Programmazione Matematica: Introduzione

Daniele Vigo
D.E.I. – Università di Bologna
daniele.vigo@unibo.it

rev. 1.0 – 2023

Preliminari

Notazione

- R : insieme dei numeri reali (R^n : spazio vettoriale a n dimensioni)
- Z : insieme dei numeri interi (Z+ : numeri interi positivi)
- $[a, b] = \{x \in R : a \leq x \leq b\}$ (intervallo chiuso)
- $(a, b) = \{x \in R : a < x < b\}$ (intervallo aperto)
- $\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$: norma euclidea; $|z|$: valore assoluto dello scalare $z \in R$
- $Q = \{q_1, \ldots q_n\}$: insieme degli n elementi $q_1, \ldots, q_n$
- $Q = \{x \in R^n : P(x)\}$: insieme dei punti di R^n che soddisfano le condizioni P
- $|Q|$: cardinalità dell'insieme Q
- $\text{argmin}\{f(i) : i \in I\}$: $i_* \in I$ tale che $f(i^*) = \min\{f(i) : i \in I\}$
- $\lfloor z \rfloor = \max\{i \in Z : i \leq z\}$; $\lceil z \rceil = \min\{i \in Z : i \geq z\}$

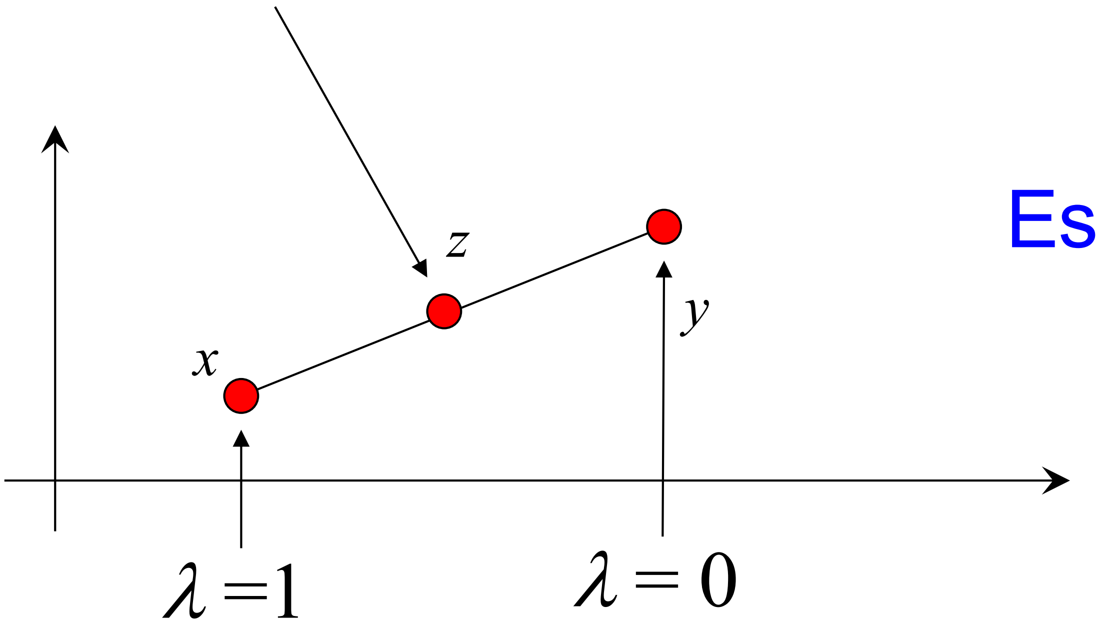
Notazione

- $\triangleright\ x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$: vettore colonna n dimensionale ($x \in R^n$)
- $\triangleright\ c^T = [c_1, \ldots, c_n]$: vettore riga n-dimensionale
- $\triangleright\ c^T x = \sum_{j=1}^{n} c_j x_j$: prodotto scalare (o anche cx)
- $\triangleright\ A = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \ldots & a_{mn} \end{bmatrix}$: matrice $m \times n$
- $\triangleright\ A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} = \begin{bmatrix} a^1 \\ \vdots \\ a^m \end{bmatrix} = [A_1, \ldots, A_n] = [a_1, \ldots, a_n]$
- $\triangleright\ Ax = \begin{bmatrix} \sum_{j=1}^{n} a_{1j} x_j \\ \vdots \\ \sum_{j=1}^{n} a_{mj} x_j \end{bmatrix} = \begin{bmatrix} a_1^T x \\ \vdots \\ a_m^T x \end{bmatrix} = \begin{bmatrix} a^1 x \\ \vdots \\ a^m x \end{bmatrix}$
- $\triangleright\ Ax = b \quad \rightarrow \begin{cases} a_1^T x = b_1 \\ \vdots \\ a_m^T x = b_m \end{cases} \equiv \begin{cases} a^1 x = b_1 \\ \vdots \\ a^m x = b_m \end{cases}$
- $\triangleright$ rango(A): rango di A
- $\triangleright$ det(A): determinante di A
- $\triangleright\ A^{-1}$: matrice inversa di A

