RILASSAMENTO LINEARE

Trasforma un PLI in un PL.

$$\begin{cases} 3x_1 + 4x_2 + 2x_3 \le 60 \\ 2x_1 + x_2 + 2x_3 \le 40 \\ x_1 + 3x_2 + 2x_3 \le 80 \\ x_1,\ x_2,\ x_3 \in \mathbb{Z}^3. \end{cases} \quad \Rightarrow \quad \begin{cases} 3x_1 + 4x_2 + 2x_3 \le 60 \\ 2x_1 + x_2 + 2x_3 \le 40 \\ x_1 + 3x_2 + 2x_3 \le 80 \\ x_1,\ x_2,\ x_3 \ge 0. \end{cases}$$

$$\begin{cases} 3x_1 + 4x_2 + 2x_3 \le 60 \\ 2x_1 + x_2 + 2x_3 \le 40 \\ x_1 + 3x_2 + 2x_3 \le 80 \\ x_1,\ x_2,\ x_3 \in \{0,1\}. \end{cases} \quad \Rightarrow \quad \begin{cases} 3x_1 + 4x_2 + 2x_3 \le 60 \\ 2x_1 + x_2 + 2x_3 \le 40 \\ x_1 + 3x_2 + 2x_3 \le 80 \\ x_1 \le 1, x_2 \le 1, x_3 \le 1 \\ x_1,\ x_2,\ x_3 \ge 0. \end{cases}$$

Ora applico il metodo del simplesso oppure la risoluzione grafica.

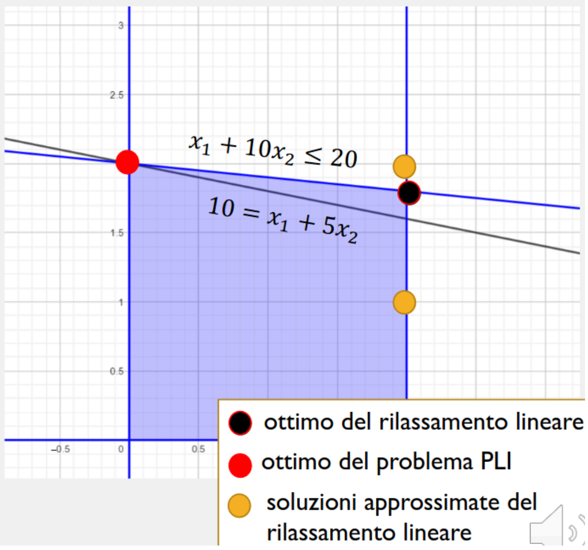
Se la soluzione ottimale è <u>INTERA</u> e <u>SODDISFA I VINCOLI di entrambi</u> i PL allora posso dire che la soluzione è <u>AMMISSIBILE</u>.
Non posso dire niente se la soluzione è ottima.

- Non vi è nessuna garanzia che la soluzione ottima di un problema PLI corrisponda alla soluzione approssimata del rilassamento lineare

Esempio

$$\max z = x_1 + 5x_2$$
$$\begin{cases} x_1 + 10x_2 \le 20 \\ x_1 \le 2 \\ x_1, x_2 \in \mathbb{Z}^2 \end{cases}$$

- La soluzione ottima del rilassamento lineare è $\left(2, \frac{9}{5}\right)$
- La soluzione intera ottenuta approssimando $\left(2, \frac{9}{5}\right)$ alle soluzione intera più vicine sono $(2,2)$ e $(2,1)$
- La soluzione ottima del problema PLI è $(0,2)$



Se il risultato non è intero allora:
- PLI di MAX -> Upper Bound
- PLI di MIN -> Lower Bound

Se ho un PLI di MAX è la soluzione ottimale è 4, allora nel PL mi uscirà come valore della funzione obiettivo qualcosa come 4,1 o 4,4.

BRANCH AND BOUND

B&B PER PROBLEMI LINEARI INTERI BINARI

PUNTO 1

- Verificare che i vincoli di non negatività siano rispettati.

- La variabile incombente è il valore della funzione obiettivo.
  Variabile incombente = +∞ se è un PL di MAX
- -∞ se è un PL di MIN.

PUNTO 2

Usando il rilassamento lineare trovo la soluzione ottimale del problema completo. Questa soluzione diventerà la soluzione "incombente".
Se la soluzione mi viene 10,5 posso semplificarla con 10 (per difetto se è di MAX) oppure con 11 (per eccesso se è di MIN).

Attenzione, già da subito devo effettuare il PUNTO 5. Se la soluzione è ottima allora posso eliminare il ramo. Il B&B è concluso

PUNTO 3 - BRANCHING

Ad ogni iterazione, per muoversi nell'albero, posso scegliere se usare il Depth First oppure il Best Bound First. A meno che l'esercizio non mi chieda di usarne uno specifico. Sono spiegati entrambi nella teoria.

Ora partiziono l'area ammissibile. Per farlo scelgo la x_i più lontana da un numero intero.

Dopo aver scelto x_i, bisogna <u>fissarla</u>. Essa diventerà una variabile di branching. Questo posso farlo solo con le variabili binarie.

A quel punto ci sarà un problema con x_i=0 e x_i=1.

PUNTO 4 - BOUNDING

Usare il rilassamento lineare per trovare le 2 soluzioni ottimali.

I limiti sono <= se è un problema di MAX, viceversa per un MIN.

PUNTO 5 - FATHOMING

LA SOLUZIONE RISPETTA I VINCOLI DI ENTRAMBI I PL?

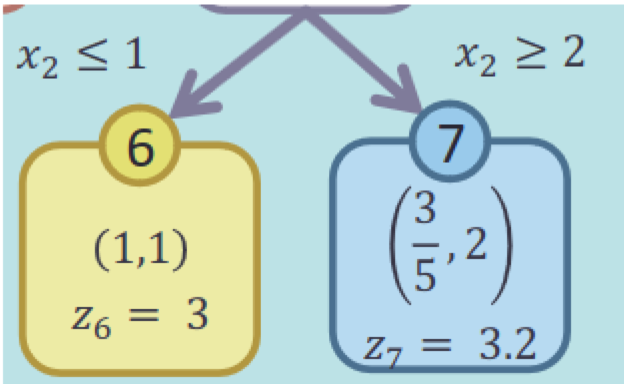
- NO, allora la soluzione è impossibile, elimino il ramo
- SI, vado avanti

LA SOLUZIONE È MIGLIORE DI QUELLA INCOMBENTE?
- SI, vado avanti
- NO, elimino il ramo

Cosa significa migliore?
- se è un PL di MAX allora deve essere la più alta possibile.
- se è un PL di MIN allora deve essere la più bassa possibile

