

Dispensa di Ottimizzazione

In formato presentazione A5 orizzontale

Questa dispensa è pensata per fornire una trattazione completa degli argomenti fondamentali dell'ottimizzazione matematica, con un'esposizione rigorosa ma accessibile, adatta alla preparazione di esami universitari.

1. Programmazione Lineare

La **Programmazione Lineare** (PL) è una tecnica di ottimizzazione in cui la funzione obiettivo e tutti i vincoli sono espressi da relazioni lineari. È ampiamente utilizzata in ambiti produttivi, logistici, finanziari e gestionali.

1.1 Forma standard di un problema di PL

Un problema di programmazione lineare si scrive nella seguente forma standard:

$$\begin{array}{ll}\min & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}\end{array}$$

dove:

- $\mathbf{x} \in \mathbb{R}^n$ è il vettore delle variabili decisionali;
- $\mathbf{c} \in \mathbb{R}^n$ è il vettore dei coefficienti della funzione obiettivo;
- $A \in \mathbb{R}^{m \times n}$ è la matrice dei coefficienti dei vincoli;

- $\mathbf{b} \in \mathbb{R}^m$ è il vettore dei termini noti;
- $\mathbf{x} \geq \mathbf{0}$ rappresenta il vincolo di non negatività.

1.2 Forma canonica

Alternativamente, in forma canonica:

$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

Le disuguaglianze possono essere trasformate in uguaglianze con l'introduzione di variabili di scarto (slack).

1.3 Ipotesi della Programmazione Lineare

Affinché un problema sia considerato lineare, devono essere verificate:

- **Linearità:** la funzione obiettivo e i vincoli sono combinazioni lineari.
- **Additività:** ogni termine contribuisce indipendentemente al risultato.
- **Divisibilità:** le variabili possono assumere qualsiasi valore reale.
- **Determinismo:** tutti i dati del problema sono noti con certezza.

1.4 Interpretazione geometrica

L'insieme delle soluzioni ammissibili di un problema di PL è un **poliedro convesso**. La funzione obiettivo lineare definisce iperpiani in \mathbb{R}^n .

Teorema fondamentale: Se esiste una soluzione ottima, essa si trova in un vertice del poliedro.

1.5 Esempio pratico

$$\begin{array}{ll}\max & 3x_1 + 2x_2 \\ \text{s.t.} & x_1 + x_2 \leq 4 \\ & 2x_1 + x_2 \leq 5 \\ & x_1, x_2 \geq 0\end{array}$$

Introducendo variabili di slack s_1, s_2 per passare alla forma standard:

$$\begin{array}{ll}x_1 + x_2 + s_1 &= 4 \\ 2x_1 + x_2 + s_2 &= 5 \\ x_1, x_2, s_1, s_2 &\geq 0\end{array}$$

1.6 Teorema di esistenza dell'ottimo

Teorema di Weierstrass: Se l'insieme delle soluzioni ammissibili è non vuoto, chiuso e limitato, e la funzione obiettivo è continua, allora esiste almeno una soluzione ottima.

Nel caso della PL, se il politopo ammissibile è limitato, la funzione lineare raggiunge un massimo (o minimo) in uno dei suoi vertici.

1.7 Soluzione multipla e degenerazione

- **Soluzione multipla:** può esistere più di una soluzione ottima, se l'ottimo si trova su una faccia del politopo.
- **Degenerazione:** si verifica quando più di una base ammissibile corrisponde alla stessa soluzione.

1.8 Problemi illimitati o infeasibili

- **Illimitato:** la funzione obiettivo può crescere indefinitamente. Succede quando manca un vincolo che “limita” la direzione dell’ottimo.
- **Infeasible:** l’insieme ammissibile è vuoto (i vincoli sono incompatibili).

2. Metodo del Simplexso

Il **Metodo del Simplexso** è un algoritmo iterativo per la risoluzione dei problemi di Programmazione Lineare (PL) in forma standard. L’idea di base è di spostarsi da un vertice del politopo ammissibile a un altro, migliorando la funzione obiettivo a ogni passo, fino a trovare una soluzione ottima.

