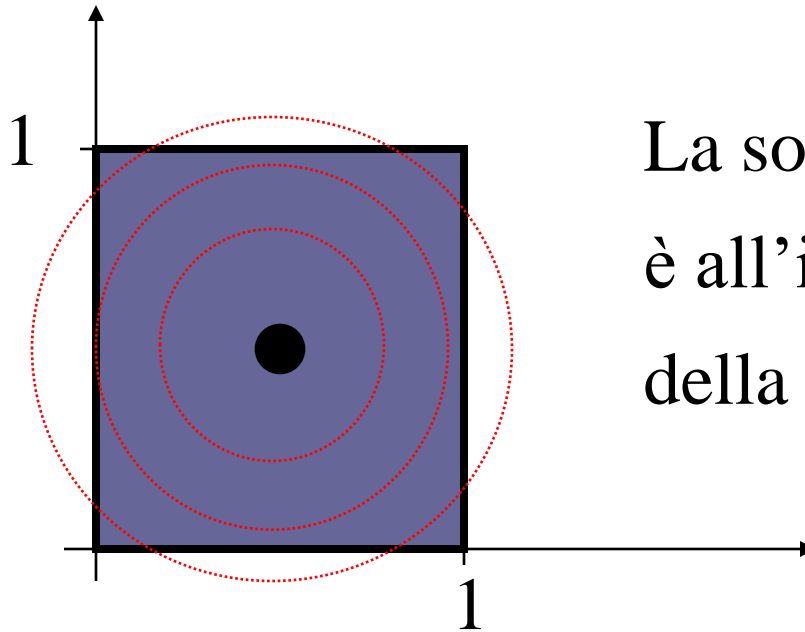
$\Rightarrow 2^{100} \sim 10^{30}$ valutazioni

Esempio 3

- problema continuo:

$$F = [0, 1]^2 \subset \mathbb{R}^2$$

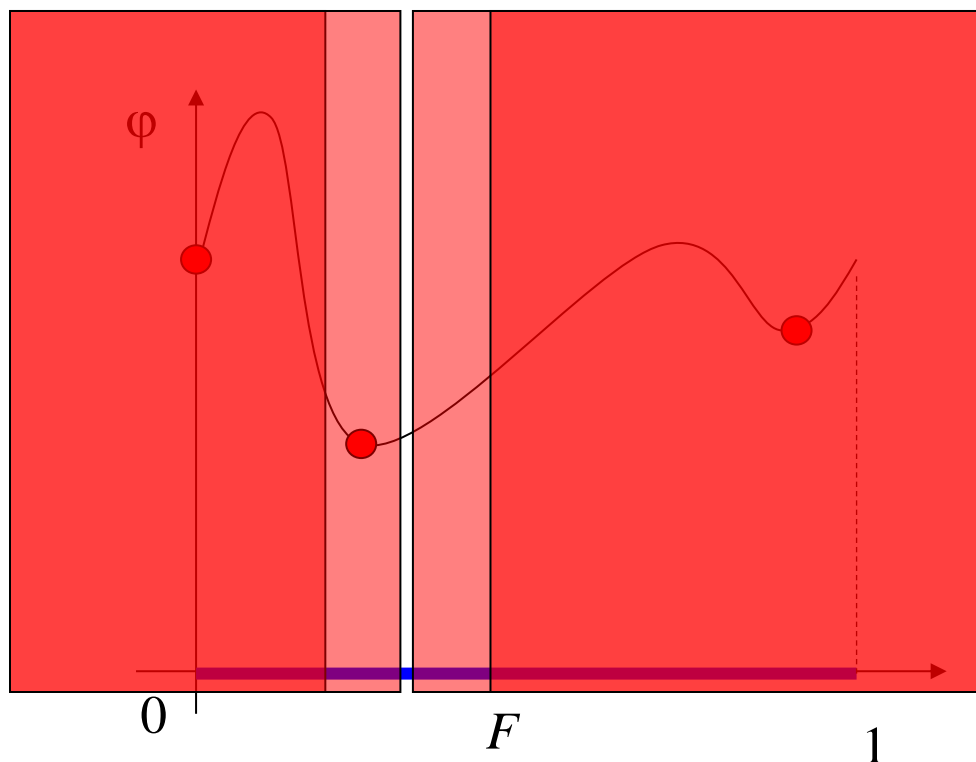
$$\min \varphi(x) = (x_1 - 1/2)^2 + (x_2 - 1/2)^2$$



La soluzione ottima
è all'interno
della regione ammissibile

Algoritmi Numerici

- Un algoritmo non ha visione **completa** di φ ed F
- valuta $\varphi(x)$ in una sequenza di punti $x \in F$





Algoritmi Numerici (2)

Gli algoritmi per i problemi di ottimizzazione sono generalmente di tipo **iterativo**:

1. Sia $x_0 (\in F)$ una soluzione iniziale;

$k := 0$;

2. **repeat**

2.1 verifica l'ottimalità (locale) di x_k

2.2 se x_k non ottima genera $x_{k+1} (\in F)$ tale che

$\varphi(x_{k+1}) \leq \varphi(x_k)$ e poni $k := k + 1$;

until (x_k ottima) o (*condizione di terminazione*)





Algoritmi Numerici (3)

