

# **COMPLEMENTI DI RICERCA OPERATIVA**

Prof. Marco Trubian  
6 CFU

**Luca Cappelletti**

Lecture Notes  
Year 2017/2018



Magistrale Informatica  
Università di Milano  
Italy  
5 maggio 2018

# Indice

<b>I</b>	<b>Programmazione non lineare</b>	<b>3</b>
<b>1</b>	<b>Alcune definizioni base</b>	<b>4</b>
1.1	La convessità . . . . .	4
1.2	Massimi e minimi locali . . . . .	4
<b>2</b>	<b>Ottimizzazione non vincolata</b>	<b>5</b>
2.1	Condizioni necessarie di ottimalità del primo ordine . . . . .	5
2.2	Condizioni necessarie di ottimalità del secondo ordine . . . . .	5
2.3	Condizioni necessarie di ottimalità in senso stretto del secondo ordine . . . . .	5
<b>3</b>	<b>Programmazione quadratica</b>	<b>6</b>
<b>4</b>	<b>Convergenza</b>	<b>7</b>
4.1	Algoritmo convergente localmente e globalmente . . . . .	7
4.1.1	Convergente localmente . . . . .	7
4.1.2	Convergente globalmente . . . . .	7
4.2	Velocità di convergenza . . . . .	7
4.2.1	Q-lineare . . . . .	7
4.2.2	Q-superlineare . . . . .	7
4.2.3	Q-quadratica . . . . .	7
<b>5</b>	<b>Metodi di ottimizzazione continua</b>	<b>8</b>
5.1	Condizioni di Wolfe . . . . .	8
5.2	Metodo di Armijo per stabilire la stepsize . . . . .	8
5.3	Convergenza di metodi LS approssimati . . . . .	8
<b>6</b>	<b>Metodi di ottimizzazione</b>	<b>10</b>
6.1	Metodi a discesa rapida . . . . .	10
6.2	Metodi Newton . . . . .	10
6.2.1	Metodi Newton modificati . . . . .	11
6.3	Metodi Quasi-Newton . . . . .	11
6.4	Metodi del gradiente coniugato . . . . .	11
6.5	Metodi Trust Region . . . . .	12
6.5.1	Metodo zampa di cane . . . . .	13
6.5.2	Proprietà della matrice B . . . . .	13
<b>7</b>	<b>Ottimizzazione vincolata</b>	<b>14</b>
7.1	Condizioni di ottimalità del secondo ordine . . . . .	15
7.2	Modello quadratico con vincoli lineari . . . . .	15
7.2.1	Metodo dell'insieme attivo primale per problemi quadratici convessi (Primal Active set) . . . . .	15
<b>8</b>	<b>Algoritmi</b>	<b>17</b>
8.1	Metodo della penalità quadratica . . . . .	17
8.2	Metodo delle barriere . . . . .	17
8.3	Metodo della proiezione del gradiente . . . . .	18
8.3.1	Caso con vincoli generici non lineari . . . . .	18
8.4	Augmented lagrangean method . . . . .	19
8.5	Sequential quadratic programming (SQP) . . . . .	19

<b>II Programmazione lineare intera</b>	<b>21</b>
<b>9 TSP simmetrico</b>	<b>22</b>
9.1 Problema di separazione . . . . .	22
9.2 Albero di Gomory-Hu . . . . .	23
9.3 Comb inequalities (Disuguaglianze a pettine) . . . . .	23
<b>10 Cover Inequalities</b>	<b>25</b>
10.1 Irrobustire una cover inequality . . . . .	25
10.2 Sollevare le cover inequalities . . . . .	25
10.3 Separazione delle cover inequalities . . . . .	26
<b>11 Dualità lagrangiana</b>	<b>27</b>
11.1 Rilassamento lagrangiano . . . . .	27
11.2 Risolvere il lagrangiano duale . . . . .	28
<b>12 Tecnica di generazione delle colonne</b>	<b>29</b>
12.1 Cutting stock . . . . .	29
12.1.1 Formulazione classica . . . . .	29
12.1.2 Formulazione estesa . . . . .	29
12.2 Master Problem . . . . .	30
12.3 Master Problem ridotto . . . . .	30
12.4 Problema di Pricing . . . . .	30

## **Parte I**

# **Programmazione non lineare**

## Alcune definizioni base

### 1.1 La convessità

**Definizione 1.1.1 (Insieme convesso).** Un insieme  $X \subset \mathbb{R}^n$  è convesso se comunque presi due punti  $\underline{x}, \underline{y} \in X$ , allora  $\lambda \underline{x} + (1 - \lambda) \underline{y} \in X$ , per ogni  $\lambda \in [0, 1]$ .

La proprietà di convessità è invariante rispetto alle operazioni di moltiplicazione con uno scalare, unione e intersezione con un altro insieme convesso.

**Definizione 1.1.2 (Funzione convessa).** Una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  è convessa se il suo dominio è un insieme convesso  $X \subseteq \mathbb{R}^n$  e comunque presi due punti  $\underline{x}, \underline{y} \in X$  vale la relazione:

$$f(\lambda \underline{x} + (1 - \lambda) \underline{y}) \leq \lambda f(\underline{x}) + (1 - \lambda) f(\underline{y}) \quad \forall \lambda \in [0, 1]$$

La proprietà di convessità è invariante rispetto a moltiplicazione con uno scalare e somma tra funzioni convesse.

Vale inoltre che la funzione max di una o più funzioni convesse e che il luogo dei punti  $\underline{x}$  per i quali vale che  $f(\underline{x}) \leq \alpha$  è convesso.

**Definizione 1.1.3 (Problema convesso).** Un problema di ottimizzazione con funzione obiettivo e regione ammissibile entrambe convesse viene detto problema convesso.

### 1.2 Massimi e minimi locali

**Definizione 1.2.1 (Minimo globale).** Un punto  $\underline{x}^* \in X$  è un punto di minimo globale di  $f(\underline{x})$  se:

$$f(\underline{x}^*) \leq f(\underline{x}) \quad \forall \underline{x} \in X$$

**Definizione 1.2.2 (Minimo locale).** Un punto  $\underline{x}^* \in X$  è un punto di minimo locale di  $f(\underline{x})$  se esiste un intorno aperto  $I(\underline{x}^*, \epsilon)$  di  $\underline{x}^*$  avente raggio  $\epsilon > 0$  tale che:

$$f(\underline{x}^*) \leq f(\underline{x}) \quad \forall \underline{x} \in X \cap I(\underline{x}^*, \epsilon)$$

# 2

## Ottimizzazione non vincolata

**Definizione 2.0.1 (Direzione di discesa).** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , un vettore  $\underline{d} \in \mathbb{R}^n$  si dice direzione di discesa per  $f$  in  $\underline{x}$  se:

$$\exists \lambda > 0 : f(\underline{x} + \lambda \underline{d}) < f(\underline{x})$$

**Definizione 2.0.2 (Derivata direzionale).** Sia data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , un vettore  $\underline{d} \in \mathbb{R}^n$  e un punto dove  $f$  è definita. Se esiste il limite:

$$\lim_{\lambda \rightarrow 0^+} \frac{f(\underline{x} + \lambda \underline{d}) - f(\underline{x})}{\lambda}$$

allora tale limite prende il nome di derivata direzionale della funzione  $f$  nel punto  $\underline{x}$  lungo la direzione  $\underline{d}$

### 2.1 Condizioni necessarie di ottimalità del primo ordine

**Teorema 2.1.1 (Condizioni necessarie di ottimalità del primo ordine).** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , derivabile in  $\underline{x}^* \in \mathbb{R}^n$ , condizione necessaria affinché il punto  $\underline{x}^*$  sia un minimo locale per  $f$  è che il gradiente della funzione calcolato in  $\underline{x}^*$  sia nullo.

### 2.2 Condizioni necessarie di ottimalità del secondo ordine

**Teorema 2.2.1 (Condizioni necessarie di ottimalità del secondo ordine).** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  di classe  $\underline{C}^2(\underline{x}^*)$ , condizione necessaria affinché il punto  $\underline{x}^*$  sia un minimo locale per  $f$  è che il gradiente della funzione calcolato in  $\underline{x}^*$  sia nullo e che valga la relazione seguente:

$$\underline{d}^T H(\underline{x}^*) \underline{d} \geq 0 \forall \underline{d} \in \mathbb{R}^n$$

Cioè l'hessiana è definita come **semipositiva**.

### 2.3 Condizioni necessarie di ottimalità in senso stretto del secondo ordine

**Teorema 2.3.1 (Condizioni necessarie di ottimalità del secondo ordine).** Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  di classe  $\underline{C}^2(\underline{x}^*)$ , condizione necessaria affinché il punto  $\underline{x}^*$  sia un minimo locale in **senso stretto** per  $f$  è che il gradiente della funzione calcolato in  $\underline{x}^*$  sia nullo e che valga la relazione seguente:

$$\underline{d}^T H(\underline{x}^*) \underline{d} > 0 \forall \underline{d} \in \mathbb{R}^n$$

Cioè l'hessiana è definita come **positiva**.

# 3

## Programmazione quadratica

Nella programmazione quadratica si approssima  $f$  con il seguente modello quadratico:

$$\min f(\underline{x}) = \frac{1}{2} \underline{x}^T Q \underline{x} + \underline{b}^T \underline{x}$$

Esistono pochi casi possibili:

**$Q$  non è semidefinita positiva**  $f$  non ha un punto di minimo.

**$Q$  è definita positiva**  $\underline{x}^* = Q^{-1} \underline{b}$  è l'unico minimo globale.

**$Q$  è definita semi-positiva**  **$Q$  non è singolare**  $\underline{x}^* = Q^{-1} \underline{b}$  è l'unico minimo globale.

