

Introduzione alla Ricerca Operativa

Ricerca operativa

Giovanni Righini



UNIVERSITÀ DEGLI STUDI
DI MILANO

Definizione

La **Ricerca Operativa**
(**Operations Research / Management Science**)

è il settore della matematica applicata
che studia

modelli matematici ed **algoritmi**
per la risoluzione di **problemi decisionali**
(problemi di **ottimizzazione**).



La ricerca operativa e la matematica

In ricerca operativa la matematica viene usata come un **linguaggio** per descrivere **modelli di problemi decisionali**.

Le **proprietà del modello matematico** sono il punto di partenza per la progettazione di opportuni **algoritmi risolutivi**.

A differenza della statistica, la ricerca operativa non parte dai **dati**, ma dal **modello** del problema.

A differenza della simulazione numerica, la ricerca operativa non studia modelli (matematici) di **sistemi fisici**, ma modelli (matematici) di **problemi decisionali**.

Modelli matematici

Fisica, ingegneria

- Sistemi naturali o artificiali
- Descritti da equazioni
- Soluzione unica
- Senza decisioni né obiettivi

Ricerca operativa

- Problemi decisionali
- Descritti da disequazioni
- Molte soluzioni possibili
- Obiettivo/i da ottimizzare

Perché i modelli matematici?

A fronte di un problema da risolvere, non si deve mai by-passare la fase della formulazione matematica.

Per molti motivi:

- Per comprendere davvero il problema.
- Per comunicarlo ad altri (incluso il calcolatore).
- Per classificarlo.
- Per comprenderne la complessità.
- Per capire quale tipo di metodo è meglio usare.
- Per poter eventualmente usare software già pronto.
- Per riconoscere sottoproblemi e scomporlo.
- Per mantenere distinto il problema dal metodo risolutivo.

La ricerca operativa e l'informatica

La **ricerca operativa** non si propone di sviluppare nuova tecnologia, ma di utilizzare nel modo migliore quella esistente.

E' quindi più vicina alla **computer science** che all'**information technology**.

La **ricerca operativa** è la matematica degli algoritmi di ottimizzazione, che possono essere anche molto sofisticati ed il cui sviluppo richiede tipicamente eccellenti doti di programmazione.

Algoritmi vs tecnologia

LP Progress: An Example



A Production Planning Model

401,640 cons. 1,584,000 vars. 9,498,000 nonzeros

Solution time line (2.0 GHz P4):		Speedup
▪ 1988 (CPLEX 1.0):	29.8 days	1x
▪ 1997 (CPLEX 5.0):	1.5 hours	480x
▪ 2003 (CPLEX 9.0):	59.1 seconds	43500x

Solving IN 1988: 82 years (machines 1000x slower)

La ricerca operativa e i Big Data

L'obiettivo della **ricerca operativa** è di supportare i **processi decisionali**, utilizzando al meglio i **dati** disponibili (in forma digitale).



Dati \Rightarrow **Decisioni**



Per sviluppare un progetto di ricerca operativa non servono necessariamente **big data**, ma piuttosto i right **data**.

Attributi di una buona decisione

La **ricerca operativa** supporta il decisore nel prendere una **decisione**

- **efficace**: raggiunge lo scopo;
- **efficiente**: raggiunge lo scopo consumando poche risorse;
- **tempestiva**: coerente con l'orizzonte temporale del livello decisionale (strategico, tattico, operativo);
- **robusta**: rimane buona (per lo meno ammissibile) anche in seguito a piccole variazioni nel valore dei dati;
- **giustificabile**: può essere dimostrata razionale ad altri.

Ricerca operativa e processi decisionali

INTELLIGENZA

Svelare il nesso
tra azioni ed effetti

Calcolatore
(non responsabile)

LIBERTA'

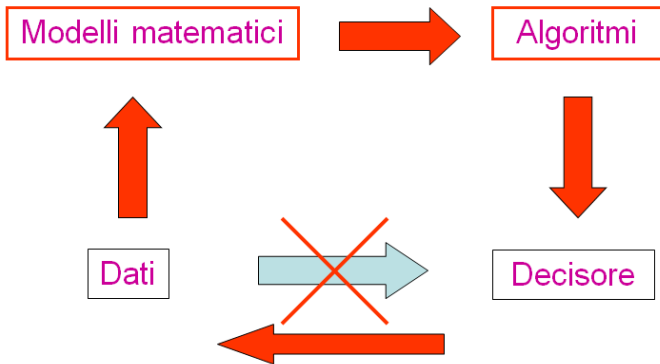
Scegliere un'azione
tra quelle possibili

Uomo
(responsabile)

Una scelta è tanto più libera e razionale quanto più è informata.

Il tipico prodotto di un progetto di ricerca operativa è un sistema di supporto alle decisioni (DSS: Decision Support System)

Supporto alle decisioni



Il **decisore** non viene sostituito dal **DSS**.

Un po' di storia

La Ricerca Operativa nacque in Gran Bretagna durante la II guerra mondiale da un gruppo di studio multidisciplinare denominato "Research on Military Operations".



Patrick Blackett (1897-1974)
Premio Nobel per la Fisica (1948)

Un po' di storia

Lo scopo era di affrontare con metodi scientifico alcuni problemi di logistica militare.

- Dove localizzare i radar per sorvegliare nel modo migliore lo spazio aereo sulla Manica in previsione di attacchi aerei della Luftwaffe?
- Come comporre le squadriglie di piloti della Royal Air Force per ingaggiare battaglie aeree?
- Come dimensionare i convogli di navi per attraversare l'Atlantico in modo da minimizzare gli effetti degli attacchi dei sottomarini tedeschi?
- A quale profondità far esplodere le cariche anti-sommergibile?

I risultati di questi primi studi ebbero un effetto decisivo per la vittoria degli Alleati nella Battaglia d'Inghilterra (1940-41) e per l'esito finale della seconda guerra mondiale.

Un po' di storia

Terminata la guerra, la Ricerca Operativa venne progressivamente applicata in ogni ambito civile, industriale ed economico, sviluppandosi di pari passo con la **Computer Science**.

Si evidenziarono due settori principali:
Operations Research (ingegneria) e
Management Science (economia).

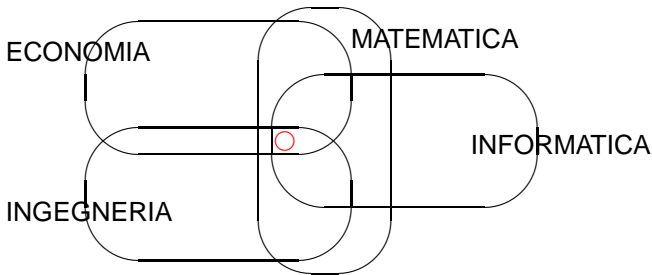
L'**Associazione Italiana di Ricerca Operativa** fu fondata nel 1961.



George B. Dantzig
(1914-2005)

Negli USA negli anni Novanta le due comunità scientifiche **ORSA = Op. Res. Soc. Of America** e **TIMS = The Inst. of Mgmt. Sc.** si fusero, originando l'attuale **INFORMS (Institute For Operations Research and the Management Sciences)**, www.informs.org.

Interdisciplinarietà



Ricerca Operativa è il nome disciplinare dell'intersciplinarietà.
Nelle università di tutto il mondo i ricercatori in **ricerca operativa** si possono trovare indifferentemente in dipartimenti di **matematica**, **informatica**, **ingegneria** o **economia**.

La Ricerca Operativa fino a ieri

Fino a pochi anni fa, la ricerca operativa era più conosciuta e sviluppata...

- ...nel mondo anglosassone,
- ...negli enti militari,
- ...in alcune grandi aziende (compagnie aeree, case automobilistiche,...),

mentre era meno conosciuta e sviluppata...

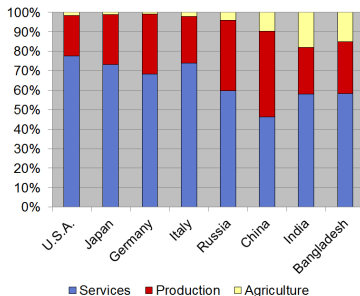
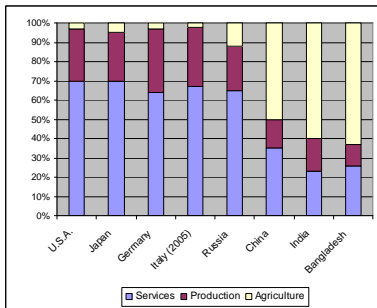
- ...in Italia,
- ...presso le amministrazioni pubbliche,
- ...nelle piccole e medie imprese,
- ...presso l'opinione pubblica.

Fattori di sviluppo

Esistono alcuni fattori **di scala mondiale** e **di lungo termine** che da alcuni anni spingono fortemente lo sviluppo della Ricerca Operativa.

- La **globalizzazione dei mercati** richiede maggiore competitività nel settore privato.
- L'**integrazione europea** richiede maggiore efficienza nel settore pubblico.
- L'**emergenza ambientale ed energetica** pone problemi complessi, ineludibili.
- Lo spostamento dell'economia dalla produzione ai **servizi** richiede un approccio scientifico a problemi nuovi.
- Esiste una disponibilità senza precedenti di **dati (ICT, big data)** e di **software** dedicato (simulazione, ottimizzazione,...).

La service-based economy



Più del 70% del PIL in Occidente proviene dal settore dei servizi
Service Science Management & Engineering (IBM, 2005).

La piramide del valore



Il valore aumenta dal livello **DATI** al livello **DECISIONI**

Data value spectrum



Analytics

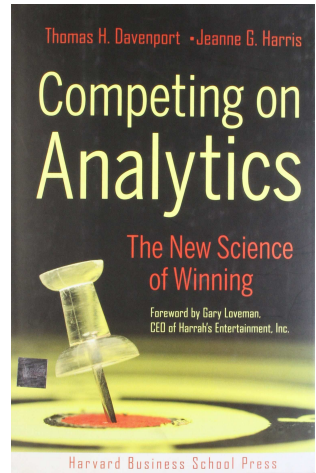
T.H. Davenport, J.G. Harris (2007)
Competing on Analytics: the New Science of Winning

I. Ayres (2007)
Super Crunchers: Why Thinking-by-Numbers is the New Way to Be Smart

S. Baker (2008)
The Numerati

T. May (2009)
The New Know: Innovation Powered by Analytics

...e molti altri a seguire.



Analytics

Gli 8 livelli di **analytics** secondo SAS:

1. *Standard reports*: Rapporti riassuntivi periodici su un sistema
2. *Ad hoc reports*: Rapporti specifici su un argomento/sotto-sistema
3. *Domande di approfondimento*: Ordinamento ed esplorazione dei dati, identificazione problemi
4. *Allerte*: Segnalazioni automatiche di problemi specifici
5. *Analisi statistica* nello spazio e nel tempo; valori medi, varianze, distribuzioni
6. *Previsioni*: analisi di serie temporali, modelli di evoluzione di un sistema
7. *Modelli predittivi*: simulazione, teoria delle code,...
8. *Ottimizzazione*: programmazione matematica.

Modelli **descrittivi**, **predittivi**, **prescrittivi**.

Smarter Planet (IBM, 2008)

<http://www.ibm.com/think>



Smart traffic
systems



Intelligent
oil field
technologies



Smart food
systems



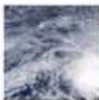
Smart
healthcare



Smart energy
grids



Smart retail



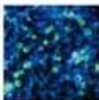
Smart water
management



Smart supply
chains



Smart
countries



Smart
weather

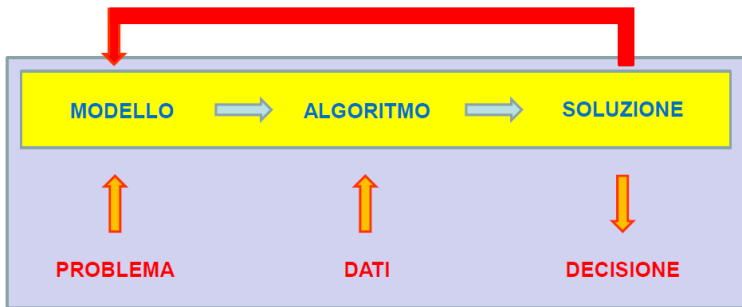


Smart
regions



Smart cities

Algoritmi intelligenti o decisori intelligenti?



Il **modello** ed i **dati** devono sempre essere soggetti a continuo miglioramento. Solo così si genera **conoscenza**.

Sbocchi occupazionali

The Best and Worst Jobs

The Best	The Worst
1. Mathematician	200. Lumberjack
2. Actuary	199. Dairy Farmer
3. Statistician	198. Taxi Driver
4. Biologist	197. Seaman
5. Software Engineer	196. EMT
6. Computer Systems Analyst	195. Roofer
7. Historian	194. Garbage Collector
8. Sociologist	193. Welder
9. Industrial Designer	192. Roustabout
10. Accountant	191. Ironworker
11. Economist	190. Construction Worker
12. Philosopher	189. Mail Carrier
13. Physicist	188. Sheet Metal Worker
14. Parole Officer	187. Auto Mechanic
15. Meteorologist	186. Butcher
16. Medical Laboratory Technician	185. Nuclear Decontamination Tech
17. Paralegal Assistant	184. Nurse (LN)
18. Computer Programmer	183. Painter
19. Motion Picture Editor	182. Child Care Worker
20. Astronomer	181. Firefighter

Professioni matematiche

Le professioni valutate sono definite così.

- **Mathematician:** Applies mathematical theories and formulas to teach or solve problems in a business, educational or industrial climate.
- **Actuary:** Interprets statistics to determine probabilities of accidents, sickness and death and loss of property from theft and natural disasters.
- **Statistician:** Tabulates, analyzes and interprets the numerical results of experiments and surveys.

Sono rispettivamente gli esperti di modelli **prescrittivi**, **predittivi** e **descrittivi**.

Best jobs 2017 (Careercast.com)

Rank	Job
1	Statistician
2	Medical services manager
 3	Operations research analyst
4	Information security analyst
5	Data scientist
6	University professor
7	Mathematician
8	Software engineer

Alcuni siti

AIRO - Associazione Italiana Ricerca Operativa
www.airo.org

EURO - Associazione Europea di R.O.
www.euro-online.org

INFORMS - INstitute For Operations Research and the Management
Sciences
www.informs.org

Informazioni sul corso

Il corso di Ricerca Operativa

Nel vostro curriculum il corso di R.O. ha lo scopo di:

- spostare il fuoco dai *metodi* ai *problemi* e dai *calcolatori* alle loro *applicazioni*;
- indicare l'esistenza di una figura professionale ben precisa, con possibilità di lavoro sia dipendente che autonomo, con apertura tanto verso le applicazioni quanto verso la ricerca scientifica;
- collegare tra loro discipline diverse (matematica del continuo e del discreto, programmazione, algoritmi e strutture-dati, calcolo delle probabilità e statistica, calcolo numerico,...)

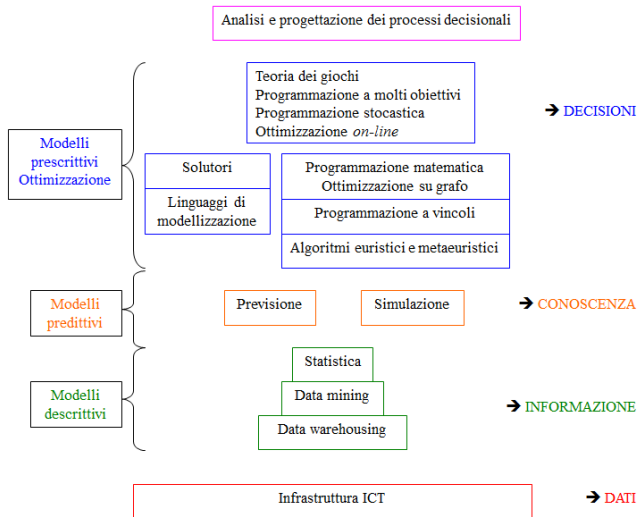
Il corso contiene una panoramica “in larghezza”, non “in profondità”.

Il corso di Ricerca Operativa

Al termine di questo corso:

- saprete riconoscere un problema di ottimizzazione quando ne incontrate uno;
- saprete classificarlo;
- saprete scriverne il modello matematico;
- saprete eventualmente risolverlo usando solutori *general-purpose*;
- non avrete imparato nei dettagli alcun algoritmo;
- non avrete imparato a progettare e realizzare algoritmi di ottimizzazione;
- non sarete diventati esperti di alcun settore applicativo in particolare.

Una panoramica sulla ricerca operativa



Il percorso *Analytics and Optimization*

Nella laurea magistrale in informatica è attivo un percorso denominato *Analytics and Optimization* nel quale vengono erogati i seguenti insegnamenti (in lingua inglese), per i quali il corso di Ricerca Operativa è propedeutico:

- Complements of operational research: algoritmi di ottimizzazione per problemi NP-hard.
- Combinatorial optimization: algoritmi per problemi di ottimizzazione su grafo polinomiali.
- Heuristic algorithms: algoritmi euristici e di approssimazione per problemi NP-hard.
- Decision methods and models: programmazione a molti obiettivi e teoria dei giochi.
- Simulation: simulazione a eventi discreti e ad agenti.
- Logistics: modelli di previsione, teoria delle code, gestione delle scorte, ottimizzazione logistica.

Programmazione matematica

I problemi di decisione possono essere classificati in base a tre caratteristiche principali:

- Numero di obiettivi
- Numero di decisori
- Grado di incertezza

Consideriamo i problemi con un solo obiettivo, un solo decisore, deterministici.

La programmazione matematica presuppone la formulazione del problema in termini di **modello matematico**.

Modelli di programmazione matematica

Gli ingredienti fondamentali di un modello di programmazione matematica sono:

- Dati
- Variabili
- Vincoli
- Funzione obiettivo

Un **algoritmo** calcola una **soluzione**, cioè un'assegnamento di valori alle variabili.

Nei problemi di **esistenza** si vuole trovare una soluzione **ammissibile**, cioè tale da soddisfare tutti i vincoli.

Nei problemi di **ottimizzazione** si vuole trovare una soluzione **ottima**, cioè tale da massimizzare/minimizzare la funzione obiettivo.

Esempio

Cioè una funzione definita da un polinomio di grado 1, il cui grafico è rappresentato da una retta

minimize $f(x)$
subject to $x \in \mathcal{X}$

Se **obiettivo** e **vincoli** sono rappresentati da **funzioni lineari** delle **variabili**, il problema è di **programmazione lineare**.

Altrimenti è di **programmazione non-lineare**.

Se le **variabili** sono vincolate ad assumere valori interi (o addirittura binari), allora il problema è di **ottimizzazione discreta**.

Il programma del corso

Il corso è suddiviso in 4 parti:

1. Programmazione lineare:

- proprietà fondamentali della PL
- soluzione per via geometrica
- forma standard e algoritmo del simplesso
- interpretazione economica della PL
- analisi post-ottimale
- teoria della dualità

2. Programmazione lineare a due obiettivi:

- soluzioni Pareto-ottime
- metodo dei pesi e metodo dei vincoli

3. Programmazione lineare intera:

- proprietà fondamentali della PLI
- rilassamenti
- branch-and-bound

4. Programmazione non-lineare:

- ottimalità locale e globale
- metodi iterativi e loro proprietà

Testi di riferimento

- A. Colorni, *Ricerca Operativa*, Ed. Zanichelli, 1984
- C. Vercellis, *Modelli e decisioni*, Progetto Leonardo, Ed. Esculapio, Bologna 1997
- Hillier, Liebermann, *Introduzione alla Ricerca Operativa*, Franco Angeli, 1999
- S. Martello, *Lezioni di ricerca operativa*, Progetto Leonardo, 2002
- R. Tadei, *Elementi di Ricerca Operativa*, Progetto Leonardo 2005
- M. Pappalardo, *Lezioni di Ricerca Operativa*, 2006
- P. Serafini, *Ricerca Operativa*, 2009
- M. Pappalardo, *Ricerca Operativa*, Pisa University Press, 2010
- M. Bruglieri, *Ricerca Operativa*, Zanichelli, 2012
- M. Fischetti, *Lezioni di ricerca operativa*, Libreria progetto, 2014

Il corso di Ricerca Operativa

L'esame consiste in una **prova al calcolatore** e in una **prova orale** che pesa 6/30.

Tradizionalmente la prova scritta di R.O. è concepita così:

- dato un problema già classificato e modellizzato,
- dato un esempio piccolo,
- calcolare a mano la soluzione, applicando l'algoritmo opportuno.

Corrisponde a fare in piccolo ciò che è **compito del calcolatore**.

Nel nostro corso invece la prova scritta di R.O. è concepita così:

- dato un problema realistico descritto a parole,
- dato un esempio "grande" (= non risolubile a mano),
- scriverne il modello e classificarlo,
- scegliere lo strumento software opportuno,
- preparare l'input, interpretare l'output.

Corrisponde al **compito del ricercatore operativo**.

Laboratorio

La parte più importante del corso non è tanto quella teorica, quanto quella che si sviluppa in **laboratorio** e che serve a sviluppare **competenze di modellistica matematica dei problemi decisionali**.

Sulla webpage del corso sono disponibili molti **esercizi d'esame risolti e commentati**.

Useremo come strumenti:

- il foglio elettronico con il componente aggiuntivo "Risolutore";
- il solutore (gratuito) *glpsol* con l'interfaccia *gusek*.

Programmazione lineare

Ricerca operativa

Giovanni Righini



UNIVERSITÀ DEGLI STUDI
DI MILANO

Programmazione lineare (PL)

Un problema è di **programmazione lineare** (Linear Programming) quando:

- le **variabili** hanno un dominio **continuo**;
- i **vincoli** sono **equazioni e disequazioni lineari**;
- la **funzione obiettivo** è una **funzione lineare delle variabili**.

Nella sua **forma generale** un problema di PL si presenta così:

$$\text{maximize/minimize } z = cx \quad (1)$$

$$\text{subject to } A_1x \geq b_1 \quad (2)$$

$$A_2x \leq b_2 \quad (3)$$

$$A_3x = b_3 \quad (4)$$

$$x' \geq 0 \quad (5)$$

$$x'' \text{ libere} \quad (6)$$

I vincoli possono essere di tipo \leq , \geq o $=$.

Alcune variabili possono essere vincolate a valori **non-negativi**.

Forma alle disuguaglianze

I problemi di PL possono essere riformulati nella forma “**alle disuguaglianze**”, che è utile per l'interpretazione geometrica del problema. Per passare dalla **forma generale** alla **forma alle**

disuguaglianze, occorre eliminare dal modello i **vincoli di uguaglianza** e le **variabili libere**.

Eliminazione vincoli di uguaglianza

I vincoli di uguaglianza si possono eliminare semplicemente per sostituzione. Ad esempio:

$$\begin{array}{llllll} \text{maximize } z = & 3x_1 & -2x_2 & +5x_3 & & \\ \text{s.t.} & 2x_1 & -x_2 & -x_3 & \leq & 8 \\ & 3x_1 & +2x_2 & & \geq & -6 \\ & & x_2 & -2x_3 & = & 1 \\ & & x_2, & x_3 & \geq & 0 \end{array}$$

Da $x_2 - 2x_3 = 1$ si ricava $x_2 = 2x_3 + 1$.
Sostituendo x_2 nel modello si ottiene:

$$\begin{array}{llllll} \text{maximize } z = & 3x_1 & +x_3 & -2 & & \\ \text{s.t.} & 2x_1 & -3x_3 & \leq & 9 & \\ & 3x_1 & +4x_3 & \geq & -8 & \\ & & 2x_3 & \geq & -1 & \\ & & x_3 & \geq & 0 & \end{array}$$

Eliminazione variabili libere

Le variabili libere si possono eliminare sostituendole con la differenza tra due variabili non-negative. Ad esempio, ponendo $x_1 = x_4 - x_5$ nel modello:

$$\begin{array}{llll} \text{maximize } z = & 3x_1 & +x_3 & -2 \\ \text{s.t.} & 2x_1 & -3x_3 & \leq 9 \\ & 3x_1 & +4x_3 & \geq -8 \\ & & 2x_3 & \geq -1 \\ & & x_3 & \geq 0 \end{array}$$

si ottiene

$$\begin{array}{llllll} \text{maximize } z = & +x_3 & +3x_4 & -3x_5 & -2 \\ \text{s.t.} & -3x_3 & +2x_4 & -2x_5 & \leq 9 \\ & 4x_3 & +3x_4 & -3x_5 & \geq -8 \\ & 2x_3 & & & \geq -1 \\ & x_3, & x_4, & x_5 & \geq 0 \end{array}$$

Forma alle disuguaglianze

$$\begin{array}{llllll} \text{maximize } z = & +x_3 & +3x_4 & -3x_5 & -2 & \\ \text{s.t.} & -3x_3 & +2x_4 & -2x_5 & \leq & 9 \\ & 4x_3 & +3x_4 & -3x_5 & \geq & -8 \\ & 2x_3 & & & \geq & -1 \\ & x_3, & x_4, & x_5 & \geq & 0 \end{array}$$

I termini costanti nella f.o. possono essere trascurati.

I vincoli ridondanti possono essere eliminati.

Tutte le disequazioni devono essere coerenti in segno e opposte all'obiettivo:

- massimizzazione con vincoli di \leq ;
- minimizzazione con vincoli di \geq .

$$\begin{array}{llllll} \text{maximize } w = & +x_3 & +3x_4 & -3x_5 & & \\ \text{s.t.} & -3x_3 & +2x_4 & -2x_5 & \leq & 9 \\ & -4x_3 & -3x_4 & +3x_5 & \leq & 8 \\ & x_3, & x_4, & x_5 & \geq & 0 \end{array}$$

Rappresentazione matriciale

$$\begin{array}{ll} \text{maximize } w = & +x_3 \quad +3x_4 \quad -3x_5 \\ \text{s.t.} & -3x_3 \quad +2x_4 \quad -2x_5 \leq 9 \\ & -4x_3 \quad -3x_4 \quad +3x_5 \leq 8 \\ & x_3, \quad x_4, \quad x_5 \geq 0 \end{array}$$

Lo stesso modello si può rappresentare in modo più compatto usando la notazione matriciale.

$$\begin{array}{ll} \text{maximize } w = & c^T x \\ \text{s.t.} & Ax \leq b \\ & x \geq 0 \end{array}$$

dove

$$c^T = [\ 1 \quad 3 \quad -3 \]$$
$$A = \begin{bmatrix} -3 & 2 & -2 \\ -4 & -3 & 3 \end{bmatrix} x = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} \quad b = \begin{bmatrix} 9 \\ 8 \end{bmatrix}$$

Interpretazione geometrica della PL

Ogni **soluzione** x è un assegnamento di valore alle variabili. Quindi corrisponde ad un punto in uno spazio continuo ad n dimensioni, dove n è il numero di **variabili** nel modello.

Ogni **vincolo di uguaglianza** $ax = b$ corrisponde ad un **iperpiano**.

Ogni **vincolo di disuguaglianza** $ax \leq b$ corrisponde ad un **semispazio**.

Il **sistema dei vincoli** nel modello alle disuguaglianze corrisponde all'**intersezione dei corrispondenti semispazi**.

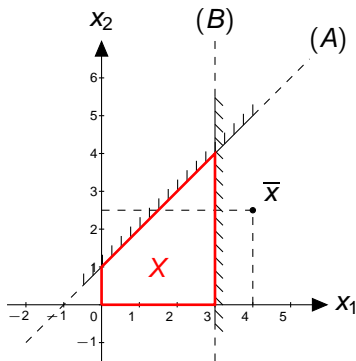
L'intersezione di semispazi è un **poliedro**.

I semispazi sono **convessi**.

L'intersezione di insiemi convessi è un insieme convesso.

Quindi **i poliedri sono convessi**.

Interpretazione geometrica della PL



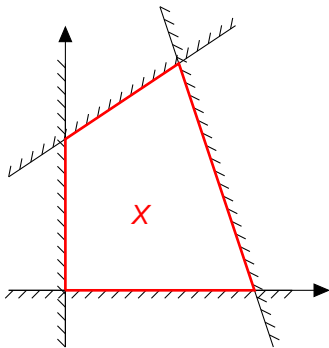
$$n = 2$$

$$\bar{x} = \begin{bmatrix} 4 \\ 2.5 \end{bmatrix}$$

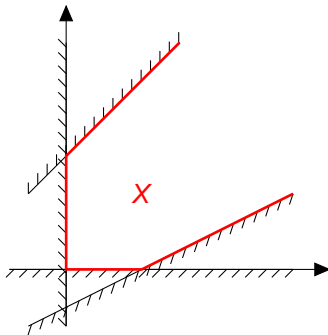
Regione ammissibile:

$$\text{X} = \begin{cases} -x_1 + x_2 \leq 1 & (A) \\ x_1 \leq 3 & (B) \\ x \geq 0 \end{cases}$$

Interpretazione geometrica della PL

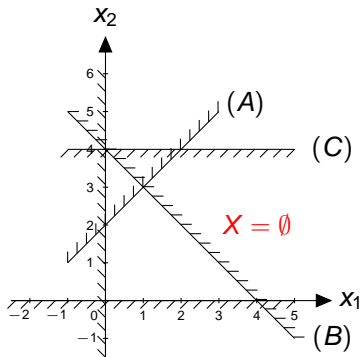


Poliedro limitato (politopo)



Poliedro illimitato

Interpretazione geometrica della PL

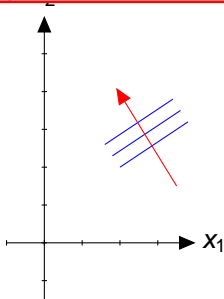


Poliedro vuoto

$$X = \begin{cases} -x_1 + x_2 \leq 2 & (A) \\ x_1 + x_2 \leq 4 & (B) \\ x_2 \geq 4 & (C) \\ x \geq 0 \end{cases}$$

Interpretazione geometrica della PL

Tutti i punti che giacciono sulla retta hanno lo stesso valore di z . La freccia indica in che direzione si sta migliorando.



$$\text{minimize } z = 2x_1 - 3x_2$$

Poiché la funzione obiettivo è lineare, tutte le **soluzioni equivalenti** giacciono su uno stesso **iperpiano**.

La funzione obiettivo corrisponde ad un **fascio di iperpiani paralleli**, ordinati come i corrispondenti valori dell'obiettivo.

La **direzione di ottimizzazione** (cioè minimizzazione o massimizzazione) definisce l'ordinamento degli iperpiani del fascio.

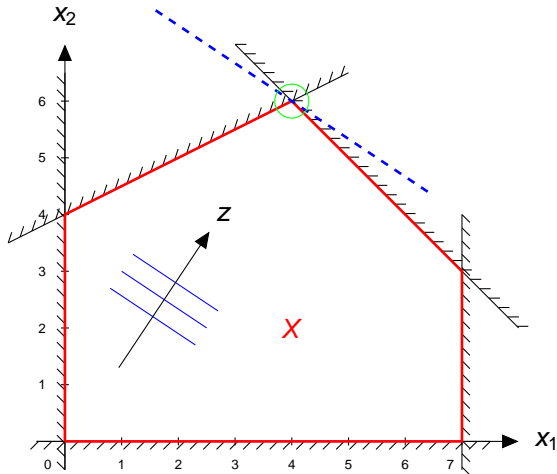
La direzione della freccia in questo caso è in alto a sx dato che devo minimizzare e il coefficiente angolare di x_1 è >0 e quello di x_2 è <0

Interpretazione geometrica della PL

Per la **convessità del poliedro** che rappresenta la regione ammissibile e per la **linearità delle curve di livello** della funzione obiettivo, possono darsi tre casi:

- **il poliedro è vuoto**: non esistono soluzioni ammissibili;
- **il poliedro è illimitato** nella direzione di ottimizzazione: non esiste un valore ottimo finito;
- esiste almeno un **vertice del poliedro** che corrisponde al **valore ottimo**.

Interpretazione geometrica della PL



Forma standard

$$\begin{array}{llllll} \text{maximize } w = & +x_1 & +3x_2 & -3x_3 & & \\ \text{s.t.} & -3x_1 & +2x_2 & -2x_3 & \leq & 9 \\ & -4x_1 & -3x_2 & +3x_3 & \leq & 8 \\ & x_1, & x_2, & x_3 & \geq & 0 \end{array}$$

La funzione obiettivo viene posta in forma di **minimizzazione**.