Combinazione Convessa

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

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



Es. $x,y \in R^2$

Combinazione Convessa (2)

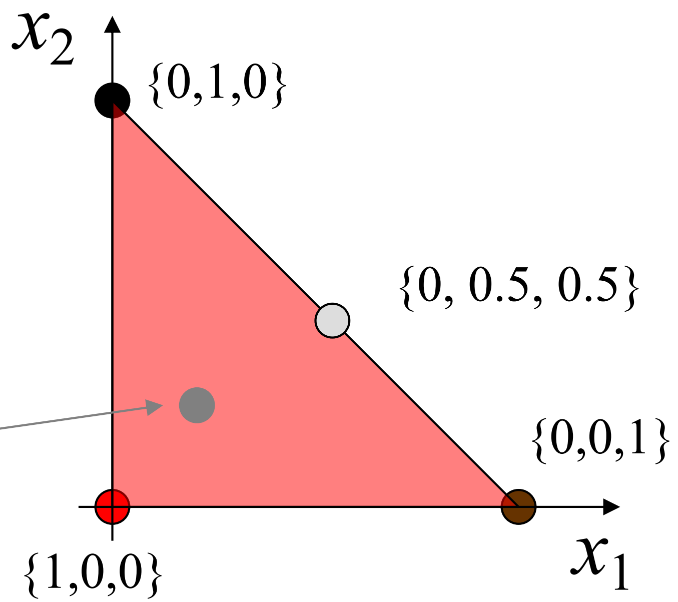
Def.: Combinazione convessa di K punti $p_1, p_2, \dots, p_K \in R^n$

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

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

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

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



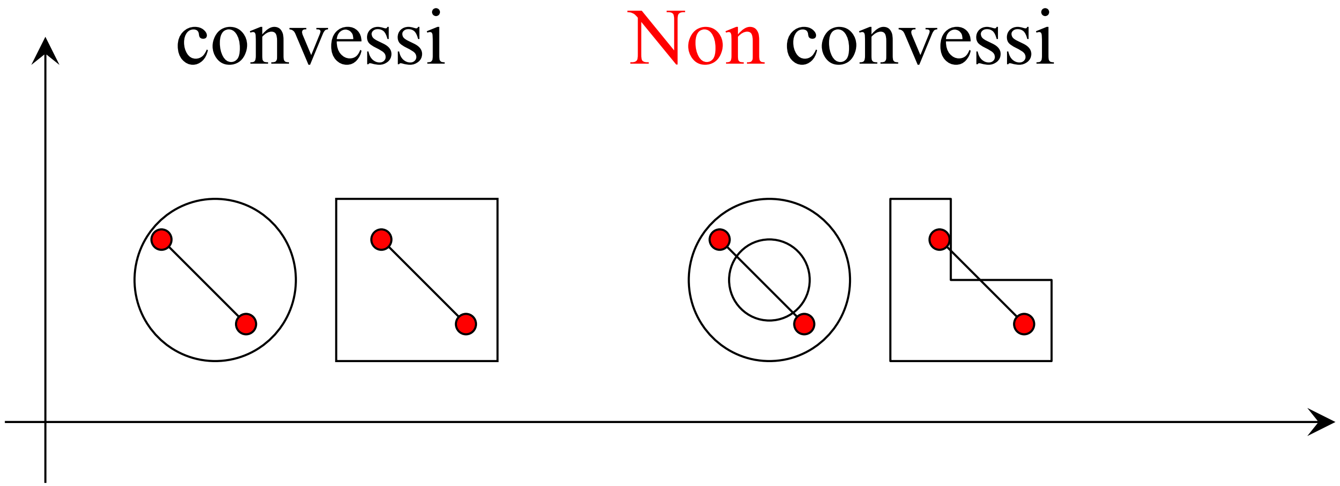
Insiemi Convessi

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

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

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

Es. $x, y \in R^2$



Proprietà degli Insiemi Convessi

Proprietà 0: R^n è convesso (ovvio)

Proprietà 1: Dati F_i convessi $\Rightarrow$

$F = \cap\, F_i$ è **convesso**

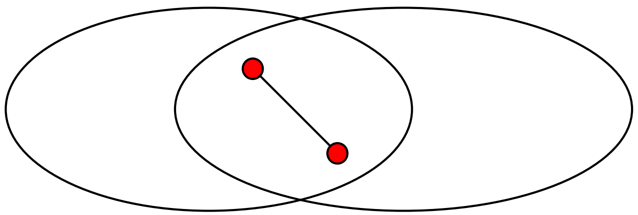
DIM.:

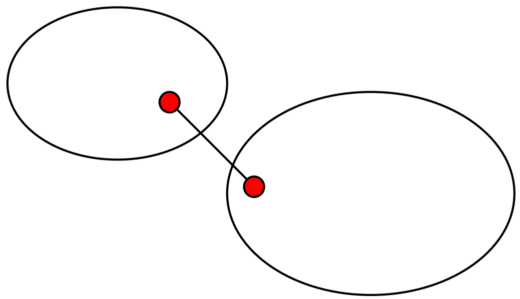
$$x,y \in F \Rightarrow x,y \in F_i \;\forall\, i$$

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

$$\Rightarrow z \in \cap\, F_i \;\square$$

$\cap\, F_i$





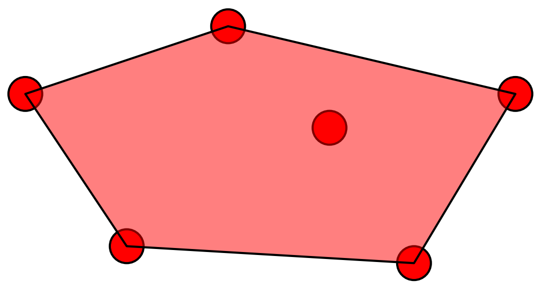
$\cup\, F_i$

Vertici ed Insiemi Convessi

Def.: z è vertice di un insieme convesso F

$\Leftrightarrow$ non è combinazione convessa di altri punti di F

Def.: Dato un insieme di punti $P = \{p_1, p_2, \ldots, p_K\} \subset R^n$ si dice chiusura convessa di P, $conv$(P) il più piccolo insieme convesso che contiene P.

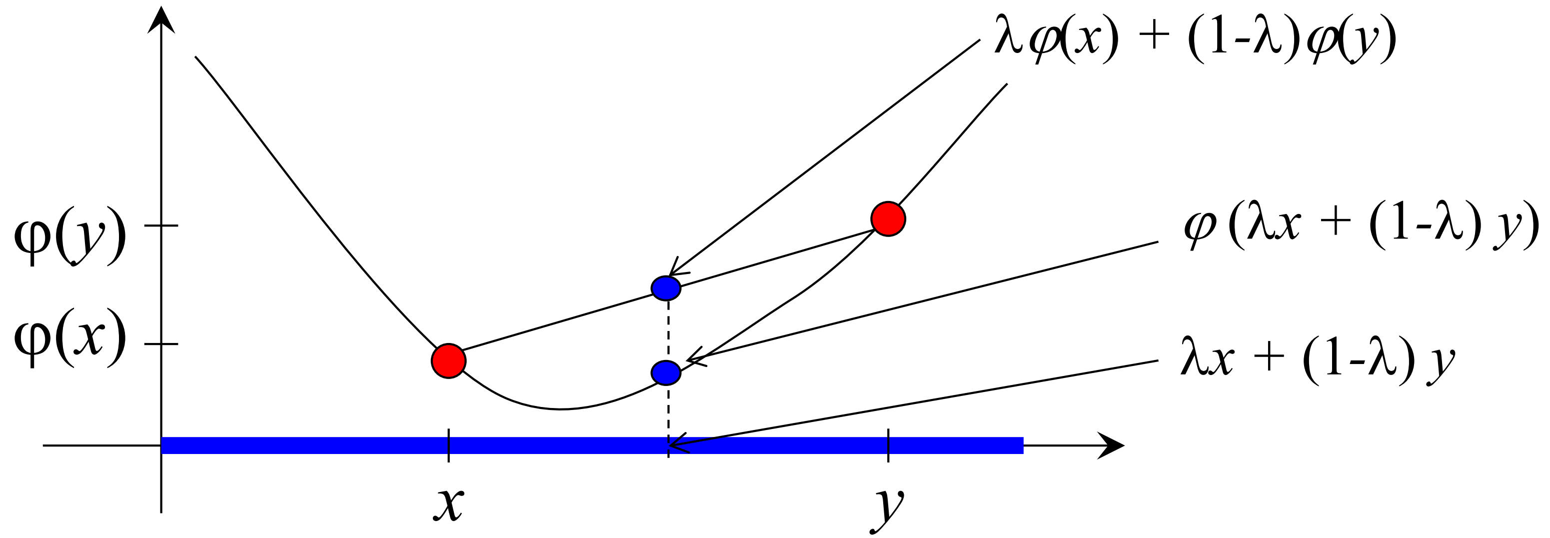


Funzioni Convesse

Def.: Dato $F \subseteq R^n$ convesso, $\varphi : F \rightarrow R$ è convessa in F se $\forall\, x,y \in F, \forall\, \lambda \in [0,1]$, si ha