2.1 Forma standard del problema

$$\begin{array}{ll}\min & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0\end{array}$$

2.2 Definizione di base e soluzione di base

Una **base** è un insieme di m colonne linearmente indipendenti della matrice $A \in \mathbb{R}^{m \times n}$. Le variabili associate sono chiamate **variabili di base**.

Una **soluzione di base** si ottiene ponendo a zero le variabili non di base (non in B) e risolvendo $A_B \mathbf{x}_B = \mathbf{b}$.

2.3 Condizioni di ottimalità

Una soluzione di base ammissibile \mathbf{x} è ottima se e solo se tutti i **costi ridotti** sono non negativi:

$$\bar{c}_j = c_j - \mathbf{c}_B^\top A_B^{-1} A_j \geq 0 \quad \forall j \notin B$$

dove:

- \mathbf{c}_B sono i coefficienti delle variabili di base;
- A_B è la matrice delle colonne di base;
- A_j è la colonna j -esima di A ;
- \bar{c}_j è il costo ridotto della variabile x_j .

2.4 Fasi dell'algoritmo

1. Selezionare una soluzione di base iniziale ammissibile.
2. Calcolare i costi ridotti.
3. Se $\bar{c}_j \geq 0$ per tutti j , la soluzione è ottima.
4. Altrimenti, scegliere una variabile entrante con $\bar{c}_j < 0$.
5. Determinare la variabile uscente usando il **test del rapporto minimo**:

$$\min_i \left\{ \frac{x_{B_i}}{(A_B^{-1} A_j)_i} \mid (A_B^{-1} A_j)_i > 0 \right\}$$

6. Eseguire l'operazione di pivot (aggiornamento della base).
7. Ripetere dal punto 2.

2.5 Criteri di scelta delle variabili

- **Entrante:** scegliere la variabile non basica con il costo ridotto più negativo (regola del massimo miglioramento), oppure la più a sinistra (regola di Bland).
- **Uscente:** determinata dal test del rapporto minimo.

2.6 Degenerazione e ciclicità

La **degenerazione** si verifica quando almeno una variabile di base è nulla. In presenza di degenerazione, l'algoritmo può non migliorare la funzione obiettivo. La **regola di Bland** previene cicli infiniti.

2.7 Esempio numerico

Consideriamo il problema:

$$\begin{aligned} \max \quad & z = 3x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 4 \\ & 2x_1 + x_2 \leq 5 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Forma standard con variabili di slack s_1, s_2 :

$$\begin{aligned} x_1 + x_2 + s_1 &= 4 \\ 2x_1 + x_2 + s_2 &= 5 \\ x_1, x_2, s_1, s_2 &\geq 0 \end{aligned}$$

Si costruisce la tabella iniziale del simplesso e si procede con l'algoritmo finché si verifica l'ottimalità.

2.8 Soluzione illimitata e infeasibile

- Se tutti gli elementi della colonna pivot sono ≤ 0 , il problema è **illimitato**.
- Se non esiste base ammissibile iniziale, si utilizza il **metodo delle due fasi**.

3. Metodo delle Due Fasi

Il **Metodo delle Due Fasi** è una tecnica per trovare una soluzione di base ammissibile iniziale in problemi di Programmazione Lineare in cui non è immediatamente disponibile.

3.1 Motivazione

In alcuni casi, come vincoli di uguaglianza o vincoli \geq , non si riesce a costruire facilmente una base ammissibile. In tali situazioni si introducono **variabili artificiali** e si risolve un **problema ausiliario**.

3.2 Struttura del metodo

Il metodo prevede due fasi distinte:

Fase 1: Problema ausiliario

- Si introduce una variabile artificiale per ciascun vincolo che impedisce di costruire direttamente una base ammissibile.
- Si definisce una nuova funzione obiettivo:

$$\min \quad w = \sum_{j \in A} x_j^a$$

dove x_j^a sono le variabili artificiali.

- Si risolve il problema con il **simplex**.

Interpretazione: se il minimo è positivo, il problema originale è **infeasible**.

Fase 2: Ritorno al problema originale

- Se la soluzione ottima ha valore zero e le variabili artificiali sono tutte nulle, si rimuovono queste variabili e si risolve il problema originale partendo dalla base trovata.