Tutti i vincoli di disuguaglianza vengono posti in forma di **uguaglianza**, introducendo opportune **variabili non-negative di scarto (slack)** o di **surplus**.

$$\begin{array}{llllllll} \text{minimize } z = & -x_1 & -3x_2 & +3x_3 & & & & \\ \text{s.t.} & -3x_1 & +2x_2 & -2x_3 & +x_4 & & = & 9 \\ & -4x_1 & -3x_2 & +3x_3 & & +x_5 & = & 8 \\ & x_1, & x_2, & x_3, & x_4, & x_5, & \geq & 0 \end{array}$$

Forma standard

Mettendo in forma standard un problema alle disuguaglianze con m vincoli e n variabili si ottiene un modello con m vincoli e $n + m$ variabili, **tutte non-negative**.

Il sistema dei vincoli è un sistema di m equazioni lineari in $n + m$ variabili.

Se non ci sono vincoli ridondanti, la matrice dei coefficienti ha rango m .

Il sistema quindi ha una soluzione univocamente determinabile se eliminiamo gli n gradi di libertà in eccesso, fissando n variabili.

Ad ogni **variabile nulla** nella forma standard corrisponde un **vincolo attivo** nella forma alle disuguaglianze.

Fissare n variabili a 0 nella forma standard corrisponde a scegliere un punto in cui n vincoli sono attivi nella forma alle disuguaglianze.

Soluzioni di base

Una base è un sottinsieme di m variabili scelte tra le $n + m$ della forma standard.

$$[B \mid N]$$

Il numero di basi è combinatorio: cresce esponenzialmente con m e n .

Una volta scelta la base, il sistema si può riscrivere come

$$Bx_B + Nx_N = b$$

La soluzione del sistema $m \times m$ che si ottiene dopo aver fissato a 0 tutte le n variabili fuori base è una **soluzione di base**.

Per ottenerla bisogna invertire la matrice B formata dalla base.

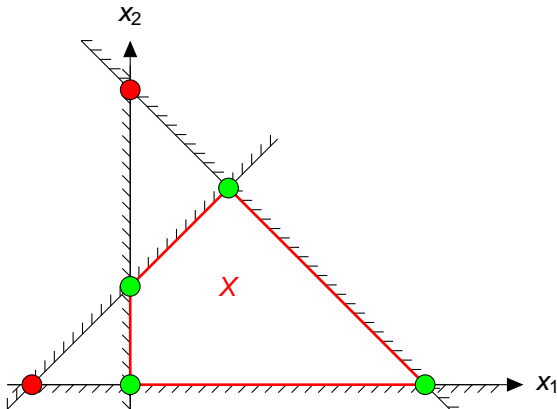
$$x_B = B^{-1}b - B^{-1}Nx_N$$

da cui

$$x_N = 0 \quad x_B = B^{-1}b.$$

Soluzioni di base

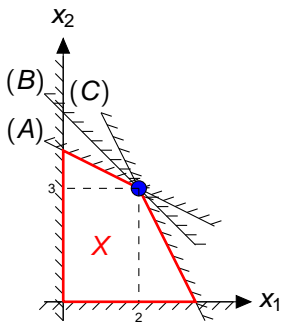
Tutti i vertici del poliedro sono soluzioni di base ma non è detto il viceversa: esistono anche soluzioni di base non ammissibili (quando $x_B \not\geq 0$).



Degenerazione

Quando una **variabile in base** risulta avere valore nullo, si ha **degenerazione**: più **soluzioni di base** coincidono.

In altri termini, più di n vincoli sono attivi nello stesso punto in uno spazio ad n dimensioni.



$$\begin{array}{llll} \text{minimize } z = & -x_1 & -x_2 & \\ \text{s.t.} & x_1 & +2x_2 & \leq 8 \quad (A) \\ & x_1 & +x_2 & \leq 5 \quad (B) \\ & 2x_1 & +x_2 & \leq 7 \quad (C) \\ & x_1, & x_2 & \geq 0 \end{array}$$

La soluzione $x = [2 \ 3 \ 0 \ 0 \ 0]$ corrisponde alle basi $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 2, 5\}$.

Teorema fondamentale della PL

Dato un problema lineare in forma standard

$$z = \min\{c^T x : Ax = b, x \geq 0\}$$

con A di rango m

- se esiste una **soluzione ammissibile**,
esiste anche una **soluzione ammissibile di base**;
- se esiste una **soluzione ottima**,
esiste anche una **soluzione ottima di base**.

Perciò un problema lineare nel **continuo** può essere risolto come problema combinatorio (**discreto**), limitandosi a considerare solo le **soluzioni di base**.

Metodi risolutivi

La complessità computazionale della programmazione lineare è **polinomiale**, tramite l'**algoritmo dell'ellissoide** (Khachiyan, 1979). Il metodo di gran lunga più diffuso per risolvere i problemi di PL però è l'**algoritmo del simplesso** (Dantzig, 1947).

L'algoritmo del simplesso non dà garanzia di terminare in un numero di iterazioni limitato da un polinomio nelle dimensioni dell'esempio, ma in pratica è molto veloce. Ne esistono diverse versioni e molte implementazioni, anche estremamente sofisticate.

L'algoritmo garantisce di terminare in un **numero finito di passi**, garantendo una di queste tre situazioni:

- la soluzione corrente è **ottima**;
- non esiste soluzione ammissibile (problema inammissibile);
- non esiste soluzione ottima finita (problema illimitato).

L'algoritmo del simplesso procede iterativamente da una **soluzione di base** ad una **adiacente**.

L'algoritmo del simplesso

Ricerca operativa

Giovanni Righini



UNIVERSITÀ DEGLI STUDI
DI MILANO

Forma canonica

Dalla forma standard di un problema di PL

$$\begin{aligned} \text{minimize } z &= c^T x \\ \text{subject to } Ax &= b \\ x &\geq 0 \end{aligned}$$

scegliendo una base (e permutando le colonne di conseguenza) si ha

$$\begin{aligned} \text{minimize } z &= c_B^T x_B + c_N^T x_N \\ \text{subject to } Bx_B + Nx_N &= b \\ x_B, x_N &\geq 0. \end{aligned}$$

Moltiplicando a sinistra per B^{-1} :

$$\begin{aligned} \text{minimize } z &= c_B^T x_B + c_N^T x_N \\ \text{subject to } Ix_B + (B^{-1}N)x_N &= B^{-1}b \\ x_B, x_N &\geq 0. \end{aligned}$$

da cui si ha la soluzione di base $x_B = B^{-1}b - (B^{-1}N)x_N$.

Forma canonica

Sostituendo $x_B = B^{-1}b - (B^{-1}N)x_N$ in z si ha:

$$\begin{aligned} \text{minimize } z &= c_B^T B^{-1}b + (c_N^T - c_B^T B^{-1}N)x_N \\ \text{subject to } Ix_B + (B^{-1}N)x_N &= B^{-1}b \\ x_B, x_N &\geq 0. \end{aligned}$$

che si può riscrivere in modo più compatto

$$\begin{aligned} \text{minimize } z &= z_B + \bar{c}_N^T x_N \\ \text{subject to } Ix_B + \bar{N}x_N &= \bar{b} \\ x_B, x_N &\geq 0. \end{aligned}$$

Quando si pone $x_N = 0$ si ha $x_B = B^{-1}b = \bar{b}$. Se $\bar{b} \geq 0$, allora la soluzione di base è ammissibile.

Forma canonica

Esistono un numero combinatorio di forme canoniche: tante quante le possibili scelte della base.

$$\begin{aligned} \text{minimize } z &= z_B + \bar{c}_N^T x_N \\ \text{subject to } Ix_B + \bar{N}x_N &= \bar{b} \\ x_B, x_N &\geq 0. \end{aligned}$$

Un problema di PL è in forma canonica se e solo se:

- i coefficienti delle variabili di base x_B formano una matrice identità $m \times m$;
- le variabili di base x_B non compaiono nella funzione obiettivo.

Inoltre la forma canonica è forte se e solo se:

- i termini noti dei vincoli sono non-negativi ($\bar{b} \geq 0$).

Una forma canonica debole corrisponde ad una soluzione di base non ammissibile.

Il *tableau*

Una volta posto in forma canonica, un problema di PL si può rappresentare in una matrice, detta *tableau*, che è la struttura dati fondamentale sulla quale opera l'algoritmo del simplesso.

Per esempio:

$$\begin{array}{llllll}
 \text{minimize } z = & -x_1 & -2x_2 & & & \\
 \text{s.t.} & -x_1 & +2x_2 & +x_3 & & = 8 \\
 & x_1 & +x_2 & & +x_4 & = 10 \\
 & x_1 & & & & +x_5 = 7 \\
 & x \geq 0 & & & &
 \end{array}$$

è in forma canonica con le variabili di slack in base.

0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

$-z_B$	\bar{c}_N^T	0
\bar{b}	\bar{N}	I

L'algoritmo

```
while ( $\neg$ Infeasible(b, c))  $\wedge$  ( $\neg$ FeasibleBase(b)) do  
    Pivot(A, b, c)  
if Infeasible(b, c) then  
    Stop: problema inammissibile  
else  
    while ( $\neg$ Optimal(c))  $\wedge$  ( $\neg$ Unbounded(A, c)) do  
        Pivot(A, b, c)  
    if Optimal(c) then  
        Stop: soluzione ottima  
    else  
        Stop: problema illimitato
```

$Pivot(A, b, c)$

Ogni iterazione (*pivoting*) consiste in un cambio di base: una variabile in base esce dalla base e una variabile fuori base entra in base.

0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

variabili fuori
base (?)

$x = \begin{bmatrix} 0 \\ 0 \\ 8 \\ 10 \\ 7 \end{bmatrix}$

$z = 0$

variabili di
surplus

1. Scegliere un elemento *pivot* positivo su una colonna fuori base.

0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

$r = 1 \quad c = 2$

$Pivot(A, b, c)$

0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

$$r = 1 \quad c = 2$$

2. Dividere la riga r per il pivot.

0	-1	-2	0	0	0
4	-1/2	1	1/2	0	0
10	1	1	0	1	0
7	1	0	0	0	1

$Pivot(A, b, c)$

0	-1	-2	0	0	0
4	-1/2	1	1/2	0	0
10	1	1	0	1	0
7	1	0	0	0	1

3. Sottrarre ad ogni riga $i \neq r$ la riga r moltiplicata per a_{ic} .

8	-2	0	1	0	0
4	-1/2	1	1/2	0	0
6	3/2	0	-1/2	1	0
7	1	0	0	0	1

$$x = \begin{bmatrix} 0 \\ 4 \\ 0 \\ 6 \\ 7 \end{bmatrix} \quad z = -8$$

Entra in base la colonna del pivot ($c = 2$).

Esce di base la colonna corrispondente alla riga del pivot ($r = 1$) nella matrice identità.

Interpretazione algebrica

Dal punto di vista algebrico, l'iterazione corrisponde a riformulare in modo equivalente il sistema di m equazioni in n variabili.

$$\begin{array}{c|ccccc} 0 & -1 & -2 & 0 & 0 & 0 \\ 8 & -1 & 2 & 1 & 0 & 0 \\ 10 & 1 & 1 & 0 & 1 & 0 \\ 7 & 1 & 0 & 0 & 0 & 1 \end{array} \quad \left\{ \begin{array}{lcl} z & = & -x_1 - 2x_2 \\ x_3 & = & 8 + x_1 - 2x_2 \\ x_4 & = & 10 - x_1 - x_2 \\ x_5 & = & 7 - x_1 \end{array} \right.$$

Facendo pivot sull'elemento in riga 1 e colonna 2, entra in base x_2 ed esce di base x_3 .

$$x_3 = 8 + x_1 - 2x_2 \Rightarrow x_2 = 4 + \frac{1}{2}x_1 - \frac{1}{2}x_3$$

da cui, per sostituzione, si ottiene

$$\begin{array}{c|ccccc} 8 & -2 & 0 & 1 & 0 & 0 \\ 4 & -1/2 & 1 & 1/2 & 0 & 0 \\ 6 & 3/2 & 0 & -1/2 & 1 & 0 \\ 7 & 1 & 0 & 0 & 0 & 1 \end{array} \quad \left\{ \begin{array}{lcl} z & = & -8 - 2x_1 + x_3 \\ x_2 & = & 4 + \frac{1}{2}x_1 + \frac{1}{2}x_3 \\ x_4 & = & 10 - x_1 - x_2 \\ x_5 & = & 7 - x_1 \end{array} \right.$$

Interpretazione geometrica

$$A = (0, 0)$$

$$B = (0, 4)$$

L'iterazione corrisponde allo spostamento da A a B.

Soluzione A: $x = [0 \ 0 \ 8 \ 10 \ 7]$

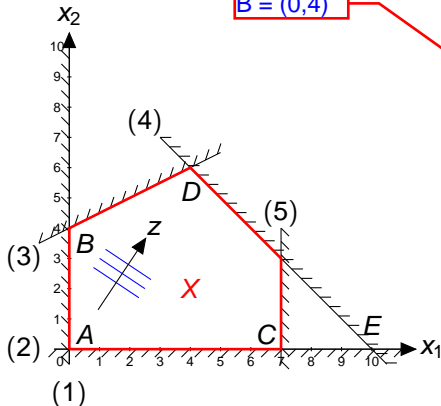
$B = \{3, 4, 5\}$, $\mathcal{N} = \{1, 2\}$, $z = 0$.

Vincoli attivi: $x_1 \geq 0$, $x_2 \geq 0$.

Soluzione B: $x = [0 \ 4 \ 0 \ 6 \ 7]$

$B = \{2, 4, 5\}$, $\mathcal{N} = \{1, 3\}$, $z = -8$.

Vincoli attivi: $x_1 \geq 0$, $x_3 \geq 0$.



Per determinare l'elemento pivot sono necessarie una **regola di scelta della colonna** e una **regola di scelta della riga**.

Test di ottimalità

Il test di ottimalità, indicato con *Optimal(c)* nello pseudocodice, riguarda i **coefficienti di costo ridotto**.

$$z = z_B + \bar{c}_N^T x_N.$$

I coefficienti \bar{c}_N^T indicano di quanto aumenterebbe la funzione obiettivo da minimizzare se le variabili fuori base, x_N , aumentassero di valore anziché valere 0, cioè entrassero in base.

Quando tutti i **coefficienti di costo ridotto** \bar{c}_N^T sono non-negativi, non esistono direzioni ammissibili miglioranti e questo garantisce **l'ottimalità della soluzione corrente**, se è ammissibile. In tal caso $z^* = z_B$.

N.B.: Possono esistere soluzioni degeneri nelle quali la condizione di ottimalità risulta verificata o no, a seconda della base scelta.

Regole di scelta della colonna

Perciò per ogni iterazione dell'algoritmo del simplesso si sceglie sempre una colonna (variabile entrante in base) che abbia **costo ridotto negativo**.

negative reduced cost

Nell'esempio precedente, partendo dal punto A esistono due colonne con costo ridotto negativo e quindi due possibili modi di fare pivot, entrambi corretti:

- facendo entrare in base x_1 la soluzione si sposta nel punto C;
- facendo entrare in base x_2 la soluzione si sposta nel punto B.

In entrambi i casi z migliora.

Nel punto B invece esiste un solo modo corretto di fare pivot (colonna x_1).

Regole di scelta della colonna

Ferma restando la regola suddetta, possono essere utilizzate diverse strategie. Ad es.:

- la colonna col minimo coefficiente di costo ridotto;
- la colonna che produce il maggior miglioramento di z ;
- la prima colonna con costo ridotto negativo, secondo un ordinamento fissato (regola di Bland);
- una colonna scelta a caso tra quelle con costo ridotto negativo.

La regola di scelta della colonna deve garantire che non si possano provocare cicli infiniti nel caso di soluzioni degeneri.

La regola di scelta della colonna garantisce che l'algoritmo del simplesso raggiunga l'ottimalità (quando una soluzione ottima esiste).

Regole di scelta della riga

La scelta della riga (variabile uscente dalla base) deve garantire l'ammissibilità. A questo scopo, una volta determinata la variabile entrante x_j , cioè lo spigolo del poliedro lungo il quale la soluzione cambia, l'unica possibilità corretta è

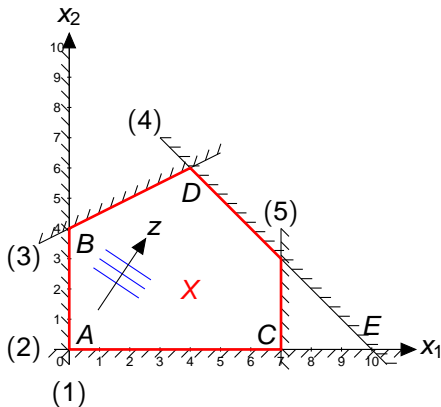
- muoversi verso l'interno del poliedro
- fermarsi appena si incontra una soluzione di base.

Ciò si traduce in

- considerare solo candidati pivot a_{ij} positivi;
- tra le righe i ad essi corrispondenti, scegliere quella che rende minimo il rapporto tra il termine noto b_i ed il candidato pivot a_{ij} .

In caso di ex-aequo, la regola di Bland impone di scegliere la riga di indice minimo, secondo una numerazione prefissata arbitrariamente.

Regole di scelta della riga: esempio



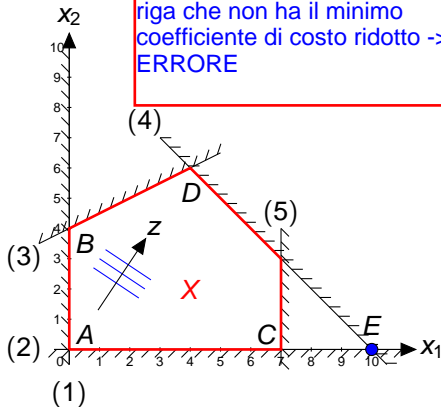
0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

-8	0	-4	-1	0	0
-8	1	-2	-1	0	0
18	0	3	1	1	0
15	0	2	1	0	1

Scegliendo un pivot negativo si ottiene una soluzione non ammissibile (direzione sbagliata).

Regole di scelta della riga: esempio

qui ho scelto come pivot la
riga che non ha il minimo
coefficiente di costo ridotto ->
ERRORE

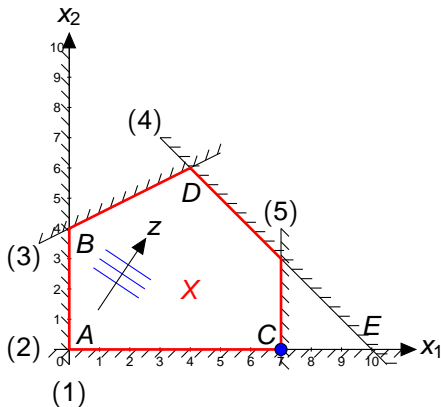


0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

10	0	-1	0	1	0
18	0	3	1	1	0
10	1	1	0	1	0
-3	0	-1	0	-1	1

La direzione è giusta ma il passo è troppo lungo: soluzione inammissibile.

Regole di scelta della riga: esempio



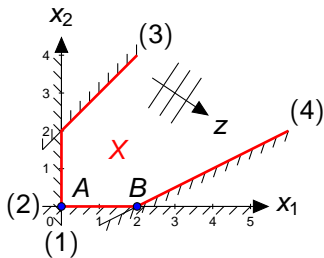
0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

7	0	-2	0	0	1
15	0	2	1	0	1
3	0	1	0	1	-1
7	1	0	0	0	1

Fermandosi sulla prima soluzione di base che si incontra, si ottiene una soluzione ammissibile.

Test di illimitatezza

Se non esistono candidati pivot positivi su una colonna con costo ridotto negativo il problema è **illimitato**.



$$\begin{aligned}
 \text{minimize } z &= -3x_1 + 2x_2 \\
 \text{s.t. } & x_1 - x_2 \geq -2 \\
 & -x_1 + 2x_2 \geq -2 \\
 & x \geq 0
 \end{aligned}$$

$$\begin{array}{c|cccc}
 0 & -3 & 2 & 0 & 0 \\
 \hline
 2 & -1 & 1 & 1 & 0 \\
 2 & 1 & -2 & 0 & 1
 \end{array}$$

A: $\mathcal{B} = \{3, 4\}$ $x = (0, 0, 2, 2)$

$$\begin{array}{c|cccc}
 6 & 0 & -4 & 0 & 3 \\
 \hline
 4 & 0 & -1 & 1 & 1 \\
 2 & 1 & -2 & 0 & 1
 \end{array}$$

B: $\mathcal{B} = \{1, 3\}$ $x = (2, 0, 4, 0)$

Variabili limitate

Può capitare che alcune variabili siano limitate non solo inferiormente ma anche superiormente:

$$0 \leq l \leq x \leq u.$$

Naturalmente i vincoli $x \geq l$ e $x \leq u$ possono essere trattati come vincoli qualsiasi, ma ciò aumenta inutilmente le dimensioni del modello.

In alternativa è possibile estendere la definizione di soluzione di base, considerando due diversi modi in cui una variabile può essere fuori base: una **soluzione di base estesa** è una soluzione nella quale n variabili hanno un valore pari al loro limite **inferiore o superiore** e le altre m formano un sistema lineare indipendente, cioè una base.

Con opportuni accorgimenti, l'algoritmo del simplesso può lavorare con soluzioni di base estese.

Inizializzazione

L'algoritmo del simplesso **mantiene l'ammissibilità** e **cerca l'ottimalità**, quando è inizializzato con una soluzione di base ammissibile.

Tuttavia può accadere che la soluzione di base iniziale sia inammissibile.

In questo caso, si può procedere in diversi modi per inizializzare l'algoritmo.

Metodo delle variabili artificiali

Data un modello di PL in forma standard (anche non canonica) con n variabili e m vincoli,

$$z = \min\{c^T x : Ax = b, x \in \mathbb{R}_+^n\}, \quad (1)$$

si introduce una variabile artificiale $u_i \geq 0$ con coefficiente unitario in ogni vincolo $i = 1, \dots, m$, definendo così un problema artificiale

$$z^a = \min\{e^T u : Ax + Iu = b, x, u \in \mathbb{R}_+^n\}, \quad (2)$$

dove $e^T = [1, \dots, 1]$.

Vale la prima condizione per la forma canonica (i coefficienti di u formano una matrice identità), ma non la seconda (le variabili u non hanno coefficienti nulli nell'obiettivo).

Per ottenere una forma canonica, si effettua la sostituzione $u_i = b_i - \sum_{j=1}^n a_{ij}x_j$ nell'obiettivo, ottenendo la riga 0 del tableau in forma canonica.

Metodo delle variabili artificiali

La soluzione iniziale $x = 0$, $u = b$ è ammissibile per costruzione e quindi si può eseguire l'algoritmo del simplesso sul problema artificiale (2).

Se il valore ottimo del problema artificiale è nullo, cioè si ottiene $z^a = 0$, allora si è trovata una soluzione ammissibile per il problema originario (1).

Altrimenti si è dimostrata l'inammissibilità del problema originario (1).

Vantaggio: si può sempre applicare.

Svantaggio: può comportare nella prima fase molti passi di pivot; lavora su un tableau più grande di quello originario.

Metodo delle variabili artificiali

Per evitare di dover eseguire m passi di pivot (come minimo) per espellere dalla base le m variabili artificiali, si possono usare alcune delle variabili di slack o surplus del problema originario, scrivendo la forma alle disuguaglianze in modo che $b \geq 0$.

$$\begin{aligned}\min z &= \sum_{j=1}^n c_j x_j \\ \text{s.t. } \sum_{j=1}^n a_{ij} x_j &\leq b_i \quad \forall i \in I_1 \\ \sum_{j=1}^n a_{ij} x_j &\geq b_i \quad \forall i \in I_2 \\ \sum_{j=1}^n a_{ij} x_j &= b_i \quad \forall i \in I_3 \\ x &\geq 0\end{aligned}$$

$$\begin{aligned}\min z &= \sum_{j=1}^n c_j x_j \\ \text{s.t. } \sum_{j=1}^n a_{ij} x_j + \hat{x}_i &= b_i \quad \forall i \in I_1 \\ \sum_{j=1}^n a_{ij} x_j - \hat{x}_i &= b_i \quad \forall i \in I_2 \\ \sum_{j=1}^n a_{ij} x_j &= b_i \quad \forall i \in I_3 \\ x, \hat{x} &\geq 0\end{aligned}$$

Metodo delle variabili artificiali

Le variabili \hat{x}_i per i vincoli $i \in I_1$ soddisfano già la prima condizione per la forma canonica.

Sia $h \in I_2$ l'indice del vincolo col massimo valore del termine noto:

$$h = \operatorname{argmax}_{i \in I_2} \{b_i\}.$$

Sottraendo ogni riga $i \in I_2 \setminus \{h\}$ dalla riga h , si ottengono vincoli equivalenti con termine noto non-negativo e variabile \hat{x}_i con coefficiente unitario.

Bisogna quindi inserire variabili artificiali solo per il vincolo h e per i vincoli in I_3 .

Metodo “big M ”

Anziché eliminare dalla formulazione del problema artificiale le variabili x , è possibile mantenerle e penalizzare nell'obiettivo le variabili u con coefficienti molto grandi.

Si risolve quindi il problema

$$z^a = \min\{c^T x + w^T u : Ax + Iu = b, x, u \in \mathbb{R}_+^n\},$$

dove $w^T = [M, \dots, M]$ è un vettore di coefficienti “abbastanza grandi”, cioè tali che ogni soluzione con una variabile $u_i > 0$ abbia costo maggiore del valore ottimo z^* del problema (1).

Vantaggio: non serve una fase di inizializzazione.

Svantaggi:

- non è detto che sia facile determinare il valore appropriato per M (potrebbe esistere una base con $u_i = \epsilon$ per qualche $i = 1, \dots, m$, con ϵ molto piccolo);
- il valore di M potrebbe provocare instabilità numerica.

Metodo di Balinski-Gomory

L'algoritmo trascura temporaneamente la funzione obiettivo e minimizza una misura dell'inammissibilità rispetto ad un vincolo violato, ripetendo l'operazione per tutti i vincoli violati, finché non raggiunge una base ammissibile oppure dimostra che il problema è inammissibile.

Per ogni vincolo violato una misura della violazione è data dal valore assoluto della corrispondente variabile (che ha valore negativo).

Esempio

$$\text{minimize } z = 2x_1 + x_2$$

$$\text{s.t. } x_1 - x_2 \leq 2$$

$$x_2 \leq 5$$

$$x_1 + x_2 \geq 4$$

$$x \geq 0$$

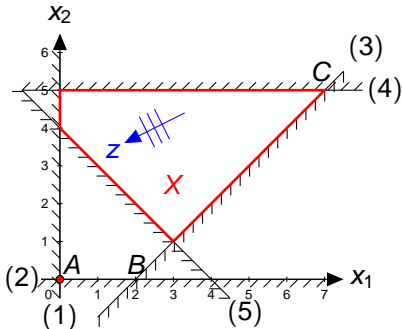


Figura: Base iniziale inammissibile: il vincolo $x_5 \geq 0$ è violato.

0	2	1	0	0	0
2	1	-1	1	0	0
5	0	1	0	1	0
-4	-1	-1	0	0	1

Solution A (infeasible)

$$B = \{3, 4, 5\}$$

$$x^T = [0 \ 0 \ 2 \ 5 \ -4]$$

$$z = 0$$

Il problema ausiliario

0	2	1	0	0	0
2	1	-1	1	0	0
5	0	1	0	1	0
-4	-1	-1	0	0	1

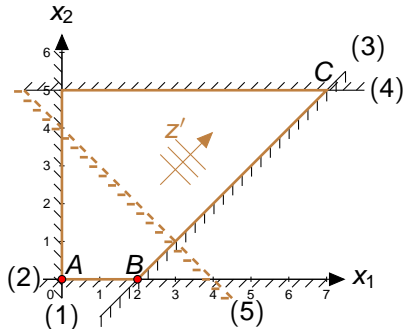
Il problema originale

-4	-1	-1	0	0	1
2	1	-1	1	0	0
5	0	1	0	1	0
0	2	1	0	0	0

Il problema ausiliario

N.B.: La forma canonica si conserva rispetto ai vincoli originali.

Il problema ausiliario



-4	-1	-1	0	0	1
2	1	-1	1	0	0
5	0	1	0	1	0
0	2	1	0	0	0

Soluzione A (inammissibile)

$$B = \{3, 4, 5\}$$

$$x^T = [0 \ 0 \ 2 \ 5 \ -4]$$

$$z = 0$$

-2	0	-2	1	0	1
2	1	-1	1	0	0
5	0	1	0	1	0
-4	0	3	-2	0	0

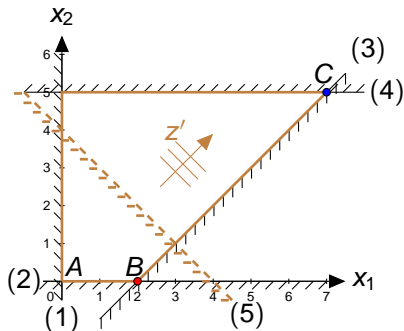
Soluzione B (inammissibile)

$$B = \{1, 4, 5\}$$

$$x^T = [2 \ 0 \ 0 \ 5 \ -2]$$

$$z = 4$$

Il problema ausiliario



-2	0	-2	1	0	1
2	1	-1	1	0	0
5	0	1	0	1	0
-4	0	3	-2	0	0

Soluzione B (inammissibile)

$$B = \{1, 4, 5\}$$

$$x^T = [2 \ 0 \ 0 \ 5 \ -2]$$

$$z = 4$$

8	0	0	1	2	1
7	1	0	1	1	0
5	0	1	0	1	0
-19	0	0	-2	-3	0

Soluzione C (ammissibile)

$$B = \{1, 2, 5\}$$

$$x^T = [7 \ 5 \ 0 \ 0 \ 8]$$

$$z = 19$$

Il problema originale

8	0	0	1	2	1
7	1	0	1	1	0
5	0	1	0	1	0
-19	0	0	-2	-3	0

Il problema ausiliario

-19	0	0	-2	-3	0
7	1	0	1	1	0
5	0	1	0	1	0
8	0	0	1	2	1

Il problema originale

Il valore della funzione obiettivo (trascurata finora) può essere peggiorato.

Test di inammissibilità

Se l'inammissibilità rispetto ad un vincolo violato è stata minimizzata ma il valore della corrispondente variabile resta negativo, questo dimostra che il problema è inammissibile e l'algoritmo termina.

Esempio: sostituendo $x_2 \leq 5$ con $x_2 \leq 1/2$ nell'ultimo esempio. Dopo due passi di pivot si ottiene il tableau seguente:

-1		0	0	1	2	1
5/2		1	0	1	1	0
1/2		0	1	0	1	0
-11/2		0	0	-2	-3	0

Il proble ausiliario è risolto all'ottimo, (tutti i costi ridotti sono non-negativi).

L'inammissibilità, quindi, non può essere ridotta ulteriormente: il problema è inammissibile.

$$0x_1 + 0x_2 + 1x_3 + 2x_4 + 1x_5 = -1$$

non ha soluzione per $x \geq 0$.

Problema ausiliario illimitato

Può capitare che il problema ausiliario sia illimitato (anche se il problema originale non lo è).

Esempio: cancellando il vincolo $x_2 \leq 5$ dall'esempio precedente.

0	2	1	0	0
2	1	-1	1	0
-4	-1	-1	0	1

Il problema originale

-4	-1	-1	0	1
2	1	-1	1	0
0	2	1	0	0

Il problema ausiliario

La seconda colonna è interamente composta da elementi negativi: problema illimitato.

Problema ausiliario illimitato

In tal caso, il pivot da scegliere è l'elemento (negativo!) sulla riga del vincolo violato.

-4	-1	-1	0	1
2	1	-1	1	0
0	2	1	0	0

4	1	1	0	-1
6	2	0	1	-1
-4	1	0	0	1

La soluzione di base risultante è un punto sul vincolo violato.

Teoria della dualità

Ricerca operativa

Giovanni Righini



UNIVERSITÀ DEGLI STUDI
DI MILANO

Teoria della dualità

La teoria della dualità per la programmazione lineare fu sviluppata da A. Tucker nel 1948, seguendo un'intuizione di J. von Neumann.

Fornisce un punto di vista diverso e molto utile per comprendere e per risolvere i problemi lineari.

E' anche il fondamento su cui si possono progettare algoritmi di ottimizzazione e di approssimazione e dimostrare le loro proprietà.

Si applica anche a problemi non-lineari e a problemi nel discreto, ma nel caso della programmazione lineare si ottengono risultati più forti.

Il problema duale

Ogni problema di PL, che d'ora in poi indichiamo come **problema primale**, ammette un altro problema di PL, che denominiamo **problema duale**.