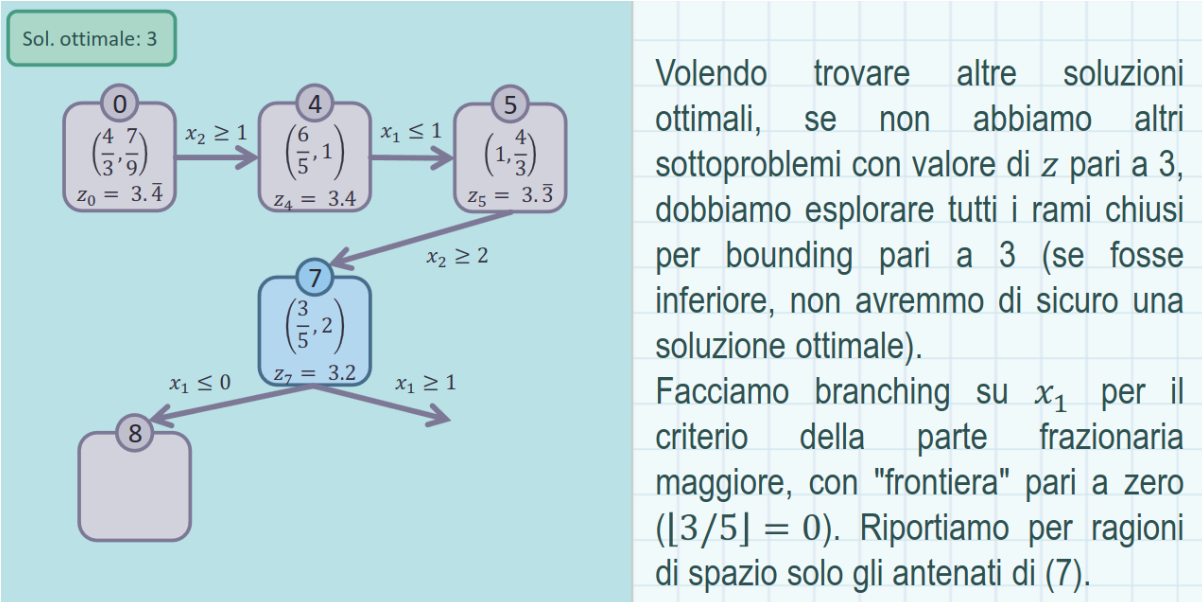
Se la variabile incombente di un PL di MIN è 10, e la soluzione della funzione obiettivo è 15, allora non va bene, non la sostituisco.
Se la variabile incombente di un PL di MAX è 10, e la soluzione della funzione obiettivo è 15, allora la sostituisco.

LA SOLUZIONE È COMPOSTA DA SOLI NUMERI INTERI?
- SI, sostituisco e chiudo il ramo
- NO, torno al punto 3 e scelgo la variabile con la funzione obiettivo più bella (non lo so bruh). A meno che, come nell'immagine qui sotto, il sottoproblema non sia già una variabile incombente. allora elimino il ramo



PUNTO 6
Quando tutti i rami sono chiusi e/o finiti allora la soluzione incombente è OTTIMALE.

SOLUZIONE MULTIPLE



Dopo di che basta semplicemente continuare il B&B finché non troviamo una nuova soluzione e tutti i rami non sono completati.

B&B PER PROBLEMI LINEARI INTERI MISTI

Guardare il PDF "B&B PLI"
Partire dalla slide 21

PUNTO 1
Verificare che i vincoli di non negatività siano rispettati.
La variabile incombente è il valore della funzione obiettivo.
All'inizio ovviamente il valore è nullo, però noi all'inizio mettiamo comunque la variabile incombente = +∞ se è un PL di MAX, viceversa se è un PL di MIN.

PUNTO 2
Usando il rilassamento continuo (capire cosa è per bene) trovo la soluzione ottimale del problema completo. Questa soluzione diventerà la soluzione "incombente".

Posso arrotondare la soluzione, dunque se ho (7/2, 7/2) posso arrotondare a (3, 3) oppure (4, 4).
Vedere ovviamente se la soluzione arrotondata è ancora ammissibile.
(per difetto se è di MAX) oppure (per eccesso se è di MIN).

Attenzione, già da subito devo effettuare il PUNTO 5. Se la soluzione è ottima allora posso eliminare il ramo. Il B&B è concluso

PUNTO 3 - BRANCHING
Ad ogni iterazione, per muoversi nell'albero, posso scegliere se usare il Depth First oppure il Best Bound First. A meno che l'esercizio non mi chieda di usarne uno specifico. Sono spiegati entrambi nella teoria.

Ora partiziono l'area ammissibile. Per farlo posso scegliere la x_i in base a:

- variabile più lontana da un numero intero
- variabile con indice più basso (prima x1, poi x2, …)

Dopo aver scelto x_i dobbiamo arrotondare la soluzione.
Se ho 7/2 allora $x_i = 3$ e $x_i = 4$

PUNTO 4 - BOUNDING
Usare il rilassamento lineare oppure guardo la soluzione grafica.

NON APPROSSIMO LE SOLUZIONI.

I limiti sono <= se è un problema di MAX, viceversa per un MIN.

PUNTO 5 - FATHOMING
LA SOLUZIONE RISPETTA I VINCOLI DI ENTRAMBI I PL?

- NO, allora la soluzione è impossibile, elimino il ramo
- SI, vado avanti

LA SOLUZIONE È MIGLIORE DI QUELLA INCOMBENTE?

- SI, vado avanti
- NO, elimino il ramo

Cosa significa migliore?

- se è un PL di MAX allora più è alta, meglio è
- se è un PL di MIN allora più è bassa, meglio è

Se la variabile incombente di un PL di MAX è 10, e la soluzione della funzione obiettivo è 10, allora elimino il ramo.

LA SOLUZIONE È INTERA

- SI, sostituisco il nuovo valore ed elimino il ramo
- NO, torno al PUNTO 3

PUNTO 6
Quando tutti i rami sono chiusi e/o finiti allora la soluzione incombente è OTTIMALE.

KNAPSACK

Oggetti	1	2	3	4
Valori	15	11	20	10
Pesi	5	2	3	4

$$\text{Capacità} = 8$$

PUNTO 1
Valore / Peso

Oggetto 1	Oggetto 2	Oggetto 3	Oggetto 4
3	11/2	20/3	5/2

Poi ordiniamo gli oggetti in ordine decrescente. Non scrivo i valori/peso perché non ci interessano più.

Oggetto 3	Oggetto 2	Oggetto 1	Oggetto 4

Ora eseguiamo il Knapsack con l'algoritmo greedy:

- Mettiamo oggetto 3
- (Capacità 8) - (Peso 3) = 5
- Valore totale = 20

- Mettiamo oggetto 2
- (Capacità 5) - (Peso 2) = 3
- Valore totale = 20 + 11

- Mettiamo oggetto 1, purtroppo non ci sta tutto
- Mettiamo solo il (Capacità rimanente) / (Peso oggetto 1)
- Quindi mettiamo $\frac{3}{5}$
- Valore totale = 20 + 11 + 15*$\frac{3}{5}$

La soluzione finale sarà ($\frac{3}{5}$, 1, 1, 0)
Il valore della funzione obiettivo sarà 40.

PUNTO 2

Ora devo trovare una soluzione AMMISSIBILE INTERA.

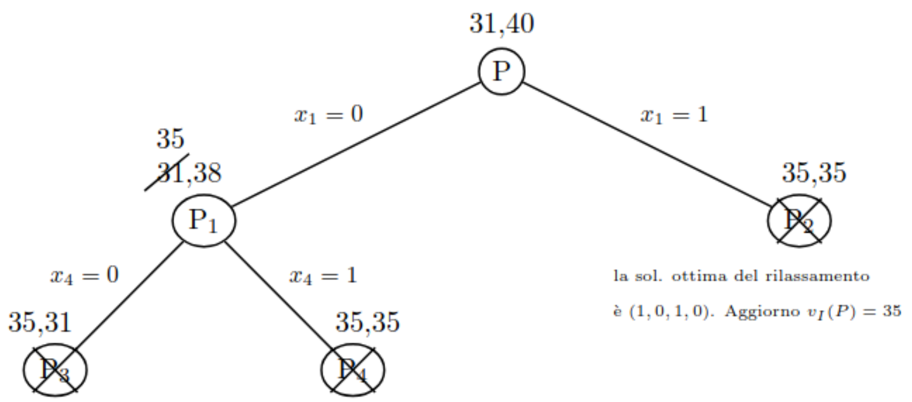
Faccio la stessa cosa di prima solo che quando arrivo ad inserire l’oggetto 1, al posto che dividerlo non lo aggiungo proprio.
Passo all’oggetto 4, purtroppo lui pesa 4, e abbiamo una capacità di 3. Quindi non ci sta e non lo aggiungiamo.