**$Q$  è singolare** non esiste una soluzione o esistono infinite soluzioni.

# 4

## Convergenza

Ovviamente un algoritmo è buono se converge rapidamente.

### 4.1 Algoritmo convergente localmente e globalmente

#### 4.1.1 Convergente localmente

Un algoritmo è convergente localmente quando converge solo se il punto da cui parte è in un intorno del punto ottimo.

#### 4.1.2 Convergente globalmente

Un algoritmo è convergente globalmente quando converge partendo da qualsiasi punto del dominio.

### 4.2 Velocità di convergenza

#### 4.2.1 Q-lineare

Quando il rapporto tra lo step  $k$  e lo step  $k + 1$  mantiene un valore costante.

#### 4.2.2 Q-superlineare

Quando il rapporto tra lo step  $k$  e lo step  $k + 1$  all'infinito è nullo.

#### 4.2.3 Q-quadratica

Quando il rapporto tra lo step  $k$  quadro e lo step  $k + 1$  mantiene un valore costante.



# 5

## Metodi di ottimizzazione continua

Si tratta del problema di determinare a partire dallo step  $k$  lo step  $k + 1$  in modo tale da avvicinarsi all'ottimo della funzione.

Esistono due strategie:

**Line search** Si determina una direzione e quindi si stabilisce una distanza con cui muoversi in questa direzione.

**Trust region** Si determina un raggio (una distanza) di confidenza e quindi si stabilisce una direzione nel limite di questa circonferenza verso cui muoversi.

Esiste chiaramente un tradeoff tra velocità e precisione nello stabilire la distanza con cui muoversi.

### 5.1 Condizioni di Wolfe

Per essere efficace, la line-search approssimata richiede che siano rispettate le condizioni di Wolfe. Esse sono due, una di **decremento sufficiente** ed una di **curvatura**, dove i coefficienti  $c$  sono tali che  $0 < c_1 < c_2 < 1$

$$\begin{aligned} f(\underline{x} + \alpha \underline{d}) &\leq f(\underline{x}) + c_1 \alpha \nabla f(\underline{x})^T \underline{d} \\ \phi(\alpha) &\leq \phi(0) + \alpha c_1 \phi'(0) \end{aligned}$$

(a) Condizione di decremento sufficiente

$$\begin{aligned} \nabla f(\underline{x} + \alpha \underline{d})^T \underline{d} &\geq c_2 \alpha \nabla f(\underline{x})^T \underline{d} \\ \phi'(\alpha) &\geq c_2 \phi'(0) \end{aligned}$$

(b) Condizione di curvatura

Figura 5.1: Condizioni di Wolfe

Le condizioni di Wolfe forti introducono un vincolo di segno sulla curvatura (valore assoluto).

### 5.2 Metodo di Armijo per stabilire la stepsize

Si itera riducendo gradualmente la distanza di un fattore  $\sigma$ , usualmente pari circa a 0.9 sino a che il valore dello step successivo è più vicino all'ottimo dello step corrente. Nei metodi Newton e quasi Newton il coefficiente della distanza è inizializzato usualmente a 1.

### 5.3 Convergenza di metodi LS approssimati

Se definiamo  $\theta_k$  come l'angolo tra  $\underline{d}_k$  e  $-\nabla f_k$ , allora possiamo determinare il coseno dell'angolo come:

$$\cos \theta_k = \frac{-\nabla f(\underline{x})_k^T \underline{d}_k}{\|\nabla f(\underline{x})_k\| \cdot \|\underline{d}_k\|}$$

**Teorema 5.3.1 (Convergenza dei metodi di ricerca lineare approssimata).** Sia  $\underline{d}_k$  una direzione di discesa e sia  $\alpha_k$  una distanza che rispetti le condizioni di Wolfe. Sia inoltre  $f$  una funzione limitata inferiormente su  $\mathbb{R}^n$ , **differenziabile continuamente** sull'insieme  $M$  che contiene  $L_f = \{\underline{x} : f(\underline{x}) \leq f(\underline{x}_0)\}$ , dove  $\underline{x}_0$  è il punto di inizio. Assumiamo inoltre che  $\nabla f$  sia **lipschitziana** sull'insieme  $M$ . Allora:

$$\sum_{j=0}^{\infty} \cos^2 \theta_j \|\nabla f(\underline{x}_j)\|^2 \leq \infty$$

*Dimostrazione.* Essendo valida la **condizione di curvatura**, allora vale la disequazione:

$$\nabla f_{k+1}^T \underline{d}_k \geq c_2 \nabla f_k^T \underline{d}_k$$

Sottraggo ad ambo i lati  $\nabla f_k \underline{d}_k$  ed ottengo:

$$(\nabla f_{k+1}^T - \nabla f_k^T) \underline{d}_k \geq (c_2 - 1) \nabla f_k^T \underline{d}_k$$

Siccome il gradiente della funzione è **lipschitziano** vale la disequazione:

$$(\nabla f_{k+1}^T - \nabla f_k^T) \underline{d}_k \leq \|\nabla f_{k+1} - \nabla f_k\| \|\underline{d}_k\| \leq L \|\underline{x}_{k+1} - \underline{x}_k\| \|\underline{d}_k\| = \alpha_k L \|\underline{d}_k\|^2$$

Da questa disequazione ricaviamo il valore di  $\alpha_k$ :

$$\alpha_k \geq \frac{c_2 - 1}{L} \frac{\nabla f_k^T \underline{d}_k}{\|\underline{d}_k\|^2}$$

Essendo quindi valida la **condizione di decremento sufficiente** vale la disequazione:

$$f_{k+1} \leq f_k + c_1 \alpha_k \underline{d}_k^T \nabla f_k$$

Sostituendo il valore di  $\alpha_k$  ottenuto al passaggio precedente ottengo:

$$f_{k+1} \leq f_k - c_1 \frac{1 - c_2}{L} \frac{(\nabla f_k^T \underline{d}_k)^2}{\|\underline{d}_k\|^2}$$

Pongo  $c = c_1 \frac{1 - c_2}{L}$  ed ottengo:

$$f_{k+1} \leq f_k - c \frac{(\nabla f_k^T \underline{d}_k)^2}{\|\underline{d}_k\|^2}$$

Sostituisco  $\cos^2 \theta_k = \frac{(-\nabla f(\underline{x})^T \underline{d}_k)^2}{\|\nabla f(\underline{x})\|^2 \|\underline{d}_k\|^2}$  ed ottengo:

$$f_{k+1} \leq f_k - c \cos^2 \theta_k \|\nabla f_k\|^2$$

Per ricorsione ottengo la sommatoria:

$$f_{k+1} \leq f_0 - c \sum_{j=0}^k \cos^2 \theta_j \|\nabla f_j\|^2$$

Siccome la funzione  $f$  è limitata inferiormente si ottiene la **condizione di Zoutendijk**:

$$c \sum_{j=0}^k \cos^2 \theta_j \|\nabla f_j\|^2 \leq f_0 - f_{k+1} < \infty$$

Questo implica che:

$$\cos^2 \theta_j \|\nabla f_j\|^2 \rightarrow 0$$

Quindi se l'algoritmo soddisfa anche la **condizione angolare** (cioè sceglie una direzione di discesa che la rispetta)  $\cos \theta_k \geq \epsilon > 0$  allora **converge**:

$$\lim_{k \rightarrow \infty} \|\nabla f(\underline{x}_k)\| = 0$$

□

## Metodi di ottimizzazione

### 6.1 Metodi a discesa rapida

Son metodi che usano come funzione di aggiornamento  $\underline{x}_{k+1} = \underline{x}_k - \alpha_k \nabla f_k$ , in cui le direzioni sono ortogonali al contorno della funzione. Non dovendo calcolare l'hessiana lo sforzo computazionale non è eccessivo e converge globalmente, ma estremamente piano quando una funzione è patologica.

**Teorema 6.1.1 (Velocità di convergenza locale dei metodi a discesa rapida).** Data una matrice  $Q$  definita positiva, la seguente relazione vale  $\forall \underline{x} \in \mathbb{R}^n$ :

$$\frac{(\underline{x}^T \underline{x})^2}{(\underline{x}^T Q \underline{x})(\underline{x}^T Q^{-1} \underline{x})} \geq \frac{4\lambda_{\min}\lambda_{\max}}{(\lambda_{\min} + \lambda_{\max})^2}$$

Dove  $\lambda_{\min}$  e  $\lambda_{\max}$  son gli autovalori minimo e massimo di  $Q$ .

Ne segue che la velocità di convergenza dei metodi a discesa rapida è **lineare** per i modelli quadratici.

$$\frac{\|\underline{x}_{k+1} - \underline{x}^*\|_Q}{\|\underline{x}_k - \underline{x}^*\|_Q} \leq \frac{(\lambda_{\max} - \lambda_{\min})}{(\lambda_{\max} + \lambda_{\min})}$$

### 6.2 Metodi Newton

Prendendo in considerazione l'approssimazione di Taylor fermata al secondo ordine, quindi con l'hessiana, otteniamo come direzione di discesa quella che minimizza  $\nabla f^T \underline{d} + \frac{1}{2} \underline{d}^T H \underline{d}$ , cioè  $\underline{d} = -(H)^{-1} \nabla f$ . Quando  $H$  è **definita positiva** vale che:

$$\nabla f^T \underline{d} = -\underline{d}^T H \underline{d} \leq -\sigma \|\underline{d}\|^2$$

Cioè quando  $H$  è **definita positiva** la direzione di Newton è la **direzione di discesa**.