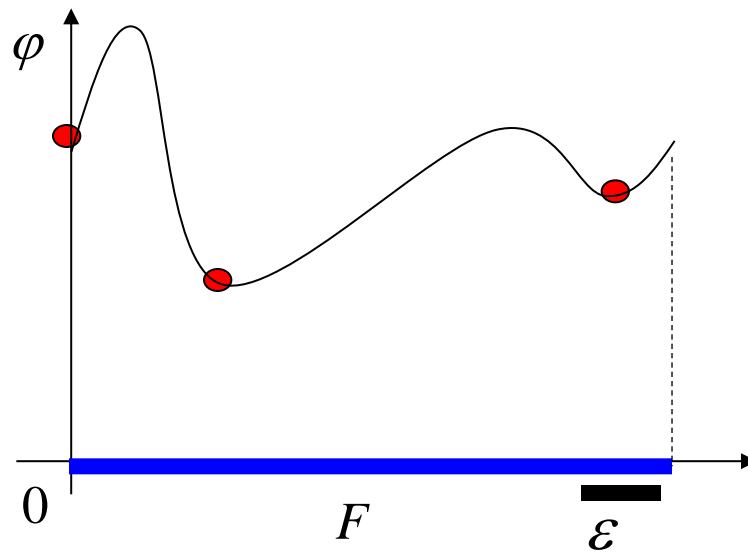
- La sequenza x_0, x_1, \dots, x_k **converge** alla soluzione ottima x^* (o ad un ottimo locale)
- Nel caso generale (φ ed F **qualsiasi**) la convergenza è ad un **ottimo locale** ed il numero di iterazioni è **molto elevato**
- Nel caso continuo si termina quando si è raggiunta l'**approssimazione** desiderata (piccole variazioni tra iterazioni successive)

Ottimi Locali ed Intorni

Def.: $y \in F$ è un **ottimo locale** se \exists un **intorno** $N \subseteq F$

tale che $\varphi(y) \leq \varphi(x) \quad \forall x \in N$

Es. $N_\varepsilon(y) := \{x \in F : \|y - x\| \leq \varepsilon, \varepsilon > 0\}$ (intorno **euclideo**)



N è **esatto** se un
ottimo locale
rispetto ad N è
ottimo globale

(**Es.** N_1)

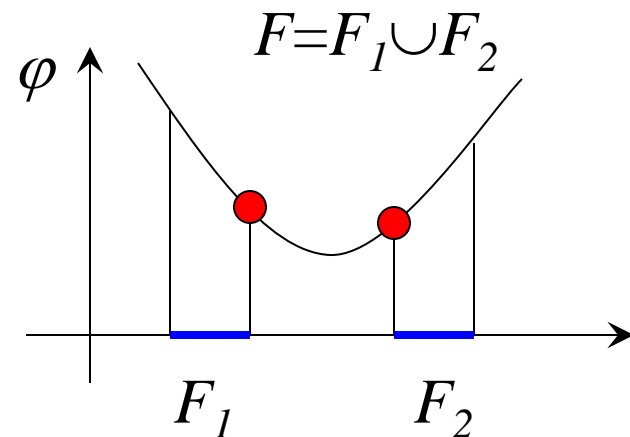
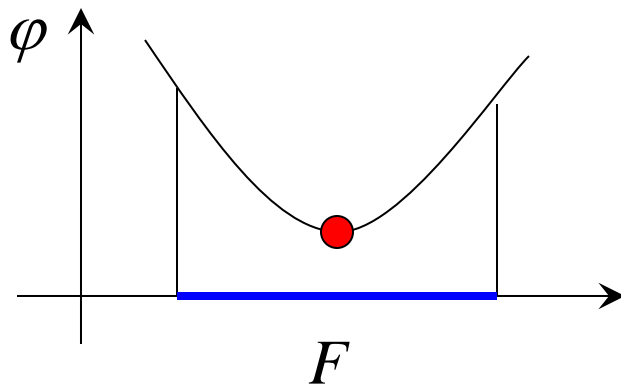
Convessità ed Intorno Euclideo

per le funzioni convesse su insiemi convessi, ottimo locale e ottimo globale coincidono.

Th.: Dato (F, φ) con $F \subseteq \mathbb{R}^n$ **convesso** e φ **convessa**, $N_\varepsilon(x) = \{y \in F : \|x - y\| \leq \varepsilon\}$ è **esatto** $\forall \varepsilon > 0$

intorno euclideo di raggio epsilon centrato in un punto x appartenente ad F.

\Rightarrow (ottimo locale \equiv ottimo globale)



Classificazione

(I vincoli di disuguaglianza $g_i(x) \leq 0$ restringono lo spazio delle soluzioni ammissibili F)

φ, g_i, h_j qualunque

I vincoli di uguaglianza $h_j(x)=0$ restringono ulteriormente lo spazio delle soluzioni ammissibili F .

\Rightarrow Progr. Non Lineare (PNL, NLP)

Non esistono algoritmi generali di ottimizzazione

\exists metodi che convergono a ottimi locali

φ, g_i convesse

h_j lineari

\Rightarrow Programmazione Convessa (PC, CP)

ottimo locale \equiv ottimo globale

\exists algoritmi ma non efficienti



Classificazione (2)

φ, g_i, h_j lineari \Rightarrow Programmazione Lineare (PL, LP)

ottimo locale \equiv ottimo globale

\exists algoritmi efficienti

(Simplexso, Interior Point, ...)

PL con variabili
interi

\Rightarrow Progr. Lineare Intera (PLI, ILP)

Problema difficile (\propto PNL)

\exists algoritmi generali

(Branch-and-Bound, ...)