La soluzione finale sarà (0, 1, 1, 0)
Il valore della funzione obiettivo sarà solo 20 + 11 = 31

PUNTO 3

Applico il metodo B&B.

Albero di enumerazione:



soluzione ottima $= (1, 0, 1, 0)$ valore ottimo $= 35$

CONCETTI BASE ANALISI

CONTINUITÀ

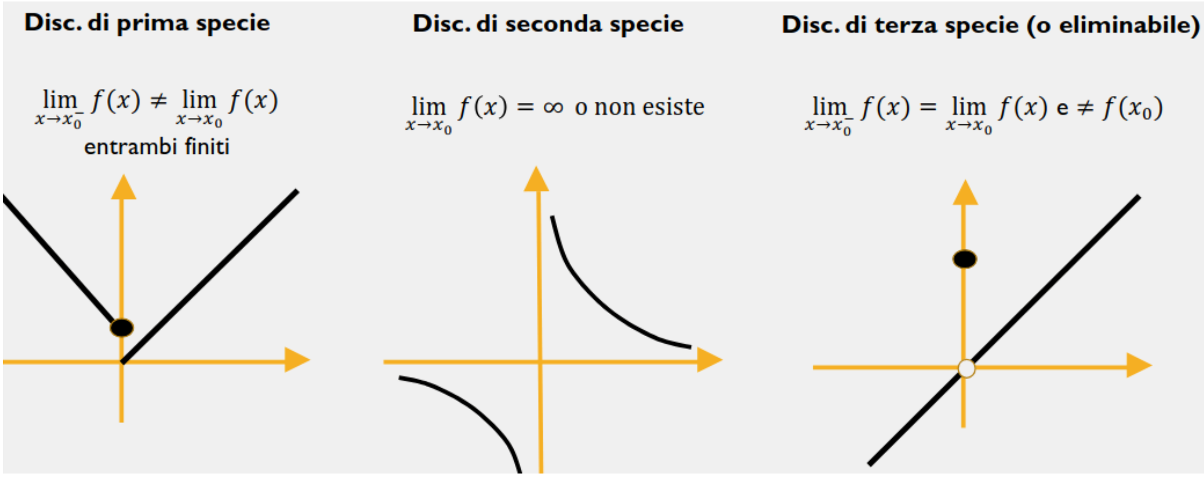
Diciamo che una funzione è continua in x0 se:

$$\lim_{x \to x_0^+} f(x) = \lim_{x \to x_0^-} f(x) = \lim_{x \to x_0} f(x) = f(x_0)$$

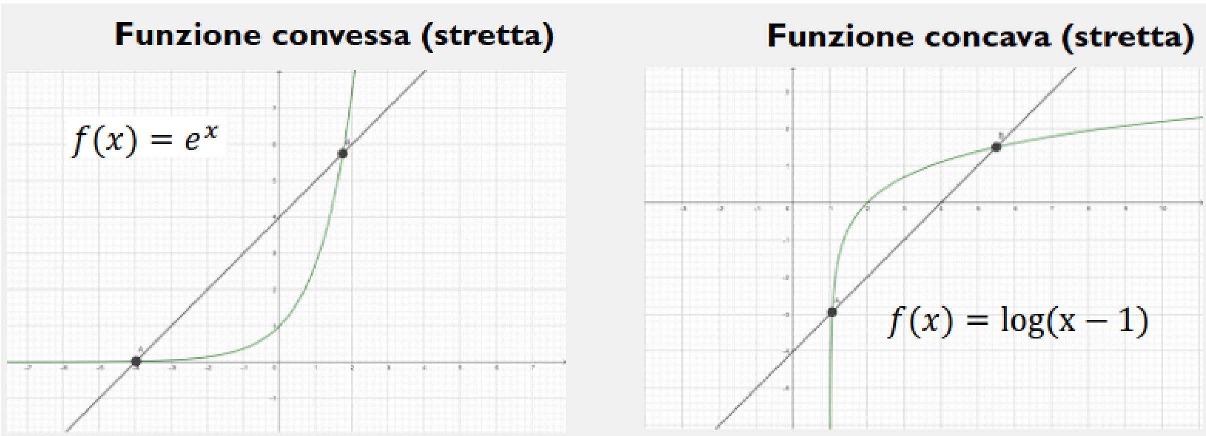
Tutte le funzioni del tipo: e, cos, sin, log, polinomiali sono sempre continue.

La somma/prodotto/composizione di funzioni continue restituisce una funzione continua.

DISCONTINUITÀ



CONVESSITÀ E CONCAVITÀ



Se per una funzione è possibile effettuare la derivata seconda allora:

- Se f’’(x) > 0 (ovvero è sopra la linea delle ascisse) allora f(x) è convessa in quello specifico intervallo
- Se f’’(x) < 0 allora f(x) è concava

Per trovare un punto di minimo/massimo locale:

1. Faccio la derivata seconda
2. Metto la derivata seconda =0 e trovo dei punti
3. Se sostituendo i punti a f’’(x) trovo 0 allora è un punto di flesso. Altrimenti è concava (<0) oppure convessa (>0).

4. Se è concava è punto di minimo, se è convessa è un punto di massimo

METODO DELLA BISEZIONE

PUNTO 1
Faccio la derivata seconda e vedo se è concava o convessa. Così capisco se è un punto di minimo/massimo

Pongo 2ε = 0.02

PUNTO 2
Faccio f'(a) e f'(b), dove l'idea è trovare 2 estremi, uno > 0 e uno < 0.

Quindi vado a caso e trovo un A e un B che siano > e < di 0

Poi faccio la media e trovo x0.

Poichè $f'(0) = 12 > 0$ e $f'(2) < 0$, possiamo porre $\underline{x} = 0$ e $\overline{x} = 2$ e

$$x_0 = \frac{\overline{x} - \underline{x}}{2} = 1$$

PUNTO 3
Ora faccio la derivata di x0 (che è xk attuale)

- Se f'(xk) > 0 ALLORA estremo inferiore = xk
- Se f'(xk) < 0 ALLORA estremo superiore = xk
- se f'(xk) = 0 ALLORA ho trovato la soluzione!

Calcoliamo $f'(x_0) \rightarrow f'(1) = -12$

$f'(x_0) < 0 \rightarrow$ poniamo $\overline{x} = 1$ (nuovo estremo superiore)

Ora devo ottenere xk+1 (che sarà il nuovo xk)
Quindi faccio la media tra estremo inferiore e estremo superiore
Quindi x-inferiore = 0 e x-superiore = 1, xk+1 sarà 0,5

$$x_{k+1} = \frac{\underline{x} + \overline{x}}{2}$$

Verifico se estremo superiore - estremo inferiore < 2ε
Se è vero posso fermarmi, ho trovato la soluzione.