Nei **modelli quadratici** con  $Q$  **definita positiva** il metodo di Newton converge in un'iterazione, altrimenti non converge. Su funzioni generiche, la qualità della direzione dipende da quanto è definita positiva la matrice hessiana.

**Teorema 6.2.1 (Convergenza dei metodi di Newton).** Sia  $f \in C^2$  e sia  $H(x)$  continuamente lipschitziana in un intorno del punto ottimo  $\underline{x}^*$ . Si assuma che valga  $\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k$ . Allora:

1. Se  $\underline{x}_0$  è sufficientemente vicino a  $\underline{x}^*$ , allora  $\{\underline{x}_k\} \rightarrow \underline{x}^*$
2.  $\{\underline{x}_k\}$  converge **quadraticamente**

3.  $\{\|\nabla f(\underline{x}_k)\|\}$  converge quadraticamente a zero.

I metodi Newton sono **convergenti localmente**.

La complessità computazionale è di  $O(n^3)$

### 6.2.1 Metodi Newton modificati

Sono metodi che modificano l'Hessiano, o rendendo la matrice positiva o scegliendo una direzione di discesa tramite metodi di discesa rapida quando necessario.

## 6.3 Metodi Quasi-Newton

Sono metodi in cui viene utilizzata un'approssimazione dell'Hessiano, che è computazionalmente costoso da calcolare. Viene utilizzata al suo posto una matrice chiamata  $G_k$  al posto di  $H_k^{-1}$ , e quindi calcolano la direzione di discesa come  $\underline{d}_k = -G_k \nabla f(\underline{x}_k)$ .

**Definizione 6.3.1 (Relazione secante).** Definendo  $\underline{p}_k = \nabla f(\underline{x}_{k+1}) - \nabla f(\underline{x}_k)$  possiamo definire la relazione secante:

$$H(\underline{x}_k) \underline{h}_k \approx \underline{p}_k \quad \text{or} \quad H(\underline{x}_k)^{-1} \underline{p}_k \approx \underline{h}_k$$

Inizializzando  $G_0 = I$ , possiamo imporre che ad ogni iterazione la matrice  $G_{k+1}$  debba soddisfare la relazione secante con la seguente uguaglianza:

$$G_{k+1} \underline{p}_k = \underline{h}_k$$

La realizzazione specifica di come si ottiene  $G_{k+1}$  partendo da  $G_k$  varia dai differenti metodi Quasi-Newton. Questi metodi impongono che  $G_k = G_k^T$  e che  $G_{k+1} - G_k$  abbia un rango basso.

Ne esistono di due categorie:

1. Matrice a rango unitario simmetrico o SR1.
2. A rango due:
  - (a) DFP
  - (b) BFGS

I **metodi a rango due** hanno alcune proprietà interessanti:

1. La matrice  $G_k$  converge a  $H(\underline{x}_k)^{-1}$  sui modelli quadratici.
2. Se  $G_0$  è definita positiva allora tutte le  $G_k$  sono definite positive.
3. Il costo computazionale è di  $O(n^2)$  in ogni iterazione.
4. La velocità di convergenza è **superlineare**.
5. In particolare il metodo BFGS garantisce convergenza globale se lo step-size rispetta le condizioni di Wolfe.

Con **Famiglia di Broyden** si intende una famiglia di formule ottenute tramite la combinazione convessa dell'aggiornamento DFP e BFGS.

## 6.4 Metodi del gradiente coniugato

Si tratta di una delle tecniche più utili per risolvere sistemi lineari di grandi dimensioni e può essere adattata per risolvere problemi di ottimizzazione non lineare. Si tratta di un metodo iterativo che risolve sistemi della forma  $A\underline{x} = \underline{b}$  dove  $A$  è una matrice definita positiva che è  $A = A^T$ .

**Definizione 6.4.1 (Vettori coniugati).** Un insieme di vettori  $\underline{p}_0, \underline{p}_1, \dots, \underline{p}_h$  è detto **coniugato** rispetto ad una matrice  $A$  se vale la relazione:

$$\underline{p}_i^T A \underline{p}_j = 0 \quad \forall i \neq j$$

Dato un punto di inizio  $\underline{x}_0$  e le **direzioni coniugate**  $\underline{p}_0, \dots, \underline{p}_{n-1}$  generiamo la sequenza  $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$  dove  $\alpha_k$  è un minimizzatore monodimensionale ed è dato da:

$$\alpha_k = -\frac{r_k^T \underline{p}_k}{\underline{p}_k^T A \underline{p}_k}$$

**Teorema 6.4.2.** Convergenza di sequenza con vettori coniugati Dato un qualsiasi  $\underline{x}_0$ , la sequenza generata da  $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{p}_k$  converge a  $\underline{x}^*$  in al più  $n$  iterazioni.

Il metodo del gradiente coniugato è un metodo a direzione coniugata con una proprietà importante:  $\underline{p}_k$  può essere ottenuta sapendo solo  $\underline{p}_{k-1}$  e  $\underline{p}_k$  risulta coniugato a tutte le direzioni precedenti.

Il vettore  $\underline{p}_k$  è definito come una combinazione lineare del gradiente e del vettore precedente:

$$\underline{p}_k = -\nabla f_k + \beta_k \underline{p}_{k-1}$$

Dove  $\beta$  è definito sfruttando la definizione di vettore coniugato:

$$\beta_k = \frac{\nabla f_k^T A \underline{p}_{k-1}}{\underline{p}_{k+1}^T A \underline{p}_{k-1}}$$

Inizialmente il valore del vettore è  $\underline{p}_0 = -\nabla f_0$ .

Esistono vari metodi del gradiente coniugato che variano in base a come viene calcolato il valore di  $\beta_{k+1}$  a partire dall'iterazione precedente.

In particolare è utile perchè utilizza poco spazio, non è computazionalmente pesante e converge rapidamente. Viene però preferita l'eliminazione gaussiana quando il problema è di piccola dimensione, essendo meno sensibile agli errori di arrotondamento.

## 6.5 Metodi Trust Region

Questi metodi realizzano un modello della funzione con lo sviluppo di Taylor fermato al secondo ordine, ma può usare sia l'Hessiano che una sua approssimazione. Tramite questo si determina un raggio della trust region e quindi si procede ad ottenere una direzione tale che minimizzi la funzione, e nel caso la discesa non sia sufficiente si riduce il raggio e si ripete l'operazione. Per un raggio sufficientemente grande, quando viene utilizzata l'Hessiana e questa è definita positiva la direzione che si ottiene coincide con quella che si otterrebbe dai metodi Newton.

Il metodo della trust region richiede di calcolare una sequenza di sottoproblemi la cui funzione obiettivo e vincoli sono entrambi quadratici.

**Teorema 6.5.1.** Il vettore  $\underline{p}^*$  è una soluzione globale per il problema di trust-region di raggio  $\Delta$ :

$$\min_{\underline{p} \in \mathbb{R}^n} m(\underline{p}) = f(\underline{x}) + \nabla f^T \underline{p} + \frac{1}{2} \underline{p}^T B \underline{p} \quad \|\underline{p}\| \leq \Delta$$

se e solo se  $\underline{p}^*$  appartiene alla trust-region ed esiste uno scalare  $\lambda \geq 0$  tale per cui:

$$\begin{cases} (B + \lambda I) \underline{p}^* &= -\nabla f \\ \lambda(\Delta - \|\underline{p}^*\|) &= 0 \\ (B + \lambda I) &\text{è definita semipositiva} \end{cases}$$

I metodi trust-region garantiscono convergenza globale grazie ad una riduzione sufficiente del modello che può essere quantificata in termini del **punto di Cauchy**, definito come il punto minimo del modello quadratico lungo la direzione  $-\nabla f$ . Questo punto può essere calcolato nell'ordine di  $O(n^2)$  e viene utilizzato per determinare se la soluzione di un sotto problema è accettabile.

Un metodo trust-region sarà globalmente convergente se un suo step  $\underline{p}_k$  dà una riduzione nel modello  $m_k$  che è almeno un multiplo positivo della decrescita ottenuta tramite lo step di Cauchy. Se scegliessimo sempre il punto di Cauchy per la discesa implementeremmo banalmente il metodo di discesa rapida con una scelta fissata della step length e questo sarebbe poco efficiente.

Dobbiamo utilizzare le informazioni che possiamo trarre dalla matrice  $B_k$ , la nostra approssimazione (o coincidente) dell'Hessiana.

### 6.5.1 Metodo zampa di cane

Questo metodo risulta applicabile quando la matrice  $B$  è definita positiva. Si inizia computando il punto  $\underline{p}^{(B)} = -B^{-1}\nabla f$  e quindi se questo punto cade nel raggio della regione ammissibile esso coincide con il punto ottimo  $\underline{p}^*$ . Altrimenti si determina una traiettoria ottimale tra un punto  $\underline{p}^{(U)} = -\frac{\nabla f^T \nabla f}{\nabla f^T B \nabla f} \nabla f$  ed il punto  $\underline{p}^{(B)}$  seguendo la curva seguente:

$$\underline{p}(\tau) = \begin{cases} \tau \underline{p}^{(U)} & \tau \in [0, 1] \\ \underline{p}^{(U)} + (\tau - 1)(\underline{p}^{(B)} - \underline{p}^{(U)}) & \tau \in [1, 2] \end{cases}$$

### 6.5.2 Proprietà della matrice B

Se la matrice  $B$  risulta definita semi positiva, allora:

1.  $\|\underline{p}(\tau)\|$  è una funzione crescente di  $\tau$
2.  $m(\underline{p}(\tau))$  è una funzione decrescente di  $\tau$
3.  $\tau$  può essere calcolata risolvendo l'equazione scalare quadratica:

$$\left\| \underline{p}^{(U)} + (\tau - 1)(\underline{p}^{(B)} - \underline{p}^{(U)}) \right\|^2 = \Delta^2$$

4. Esistono metodi iterativi per risolvere il modello quadratico.
5. La convergenza può arrivare ad essere **super-lineare**.