La corrispondenza tra i due può essere stabilita direttamente dalla forma generale, secondo questo schema:

Problema primale	Funzione obiettivo	Problema duale
Minimizzazione		Massimizzazione
m vincoli		m variabili
n variabili		n vincoli
coefficienti della f.o.		termini noti dei vincoli
termini noti dei vincoli		coefficienti della f.o.
matrice dei coefficienti A		matrice dei coefficienti A^T
vincoli di uguaglianza		variabili libere
variabili libere		vincoli di uguaglianza
vincoli di disuguaglianza \geq		variabili non-negative
variabili non-negative		vincoli di disuguaglianza \leq

Coppie primale-duale

Problema primale P :

$$\begin{aligned} \text{minimize } z &= c_1^T x_1 + c_2^T x_2 \\ \text{s.t. } A_{11}x_1 + A_{12}x_2 &\geq b_1 \\ A_{21}x_1 + A_{22}x_2 &= b_2 \\ x_1 &\geq 0 \\ x_2 &\text{ libere} \end{aligned}$$

Problema duale D :

$$\begin{aligned} \text{maximize } w &= b_1^T y_1 + b_2^T y_2 \\ \text{s.t. } A_{11}^T y_1 + A_{21}^T y_2 &\leq c_1 \\ A_{12}^T y_1 + A_{22}^T y_2 &= c_2 \\ y_1 &\geq 0 \\ y_2 &\text{ libere} \end{aligned}$$

La relazione è simmetrica: il duale del duale di P è P .

Teorema della dualità in forma debole

Data una coppia primale-duale

$$P : \text{maximize } z(x), \text{ s.t. } x \in X$$

$$D : \text{minimize } w(y), \text{ s.t. } y \in Y$$

per ogni soluzione ammissibile $\bar{x} \in X$ di P e per ogni soluzione ammissibile $\bar{y} \in Y$ di D , si ha

$$z(\bar{x}) \leq w(\bar{y}).$$

Valore funzione obiettivo
che si massimizza



Valore funzione obiettivo
che si minimizza

Dimostrazione (alle disuguaglianze)

Data una coppia primale-duale

$$P : \text{maximize } z = c^T x, \text{ s.t. } Ax \leq b, x \geq 0$$

$$D : \text{minimize } w = b^T y, \text{ s.t. } A^T y \geq c, y \geq 0$$

$$\forall \bar{x} \in X, \bar{y} \in Y, c^T \bar{x} \leq b^T \bar{y}.$$

Dimostrazione.

$$A^T \bar{y} \geq c, \bar{x} \geq 0 \Rightarrow \bar{x}^T A^T \bar{y} \geq \bar{x}^T c \Leftrightarrow c^T \bar{x} \leq \bar{x}^T A^T \bar{y}.$$

Analogamente

$$A\bar{x} \leq b, \bar{y} \geq 0 \Rightarrow \bar{y}^T A\bar{x} \leq \bar{y}^T b \Leftrightarrow b^T \bar{y} \geq \bar{y}^T A\bar{x}.$$

In ambo i casi il secondo membro è uno scalare:

$$\bar{x}^T A^T \bar{y} = (\bar{x}^T A^T \bar{y})^T = \bar{y}^T (A^T)^T (\bar{x}^T)^T = \bar{y}^T A\bar{x}.$$

Quindi

$$c^T \bar{x} \leq \bar{x}^T A^T \bar{y} = \bar{y}^T A\bar{x} \leq b^T \bar{y}.$$

Corollari

Se il valore delle due f.o. allora
quel valore è la soluzione ottima

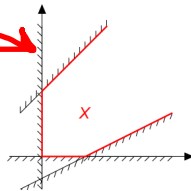
Corollario 1.

Se $\bar{x} \in X$, $\bar{y} \in Y$ e $z(\bar{x}) = w(\bar{y})$,
allora $\bar{x} = x^*$, $\bar{y} = y^*$, $z(\bar{x}) = z^*$ e $w(\bar{y}) = w^*$.

Corollario 2.

Se un problema di PL P è illimitato, allora il suo duale
inammissibile.

N.B. Non vale il viceversa.



Teorema fondamentale dell'algebra

Dato un **sistema di equazioni** lineari $Ax = b$, con A di dimensione $m \times n$ e b di dimensione m , una e una sola di queste alternative è vera:

$$\exists x \in \mathbb{R}^n : Ax = b$$

$$\exists y \in \mathbb{R}^m : y^T A = 0, y^T b \neq 0$$

In altri termini, o esiste un **certificato di ammissibilità** x , la cui esistenza dimostra che il sistema ha soluzione, o esiste un **certificato di inammissibilità** y , la cui esistenza dimostra che il problema non ha soluzione.

Un risultato simile (teorema delle alternative) si può dimostrare per i **sistemi di disequazioni**.

Lemma di Farkas

Dato un sistema di equazioni lineari $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$, con A di dimensione $m \times n$ e \mathbf{b} di dimensione m , una e una sola di queste alternative è vera:

- (i) $\exists \mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0$
- (ii) $\exists \mathbf{y} \in \mathbb{R}^m : A^T \mathbf{y} \geq 0, \mathbf{b}^T \mathbf{y} < 0$.

Dimostrazione - parte I: se (i) è vera, (ii) è falsa.

Assumiamo che (i) sia vera. Allora

$$\exists \bar{\mathbf{x}} \in \mathbb{R}^n : A\bar{\mathbf{x}} = \mathbf{b}, \bar{\mathbf{x}} \geq 0.$$

Allora

$$A^T \mathbf{y} \geq 0 \Rightarrow \bar{\mathbf{x}}^T A^T \mathbf{y} \geq 0.$$

Quindi

$$\bar{\mathbf{x}}^T A^T = \mathbf{b}^T \Rightarrow \mathbf{b}^T \mathbf{y} \geq 0.$$

Quindi (ii) è falsa.

Lemma di Farkas

Dimostrazione - parte II: se (i) è falsa, (ii) è vera.

Assumiamo che (i) sia falsa e definiamo il cono

$$C = \{q \in \mathbb{R}^m : \exists \mathbf{x}(q) \in \mathbb{R}^n : \mathbf{x}(q) \geq 0, A\mathbf{x}(q) = q\}.$$

C è convesso. Se (i) è falsa, $b \notin C$.

Quindi, per il teorema dell'iperpiano separatore,

$$\exists \mathbf{y} \in \mathbb{R}^m \setminus \{0\} : \mathbf{q}^T \mathbf{y} \geq 0 \quad \forall q \in C, \mathbf{b}^T \mathbf{y} < 0.$$

Poiché $q = A\mathbf{x}(q) \quad \forall q \in C$,

$$\mathbf{q}^T \mathbf{y} \geq 0 \quad \forall q \in C \Rightarrow \mathbf{x}(q)^T A^T \mathbf{y} \geq 0 \quad \forall q \in C.$$

Poiché $\mathbf{x}(q) \geq 0 \quad \forall q \in C$,

$$\mathbf{x}(q)^T A^T \mathbf{y} \geq 0 \quad \forall q \in C \Rightarrow A^T \mathbf{y} \geq 0.$$

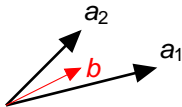
Quindi (ii) è vera.

Interpretazione geometrica

Sia a_j la generica colonna della matrice A .

$$Ax = b, x \geq 0$$

$$b = \sum_{j=1}^n a_j x_j, x_j \geq 0 \forall j = 1, \dots, n.$$



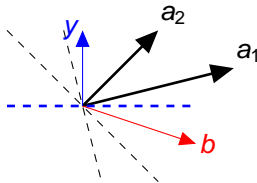
(i) è vera, (ii) è falsa: $b \in C$.

$$A^T y \geq 0, b^T y < 0$$

$$a_j^T y \geq 0 \forall j = 1, \dots, n.$$

$$b^T y < 0$$

L'iperpiano $z^T y = 0$ separa b dal cono C .



(i) è falsa, (ii) è vera: $b \notin C$.

Lemma di Farkas: variante

Dato un **sistema di disequazioni** lineari $Ax \leq b, x \geq 0$, con A di dimensione $m \times n$ e b di dimensione m , una e una sola di queste alternative è vera:

$$(i) \exists x \in \mathbb{R}^n : Ax \leq b, x \geq 0$$

$$(ii) \exists y \in \mathbb{R}^m : A^T y \geq 0, b^T y < 0, y \geq 0.$$

Dimostrazione. Introduciamo variabili di slack non-negative. La condizione (i) diventa:

$$(i) \exists x \in \mathbb{R}^n, s \in \mathbb{R}^m : Ax + Is = b, x \geq 0, s \geq 0.$$

Definendo $A' = [A \mid I]$ di dimensioni $(m \times (n + m))$ e $x' = \begin{bmatrix} x \\ s \end{bmatrix}$ di dimensione $(n + m) \times 1$, la condizione (i) diventa

$$(i) \exists x' \in \mathbb{R}^{n+m} : A'x' = b, x' \geq 0.$$

Per il Lemma di Farkas, essa è alternativa alla condizione

$$(ii) \exists y \in \mathbb{R}^m : A'^T y \geq 0, b^T y < 0.$$

Il sistema di disequazioni $A'^T y \geq 0$ equivale a $A^T y \geq 0, y \geq 0$. (c.v.d.)

Teorema della dualità in forma forte

Teorema della dualità forte.

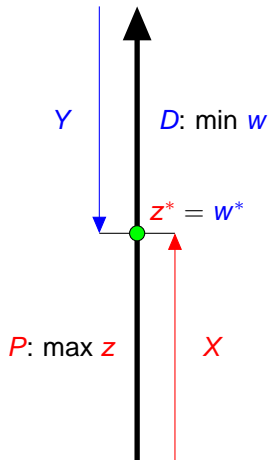
Data una coppia primale-duale

$$P : \text{maximize } z = c^T x, \text{ s.t. } Ax \leq b, x \geq 0$$

$$D : \text{minimize } w = b^T y, \text{ s.t. } A^T y \geq c, y \geq 0,$$

se uno dei due problemi ammette soluzione ottima finita, allora anche l'altro ammette soluzione ottima finita e i due valori ottimi coincidono.

Teorema della dualità in forma forte



Teorema della dualità in forma forte

Dimostrazione. Sia $y^* \in \mathbb{R}^m$ la soluzione ottima finita del duale D e sia $w^* = b^T y^*$ il suo valore.

Vogliamo dimostrare che $\exists x^* \in \mathbb{R}^n : Ax^* \leq b, x^* \geq 0$ e che il suo valore soddisfa la disequazione $c^T x^* \geq b^T y^*$.

Procedendo per assurdo, supponiamo che tale soluzione x^* del primale P non esista e utilizzando il lemma di Farkas, dimostriamo che in tal caso sarebbe violata l'ipotesi di ottimalità di y^* .

Teorema della dualità in forma forte

Assumiamo che

$$\nexists \mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0, \mathbf{c}^T \mathbf{x} \geq w^*.$$

Ciò è equivalente ad affermare che

$$\nexists \mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}, -\mathbf{c}^T \mathbf{x} \leq -w^*, \mathbf{x} \geq 0.$$

Definendo

$$A' = \begin{bmatrix} A \\ -\mathbf{c}^T \end{bmatrix} \quad b' = \begin{bmatrix} \mathbf{b} \\ -w^* \end{bmatrix}$$

rispettivamente di dimensione $(m+1) \times n$ e $(m+1) \times 1$, l'ipotesi equivale a

$$\nexists \mathbf{x} \in \mathbb{R}^n : A'\mathbf{x} \leq \mathbf{b}', \mathbf{x} \geq 0.$$

Per il Lemma di Farkas, ciò implica che

$$\exists \mathbf{y}' \in \mathbb{R}^{m+1} : A'^T \mathbf{y}' \geq 0, \mathbf{b}'^T \mathbf{y}' < 0, \mathbf{y}' \geq 0.$$

Teorema della dualità in forma forte

$$\exists \mathbf{y}' \in \Re^{m+1} : \mathbf{A}'^T \mathbf{y}' \geq 0, \mathbf{b}'^T \mathbf{y}' < 0, \mathbf{y}' \geq 0.$$

Sia

$$\mathbf{y}' = \begin{bmatrix} \mathbf{y} \\ \lambda \end{bmatrix},$$

con $\mathbf{y} \in \Re^m$ e $\lambda \in \Re$. Allora, il sistema

$$\mathbf{A}'^T \mathbf{y}' \geq 0, \mathbf{b}'^T \mathbf{y}' < 0, \mathbf{y}' \geq 0$$

equivale a

$$\mathbf{A}^T \mathbf{y} - c\lambda \geq 0, \mathbf{b}^T \mathbf{y} - \mathbf{w}^* \lambda < 0, \mathbf{y} \geq 0, \lambda \geq 0.$$

Studiamo separatamente i due casi $\lambda > 0$ e $\lambda = 0$.

Teorema della dualità in forma forte

$$A^T y - c\lambda \geq 0, b^T y - w^*\lambda < 0, y \geq 0, \lambda \geq 0.$$

Caso I: $\lambda > 0$. Dividendo tutte le disequazioni per λ e ponendo

$$\hat{y} = \frac{y}{\lambda},$$

si ha

$$\begin{cases} A^T \hat{y} \geq c \\ b^T \hat{y} < w^* \\ \hat{y} \geq 0 \end{cases}$$

Ciò implica che esista una soluzione ammissibile per il duale, il cui valore è minore del valore minimo, il che genera contraddizione.

Teorema della dualità in forma forte

Caso II: $\lambda = 0$. In tal caso il sistema

$$A^T y - c\lambda \geq 0, b^T y - w^*\lambda < 0, y \geq 0, \lambda \geq 0.$$

diventa

$$A^T y \geq 0, b^T y < 0, y \geq 0.$$

Ponendo $\hat{y} = y^* + y$, e osservando che y^* soddisfa le condizioni

$$A^T y^* \geq c, b^T y^* = w^*, y^* \geq 0,$$

si ottiene

$$A^T \hat{y} \geq c, b^T \hat{y} < w^*, \hat{y} \geq 0.$$

Anche in questo caso si ha una contraddizione, poiché \hat{y} è una soluzione ammissibile per il duale ed il suo valore è minore del valore minimo.

Teorema della dualità in forma forte

Questa dimostrazione per assurdo consente di affermare che

$$\exists \mathbf{x}^* \in \mathbb{R}^n : A\mathbf{x}^* \leq \mathbf{b}, \mathbf{x}^* \geq 0, \mathbf{c}^T \mathbf{x}^* \geq \mathbf{w}^*.$$

Per il teorema della dualità in forma debole, non è possibile che $\mathbf{c}^T \mathbf{x}^* > \mathbf{w}^*$. Quindi

$$\mathbf{z}^* = \mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^* = \mathbf{w}^*.$$

Teorema fondamentale della dualità lineare

Teorema fondamentale della dualità lineare.

Data una coppia primale-duale

$$P : \text{maximize } z(x), \text{ s.t. } x \in X$$

$$D : \text{minimize } w(y), \text{ s.t. } y \in Y,$$

esiste una sequenza finita di passi di pivot che porta l'algoritmo del simpleso a terminare, riconoscendo uno di questi quattro possibili casi:

- soluzione ottima di P e D ;
- P è illimitato e D è inammissibile;
- D è illimitato e P è inammissibile;
- sia P che D sono inammissibili.

Scarto complementare

Teorema dello scarto complementare.

Data una coppia primale-duale

$$P : \text{maximize } z = c^T x, \text{ s.t. } Ax \leq b, x \geq 0$$

$$D : \text{minimize } w = b^T y, \text{ s.t. } A^T y \geq c, y \geq 0,$$

condizione necessaria e sufficiente per l'ottimalità di due soluzioni ammissibili \bar{x} e \bar{y} è che valgano le equazioni seguenti:

$$\bar{y}^T (b - A\bar{x}) = 0$$

$$(A^T \bar{y} - c)\bar{x} = 0$$

N.B. Le condizioni sono significative per i vincoli di disuguaglianza e le loro corrispondenti variabili di slack/surplus. Per i vincoli di uguaglianza sono già implicate dall'ammissibilità delle soluzioni.

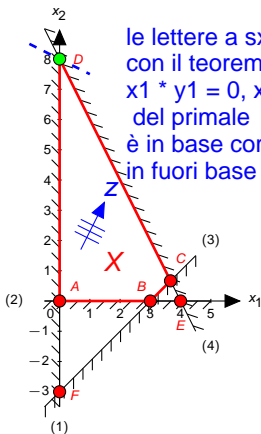
Esempio

P) $\max z = x_1 + 2x_2$

s.t. $x_1 - x_2 \leq 3$ $[y_3]$

$2x_1 + x_2 \leq 8$ $[y_4]$

$x_1, x_2 \geq 0$



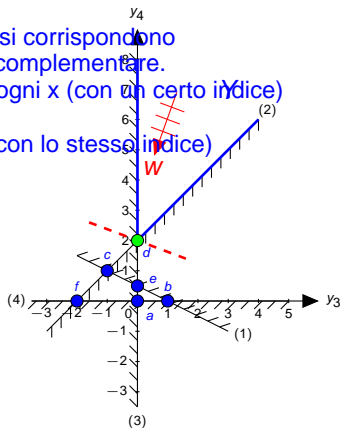
le lettere a sx e quelle a dx si corrispondono con il teorema dello scarto complementare.
 $x_1 * y_1 = 0$, $x_2 * y_2 = 0$; ad ogni x (con un certo indice) del primale è in base corrisponde la y (con lo stesso indice) in fuori base

D) $\min w = 3y_3 + 8y_4$

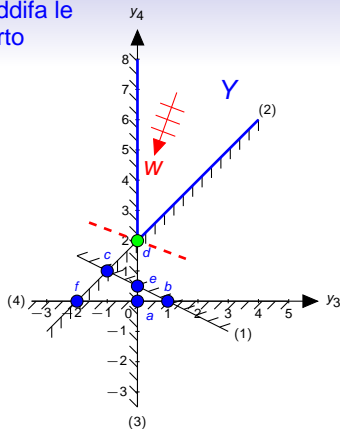
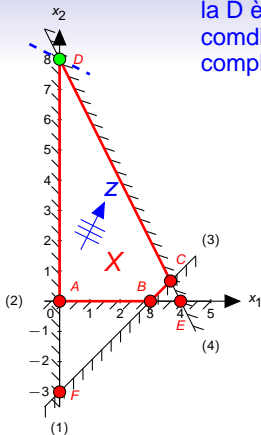
s.t. $y_3 + 2y_4 \geq 1$ $[x_1]$

$-y_3 + y_4 \geq 2$ $[x_2]$

$y_3, y_4 \geq 0$



la D è l'unica che soddisfa le
condizioni dello scarto
complementare



Sol.	x_1	x_2	x_3	x_4	$\in X$	z
A	0	0	3	8	Y	0
B	3	0	0	2	Y	3
C	$\frac{11}{3}$	$\frac{2}{3}$	0	0	Y	5
D	0	8	11	0	Y	16
E	4	0	-1	0	N	4
F	0	-3	0	11	N	-6

Sol.	y_1	y_2	y_3	y_4	$\in Y$	w
a	-1	-2	0	0	N	0
b	0	-3	1	0	N	3
c	0	0	-1	1	N	5
d	3	0	0	2	Y	16
e	0	$-\frac{3}{2}$	0	$\frac{1}{2}$	N	4
f	-3	0	-2	0	N	-6

La soluzione è
ammissibile? L'unica
ammissibile è a D

L'algoritmo del simplesso duale

Dato che i coefficienti di P e D sono gli stessi (ancorché in posizione diversa nei due modelli), entrambi i problemi di una coppia primale-duale si possono rappresentare sullo stesso tableau.

Problema primale

$$\begin{aligned} \max z = & 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 - 2x_2 \leq 2 \quad [y_3] \\ & x_2 \leq 4 \quad [y_4] \\ & x_1 + x_2 \leq 6 \quad [y_5] \\ & x_1, x_2 \geq 0 \end{aligned}$$

0	-2	-1	0	0	0
2	1	-2	1	0	0
4	0	1	0	1	0
6	1	1	0	0	1

Problema duale

$$\begin{aligned} \min w = & 2y_3 + 4y_4 + 6y_5 \\ \text{s.t.} \quad & y_3 + y_5 \geq 2 \quad [x_1] \\ & -2y_3 + y_4 + y_5 \geq 1 \quad [x_2] \\ & y_3, y_4, y_5 \geq 0 \end{aligned}$$

0	0	0	2	4	6
-2	1	0	-1	0	-1
-1	0	1	2	-1	-1

Il tableau ristretto

Problema primale

0	-2	-1	0	0	0
2	1	-2	1	0	0
4	0	1	0	1	0
6	1	1	0	0	1
	x_1	x_2	x_3	x_4	x_5

Tableau

0	-2	-1	
2	1	-2	y_3
4	0	1	y_4
6	1	1	y_5
	x_1	x_2	

Tableau ristretto

Problema duale

0	0	0	2	4	6
-2	1	0	-1	0	-1
-1	0	1	2	-1	-1
	y_1	y_2	y_3	y_4	y_5

Tableau

0	2	4	6	
-2	-1	0	-1	x_1
-1	2	-1	-1	x_2
	y_3	y_4	y_5	

Tableau ristretto

Algoritmo del semplice duale

E' possibile lavorare sul tableau del problema primale, eseguendo su di esso gli stessi passi di pivot che l'algoritmo del semplice eseguirebbe se lavorasse sul tableau del problema duale.

L'algoritmo risultante è l'algoritmo del semplice duale.

Problema primale

0		1	2	0	0	0
2		1	-1	1	0	0
3		0	1	0	1	0
-4		-1	-1	0	0	1

$$\mathcal{B}_P = \{3, 4, 5\}$$

Inammissibile, poiché $x_5 < 0$.
ho un -4, è inammissibile

Problema duale

0		0	0	2	3	-4
1		1	0	-1	0	1
2		0	1	1	-1	1

$$\mathcal{B}_D = \{1, 2\}$$

Sub-ottima, poiché il costo ridotto di y_5 è negativo.

Algoritmo del simplesso duale

Problema primale

0		1	2	0	0	0
2		1	-1	1	0	0
3		0	1	0	1	0
-4		-1	-1	0	0	1

$\mathcal{B}_P = \{3, 4, 5\}$, inammissibile

-4		0	1	0	0	1
-2		0	-2	1	0	1
3		0	1	0	1	0
4		1	1	0	0	-1

$\mathcal{B}_P = \{1, 3, 4\}$, inammissibile

Problema duale

0		0	0	2	3	-4
1		1	0	-1	0	1
2		0	1	1	-1	1

$\mathcal{B}_D = \{1, 2\}$, sub-ottima.

4		4	0	-2	3	0
1		1	0	-1	0	1
1		1	1	2	-1	0

$\mathcal{B}_D = \{2, 5\}$, sub-ottima.

Algoritmo del simplesso duale

Problema primale

-4	0	1	0	0	1
-2	0	-2	1	0	1
3	0	1	0	1	0
4	1	1	0	0	-1

$\mathcal{B}_P = \{1, 3, 4\}$, inammissibile.

-1	0	0	$\frac{1}{2}$	0	$\frac{3}{2}$
1	0	1	$-\frac{1}{2}$	0	$-\frac{1}{2}$
2	0	0	$\frac{1}{2}$	1	$\frac{1}{2}$
3	1	0	$\frac{1}{2}$	0	$-\frac{1}{2}$

$\mathcal{B}_P = \{1, 2, 4\}$

Ammissibilità raggiunta: stop.

Problema duale

4	4	0	-2	3	0
1	1	0	-1	0	1
1	1	1	2	-1	0

$\mathcal{B}_D = \{2, 5\}$, sub-ottima.

1	3	1	0	2	0
$\frac{3}{2}$	$\frac{3}{2}$	$\frac{3}{2}$	0	$\frac{3}{2}$	1
$\frac{1}{2}$	$\frac{3}{2}$	$\frac{3}{2}$	1	$\frac{3}{2}$	0

$\mathcal{B}_D = \{3, 5\}$

Ottimalità raggiunta: stop.

Algoritmo del simplesso duale

Simpleso primale: conserva l'ammissibilità e persegue l'ottimalità.

Simpleso duale: conserva l'ottimalità e persegue l'ammissibilità.

Nell'algoritmo del simplesso duale le regole di scelta del pivot sono duali:

- la riga del pivot viene scelta prima della colonna: il suo termine noto dev'essere negativo (vincolo violato);
- il pivot dev'essere negativo;
- la colonna del pivot viene scelta minimizzando il valore assoluto del rapporto tra il coefficiente di costo ridotto ed il candidato pivot.

Algoritmo del simplesso duale

L'algoritmo del simplesso duale è particolarmente utile quando la base iniziale è inammissibile e super-ottima.

E' una tipica situazione che si verifica negli *algoritmi "cutting planes"*, che vengono usati per risolvere rilassamenti continui di problemi di programmazione lineare intera o binaria.

Analisi post-ottimale

Ricerca operativa

Giovanni Righini



UNIVERSITÀ DEGLI STUDI
DI MILANO

Analisi post-ottimale

Dopo aver calcolato la **soluzione ottima** di un problema, ma prima di prendere una **decisione** conseguente, è molto importante valutare la **robustezza** della soluzione.

Infatti, i **dati** sono spesso affetti da errori, approssimazioni, incertezza, arrotondamenti,...

La domanda cui risponde l'analisi post-ottimale è: quanto è robusta la **soluzione ottima** rispetto a possibili (piccoli) cambiamenti nel valore dei **data** che sono stati usati per calcolarla?

Analisi di sensitività

Input: A, b, c .
cioè A = matrice coefficiente dei vincoli, b = valori noti, c = vettore dei coefficienti della f.o.

Output: B^*, x^*, z^* .

Cioè base ottima, soluzione ottima e valore ottimo della f.o.

Lo scopo dell'**analisi di sensitività** è di valutare l'intervallo nel quale può variare ogni coefficiente c_j e b_i senza che cambi la base ottima B^* .

La base B^* rimane ottima finché valgono le condizioni di **ammissibilità** e di **ottimalità**:

- Ammissibilità: $x_B = B^{-1}b \geq 0$.
- Ottimalità: $\bar{c}_N = c_N - c_B B^{-1}N \geq 0$.

Le condizioni di ammissibilità dipendono solo da b .

Le condizioni di ottimalità dipendono solo da c .

Variazione di un coefficiente c_j

$$\text{maximize } z = 2x_1 + x_2$$

$$\text{s.t. } -x_1 + 2x_2 \leq 12 \quad (3)$$

$$3x_1 - x_2 \leq 24 \quad (4)$$

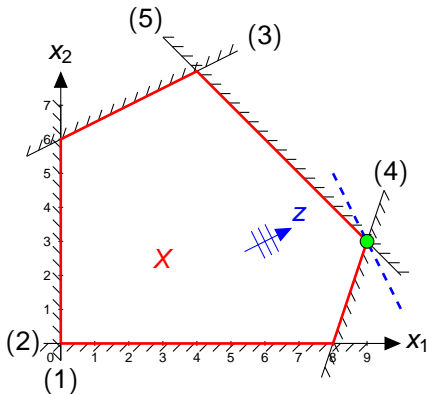
$$x_1 + x_2 \leq 12 \quad (5)$$

$$x \geq 0$$

$$\mathcal{B}^* = \{1, 2, 3\}.$$

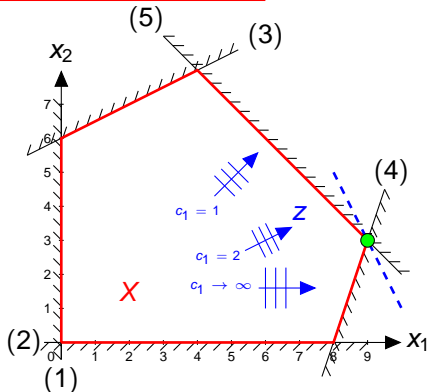
$$x^* = [9 \ 3 \ 15 \ 0 \ 0]^T.$$

$$z^* = 21.$$



Variazione di un coefficiente c_j

Perché se il coefficiente decresce significa che x_1 pesa di meno, quindi conta di meno andare verso dx ma conta di più andare verso l'alto



Quando c_1 decresce, la f.o. ruota in senso antiorario, finché la base ottima cambia per $c_1 = 1$, quando le linee di livello diventano parallele al vincolo (5).

Quando c_1 aumenta, la f.o. ruota in senso orario e le curve di livello tendono a diventare verticali per $c_1 \rightarrow \infty$. La base ottima in questo caso non cambia.

Quindi, $B^* = \{1, 2, 3\}$ è ottima per $1 \leq c_1 < \infty$.

Sebbene B^* non cambi e x^* non cambi, z^* cambia perché dipende da c_1 :

$$z^*(c_1) = x_1^* c_1 + x_2^* = 9c_1 + 3.$$

Variazione di un coefficiente c_j

Tutti i dati (c^* e a^*) necessari per l'analisi di sensitività sono contenuti nel tableau all'ottimo.

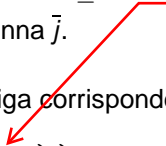
Supponiamo di analizzare un problema che nella forma alle disuguaglianze ha

- funzione obiettivo da massimizzare,
- vincoli di disuguaglianza \leq .

Consideriamo una colonna \bar{j} .

devo fare il rapporto tra i valori delle colonne fuori base su r .

Caso 1: $\bar{j} \in \mathcal{B}$ e \bar{r} è la riga corrispondente.


$$\max \left\{ -\infty, \max_{j \in \mathcal{N}} \left\{ \frac{-c_j^*}{a_{\bar{r}j}^{*+}} \right\} \right\} \leq \Delta c_{\bar{j}} \leq \min \left\{ \min_{j \in \mathcal{N}} \left\{ \frac{-c_j^*}{a_{\bar{r}j}^{*-}} \right\}, +\infty \right\}.$$

(infinito (sia nel caso max che minimo) si verifica quando non ci sono possibilità di rendere la f.o. parallela ad un vincolo.

Caso 2: $\bar{j} \in \mathcal{N}$

$$\Delta c_{\bar{j}} \leq c_{\bar{j}}^*.$$

Variazione di un coefficiente c_j

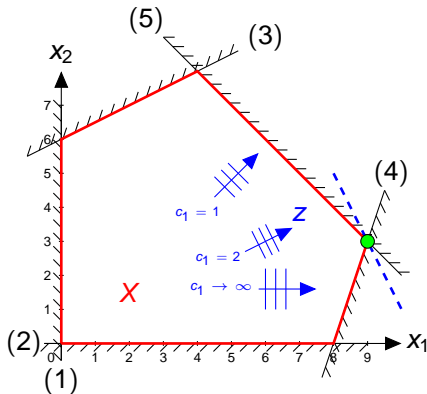


Tableau all'ottimo:

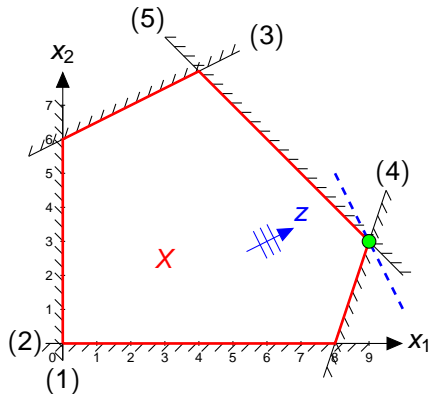
21	0	0	0	1/4	5/4
15	0	0	1	3/4	-5/4
9	1	0	0	1/4	1/4
3	0	1	0	-1/4	3/4

$\bar{j} = 1$, variabile in base, $\bar{r} = 2$.

$$\max\left\{\frac{-1/4}{1/4}, \frac{-5/4}{1/4}\right\} \leq \Delta c_1 < +\infty$$

$$-1 \leq \Delta c_1 < +\infty$$

Variazione di un coefficiente b_i



$$\text{maximize } z = 2x_1 + x_2$$

$$\text{s.t. } -x_1 + 2x_2 \leq 12 \quad (3)$$

$$3x_1 - x_2 \leq 24 \quad (4)$$

$$x_1 + x_2 \leq 12 \quad (5)$$

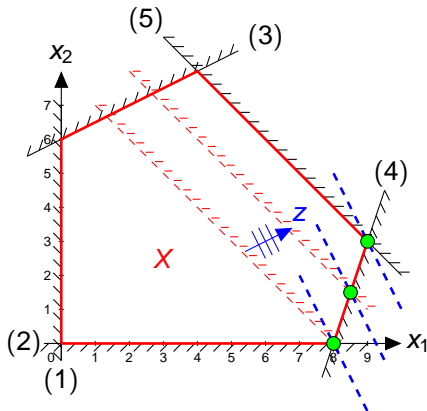
$$x \geq 0$$

$$B^* = \{1, 2, 3\}.$$

$$x^* = [9 \ 3 \ 15 \ 0 \ 0]^T.$$

$$z^* = 21.$$

Variazione di un coefficiente b_i



Quando b_3 decresce, il vincolo (3) trasla verso il basso e a sinistra, finché il vincolo $x_2 \geq 0$ diventa attivo per $b_3 = 8$.