PUNTO 4
Dobbiamo stabilire un criterio di arresto:

- xk+1 - xk < 2ε a quanto pare questo non viene usato
- la traccia mi dice quante iterazioni fare
- se i valori di xk tendono a 0 oppure a ∞
- se si verificano cicli, ovvero i valori di xk e xk+1 continuano a scambiarsi

Se non mi arresto torno al **PUNTO 3.**

ALGORITMO DI NEWTON

- faccio derivata prima e derivata seconda
- Scelgo 2ε = 0.02
- Scelgo xk = 1 (oppure no, dipende dai casi)

Si ponga $x_{k+1} = x_k - \dfrac{f'(x_k)}{f''(x_k)}$

I criteri di arresto è:
xk+1 - xk < 2ε

GRADIENTE

$$f = 15x_1 + 2(x_2)^3 - 3x_1(x_3)^2$$

$$\downarrow$$

$$\nabla f = \begin{bmatrix} 15 - 3(x_3)^2 & 6(x_2)^2 & -6x_1x_3 \end{bmatrix}$$

Quindi per la prima posizione derivo tutti gli elementi che hanno x1.
Quindi 15x1 diventa 15
-3x1 (x3)^2 diventa -3(x3)^2
(non derivo x3 in quanto è derivata parziale per x1)
quando ho (x-1)y e voglio fare la derivata lo risolvo tutto, altrimenti faccio solo casini

HESSIANA

$$\nabla f = [15 - 3(x_3)^2 \quad 6(x_2)^2 \quad -6x_1x_3]$$

$$\nabla^2 f = \begin{bmatrix} 0 & 0 & -6x_3 \\ 0 & 12x_2 & 0 \\ -6x_3 & 0 & -6x_1 \end{bmatrix}$$

Anche qua fare derivate parziali (per x1, per x2, per x3)

TROVARE PUNTO DI MASSIMO/MINIMO

Data una f(x)

- faccio gradiente
- metto quello che trovo = 0 dentro un sistema. In questo modo trovo un punto stazionario
- Faccio hessiana

A   B
C   D

- Calcolo il determinante -> A*D - C*B
- Traccia = A+D

- Entrambi positivi -> convessa, punto di minimo
- Entrambi negativi -> concava, punto di massimo
- Uno dei due è uguale a 0 -> semidefinita positiva oppure semidefinita negativa, quindi è convessa/concava/punto di sella

- Uno negativo, e l'altro positivo -> non posso dire nulla oppure uso gli autovalori

AUTOVALORI

Mi invento una variabile a cazzo -> t

(A - t) (D - t) - 1 = 0

trovo t1 e t2 con la classica formula:
-b +- radice(b^2 - 4ab) / 2a

Uso lo stesso criterio usato prima.
Se sono entrambi diversi allora è **indefinita**.

METODO DEL GRADIENTE

Iterazione numero k
- poniamo ε1=0,01 ed ε2=0,1
- il punto (x, y) da cui partiamo lo dice la traccia
- calcoliamo gradiente di (x, y)
- Creiamo D^k
- D^k = gradiente(x, y)
- Se è MIN allora moltiplicare tutto per -1

Dobbiamo calcolare il punto successivo
- Il nuovo punto è x^k+1 = (x, y) + a*gradiente(x, y)
- Esce qualcosa come [a a] oppure [2a 1/2a]

Devo calcolare a!
- per calcolare a -> sostituire [a a] alla f(x) iniziale, la sostituisco con tutte le x1 e x2
- Poi faccio la derivata di quello che mi esce fuori
- Poi metto =0 e trovo il valore di a

Ora sostituisco a dentro [a a]
Questo è il nuovo punto.

CRITERIO DI ARRESTO
- f(x^k) - f(x^k+1) > ε1
- gradiente(x^k+1) > ε2
- appena uno di questi è falso, si interrompe il metodo

Altrimenti torniamo a capo, e calcoliamo il gradiente inserendo [a a]

Confrontiamo il f(x, y iniziali) con f(a, a dell'ultima iterazione)
Se è di MIN e diminuisce, allora migliora
Se migliora faccio
- gradiente(a, a)
- se viene (0, 0) può essere un punto di minimo/massimo

- a quel punto facciamo traccia e determinante, etc…

METODO DI NEWTON

- poniamo ε1=0,01 ed ε2=0,1
- Ci daranno un punto (x, y)
- gradiente(x, y)
- hessiana

Calcoliamo il nuovo punto x^k+1
- gradiente(x, y) = [v1 v2] hessiana

Ora immagina hessian così:
A   B
C   D

- -x = A*v1 + B*v2
- -y = C*v1 + D*v2
- ricordati di negare x e y
- troviamo v1 e v2, che è il nuovo punto

Ora devo fare:
x^k(che un punto x, y) + [v1 v2]
Questo è il punto x^k+1
Ora faccio gradiente(x^k+1)

Se è (0, 0) allora faccio traccia e determinante per trovare massimo/minimo…

KKT

Data la funzione MIN x+y
Con vincoli:
- x - 4y > 3
- -x + y^2 > 0

Faccio il gradiente di f(x) -> [A B]
Dove A e B sono i numeri che ti escono

Riscrivo tutti i vincoli come:

h1(x, y) = -x + 4y + 3 < 0

h(x, y) < 0
u1 > 0
Calcolo il gradiente -> [-1 4]

h2(x, y) = x - y^2 < 0
u2 > 0
Calcolo il gradiente [1 -2y]

Ora dentro una grande parentesi graffa scriviamo tutta questa roba

---

Se è un problema di MINIMO
A = -u1(-1) - u2(1)
B = -u1(4) - u2(-2y)

Se è un problema di MASSIMO
A = +u1(-1) + u2(1)
B = +u1(4) + u2(-2y)

---

u1*(-x + 4y + 3) = 0
u2*(x - y^2) = 0

-x + 4y + 3 < 0
x - y^2 < 0

u1, u2 > 0
Quindi ho 2^2 vincoli da verificare

u1=0 u2=0
u1>0 u2>0
u1>0 u2=0
u1=0 u2>0

Partiamo dal primo, ricordati che A=1 e B=1

A = -u1(-1) - u2(1) -> A=0
B = -u1(4) - u2(-2y) -> B=0
u1*(-x + 4y + 3) = 0 -> 0=0
u2*(x - y^2) = 0 -> 0=0
-x + 4y + 3 < 0
x - y^2 < 0
u1, u2 > 0

Il fatto che A=0 e B=0 è in quasi tutti i casi IMPOSSIBILE

Facciamo il secondo con u1>0 e u2>0

```
A = -u1(-1) - u2(1)
B = -u1(4) - u2(-2y)
u1*(-x + 4y + 3) = 0 -> -3=0
u2*(x - y^2) = 0 -> y=0
-x + 4y + 3 < 0
x - y^2 < 0
u1, u2 > 0
```

Peccato! Perché -3=0, altrimenti avrei trovato un candidato.

Facciamo il quarto con u1=0 e u2>0

```
A = -u1(-1) - u2(1) -> A=-u2 -> u2=-1
B = -u1(4) - u2(-2y) -> B=-u2(-2y)=2yu2
u1*(-x + 4y + 3) = 0 -> 0=0
u2*(x - y^2) = 0 -> y=0
-x + 4y + 3 < 0
x - y^2 < 0
u1, u2 > 0
```

Essendo che siamo in un problema di MINIMO u2=-1 è un candidato per punto di MASSIMO. Ovviamente vale il viceversa.

TEORIA

PROGRAMMAZIONE LINEARE INTERA

Le variabili di decisione devono essere intere.
Le applicazioni di PLI sono principalmente problemi di produzione, logistica, assegnamento.
Infatti spesso richiedono delle variabili intere.

Mentre se solo alcune variabili sono intere e altre no, allora si parla di PLM (programmazione lineare mista).

Nella PLI le soluzioni ammissibili non sono infinite, ma sono un numero finito.
In sostanza non abbiamo più una regione ammissibile convessa.
Infatti il nostro poligono ora può assumere solo qualche soluzione.

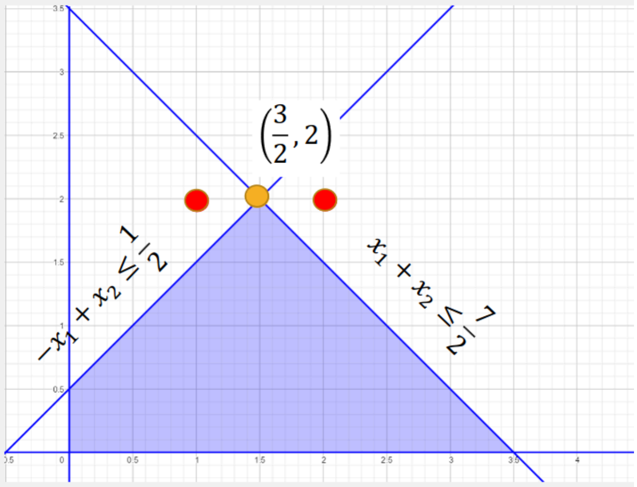
Inoltre il simplesso non funziona quando si tratta di risolvere PLI.
Esiste un metodo per trasformare un PLI in un PL. Questo metodo si chiama “rilassamento lineare”.