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

Es. $F = [0,1] \subset R$



Problemi di ottimizzazione

Problemi di Ottimizzazione

- $x = (x_1, \ldots, x_n) \in \mathrm{R}^n$: 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 ...
- $F \subseteq \mathrm{R}^n$: insieme delle soluzioni ammissibili
  (regione ammissibile)
- $\varphi : F \rightarrow \mathrm{R}$: funzione obiettivo (f. costo)

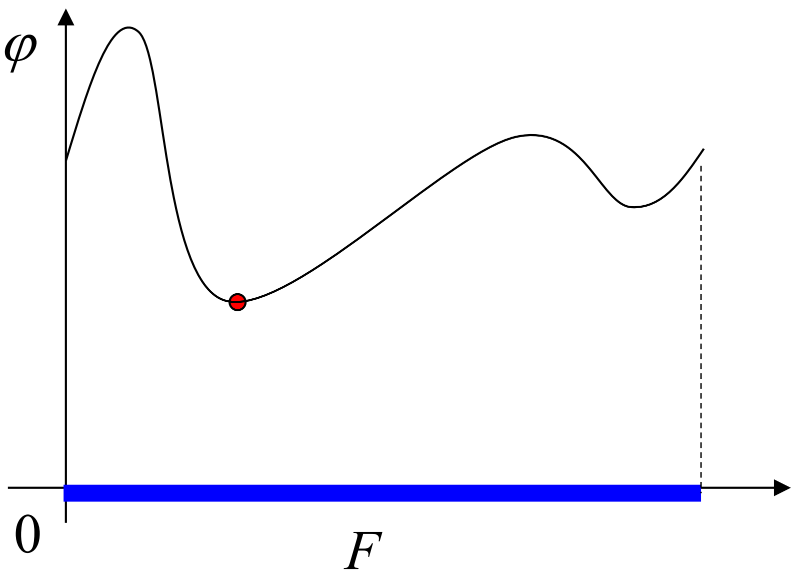
$$\boxed{(P) \quad \min_{x \in F} \varphi(x)}$$

Problemi di Ottimizzazione (2)

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

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

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



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 $x \in F$

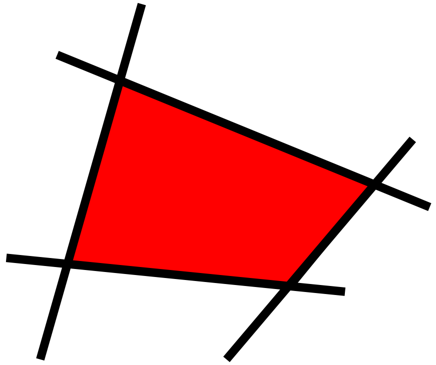
  Es. $[0,1]^2$; *x* intere nell'ipercubo di lato 1

  Elenco la regione ammissibile che ci serve: da 1 a 10 etc...

- implicitamente: servendosi di equazioni e disequazioni

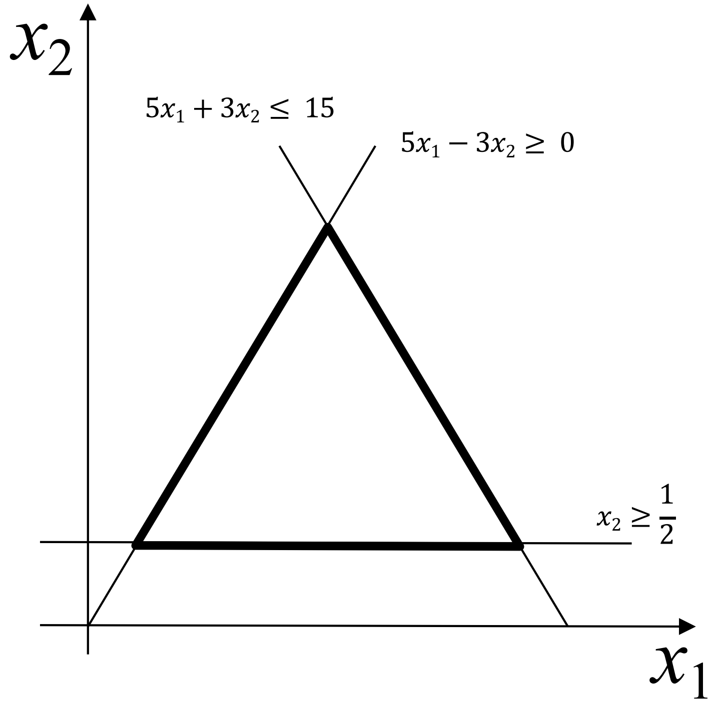
  Definisco la regione ammissibile attraverso una funzione

$$F := \{x \in \mathrm{R}^n : g_i(x) \leq 0, (i=1, \ldots, m)$$
$$h_j(x) = 0, (j=1, \ldots, p)\}$$