# 7

## Ottimizzazione vincolata

Un problema è detto di ottimizzazione vincolata quando il dominio è limitato da una o più funzioni. In presenza di vincoli, problemi complessi possono divenire semplici: funzioni non convesse su  $\mathbb{R}$  per esempio potrebbero esserlo sul dominio di definizione.

Le condizioni di ottimalità possono essere ottenute tramite la funzione lagrangiana.

**Teorema 7.0.1.** Data una funzione  $f(\underline{x})$  e  $\underline{h}$  vincoli bilateri, con  $f, \underline{h} \in C^1$ . Se i gradienti dei vincoli nel punto ottimo  $\underline{x}^*$  sono **linearmente indipendenti**, se  $\underline{x}^*$  sono di minimo locale di  $f(\underline{x})$  ed esso soddisfa i vincoli  $\underline{h}$ , allora esiste un vettore  $\underline{\lambda}^*$  tale che  $(\underline{x}^*, \underline{\lambda}^*)$  è un punto stazionario della funzione Lagrangiana  $L$ :

$$\begin{cases} \frac{\partial L}{\partial x_i} = \frac{\partial f(\underline{x}^*)}{\partial x_i} + \sum_{j=1}^h \lambda_j^* \frac{\partial h_j(\underline{x}^*)}{\partial x_i} = 0 \\ \frac{\partial L}{\partial \lambda_i} = h_i(\underline{x}^*) = 0 \end{cases}$$

**Definizione 7.0.2 (Condizioni di regolarità).** Un punto  $\underline{x}^*$  soddisfa le condizioni di regolarità se non esiste un vettore  $\underline{h}$  tale per cui tutti i gradienti dei vincoli attivi sono linearmente indipendenti in quel punto. Un punto che soddisfa queste condizioni è detto **regolare**.

**Teorema 7.0.3 (Condizioni sufficienti del prim'ordine per problemi convessi).** Data una funzione  $f(\underline{x})$  ed un vettore  $\underline{h}$  di vincoli bilateri, con  $f, \underline{h} \in C^1$ . Se la matrice jacobiana  $J(\underline{x}^*)$  ha rango massimo  $\|\underline{h}\|$ , se esiste un vettore  $\underline{\lambda}^*$  tale che  $(\underline{x}^*, \underline{\lambda}^*)$  è un punto stazionario della funzione lagrangiana  $L$ , allora  $\underline{x}^*$  è un minimo locale di  $f(\underline{x})$ .

Mettendo assieme tutti questi risultati matematici otteniamo le **condizioni di Karush-Kuhn-Tucker**, o condizioni KKT

**Teorema 7.0.4 (Condizioni di Karush-Kuhn-Tucker).** Sia  $f$  una funzione,  $h_i$  con  $i \in \{1, \dots, s\}$  dei vincoli bilateri e  $g_j$  con  $j \in \{1, \dots, m\}$  dei vincoli monolateri e sia l'insieme  $X$  definito come:

$$X = \{x \in \mathbb{R}^n : g_j(x) \leq 0, h_i(x) = 0 \quad \forall i, j\} \quad \text{e} \quad f, g_j, h_i \in C^1(X) \quad \forall i, j$$

Se  $x^*$  è un punto regolare in  $X$  e un punto di minimo locale per  $f \in X$ , allora esistono  $s$  moltiplicatori  $\lambda_i \in \mathbb{R}$  e  $m$   $\mu_j \geq 0$  tali che:

$$\begin{aligned} \nabla f(x^*) + \sum_{i=1}^s \lambda_i \nabla h_i(x^*) + \sum_{j=1}^m \mu_j \nabla g_j(x^*) &= 0 \\ \mu_j g_j(x^*) &= 0 \end{aligned}$$

**Teorema 7.0.5 (Condizioni KKT nel caso convesso).** Se le funzioni  $f, \underline{g}, \underline{h} \in C^1$  sono **funzioni convesse** le condizioni KKT sono condizioni sufficienti.

**Definizione 7.0.6 (Cono critico).** Dato un punto  $\underline{x}^*$  ed i vettori di moltiplicatori  $\underline{\lambda}^*$  e  $\underline{\mu}^*$  che soddisfano le condizioni KKT, viene chiamato **cono critico** l'insieme delle direzioni di discesa verso il punto  $\underline{x}^*$ .

$$C(\underline{x}^*, \underline{\lambda}^*, \underline{\mu}^*) = \left\{ \underline{d} \in F(\underline{x}^*) : \nabla h_j(\underline{x}^*)^T \underline{d} = 0, j \in E; g_j(\underline{x}^*)^T \underline{d}, j \in I, \text{ con } \underline{\lambda}^* > 0 \right\}$$

Queste direzioni sono ortogonali al gradiente di  $f$ .

## 7.1 Condizioni di ottimalità del secondo ordine

**Teorema 7.1.1 (Matrice hessiana semi-definita positiva nel punto minimo).** Dato un problema di ottimizzazione in cui le funzioni  $f, \underline{g}, \underline{h} \in C^2$ . Se  $\underline{x}^*$  è un minimo locale che appartiene alla regione ammissibile ed esistono due vettori  $\underline{\lambda}^*$  e  $\underline{\mu}^*$  che soddisfano le condizioni KKT, allora:

$$\underline{d}^T \nabla_{\underline{x}, \underline{x}}^2 L(\underline{x}^*, \underline{\lambda}^*, \underline{\mu}^*) \underline{d} \geq 0, \quad \forall \underline{d} \in \text{Cono critico}$$

Se la matrice hessiana è **definita positiva** nel cono critico, allora le condizioni KKT divengono sufficienti.

**Teorema 7.1.2 (Punto di minimo locale stretto).** Dato un problema di ottimizzazione in cui le funzioni  $f, \underline{g}, \underline{h} \in C^2$ . Se  $\underline{x}^*$  appartiene alla regione ammissibile, esistono due vettori  $\underline{\lambda}^*$  e  $\underline{\mu}^*$  che soddisfano le condizioni KKT e vale la seguente relazione:

$$\underline{d}^T \nabla_{\underline{x}, \underline{x}}^2 L(\underline{x}^*, \underline{\lambda}^*, \underline{\mu}^*) \underline{d} > 0, \quad \forall \underline{d} \in \text{Cono critico}$$

Allora  $\underline{x}^*$  è un punto di minimo locale stretto.

## 7.2 Modello quadratico con vincoli lineari

Un problema quadratico del tipo:

$$\begin{aligned} \min q(\underline{x}) &= \frac{1}{2} \underline{x}^T Q \underline{x} + \underline{b}^T \underline{x} \\ A \underline{x} &\geq \underline{d} \end{aligned}$$

è risolvibile facilmente quando la matrice  $Q$  è definita positiva.

### 7.2.1 Metodo dell'insieme attivo primale per problemi quadratici convessi (Primal Active set)

I metodi **Primal Active set** ad ogni iterazione risolvono un sottoproblema quadratico in cui i vincoli di disuguaglianza sono imposti come uguaglianze sul **working set**  $W_k$ .

I gradienti  $\underline{\alpha}_i^T$  dei vincoli sul set  $W_k$  sono linearmente indipendenti.

Per prima cosa viene controllato se  $\underline{x}_k$  minimizza la funzione quadratica  $q(\underline{x})$  su  $W_k$  e se questo non avviene viene calcolato uno step  $p$  risolvendo un sotto-problema quadratico sul set  $W_k$ , vincolato da uguaglianze:



$$\begin{aligned}
\underline{\mathbf{p}} &= \underline{\mathbf{x}} - \underline{\mathbf{x}}_k \\
\underline{\mathbf{g}}_k &= Q\underline{\mathbf{x}}_k + \underline{\mathbf{b}} \\
\underline{\mathbf{p}}_k &= \operatorname{argmin} q(\underline{\mathbf{x}}) = \operatorname{argmin} q(\underline{\mathbf{x}}_k + \underline{\mathbf{p}}) = \operatorname{argmin} \frac{1}{2} \underline{\mathbf{p}}^T Q \underline{\mathbf{p}} + \underline{\mathbf{g}}_k^T \underline{\mathbf{p}} + \rho_k \\
\underline{\mathbf{p}}_k &= \frac{1}{2} \underline{\mathbf{x}}_k^T Q \underline{\mathbf{x}}_k + \underline{\mathbf{b}}^T \underline{\mathbf{x}}_k \\
\underline{\boldsymbol{\alpha}}_i^T \underline{\mathbf{p}}_k &= 0 \quad \forall i \in W_q
\end{aligned}$$

Per ogni  $i \in W_k$  vale che:

$$\underline{\boldsymbol{\alpha}}_i^T (\underline{\mathbf{x}}_k + \alpha \underline{\mathbf{p}}_k) = \underline{\boldsymbol{\alpha}}_i^T \underline{\mathbf{x}}_k = \underline{\mathbf{d}}_i \forall \alpha$$

Se  $\underline{\mathbf{p}}_k \neq 0$  allora aggiorno il valore di  $\alpha_k$  ed ottengo il prossimo punto,  $\underline{\mathbf{x}}_{k+1} = \underline{\mathbf{x}}_k + \alpha_k \underline{\mathbf{p}}_k$ :

$$\alpha_k = \min \left\{ 1, \min \left\{ \frac{\underline{\mathbf{d}}_i - \underline{\boldsymbol{\alpha}}_i^T \underline{\mathbf{x}}_k}{\underline{\boldsymbol{\alpha}}_i^T \underline{\mathbf{p}}_k} : i \in W_k, \underline{\boldsymbol{\alpha}}_i^T \underline{\mathbf{p}}_k < 0 \right\} \right\}$$

Altrimenti testo le condizioni KKT o aggiorno il set di lavoro  $W_k$ .