3.3 Esempio

Consideriamo il problema:

$$\begin{array}{ll} \min & z = 3x_1 + 2x_2 \\ \text{s.t.} & x_1 + x_2 = 5 \\ & x_1 + 3x_2 \geq 3 \\ & x_1, x_2 \geq 0 \end{array}$$

Preparazione della fase 1

- Primo vincolo: uguaglianza \Rightarrow si aggiunge variabile artificiale a_1 .
- Secondo vincolo: \geq si trasforma in uguaglianza con variabile di surplus $-s_2$ e variabile artificiale a_2 .

Sistema modificato:

$$\begin{array}{l} x_1 + x_2 + a_1 = 5 \\ x_1 + 3x_2 - s_2 + a_2 = 3 \end{array}$$

Funzione obiettivo della Fase 1:

$$\min \quad w = a_1 + a_2$$

Si applica il simplesso al problema ausiliario.

3.4 Commenti sul metodo

- Alternativa al **Big M method**, più stabile numericamente.
- Garantisce l'individuazione di una base ammissibile senza introdurre costanti numeriche arbitrarie.

3.5 Conclusione

Il metodo delle due fasi è essenziale per affrontare problemi in cui la base iniziale non è ovvia o non esiste esplicitamente. È robusto e permette l'applicazione del metodo del simplesso in modo generalizzato.

4. Dualità nella Programmazione Lineare

Ogni problema di Programmazione Lineare (**primale**) ha associato un **problema duale**, che fornisce importanti informazioni teoriche ed economiche.

4.1 Formulazione del duale

Dato il problema primale in forma canonica:

$$\begin{array}{ll}\max & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq 0\end{array}$$

Il problema duale corrispondente è:

$$\begin{array}{ll}\min & \mathbf{b}^\top \mathbf{y} \\ \text{s.t.} & A^\top \mathbf{y} \geq \mathbf{c} \\ & \mathbf{y} \geq 0\end{array}$$

4.2 Interpretazione economica

- Ogni vincolo del primale corrisponde a una variabile nel duale.
- Ogni variabile del primale corrisponde a un vincolo nel duale.
- Le variabili duali \mathbf{y} rappresentano i **prezzi ombra**, ovvero quanto aumenterebbe la funzione obiettivo per unità aggiuntiva della risorsa.

4.3 Teoremi fondamentali

Dualità debole: Se \mathbf{x} è ammissibile per il primale e \mathbf{y} è ammissibile per il duale, allora

$$\mathbf{c}^\top \mathbf{x} \leq \mathbf{b}^\top \mathbf{y}$$

Dualità forte: Se esiste una soluzione ottima per entrambi i problemi e sono entrambi ammissibili, allora

$$\mathbf{c}^\top \mathbf{x}^* = \mathbf{b}^\top \mathbf{y}^*$$

Condizioni di complementarità: Se \mathbf{x}^* e \mathbf{y}^* sono soluzioni ottime del primale e del duale, allora:

$$y_i^* (A_i \mathbf{x}^* - b_i) = 0 \quad \forall i \quad (\text{vincoli attivi} \Leftrightarrow \text{moltiplicatori positivi})$$

4.4 Strategie risolutive e uso del duale

- In alcuni casi è più efficiente risolvere il duale anziché il primale.
- Il **metodo del simplesso duale** è particolarmente utile quando si parte da una soluzione ottima duale ma non ammissibile per il primale.
- Le soluzioni duali permettono di stabilire bounds per il valore ottimo.

4.5 Esempio numerico

Primale:

$$\begin{array}{ll}\max & 5x_1 + 3x_2 \\ \text{s.t.} & 2x_1 + x_2 \leq 8 \\ & x_1 + 3x_2 \leq 9 \\ & x_1, x_2 \geq 0\end{array}$$

Duale:

$$\begin{array}{ll}\min & 8y_1 + 9y_2 \\ \text{s.t.} & 2y_1 + y_2 \geq 5 \\ & y_1 + 3y_2 \geq 3 \\ & y_1, y_2 \geq 0\end{array}$$

4.6 Conclusioni

La teoria della dualità permette non solo di analizzare il problema da un punto di vista alternativo, ma anche di sviluppare algoritmi efficienti e di giustificare economicamente le soluzioni.