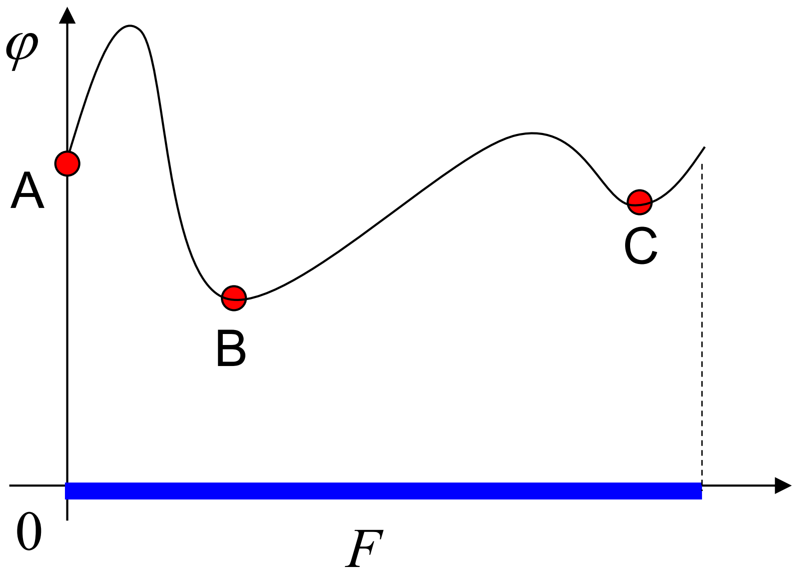
Esempio di regione ammissibile

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



Minimi Locali e Globali

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

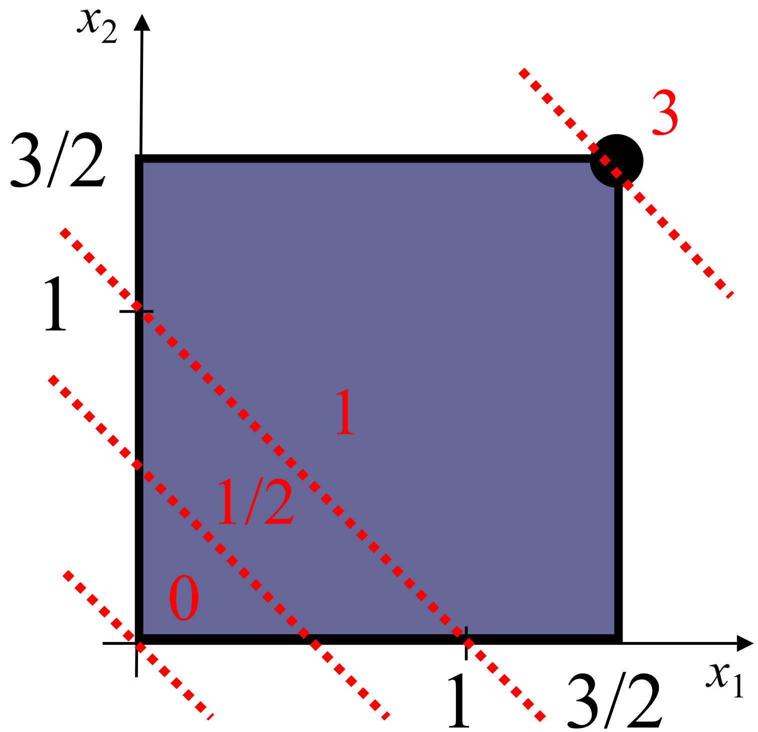


(P) richiede di trovare almeno un ottimo globale

Esempio 1

- problema continuo:

$$F = [0,3/2]^2 \subset R^2 \qquad \max \varphi(x) = x_1 + x_2$$

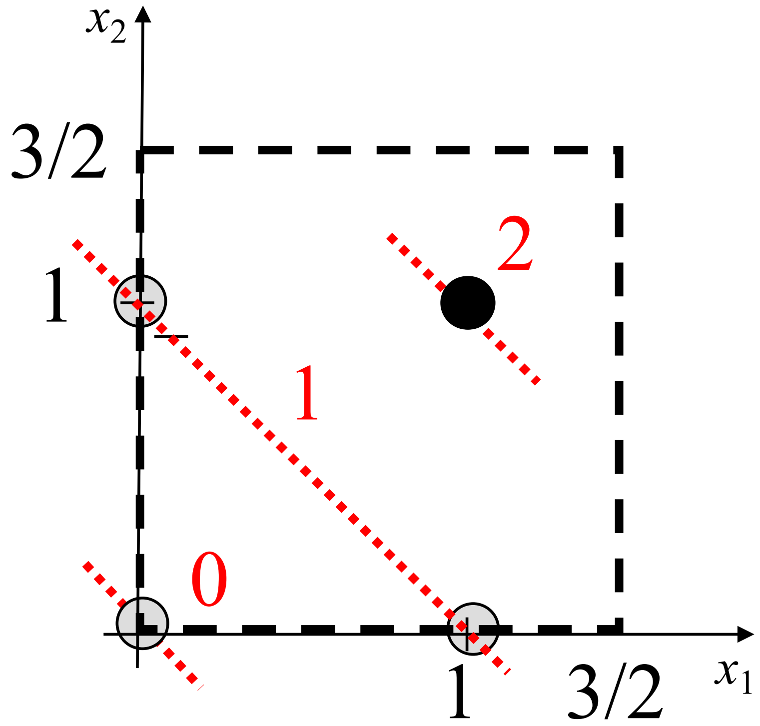


$x_1 + x_2 = \text{costante}$

Esempio 2

- problema discreto:

$F = \{0, 3/2\}^2 \subset R^2 \qquad \max \varphi(x) = x_1 + x_2$



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

se $F=\{0,1\}^{100}$

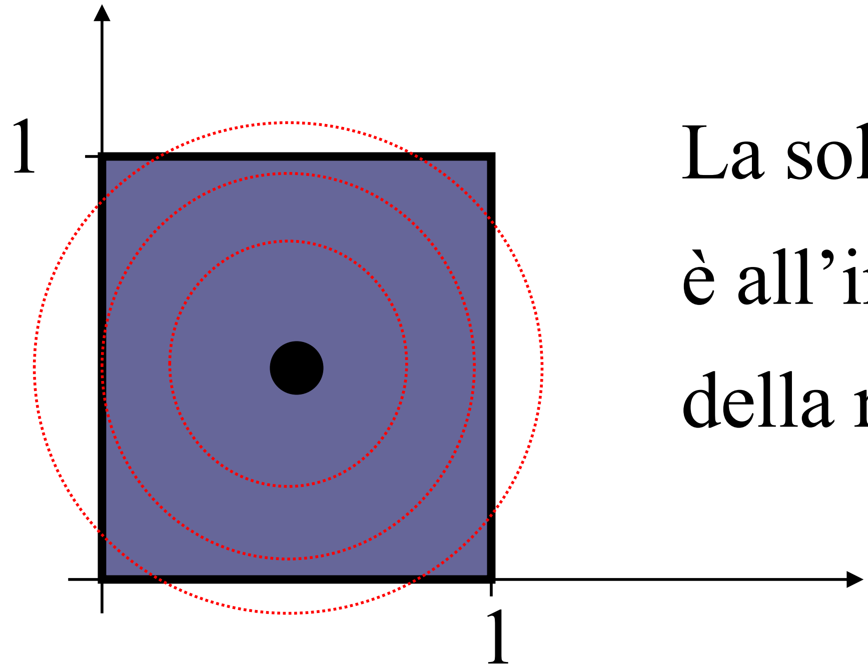
$\Rightarrow 2^{100} \sim 10^{30}$ valutazioni

Esempio 3

- problema continuo:

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

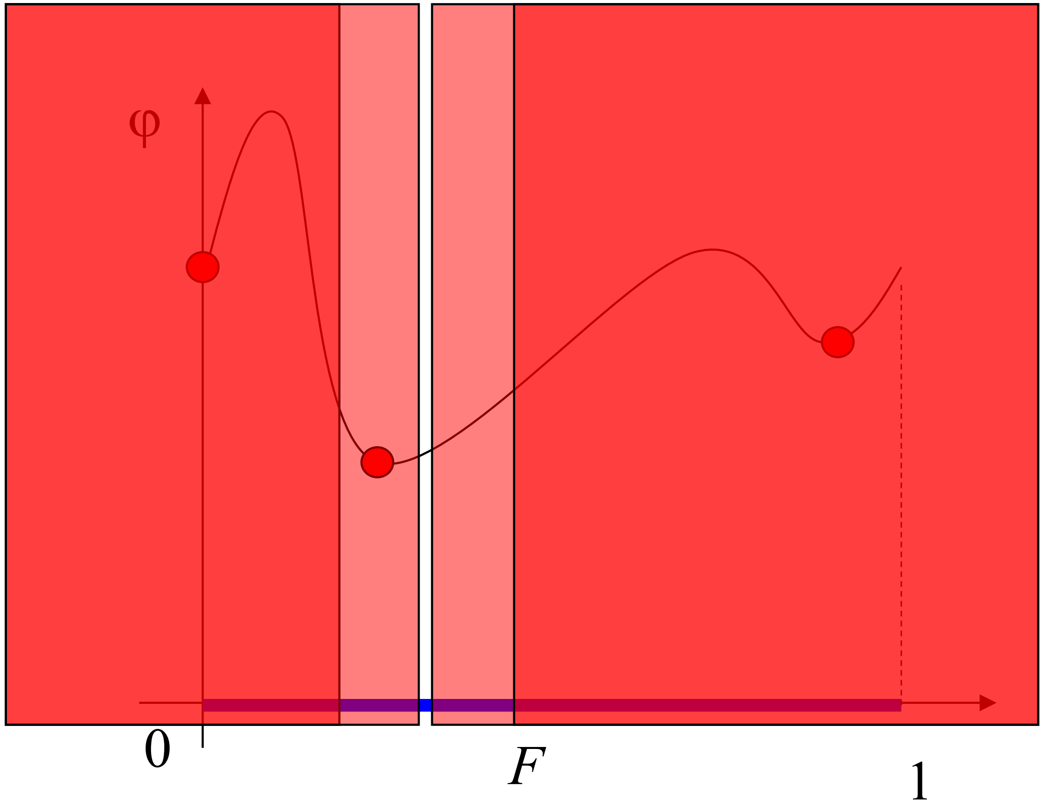
$$\min \varphi(x) = (x_1 - 1/2)^2 + (x_2 - 1/2)^2$$



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)

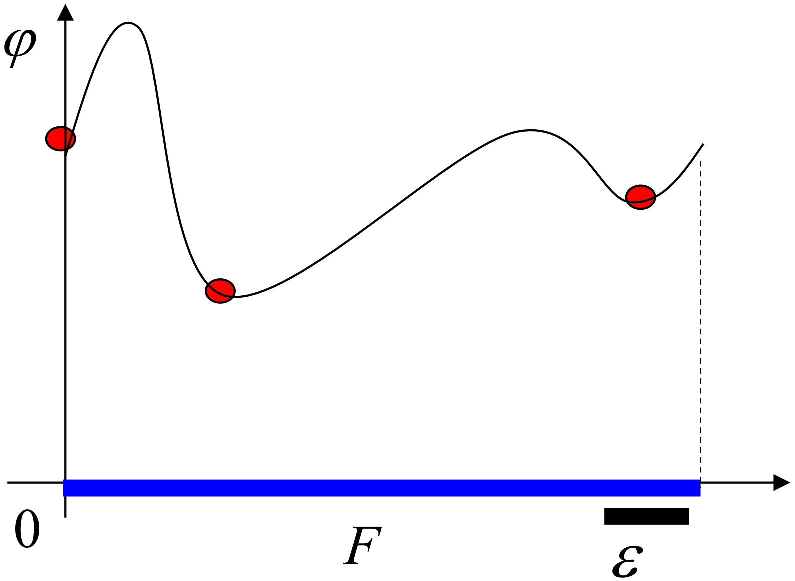
- La sequenza $x_0, x_1, \ldots, x_k$ converge alla soluzione ottima x^* (o ad un ottimo locale)
- Nel caso generale (φ ed F qualsiasi) la convergenza è ad un ottimo locale ed il numero di iterazioni è molto elevato
- Nel caso continuo si termina quando si è raggiunta l'approssimazione desiderata (piccole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



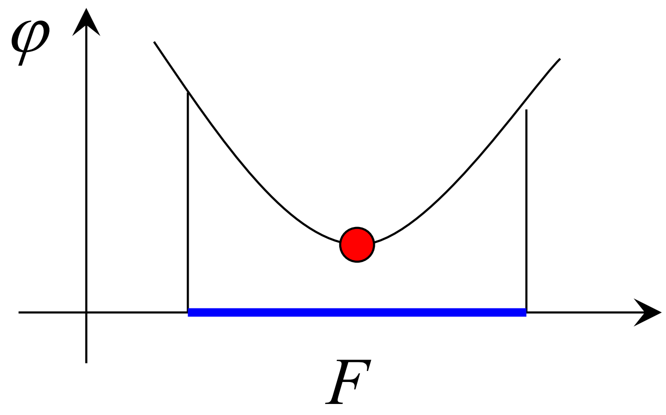
N è esatto se un ottimo locale rispetto ad N è ottimo globale