Quando b_3 aumenta, il vincolo (3) trasla verso l'alto e a destra, finché il vincolo (1) diventa attivo per $b_3 = 24$.

Quindi, $B^* = \{1, 2, 3\}$ è ottima per $8 \leq b_3 \leq 24$.

Benché B^* non cambi, x^* e z^* cambiano perché dipendono da

$$b_3: x_1^*(b_3) = 6 + \frac{1}{4}b_3.$$

$$x_2^*(b_3) = -6 + \frac{3}{4}b_3.$$

$$z^*(b_3) = 6 + \frac{5}{4}b_3.$$

Variazione di un coefficiente b_i

Tutti i dati (b^* e a^*) necessari per l'analisi di sensitività sono contenuti nel tableau all'ottimo.

Supponiamo di analizzare un problema che nella forma alle disuguaglianze ha

- funzione obiettivo da massimizzare,
- vincoli di disuguaglianza \leq .

Consideriamo una riga \bar{i} e sia \bar{j} è la colonna della variabile di slack corrispondente.

Caso 1: \bar{i} attivo.

$$\max \left\{ -\infty, \max_i \left\{ \frac{-b_i^*}{a_{i\bar{j}}^*} \right\} \right\} \leq \Delta b_{\bar{i}} \leq \min \left\{ \min_i \left\{ \frac{-b_i^*}{a_{i\bar{j}}^*} \right\}, +\infty \right\}.$$

Caso 2: \bar{i} non attivo.

$$\Delta b_{\bar{i}} \geq -x_{\bar{j}}^*.$$

Variazione di un coefficiente b_i

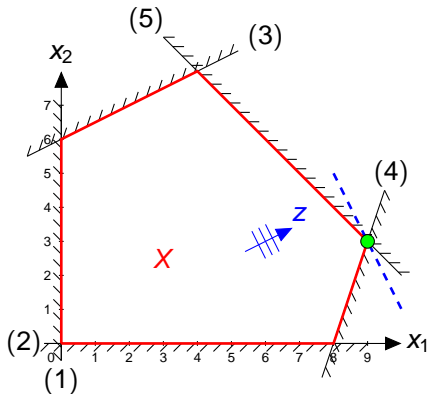


Tableau all'ottimo:

21	0	0	0	1/4	5/4
15	0	0	1	3/4	-5/4
9	1	0	0	1/4	1/4
3	0	1	0	-1/4	3/4

$\bar{i} = 3$, vincolo attivo, $\bar{j} = 5$.

$$\max\left\{\frac{-9}{1/4}, \frac{-3}{3/4}\right\} \leq \Delta b_3 \leq \frac{-15}{-5/4}$$

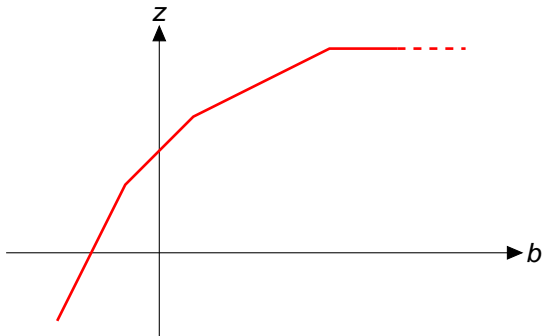
$$-4 \leq \Delta b_3 \leq 12$$

Può scendere di 4 unità e salire di 12 unità

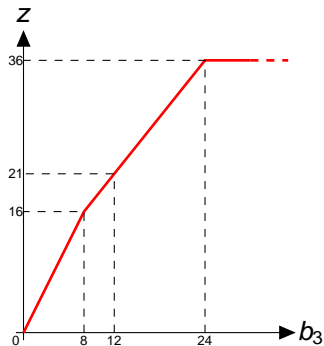
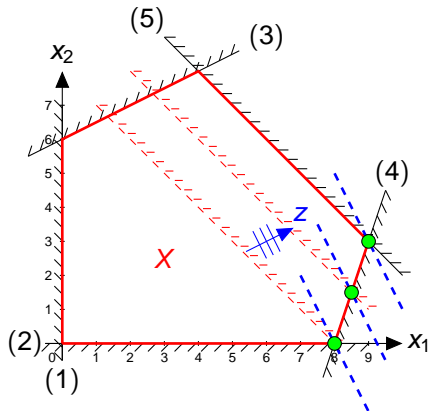
Analisi parametrica

L'analisi parametrica studia come z^* dipende dal valore del termine noto di un vincolo prescelto.

Il risultato è una funzione lineare a tratti: ogni suo segmento corrisponde ad una base ottima ed ogni punto di discontinuità ad un cambio di base.



Analisi parametrica



Interpretazione economica della PL

$$\begin{aligned} \text{maximize } z &= 6x_1 + 14x_2 + 13x_3 \\ \text{s.t. } 0.5x_1 + 2x_2 + x_3 &\leq 24 \\ x_1 + 2x_2 + 4x_3 &\leq 60 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

Interpretazione economica:

- tre prodotti richiedono due risorse;
- le variabili rappresentano le quantità prodotte;
- i coefficienti della f.o. rappresentano i profitti unitari;
- i termini noti rappresentano le quantità di risorsa disponibili.

Tableau all'ottimo:

294		0	9	0	11	1/2
36		1	6	0	4	-1
6		0	-1	1	-1	1/2

$$B^* = \{1, 3\} \quad x^* = [36 \ 0 \ 6 \ 0 \ 0] \quad z^* = 294$$

Interpretazione economica della PL

$$\begin{aligned} \text{maximize } z &= 6x_1 + 14x_2 + 13x_3 \\ \text{s.t. } 0.5x_1 + 2x_2 + x_3 &\leq 24 \\ x_1 + 2x_2 + 4x_3 &\leq 60 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

z da minimizzare, sto cambiando il segno

coefficienti di costo ridotto, cioè i coefficienti delle variabili di slack

294	0	9	0	11	1/2
36	1	6	0	4	-1
6	0	-1	1	-1	1/2

Cioè se la risorsa fosse 23 anziché 24. La variabile di slack in questo vincolo è x_4 , quindi diminuirebbe di 11 (coeff. di x_4)

$$z = 294 - 9x_2 - 11x_4 - \frac{1}{2}x_5$$

Se diminuisse di un'unità la quantità di risorsa 1 disponibile, z peggiorerebbe di 11 unità.

I c.c.r. delle colonne di slack all'ottimo indicano i **prezzi-ombra** delle corrispondenti risorse, cioè il massimo prezzo a cui conviene comprare la risorsa e il minimo prezzo a cui conviene venderla.

Il prezzo-ombra di risorse non scarse è nullo. -> perchè non la sto usando tutta, quindi non ha senso nemmeno comprarne altra

Interpretazione economica della PL

$$\begin{aligned} \text{maximize } z &= 6x_1 + 14x_2 + 13x_3 \\ \text{s.t. } 0.5x_1 + 2x_2 + x_3 &\leq 24 \\ x_1 + 2x_2 + 4x_3 &\leq 60 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

294	0	9	0	11	1/2
36	1	6	0	4	-1
6	0	-1	1	-1	1/2

Se si volesse produrre un'unità di **prodotto 2**, si avrebbe

- un ricavo marginale pari a 14 (valore di c_2)
- un consumo di risorse pari a $[2 \ 2]$, che si traduce in un costo pari a $2 \times 11 + 2 \times \frac{1}{2} = 23$

e quindi un profitto marginale pari a -9 (non conveniente), che è infatti il costo ridotto di x_2 .

La produzione subisce un impatto negativo di 23 per ottenere un guadagno di 14. QUindi non ne vale la pena, $23 - 14 = 9 \rightarrow$ perdita netta di 9. Quando si ha un costo ridotto 0 il costo marginale e il profitto si equilibrano, vedi esempio slide successiva

Interpretazione economica della PL

$$\begin{aligned} \text{maximize } z &= 6x_1 + 14x_2 + 13x_3 \\ \text{s.t. } 0.5x_1 + 2x_2 + x_3 &\leq 24 \\ x_1 + 2x_2 + 4x_3 &\leq 60 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

294		0	9	0	11	1/2
36		1	6	0	4	-1
6		0	-1	1	-1	1/2

Il coefficiente di costo ridotto di variabili basiche è nullo, perché i ricavi marginali e i costi marginali risultano uguali.

Per esempio, per la variabile x_1 si ha:

$$6 = \frac{1}{2} \times 11 + 1 \times \frac{1}{2}.$$

Costi ridotti

Il costo ridotto \bar{c}_j di ogni variabile x_j è dato da

$$\bar{c}_j = c_j - \sum_i a_{ij} \lambda_i,$$

dove

- c_j è il coefficiente di x_j nella f.o.,
- a_{ij} è il coefficiente sulla riga i e colonna j nella matrice dei vincoli;
- λ_i è il prezzo-ombra del vincolo i .

Programmazione (lineare) a molti obiettivi

Ricerca operativa

Giovanni Righini



UNIVERSITÀ DEGLI STUDI
DI MILANO

Programmazione a molti obiettivi

La programmazione a molti obiettivi è l'estensione della programmazione matematica al caso in cui siano presenti più funzioni obiettivo in conflitto tra loro, cioè tali che il miglioramento rispetto ad una comporti un peggioramento rispetto ad un'altra.

L'ambito di applicazione della programmazione a multi-obiettivi è vastissimo.

In questo corso ci limiteremo a considerare il caso di problemi di programmazione **lineare** con **due obiettivi**.

Due fasi distinte

In presenza di problemi di ottimizzazione con più obiettivi, il processo risolutivo prevede due fasi distinte:

- Prima fase: calcolo della **regione Pareto-ottima**, cioè dell'insieme delle **soluzioni ammissibili non-dominate**.
- Seconda fase: **scelta di una soluzione** tra quelle Paretiane, individuate nella prima fase.

La prima fase è puramente algoritmica, mentre la seconda implica scelte del decisore.

Per risolvere un problema di ottimizzazione con più obiettivi non si può più utilizzare il concetto di **soluzione ottima**.

Se gli obiettivi sono in conflitto, nessuna soluzione è **ottima**.

Il concetto di **soluzione ottima** viene quindi sostituito da quello di **soluzione non-dominata**.

f.o. in conflitto

Dominanza

soluzioni ammissibili

Dati gli obiettivi $f_1(x), f_2(x), \dots, f_k(x)$ da minimizzare e date due soluzioni ammissibili x' e x'' , x' domina x'' se e solo se

Deve esistere almeno un obiettivo per cui $f_j(x')$ è strettamente migliore di $f_j(x'')$

$$\begin{cases} f_i(x') \leq f_i(x'') \quad \forall i = 1, \dots, k \\ \exists j \in \{1, \dots, k\} : f_j(x') < f_j(x'') \end{cases}$$

\leq perchè sto minimizzando.
Per ogni f.o. x' deve essere NON peggio di x''

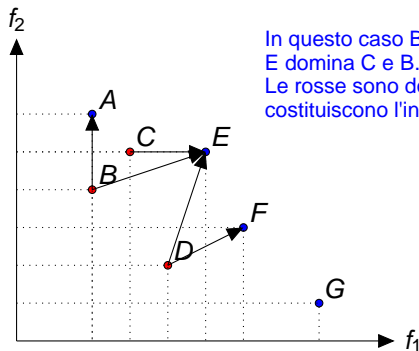
In caso di massimizzazione, bisogna invertire il segno delle disequazioni.

L'insieme delle soluzioni non-dominate è la regione Pareto-ottima del problema.

Obiettivo dell'analisi di un problema a multi-obiettivi è anzitutto determinare con opportuni algoritmi la sua regione Pareto-ottima (prima fase).

Regione Pareto-ottima

Le soluzioni e la regione Pareto-ottima possono essere rappresentate non solo nello spazio delle variabili, ma anche nello **spazio degli obiettivi**.



In questo caso B è dominata sia da A che E.
E domina C e B.

Le rosse sono dominate, mentre le blu costituiscono l'insieme Pareto ottimo.

Soluzioni **Paretiane** e **dominate** nello spazio degli obiettivi
(due obiettivi da massimizzare).

Metodo dei pesi

Il metodo dei pesi consiste nell'ottimizzare una **combinazione convessa delle funzioni-obiettivo**.

maximize $f_1(x)$
maximize $f_2(x)$
maximize \dots
maximize $f_k(x)$
s.t. $x \in X$

maximize $\sum_{i=1}^k \lambda_i f_i(x)$
s.t. $x \in X$

con $\lambda_i \geq 0 \ \forall i = 1, \dots, k$
e $\sum_{i=1}^k \lambda_i = 1$.

Ovviamente la soluzione ottima del problema risultante dipende dal vettore dei pesi λ .

Con due obiettivi lineari e vincoli lineari si può calcolare la regione Paretiana con l'**analisi parametrica**.

Esempio

L'ottimo può essere su questi due tratti, che si ottiene combinando le due f.o.

$$\text{maximize } f_1 = 2x_1 + x_2$$

$$\text{maximize } f_2 = x_2$$

$$\text{s.t. } 3x_1 + x_2 \leq 18$$

$$x_1 + x_2 \leq 8$$

$$x_1 + 3x_2 \leq 18$$

$$x_1, x_2 \geq 0$$

$$\lambda = [\alpha \quad 1 - \alpha] \quad 0 \leq \alpha \leq 1$$

una f.o. la peso con alfa, l'altra con 1-alfa

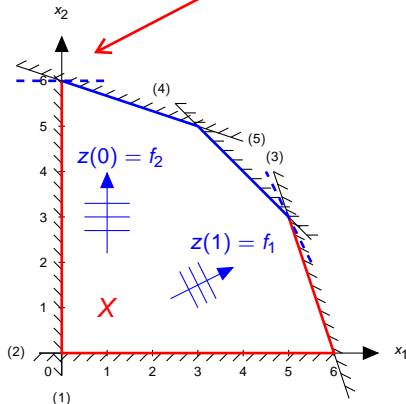
$$\text{maximize } z = 2\alpha x_1 + x_2$$

$$\text{s.t. } 3x_1 + x_2 \leq 18$$

$$x_1 + x_2 \leq 8$$

$$x_1 + 3x_2 \leq 18$$

$$x_1, x_2 \geq 0$$



Esempio

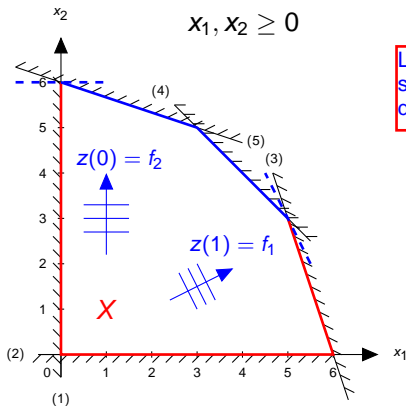
maximize $z = 2\alpha x_1 + x_2$

s.t. $3x_1 + x_2 \leq 18$

$x_1 + x_2 \leq 8$

$x_1 + 3x_2 \leq 18$

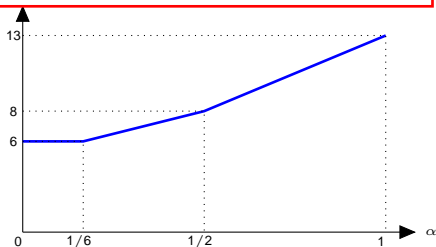
$x_1, x_2 \geq 0$



Analisi parametrica su α :

$$\begin{array}{lll} 0 \leq \alpha \leq \frac{1}{6} & x^* = \begin{bmatrix} 0 \\ 6 \end{bmatrix} & z^* = 6 \\ \frac{1}{6} \leq \alpha \leq \frac{1}{2} & x^* = \begin{bmatrix} 3 \\ 5 \end{bmatrix} & z^* = 5 + 6\alpha \\ \frac{1}{2} \leq \alpha \leq 1 & x^* = \begin{bmatrix} 5 \\ 3 \end{bmatrix} & z^* = 3 + 10\alpha \end{array}$$

L'analisi parametrica è effettuata sull'obiettivo non sui termini noti, non è un problema grazie alla teoria della dualità.



Metodo dei vincoli

Il metodo dei vincoli consiste nell'ottimizzare una delle funzioni-obiettivo, trasformando le altre in vincoli con un termine noto parametrico.

$$\begin{array}{ll} \text{maximize } f_1(x) & \text{maximize } f_1(x) \\ \text{maximize } f_2(x) & \text{s.t. } x \in X \\ \text{maximize } \dots & f_i(x) \geq \beta_i \quad \forall i = 2, \dots, k. \\ \text{maximize } f_k(x) & \\ \text{s.t. } x \in X & \end{array}$$

Ovviamente la soluzione ottima del problema risultante dipende dal vettore dei termini noti β .

Con due obiettivi lineari e vincoli lineari si può calcolare la regione Paretiana con l'analisi parametrica.

Esempio

$$\text{maximize } f_1 = 2x_1 + x_2$$

$$\text{maximize } f_2 = x_2$$

$$\text{s.t. } 3x_1 + x_2 \leq 18$$

$$x_1 + x_2 \leq 8$$

$$x_1 + 3x_2 \leq 18$$

$$x_1, x_2 \geq 0$$

$$\text{maximize } f_1 = 2x_1 + x_2$$

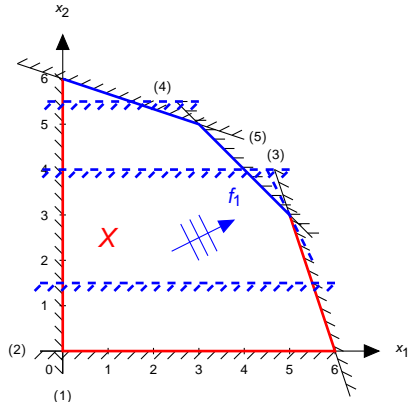
$$\text{s.t. } 3x_1 + x_2 \leq 18$$

$$x_1 + x_2 \leq 8$$

$$x_1 + 3x_2 \leq 18$$

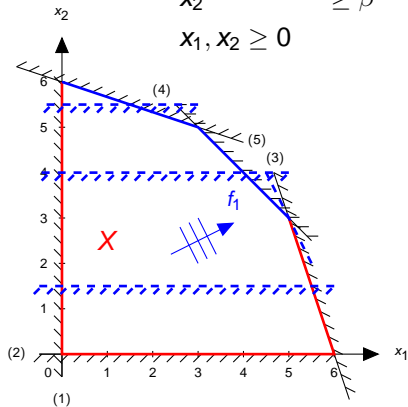
$$x_2 \geq \beta$$

$$x_1, x_2 \geq 0$$



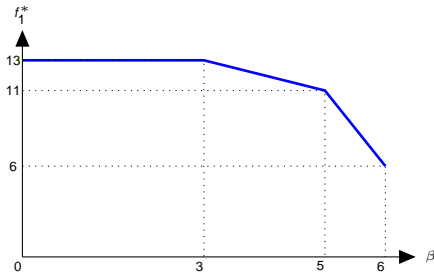
Esempio

$$\begin{aligned}
 &\text{maximize } f_1 = 2x_1 + x_2 \\
 &\text{s.t. } 3x_1 + x_2 \leq 18 \\
 &\quad x_1 + x_2 \leq 8 \\
 &\quad x_1 + 3x_2 \leq 18 \\
 &\quad x_2 \geq \beta \\
 &\quad x_1, x_2 \geq 0
 \end{aligned}$$



Analisi parametrica su β :

$$\begin{aligned}
 \beta \leq 3 & \quad x^* = \begin{bmatrix} 5 \\ 3 \end{bmatrix} & \quad f_1^* = 13 \\
 3 \leq \beta \leq 5 & \quad x^* = \begin{bmatrix} 8 - \beta \\ \beta \end{bmatrix} & \quad f_1^* = 16 - \beta \\
 5 \leq \beta \leq 6 & \quad x^* = \begin{bmatrix} 18 - 3\beta \\ \beta \end{bmatrix} & \quad f_1^* = 36 - 5\beta
 \end{aligned}$$

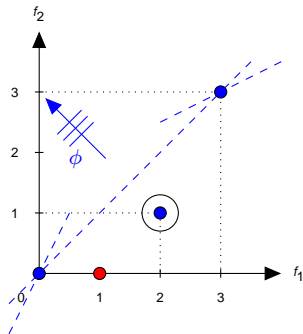


Regioni paretiane continue e discrete

Per problemi lineari nel continuo, il metodo dei pesi e dei vincoli generano correttamente la regione paretiana.

Nel discreto, invece, il metodo dei pesi in generale non garantisce di trovare tutte le soluzioni paretiane: possono esistere soluzioni paretiane che non sono ottime per alcuna scelta dei pesi.

$$\begin{aligned} &\text{minimize } f_1 = y_1 + y_2 + y_3 \\ &\text{maximize } f_2 = x_1 + x_2 + x_3 \\ &\text{s.t. } 2x_1 \leq y_2 + y_3 \\ &\quad 2x_2 \leq y_1 + y_3 \\ &\quad 2x_3 \leq y_1 + y_2 \\ &\quad x \in \{0, 1\} \\ &\quad y \in \{0, 1\} \end{aligned}$$



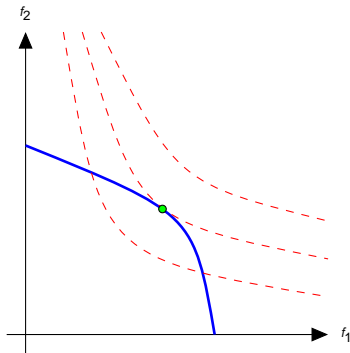
Seconda fase: scelta di una soluzione

La seconda fase del processo decisionale può essere supportata da metodi quantitativi, benché richieda una scelta, non demandabile ad un algoritmo, da parte del decisore.

Esistono vari metodi utilizzabili a questo scopo. Ad esempio:

- Metodo delle curve di indifferenza
- Criterio della massima curvatura
- Criterio del punto utopia
- Criterio degli standard

Metodo delle curve di indifferenza



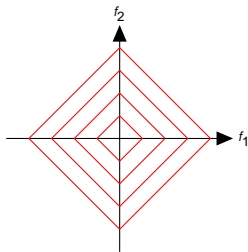
La soluzione scelta è quella in cui una delle curve di indifferenza risulta tangente alla regione Paretiana.

Metodo delle curve di indifferenza

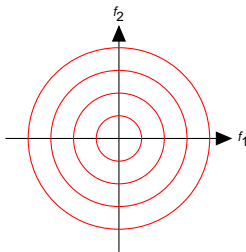
Alcune curve di indifferenza comunemente usate per avere un'espressione analitica:

$$\omega(f(x)) = \left[\sum_{i=1}^k (\lambda_i f_i(x))^p \right]^{\frac{1}{p}}.$$

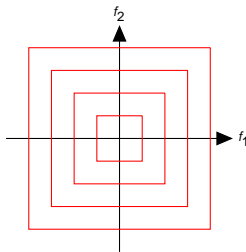
Ad esempio, con $k = 1$ e $\lambda_1 = \lambda_2 = 1$:



$p = 1$

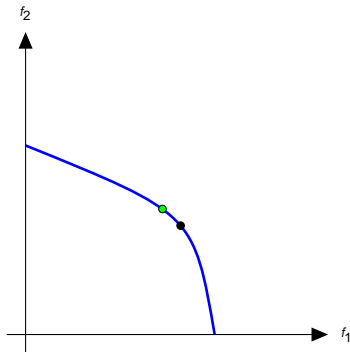


$p = 2$



$p \rightarrow \infty$

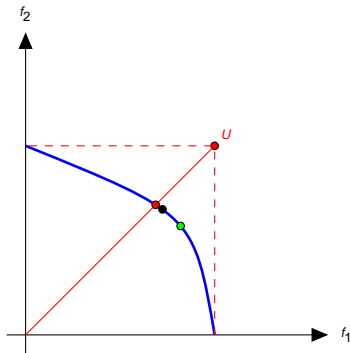
Criterio del punto di massima curvatura



La soluzione scelta è quella per cui ad un piccolo miglioramento di un obiettivo corrisponde un grande peggioramento dell'altro.

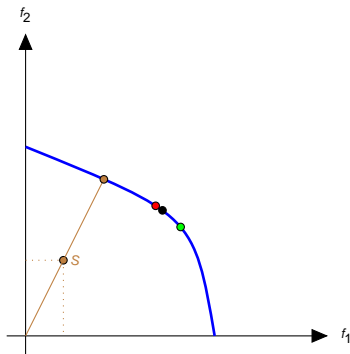
Criterio del punto-utopia

Il **punto-utopia** è la soluzione (in generale non-ammissibile) che nello spazio degli obiettivi ha come coordinate i valori ottimi di ciascuno.



Criterio degli standard

Gli **standard** sono valori-soglia al di sotto dei quali non si vuole che gli obiettivi possano peggiorare.



Programmazione lineare intera

Ricerca Operativa

Giovanni Righini



UNIVERSITÀ DEGLI STUDI
DI MILANO

Ottimizzazione nel discreto

Esistono diverse classi di problemi di ottimizzazione con **variabili discrete**:

- IP: integer programming
- BP: binary programming
- MIP: mixed-integer programming
- CO: combinatorial optimization

Considereremo solo modelli lineari: **Integer Linear Programming (ILP)**.

Per ottimizzare nel discreto possiamo:

- selezionare “buone” *formulazioni lineari* e migliorarle fino a poterle risolvere un problema di PLI come problema di PL;
- *scomporre* il problema in sotto-problemi più piccoli e più facili;
- eseguire una *enumerazione implicita* delle soluzioni.

Ottimalità

Dato un problema di ottimizzazione discreta P

$$z^* = \max\{z(x) : x \in X \subseteq \mathcal{Z}^n\}$$

l'ottimalità si dimostra calcolando un *upper bound* \bar{z} e un *lower bound* \underline{z} , tali che

$$\underline{z} \leq z^* \leq \bar{z}.$$

- Se P è un problema di minimizzazione, \bar{z} è un bound primale e \underline{z} è un bound duale.
- Se P è un problema di massimizzazione, \underline{z} è un bound primale e \bar{z} è un bound duale.

La differenza $\bar{z} - \underline{z}$ è detta **gap di ottimalità**.

Quando $\bar{z} - \underline{z} = 0$ si ha la *garanzia di ottimalità*.

Bounds primali

Un **bound primale** \bar{z} è dato dal valore della funzione obiettivo $z(x)$ in una qualsiasi soluzione ammissibile $\bar{x} \in X$.

$$\bar{z} = z(\bar{x}), \quad \bar{x} \in X.$$

Bounds primali possono essere calcolati in vari modi:

- con algoritmi euristici o meta-euristici (ricerca locale, GRASP,...);
- con algoritmi di approssimazione con garanzia: in tal caso si ha anche un bound duale.

Per alcuni problemi di ottimizzazione discreta è difficile anche trovare una soluzione ammissibile (cioè calcolare un bound primale).

Bounds duali

Un **bound duale** è dato dal valore della funzione obiettivo $z(x)$ in corrispondenza di una soluzione super-ottima \bar{x} . Quindi in generale \bar{x} non è ammissibile.

Ci sono due tecniche principali per calcolare un bound duale per un problema P :

- risolvere all'ottimo un **rilassamento** R di P ;
- trovare una soluzione ammissibile al **duale** D di P .

Rilassamenti

Dato un problema

$$P = \min\{z_P(x) : x \in X(P)\}$$

un problema

$$R = \min\{z_R(x) : x \in X(R)\}$$

è un **rilassamento** di P se valgono le seguenti due condizioni:

- $X(P) \subseteq X(R)$
- $z_R(x) \leq z_P(x) \quad \forall x \in X(P)$.

[In caso di massimizzazione, le disequazioni vanno invertite.]

Corollario: $z_R^* \leq z_P^*$.

Ci sono molti tipi diversi di rilassamento.

Un rilassamento è tanto migliore quando più il suo valore ottimo z_R^* è vicino a z_P^* .

Rilassamento lineare continuo

Quando P è un problema di ottimizzazione discreta

$$P) \min\{z(x) : x \in X, x \in \mathcal{Z}_+^n\},$$

il suo *rilassamento continuo* C è ottenuto da P trascurando le condizioni di integralità:

$$C) \min\{z(x) : x \in X, x \in \mathbb{R}_+^n\}.$$

Quando P è un problema di ottimizzazione *lineare* discreta

$$P) \min\{cx : Ax \leq b, x \in \mathcal{Z}_+^n\},$$

il suo rilassamento continuo LP

$$LP) \min\{cx : Ax \leq b, x \in \mathbb{R}_+^n\}$$

è un problema di programmazione lineare (che sappiamo come risolvere molto efficacemente).

Se $x_{LP}^* \in \mathcal{Z}_+^n$, allora $x_P^* = x_{LP}^*$.

Rilassamenti combinatori

Il rilassamento combinatorio C di un problema di ottimizzazione combinatoria P è ancora un problema di ottimizzazione combinatoria, ma tipicamente molto più facile da risolvere.

Esempio 1:

- P : il TSP asimmetrico;
- C : il problema di matching bipartito di costo minimo.

Esempio 2:

- P : il TSP simmetrico;
- C : il problema dell'1-albero ricoprente di costo minimo.

Rilassamento Lagrangeano

Il rilassamento Lagrangeano LR di un problema di ottimizzazione (lineare discreto) P si ottiene rimuovendo alcuni vincoli e aggiungendo all'obiettivo termini di penalità per la loro violazione.

$$P) \min\{z(x) : Ax \leq b, x \in X \subseteq \mathcal{Z}_+^n\}$$

$$LR) \min\{z_{LR}(x, \lambda) = z(x) + \lambda(Ax - b) : x \in X \subseteq \mathcal{Z}_+^n\}$$

con $\lambda \geq 0$.

metto + perchè sto minimizzando. La penalità è lambda

Esso soddisfa entrambe le condizioni per essere un rilassamento:

- Vincoli: $\{x : Ax \leq b, x \in X\} \subseteq \{x : x \in X\}$
- Obiettivo:
 - $Ax - b \leq 0$ per tutte le soluzioni ammissibili per P ;
 - $\lambda(Ax - b) \leq 0$ per tutte le soluzioni ammissibili per P ;
 - $z_{LR}(x, \lambda) = z(x) + \lambda(Ax - b) \leq z(x)$ per tutte le soluzioni ammissibili per P .

Rilassamento surrogato

Il rilassamento surrogato S di un problema di ottimizzazione (lineare discreto) P si ottiene sostituendo un insieme di vincoli con una loro combinazione convessa.

$$P) \min\{z(x) : Ax \leq b, x \in X \subseteq \mathcal{Z}_+^n\}$$

$$S) \min\{z(x) : \lambda^T Ax \leq \lambda^T b, x \in X \subseteq \mathcal{Z}_+^n\}$$

con $\lambda \geq 0$.

Esso soddisfa le due condizioni per essere un rilassamento:

- Vincoli: $Ax \leq b$ implica $\lambda^T Ax \leq \lambda^T b$ (ma non viceversa).
- Obiettivo: banale, perché non cambia.

Rilassamenti e bounds

I rilassamenti lineare, Lagrangeano e surrogato possono fornire in generale bounds diversi.

In caso di minimizzazione valgono le seguenti relazioni:

$$z_{LP}^* \leq z_{LR}^* \leq z_S^* \leq z^*.$$

Con z_{LR}^* e z_S^* qui si indicano i migliori bounds ottenibili dal rilassamento Lagrangeano e surrogato, scegliendo cioè nel modo migliore i moltiplicatori λ .

Dualità

La seconda tecnica per ottenere un bound duale consiste nel calcolare una soluzione ammissibile per il problema duale di P o per il duale di un suo rilassamento.

Problema lineare duale:

Tolgo la condizione di integralità

$$P)z^* = \min\{cx : Ax \geq b, x \in \mathbb{Z}_+^n\}$$

$$D)w^* = \max\{yb : yA \leq c, y \in \mathbb{R}_+^m\}$$

formano una coppia primale-duale debole.