5. Programmazione Lineare Intera (PLI)

La **Programmazione Lineare Intera** (PLI) è una variante della Programmazione Lineare in cui alcune o tutte le variabili decisionali devono assumere valori interi.

5.1 Classificazione dei problemi interi

- **PLI**: tutte le variabili devono essere intere.
- **PLIM** (mista): solo alcune variabili devono essere intere.
- **PLIB** (binaria): le variabili intere assumono valori in $\{0, 1\}$.

5.2 Modello generale

$$\begin{aligned} \min \text{ (o max)} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}^n \text{ o } \mathbf{x} \in \mathbb{R}^n \text{ con alcune } x_i \in \mathbb{Z} \end{aligned}$$

5.3 Rilassamento continuo

Si ottiene eliminando i vincoli di interezza. Serve come base per:

- Costruzione di limiti inferiori (o superiori) per il problema intero;
- Guidare metodi di soluzione come Branch and Bound.

5.4 Formulazione ideale

Una **formulazione ideale** è una formulazione in cui il rilassamento continuo produce sempre soluzioni intere. È difficile da ottenere in generale, ma utile per ottenere limiti stretti e migliorare le prestazioni degli algoritmi.

5.5 Teorema fondamentale della PLI

Ogni punto estremo (vertice) del politopo associato a una formulazione ideale del problema intero corrisponde a una soluzione ammissibile intera.

5.6 Algoritmi principali

5.6.1 Branch and Bound (B&B)

- Algoritmo di esplorazione dell'albero delle soluzioni.
- Si basa sulla separazione dei casi (branching) e sul calcolo di limiti (bounding).
- Prevede **potatura** di rami non promettenti:
 - Se il rilassamento è **inammissibile** \Rightarrow si pota.
 - Se la soluzione è **intera ma peggiore** del best known \Rightarrow si pota.
 - Se il bound è **peggiore** del best known \Rightarrow si pota.
- La **scelta della variabile** da fissare e del **nodo** da esplorare influenzano le prestazioni.
- Il semplice utilizzato in ciascun nodo è solitamente il **primale**, ma in alcuni casi il duale può essere più efficiente.

5.6.2 Cutting Planes (Piani di taglio)

- Si risolve il rilassamento continuo.
- Se la soluzione non è intera, si aggiunge un **taglio** (vincolo) valido per l'insieme intero ma violato dalla soluzione attuale.
- Si ripete fino a ottenere una soluzione intera.
- Il taglio più noto è il **taglio di Gomory**.

5.6.3 Branch and Cut

- Combina B&B e piani di taglio.
- Si generano tagli durante l'esplorazione dell'albero.

5.7 Tagli di Gomory (Gomory Cuts)

- Derivati dall'ultima riga della tabella del simplesso.
- Si prende la parte frazionaria del termine noto e si genera un vincolo che esclude la soluzione non intera.
- I tagli vengono aggiunti iterativamente fino a ottenere una soluzione intera.

5.8 Esempio: problema dello zaino (knapsack)

$$\begin{aligned} \max \quad & \sum_{i=1}^n p_i x_i \\ \text{s.t.} \quad & \sum_{i=1}^n w_i x_i \leq W \\ & x_i \in \{0, 1\} \end{aligned}$$

Interpretazione: Si devono scegliere oggetti da inserire in uno zaino di capacità W massimizzando il valore totale.

5.9 Unimodularità

- Se la matrice dei vincoli è **totalmente unimodulare** e il termine noto \mathbf{b} è intero, allora tutte le soluzioni dei vertici del rilassamento sono intere.
- Questo garantisce una formulazione ideale.

5.10 Conclusioni

La PLI è centrale per numerosi problemi pratici: scheduling, logistica, pianificazione della produzione. Richiede tecniche specifiche rispetto alla PL continua, ed è un campo di intensa ricerca algoritmica.

6. Problema del Trasporto

Il **Problema del Trasporto** è una particolare classe di problemi di Programmazione Lineare caratterizzata da una struttura a rete bipartita. L'obiettivo è determinare il modo più economico per trasportare una merce da un insieme di fornitori a un insieme di clienti, rispettando capacità e richieste.