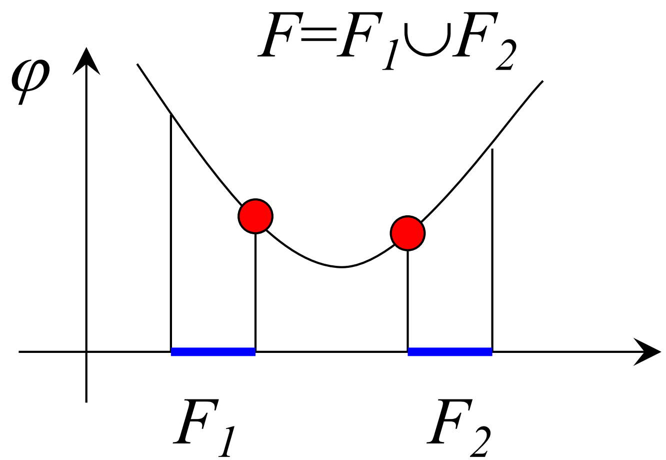
(Es. N_1)

Convessità ed Intorno Euclideo

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

$\Rightarrow$ (ottimo locale $\equiv$ ottimo globale)





Dimostrazione

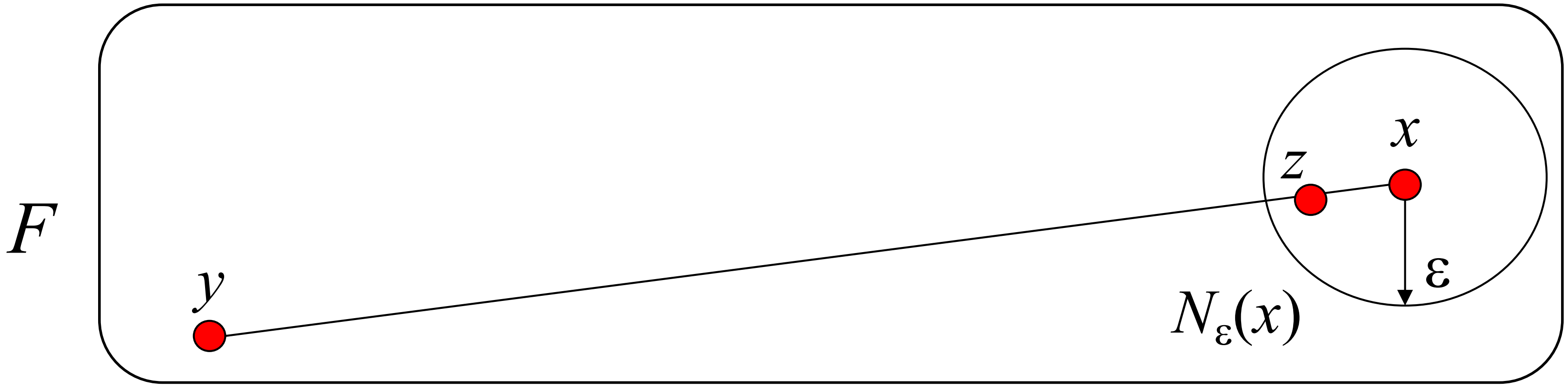
x = ottimo locale rispetto ad N_ε ma non globale !!; lo è $y \in F$;

$z = \lambda x + (1- \lambda)\, y$ in $N_\varepsilon(x)$ (λ prossimo a 1) ;

$$\varphi(z) = \varphi(\lambda x + (1-\lambda)\, y) \le \lambda \; \varphi(x) + (1-\lambda)\, \varphi(y) \;\Rightarrow$$

$$\Rightarrow \quad \varphi(y) \ge [\varphi(z) - \lambda\, \varphi(x)]/(1-\lambda); \;\; z \in N_\varepsilon(x) \Rightarrow \varphi(z) \ge \varphi(x)$$

$$\Rightarrow \quad \varphi(y) \ge [\varphi(x) - \lambda\, \varphi(x)]/(1-\lambda) = \varphi(x) \text{ (contraddizione !!)}$$



Classificazione

φ, g_i, h_j qualunque $\Rightarrow$ Progr. Non Lineare (PNL, NLP)

Non esistono algoritmi generali di ottimizzazione

$\exists$ metodi che convergono a ottimi locali

φ, g_i, convesse $\Rightarrow$ Programmazione Convessa (PC,CP)

h_j lineari

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 (Simplesso, Elissoide …)

PL con variabili intere $\Rightarrow$ Progr. Lineare Intera (PLI, ILP)

Problema difficile ($\propto$ PNL)

$\exists$ algoritmi generali (Branch-and-Bound, Branch-and-Cut, Branch-and-Price …)