Non ci possono essere due edge con un vertice in comune

Problema duale combinatorio:

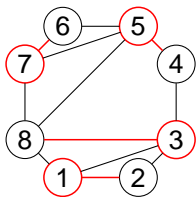
Il problema del massimo matching e il problem del *minimum vertex cover*

$$P)z^* = \max\{1x : Ax \leq 1, x \in \mathcal{B}_+^{|E|}\}$$

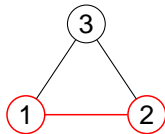
$$D)w^* = \min\{1y : yA \geq 1, y \in \mathcal{B}_+^{|V|}\}$$

dove A è la matrice di incidenza di un grafo $G = (V, E)$, formano una coppia primale-duale debole.

Esempio



$$z^* = 4 \quad w^* = 4$$



$$z^* = 1$$

$$z_{LP}^* = z\left(\left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]\right) = \frac{3}{2}$$

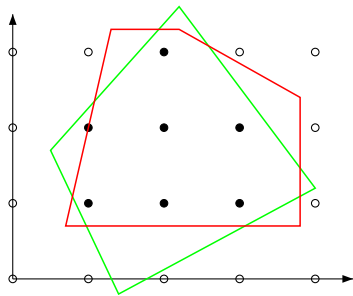
$$w_{LP}^* = w\left(\left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]\right) = \frac{3}{2}$$

$$w^* = 2$$

Il minimo vertex cover, così come il massimo matching richiede 4 vertici, bastano quelli per coprire tutti i vertici del grafo. Non sempre sono uguali, infatti a dx il matching è 1 mentre il minimo vertex cover è 2.

Formulazioni lineari

I problemi di ottimizzazione (lineare) discreti *non* hanno una formulazione unica.



Formulazioni

Dal momento che non sono uniche, ha senso

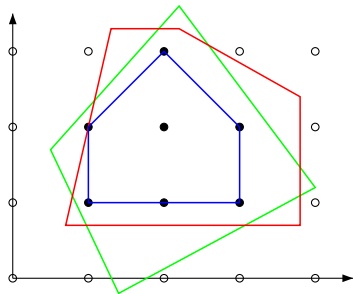
- *confrontare* formulazioni,
- *migliorare* formulazioni.

Una formulazione migliore si traduce in un algoritmo più efficiente.

La **formulazione ideale** di un **problema di programmazione lineare discreta** è quella che consente di risolverlo come se fosse un problema di **programmazione lineare nel continuo**.

Formulazione ideale

La formulazione di un problema di programmazione lineare corrisponde ad un *poliedro*.



I vincoli della **formulazione ideale** corrispondono al *guscio convesso* delle soluzioni intere.

Guscio convesso (*convex hull*)

Dato un insieme discreto

$$X = \{x_1, \dots, x_t\} \text{ with } x_i \in \mathbb{R}^n \forall i = 1, \dots, t,$$

il suo *guscio convesso* è il poliedro

$$\text{conv}(X) = \{x \in \mathbb{R}^n : x = \sum_{i=1}^t \lambda_i x_i, \sum_{i=1}^t \lambda_i = 1, \lambda_i \geq 0 \forall i = 1, \dots, t\}.$$

è la combinazione
convessa dei punti estremi.
Lamda è un coefficiente
non negativo.

E' un *poliedro* i cui punti estremi sono elementi dell'insieme discreto X .

Data una formulazione P e l'insieme discreto X delle sue soluzioni ammissibili, vale la relazione

$$X \subseteq \text{conv}(X) \subseteq P.$$

Polyhedral combinatorics

In generale

- non conosciamo la formulazione ideale dei problemi di ottimizzazione lineare intera;
- il numero dei vincoli del guscio convesso può crescere esponenzialmente con la dimensione dell'istanza.

Conosciamo la formulazione ideale solo per alcuni particolari problemi di ottimizzazione discreta: il problema del cammino minimo su grafo, il problema del matching bipartito di costo minimo, il problema dell'albero ricoprente di costo minimo,...

La disciplina che studia come selezionare e migliorare le formulazioni lineari dei problemi di PLI è la *polyhedral combinatorics*.

Scelta della formulazione: esempio

In molti problemi di ottimizzazione discreta con vincoli di capacità (Bin Packing Problem, Facility Location Problem,...), ci sono vincoli di questa forma:

\mathcal{N} è l'insieme di oggetti mentre

\mathcal{M} è un insieme di contenitori

$x_{ij}=1$ significa che ho messo i dentro j

$$\sum_{i \in \mathcal{N}} x_{ij} \leq |\mathcal{N}| y_j \quad \forall j \in \mathcal{M},$$

capacità del contenitore
 N vale 1 se y si usa, 0 altrimenti

che esprime una condizione logica che lega le variabili x e y :

$$\begin{cases} \exists (i, j) \in \mathcal{N} \times \mathcal{M} : x_{ij} > 0 & \Rightarrow y_j = 1 \\ \exists j \in \mathcal{M} : y_j = 0 & \Rightarrow x_{ij} = 0 \quad \forall i \in \mathcal{N}. \end{cases}$$

se ij vale 1 significa per forza che
ho messo un oggetto in j

La stessa condizione può essere espressa con

$$x_{ij} \leq y_j \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{M}. \quad (2)$$

La formulazione (1) richiede $|\mathcal{M}|$ vincoli.

La formulazione (2) richiede $|\mathcal{M}| |\mathcal{N}|$ vincoli.

detti vincoli di capacità

Queste due formulazioni non sono equivalenti per quanto riguarda il rilassamento continuo, infatti definiscono poliedri diversi. Per il solutore è meglio la seconda, ma come facciamo a capire quale è meglio?

Scelta della formulazione: esempio

Sommando tra loro i vincoli (2) per ogni $i \in \mathcal{N}$ si ottiene

$$\sum_{i \in \mathcal{N}} x_{ij} \leq \sum_{i \in \mathcal{N}} y_j \quad \forall j \in \mathcal{M}$$

tutti i lambda sono uguali a 1

cioè proprio i vincoli (1): $\sum_{i \in \mathcal{N}} x_{ij} \leq |\mathcal{N}| y_j \quad \forall j \in \mathcal{M}$.

Quindi ogni vincolo (1) è un vincolo *surrogato* di alcuni vincoli (2).

I vincoli (2) implicano i vincoli (1) ma non viceversa.

Ci sono soluzioni che soddisfano (1) ma violano (2):

$$\begin{cases} x_{ij} = 1 & \forall j \in \mathcal{M}, \forall i \in \mathcal{N} : i \in [k(j-1) + 1, \dots, kj] \\ y_j = 1/|\mathcal{M}| & \forall j \in \mathcal{M} \end{cases}$$

dove $k = |\mathcal{N}|/|\mathcal{M}|$.

I vincoli (2) generano una **migliore formulazione** rispetto ai vincoli (1).

Il poliedro con i vincoli (2) contiene il poliedro con i vincoli (1).

Algoritmi “cutting planes”

Dato un problema di PLI

$$P^{(k)} = \max\{cx : Ax \leq b, x \in \mathcal{Z}_+^n\}$$

consideriamo il suo rilassamento continuo

$$L^{(k)} = \max\{cx : Ax \leq b, x \in \mathbb{R}_+^n\}$$

(Cutting planes)

e la sua soluzione ottima $x^{*(k)}$. Quindi, generiamo un insieme di *disuguaglianze valide* $Qx \leq q$ tali che.

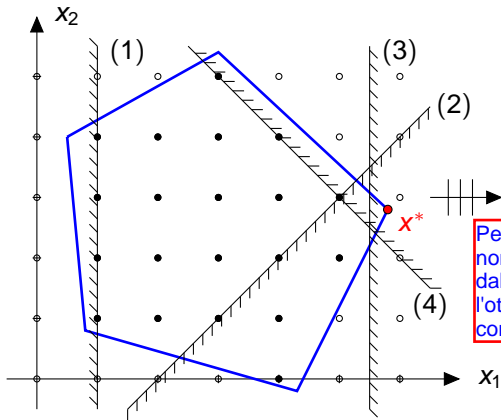
- $Qx \leq q \quad \forall x \in \mathcal{Z}_+^n : Ax \leq b$
- $Qx^{*(k)} > q$ (rendere inammissibile il punto frazionario)

e otteniamo così una formulazione più stretta

$$P^{(k+1)} = \max\{cx : Ax \leq b, Qx \leq q, x \in \mathcal{Z}_+^n\}.$$

Si ottiene così un nuovo poliedro con una formulazione più stretta dato che ha dei vincoli in più

Disuguaglianze valide: esempio



Perchè sto tagliando ma non sto ottimizzando dall'altra parte, l'ottimizzazione rimane comunque il pallino rosso.

La disequazione (1) è valida ma inutile: non "taglia" x^* .

La disequazione (2) non è valida: "taglia" alcune soluzioni ammissibili intere.

La disequazione (3) è valida e utile.

La disequazione (4) è anche *facet defining*.

Procedura di Chvátal-Gomory

Consideriamo un problema di PLI con insieme ammissibile

$$X = \{x \in \mathbb{Z}_+^n : Ax \leq b\}$$

condizione di integralità + vincoli lineari

dove A ha m righe e n colonne.

Scegliamo un vettore $u \in \mathbb{R}_+^m$:

- $\sum_{j=1}^n ua_j x_j \leq ub$ è valida perché $ax \leq b$ e $u \geq 0$. (sto moltiplicando a sx i vincoli)
- $\sum_{j=1}^n \lfloor ua_j \rfloor x_j \leq ub$ è valida perché $x \geq 0$. (arrotondo per difetto i coeff del primo membro)
- $\sum_{j=1}^n \lfloor ua_j \rfloor x_j \leq \lfloor ub \rfloor$ è valida perché x è intero. (arrotondo per difetto anche il termine noto)

Ogni disuguaglianza valida può essere generata con questa procedura in un numero finito di passi.

L'efficacia della procedura dipende dalla scelta di u .

Algoritmi “cutting planes”

Gli algoritmi “cutting planes” iterativamente risolvono il rilassamento continuo L di un problema discreto P e rafforzano la sua formulazione generando ulteriori vincoli (*cutting planes*), in modo tale che la soluzione ottima del rilassamento continuo all’iterazione k diventi inammissibile all’iterazione $k + 1$.

- Pro:
 - se i piani di taglio sono generati in modo efficace, l’algoritmo può garantire di trovare la soluzione ottima discreta senza fare ricorso ad altre tecniche (ad es. enumerazione implicita);
 - una formulazione più stretta, anche se non ideale, può fornire *bounds duali* più efficaci in un algoritmo branch-and-bound.
- Contro:
 - è necessaria una procedura apposita per generare iterativamente disuguaglianze valide e utili: è chiamata *algoritmo di separazione*. Se il problema originale è difficile (*NP-hard*), anche il problema di separazione lo è.

Algoritmi “cutting planes”: pseudo-codice

Begin

$t:=0$; $P^{(0)}:=P$; [P è il rilassamento continuo]

repeat

$z^{*(t)}:=\max\{cx : x \in P^{(t)}\}$

$x^{*(t)}:=\operatorname{argmax}\{cx : x \in P^{(t)}\}$

if $x^{*(t)} \notin \mathcal{Z}^n$ **then**

Genera una disuguaglianza valida $\pi x \leq \pi_0 : \pi x^{*(t)} > \pi_0$

$P^{(t+1)}:=P^{(t)} \cap \{x : \pi x \leq \pi_0\}$

$t := t + 1$

end if

until $(x^{*(t)} \in \mathcal{Z}^n) \vee$ (no inequalities found)

End

t = indice iterazione

se la disuguaglianza non è intera procedo iterativamente.

Si esce dal ciclo quando trovo l'ottimo nel discreto oppure non si riesce a trovare una disuguaglianza valida.

Algoritmi “cutting planes”

Dopo ogni iterazione $z^{*(t)}$ è un bound duale valido.

Può capitare che non venga trovata nessuna disuguaglianza valida se l'algoritmo di separazione è ristretto a cercarla all'interno di specifici sottinsiemi di disuguaglianze con una struttura particolare, che non bastano per descrivere completamente il guscio convesso del problema di PLI.

Tagli di Gomory

Data una soluzione frazionaria x^* del rilassamento continuo di un problema di PLI, si utilizza la procedura di Chvátal-Gomory sul vincolo associato ad una variabile frazionaria: si ottiene così una disuguaglianza valida violata da x^* .

Dato un problema di PLI

obiettivo: vincoli, variabili, condizioni di integralità

$$P) \max\{cx : ax = b, x \geq 0, x \in \mathbb{Z}^n\}$$

ed il suo rilassamento continuo

$$LP) \max\{cx : ax = b, x \geq 0\}$$

siano x^* e z^* la soluzione ottima di LP e il suo valore.

$$z^* = \bar{a}_{00} + \sum_{j \in N^*} \bar{a}_{0j} x_j^*$$

a segntato indica (riga, colonna) del tableau

$$\begin{cases} x_{B^* i}^* + \sum_{j \in N^*} \bar{a}_{ij} x_j^* = \bar{a}_{i0} & \forall i = 1, \dots, m \\ x^* \geq 0 \end{cases} \quad (3)$$

dove B^* e N^* sono gli indici delle variabili in base e fuori base in x^* .

Tagli di Gomory

Se x^* non è intero, esiste almeno un vincolo \hat{i} tale che \bar{a}_{i0} non è intero.

Eseguendo la procedura di Chvátal-Gomory su di esso si ottiene:

$$x_{B^*\hat{i}} + \sum_{j \in N^*} \lfloor \bar{a}_{ij} \rfloor x_j \leq \lfloor \bar{a}_{i0} \rfloor.$$

Sottraendo questa disuguaglianza dal vincolo di uguaglianza

$$x_{B^*\hat{i}} + \sum_{j \in N^*} \bar{a}_{ij} x_j^* = \bar{a}_{i0}$$

si ottiene il **taglio di Gomory**:

$$\sum_{j \in N^*} f_{ij} x_j \geq f_{i0}$$

dove $f_{ij} = \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor$ e $f_{i0} = \bar{a}_{i0} - \lfloor \bar{a}_{i0} \rfloor$.

Anche la variabile di slack/surplus associata a questa disuguaglianza è intera.

Un esempio

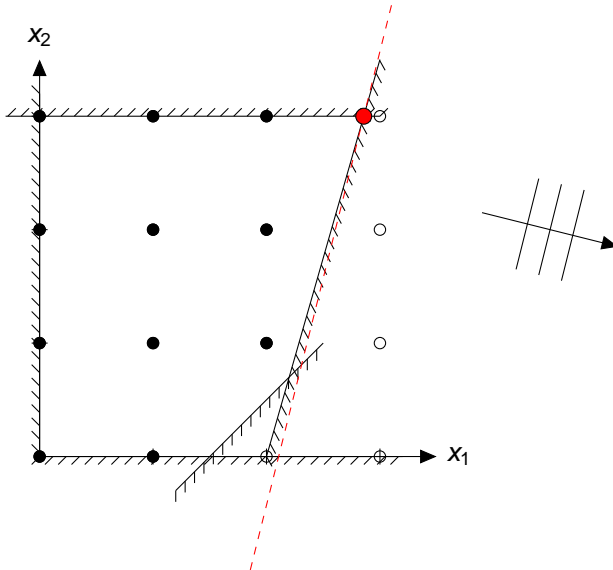
$$\begin{aligned}\text{maximize } z &= 4x_1 - x_2 \\ 7x_1 - 2x_2 &\leq 14 \\ x_2 &\leq 3 \\ 2x_1 - 2x_2 &\leq 3 \\ x &\geq 0 \text{ (integer)}\end{aligned}$$

Risolvendo il rilassamento continuo, si ottiene $B^* = \{1, 2, 5\}$,

$N^* = \{3, 4\}$:

$$\begin{aligned}z &= \frac{59}{7} - \frac{4}{7}x_3 - \frac{1}{7}x_4 \\ x_1 &+ \frac{1}{7}x_3 + \frac{2}{7}x_4 = \frac{20}{7} \\ x_2 &+ x_4 = 3 \\ -\frac{2}{7}x_3 + \frac{10}{7}x_4 + x_5 &= \frac{23}{7} \\ x &\geq 0\end{aligned}$$

Un esempio



Un esempio

$$\begin{aligned} z &= \frac{59}{7} - \frac{4}{7}x_3 - \frac{1}{7}x_4 \\ x_1 + \frac{1}{7}x_3 + \frac{2}{7}x_4 &= \frac{20}{7} \\ x_2 + x_4 &= 3 \\ -\frac{2}{7}x_3 + \frac{10}{7}x_4 + x_5 &= \frac{23}{7} \\ x &\geq 0 \end{aligned}$$

Dal primo vincolo si genera un taglio di Gomory:

$$x_1^* = \frac{20}{7} \Rightarrow \frac{1}{7}x_3 + \frac{2}{7}x_4 \geq \frac{6}{7}.$$

La sua variabile ausiliaria è

$$s_1 = -\frac{6}{7} + \frac{1}{7}x_3 + \frac{2}{7}x_4.$$

Un esempio

Dai vincoli

$$\begin{aligned}x_1 + \frac{1}{7}x_3 + \frac{2}{7}x_4 &= \frac{20}{7} \\ x_2 + x_4 &= 3\end{aligned}$$

si ottiene

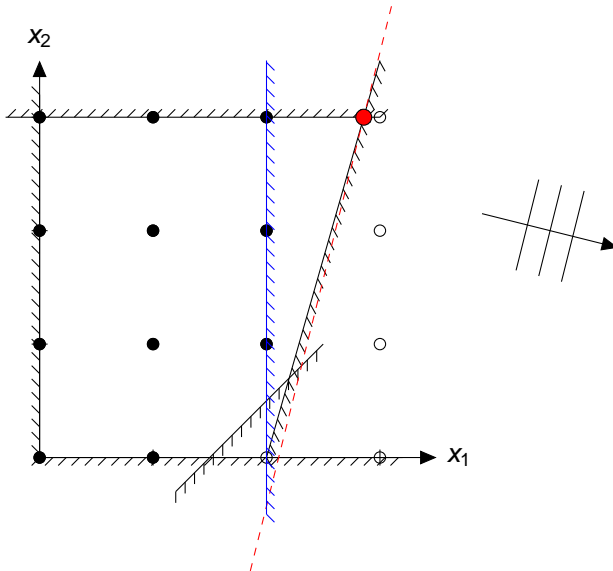
$$\begin{aligned}x_3 &= -7x_1 + 2x_2 + 14 \\ x_4 &= -x_2 + 3\end{aligned}$$

e l'equazione del taglio di Gomory

$$\frac{1}{7}x_3 + \frac{2}{7}x_4 \geq \frac{6}{7}$$

si può riscrivere come

$$x_1 \leq 2.$$

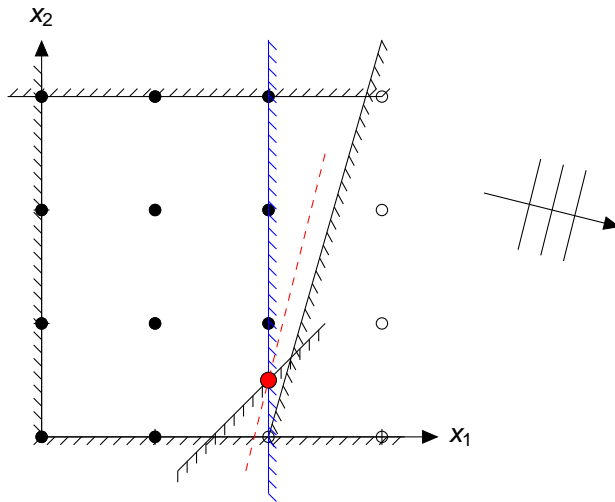


Un esempio

Ri-ottimizzando si ottiene:

$$\begin{aligned} z &= \frac{15}{2} - \frac{1}{2}x_5 - 3s_1 \\ x_1 \quad \quad \quad &+ s_1 = 2 \\ x_2 \quad \quad -\frac{1}{2}x_5 + s_1 &= \frac{1}{2} \\ x_3 \quad \quad -x_5 - 5s_1 &= 1 \\ x_4 + \frac{1}{2}x_5 - s_1 &= \frac{5}{2} \\ x, s &\geq 0 \end{aligned}$$

Un esempio



Un esempio

$$\begin{aligned} z &= \frac{15}{2} - \frac{1}{2}x_5 - 3s_1 \\ x_1 \quad \quad \quad + s_1 &= 2 \\ x_2 \quad \quad - \frac{1}{2}x_5 + s_1 &= \frac{1}{2} \\ x_3 \quad \quad - x_5 - 5s_1 &= 1 \\ x_4 + \frac{1}{2}x_5 - s_1 &= \frac{5}{2} \\ x, s &\geq 0 \end{aligned}$$

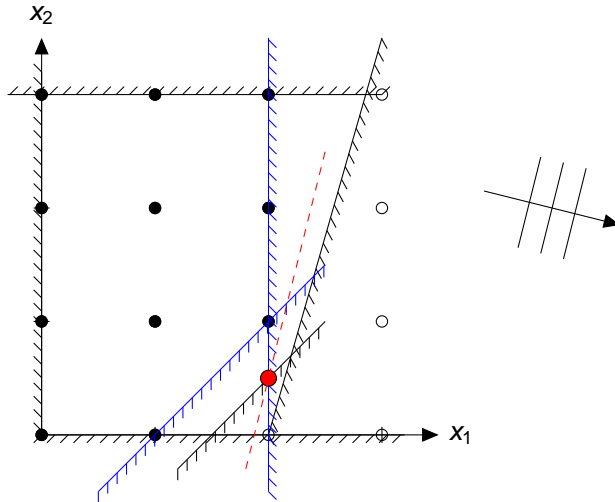
Dal secondo vincolo si può generare un taglio di Gomory:

$$x_2^* = \frac{1}{2} \Rightarrow \frac{1}{2}x_5 \geq \frac{1}{2} \Rightarrow x_1 - x_2 \leq 1.$$

La sua variabile ausiliaria è

$$s_2 = -\frac{1}{2} + \frac{1}{2}x_5.$$

Un esempio



Un esempio

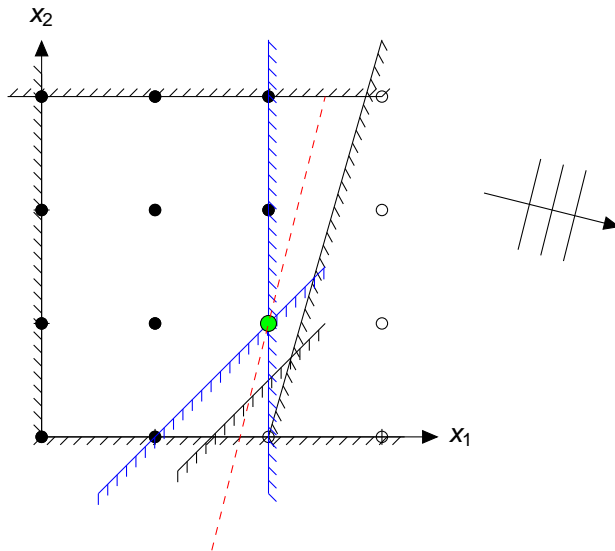
Ri-ottimizzando ancora, si ottiene:

$$\begin{array}{rcll} z = 7 & -3s_1 - s_2 & & \\ x_1 & +s_1 & & = 2 \\ x_2 & +s_1 - s_2 & & = 1 \\ x_3 & -5s_1 - 2s_2 & & = 2 \\ x_4 & -s_1 + s_2 & & = 2 \\ x_5 & & -2s_2 & = 1 \end{array}$$

$$x, s \geq 0$$

Ora la soluzione ottima del rilassamento continuo è intera e quindi è anche la soluzione ottima discreta.

Un esempio



Branch-and-bound

Giovanni Righini

Ricerca Operativa



UNIVERSITÀ DEGLI STUDI
DI MILANO

Ottimizzazione discreta

I problemi di ottimizzazione discreta in generale sono molto difficili da risolvere perché:

- il numero di soluzioni cresce esponenzialmente con numero di variabili;
- gli strumenti del calcolo differenziale, come le derivate (utili per caratterizzare i punti di ottimo) non sono disponibili.

A causa della **esplosione combinatoria** del numero di soluzioni, **l'enumerazione esplicita** non è praticabile.

Tuttavia esistono tecniche di **enumerazione implicita**:

- branch-and-bound,
- programmazione dinamica.

Branch-and-bound

In un algoritmo branch-and-bound

- un problema difficile \mathcal{P} viene ricorsivamente scomposto in più sotto-problemi $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$ più facili.

La scomposizione (**branching**, cioè ramificazione) deve rispettare la seguente condizione per assicurare la correttezza dell'algoritmo:

$$\mathcal{X}(\mathcal{P}) = \bigcup_{i=1}^n \mathcal{X}(\mathcal{F}_i).$$

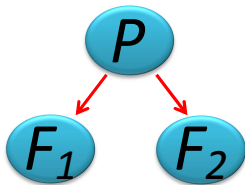
- la soluzione ottima di \mathcal{P} è determinata confrontando le soluzioni ottime dei sotto-problemi originati da esso.

In caso di minimizzazione:

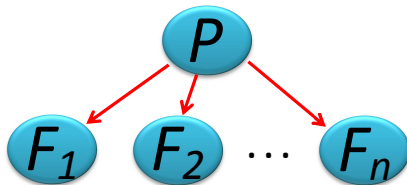
$$z^*(\mathcal{P}) = \min_{i=1, \dots, n} \{z^*(\mathcal{F}_i)\}.$$

Il branch-and-bound tree

La scomposizione ricorsiva di problemi in sotto-problemi genera un'arborescenza (detta anche *decision tree* o *search tree*), in cui la radice corrisponde al problema originale \mathcal{P} ed ogni altro nodo corrisponde ad un sotto-problema.



Branching binario



Branching n -ario

Branching

A scopo di efficienza, la scomposizione solitamente implica una partizione di $\mathcal{X}(\mathcal{P})$ in sottinsiemi disgiunti di modo che nessuna soluzione debba essere (implicitamente) considerata più di una volta:

$$\mathcal{X}(\mathcal{F}_i) \cap \mathcal{X}(\mathcal{F}_j) = \emptyset \quad \forall i \neq j = 1, \dots, n.$$

Ci sono due modi principali di fare branching:

- fissaggio di variabili;
- inserzione di vincoli.

Ogni sotto-problema è una **restrizione** del suo predecessore ed un **rilassamento** dei suoi successori.

Branching binario

Regole di branching comuni sono le seguenti.

- **Branching su una variabile binaria.**

Una variabile binaria x viene selezionata.

Due sotto-problemi vengono generati fissando $x = 0$ in uno e $x = 1$ nell'altro.

- **Branching su un vincolo intero.**

Vengono scelti un vettore di variabili intere (x_1, x_2, \dots, x_n) , un opportuno vettore di coefficienti interi (a_1, a_2, \dots, a_n) e un opportuno termine noto intero k .

Vengono generati due sotto-problemi inserendo i vincoli $ax \leq k$ in uno e $ax \geq k + 1$ nell'altro.

Branching n -ario

Regole di branching n -ario sono le seguenti.

- **Branching su una variabile intera.**

Viene selezionata una variabile intera $x \in [1, \dots, n]$.

Vengono generati n sotto-problemi fissando $x = 1, x = 2, \dots, x = n$.

- **Branching su n variabili binarie.**

Viene scelto un vettore di n variabili binarie (x_1, x_2, \dots, x_n) .

Vengono generati $n + 1$ sotto-problemi fissando alcune variabili come segue (una riga per ogni sotto-problema):

$$x_1 = 1$$

$$x_1 = 0, x_2 = 1$$

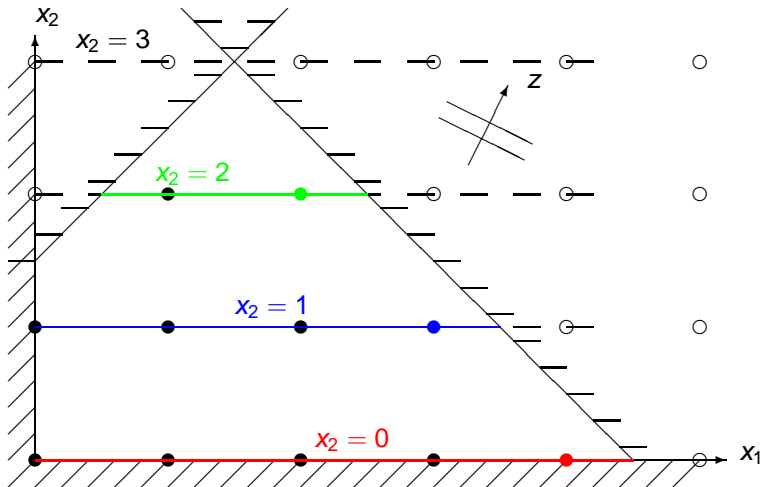
$$x_1 = x_2 = 0, x_3 = 1$$

...

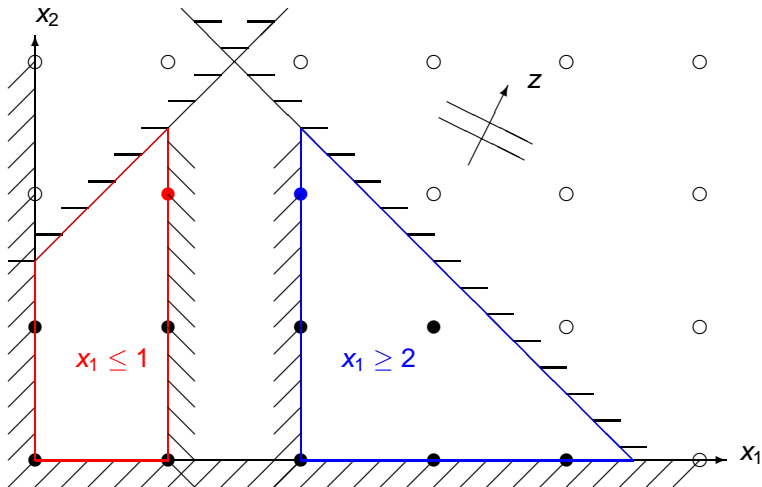
$$x_1 = x_2 = \dots = x_{n-1} = 0, x_n = 1$$

$$x_1 = x_2 = \dots = x_n = 0$$

Branching tramite fissaggio di variabili



Branching tramite inserzione di vincoli



Foglie dell'albero

Solitamente un sotto-problema è “risolto” dal branching, cioè è sostituito da altri sotto-problemi.

Tuttavia questa procedura ricorsiva termina quando il sotto-problema corrente...

- ...è inammissibile;
- ...è risolto all'ottimo;
- ...può essere rtrascurato.

Tutti e tre i casi possono essere scoperti risolvendo un **rilassamento** del sotto-problema corrente.

Rilassamenti

Dato un problema \mathcal{P} ,

$$\begin{aligned} &\text{minimize } z_{\mathcal{P}}(x) \\ &\text{s.t. } x \in \mathcal{X}_{\mathcal{P}} \end{aligned}$$

un problema \mathcal{R}

$$\begin{aligned} &\text{minimize } z_{\mathcal{R}}(x) \\ &\text{s.t. } x \in \mathcal{X}_{\mathcal{R}} \end{aligned}$$

è un **rilassamento** di \mathcal{P} se e solo se valgono le due condizioni:

- $\mathcal{X}_{\mathcal{P}} \subseteq \mathcal{X}_{\mathcal{R}}$
- $z_{\mathcal{R}}(x) \leq z_{\mathcal{P}}(x) \quad \forall x \in \mathcal{X}_{\mathcal{P}}.$

Il valore ottimo del rilassamento non è mai peggiore del valore ottimo del problema originale:

$$z_{\mathcal{R}}^* \leq z_{\mathcal{P}}^*.$$

Rilassamenti

Come conseguenza della definizione di rilassamento, valgono questi corollari.

Corollario 1. Se \mathcal{R} è inammissibile, anche \mathcal{P} è inammissibile.

Corollario 2. Se x^* è ottima per \mathcal{R} ed è ammissibile per \mathcal{P} e $z_{\mathcal{R}}(x) = z_{\mathcal{P}}(x)$, allora x^* è ottima anche per \mathcal{P} .

Corollario 3. Se $z_{\mathcal{R}}^* \geq \bar{z}$, allora $z_{\mathcal{P}}^* \geq \bar{z}$.

Il Corollario 3 è sfruttato nell'operazione di **bounding**.

Bounding

Il bounding consiste nell'associare un **bound duale** ad ogni sotto-problema \mathcal{F} .

Poiché

$$z_{\mathcal{R}}^* \leq z_{\mathcal{P}}^*$$

il valore ottimo di $\mathcal{R}(\mathcal{F})$ (un rilassamento di \mathcal{F}) fornisce un bound duale ogni sotto-problema \mathcal{F} :

$$z_{\mathcal{R}(\mathcal{F})}^* \leq z_{\mathcal{F}}^*.$$

Il bound duale è confrontato con un **bound primale** che corrisponde al valore $z_{\mathcal{P}}(\bar{x})$ di una soluzione ammissibile $\bar{x} \in \mathcal{X}(\mathcal{P})$.