6.1 Definizione formale

Siano:

- m le origini (depositi, fabbriche);
- n le destinazioni (clienti, magazzini);
- a_i la quantità disponibile nel deposito i ;
- b_j la quantità richiesta dal cliente j ;
- c_{ij} il costo unitario di trasporto dall'origine i alla destinazione j ;
- x_{ij} la quantità trasportata da i a j .

Modello matematico:

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} = a_i \quad \forall i = 1, \dots, m \quad (\text{vincoli di offerta}) \\ & \sum_{i=1}^m x_{ij} = b_j \quad \forall j = 1, \dots, n \quad (\text{vincoli di domanda}) \\ & x_{ij} \geq 0 \quad \forall i, j \end{aligned}$$

6.2 Equilibrio

Il problema è **bilanciato** se:

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

Altrimenti, si aggiunge una **origine o destinazione fittizia** per bilanciare.

6.3 Struttura e proprietà

- Matrice dei vincoli ha struttura speciale: ogni variabile x_{ij} compare in un solo vincolo di riga e uno di colonna.
- Se i dati (a_i, b_j, c_{ij}) sono interi, allora esiste una soluzione ottima intera.
- Il numero massimo di variabili base non nulle in una soluzione di base è $m + n - 1$.

6.4 Metodo di risoluzione

1. **Fase 1:** Calcolo di una **soluzione iniziale ammissibile**:

- Metodo dell'angolo Nord-Ovest
- Metodo di Vogel
- Metodo del costo minimo

2. **Fase 2: Ottimizzazione** tramite Metodo del Potenziale o MODI.

6.5 Metodo del Potenziale (MODI)

- Si assegnano potenziali u_i per le righe (origini) e v_j per le colonne (destinazioni) in modo che:

$$u_i + v_j = c_{ij} \quad \text{per ogni } (i, j) \text{ in base}$$

- Si calcola il **costo ridotto** $c'_{ij} = c_{ij} - u_i - v_j$
- Se tutti i $c'_{ij} \geq 0$, la soluzione è ottima.
- Se esiste un $c'_{ij} < 0$, si modifica la base lungo un ciclo chiuso per migliorare la soluzione.

6.6 Esempio semplice

Dati due depositi con $a_1 = 20$, $a_2 = 30$ e tre clienti con $b_1 = 10$, $b_2 = 25$, $b_3 = 15$ e matrice dei costi:

$$C = \begin{bmatrix} 4 & 8 & 5 \\ 6 & 3 & 7 \end{bmatrix}$$

Si costruisce la soluzione iniziale con Nord-Ovest o Vogel e si ottimizza con MODI.

6.7 Interpretazione economica

- I potenziali u_i, v_j rappresentano costi duali.
- Il problema ha una struttura totalmente unimodulare \Rightarrow soluzione intera.

6.8 Conclusioni

Il problema del trasporto è fondamentale in logistica, supply chain, network flow, ed è alla base di molte estensioni come:

- Problema del trasbordo;
- Problema dell'assegnamento;
- Problema a più fasi (multi-echelon).

7. Problemi di Rete

I **problemi di rete** sono una classe di problemi di ottimizzazione in cui le variabili decisionali rappresentano flussi su archi di un grafo orientato. Sono utilizzati in logistica, trasporti, telecomunicazioni, produzione e project management.

7.1 Struttura generale

Una rete è un grafo orientato $G = (N, A)$, dove:

- N è l'insieme dei nodi (vertici),
- A è l'insieme degli archi (i, j) , ciascuno con:
 - costo c_{ij} ,
 - capacità u_{ij} ,
 - variabile di flusso x_{ij} .

Equilibrio di flusso: ogni nodo ha una domanda/offerta d_i e deve soddisfare:

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = d_i$$

7.2 Problema del Flusso a Costo Minimo

Obiettivo: trovare un flusso di costo minimo che soddisfi la domanda/offerta in ogni nodo.

$$\begin{aligned}
& \min \quad \sum_{(i,j) \in A} c_{ij} x_{ij} \\
& \text{s.t.} \quad \sum_j x_{ij} - \sum_j x_{ji} = d_i \quad \forall i \in N \\
& \quad \quad 0 \leq x_{ij} \leq u_{ij}
\end{aligned}$$

Nota: se $d_i > 0$ allora i è un nodo *fornitore*, se $d_i < 0$ è un nodo *ricevente*.