# 8

## Algoritmi

### 8.1 Metodo della penalità quadratica

Questo metodo trasforma il problema vincolato in uno privo di vincoli, sfruttando una **penalty function**.

Si parte quindi da un problema vincolato:

$$\begin{aligned} \min f(\underline{\mathbf{x}}) \\ \underline{\mathbf{h}}(\underline{\mathbf{x}}) = \underline{\mathbf{0}} \end{aligned}$$

Si costruisce una **penalty function** utilizzando i vincoli:

$$p(\underline{\mathbf{x}}) = \sum_{j=1}^h h_j^2(\underline{\mathbf{x}})$$

Figura 8.1: Penalty function

Il modello che si ottiene è quindi la somma pesata tra la funzione obiettivo e la **penalty function**:

$$\min q(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}}) + \alpha \sum_{j=1}^h h_j^2(\underline{\mathbf{x}})$$

Aumentando il parametro  $\alpha$  a  $+\infty$ , aumentiamo la penalità della violazione dei vincoli, aumentandone la severità.

Possiamo usare su questo problema le tecniche tratte dall'ottimizzazione svincolata.

### 8.2 Metodo delle barriere

Dato un problema vincolato con disuguaglianze:

$$\begin{aligned} \min f(\underline{\mathbf{x}}) \\ \underline{\mathbf{g}}(\underline{\mathbf{x}}) \leq \underline{\mathbf{0}} \end{aligned}$$

Si procede dividendo la regione ammissibile in due set: uno definito dalla frontiera dei vincoli ( $\underline{\mathbf{g}}(\underline{\mathbf{x}}) = \underline{\mathbf{0}}$ ) ed uno dall'interno ( $\underline{\mathbf{g}}(\underline{\mathbf{x}}) < \underline{\mathbf{0}}$ ).

Questo metodo è applicabile solo quando il set interno non è vuoto. Viene utilizzata una **funzione di barriera** che aumenta a  $\infty$  quando il punto  $\underline{\mathbf{x}}$  tende al set di frontiera.

$$v(\underline{\mathbf{x}}) = \sum_{j=1}^k \log(-g_j^2(\underline{\mathbf{x}}))$$

Figura 8.2: Barrier function

Mettendole assieme si ottiene:

$$\min q(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}}) + \alpha v(\underline{\mathbf{x}})$$

### 8.3 Metodo della proiezione del gradiente

Viene applicato su problemi con vincoli lineari:

$$\begin{aligned} \min f(\underline{\mathbf{x}}) \\ A\underline{\mathbf{x}} = \underline{\mathbf{b}} \end{aligned}$$

Si inizia con una soluzione ammissibile  $\underline{\mathbf{x}}' : A\underline{\mathbf{x}}' = \underline{\mathbf{b}}$  e si cerca una soluzione migliore  $\underline{\mathbf{x}} = \underline{\mathbf{x}}' + \alpha \underline{\mathbf{d}}$  lungo la direzione  $\underline{\mathbf{d}}$  che deve essere normalizzata, deve minimizzare la derivata direzionale  $\nabla f(\underline{\mathbf{x}}')^T \underline{\mathbf{d}}$  in  $\underline{\mathbf{x}}'$  ed il punto  $\underline{\mathbf{x}}$  deve essere ammissibile, cioè  $A\underline{\mathbf{d}} = \underline{\mathbf{0}}$ .

Da questo si va a costruire il problema di minimizzazione della derivata direzionale, da cui si ottiene che la direzione  $\underline{\mathbf{d}}$  migliore è:

$$\underline{\mathbf{d}} = \frac{(I - A^T (AA^T)^{-1} A) \nabla f(\underline{\mathbf{x}}')}{\|(I - A^T (AA^T)^{-1} A) \nabla f(\underline{\mathbf{x}}')\|}$$

La direzione che migliora di più  $f(\underline{\mathbf{x}})$  è l'anti-gradiente e  $\underline{\mathbf{d}}$  risulta esserne la **proiezione** sull'iperpiano  $A\underline{\mathbf{x}} = \underline{\mathbf{d}}$ . La matrice  $P = (I - A^T (AA^T)^{-1} A) \nabla f(\underline{\mathbf{x}}')$  è detta **matrice di proiezione**.

Si procede quindi con  $\underline{\mathbf{d}} = -P \nabla f(\underline{\mathbf{x}}_k)$  e si ottiene il punto successivo tramite:

$$\underline{\mathbf{x}}_{k+1} = \underline{\mathbf{x}}_k + \alpha \underline{\mathbf{d}}$$

Dove  $\alpha$  può essere ottenuto con per esempio il metodo di Armijo.

#### 8.3.1 Caso con vincoli generici non lineari

Nel caso con vincoli generici viene utilizzato uno sviluppo di Taylor per individuare dei vincoli lineari nell'intorno del punto corrente  $\underline{\mathbf{x}}'$ .

Partendo quindi da un problema del tipo:

$$\begin{aligned} \min f(\underline{\mathbf{x}}) \\ \underline{\mathbf{h}}(\underline{\mathbf{x}}) = \underline{\mathbf{0}} \end{aligned}$$

Si svolge lo sviluppo di Taylor:

$$h_j(\underline{\mathbf{x}}) = h_j(\underline{\mathbf{x}}') + \nabla h_j(\underline{\mathbf{x}}')^T (\underline{\mathbf{x}} - \underline{\mathbf{x}}')$$

Da cui si ottiene:

$$\nabla h_j(\underline{\mathbf{x}}')^T \underline{\mathbf{x}} - \nabla h_j(\underline{\mathbf{x}}')^T \underline{\mathbf{x}}' = 0$$

Imponendo quindi  $A = \nabla h_j(\underline{\mathbf{x}}')^T$  e  $\underline{\mathbf{b}} = \nabla h_j(\underline{\mathbf{x}}')^T \underline{\mathbf{x}}'$  ci si riconduce al caso precedente:

$$\begin{aligned} \min f(\underline{\mathbf{x}}) \\ A\underline{\mathbf{x}} = \underline{\mathbf{b}} \end{aligned}$$

Con l'eccezione che ora la matrice di proiezione  $P$  dipende da  $A$ , che a sua volta ora dipende da  $\underline{\mathbf{x}}'$ . Pertanto la direzione da usare sarà  $\underline{\mathbf{d}} = -P(\underline{\mathbf{x}}_k)\nabla f(\underline{\mathbf{x}}_k)$ .

Per costruzione del metodo però, molto probabilmente il punto ottenuto  $\underline{\mathbf{x}}_{k+1}$  non soddisfa i vincoli non lineari originali, per cui si procede con uno step correttivo ottenendo  $\underline{\mathbf{x}}_{k+1\text{corretto}}$  con:

$$P(\underline{\mathbf{x}}_k)(\underline{\mathbf{x}}_{k+1\text{corretto}} - \underline{\mathbf{x}}_{k+1}) = \underline{\mathbf{0}} \quad \wedge \quad \underline{\mathbf{h}}(\underline{\mathbf{x}}_{k+1\text{corretto}}) = 0$$

Da cui si ottiene che  $\underline{\mathbf{x}}_{k+1\text{corretto}}$  risulta:

$$\underline{\mathbf{x}}_{k+1\text{corretto}} \approx \underline{\mathbf{x}}_{k+1} - A^T (AA^T)^{-1} \underline{\mathbf{h}}(\underline{\mathbf{x}}_{k+1})$$

Lo step correttivo viene applicato fino a che il valore di  $\underline{\mathbf{h}}(\underline{\mathbf{x}}_{k+1})$  è sufficientemente piccolo, mentre l'algoritmo complessivo si interrompe quando  $P(\underline{\mathbf{x}}_k)\nabla f(\underline{\mathbf{x}}_k) \approx 0$ .

## 8.4 Augmented lagrangean method

Questo metodo combina l'uso di una funzione lagrangiana con una funzione di penalità quadratica, l'idea è quella di approssimare i moltiplicatori di Lagrange. Si inizia con un problema generico:

$$\begin{aligned} \min f(\underline{\mathbf{x}}) \\ \underline{\mathbf{h}}(\underline{\mathbf{x}}) = \underline{\mathbf{0}} \end{aligned}$$

E si realizza una **funzione lagrangiana incrementata**:

$$L(\underline{\mathbf{x}}, \underline{\boldsymbol{\lambda}}, \rho) = f(\underline{\mathbf{x}}) + \sum_{j=1}^h \lambda_j h_j(\underline{\mathbf{x}}) + \rho \sum_{j=1}^h h_j^2(\underline{\mathbf{x}})$$

Figura 8.3: Augmented lagrangean function

1. Nel caso in cui  $\lambda_j = 0$  si ottiene la funzione di penalità.
2. Se è noto  $\lambda^* \forall \rho > 0$  minimizzando  $L(\underline{\mathbf{x}}, \underline{\boldsymbol{\lambda}}, \rho)$  rispetto a  $\underline{\mathbf{x}}$  si ottiene  $\underline{\mathbf{x}}^*$
3. Se  $\underline{\boldsymbol{\lambda}}^k$  è una approssimazione valida di  $\underline{\boldsymbol{\lambda}}^*$ , allora possiamo approssimare  $\underline{\mathbf{x}}^*$  minimizzando  $L(\underline{\mathbf{x}}, \underline{\boldsymbol{\lambda}}^k, \rho)$  anche per valori bassi di  $\rho$ .
4. Il parametro  $\rho$  deve garantire che  $L(\underline{\mathbf{x}}, \underline{\boldsymbol{\lambda}}^k, \rho)$  abbia un minimo locale rispetto a  $\underline{\mathbf{x}}$  e non solamente un punto stazionario.