Se il bound duale di \mathcal{F} risulta essere non-migliore del bound primale, allora \mathcal{F} può essere scartato.

If $z_{\mathcal{R}(\mathcal{F})}^* \geq z_{\mathcal{P}}(\bar{x})$ then Fathom \mathcal{F} .

Bounding

La correttezza del bounding è data dalla concatenazione di due disuguaglianze.

- La prima garantisce che nessuna soluzione può esistere in $\mathcal{X}(\mathcal{F})$ con un valore migliore di $z_{\mathcal{R}(\mathcal{F})}^*$, poiché

$$z_{\mathcal{F}}^* \geq z_{\mathcal{R}(\mathcal{F})}^*.$$

- La seconda è $z_{\mathcal{R}(\mathcal{F})}^* \geq z_{\mathcal{P}}(\bar{x})$.

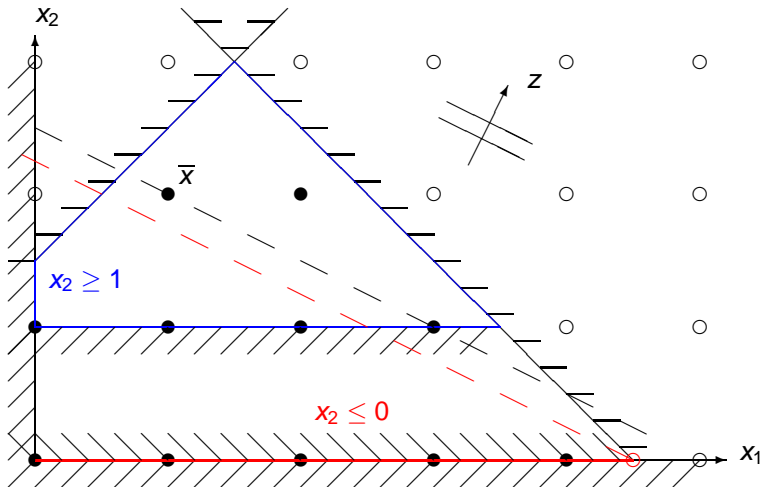
Concatenandole si conclude che

$$z_{\mathcal{F}}^* \geq z_{\mathcal{R}(\mathcal{F})}^* \geq z_{\mathcal{P}}(\bar{x})$$

che significa che risolvere il problema \mathcal{F} all'ottimo è inutile, perché esso non può fornire alcuna soluzione migliore di quella già nota, \bar{x} .

Scartare sotto-problemi in un algoritmo branch-and-bound è cruciale per risparmiare tempo e memoria.

Esempio



Strategia di visita dell'albero

Ogni volta che due o più sotto-problemi vengono generati, essi vengono appesi ad una lista di **nodi aperti**, cioè di sotto-problemi da risolvere.

Questo è necessario perché l'algoritmo viene eseguito su una macchina seriale e i sotto-problemi non possono essere esaminati in parallelo.

La politica seguita per decidere quali nodi visitare per primi è detta **search strategy**.

Il *sotto-problema corrente* è quello che viene risolto ad un generico istante durante l'esecuzione dell'algoritmo.

Strategia di visita dell'albero

Si possono usare vari criteri per gestire la lista dei nodi aperti:

- FIFO: breadth-first search
- LIFO: depth-first search
- Lista ordinata: best-first search

La Best-first search è solitamente basata sul valore del bound duale: vengono esplorati prima i nodi più promettenti.

Per mantenere la lista ordinata è utile utilizzare uno *heap*.

Modelli di ottimizzazione discreta

Giovanni Righini

Ricerca Operativa



UNIVERSITÀ DEGLI STUDI
DI MILANO

Ottimizzazione discreta

Molto spesso le variabili nei problemi di ottimizzazione rappresentano **quantità**, che possono essere **continue** o **discrete**.

Il secondo caso si ha quando le quantità sono necessariamente multipli interi di unità non frazionabili: numero di pallets in un container, numero di persone in un gruppo, numero di veicoli da utilizzare per un trasporto...

Questi casi danno origine a modelli con **variabili intere**, solitamente non-negative e con un dominio finito.

Variabili binarie

In altri casi, invece, le variabili **non rappresentano quantità** e quindi

- non hanno un'unità di misura
- non ammettono approssimazioni.

Si tratta dei modelli con **variabili binarie**, che hanno come dominio l'insieme $\{0, 1\}$.

Le variabili binarie hanno un'enorme importanza dal punto di vista modellistico.

$$x_i = \begin{cases} 1 & \text{capita evento } i \\ 0 & \text{non capita evento } i \end{cases}$$

Variabili binarie

Le relazioni tra variabili binarie esprimono condizioni logiche:

$$\sum_{i=1}^N x_i \leq 1 \Leftrightarrow \text{Non deve capitare più di uno tra } N \text{ eventi}$$

$$\sum_{i=1}^N x_i = 1 \Leftrightarrow \text{Deve capitare uno tra } N \text{ possibili eventi}$$

$$\sum_{i=1}^N x_i \geq 1 \Leftrightarrow \text{Deve capitare almeno uno di } N \text{ possibili eventi}$$

$$x_1 = x_2 \Leftrightarrow \text{I due eventi devono capitare entrambi oppure nessuno dei due}$$

$$x_1 \leq x_2 \Leftrightarrow \text{L'evento 1 può verificarsi solo se si verifica l'evento 2}$$

Variabili binarie

Le variabili binarie sono usate per selezionare sottinsiemi di un insieme:

$$\sum_{i=1}^N c_i x_i \Leftrightarrow \sum_{i \in S} c_i$$

dove S è un sottinsieme di $\{1, \dots, N\}$ corrispondente al vettore caratteristico x :

$$x_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$

Variabili binarie

Le variabili binarie sono usate per eliminare i “se” dai modelli.

$$\begin{cases} 0 \leq y \leq u & \text{se } x = 1 \\ y = 0 & \text{se } x = 0 \end{cases} \Leftrightarrow 0 \leq y \leq ux$$

Esempio: rappresentazione di costi fissi.

Se investo e produco, ho costi $c(y) = c_f + c_v y$, con $0 \leq y \leq Q$.

Se non investo e non produco, ho costi nulli $c(y) = 0$ e produzione nulla $y = 0$.

Rappresentiamo la scelta con una variabile binaria x .

$$x = \begin{cases} 1 & \text{investo e produco} \\ 0 & \text{non investo e non produco} \end{cases}$$

Ora il modello può essere espresso così:

$$\begin{aligned} c(y) &= c_f x + c_v y \\ 0 &\leq y \leq Qx \end{aligned}$$

Attivazione/disattivazione di vincoli

Le variabili binarie possono essere usate anche per attivare e disattivare vincoli.

$$y \leq Q + Mx$$

con M “abbastanza grande”, equivale a

$$\begin{cases} y \leq Q & \text{se } x = 0 \\ y \text{ qualsiasi} & \text{se } x = 1 \end{cases}$$

Esempio: vincoli disgiuntivi

Supponiamo di voler imporre il vincolo

$$|a - b| \geq k$$

essendo a e b due variabili continue non-negative e $k > 0$ dato. Scritto così, il vincolo non è lineare ed è un vincolo disgiuntivo.

$$|a - b| \geq k \Leftrightarrow (a - b \geq k) \vee (a - b \leq -k)$$

Si può linearizzare introducendo una variabile binaria x ed una costante M “abbastanza grande”:

$$\begin{cases} a - b \geq k - Mx \\ a - b \leq -k + M(1 - x) \end{cases}$$

A seconda del valore di x , uno dei due vincoli viene imposto mentre l'altro risulta disattivato.

Esempio: regioni non convesse

Supponiamo di voler imporre che un punto di coordinate (x, y) non sia interno ad un rettangolo con lati paralleli agli assi, base $2a$, altezza $2b$ e centro nell'origine.

La regione ammissibile definita da questo vincolo non è convessa. Il punto non è interno quando

$$(x \leq -a) \vee (x \geq a) \vee (y \leq -b) \vee (y \geq b).$$

Almeno una delle quattro condizioni deve essere vera. Introduciamo 4 variabili binarie w'_x , w''_x , w'_y e w''_y , che disattivano i vincoli quando valgono 1.

$$\left\{ \begin{array}{l} x \geq a - Mw'_x \\ x \leq -a + Mw''_x \\ y \geq b - Mw'_y \\ y \leq -b + Mw''_y \\ w'_x + w''_x + w'_y + w''_y \leq 3 \end{array} \right.$$

Esempio: problemi di scheduling

Nei problemi di scheduling bisogna decidere in che ordine eseguire n jobs di durata nota (*processing time*) $p_i \forall i = 1, \dots, n$ su una o più macchine.

Indicando con una variabile t_i l'istante di inizio di ogni job i , i vincoli di non-sovrapposizione tra jobs assegnati alla stessa macchina sono del tipo:

$$\begin{cases} t_j \geq t_i + p_i & \text{se } i \text{ precede } j \\ t_i \geq t_j + p_j & \text{se } j \text{ precede } i \end{cases}$$

Uno dei due vincoli deve essere imposto, mentre l'altro deve essere disattivato. Introducendo una variabile binaria x_{ij} per ogni coppia (non ordinata) $[i, j]$, si ha

$$\begin{cases} t_j - t_i \geq p_i - Mx_{ij} \\ t_i - t_j \geq p_j - M(1 - x_{ij}) \end{cases}$$

Occorrono però $n(n-1)/2$ variabili binarie.

Programmazione non-lineare

Giovanni Righini

Ricerca Operativa



UNIVERSITÀ DEGLI STUDI
DI MILANO

Programmazione non-lineare (PNL)

La **programmazione non-lineare**, o **PNL** (**Non-linear Programming**, **NLP**) studia problemi di ottimizzazione in cui la funzione obiettivo o alcuni vincoli sono non-lineari.

Applicazioni:

- economie di scala,
- minimizzazione dell'errore quadratico medio in problemi di
 - controllo ottimo,
 - classificazione automatica,
 - machine learning,
 - fitting di dati sperimentali,
- riformulazioni quadratiche,
- modelli di sistemi fisici non lineari,
- modelli che implicano l'uso di distanza Euclidea,
- eccetera...

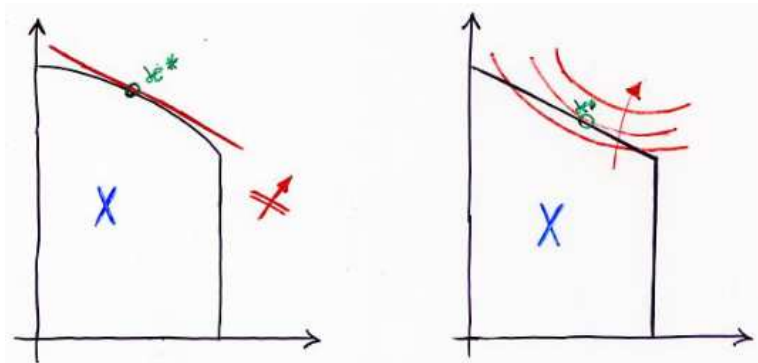
Programmazione non-lineare (PNL)

Forma generale:

$$\begin{aligned} \text{minimize } z &= f(x) \\ \text{s.t. } h_i(x) &= 0 & \forall i \\ g_j(x) &\leq 0 & \forall j \\ x &\in \mathbb{R}^n \end{aligned} \tag{1}$$

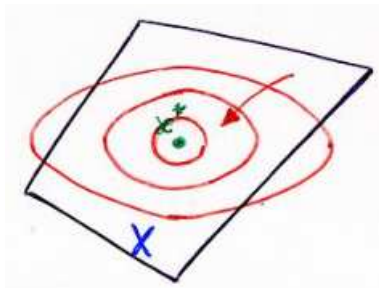
dove $f(x)$, $g(x)$ e $h(x)$ possono essere funzioni non-lineari.

Programmazione non-lineare (PNL)



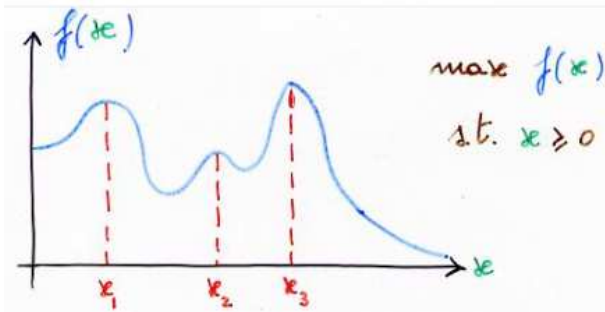
In generale, la soluzione ottima può non essere all'intersezione dei vincoli.

Programmazione non-lineare (PNL)



Non è neppure detto che sia necessariamente sulla frontiera della regione ammissibile.

Ottimalità locale e globale



Le soluzioni x_1 e x_2 sono **ottimi locali**.

La soluzione x_3 è un **ottimo globale**.

Ottimalità locale e globale

Ottimalità globale. Una soluzione $x^* \in X$ è un **minimo globale** se e solo se

$$f(x^*) \leq f(x) \quad \forall x \in X.$$

Ottimalità locale. Una soluzione $\bar{x} \in X$ è un **minimo locale** se e solo se

$$\exists \epsilon > 0 : f(\bar{x}) \leq f(x) \quad \forall x \in X : \|\bar{x} - x\| \leq \epsilon.$$

L'insieme delle soluzioni $x \in X : \|\bar{x} - x\| \leq \epsilon$ è un **intorno** di \bar{x} .

Ottimalità locale e globale

Per trovare un ottimo globale si dovrebbero enumerare tutti gli ottimi locali e scegliere il migliore.

Tuttavia, l'enumerazione completa degli ottimi locali in generale non è fattibile in pratica

- per il loro grande numero;
- perché non è noto un metodo algoritmico per eseguirla in modo efficiente.

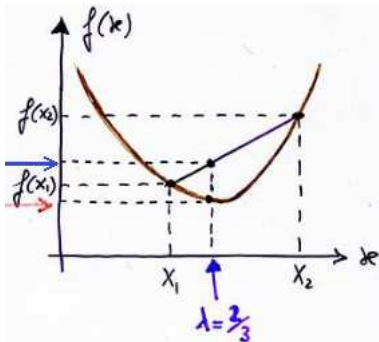
Un'importante eccezione positiva è la **programmazione convessa**. Un problema di minimizzazione non-lineare è convesso quando

- la funzione-obiettivo è una **funzione convessa**;
- la regione ammissibile è un **insieme convesso**.

Funzioni convesse

Una funzione $f(x)$ è convessa se e solo se per ogni coppia di punti x_1 e x_2 nel suo dominio e per $0 \leq \lambda \leq 1$

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$



Insiemi convessi

Un insieme X è convesso se e solo se per ogni coppia di punti x_1 e x_2 in esso, tutte le loro combinazioni convesse appartengono all'insieme:

$$\forall x_1, x_2 \in X \quad \forall 0 \leq \lambda \leq 1 \quad \lambda x_1 + (1 - \lambda)x_2 \in X.$$



Programmazione convessa

La regione ammissibile è convessa quando

- tutti i vincoli di uguaglianza $h(x) = 0$ sono lineari;
- tutti i vincoli di disuguaglianza, riscritti in forma $g(x) \leq 0$ sono convessi.

La funzione-obiettivo da **minimizzare** deve essere convessa (deve essere concava in caso di massimizzazione).

Se entrambe queste condizioni sono soddisfatte, il problema è di programmazione convessa e quindi:

- l'ottimalità locale implica quella globale;
- se esistono più ottimi, essi formano un insieme convesso.

Ottimizzazione vincolata e non vincolata

Distinguiamo tra

- Unconstrained NLP: minimizzare una funzione non lineare senza ulteriori vincoli.
- Constrained NLP: minimizzare $f(x)$, con $x \in X$: le non-linearità possono essere tanto nell'obiettivo quanto nei vincoli.

Ottimizzazione non vincolata

Assumiamo che la funzione-obiettivo $f(x)$ da minimizzare sia **continua** e **differenziabile**.

Il gradiente di una funzione $f(x_1, x_2, \dots, x_n)$ è il vettore delle sue derivate parziali di primo ordine

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n} \right]^T.$$

L'Hessiano di una funzione $f(x_1, x_2, \dots, x_n)$ è la matrice delle sue derivate parziali di secondo ordine

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}.$$

Caratterizzazione dei minimi locali

Condizioni necessarie del primo ordine.

$$\nabla f(\bar{x}) = 0$$

Condizioni necessarie del secondo ordine.

$$\nabla^2 f(\bar{x}) \geq 0$$

Condizioni sufficienti del secondo ordine.

$$\nabla^2 f(\bar{x}) > 0$$

Algoritmi

Se le derivate prime e seconde sono note (il che non è garantito, in generale), si possono enumerare i punti nei quali sono soddisfatte le condizioni analitiche.

Gli algoritmi per l'ottimizzazione non-lineare sono **algoritmi iterativi**, che **convergono verso** un minimo locale.

Partono da una soluzione data $x^{(0)}$ e calcolano una sequenza di soluzioni tali che il valore di $f(x)$ diminuisce monotonicamente.

Si fermano quando il miglioramento ottenuto o il passo compiuto sono più piccoli di una data soglia.

Ad ogni iterazione k , l'algoritmo calcola una direzione $d^{(k)}$ (vettore) e un passo s_k (scalare) tali che:

$$x^{(k+1)} = x^{(k)} + s_k d^{(k)}.$$

Le due principali strategie sono:

- line search;
- trust regions.

Algoritmi *line search*

Negli algoritmi *line search*, le scelte più comuni per definire la direzione $d^{(k)}$ sono:

- (metodo del gradiente): la direzione opposta a quella del gradiente, $-\nabla f(x^{(k)})$;
- (metodo di Newton): una direzione $-B^{-1}\nabla f(x^{(k)})$, dove B è una matrice semi-definita positiva;
- (metodo del gradiente coniugato): una direzione $-\nabla f(x^{(k)}) + \beta_k d^{(k-1)}$.

Metodo del gradiente

Per il teorema di Taylor

$$f(\mathbf{x}^{(k)} + s_k \mathbf{d}^{(k)}) = f(\mathbf{x}^{(k)}) + s_k \mathbf{d}^{(k)T} \nabla f(\mathbf{x}^{(k)}) + \frac{1}{2} s_k^2 \mathbf{d}^{(k)T} \nabla^2 f(\mathbf{x}^{(k)}) \mathbf{d}^{(k)} + \dots$$

Trascurando i termini dal secondo ordine in poi, si ha l'approssimazione

$$f(\mathbf{x}^{(k)} + s_k \mathbf{d}^{(k)}) \approx f(\mathbf{x}^{(k)}) + s_k \mathbf{d}^{(k)T} \nabla f(\mathbf{x}^{(k)})$$

che decresce più rapidamente nella direzione opposta a quella del *gradiente*.

$$\mathbf{d}^{(k)} = - \frac{\nabla f(\mathbf{x}^{(k)})}{\|\nabla f(\mathbf{x}^{(k)})\|}.$$

Un vantaggio di questo metodo, detto *steepest descent method* (o *gradient method*) è che richiede solo il calcolo del gradiente, non delle derivate seconde.

Metodo di Newton

Assumendo $s_k = 1$ e trascurando i termini dal terzo ordine in poi, si ha l'approssimazione

$$f(x^{(k)} + d^{(k)}) = f(x^{(k)}) + d^{(k)T} \nabla f(x^{(k)}) + \frac{1}{2} d^{(k)T} \nabla^2 f(x^{(k)}) d^{(k)}.$$

La direzione che minimizza questa quantità è la *direzione di Newton*:

$$d^{(k)} = -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}).$$

Il *metodo di Newton* è veloce e accurato, ma richiede il calcolo dell'Hessiano $\nabla^2 f(x^{(k)})$ e può essere usato solo quando $\nabla^2 f(x^{(k)})$ è definito positivo.

Metodi *quasi-Newton* sono stati ideati per ovviare a questo limite.

Scelta del passo

Una volta scelta la direzione $d^{(k)}$, rimane un problema di minimizzazione ad una sola variabile

$$\text{minimize } f(x^{(k+1)}) = f(x^{(k)} + s_k d^{(k)})$$

dove la variabile è lo scalare $s_k \geq 0$.

L'ottimizzazione esatta di s_k non è indispensabile; una buona approssimazione è sufficiente per avviare l'iterazione successiva dopo aver migliorato $f(x)$.

Gli algoritmi per determinare il passo possono essere classificati in

- algoritmi che richiedono il calcolo della derivata,
- algoritmi *derivative-free*.

Metodo di bisezione

Richiede il calcolo della derivata.

Dato un intervallo iniziale $r = [a, b]$ per s_k :

1. calcolare $\nabla f(x^{(k)} + \frac{a+b}{2}d^{(k)})$;
2. se è positiva, porre $r := [a, \frac{a+b}{2}]$;
3. se è negativa, porre $r := [\frac{a+b}{2}, b]$;
4. ripetere finché r è abbastanza piccolo.

I numeri di Fibonacci

La sequenza dei numeri di Fibonacci inizia con $F_0 = 0$ e $F_1 = 1$ e si ricava applicando la ricorsione

$$F_k = F_{k-1} + F_{k-2}.$$

Si ottiene così:

k	0	1	2	3	4	5	6	7	8	9	10	11	...
F_k	0	1	1	2	3	5	8	13	21	34	55	89	...

Tabella: I primi numeri di Fibonacci.

Una proprietà

Proprietà. Dati 4 numeri di Fibonacci consecutivi a partire dal k -esimo, si ha

$$F_{k+1}F_{k+2} - F_kF_{k+3} = (-1)^k \quad \forall k \geq 0.$$

Esempio.

$$F_6F_7 - F_5F_8 = 8 \times 13 - 5 \times 21 = 104 - 105 = -1 \quad (k = 5, \text{ dispari})$$

$$F_7F_8 - F_6F_9 = 13 \times 21 - 8 \times 34 = 273 - 272 = 1 \quad (k = 6, \text{ pari}).$$

Dimostrazione (per induzione).

$$F_{k+1}F_{k+2} - F_kF_{k+3} = (-1)^k \quad \forall k \geq 0.$$

- **Base dell'induzione (per $k = 0$):**

$$F_1F_2 - F_0F_3 = 1 \times 1 - 0 \times 2 = 1^0$$

- **Passo induttivo (da $k - 1$ a k per ogni $k \geq 1$):**

$$F_kF_{k+1} - F_{k-1}F_{k+2} = (-1)^{k-1} \Rightarrow F_{k+1}F_{k+2} - F_kF_{k+3} = (-1)^k.$$

Infatti:

$$\begin{aligned} F_{k+1}F_{k+2} - F_kF_{k+3} &= [F_{k+1}(F_k + F_{k+1})] - [F_k(2F_{k+1} + F_k)] = \\ &= F_kF_{k+1} + F_{k+1}^2 - 2F_kF_{k+1} - F_k^2 = (F_{k+1}^2 - F_k^2) - F_kF_{k+1} = \\ &= (F_{k+1} + F_k)(F_{k+1} - F_k) - F_kF_{k+1} = F_{k+2}F_{k-1} - F_kF_{k+1} = \text{(per ipotesi)} \\ &= -(-1)^{k-1} = (-1)^k. \quad [\text{c.v.d.}] \end{aligned}$$

Il problema

- È data una funzione continua $f(x)$ di una sola variabile x .
- Si vuole cercare un punto di minimo di $f(x)$.
- È dato un intervallo di incertezza iniziale I^0 .
- È richiesto un massimo intervallo di incertezza finale Δ .
- Si assume che la funzione sia *unimodale* nell'intervallo I^0 , cioè abbia un solo punto di minimo nell'intervallo.
- Si suppone di non poter/voler calcolare la derivata prima di $f(x)$, come invece è richiesto dal metodo di bisezione.
- Si suppone che sia possibile valutare $f(x)$ in punti diversi, purché distanti tra loro almeno ϵ (risoluzione).

Osservazione. Sarebbe ancora possibile usare il metodo di bisezione, valutando in ogni intervallo due punti distanti ϵ tra loro, posti al centro dell'intervallo stesso. L'informazione che se ne ricaverebbe sarebbe equivalente a quella fornita dal calcolo della derivata prima al centro dell'intervallo. In tal modo sarebbero necessarie *due* valutazioni della funzione ad ogni iterazione. Con il metodo dei numeri di Fibonacci, invece, basta *una* valutazione della funzione ad ogni iterazione.

Iterazione generica

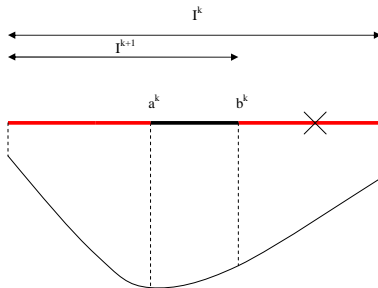
Alla generica iterazione k si ha un intervallo di incertezza I^k .

Si considerino due punti a^k e b^k interni all'intervallo, che lo dividono in tre parti.

Si conosca il valore della funzione $f(x)$ nei due punti interni.

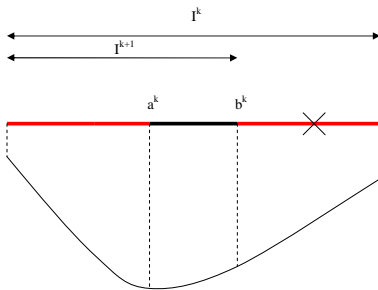
Se $f(a^k) > f(b^k)$, allora il minimo di $f(x)$ non cade nella prima parte.

Se $f(a^k) < f(b^k)$, allora il minimo di $f(x)$ non cade nella terza parte.



Punti di valutazione

Alla successiva iterazione l'intervallo di incertezza risulta composto da due delle tre parti dell'intervallo di incertezza precedente.



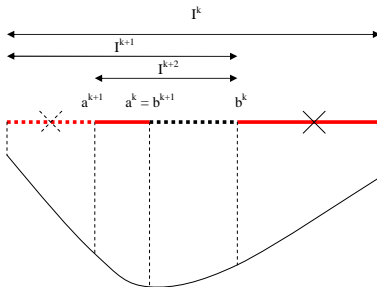
Uno dei due punti interni precedenti diventa un estremo dell'intervallo di incertezza.

Punti di valutazione

Per simmetria, ad ogni iterazione i punti in cui valutare $f(x)$ sono scelti in modo simmetrico nell'intervallo di incertezza corrente.

Si ha quindi:

$$I^k = I^{k+1} + I^{k+2} \quad \forall k \geq 0.$$

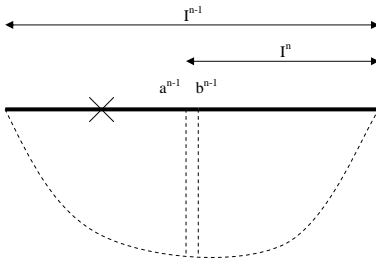


In uno dei due punti interni la funzione è già stata valutata in precedenza.

Intervallo finale

Osservazione. Più è grande l'intervallo “scartato” all'iterazione k e più risultano piccoli quelli “scartabili” all'iterazione $k + 1$.

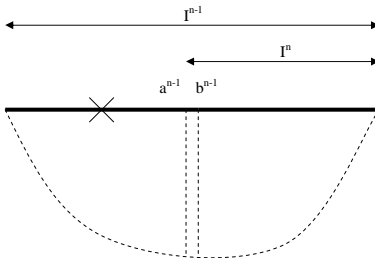
L'iterazione di massima efficacia è quella che consente di scartare metà dell'intervallo di incertezza, valutando due punti interni distinti vicinissimi tra loro (a distanza ϵ).



Tuttavia all'iterazione successiva il primo e il terzo intervallo sarebbero larghi solo ϵ e quindi si avrebbe un'iterazione di minima efficacia.

Intervallo finale

Si vuole quindi arrivare a compiere un'iterazione di massima efficacia *per ultima*.



Si ha perciò:

$$I^{n-1} = 2I^n - \epsilon.$$

Progressione delle iterazioni

Dalle due relazioni

$$l^k = l^{k+1} + l^{k+2} \quad \forall k \geq 0$$

$$l^{n-1} = 2l^n - \epsilon$$

si ricava:

$$l^{n-2} = l^{n-1} + l^n = 3l^n - \epsilon$$

$$l^{n-3} = l^{n-2} + l^{n-1} = 5l^n - 2\epsilon$$

$$l^{n-4} = l^{n-3} + l^{n-2} = 8l^n - 3\epsilon$$

...

$$l^{n-k} = l^{n-k+1} + l^{n-k+2} = F_{k+2}l^n - F_k\epsilon \quad \forall k \geq 1$$

...

$$l^2 = l^3 + l^4 = F_n l^n - F_{n-2}\epsilon$$

$$l^1 = l^2 + l^3 = F_{n+1} l^n - F_{n-1}\epsilon$$

$$l^0 = l^1 + l^2 = F_{n+2} l^n - F_n\epsilon.$$

Perciò:

$$l^0 = F_{n+2} l^n - F_n\epsilon \quad \text{ovvero} \quad l^n = \frac{l^0}{F_{n+2}} + \frac{F_n}{F_{n+2}}\epsilon.$$

Numero di iterazioni

Dalla relazione

$$I^n = \frac{I^0}{F_{n+2}} + \frac{F_n}{F_{n+2}} \epsilon$$

e dal requisito sull'incertezza finale

$$I^n \leq \Delta$$

si ricava il numero di iterazioni necessarie:

$$\bar{n} = \min\{n \mid \frac{I^0}{F_{n+2}} + \frac{F_n}{F_{n+2}} \epsilon \leq \Delta\}.$$

Scelta del primo punto interno

Dalle relazioni

$$l^{\bar{n}} = \frac{l^0}{F_{\bar{n}+2}} + \frac{F_{\bar{n}}}{F_{\bar{n}+2}} \epsilon$$

e

$$l^1 = F_{\bar{n}+1} l^{\bar{n}} - F_{\bar{n}-1} \epsilon$$

si ricava l^1 in funzione di l^0 :

$$\begin{aligned} l^1 &= F_{\bar{n}+1} \left(\frac{l^0}{F_{\bar{n}+2}} + \frac{F_{\bar{n}}}{F_{\bar{n}+2}} \epsilon \right) - F_{\bar{n}-1} \epsilon = \\ &= \frac{F_{\bar{n}+1}}{F_{\bar{n}+2}} l^0 + \frac{\epsilon}{F_{\bar{n}+2}} (F_{\bar{n}+1} F_{\bar{n}} - F_{\bar{n}+2} F_{\bar{n}-1}) = \\ &= \frac{F_{\bar{n}+1}}{F_{\bar{n}+2}} l^0 + \frac{\epsilon}{F_{\bar{n}+2}} (-1)^{\bar{n}-1}. \end{aligned}$$

Esempio

- È dato un intervallo di incertezza iniziale $I^0 = [0, 100]$.
- È richiesto un massimo intervallo di incertezza finale $\Delta = 2$.
- È data una risoluzione $\epsilon = 1$.

Esempio

Scelta del numero di iterazioni

$$\bar{n} = \min\{n \mid \frac{100}{F_{n+2}} + \frac{F_n}{F_{n+2}} \leq 2\} = 9.$$

Infatti si ha:

$$\text{per } n = 8: \frac{100}{F_{10}} + \frac{F_8}{F_{10}} = 100/55 + 21/55 = 121/55 > 2,$$

$$\text{per } n = 9: \frac{100}{F_{11}} + \frac{F_9}{F_{11}} = 100/89 + 34/89 = 134/89 < 2.$$

k	0	1	2	3	4	5	6	7	8	9	10	11	...
F_k	0	1	1	2	3	5	8	13	21	34	55	89	...