Questo metodo però non è affidabile per trovare la soluzione ottimale di un PLI. Però è molto efficace nel trovare una soluzione ammissibile in quanto è molto facile capire se lo è oppure no.

- Una soluzione ottima di un problema PL non è necessariamente ammissibile dopo l’arrotondamento

Esempio

$$\max z = x_2$$
$$\begin{cases} -x_1 + x_2 \leq \frac{1}{2} \\ x_1 + x_2 \leq \frac{7}{2} \\ x_1, x_2 \in \mathbb{Z}^2 \end{cases}$$



- La soluzione ottima del rilassamento lineare è il punto $\left(\frac{3}{2}, 2\right)$
- Le soluzioni ottenute approssimando $x_1 = \frac{3}{2}$ all’intero più vicino sono entrambe non ammissibili

Nei problemi di trasporto quando rilasso un PLI trovo sempre e comunque una soluzione intera, e quindi ammissibile.

Ad ogni modo il rilassamento lineare viene usato insieme ad un’altro metodo -> Branch and Bound.

Esiste anche la PLB (binaria). Le variabili binarie valgono 0 oppure 1 e vengono usate per:
- l’analisi di investimenti
- spedizioni di beni (calcolo del percorso ottimo, dove ogni arco vale 0 oppure 1)
- scelta di un posto per un servizio

$$\max z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

soggetto ai seguenti vincoli:

$$\begin{cases} 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \\ x_3 + x_4 \leq 1 \\ x_3 \leq x_1 \\ x_4 \leq x_2 \\ x_1, x_2, x_3, x_4 \in \{0,1\} \end{cases}$$

Il problema della PLB è che date *n* variabili binarie le soluzioni possibili sono 2^n.

BRANCH AND BOUND

È una tecnica di enumerazione implicita.

- valuta tutte le soluzioni fino a trovare quella ottima
- scarta alcune soluzioni, dimostrando che non sono ottime
- si basa sul concetto di dividi et impera

Questa tecnica divide un problema in più sottoproblemi, dove ogni sottoproblema è una porzione dell'area ammissibile.
Sia z* = f(x*) la soluzione ottima del problema completo.
SIa z° = f(z°) la soluzione ottima di un sottoproblema.
Allora si ha che f(x°) <= f(x*)
Questo se il problema è di MASSIMO, se il problema è di MINIMO allora f(x°) >= f(x*)

Il B&B fa uso di 3 tecniche differenti:

- **branching**, partizione dell'area ammissibile in più sottoproblemi
- **bounding**, determinazione di un limite superiore/inferiore
- **fathoming**, eliminazione di sottoproblemi

Durante il Branching ci sono 2 modi per muoversi nell'albero:

- **Depth-First** -> ovvero scelgo il sottoproblema creato più di recente. semplice da implementare, permette di ottenere presto delle soluzioni ammissibili (ci si avvicina più rapidamente alle foglie) e limita la memoria necessaria per memorizzare l'albero delle soluzioni, visto che si tendono a chiudere molti nodi per inammissibilità e rimangono pochi nodi contemporaneamente aperti. Per contro, esplora sotto-alberi con soluzioni scadenti.

- **Best Bound First** -> scelgo il sottoproblema con il miglior bound, limita di molto il numero di nodi visitati. Per contro, l'esplorazione tende a rimanere a livelli poco profondi, dove i problemi sono meno vincolati e, di conseguenza, i bound sono più promettenti ma difficilmente si ottengono presto soluzioni ammissibili che migliorino la soluzione incombente corrente. Questo impedisce di applicare efficacemente le regole di fathoming, maggiore la richiesta di memoria

Invece i criteri del fathoming sono questi:

Ricordando che i criteri di fathoming derivanti dall'analisi di un problema rilassato :

Criterio 1: E' stata trovata una soluzione ottima intera del sotto-problema

Criterio 2: La soluzione ottima Z del sotto-problema deve essere peggiore della soluzione incombente Z^* ($Z \leq Z^*$ per un problema di Max e viceversa per un problema di Min)

Criterio 3: Il sotto-problema non presenta soluzioni ammissibili

Quando tutti i nodi sono dichiarati fathomed (chiusi) il B&B si arresta. La soluzione incombente attuale diventa la soluzione ottima.

Il B&B inizia mettendo la variabile incombente a -oo se è un PL di MAX, +oo se è un PL di MIN.

Rilassamento Lagrangiano

L'intero insieme di vincoli funzionali $\boldsymbol{Ax} \leq \boldsymbol{b}$ vine cancellato (o quasi interamente cancellato) e la funzione obiettivo

$$Max\ Z = \boldsymbol{cx}$$

Viene sostituita da

$$Max\ Z = \boldsymbol{cx}\text{-}\boldsymbol{\lambda}(\boldsymbol{Ax} - \boldsymbol{b})$$

Il problema $\quad Max\ Z = \boldsymbol{cx}\text{-}\boldsymbol{\lambda}(\boldsymbol{Ax} - \boldsymbol{b})$

s.t. $x \geq 0$

È detto problema Lagrangiano e ha la proprietà che per ogni $\lambda \geq 0$ fissato la sua soluzione ottima costituisce un upper bound sull'ottimo del problema originario (lower bound se il problema è di Min).

Il bound ottenibile con il rilassamento Lagrangiano è quindi almeno tanto buono quanto quello ottenibile con il rilassamento lineare.

L'efficacia pratica del rilassamento Lagrangiano è legata all'efficienza con cui si riesce a risolvere il problema duale del problema Lagrangiano.

Non è però efficiente come il rilassamento LP nel calcolare il primo e il terzo dei test di fathoming che ricordiamo di seguito.

Il B&B può essere usato anche per trovare soluzioni buone Z (quasi-ottime). Esse hanno valori molto vicini a quelli di una soluzione ottima Z**.

(1 - ⍼)Z** <= Z

Dove ⍼ è una costante che indica la percentuale di approssimazione.
Quindi se una possibile soluzione incombente Z* soddisfa:

(1 - ⍼)Z** <= Z*

Allora si può terminare con Z* che è soluzione quasi-ottima.

Questo viene fatto se si vuole fare uno sforzo computazionale basso oppure se non si ha tempo per trovare una soluzione ottimale.

Immagina di voler trovare l'itinerario più veloce (ottimo) per andare a scuola:

- Z**: il tempo minimo possibile (20 minuti).
- Ti accontenti di arrivare in un tempo vicino a quello ottimo (ad esempio, con uno scarto massimo del 5%).
- Perciò accetti una soluzione quasi-ottima che richieda al massimo: (1 − 0.05) × 20 = 19 cioè 21 minuti. Se trovi un itinerario che richiede meno di 21 minuti, ti fermi.

Ma come facciamo a sapere che valore ha la soluzione ottima Z**
prima di risolvere il sistema? Possiamo usare la soluzione migliore trovata fino a quel momento.

Poiché Bound rappresenta un limite superiore valido per Z**, possiamo sostituire Z** con Bound nelle formule precedenti:

$(1 - \alpha)\ \text{Bound} \leq Z*$

Anche se la soluzione associata a Bound potrebbe non essere **ammissibile** (ad esempio, potrebbe violare vincoli di interezza), essa rimane un valido limite superiore per Z**. Quindi è utile per il calcolo e per prendere decisioni sul B&B.