Si procede quindi iterativamente, dati i valori iniziali di  $\underline{\boldsymbol{\lambda}}^k$  e  $\rho$ , finchè  $\|L(\underline{\mathbf{x}}, \underline{\boldsymbol{\lambda}}^k, \rho)\| > \epsilon$  si procede ad ottenere il punto ottimo  $\underline{\mathbf{x}}^*_k$  risolvendo  $L(\underline{\mathbf{x}}, \underline{\boldsymbol{\lambda}}^k, \rho)$  rispetto a  $\underline{\mathbf{x}}$  con un qualsiasi metodo di ottimizzazione svincolata, quindi si aggiorna il valore di  $\lambda$  come  $\lambda_j^{k+1} = \lambda_j^k + 2\rho h_j(\underline{\mathbf{x}}^*_k)$ . Eventualmente viene aggiornato anche il valore di  $\rho$ .

## 8.5 Sequential quadratic programming (SQP)

L'idea è di applicare il metodo di Newton per identificare  $(\underline{\mathbf{x}}^*, \underline{\boldsymbol{\lambda}}^*)$  dalle condizioni KKT di un problema vincolato, riducendo ogni step del metodo di Newton in un problema di programmazione quadratica. Considerando un problema generale:

$$\begin{aligned} \min f(\underline{\mathbf{x}}) \\ \underline{\mathbf{g}}(\underline{\mathbf{x}}) &\leq 0 \\ \underline{\mathbf{h}}(\underline{\mathbf{x}}) &= 0 \end{aligned}$$

Possiamo costruirne il modello lagrangiano:

$$L(\underline{\mathbf{x}}, \underline{\boldsymbol{\lambda}}, \underline{\boldsymbol{\mu}}) = f(\underline{\mathbf{x}}) + \sum_{j=1}^k \lambda_j g_j(\underline{\mathbf{x}}) + \sum_{j=1}^h \mu_j h_j(\underline{\mathbf{x}})$$

Data un'approssimazione della soluzione e dei moltiplicatori lagrangiani  $(\underline{\mathbf{x}}_k, \underline{\boldsymbol{\lambda}}_k, \underline{\boldsymbol{\mu}}_k)$ , nota la matrice hessiana di  $L$  possiamo scrivere:

$$\nabla^2 L(\underline{\mathbf{x}}_k) = H(\underline{\mathbf{x}}_k) + \sum_{j=1}^k \lambda_j^k \nabla^2 g_j(\underline{\mathbf{x}}_k) + \sum_{j=1}^h \mu_j^k \nabla^2 h_j(\underline{\mathbf{x}}_k)$$

Per aggiornare i valori dobbiamo identificare la direzione di Newton  $\underline{\mathbf{d}}$  tramite il sotto-problema quadratico seguente:

$$\begin{aligned} \min \phi(\underline{\mathbf{x}}) &= f(\underline{\mathbf{x}}_k) + \nabla f(\underline{\mathbf{x}}_k)^T \underline{\mathbf{d}} + \frac{1}{2} \underline{\mathbf{d}}^T \nabla^2 L(\underline{\mathbf{x}}_k) \underline{\mathbf{d}} \\ \underline{\mathbf{g}}(\underline{\mathbf{x}}_k) + \left[ \frac{\partial \underline{\mathbf{g}}(\underline{\mathbf{x}}_k)}{\partial \underline{\mathbf{x}}} \right] \underline{\mathbf{d}} &\leq 0 \\ \underline{\mathbf{h}}(\underline{\mathbf{x}}_k) + \left[ \frac{\partial \underline{\mathbf{h}}(\underline{\mathbf{x}}_k)}{\partial \underline{\mathbf{x}}} \right] \underline{\mathbf{d}} &= 0 \end{aligned}$$

Risolvendo il problema si ottiene  $\underline{\mathbf{x}}_{k+1} = \underline{\mathbf{x}}_k + \underline{\mathbf{d}}_k$  ed i moltiplicatori lagrangiani aggiornati  $\underline{\boldsymbol{\lambda}}_{k+1}$  e  $\underline{\boldsymbol{\mu}}_{k+1}$ . L'iterazione si interrompe basandosi su un criterio costruito sulla norma della direzione migliorante  $\underline{\mathbf{d}}$ .

Il Sequential quadratic programming identifica un punto che soddisfa le condizioni KKT, per cui tutti i punti non regolari sono evitati dall'algoritmo.

## **Parte II**

# **Programmazione lineare intera**

# 9

## TSP simmetrico

Prendiamo in considerazione il problema del commesso viaggiatore su un grafo non orientato  $G = (N, E)$ :

$\min \sum_{e \in E} c_e x_e$	Il costo degli archi inseriti deve essere minimo
$\sum_{e \in \delta(i)} x_e = 2 \quad \forall i$	Si deve entrare ed uscire da ogni nodo esattamente una volta
$\sum_{e \in E(S)} x_e \leq  S  - 1 \quad \forall S \subset N; 2 \leq  S  \leq  N  - 1$	Non possono esistere sotto-cicli, cioè in ogni sotto-insieme di $n$ nodi vengono percorsi al $n - 1$ archi
$x_e \in 0, 1 \quad \forall e \in E$	Ogni arco può essere percorso o non percorso.

Figura 9.1: Modello TSP

È possibile riscrivere il vincolo dell'assenza di sotto-cicli (SEC, sub-tour elimination constraints) come il fatto che da ogni sotto-insieme di nodi entrino ed escano almeno due archi:

$\min \sum_{e \in E} c_e x_e$	Il costo degli archi inseriti deve essere minimo
$\sum_{e \in \delta(i)} x_e = 2 \quad \forall i$	Si deve entrare ed uscire da ogni nodo esattamente una volta
$\sum_{e \in \delta(S)} x_e \geq 2 \quad \forall S \subset N; 2 \leq  S  \leq  N  - 1$	Non possono esistere sotto-cicli, cioè in ogni sotto-insieme di nodi devono entrare ed uscire almeno due archi.
$x_e \in 0, 1 \quad \forall e \in E$	Ogni arco può essere percorso o non percorso.

Figura 9.2: Modello TSP

### 9.1 Problema di separazione

I vincoli SEC sono in numero esponenziale e quindi si procede risolvendo il modello ignorandoli. Data quindi una soluzione  $\underline{x}^*$ , se essa non definisce un **ciclo hamiltoniano**, si cerca di individuare un vincolo SEC violato, cioè si identifica un sotto insieme  $S$  per cui la disequazione  $\sum_{e \in \delta(S)} x_e \geq 2$  risulta falsa.

Costruito un grafo composto dagli archi che sono stati scelti in  $\underline{x}^*$ , il problema di separazione coincide ad identificare un taglio di capacità  $< 2$  che coincide con il taglio di capacità minima. Quest'ultimo problema coincide, a sua volta, con il problema di calcolare il flusso massimo, che però è definito su un grafo orientato.

È possibile però evitare questa trasformazione utilizzando l'albero di Gomory-Hu.

## 9.2 Albero di Gomory-Hu

Si tratta di un albero ricoprente  $(N, T)$  degli  $G$  nodi di  $G' = (N, E^*)$ . I lati che lo compongono non devono essere necessariamente essere i lati scelti in  $\underline{x}^*$ . Dato un lato  $e$ , la sua rimozione crea due componenti connesse con insiemi di vertici  $S, N \setminus S$ . Ad ognuno di questi lati  $e$  è associato un peso  $w(e)$  tale che:

1. Per ogni coppia di nodi, il valore di costo minimo che li separa in  $G$  è quello dell'unico cammino che li congiunge nell'albero, cioè il peso del taglio minimo dei due nodi nell'albero.
2. Per ogni lato  $e$  dell'albero, il peso  $w(e)$  è quello del taglio associato ad  $e$  in  $G'$ .

Per trovare il taglio di peso minimo in  $G'$  risulta sufficiente cercare il lato  $e$  di peso minimo nell'albero  $T$ .

## 9.3 Comb inequalities (Disuguaglianze a pettine)

Si tratta di una famiglia di disuguaglianze usate nella risoluzione del TSP simmetrico. Viene definita considerando un insieme di nodi  $H$ , il manico, e  $T$ , i denti, tali che:

$$\begin{aligned} H, T_1, T_2, \dots, T_t &\subseteq N \\ T_j \setminus H &\neq \emptyset \forall j \\ T_j \cap H &\neq \emptyset \forall j \\ T_j \cap T_i &= \emptyset \forall j, i: i \neq j \\ t &\geq 3 \quad \wedge \quad \text{dispari} \end{aligned}$$

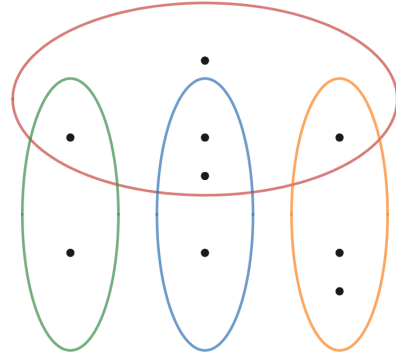


Figura 9.3: Comb Inequalities

La disuguaglianza afferma che:

$$x(E(H)) + \sum_{j=1}^t x(E(T_j)) \leq |H| + \sum_{j=1}^t |T_j| - \frac{3t+1}{2}$$

Dove  $x(E(S)) = \sum_{e \in E(S)} x_e$  è l'insieme dei lati in un ciclo hamiltoniano i cui vertici appartengono entrambi all'insieme di vertici  $S \subseteq N$ . È possibile ricavare tale relazione a partire dai vincoli di grado:

$$2x(E(S)) + x(\delta(S)) = 2|S|$$

Dove  $x(\delta(S)) = \sum_{e \in \delta(S)} x_e$  è l'insieme di lati in un ciclo hamiltoniano che hanno un estremo in  $S$  ed uno in  $N \setminus S$ .