7.3 Problema del Flusso Massimo

Dato un grafo con nodo sorgente s e nodo pozzo t , si vuole massimizzare il flusso da s a t .

$$\begin{aligned}
& \max \quad \sum_j x_{sj} \\
& \text{s.t.} \quad \sum_j x_{ij} - \sum_j x_{ji} = 0 \quad \forall i \notin \{s, t\} \\
& \quad \quad 0 \leq x_{ij} \leq u_{ij}
\end{aligned}$$

Algoritmo classico: Edmonds-Karp (implementazione del metodo di Ford-Fulkerson con BFS).

7.4 Problema del Cammino Minimo

Dati: un grafo orientato con pesi non negativi c_{ij} e un nodo sorgente s .

Obiettivo: determinare il percorso di costo minimo da s a tutti gli altri nodi.

Algoritmo: Dijkstra (greedy), Bellman-Ford (anche per pesi negativi).

7.5 Problema di Assegnamento

Si tratta di un caso particolare di problema di trasporto con n fornitori e n clienti, dove ogni fornitore deve essere assegnato esattamente a un cliente.

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} = 1 \quad \forall i \\ & \sum_{i=1}^n x_{ij} = 1 \quad \forall j \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

Algoritmo: metodo ungherese.

7.6 Algoritmi generali

- Metodo del cammino aumentante (per flusso massimo)
- Algoritmo di ciclo negativo (per flusso di costo minimo)
- Metodo del simplesso su reti
- Tecniche di scaling

7.7 Proprietà notevoli

- Le matrici di incidenza dei problemi di rete sono totalmente unimodulari.

- Se dati e domande sono interi, anche le soluzioni ottime lo sono.
- La struttura sparsa permette algoritmi più efficienti dei metodi generali di PL.

7.8 Applicazioni

- Ottimizzazione della logistica
- Pianificazione dei trasporti
- Gestione reti elettriche e idriche
- Project management (grafi di attività)
- Allocazione ottima di risorse (task assignment)

8. Geometria della Programmazione Lineare e Teoremi Fondamentali

L'analisi geometrica fornisce un'intuizione potente per comprendere il comportamento dei problemi di Programmazione Lineare (PL), soprattutto in dimensioni basse.

8.1 Spazio delle soluzioni ammissibili

In un problema di PL:

$$\begin{array}{ll} \min \text{ (o max)} & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{b} \end{array}$$

l'insieme delle soluzioni ammissibili è un **poliedro** convesso, ovvero:

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}$$

Definizioni geometriche

- Un **poliedro** è l'intersezione di un numero finito di semispazi.
- Un **vertice** (o punto estremo) è un punto che non può essere scritto come combinazione convessa di altri due punti di \mathcal{P} .
- Una **faccia** è un sottoinsieme di \mathcal{P} che soddisfa un ulteriore vincolo lineare come uguaglianza.

8.2 Funzione obiettivo e iperpiani

La funzione obiettivo $\mathbf{c}^\top \mathbf{x}$ definisce una famiglia di iperpiani paralleli:

$$\mathbf{c}^\top \mathbf{x} = z$$

che traslano nello spazio a seconda del valore z . L'algoritmo del simplesso li muove fino al punto in cui toccano per l'ultima volta \mathcal{P} , cioè in corrispondenza dell'ottimo.

8.3 Teorema dei vertici (o teorema fondamentale della PL)

Enunciato: Se un problema di PL ha una soluzione ottima, allora almeno una delle soluzioni ottime è un **vertice** del poliedro delle soluzioni ammissibili.

Conseguenza: l'algoritmo del simplesso può visitare solo i vertici, ignorando l'interno.

8.4 Esistenza di soluzione ottima

- Se l'insieme ammissibile \mathcal{P} è non vuoto e limitato, l'ottimo esiste.
- Se non è limitato, l'ottimo può non esistere (caso illimitato).
- Se $\mathcal{P} = \emptyset$, il problema è **inammissibile**.