Per problemi di grandi dimensioni, cercare la soluzione ottima Z** potrebbe essere computazionalmente proibitivo. Usare soluzioni quasi-ottime e il limite superiore Bound permette di risparmiare tempo e risorse.

Immagina di dover pianificare la consegna di pacchi in una città:

- Z**: costo minimo totale (non noto esattamente).
- Bound: un'approssimazione calcolata rilassando alcune restrizioni (esempio: ignorare alcune strade chiuse temporaneamente).
- Se trovi un percorso Z* che soddisfa: $(1 - \alpha)\ \text{Bound} \leq Z*$ puoi accettare il risultato e interrompere i calcoli, risparmiando tempo.

PROGRAMMAZIONE LINEARE MISTA

È molto simile a quella binaria eccetto che alcune variabili possono essere NON intere.
NON SO SE QUESTA ROBA VA MESSA NELLA PRATICA OPPURE NELLA TEORIA
Cosa cambia:

- Branching: quando scelgo una variabile e decido di sceglierla in base all'indice allora posso scegliere solo le variabili intere, non quelle NON intere. ***NON SO SE SIA GIUSTO***

- Invece che assegnare vincoli di 0 e 1, devo assegnare:

$$x_j \leq \lfloor x_j^* \rfloor \quad \text{e} \quad x_j \geq \lfloor x_j^* \rfloor + 1$$

se $x_j^* = \frac{7}{2}$ allora $\lfloor x_j^* \rfloor = 3$ e i due vincoli sono rispettivamente $x_j \leq 3$ e $x_j \geq 4$

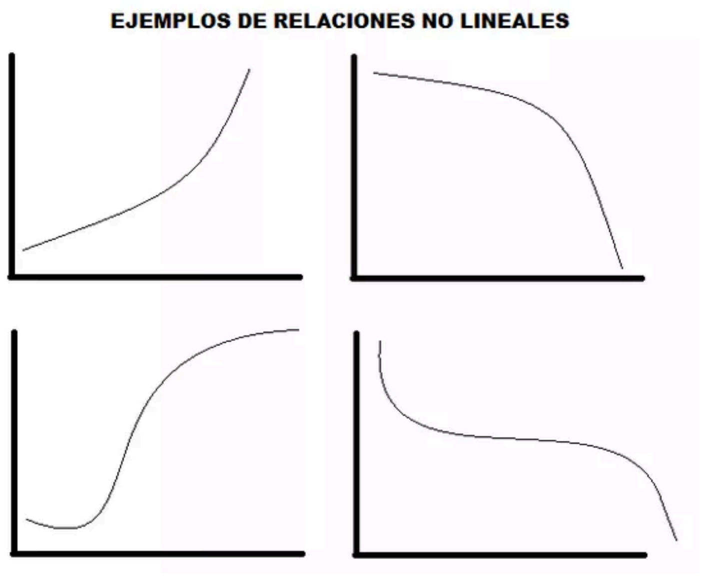
- Non posso più fissare le variabili, ciò significa che quando nella 1° fase trovo un valore ad x1 nella prossima fase devo ricalcolarlo di nuovo.
- Posso effettuare l'approssimazione del valore della funzione obiettivo SSE tutti i coefficienti sono interi.
- Il B&B si arresta solo quando le variabili che devono essere intere sono effettivamente intere. Le variabili continue non le consideriamo.

PROGRAMMAZIONE NON LINEARE

Vincoli e funzioni sono NON lineari. Ma cosa significa lineare?

- Lineare -> graficamente sono delle rette, le funzioni sono tipo: $3x_1 + 10x_2 + x_3$ …
- Non Lineare -> graficamente non sono delle rette. Una funzione non lineare può includere termini elevati a potenze maggiori di 1, radici, logaritmi, esponenziali, ecc.

Questo implica che se una funzione lineare ha una crescita costante e abbastanza prevedibile, una funzione non lineare ha una crescita che varia in base al tipo di funzione.



Ecco alcuni scenari in cui è possibile applicare la programmazione non lineare:

Elasticità del prezzo: la domanda di un prodotto può diminuire all'aumentare del prezzo.

Costi marginali variabili: il costo di un oggetto può diminuire, se la sua produzione aumenta, a causa dell'economia di scala.

Problemi di trasporto: possono essere disponibili sconti di volume per spedizioni di grandi dimensioni.

Classificazione: nel machine learning, la PNL può essere utilizzata per costruire classificatori che separano i punti di dati in diverse categorie minimizzando gli errori di classificazione.

Addestramento di reti neurali: l'addestramento di una rete neurale prevede la ricerca dei parametri ottimali che minimizzano l'errore di previsione, il che porta a un problema di ottimizzazione non lineare.

Ottimizzazione del portafoglio: la PNL può essere utilizzata per creare un portafoglio di investimenti che bilanci il rischio di un singolo investimento con il rendimento atteso di un altro.

Consideriamo un portafoglio composto da un mix di azioni scelte tra *n* possibilità.

Variabili Decisionali: x_j (j = 1 … n) rappresenta il numero di azioni del titolo j da includere nel portafoglio.
Parametri:

- μ_j: Media del rendimento su ciascuna azione del titolo j
- σ_{jj}: Varianza del rendimento su ciascuna azione del titolo j, che misura il rischio associato a quel titolo.
- σ_{ij}: Covarianza del rendimento dei titoli i e j (per i ≠ j).
- P_j: Prezzo di ogni azione del titolo j.
- B: Budget disponibile per l'investimento.
- L: Minimo rendimento accettabile per il portafoglio.

Funzioni Obiettivo e Vincoli:

$$R(x) = \sum_{j=1}^{n} \mu_j x_j \qquad \text{Rendimento totale del portafoglio}$$

$$V(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \qquad \text{Rischio totale del portafoglio}$$

L'obiettivo è massimizzare $R(x)$ minimizzando $V(x)$. Un approccio comune è quello di minimizzare $V(x)$ (rischio) con un vincolo sul rendimento minimo accettabile: **Min** $V(x)$

Con i seguenti vincoli:

$$\sum_{j=1}^{n} \mu_j x_j \geq L \qquad \text{Rendimento totale del portafoglio}$$

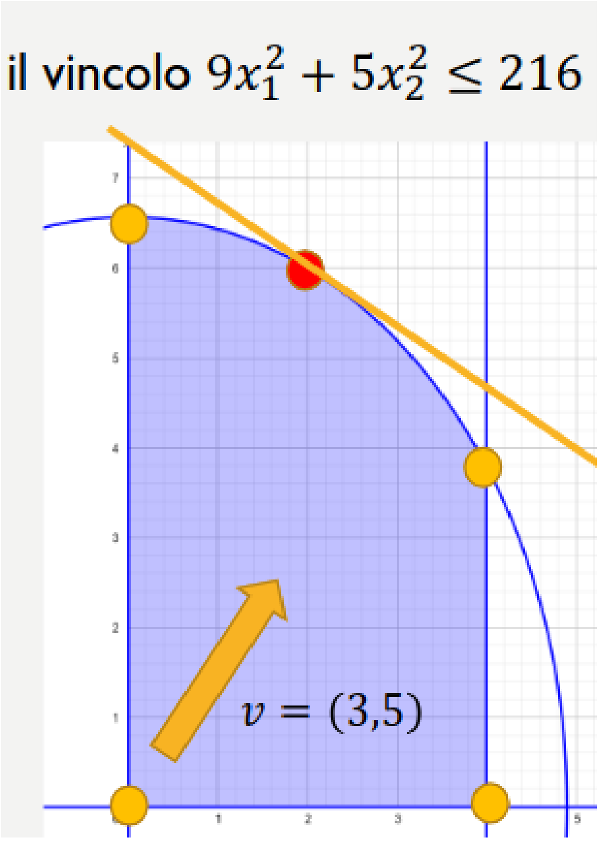
$$\sum_{j=1}^{n} P_j x_j \leq B$$

$$x_j \geq 0 \qquad per\ j = 1,2 \ldots n$$

I problemi di PNL possono essere classificati in base alle loro caratteristiche:

- **Ottimizzazione non vincolata**: non ci sono vincoli.
- **Ottimizzazione con vincoli lineari**: vincoli lineari, ma la funzione obiettivo è non lineare. Un esempio è la programmazione quadratica, come il modello di portafoglio di Markowitz.
- **Ottimizzazione convessa**: la funzione obiettivo è concava o convessa e tutti i vincoli sono convessi. Questo porta dei vantaggi in quanto esiste 1 solo punto di minimo/massimo.
- **Ottimizzazione non convessa**: i problemi che non soddisfano i requisiti di convessità. Questi sono più difficili da risolvere a causa della potenziale presenza di più punti di minimo/massimo locali.