Riscriviamo la relazione in funzione di  $x$  ed otteniamo:

$$x(E(S)) = |S| - \frac{x(\delta(S))}{2}$$

Sommiamo tale relazione sugli insiemi  $H, T_1, T_2, \dots, T_t$  ed otteniamo:

$$x(E(H)) + \sum_{j=1}^t x(E(T_j)) = |H| + \sum_{j=1}^t |T_j| - \frac{1}{2} \left( x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j)) \right)$$

La disuguaglianza a pettine risulta vera se ricaviamo che:

$$x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j)) \geq 3t + 1$$



Per farlo procediamo indicando con  $x(\delta_j(H))$  l'insieme dei lati di un ciclo hamiltoniano con un estremo in  $H \cap T_j$  ed uno non in  $H$ . Di conseguenza  $x(\delta(H)) \geq \sum_{j=1}^t x(\delta_j(H))$  (il primo termine risulta maggiore perch  vi considero anche i lati di un ciclo hamiltoniano incidenti a vertici in  $H$  ma non nell'intersezione).

Per costruzione del ciclo che pu  toccare i vertici in  $H \cap T_j$  e  $T_j \setminus H$  risulta che:

$$x(\delta_j(H)) + x(\delta(T_j)) \geq 3$$

Unendo le ultime due disuguaglianze si ottiene:

$$x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j)) \geq \sum_{j=1}^t x(\delta_j(H)) + \sum_{j=1}^t x(\delta(T_j)) \geq 3t$$

  noto dalla prima relazione di uguaglianza che  $x(\delta(S))$  sia un valore pari  $\forall S$ , mentre  $t$    per costruzione dispari. Risulta sufficiente aggiungere 1 alla destra della disequazione per ricavare la relazione cercata:

$$x(\delta(H)) + \sum_{j=1}^t x(\delta(T_j)) \geq 3t + 1$$

# 10

## Cover Inequalities

Consideriamo il set di possibili soluzioni di un problema dello zaino a pesi e peso massimo strettamente positivi  $X = \{\underline{x} \in B^n : \underline{x}\underline{a} \leq b\}$ .

**Definizione 10.0.1.** Cover Un set  $S \subseteq N$  è una **cover** se  $\underline{a}_S > b$ . Una cover è detta **minimale** se  $C \setminus \{j\}$  non è una cover per nessun  $j \in C$ .

Il **vettore di incidenza associato** alla cover  $\underline{x}^C$  **non appartiene** alla regione ammissibile definita da  $X$ .

Se un set  $T \subseteq N$  contiene una cover  $C$ , allora anche  $T$  è una cover.

**Proposizione 10.0.2.** Validità di una cover inequality Se  $C \subseteq N$  è una cover per  $X$ , allora la cover inequality

$$\sum_{j \in C} x_j \leq |C| - 1$$

È valida per  $X$

### 10.1 Irrobustire una cover inequality

**Proposizione 10.1.1.** Se  $C$  è una cover per  $X$ , la cover inequality estesa

$$\sum_{j \in E(C)} x_j \leq |C| - 1$$

è valida per  $X$ , dove  $E(C) = C \cup \{j : a_j \geq a_i \quad \forall i \in C\}$

### 10.2 Sollevare le cover inequalities

In generale il problema risulta essere trovare il miglior valore possibile di un peso  $\alpha_j, j \in N \setminus C$  in modo tale che la disuguaglianza

$$\sum_{j \in C} x_j + \sum_{j \in N \setminus C} \alpha_j x_j \leq |C| - 1$$

sia valida in  $X$ .

La procedura di sollevamento delle cover inequalities è un metodo per produrre questo set di valori.

Siano  $j_1, \dots, j_r$  un insieme ordinato di  $N \setminus C$  e sia  $t$  uno scalare inizialmente di valore pari a 1. Si inizia dalla migliore uguaglianza identificata sino ad ora:

$$\sum_{j \in C} x_j + \sum_{i=1}^{t-1} \alpha_{j_i} \underline{x}_{j_i} \leq |C| - 1$$

E si procede ad identificare il massimo valore di  $\alpha_{j_i}$  per cui essa risulta essere valida risolvendo il seguente problema dello zaino:

$$\begin{aligned} \xi_t = \max \sum_{j \in C} x_j + \sum_{i=1}^{t-1} \alpha_{j_i} \underline{x}_{j_i} \\ \sum_{j \in C} x_j + \sum_{i=1}^{t-1} a_{j_i} \underline{x}_{j_i} \leq b - a_{j_t} \\ x \in \{0, 1\}^{|C|+t-1} \end{aligned}$$

Figura 10.1: Problema dello zaino per il sollevamento delle cover inequalities

Si inizializza  $\alpha_{j_t} = |C| - 1 - \xi_t$  e si interrompe l'iterazione quando  $t = r$ .

### 10.3 Separazione delle cover inequalities

Sia  $\mathcal{F}$  la famiglia delle cover inequalities per  $X$  e procediamo ad esaminare il problema della separazione su questa famiglia. Dato un punto non-intero  $\underline{x}^* \in [0, 1]^n$  vogliamo verificare se esso soddisfa o meno tutte le cover inequalities. Questo può essere formalizzato come:

$$\sum_{j \in C} (1 - x_j) \geq 1$$

La domanda che ci poniamo quindi è la seguente: esiste un set  $C \subseteq N$  tale che  $\sum_{j \in C} a_j > b$  per cui  $\sum_{j \in C} (1 - x_j^*) < 1$ ?

Espresso in termini formali, la seguente disequazione è valida?

$$\xi = \min \left\{ \sum_{j \in N} (1 - x_j^*) z_j : \sum_{j \in N} a_j z_j > b, z \in B^n \right\} < 1$$

**Teorema 10.3.1.** 1. Se  $\xi \geq 1$  allora  $x^*$  soddisfa tutte le cover inequalities.

2. Se  $\xi \leq 1$  con come soluzione ottimale  $z^R$ , allora la cover inequality  $\sum_{j \in R} x_j \leq |R| - 1$  taglia il punto  $\underline{x}^*$  di un valore pari  $1 - \xi$ .

# 11

## Dualità lagrangiana

### 11.1 Rilassamento lagrangiano

Viene utilizzata per risolvere problemi di programmazione intera. Dato un problema con vincoli complessi:

$$\begin{aligned} z = \max_{\underline{\mathbf{x}}} & \underline{\mathbf{c}}^T \underline{\mathbf{x}} \\ & D\underline{\mathbf{x}} \leq \underline{\mathbf{b}} \\ & \underline{\mathbf{x}} \in X \end{aligned}$$

Il seguente è il corrispondente **rilassamento lagrangiano**:

**Definizione 11.1.1.** Rilassamento lagrangiano Dato un vettore di moltiplicatori di Lagrange  $\underline{\lambda} \in \mathbb{R}^m, \underline{\lambda} \geq 0$ , si definisce **rilassamento lagrangiano** del problema  $P$  il seguente problema di ottimizzazione:

$\begin{aligned} z = \max_{\underline{\mathbf{x}}} & \underline{\mathbf{c}}^T \underline{\mathbf{x}} \\ & D\underline{\mathbf{x}} \leq \underline{\mathbf{b}} \\ & \underline{\mathbf{x}} \in X \end{aligned}$ <p>(a) Problema <math>P</math></p>	$\begin{aligned} z_{RL} = \max_{\underline{\mathbf{x}}} & \underline{\mathbf{c}}^T \underline{\mathbf{x}} - \underline{\lambda}^T (D\underline{\mathbf{x}} - \underline{\mathbf{b}}) \\ & C\underline{\mathbf{x}} \leq \underline{\mathbf{d}} \\ & \underline{\mathbf{x}} \in X \end{aligned}$ <p>(b) Rilassamento lagrangiano di <math>P</math></p>
---	--

Figura 11.1: Il rilassamento lagrangiano

**Teorema 11.1.2 (Condizioni di integralità).** Dato un problema di PLI  $P$ , il suo rilassamento continuo  $RC$  ed il suo rilassamento lagrangiano  $RL$ , se questi ammettono soluzione ottima, per qualsiasi vettore dei costi della funzione obiettivo di  $P$  se:

$$\text{Conv}\{\underline{\mathbf{x}} \in \mathbb{Z}^n : C\underline{\mathbf{x}} \geq \underline{\mathbf{d}}, \underline{\mathbf{d}} \geq b\mathbf{m}0\} = \{\underline{\mathbf{x}} \in \mathbb{R}^n : C\underline{\mathbf{x}} \geq \underline{\mathbf{d}}, \underline{\mathbf{d}} \geq b\mathbf{m}0\}$$

Allora  $\underline{z}_{RL}^* = \underline{z}_{RC}^*$ , cioè il rilassamento lagrangiano non può ottenere una limitazione inferiore sul valore ottimo migliore di quella ottenibile dal rilassamento lineare.