Scelta del primo punto interno

$$\begin{aligned} I^1 &= \frac{F_{\bar{n}+1}}{F_{\bar{n}+2}} I^0 + \frac{\epsilon}{F_{\bar{n}+2}} (-1)^{\bar{n}-1} = \\ &= \frac{55}{89} 100 + \frac{1}{89} (-1)^8 = 5500/89 + 1/89 = 5501/89 \approx 61,81. \end{aligned}$$

Esempio

Iterazioni. Gli intervalli di incertezza successivi risultano avere le seguenti ampiezze:

$$I^2 = I^0 - I^1 = \frac{8900 - 5501}{89} = \frac{3399}{89}$$

$$I^3 = I^1 - I^2 = \frac{5501 - 3399}{89} = \frac{2102}{89}$$

$$I^4 = I^2 - I^3 = \frac{3399 - 2102}{89} = \frac{1297}{89}$$

$$I^5 = I^3 - I^4 = \frac{2102 - 1297}{89} = \frac{805}{89}$$

$$I^6 = I^4 - I^5 = \frac{1297 - 805}{89} = \frac{492}{89}$$

$$I^7 = I^5 - I^6 = \frac{805 - 492}{89} = \frac{313}{89}$$

$$I^8 = I^6 - I^7 = \frac{492 - 313}{89} = \frac{179}{89}$$

$$I^9 = I^7 - I^8 = \frac{313 - 179}{89} = \frac{134}{89}.$$

Conclusioni

- Il metodo dei numeri di Fibonacci consente di approssimare il minimo di una funzione di una sola variabile continua.
- Deve essere noto un intervallo di incertezza iniziale e la funzione deve essere unimodale in esso.
- Il metodo non richiede il calcolo della derivata prima della funzione.
- Il metodo richiede di valutare la funzione in un numero di punti dello stesso ordine di grandezza del numero di iterazioni.

Ottimizzazione vincolata

Nell'ottimizzazione non-lineare vincolata consideriamo anche l'effetto di

- vincoli di uguaglianza $h_i(\mathbf{x}) = 0 \quad \forall i \in \mathcal{E}$
- vincoli di disuguaglianza $g_j(\mathbf{x}) \leq 0 \quad \forall j \in \mathcal{I}$.

Un vincolo di disuguaglianza $j \in \mathcal{I}$ è attivo in una soluzione $\bar{\mathbf{x}}$ se e solo se $g_j(\bar{\mathbf{x}}) = 0$.

Vincoli di uguaglianza

Consideriamo un vincolo di uguaglianza $h(x) = 0$ ed un punto \bar{x} su di esso.

Indichiamo con $\nabla h(\bar{x})$ la direzione della normale al vincolo in \bar{x} .

Consideriamo un passo infinitesimo da \bar{x} lungo una direzione d .

Per mantenere l'ammissibilità rispetto al vincolo, d deve essere tale che:

$$\nabla h(\bar{x})^T d = 0.$$

Il passo produce un miglioramento nel valore di $f(x)$ se e solo se

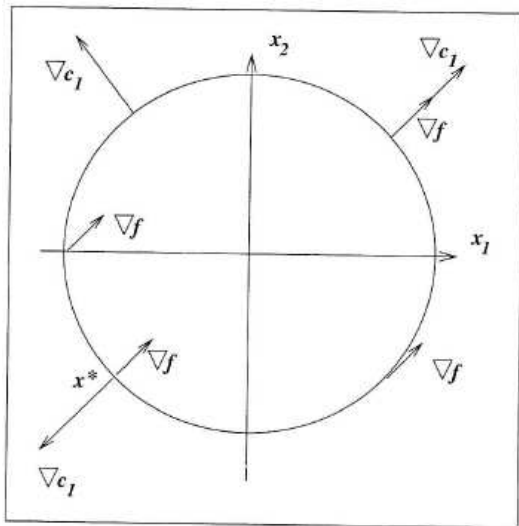
$$\nabla f(\bar{x})^T d < 0.$$

Quindi un passo migliorante *non* è possibile se

$$\nabla h(\bar{x}) = \lambda \nabla f(\bar{x})$$

per qualche $\lambda \neq 0$.

Vincoli di uguaglianza



Vincoli di disuguaglianza

Consideriamo due vincoli di disuguaglianza $g_1(x) \geq 0$, $g_2(x) \geq 0$ ed un punto \bar{x} , dove entrambi sono attivi.

Indichiamo con $\nabla g_1(\bar{x})$ e $\nabla g_2(\bar{x})$ la direzione della normale ai vincoli in \bar{x} .

Poiché i vincoli sono in forma di \geq , il gradiente punta verso l'interno della regione ammissibile.

Consideriamo un passo infinitesimo da \bar{x} lungo una direzione d .

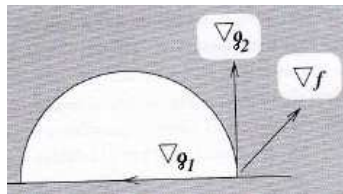
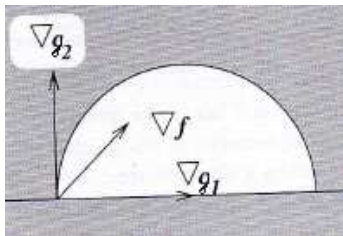
Per mantenere l'ammissibilità rispetto ai vincoli, d deve essere tale che:

$$\nabla g_1(\bar{x})^T d \geq 0 \quad \text{e} \quad \nabla g_2(\bar{x})^T d \geq 0.$$

Il passo produce un miglioramento di $f(x)$ se e solo se

$$\nabla f(\bar{x})^T d < 0.$$

Vincoli di disuguaglianza



Direzioni ammissibili

Una direzione d è ammissibile in \bar{x} se e solo se:

$$\nabla h_i(\bar{x})^T d = 0 \quad \forall i \in \mathcal{E} \quad \text{e} \quad \nabla g_i(\bar{x})^T d \geq 0 \quad \forall i \in \mathcal{A}(\bar{x}),$$

dove $\mathcal{A}(\bar{x})$ indica l'insieme dei vincoli di disuguaglianza attivi in \bar{x} .

Dobbiamo considerare solo i gradienti $\nabla g_i(\bar{x})$ linearmente indipendenti.

Per definire le direzioni in modo univoco, normalizziamo d in modo che abbia norma unitaria.

Un algoritmo delle direzioni ammissibili è un algoritmo iterativo che seleziona una direzione ammissibile ad ogni iterazione e poi calcola un passo ottimale lungo di essa risolvendo un problema non-lineare a singola variabile *vincolato*.

Programmazione non-lineare

Giovanni Righini

Ricerca Operativa



UNIVERSITÀ DEGLI STUDI
DI MILANO

Programmazione non-lineare (PNL)

La **programmazione non-lineare**, o **PNL** (**Non-linear Programming**, **NLP**) studia problemi di ottimizzazione in cui la funzione obiettivo o alcuni vincoli sono non-lineari.

Applicazioni:

- economie di scala,
- minimizzazione dell'errore quadratico medio in problemi di
 - controllo ottimo,
 - classificazione automatica,
 - machine learning,
 - fitting di dati sperimentali,
- riformulazioni quadratiche,
- modelli di sistemi fisici non lineari,
- modelli che implicano l'uso di distanza Euclidea,
- eccetera...

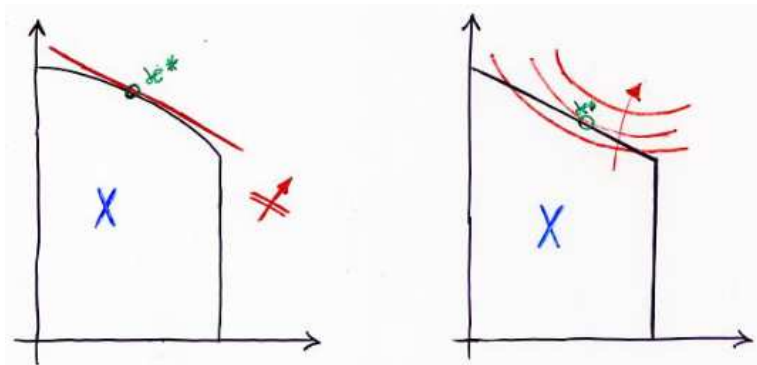
Programmazione non-lineare (PNL)

Forma generale:

$$\begin{aligned} \text{minimize } z &= f(x) \\ \text{s.t. } h_i(x) &= 0 & \forall i \\ g_j(x) &\leq 0 & \forall j \\ x &\in \mathbb{R}^n \end{aligned} \tag{1}$$

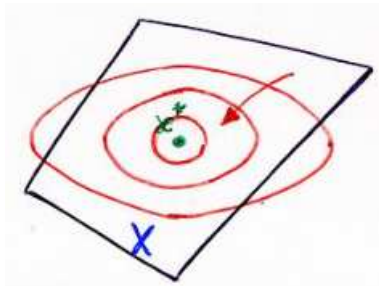
dove $f(x)$, $g(x)$ e $h(x)$ possono essere funzioni non-lineari.

Programmazione non-lineare (PNL)



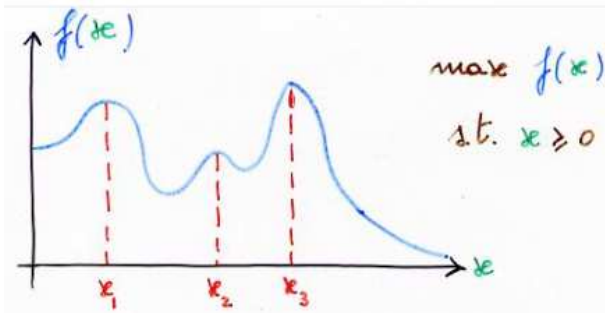
In generale, la soluzione ottima può non essere all'intersezione dei vincoli.

Programmazione non-lineare (PNL)



Non è neppure detto che sia necessariamente sulla frontiera della regione ammissibile.

Ottimalità locale e globale



Le soluzioni x_1 e x_2 sono **ottimi locali**.

La soluzione x_3 è un **ottimo globale**.

Ottimalità locale e globale

Ottimalità globale. Una soluzione $x^* \in X$ è un **minimo globale** se e solo se

$$f(x^*) \leq f(x) \quad \forall x \in X.$$

Ottimalità locale. Una soluzione $\bar{x} \in X$ è un **minimo locale** se e solo se

$$\exists \epsilon > 0 : f(\bar{x}) \leq f(x) \quad \forall x \in X : \|\bar{x} - x\| \leq \epsilon.$$

L'insieme delle soluzioni $x \in X : \|\bar{x} - x\| \leq \epsilon$ è un **intorno** di \bar{x} .

Ottimalità locale e globale

Per trovare un ottimo globale si dovrebbero enumerare tutti gli ottimi locali e scegliere il migliore.

Tuttavia, l'enumerazione completa degli ottimi locali in generale non è fattibile in pratica

- per il loro grande numero;
- perché non è noto un metodo algoritmico per eseguirla in modo efficiente.

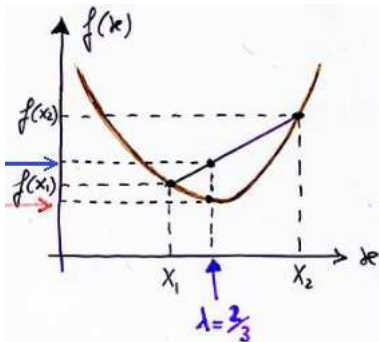
Un'importante eccezione positiva è la **programmazione convessa**. Un problema di minimizzazione non-lineare è convesso quando

- la funzione-obiettivo è una **funzione convessa**;
- la regione ammissibile è un **insieme convesso**.

Funzioni convesse

Una funzione $f(x)$ è convessa se e solo se per ogni coppia di punti x_1 e x_2 nel suo dominio e per $0 \leq \lambda \leq 1$

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2).$$



Insiemi convessi

Un insieme X è convesso se e solo se per ogni coppia di punti x_1 e x_2 in esso, tutte le loro combinazioni convesse appartengono all'insieme:

$$\forall x_1, x_2 \in X \quad \forall 0 \leq \lambda \leq 1 \quad \lambda x_1 + (1 - \lambda)x_2 \in X.$$



Programmazione convessa

La regione ammissibile è convessa quando

- tutti i vincoli di uguaglianza $h(x) = 0$ sono lineari;
- tutti i vincoli di disuguaglianza, riscritti in forma $g(x) \leq 0$ sono convessi.

La funzione-obiettivo da **minimizzare** deve essere convessa (deve essere concava in caso di massimizzazione).

Se entrambe queste condizioni sono soddisfatte, il problema è di programmazione convessa e quindi:

- l'ottimalità locale implica quella globale;
- se esistono più ottimi, essi formano un insieme convesso.

Ottimizzazione vincolata e non vincolata

Distinguiamo tra

- Unconstrained NLP: minimizzare una funzione non lineare senza ulteriori vincoli.
- Constrained NLP: minimizzare $f(x)$, con $x \in X$: le non-linearità possono essere tanto nell'obiettivo quanto nei vincoli.

Ottimizzazione non vincolata

Assumiamo che la funzione-obiettivo $f(x)$ da minimizzare sia **continua** e **differenziabile**.

Il gradiente di una funzione $f(x_1, x_2, \dots, x_n)$ è il vettore delle sue derivate parziali di primo ordine

$$\nabla f(x) = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n} \right]^T.$$

L'Hessiano di una funzione $f(x_1, x_2, \dots, x_n)$ è la matrice delle sue derivate parziali di secondo ordine

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}.$$

Caratterizzazione dei minimi locali

Condizioni necessarie del primo ordine.

$$\nabla f(\bar{x}) = 0$$

Condizioni necessarie del secondo ordine.

$$\nabla^2 f(\bar{x}) \geq 0$$

Condizioni sufficienti del secondo ordine.

$$\nabla^2 f(\bar{x}) > 0$$

Algoritmi

Se le derivate prime e seconde sono note (il che non è garantito, in generale), si possono enumerare i punti nei quali sono soddisfatte le condizioni analitiche.

Gli algoritmi per l'ottimizzazione non-lineare sono **algoritmi iterativi**, che **convergono verso** un minimo locale.

Partono da una soluzione data $x^{(0)}$ e calcolano una sequenza di soluzioni tali che il valore di $f(x)$ diminuisce monotonicamente.

Si fermano quando il miglioramento ottenuto o il passo compiuto sono più piccoli di una data soglia.

Ad ogni iterazione k , l'algoritmo calcola una direzione $d^{(k)}$ (vettore) e un passo s_k (scalare) tali che:

$$x^{(k+1)} = x^{(k)} + s_k d^{(k)}.$$

Velocità di convergenza

Sia $\{x_k\}$ una sequenza in \mathbb{R}^n che converge a x^* .

Convergenza lineare:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = r < 1.$$

Convergenza superlineare:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Convergenza quadratica:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} = M.$$

Algoritmi *line search*

Le due principali strategie sono:

- line search;
- trust regions.

Negli algoritmi *line search*, le scelte più comuni per definire la direzione $d^{(k)}$ sono:

- (metodo del gradiente): la direzione opposta a quella del gradiente, $-\nabla f(x^{(k)})$;
- (metodo di Newton): una direzione $-B^{-1}\nabla f(x^{(k)})$, dove B è una matrice semi-definita positiva;
- (metodo del gradiente coniugato): una direzione $-\nabla f(x^{(k)}) + \beta_k d^{(k-1)}$.

Metodo del gradiente

Per il teorema di Taylor

$$f(\mathbf{x}^{(k)} + s_k \mathbf{d}^{(k)}) = f(\mathbf{x}^{(k)}) + s_k \mathbf{d}^{(k)T} \nabla f(\mathbf{x}^{(k)}) + \frac{1}{2} s_k^2 \mathbf{d}^{(k)T} \nabla^2 f(\mathbf{x}^{(k)}) \mathbf{d}^{(k)} + \dots$$

Trascurando i termini dal secondo ordine in poi, si ha l'approssimazione

$$f(\mathbf{x}^{(k)} + s_k \mathbf{d}^{(k)}) \approx f(\mathbf{x}^{(k)}) + s_k \mathbf{d}^{(k)T} \nabla f(\mathbf{x}^{(k)})$$

che decresce più rapidamente nella direzione opposta a quella del *gradiente*.

$$\mathbf{d}^{(k)} = - \frac{\nabla f(\mathbf{x}^{(k)})}{\|\nabla f(\mathbf{x}^{(k)})\|}.$$

Un vantaggio di questo metodo, detto *steepest descent method* (o *gradient method*) è che richiede solo il calcolo del gradiente, non delle derivate seconde.

Metodo di Newton

Assumendo $s_k = 1$ e trascurando i termini dal terzo ordine in poi, si ha l'approssimazione

$$f(\mathbf{x}^{(k)} + \mathbf{d}^{(k)}) = f(\mathbf{x}^{(k)}) + \mathbf{d}^{(k)T} \nabla f(\mathbf{x}^{(k)}) + \frac{1}{2} \mathbf{d}^{(k)T} \nabla^2 f(\mathbf{x}^{(k)}) \mathbf{d}^{(k)}.$$

La direzione che minimizza questa quantità è la *direzione di Newton*:

$$\mathbf{d}^{(k)} = -\nabla^2 f(\mathbf{x}^{(k)})^{-1} \nabla f(\mathbf{x}^{(k)}).$$

Il *metodo di Newton* è veloce e accurato, ma richiede il calcolo dell'Hessiano $\nabla^2 f(\mathbf{x}^{(k)})$ e può essere usato solo quando $\nabla^2 f(\mathbf{x}^{(k)})$ è definito positivo.

Metodi *quasi-Newton*, basati sull'approssimazione dell'Hessiano, sono stati ideati per ovviare a questo limite.

Metodi trust region

I metodi *trust region* richiedono di

- approssimare la funzione $f()$ con un modello $m_k()$, che viene aggiornato ad ogni iterazione k ;
- cercare un minimo di $m_k()$ in un intorno di raggio p_k della soluzione corrente x_k .

Se la diminuzione del valore dell'obiettivo non è “grande abbastanza”, il raggio viene diminuito e l'ottimizzazione viene ripetuta.

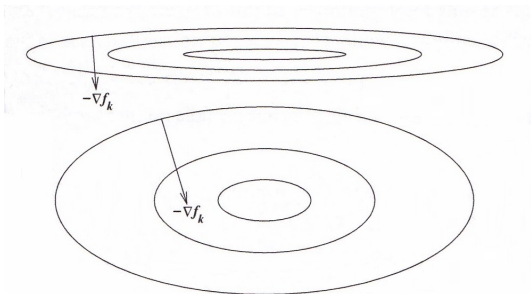
Di solito il modello m è quadratico

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T B_k p$$

e usa il gradiente ∇f_k e l'Hessiano o una sua approssimazione B_k .

Scaling

Gli algoritmi di programmazione non-lineare possono essere più o meno robusti rispetto allo **scaling**, che si ha, ad esempio, cambiando l'unità di misura di alcune grandezze.



L'**invarianza di scala** è una proprietà desiderabile degli algoritmi, che li rende più **robusti**.

Programmazione non-lineare: caso monodimensionale

Giovanni Righini

Ricerca Operativa



UNIVERSITÀ DEGLI STUDI
DI MILANO

Scelta del passo

Una volta scelta la direzione $d^{(k)}$ nei metodi line search rimane un problema di minimizzazione ad una sola variabile

$$\text{minimize } f(x^{(k+1)}) = f(x^{(k)} + s_k d^{(k)})$$

dove la variabile è lo scalare $s_k \geq 0$.

L'ottimizzazione esatta di s_k non è indispensabile; una buona approssimazione è sufficiente per avviare l'iterazione successiva dopo aver migliorato $f(x)$.

Gli algoritmi per determinare il passo possono essere classificati in

- algoritmi che richiedono il calcolo della derivata,
- algoritmi *derivative-free*.

Metodo di bisezione

Richiede il calcolo della derivata.

Dato un intervallo iniziale $r = [a, b]$ per s_k :

1. calcolare $\nabla f(x^{(k)} + \frac{a+b}{2}d^{(k)})$;
2. se è positiva, porre $r := [a, \frac{a+b}{2}]$;
3. se è negativa, porre $r := [\frac{a+b}{2}, b]$;
4. ripetere finché r è abbastanza piccolo.

I numeri di Fibonacci

La sequenza dei numeri di Fibonacci inizia con $F_0 = 0$ e $F_1 = 1$ e si ricava applicando la ricorsione

$$F_k = F_{k-1} + F_{k-2}.$$

Si ottiene così:

k	0	1	2	3	4	5	6	7	8	9	10	11	...
F_k	0	1	1	2	3	5	8	13	21	34	55	89	...

Tabella: I primi numeri di Fibonacci.

Una proprietà

Proprietà. Dati 4 numeri di Fibonacci consecutivi a partire dal k -esimo, si ha

$$F_{k+1}F_{k+2} - F_kF_{k+3} = (-1)^k \quad \forall k \geq 0.$$

Esempio.

$$F_6F_7 - F_5F_8 = 8 \times 13 - 5 \times 21 = 104 - 105 = -1 \quad (k = 5, \text{dispari})$$

$$F_7F_8 - F_6F_9 = 13 \times 21 - 8 \times 34 = 273 - 272 = 1 \quad (k = 6, \text{pari}).$$

Dimostrazione (per induzione).

$$F_{k+1}F_{k+2} - F_kF_{k+3} = (-1)^k \quad \forall k \geq 0.$$

- Base dell'induzione (per $k = 0$):

$$F_1F_2 - F_0F_3 = 1 \times 1 - 0 \times 2 = 1^0$$

- Passo induttivo (da $k - 1$ a k per ogni $k \geq 1$):

$$F_kF_{k+1} - F_{k-1}F_{k+2} = (-1)^{k-1} \Rightarrow F_{k+1}F_{k+2} - F_kF_{k+3} = (-1)^k.$$

Infatti:

$$\begin{aligned} F_{k+1}F_{k+2} - F_kF_{k+3} &= [F_{k+1}(F_k + F_{k+1})] - [F_k(2F_{k+1} + F_k)] = \\ &= F_kF_{k+1} + F_{k+1}^2 - 2F_kF_{k+1} - F_k^2 = (F_{k+1}^2 - F_k^2) - F_kF_{k+1} = \\ &= (F_{k+1} + F_k)(F_{k+1} - F_k) - F_kF_{k+1} = F_{k+2}F_{k-1} - F_kF_{k+1} = \text{(per ipotesi)} \\ &= -(-1)^{k-1} = (-1)^k. \quad [\text{c.v.d.}] \end{aligned}$$

Il problema

- È data una funzione continua $f(x)$ di una sola variabile x .
- Si vuole cercare un punto di minimo di $f(x)$.
- È dato un intervallo di incertezza iniziale I^0 .
- È richiesto un massimo intervallo di incertezza finale Δ .
- Si assume che la funzione sia *unimodale* nell'intervallo I^0 , cioè abbia un solo punto di minimo nell'intervallo.
- Si suppone di non poter/voler calcolare la derivata prima di $f(x)$, come invece è richiesto dal metodo di bisezione.
- Si suppone che sia possibile valutare $f(x)$ in punti diversi, purché distanti tra loro almeno ϵ (risoluzione).

Osservazione. Sarebbe ancora possibile usare il metodo di bisezione, valutando in ogni intervallo due punti distanti ϵ tra loro, posti al centro dell'intervallo stesso. L'informazione che se ne ricaverebbe sarebbe equivalente a quella fornita dal calcolo della derivata prima al centro dell'intervallo. In tal modo sarebbero necessarie *due* valutazioni della funzione ad ogni iterazione. Con il metodo dei numeri di Fibonacci, invece, basta *una* valutazione della funzione ad ogni iterazione.

Iterazione generica

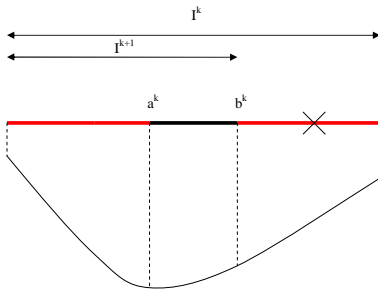
Alla generica iterazione k si ha un intervallo di incertezza I^k .

Si considerino due punti a^k e b^k interni all'intervallo, che lo dividono in tre parti.

Si conosca il valore della funzione $f(x)$ nei due punti interni.

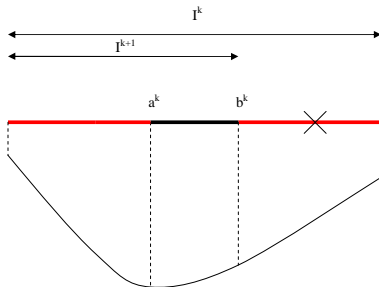
Se $f(a^k) > f(b^k)$, allora il minimo di $f(x)$ non cade nella prima parte.

Se $f(a^k) < f(b^k)$, allora il minimo di $f(x)$ non cade nella terza parte.



Punti di valutazione

Alla successiva iterazione l'intervallo di incertezza risulta composto da due delle tre parti dell'intervallo di incertezza precedente.



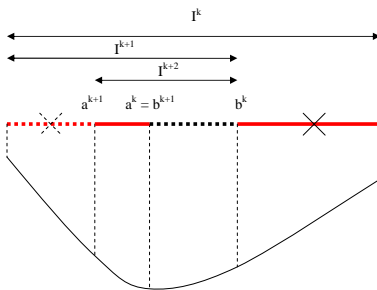
Uno dei due punti interni precedenti diventa un estremo dell'intervallo di incertezza.

Punti di valutazione

Per simmetria, ad ogni iterazione i punti in cui valutare $f(x)$ sono scelti in modo simmetrico nell'intervallo di incertezza corrente.

Si ha quindi:

$$I^k = I^{k+1} + I^{k+2} \quad \forall k \geq 0.$$

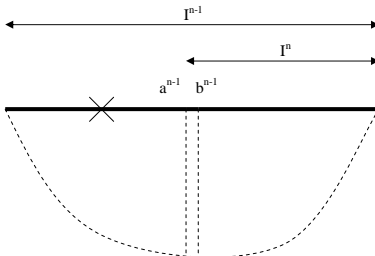


In uno dei due punti interni la funzione è già stata valutata in precedenza.

Intervallo finale

Osservazione. Più è grande l'intervallo “scartato” all'iterazione k e più risultano piccoli quelli “scartabili” all'iterazione $k + 1$.

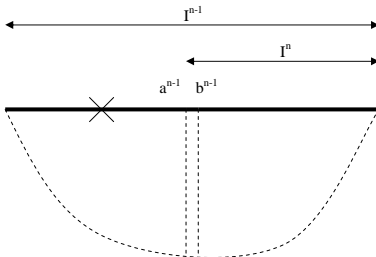
L'iterazione di massima efficacia è quella che consente di scartare metà dell'intervallo di incertezza, valutando due punti interni distinti vicinissimi tra loro (a distanza ϵ).



Tuttavia all'iterazione successiva il primo e il terzo intervallo sarebbero larghi solo ϵ e quindi si avrebbe un'iterazione di minima efficacia.

Intervallo finale

Si vuole quindi arrivare a compiere un'iterazione di massima efficacia *per ultima*.



Si ha perciò:

$$I^{n-1} = 2I^n - \epsilon.$$

Progressione delle iterazioni

Dalle due relazioni

$$l^k = l^{k+1} + l^{k+2} \quad \forall k \geq 0$$

$$l^{n-1} = 2l^n - \epsilon$$

si ricava:

$$l^{n-2} = l^{n-1} + l^n = 3l^n - \epsilon$$

$$l^{n-3} = l^{n-2} + l^{n-1} = 5l^n - 2\epsilon$$

$$l^{n-4} = l^{n-3} + l^{n-2} = 8l^n - 3\epsilon$$

...

$$l^{n-k} = l^{n-k+1} + l^{n-k+2} = F_{k+2}l^n - F_k\epsilon \quad \forall k \geq 1$$

...

$$l^2 = l^3 + l^4 = F_n l^n - F_{n-2}\epsilon$$

$$l^1 = l^2 + l^3 = F_{n+1} l^n - F_{n-1}\epsilon$$

$$l^0 = l^1 + l^2 = F_{n+2} l^n - F_n \epsilon.$$

Perciò:

$$l^0 = F_{n+2} l^n - F_n \epsilon \quad \text{ovvero} \quad l^n = \frac{l^0}{F_{n+2}} + \frac{F_n}{F_{n+2}} \epsilon.$$

Numero di iterazioni

Dalla relazione

$$I^n = \frac{I^0}{F_{n+2}} + \frac{F_n}{F_{n+2}}\epsilon$$

e dal requisito sull'incertezza finale

$$I^n \leq \Delta$$

si ricava il numero di iterazioni necessarie:

$$\bar{n} = \min\{n \mid \frac{I^0}{F_{n+2}} + \frac{F_n}{F_{n+2}}\epsilon \leq \Delta\}.$$

Formula di Binet:

$$F_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right)$$

Scelta del primo punto interno

Dalle relazioni

$$I^{\bar{n}} = \frac{I^0}{F_{\bar{n}+2}} + \frac{F_{\bar{n}}}{F_{\bar{n}+2}}\epsilon$$

e

$$I^1 = F_{\bar{n}+1}I^{\bar{n}} - F_{\bar{n}-1}\epsilon$$

si ricava I^1 in funzione di I^0 :

$$\begin{aligned} I^1 &= F_{\bar{n}+1} \left(\frac{I^0}{F_{\bar{n}+2}} + \frac{F_{\bar{n}}}{F_{\bar{n}+2}}\epsilon \right) - F_{\bar{n}-1}\epsilon = \\ &= \frac{F_{\bar{n}+1}}{F_{\bar{n}+2}} I^0 + \frac{\epsilon}{F_{\bar{n}+2}} (F_{\bar{n}+1}F_{\bar{n}} - F_{\bar{n}+2}F_{\bar{n}-1}) = \\ &= \frac{F_{\bar{n}+1}}{F_{\bar{n}+2}} I^0 + \frac{\epsilon}{F_{\bar{n}+2}} (-1)^{\bar{n}-1}. \end{aligned}$$

Esempio

- È dato un intervallo di incertezza iniziale $I^0 = [0, 100]$.
- È richiesto un massimo intervallo di incertezza finale $\Delta = 2$.
- È data una risoluzione $\epsilon = 1$.

Esempio

Scelta del numero di iterazioni

$$\bar{n} = \min\left\{n \mid \frac{100}{F_{n+2}} + \frac{F_n}{F_{n+2}} \leq 2\right\} = 9.$$

Infatti si ha:

$$\text{per } n = 8: \frac{100}{F_{10}} + \frac{F_8}{F_{10}} = 100/55 + 21/55 = 121/55 > 2,$$

$$\text{per } n = 9: \frac{100}{F_{11}} + \frac{F_9}{F_{11}} = 100/89 + 34/89 = 134/89 < 2.$$

k	0	1	2	3	4	5	6	7	8	9	10	11	...
F_k	0	1	1	2	3	5	8	13	21	34	55	89	...

Scelta del primo punto interno

$$\begin{aligned} I^1 &= \frac{F_{\bar{n}+1}}{F_{\bar{n}+2}} I^0 + \frac{\epsilon}{F_{\bar{n}+2}} (-1)^{\bar{n}-1} = \\ &= \frac{55}{89} 100 + \frac{1}{89} (-1)^8 = 5500/89 + 1/89 = 5501/89 \approx 61,81. \end{aligned}$$

Esempio

Iterazioni. Gli intervalli di incertezza successivi risultano avere le seguenti ampiezze:

$$I^2 = I^0 - I^1 = \frac{8900 - 5501}{89} = \frac{3399}{89}$$

$$I^3 = I^1 - I^2 = \frac{5501 - 3399}{89} = \frac{2102}{89}$$

$$I^4 = I^2 - I^3 = \frac{3399 - 2102}{89} = \frac{1297}{89}$$

$$I^5 = I^3 - I^4 = \frac{2102 - 1297}{89} = \frac{805}{89}$$

$$I^6 = I^4 - I^5 = \frac{1297 - 805}{89} = \frac{492}{89}$$

$$I^7 = I^5 - I^6 = \frac{805 - 492}{89} = \frac{313}{89}$$

$$I^8 = I^6 - I^7 = \frac{492 - 313}{89} = \frac{179}{89}$$

$$I^9 = I^7 - I^8 = \frac{313 - 179}{89} = \frac{134}{89}.$$

Conclusioni

- Il metodo dei numeri di Fibonacci consente di approssimare il minimo di una funzione di una sola variabile continua.
- Deve essere noto un intervallo di incertezza iniziale e la funzione deve essere unimodale in esso.
- Il metodo non richiede il calcolo della derivata prima della funzione.
- Il metodo richiede di valutare la funzione in un numero di punti dello stesso ordine di grandezza del numero di iterazioni.

Programmazione non-lineare: ottimizzazione vincolata

Giovanni Righini

Ricerca Operativa



UNIVERSITÀ DEGLI STUDI
DI MILANO

Ottimizzazione vincolata

Nell'ottimizzazione non-lineare vincolata, oltre alla funzione obiettivo

$$\text{minimize } f(x),$$

consideriamo anche l'effetto di

- vincoli di uguaglianza $h_i(x) = 0 \forall i \in \mathcal{E}$
- vincoli di disuguaglianza $g_i(x) \geq 0 \forall i \in \mathcal{I}$.