Purtroppo si vengono a creare dei problemi:



La soluzione ottima di un problema di PNL non si trova più obbligatoriamente in un vertice della regione ammissibile.

Potrebbe essere all'interno della regione ammissibile oppure su un punto del confine (che non è un vertice).

Di conseguenza, il **metodo del simplesso** non può essere applicato direttamente ai problemi di PNL.

PNL AD 1 VARIABILE

Si risolvono in 2 modi diversi:

- **Algoritmi dicotomici:** come il metodo della bisezione
- **Algoritmi di approssimazione:** come il metodo di Newton

Metodo della Bisezione

L'idea è schiacciare/approssimare la funzione fino a trovare il punto di ottimo. Si può applicare solo se la funzione è concava/convessa, continua e derivabile.

Nel **Machine Learning**, molti algoritmi lavorano **localmente**, il che significa che possono trovare un **minimo locale** (o massimo locale), ma non garantiscono di trovare il **minimo globale** della funzione obiettivo. Questo è particolarmente importante in contesti complessi come il training di modelli di rete neurale, dove la funzione di perdita è non lineare e presenta molti minimi locali.

Il **metodo della bisezione** si applica principalmente a funzioni **monotone** o su intervalli ristretti per trovare zeri o punti critici. Nel contesto del machine learning, non è il metodo principale per evitare problemi di ottimi locali, ma offre un'idea chiara del problema: **individuare intervalli ben definiti per migliorare il risultato.**

Perché gli algoritmi si bloccano nei minimi locali?

- Gli algoritmi di ottimizzazione (come il **gradiente discesa**) cercano il minimo seguendo la pendenza locale.
- Il gradiente discesa o metodi simili si basano solo su informazioni locali, il che può portare a fermarsi in minimi che non sono i più bassi possibili.
- Un cattivo punto iniziale può dirigere l'algoritmo verso un minimo locale subottimale.

Vantaggi
- converge sempre
- richiede solo la derivata prima

Come svantaggio però è un algoritmo molto lento.

Algoritmo di Newton

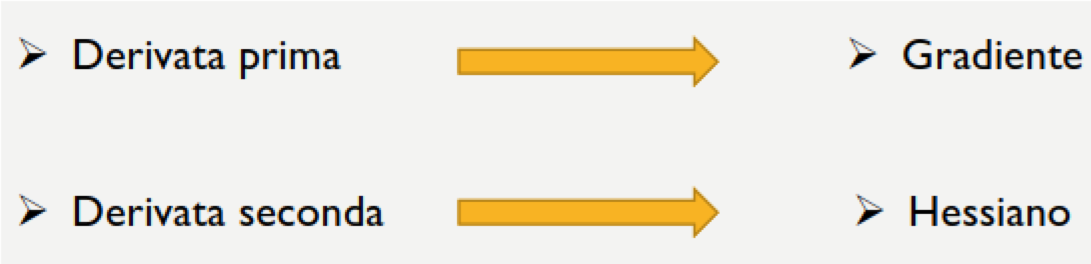
Ad ogni c'è un modo per aumentare la velocità di convergenza, infatti posso usare la formula di Taylor.

Approssimare una funzione in un punto specifico (xk + 1). La funzione quadratica ottenuta dalla serie di taylor presenta xk+1 come variabile.

Newton ha il vantaggio di avere la convergenza quadratica (molto veloce).
Svantaggi
- richiede il calcolo della derivata seconda, che potrebbe essere ostica
- la sua convergenza globale è scarsa, specie se xk è molto lontano dal punto di ottimo. Infatti se usiamo Taylor la funzione sarà "corretta" solo in punto.

Nella pratica newton va usato con strategie di ottimizzazione globale. Si comporta bene con poche variabili.

PNL CON PIÙ VARIABILI



Il **gradiente** è un vettore che contiene le derivate prime <u>parziali</u> della funzione. Indica la direzione di crescita della funzione (alto, basso).

Se il gradiente è zero, sono in un minimo/massimo/punto di sella

La **Hessiana** è una matrice quadrata che contiene le derivate seconde della funzione. Indica la curvatura di una funzione.

PNL VINCOLATA

Se ho dei vincoli nella funzione allora le strategie che abbiamo usato non vanno più bene, perché se il punto stazionario che troviamo è fuori dalla regione ammissibile siamo fottuti. Ci sono 3 possibile approcci:

DIMENSIONALITY REDUCTION

In poche parole risolvi *letteralmente* la funzione.
Non è sempre possibile usare sempre questo metodo, anzi quasi mai.

Consideriamo il seguente problema di ottimizzazione non lineare vincolato

$$\min\ (x_1 - 2)^2 + 2(x_2 - 1)^2$$

soggetto al seguente vincolo

$$x_1 + 4x_2 = 3$$

Possiamo esplicitare la variabile x_1 nel vincolo in funzione di x_2

$$x_1 = 3 - 4x_2$$

sostituire nella funzione obiettivo ottenendo un problema di ottimizzazione in una variabile

$$\min\ (3 - 4x_2 - 2)^2 + 2(x_2 - 1)^2 \Rightarrow \min\ (1 - 4x_2)^2 + 2(x_2 - 1)^2$$

$$\Rightarrow \min 18x_2^2 - 12x_2 + 3 \Rightarrow x_2 = \frac{1}{3}$$

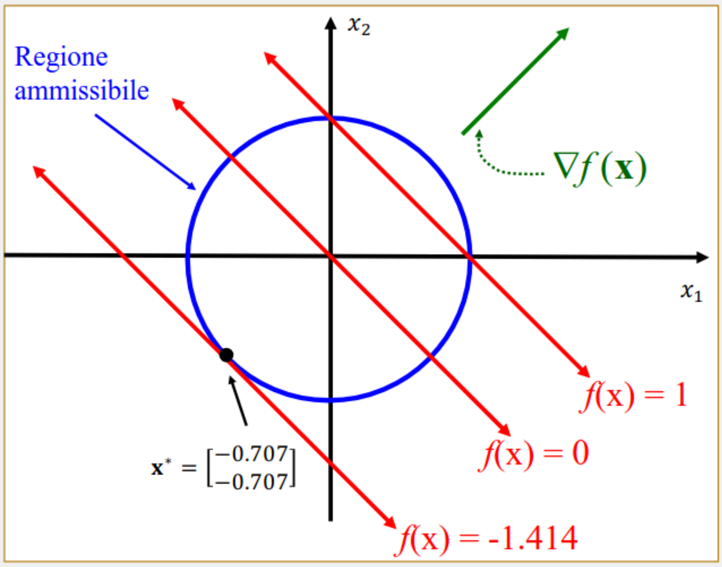
Da cui possiamo ottenere anche il valore di x_1

$$x_1 = 3 - 4x_2 \Rightarrow x_1 = \frac{5}{3}$$

Grafico

$$\min x_1 + x_2$$
$$x_1^2 + x_2^2 = 1$$

- La regione ammissibile è un cerchio di raggio unitario
- Le curve di livello della funzione obiettivo sono rette con pendenza -1
- Il punto di minimo corrisponderà a quel punto del cerchio tangente alla curva di livello avente valore più basso



Il $\nabla f(x)$, come abbiamo sempre visto, è costante.
Mentre se faccio $\nabla g(x)$ il suo risultato è variabile ed in questo caso dipende da x.
$\nabla g(x)$ punta nella stessa direzione MA opposta a quella di $\nabla f(x)$ nel punto di minimo del problema: $\nabla f\ x* = -\nabla g(x*)$

MOLTIPLICATORI DI LAGRANGE

Serve per disfarsi di vincoli scomodi. È praticamente la stessa cosa della DUALITÀ. È praticamente il KKT ma con solo i vincoli di uguaglianza.