**Teorema 11.1.3 (Condizioni di dualità lagrangiana debole).** Sia  $\underline{\bar{\mathbf{x}}}$  una soluzione ammissibile di  $P$ , avente valore  $\bar{z}_P$ . Vale che:

$$\bar{z}_P \geq \bar{z}_{RL}^* \quad \forall \underline{\lambda} \in \mathbb{R}^m, \lambda \geq 0$$

**Teorema 11.1.4 (Condizioni di ortogonalità lagrangiana).** Sia dato un problema di programmazione intera  $P$  ed il suo rilassamento lagrangiano  $RL$ , in corrispondenza di un dato vettore di moltiplicatori di Lagrange  $\underline{\lambda}^* \geq 0$ . Sia  $\underline{x}_{RL}^*$  la soluzione ottima di  $RL$ . Se vale che:

1.  $\underline{x}_{RL}^*$  è ammissibile per  $P$ , cioè  $A\underline{x}_{RL}^* \geq \underline{b}$
2.  $\underline{\lambda}^{*T} (A\underline{x}_{RL}^* - \underline{b}) = 0$

Allora la soluzione  $\underline{x}_{RL}^*$  è ottima anche per  $P$ .

**Corollario 11.1.4.1 (Condizioni di ortogonalità lagrangiana per vincoli di uguaglianza).** In questo caso se la soluzione ottima del rilassamento lagrangiano è ammissibile nel problema originario allora essa risulta anche essere ottima.

## 11.2 Risolvere il lagrangiano duale

In generale la funzione lagrangiana risulta non essere differenziabile (nei punti in cui cambia la pendenza). Si utilizza quindi al posto di un **gradiente** un **sottogradiente**.

**Definizione 11.2.1 (Sottogradiente).** Data una funzione concava  $f(\underline{\lambda}) : \mathbb{R}^m \rightarrow \mathbb{R}$  un vettore  $s(\underline{\lambda}') \in \mathbb{R}^m$  tale che:

$$f(\underline{\lambda}) \leq f(\underline{\lambda}') + s(\underline{\lambda}')^T (\underline{\lambda} - \underline{\lambda}') \forall \underline{\lambda} \in \mathbb{R}^m$$

viene detto **sottogradiente** di  $f$  in  $\underline{\lambda}' \in \mathbb{R}^m$ .

**Teorema 11.2.2 (Sottogradiente nel rilassamento lagrangiano).** Siano  $\underline{\lambda}^{(t)} \in \mathbb{R}^m$  e  $\underline{x}_{RL_{\underline{\lambda}^{(t)}}}^{*(t)} \in \mathbb{Z}^n$  due vettori che soddisfano la seguente condizione:

$$L^*(\underline{\lambda}^{(t)}) = \underline{c}^T \underline{x}^{(t)} - \underline{\lambda}^{(t)T} (A\underline{x}^{(t)} - \underline{b}) = \min_{\underline{x}} \{ \underline{c}^T \underline{x} - \underline{\lambda}^{(t)T} (A\underline{x} - \underline{b}) : C\underline{x} \geq \underline{d}; \underline{x} \geq 0; \underline{x} \in \mathbb{Z}^n \}$$

Cioè  $\underline{x}_{RL_{\underline{\lambda}^{(t)}}}^{*(t)}$  è la soluzione ottima del lagrangiano con vettore di coefficienti  $\underline{\lambda}^{(t)}$ . Allora il vettore  $s(\underline{\lambda}^{(t)}) = -(A\underline{x}^{(t)} - \underline{b})$  è un **sottogradiente** di  $L^*(\underline{\lambda})$  in  $\underline{\lambda}^{(t)}$ .

# 12

## Tecnica di generazione delle colonne

Questo problema nasce dalla necessità di fornire una tecnica di risoluzione a problemi dal numero di soluzioni esponenziali, che spesso risultano equivalenti: questo infatti manda in crisi gli elaboratori, che seguendo l'albero di ricerca devono considerare un numero esponenziale di soluzioni equivalenti. Il problema dei **cutting stock** ne è un esempio tipico.

### 12.1 Cutting stock

In questo problema è necessario tagliare delle assi (tutte di pari lunghezza) in modo da minimizzare il numero di assi utilizzate e soddisfare la richiesta totale.

#### 12.1.1 Formulazione classica

$\min \sum_{i=1}^m y_i$	Va minimizzato il numero di assi che vengono utilizzate.
$\sum_{j=1}^n l_j x_{ij} \leq L y_i \quad \forall i$	Da ogni assa $i$ si può realizzare al massimo un numero di prodotti la cui lunghezza totale non superi la lunghezza $L$ dell'assa
$\sum_{i=1}^n x_{ij} \leq d_j \quad \forall j$	Bisogna realizzare almeno un numero di prodotti per tipo pari alla domanda.
$x_{ij} \in \mathbb{Z}^+$	
$y_i \in \{0, 1\}$	

Figura 12.1: Cutting stock in formulazione classica

Il modello presenta un elevato grado di simmetria, cioè esistono tante soluzioni ottime equivalenti. Questo, come detto inizialmente, mette in crisi i risolutori ad albero.

#### 12.1.2 Formulazione estesa

Consideriamo ora una formulazione che considera come variabili  $z_h$  i possibili schemi di taglio (considerando unicamente gli schemi non dominati, cioè che garantiscono il massimo numero di prodotti dallo stesso numero di tavole). Questo schema viene

rappresentato da un vettore colonna  $\underline{\alpha} A_h = \begin{bmatrix} a_{1h} \\ \vdots \\ a_{nh} \end{bmatrix}$ , in cui ogni elemento  $a_{jh}$  rappresenta il numero di prodotti  $j$  ricavati dallo schema di taglio  $h$ .

$$\min \sum_{h=1}^H z_h$$

$$\sum_{h=1}^H a_{jh} z_h \geq d_j$$

$$z_h \in \{0, 1\} \quad \forall h$$

Vogliamo minimizzare il numero di assi utilizzate: ogni schema di taglio si applica ad una sola asse.

Il numero di prodotti realizzati deve rispondere alla domanda richiesta.

Figura 12.2: Cutting stock in formulazione estesa

Il numero  $H$  di schemi di taglio è solitamente esponenziale, per cui non è possibile elencare tutti i possibili  $z_h$  o le rispettive colonne  $A_h$ , per cui è necessaria adottare una tecnica per **generare** queste **colonne** in modo dinamico per poter arrivare a risolvere il rilassamento continuo del modello, che viene chiamato **Master Problem**.

## 12.2 Master Problem

Il master problem è il rilassamento continuo della formulazione estesa.

$$\min \sum_{h=1}^H z_h$$

$$\sum_{h=1}^H a_{jh} z_h \geq d_j$$

$$z_h \in [0, 1] \quad \forall h$$

Vogliamo minimizzare il numero di assi utilizzate: ogni schema di taglio si applica ad una sola asse.

Il numero di prodotti realizzati deve rispondere alla domanda richiesta.

Figura 12.3: Master Problem

## 12.3 Master Problem ridotto

Il master problem **ridotto** è una versione del master problem realizzata considerando un numero piccolo di colonne di  $\underline{A}$ , scelte in modo tale da continuare a rispettare i vincoli della domanda. Questo insieme ridotto viene indicato con  $\underline{\bar{A}}$ .

$$\min \sum_{h=1}^{\bar{H}} z_h$$

$$\sum_{h=1}^{\bar{H}} a_{jh} z_h \geq d_j$$

$$z_h \in [0, 1] \quad \forall h$$

Vogliamo minimizzare il numero di assi utilizzate: ogni schema di taglio si applica ad una sola asse.

Il numero di prodotti realizzati deve rispondere alla domanda richiesta.

Figura 12.4: Master Problem ridotto

## 12.4 Problema di Pricing

Una volta identificata una soluzione ottima  $z^*$  nel master problem ridotto è necessario verificare se essa risulti o meno ottima del master problem. Questo viene svolto cercando un **coefficiente di costo ridotto negativo** ottenuto tramite  $\bar{c}_h = c_h - \underline{\lambda}^{*T} \underline{A}_h$ , dove  $\underline{\lambda}^*$  è il vettore delle variabili duali ottime,  $c_h = 1$  (in questo caso) ed  $\underline{A}_h \in \underline{A} \setminus \underline{\bar{A}}$ . Il problema viene quindi formulato come:

$$\begin{aligned} \min & 1 - \sum_{j=1}^n \lambda_j^* x_j \\ & \sum_{j=1}^n l_j x_j \leq L \\ & \underline{x} \in \mathbb{Z}^n \end{aligned}$$

(a) Problema di Pricing

$$\begin{aligned} w^* &= \max \sum_{j=1}^n \lambda_j^* x_j \\ & \sum_{j=1}^n l_j x_j \leq L \\ & \underline{x} \in \mathbb{Z}^n \end{aligned}$$

(b) Problema di Pricing riscritto

Figura 12.5: Problema di Pricing

Se  $w^* > 1$  allora  $\underline{x}$  è il nuovo vettore colonna da aggiungere alla matrice  $\bar{A}$  e si ripete la ricerca dell'ottimo nel **master problem ridotto**, altrimenti se  $w^* \leq 1$  il vettore  $\underline{z}^*$  è soluzione ottima anche del **master problem**.

La procedura di generazione di colonne però si arresta solo quando il termine  $w^* \leq 0$ , altrimenti non si ha garanzie che la soluzione identificata sia ottima anche per la formulazione **intera**.

È importante notare che il problema di **pricing** sia risolvibile non necessariamente con tecniche di programmazione lineare intera, ma essendo un tipo di **problema dello zaino** è possibile utilizzare delle tecniche, anche euristiche, ad hoc. Solo alla fine è assolutamente necessario ricorrere a metodi esatti per verificare che non esistano soluzioni accettabili.