Un vincolo di disuguaglianza $j \in \mathcal{I}$ è **attivo** in una soluzione \bar{x} se e solo se $g_j(\bar{x}) = 0$.

L'**insieme attivo** $A(\bar{x})$ è l'insieme dei vincoli attivi in \bar{x} .

In ogni punto \bar{x} ammissibile, $A(\bar{x})$ comprende sempre tutti i vincoli di uguaglianza.

Vincoli di uguaglianza

Consideriamo un vincolo $c(x) = 0$ ed un punto \bar{x} su di esso.

Indichiamo con $\nabla c(\bar{x})$ la direzione della normale al vincolo in \bar{x} .

Consideriamo un passo infinitesimo da \bar{x} lungo una direzione d .

Per mantenere l'ammissibilità rispetto al vincolo, d deve essere tale che:

$$\nabla c(\bar{x})^T d = 0.$$

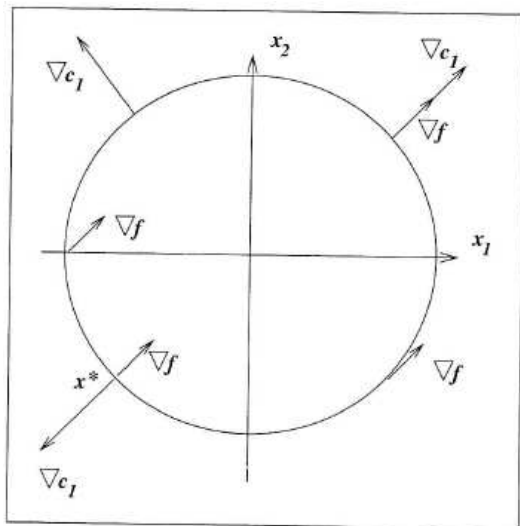
Il passo produce un miglioramento nel valore di $f(x)$ (da minimizzare) se e solo se

$$\nabla f(\bar{x})^T d < 0.$$

Quindi un passo migliorante da \bar{x} *non* è possibile se e solo se

$$\exists \bar{\lambda} \neq 0 : \nabla c(\bar{x}) = \bar{\lambda} \nabla f(\bar{x})$$

Vincoli di uguaglianza



Vincoli di uguaglianza

Un modo alternativo di formulare la stessa condizione di ottimalità in un punto \bar{x} consiste nell'introdurre la **funzione Lagrangiana**

$$\mathcal{L}(x, \lambda) = f(x) - \lambda c(x).$$

Si ha

$$\nabla_x \mathcal{L}(x, \lambda) = \nabla f(x) - \lambda \nabla c(x).$$

Quindi la condizione di ottimalità in \bar{x}

$$\exists \bar{\lambda} \neq 0 : \nabla c(\bar{x}) = \bar{\lambda} \nabla f(\bar{x})$$

equivale alla condizione

$$\exists \bar{\lambda} \neq 0 : \nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}) = 0.$$

Si tratta di una **condizione necessaria** del primo ordine, ma non sufficiente (proprio come nel caso non-vincolato).

Vincoli di disuguaglianza

Consideriamo un vincolo di disuguaglianza $g(x) \geq 0$ ed un punto \bar{x} su di esso.

Il gradiente $\nabla g(x)$ è un vettore che punta verso l'interno della regione ammissibile, dato che il vincolo è nella forma $g(x) \geq 0$ (se $f(x)$ fosse da massimizzare porremmo i vincoli di disuguaglianza nella forma $g(x) \leq 0$).

Il punto \bar{x} **non** è ottimo se esiste uno spostamento infinitesimo d tale da migliorare il valore dell'obiettivo e da mantenere l'ammissibilità, cioè tale che

$$\nabla f(\bar{x})d < 0$$

e

$$\nabla g(\bar{x})d \geq 0.$$

Tali condizioni non possono essere vere entrambe solo se

$$\exists \bar{\lambda} \geq 0 : \nabla f(\bar{x}) = \bar{\lambda} \nabla g(\bar{x}).$$

Vincoli di disuguaglianza

Quando invece il punto \bar{x} non è sul vincolo, allora si può avere uno spostamento infinitesimo d ammissibile e migliorante quando

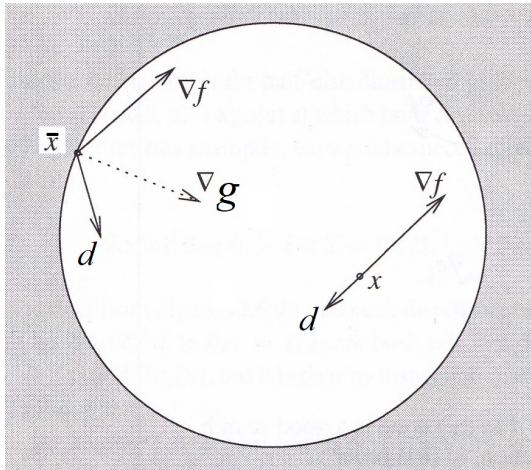
$$\nabla f(\bar{x})d < 0$$

e d è abbastanza piccolo da non superare lo slack del vincolo.

Quindi la condizione necessaria del primo ordine per l'ottimalità in \bar{x} è la stessa del caso non-vincolato

$$\nabla f(\bar{x}) = 0.$$

Vincoli di disuguaglianza



Vincoli di disuguaglianza

Un modo alternativo di formulare la stessa condizione di ottimalità in un punto \bar{x} consiste nell'introdurre la **funzione Lagrangiana**

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x).$$

Si ha

$$\nabla_x \mathcal{L}(x, \lambda) = \nabla f(x) - \lambda \nabla g(x).$$

Quindi la condizione di ottimalità in \bar{x}

$$\begin{cases} \exists \bar{\lambda} \geq 0 : \nabla f(\bar{x}) = \bar{\lambda} \nabla g(\bar{x}) & \text{se } g(\bar{x}) = 0 \\ \nabla f(\bar{x}) = 0 & \text{se } g(\bar{x}) > 0 \end{cases}$$

equivale alle condizioni

$$\begin{aligned} \exists \bar{\lambda} \geq 0 : \nabla_x \mathcal{L}(\bar{x}, \bar{\lambda}) &= 0 \\ \bar{\lambda} g(\bar{x}) &= 0 \end{aligned}$$

Due vincoli di disuguaglianza

Consideriamo due vincoli di disuguaglianza $g_1(x) \geq 0$, $g_2(x) \geq 0$ ed un punto \bar{x} , dove entrambi sono attivi.

Indichiamo con $\nabla g_1(\bar{x})$ e $\nabla g_2(\bar{x})$ la direzione della normale ai vincoli in \bar{x} .

Consideriamo un passo infinitesimo da \bar{x} lungo una direzione d .

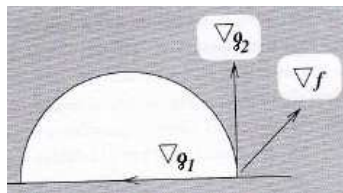
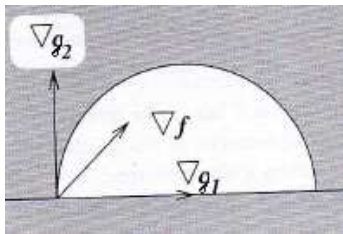
Per l'ammissibilità, d deve essere tale che:

$$\nabla g_1(\bar{x})^T d \geq 0 \quad \text{e} \quad \nabla g_2(\bar{x})^T d \geq 0.$$

Il passo produce un miglioramento di $f(x)$ se e solo se

$$\nabla f(\bar{x})^T d < 0.$$

Vincoli di disuguaglianza



Direzioni ammissibili

Una direzione d è ammissibile in \bar{x} se e solo se:

$$\nabla c_i(\bar{x})^T d = 0 \quad \forall i \in \mathcal{E} \quad \text{e} \quad \nabla c_i(\bar{x})^T d \geq 0 \quad \forall i \in \mathcal{A}(\bar{x}) \cap \mathcal{I},$$

dove $\mathcal{A}(\bar{x})$ indica l'insieme dei vincoli di disuguaglianza attivi in \bar{x} .

Proprietà *linear independence constraint qualification (LICQ)* in un punto \bar{x} : tutti i gradienti dei vincoli attivi in $\mathcal{A}(\bar{x})$ sono linearmente indipendenti.

Condizioni di ottimalità del primo ordine

Definita la funzione Lagrangiana

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(\mathbf{x})$$

si hanno le seguenti condizioni necessarie del primo ordine affinché un punto sia un minimo locale.

Condizioni di Karush-Kuhn-Tucker (KKT). Se

- \mathbf{x}^* è un minimo locale di $f(\mathbf{x})$,
- $f(\mathbf{x})$ e $c_i(\mathbf{x})$ sono funzioni continue e differenziabili,
- la LICQ è soddisfatta in \mathbf{x}^* ,

allora esiste λ^* tale che

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \lambda^*) = 0$$

$$c_i(\mathbf{x}^*) = 0 \quad \forall i \in \mathcal{E}$$

$$c_i(\mathbf{x}^*) \geq 0 \quad \forall i \in \mathcal{I}$$

$$\lambda_i^* \geq 0 \quad \forall i \in \mathcal{I}$$

$$\lambda_i^* c_i(\mathbf{x}^*) = 0 \quad \forall i \in \mathcal{E} \cup \mathcal{I}$$

Complementarità

Le condizioni di complementarità

$$\lambda_i^* c_i(\mathbf{x}^*) = 0 \quad \forall i \in \mathcal{E} \cup \mathcal{I}$$

richiedono che

- o il vincolo $c_i(x)$ sia attivo,
- o il corrispondente moltiplicatore λ_i sia nullo,
- o entrambe le cose.

Poiché $\lambda_i^* = 0 \quad \forall i \notin A(\mathbf{x}^*)$, la condizione del primo ordine si può riscrivere come

$$\nabla f(\mathbf{x}^*) - \sum_{i \in A(\mathbf{x}^*)} \lambda_i^* \nabla c_i(\mathbf{x}^*) = 0$$

Complementarità stretta

Si ha complementarità stretta quando solo una tra λ_i^* e $c_i(x^*)$ è nulla $\forall i \in A(x^*)$.

Per uno stesso punto x^* potrebbero esistere diversi λ_i^* che soddisfano le condizioni KKT.

Ma se valgono le condizioni LICQ, allora λ_i^* è unico.

Condizioni necessarie del secondo ordine

Assumiamo che $f(x)$ e $c_i(x)$ siano tutte continue e differenziabili fino al secondo ordine.

Siano

- $\mathcal{F}(x^*)$, l'insieme delle direzioni ammissibili in x^* ;
- λ^* un vettore di moltiplicatori Lagrangiani che soddisfa le KKT in x^* .

Cono critico.

$$\mathcal{C}(x^*, \lambda^*) = \{w \in \mathcal{F}(x^*) : \nabla c_i(x^*)^T w = 0 \ \forall i \in A(x^*) \cap \mathcal{I} : \lambda_i^* > 0\}.$$

Quindi:

$$w \in \mathcal{C}(x^*, \lambda^*) \Leftrightarrow \begin{cases} \nabla c_i(x^*)^T w = 0 & \forall i \in \mathcal{E} \\ \nabla c_i(x^*)^T w = 0 & \forall i \in A(x^*) \cap \mathcal{I} : \lambda_i^* > 0 \\ \nabla c_i(x^*)^T w \geq 0 & \forall i \in A(x^*) \cap \mathcal{I} : \lambda_i^* = 0 \end{cases}$$

Cono critico

Dato che $\lambda_i^* = 0 \ \forall i \notin A(x^*)$,

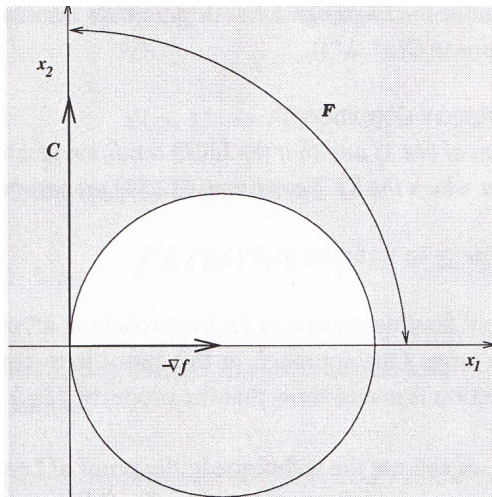
$$w \in \mathcal{C}(x^*, \lambda^*) \Rightarrow \lambda_i^* \nabla c_i(x^*)^T w = 0 \ \forall i \in \mathcal{E} \cup \mathcal{I}.$$

Dalla definizione di $\mathcal{L}(x, \lambda)$ e dalle KKT

$$w \in \mathcal{C}(x^*, \lambda^*) \Rightarrow w^T \nabla f(x^*) = \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i^* w^T \nabla c_i(x^*) = 0.$$

Quindi il cono critico contiene quelle direzioni ammissibili per le quali le derivate prime non danno informazioni sufficienti.

Cono critico



Condizioni del secondo ordine

Condizioni necessarie del secondo ordine. Sia x^* un minimo locale di $f(x)$ in cui sono soddisfatte le LICQ. Sia λ^* un vettore di moltiplicatori Lagrangiani che soddisfa le KKT in x^* . Allora

$$w^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w \geq 0 \quad \forall w \in \mathcal{C}(x^*, \lambda^*).$$

Condizioni sufficienti del secondo ordine. Sia x^* una soluzione ammissibile e sia λ^* un vettore di moltiplicatori Lagrangiani che soddisfa le KKT in x^* . Se

$$w^T \nabla_{xx}^2 \mathcal{L}(x^*, \lambda^*) w > 0 \quad \forall w \in \mathcal{C}(x^*, \lambda^*), w \neq 0,$$

allora x^* è un minimo locale di $f(x)$.

Algoritmi

Per ogni dato sottoinsieme di vincoli attivi (il *working set*), è possibile risolvere un problema di PNL non vincolata.

Tuttavia questo metodo soffre per l'esplosione combinatoria nel numero di sottinsieme che è necessario considerare.

I **metodi *active set*** eseguono una ricerca “intelligente”, scartando a priori alcuni sottinsiemi.

I **metodi del punto interno** o **metodi a barriera** invece producono sequenze di punti che non rendono attivo alcun vincolo di disuguaglianza, bensì si avvicinano asintoticamente al contorno della regione ammissibile.

Funzioni di merito e filtri

In generale gli algoritmi di PNL devono bilanciare due effetti di ogni passo:

- il miglioramento della funzione obiettivo
- il peggioramento nella violazione di alcuni vincoli

Una **funzione di merito** combina insieme i due effetti tramite un opportuno *penalty parameter* μ .

Una funzione di merito $\Phi(x, \mu)$ è **esatta** quando esiste un valore scalare positivo μ^* tale che per ogni valore $\mu > \mu^*$ ogni minimo locale del problema di PNL vincolata è un minimo locale di $\Phi(x, \mu)$.

Funzioni di merito

Un esempio è la l_1 -*penalty function*:

$$\Phi_1(\mathbf{x}, \mu) = f(\mathbf{x}) + \mu \sum_{i \in \mathcal{E}} |c_i(\mathbf{x})| + \mu \sum_{i \in \mathcal{I}} [c_i(\mathbf{x})]^-$$

dove $[k]^-$ indica $\max\{0, -k\}$.

La funzione $\Phi_1(\mathbf{x}, \mu)$ non è differenziabile ovunque, ma è esatta.

Il valore soglia è dato da

$$\mu^* = \max_{i \in \mathcal{E} \cup \mathcal{I}} \{|\lambda_i^*|\},$$

dove λ_i^* indica il vettore dei moltiplicatori duali corrispondenti ad una soluzione ottima \mathbf{x}^* .

Dato che λ_i^* non è noto a priori, occorre iterativamente ri-calibrare il valore di μ .

Funzioni di merito

Un altro esempio è la l_2 -*penalty function* che nel caso di vincoli di uguaglianza ha la forma

$$\Phi_2(x, \mu) = f(x) + \mu \|c_i(x)\|_2.$$

Anche questa non è differenziabile perché la derivata non è definita dove $c(x) = 0$.

Funzioni di merito

La funzione di merito *Fletcher's augmented Lagrangian* è sia differenziabile che esatta:

$$\Phi_F(x, \mu) = f(x) - \lambda(x)^T c(x) + \frac{1}{2} \mu \sum_{i \in \mathcal{E}} c_i(x)^2$$

dove

$$\lambda(x) = [A(x)A(x)^T]^{-1} A(x) \nabla f(x)$$

e $A(x)$ indica lo Jacobiano di $c(x)$.

Tuttavia è pesante a causa del calcolo di $\lambda(x)$.

Funzioni di merito

La funzione di merito Lagrangiana aumentata è

$$\mathcal{L}_A(\mathbf{x}, \lambda, \mu) = f(\mathbf{x}) - \lambda^T \mathbf{c}(\mathbf{x}) + \frac{1}{2}\mu \|\mathbf{c}(\mathbf{x})\|_2^2.$$

Si accetta un punto prossimo $(\mathbf{x}^{k+1}, \lambda^{k+1})$ se la \mathcal{L}_A diminuisce rispetto al punto corrente (\mathbf{x}, λ) .

Gli algoritmi che usano questa funzione di merito includono criteri per modificare opportunamente i valori di λ e μ .

Derivate direzionali

Le funzioni non differenziabili hanno tuttavia **derivate direzionali**: data una funzione $f(x)$ ed una direzione p , la derivata direzionale di $f(x)$ nella direzione p è

$$D(f(x), p) = \lim_{\epsilon \rightarrow 0} \frac{f(x + \epsilon p) - f(x)}{\epsilon}.$$

Quando $f(x)$ è continua e differenziabile in un intorno di x , si ha

$$D(f(x), p) = \nabla f(x)^T p.$$

Derivate direzionali

In un metodo *line search* la condizione per accettare un passo α è che sia abbastanza piccolo affinché la disequazione

$$\Phi(\mathbf{x} + \alpha \mathbf{p}, \mu) \leq \Phi(\mu, \mathbf{x}) + \eta \alpha D(\Phi(\mathbf{x}, \mu), \mathbf{p})$$

sia soddisfatta per qualche $0 \leq \eta \leq 1$.

I metodi *trust region* usano tipicamente un modello quadratico q per stimare il valore di Φ dopo un passo p .

La condizione sufficiente per accettare un passo è

$$\Phi(\mathbf{x} + p, \mu) \leq \Phi(\mathbf{x}, \mu) - \eta(q(0) - q(p))$$

per qualche $0 \leq \eta \leq 1$.

Filtri

Negli algoritmi basati sui filtri l'ottimalità e l'ammissibilità vengono trattate come due obiettivi distinti, come nella PMO, e vengono accettate le soluzioni x non-dominate, cioè quelle per cui non è stata trovata in precedenza alcuna soluzione x' con $f(x') \leq f(x)$ e $h(x') \leq h(x)$, dove

$$h(x) = \sum_{i \in \mathcal{E}} |c_i(x)| + \sum_{i \in \mathcal{I}} [c_i(x)]^-$$

indica una misura della violazione dei vincoli.

Nei metodi *line search* una soluzione $x^{k+1} = x^k + \alpha_k p_k$ viene accettata se (f^{k+1}, h^{k+1}) è una coppia di valori non-dominata.

Nei metodi *trust region*, se una soluzione x^{k+1} non viene accettata, si riduce il raggio e si ripete l'iterazione.

In entrambi i casi vengono intercalate iterazioni di ripristino dell'ammissibilità (*feasibility restoration phases*), dove viene minimizzata solo $h(x)$.

L'algoritmo del simplesso rivisto

Ricerca operativa

Giovanni Righini



UNIVERSITÀ DEGLI STUDI
DI MILANO

L'algoritmo *revised simplex*

Per eseguire i test di ammissibilità e ottimalità e scegliere il pivot su cui eseguire la prossima iterazione non è necessario conoscere tutti i coefficienti del tableau.

L'idea quindi è di rappresentare il tableau in un modo alternativo, ma equivalente, risparmiando alcune operazioni.

A questo scopo si sfrutta la dualità.

Coppia duale

Consideriamo un problema di PL in forma standard

$$\begin{aligned} \text{P) minimize } z &= c^T x \\ \text{subject to } Ax &= b \\ x &\geq 0 \end{aligned}$$

con $n + m$ variabili non-negative e m vincoli di uguaglianza.

Il suo duale è

$$\begin{aligned} \text{D) maximize } w &= b^T y \\ \text{subject to } A^T y &\leq c \end{aligned}$$

con m variabili libere e $m + n$ vincoli di disuguaglianza.

Base primale

Scelta una base di m colonne, il primale si può riscrivere come segue:

$$\begin{aligned} \text{P) minimize } z &= c_B^T x_B + c_N^T x_N \\ \text{subject to } Bx_B + Nx_N &= b \\ x_B, x_N &\geq 0. \end{aligned}$$

dove

$$\begin{aligned} A &= [B|N] \\ x^T &= [x_B|x_N] \\ c^T &= [c_B|c_N] \end{aligned}$$

Base duale

Il duale si può mettere a sua volta in forma standard, inserendo variabili non-negative di slack:

$$\begin{aligned} \text{D) maximize } w &= b^T y \\ \text{subject to } A^T y + s &= c \end{aligned}$$

con m variabili y libere, $m + n$ variabili $s \geq 0$ e $m + n$ equazioni.

Dato che $A^T = \begin{bmatrix} B^T \\ N^T \end{bmatrix}$, il duale si può riscrivere come segue:

$$\begin{aligned} \text{D) maximize } w &= b^T y \\ \text{subject to } B^T y + s_B &= c_B \\ N^T y + s_N &= c_N. \end{aligned}$$

Per il teorema degli scarti complementari, per ogni coppia di basi primale/duale che si corrispondono si ha

$$x_i s_i = 0.$$

Quindi $s_B = 0$.

Base duale

Da $B^T y = c_B$ si ottiene

$$y = (B^T)^{-1} c_B.$$

Da $N^T y + s_N = c_N$, si ottiene

$$\begin{aligned} s_N &= c_N - N^T (B^T)^{-1} c_B = \\ &= c_N - N^T (B^{-1})^T c_B = \\ &= c_N - (B^{-1} N)^T c_B. \end{aligned} \tag{1}$$

Nel primale invece si ha

$$x_B = B^{-1} b$$

$$x_N = 0.$$

Così le soluzioni primale e duale si possono calcolare a partire da B , N e i dati **iniziali**.

Cambio di base

Per eseguire un passo di pivot, si sceglie una variabile primale fuori base con costo ridotto negativo, cioè con colonna $q \in N$ tale che $s_q < 0$.

Indichiamo con $T = B^{-1}A$ il tableau corrente (che non vogliamo calcolare esplicitamente). Sia T_q la sua colonna (l'unica che calcoliamo esplicitamente) corrispondente alla variabile x_q :

$$T_q = B^{-1}A_q,$$

dove A_q indica la colonna q della matrice A .

Indichiamo con $p \in B$ l'indice della variabile primale uscente:

$$p = \operatorname{argmin}_{i: T_{iq} > 0} \left\{ \frac{(x_B)_i}{T_{iq}} \right\},$$

dove $(x_B)_i$ indica il valore della variabile che è in base sulla riga i .

Il nuovo valore di x_q , entrando in base, è $x'_q = \frac{(x_B)_p}{T_{pq}}$.

Cambio di base

Diamo una descrizione algebrica dell'operazione di pivot dalla soluzione di base x alla soluzione di base x' , sapendo che:

$$Ax = b \quad Ax' = b$$

perchè entrambe le soluzioni sono ammissibili,

$$x_N = 0 \quad x'_i = 0 \quad \forall i \in N \setminus \{q\}.$$

Poiché

$$b = Ax = Bx_B$$

$$b = Ax' = Bx'_B + A_q x'_q,$$

si ha

$$x'_B = x_B - B^{-1} A_q x'_q.$$

Nel duale

$$y^T = c_B^T B^{-1}$$

$$A_q^T y + s_q = c_q \quad \text{ossia} \quad s_q = c_q - y^T A_q.$$

Obiettivo

Il valore dell'obiettivo z dopo il pivot è

$$\begin{aligned} z' &= c^T x' = c_B^T x'_B + c_q x'_q = \\ &= c_B^T x_B - c_B^T B^{-1} A_q x'_q + c_q x'_q = \\ &= c_B^T x_B - y^T A_q x'_q + c_q x'_q = \\ &= c_B^T x_B - (c_q - s_q) x'_q + c_q x'_q = \\ &= c_B^T x_B + s_q x'_q = z + s_q x'_q. \end{aligned}$$

Se l'iterazione non è degenera, $s_q < 0$ e $x'_q > 0$ implicano $z' < z$.

Quindi tutto ciò che serve per procedere con le iterazioni dell'algoritmo del simplesso (x , y , s , z) può essere calcolato a partire dai dati iniziali eseguendo solo prodotti tra vettore e matrice e non tra matrici, pur di conoscere B^{-1} .

Fattorizzazione LU

Per calcolare rapidamente l'inversa di B , si rappresenta $B = LU$ come prodotto tra

- una matrice L *unit lower triangular* (gli elementi sulla diagonale hanno valore 1 e sopra la diagonale hanno valore 0);
- una matrice U *upper triangular* (gli elementi sotto la diagonale hanno valore 0).

Il calcolo di $T_q = B^{-1}A_q$, cioè tale che $BT_q = A_q$, si divide in due step:

- Trovare $d : Ld = A_q$
- Trovare $T_q : UT_q = d$.

Analogamente, il calcolo di $y = (B^{-1})^T c_B$, cioè tale che $B^T y = c_B$, si divide in due step:

- Trovare $d : U^T d = c_B$
- Trovare $y : L^T y = d$.

Tutti questi step sono calcolabili efficientemente per eliminazione Gaussiana.

Aggiornamento di L e U

Nel passaggio da B a B' , quando x_q entra in base e x_p esce, bisogna aggiornare efficientemente L e U .

$L^{-1}B = U$ è triangolare superiore.

Sia j l'indice della colonna di U che corrisponde a x_p .

$L^{-1}B'$ è ancora triangolare superiore tranne che nella colonna j .

Aggiornamento di L e U

Con

- una permutazione ciclica delle colonne che sposta j in fondo e tutte le colonne da $j + 1$ a n a sinistra di una posizione,
- una permutazione ciclica delle righe che sposta la riga j in fondo e tutte le righe da $j + 1$ a n in alto di una posizione,

si ottiene una nuova matrice triangolare superiore con in aggiunta alcuni elementi non-zero sull'ultima riga.

Essa si può esprimere come prodotto tra una matrice L' identica a L tranne che nell'ultima riga ed una matrice U' identica alla matrice permutata tranne che nell'ultimo elemento in basso a destra.

Anche questi nuovi coefficienti si ricavano in modo efficiente per eliminazione Gaussiana.

Esempio

$$L^{-1}B = U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\ & u_{22} & u_{23} & u_{24} & u_{25} \\ & & u_{33} & u_{34} & u_{35} \\ & & & u_{44} & u_{45} \\ & & & & u_{55} \end{bmatrix} \quad L^{-1}A_q = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix}$$

Supponiamo che la colonna modificata sia $j = 2$.

$$L^{-1}B' = \begin{bmatrix} u_{11} & w_1 & u_{13} & u_{14} & u_{15} \\ & w_2 & u_{23} & u_{24} & u_{25} \\ & w_3 & u_{33} & u_{34} & u_{35} \\ & w_4 & & u_{44} & u_{45} \\ & w_5 & & & u_{55} \end{bmatrix}$$

Indichiamo con P_j la matrice della permutazione ciclica da j a n .

$$P_j L^{-1} B' P_j^T = \begin{bmatrix} u_{11} & u_{13} & u_{14} & u_{15} & w_1 \\ & u_{33} & u_{34} & u_{35} & w_3 \\ & & u_{44} & u_{45} & w_4 \\ & & & u_{55} & w_5 \\ & u_{23} & u_{24} & u_{25} & w_2 \end{bmatrix}$$

Esempio

La matrice

$$P_j L^{-1} B' P_j^T = \begin{bmatrix} u_{11} & u_{13} & u_{14} & u_{15} & w_1 \\ & u_{33} & u_{34} & u_{35} & w_3 \\ & & u_{44} & u_{45} & w_4 \\ & & & u_{55} & w_5 \\ & u_{23} & u_{24} & u_{25} & w_2 \end{bmatrix}$$

è il prodotto di due matrici triangolari L_1 e U_1 :

$$L_1 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ 0 & l_{52} & l_{53} & l_{54} & 1 \end{bmatrix} \quad U_1 = \begin{bmatrix} u_{11} & u_{13} & u_{14} & u_{15} & w_1 \\ & u_{33} & u_{34} & u_{35} & w_3 \\ & & u_{44} & u_{45} & w_4 \\ & & & u_{55} & w_5 \\ & & & & \hat{w}_2 \end{bmatrix}$$

Esempio

I valori dei nuovi coefficienti si calcolano per eliminazione Gaussiana.

$$\begin{bmatrix} u_{11} & u_{13} & u_{14} & u_{15} & w_1 \\ & u_{33} & u_{34} & u_{35} & w_3 \\ & & u_{44} & u_{45} & w_4 \\ & & & u_{55} & w_5 \\ & u_{23} & u_{24} & u_{25} & w_2 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ 0 & l_{52} & l_{53} & l_{54} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{13} & u_{14} & u_{15} & w_1 \\ & u_{33} & u_{34} & u_{35} & w_3 \\ & & u_{44} & u_{45} & w_4 \\ & & & u_{55} & w_5 \\ & & & & \hat{w}_2 \end{bmatrix}$$

$$l_{52} u_{33} = u_{23}$$

$$l_{52} u_{34} + l_{53} u_{44} = u_{24}$$

$$l_{52} u_{35} + l_{53} u_{45} + l_{54} u_{55} = u_{25}$$

$$l_{52} w_3 + l_{53} w_4 + l_{54} w_5 + \hat{w}_2 = w_2$$

Quindi la nuova fattorizzazione $B' = L' U'$ si ottiene così:

$$L_1 U_1 = P_j L^{-1} B' P_j^T \Leftrightarrow L_1 U_1 = (P_j L^{-1} L') (U' P_j^T)$$

$$L_1 = P_j L^{-1} L' \text{ e quindi } L' = L P_j^T L_1$$

$$U_1 = U' P_j^T \text{ e quindi } U' = U_1 P_j.$$

DOMANDE RICERCA OPERATIVA

Programmazione Lineare (PL)

Cap.2 – Programmazione Lineare

Come riconosco dal tableau che mi trovo in una soluzione di base degenera?

Cap. 4 – Teoria della Dualità

Enunciare il teorema dello scarto complementare

Enunciare il teorema della dualità nella forma debole e nella forma forte

Descrivere quali sono i criteri per eseguire un passo di pivot nell'algoritmo del simplesso duale

Simplesso duale come funziona e perché lo usiamo?

Bound duale / Bound primario

Lemma di Farkas, enunciarlo, a cosa serve, dove si usa

Cap. 6 – Programmazione Lineare a due obiettivi

Dare la definizione di dominanza nella programmazione a molti obiettivi

Definizione di soluzione non dominata ed esempi

Programmazione multilivello

Soluzione paretiana

Metodo delle curve di indifferenza spiegare come funziona

Criterio del punto utopia

Programmazione Lineare Intera (PLI)

Cap. 7-8 – Programmazione Lineare Intera + Branch and Bound

Dare la definizione di rilassamento

Definire le regole di branching

Dare la definizione di bound

Parlare del branching

Branch and bound. Abbandono l'ottimo per una soluzione approssimata in modo da avere un algoritmo che non ci metta troppo, come faccio?

Quali condizioni deve soddisfare il branching nel branching and bound

Taglio di Gomory, come si fa a generare?

Programmazione Non Lineare (PNL)

Cap. 10a – PNL non vincolata

Dare la definizione di minimo locale

Dare la definizione di velocità di convergenza

Descrivere l'algoritmo del gradiente

Algoritmi iterativi (credo intenda il metodo di bisezione)

Line search e trust region

Cos'è la programmazione convessa?

Come capisco che un problema è di programmazione convessa?

Metodo di Newton

Cap. 10b – PNL monodimensionale

Nella PNL, quando ho scelto la direzione, come scelgo il passo?

Cap. 10c – PNL vincolata

Metodi Active set e metodi a barriera

Karush Kuhn Tucker

Altro

Cap. 11 – Algoritmo del Simplexso rivisto

Simplexso rivisto - ha disegnato la matrice, le formule e descritto i passi [per il 30L]

Regola di Bland

Quando ho l'unicità della soluzione in PL, PLI e PNL