Il punto di ottimo della lagrangiana mi da lo stesso punto di ottimo del problema originale.

$$\min x_1 + x_2$$

soggetto ai seguenti vincoli

$$x_1^2 + x_2^2 = 1$$

$$\Longrightarrow L(x_1, x_2, \lambda_1) = x_1 + x_2 + \lambda_1(x_1^2 + x_2^2 - 1)$$

$\lambda 1$ perché abbiamo solo un vincolo, altrimenti avremmo dovuto sommare anche $\lambda 2$, $\lambda 3$…

Ora diventa un PNL non vincolata, in quanto non ci sono più vincoli. Proviamo ad usare il gradiente per trovare dei punti stazionari.

$$1 + 2\lambda_1 x_1 = 0$$

$$1 + 2\lambda_1 x_2 = 0$$

$$(x_1^2 + x_2^2 - 1) = 0 \Longrightarrow 2x_1^2 = 1$$

Ora risolviamo 2x^2 = 1

Da cui otteniamo due punti stazionari

$\mathbf{x}^* =$

$$A = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \text{ con } \lambda_1 = -\frac{\sqrt{2}}{2} \qquad B = \left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right) \text{ con } \lambda_1 = \frac{\sqrt{2}}{2}$$

Punto A = massimo
Punto B = minimo

Il problema di questo metodo è che non dice se è un punto di massimo/minimo, ma solo se è un punto stazionario.

È necessario avere i requisiti necessari per determinare i punti critici e la natura (massimo, minimo o sella) di una soluzione.

CONDIZIONI DEL PRIMO ORDINE (non li chiedono, spero)
Ogni punto stazionario "vero" della funzione è un punto stazionario della lagrangiana.
Quindi durante il calcolo della lagrangiana, quando trovo i suoi stazionari, potrebbero essere stazionari anche della funzione.

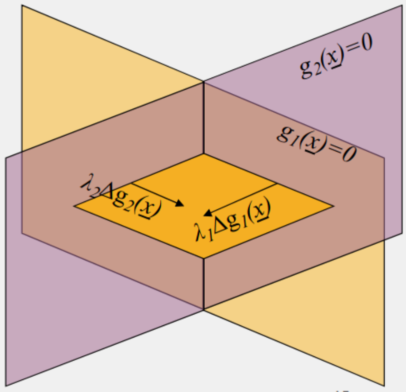
- L'insieme dei vincoli di fatto restringe lo spazio delle soluzioni ammissibili del problema al sottospazio intersezione dato dell'intersezione dei vari vincoli.
- Il gradiente della funzione $\nabla f(\boldsymbol{x}^*)$ nel punto di ottimo $\boldsymbol{x}^*$ deve risultare ortogonale a tale sottospazio
- Il vettore ortogonale a tale sotto-spazio è dato da

$$\sum_{i=1}^{m} \lambda_i^* \cdot \nabla g_i(\boldsymbol{x}^*)$$

- Dobbiamo quindi avere che

$$\nabla f(\boldsymbol{x}^*) = -\sum_{i=1}^{m} \lambda_i^* \cdot \nabla g_i(\boldsymbol{x}^*)$$

- Quest'ultima condizione significa che nel punto di ottimo il gradiente della funzione f può essere riscritto come combinazione lineare dei gradienti dei vincoli



15

CONDIZIONI DEL SECONDO ORDINE
Controllano il valore di matrice hessiana della funzione lagrangiana sulle variabili iniziali.
Cosa cazzo è y?
Sono i vettori ortogonali ai gradienti dei vertici.

CONDIZIONI DI KKT

Le **condizioni di Karush-Kuhn-Tucker (KKT)** sono un insieme di condizioni che caratterizzano i punti di ottimo per PNL Vincolata.

NON sono un algoritmo. Sono una generalizzazione del metodo dei moltiplicatori di Lagrange e si applicano a problemi con vincoli di uguaglianza e disuguaglianza.

- vincoli di = "tolgono dimensioni", limitando a spazi (n-k)-dimensionali, soluzioni di funzioni n-dimensionali
- vincoli di disuguaglianza restringono l'area in cui cercare, ma l'area rimane n-dimensionale

CONDIZIONI DI QUALIFICAZIONE

Prima di fare il KKT verifico la regolarità dei vincoli (basta verificarne uno):

- indipendenza lineare dei vincoli -> il gradiente della disuguaglianza e gradiente della uguaglianza devono essere linearmente indipendenti (det != 0)
- Slater Condition -> Il problema deve essere convesso.
- Vincoli lineari -> se tutti i vincoli sono funzioni affini, allora la condizione di regolarità è sempre verificata in x

Dunque:
Le condizioni KKT sono necessarie ma non sufficienti per trovare un punto di ottimo.
Ovvero troviamo un punto stazionario, ma non sappiamo se è minimo o massimo

Si suppone che $f\colon \mathbb{R}^n \to \mathbb{R}$ e che $g_i\colon \mathbb{R}^n \to \mathbb{R}$ e $h_j\colon \mathbb{R}^n \to \mathbb{R}$. Inoltre si suppone che esse siano funzioni continuamente differenziabili nell'intorno del punto x^*. Se x^* è un punto di minimo locale che soddisfa le condizioni di regolarità dei vincoli, allora esistono dei moltiplicatori μ_j, con $j = 1, \dots, l$, e λ_i, con $i = 1, \dots, m$, tali che:

1. $\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{l} \mu_j \nabla h_j(x^*) = 0;$
2. $\forall i \in \{1, \dots, m\} : g_i(x^*) \leq 0;$
3. $\forall j \in \{1, \dots, l\} : h_j(x^*) = 0;$
4. $\forall i \in \{1, \dots, m\} : \lambda_i \geq 0;$
5. $\forall i \in \{1, \dots, m\} : \lambda_i g_i(x^*) = 0.$

PRIMA CONDIZIONE
Condizione di annullamento del gradiente della funzione lagrangiana.
SECONDA E TERZA CONDIZIONE
Sono i vincoli di ammissibilità del punto.
QUARTA CONDIZIONE
Condizione di non negatività del moltiplicatore associato ai vincoli di disuguaglianza.
QUINTA CONDIZIONE - CONDIZIONE DI COMPLEMENTARIETÀ
Il moltiplicatore di un vincolo inattivo deve essere nullo.

- Alla fine Se è il PNL è CONVESSO -> allora è minimo globale
- Altrimenti devo verificare con l'hessiana lagrangiana se sono punti stazionari (massimo/minimo). Uso le condizioni del secondo ordine.

Se il vincolo di uguaglianza fosse una retta, i punti di massimo o minimo potrebbero essere solo su quella retta