8.5 Relazione con le basi

Una **base** corrisponde a un vertice del politopo: una scelta di m vincoli attivi linearmente indipendenti su n variabili.

Teorema di base-soluzione:

Ogni vertice del politopo ammissibile corrisponde a una soluzione di base (ammissibile). Viceversa, ogni soluzione di base ammissibile è un vertice.

8.6 Dualità geometrica

- Il duale geometrico di un problema massimizzazione è la minimizzazione di un supporto sull'insieme dei piani che contengono il politopo.
- Le soluzioni duali corrispondono a iperpiani di supporto al politopo.

8.7 Degenerazione

Un vertice è **degenere** se più di n vincoli sono attivi. In tal caso, la base non è univoca. Si può avere ciclicità nel simplesso.

8.8 Visualizzazione bidimensionale

- In \mathbb{R}^2 , il poliedro è un poligono.
- La funzione obiettivo è una retta mobile.
- La soluzione ottima è a un vertice toccato per ultimo da tale retta.

8.9 Interpretazione economica

La geometria aiuta a visualizzare:

- Prezzi ombra (derivati dai gradienti dei vincoli attivi)
- Elasticità delle risorse (modifica dei vertici)
- Effetti dei tagli (intersezione aggiuntiva con il politopo)

9. Conclusioni e Strategia per la Preparazione all'Esame

In questa dispensa abbiamo affrontato in modo sistematico i principali argomenti della programmazione lineare, della programmazione intera e delle tecniche risolutive più diffuse nell'ambito dell'ottimizzazione. Lo scopo è fornire una base solida per sostenere con successo un esame universitario di Decision Science.

9.1 Riepilogo degli argomenti trattati

- Modelli di ottimizzazione e formulazione matematica
- Programmazione lineare: forme standard e canoniche
- Metodo del simpleso: geometria, basi, ottimalità, degenerazione
- Metodo delle due fasi e Big M
- Dualità: formulazione, interpretazione, teoremi fondamentali
- Programmazione intera: rilassamenti, formulazioni ideali
- Branch and Bound: potatura, strategie di branching

- Cutting Planes: tagli di Gomory, tagli validi
- Problemi di trasporto e di rete: flussi, cammini, assegnamenti
- Teoremi geometrici e fondamentali della PL
- Esempi svolti e spiegati

9.2 Suggerimenti per lo studio

1. Parti dai fondamentali: assicurati di comprendere bene modelli, formulazioni e significato dei vincoli.
2. Apprendi con la geometria: la visualizzazione di politopi e iperpiani aiuta a interiorizzare concetti.
3. Risolvi esercizi a mano: il simplesso, la dualità e il B&B vanno allenati anche manualmente.
4. Rivedi dimostrazioni chiave: dualità forte/debole, condizioni di ottimalità, formulazione del duale.
5. Confronta primale e duale: costruisci le corrispondenze per apprendere simmetrie e intuizioni.
6. Impara a motivare algoritmi: il simplesso duale si usa in presenza di ammissibilità duale, non primale.
7. Riconosci pattern: molti problemi applicativi (zaino, trasporto, rete) si riconducono a strutture note.

9.3 In sede d'esame

- Preparati a esporre in modo chiaro **definizioni formali**, **algoritmi** e **dimostrazioni**.
- Allenati con **esempi numerici** completi, dall'inizio alla fine, per ciascun metodo.
- Sii in grado di passare da formulazioni testuali a modelli matematici completi.
- Rivedi bene l'uso delle **basi**, l'effetto dei **tagli**, la scelta dei **nodi** e delle **variabili** in B&B.
- Conosci le differenze e gli usi del **simplesso primale** e **duale**, e del metodo delle due fasi.

Buono studio!

Questa dispensa, strutturata come una guida organica, è pensata per accompagnarti nella comprensione teorica e nella pratica risolutiva. Con padronanza dei concetti esposti, sarai pronto ad affrontare l'esame con sicurezza e successo.

10. Esempi Svolti e Commentati

In questa sezione presentiamo alcuni esempi completi che illustrano l'applicazione pratica dei metodi discussi nei capitoli precedenti. Ogni esempio è corredato da spiegazioni, passaggi dell'algoritmo e interpretazioni.

10.1 Esempio 1: Problema di produzione

Testo: Un'azienda produce due prodotti x_1 e x_2 che generano rispettivamente profitti unitari pari a 6 e 5. I vincoli di risorse sono:

$$\begin{aligned}x_1 + x_2 &\leq 5 && \text{(tempo disponibile)} \\3x_1 + 2x_2 &\leq 12 && \text{(materie prime)} \\x_1, x_2 &\geq 0\end{aligned}$$

Obiettivo: Massimizzare il profitto:

$$\max z = 6x_1 + 5x_2$$

Passaggi:

1. Introduzione delle variabili slack s_1, s_2 :

$$\begin{aligned}x_1 + x_2 + s_1 &= 5 \\3x_1 + 2x_2 + s_2 &= 12\end{aligned}$$

2. Costruzione della tabella iniziale del simplesso con s_1, s_2 in base.
3. Calcolo dei costi ridotti, scelta della variabile entrante e uscente.

4. Iterazioni successive fino all'ottimalità.

Soluzione ottima:

$$x_1^* = 2, \quad x_2^* = 3, \quad z^* = 6(2) + 5(3) = 27$$

10.2 Esempio 2: PLI con Branch and Bound

Testo: Si consideri il problema:

$$\begin{aligned} \max \quad & z = 3x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 4 \\ & 2x_1 + x_2 \leq 5 \\ & x_1, x_2 \in \mathbb{Z}_+ \end{aligned}$$

Procedura:

- Risolviamo il rilassamento continuo con il simplesso: otteniamo $x_1 = 1.5$, $x_2 = 2.5$, $z = 8.5$.
- Branch su x_2 : creiamo due nodi $x_2 \leq 2$ e $x_2 \geq 3$.
- Ripetiamo la procedura per ciascun ramo.
- Confronto dei valori interi ottenuti: selezioniamo il massimo.

Soluzione intera ottima: $x_1 = 2$, $x_2 = 2$, $z = 10$

10.3 Esempio 3: Dualità

Primale:

$$\begin{aligned} \max \quad & 4x_1 + 3x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 \leq 8 \\ & x_1 + 2x_2 \leq 6 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Dualità:

$$\begin{array}{ll}\min & 8y_1 + 6y_2 \\ \text{s.t.} & 2y_1 + y_2 \geq 4 \\ & y_1 + 2y_2 \geq 3 \\ & y_1, y_2 \geq 0\end{array}$$

Soluzioni ottime: Entrambe hanno valore ottimo $z = 18$

Osservazioni:

- Il valore della funzione obiettivo coincide (dualità forte).
- I vincoli attivi del primale corrispondono a variabili duali positive.

10.4 Esempio 4: Tagli di Gomory

Risoluzione di un problema intero mediante tagli:

$$\begin{array}{ll}\max & 5x_1 + 4x_2 \\ \text{s.t.} & 6x_1 + 4x_2 \leq 24 \\ & x_1 + 2x_2 \leq 6 \\ & x_1, x_2 \in \mathbb{Z}_+\end{array}$$

Step:

1. Si risolve il rilassamento continuo \rightarrow soluzione frazionaria.
2. Si costruisce un taglio di Gomory da una riga della tabella.
3. Si aggiunge il taglio, si ricalcola con simpleso duale.
4. Si itera fino a soluzione intera.

Risultato finale: $x_1 = 2, x_2 = 2, z = 18$

10.5 Esempio 5: Metodo delle due fasi

Problema originale:

$$\begin{array}{ll}\min & x_1 + x_2 \\ \text{s.t.} & x_1 - x_2 = 1 \\ & x_1, x_2 \geq 0\end{array}$$

Fase 1:

- Aggiungiamo variabile artificiale a_1 alla prima equazione.
- Nuova funzione obiettivo: $\min a_1$

Fase 2:

- Se $a_1 = 0$ all'ottimo della fase 1, si elimina e si riprende il problema originale.

Risultato: $x_1 = 1, x_2 = 0$, valore ottimo = 1