Ottimizzazione Combinatoria
Appunti

Giovanni "Qua' Qua' dancer" Palma e Alex Basta

Contents

Chapter 1

Introduzione

Prova scritta e orale. Si studiano metodi algoritmici per ottimizzare problemi di flusso su reti e di programmazione lineare. In poche parole, impariamo come prendere decisioni. Appunti presi in base alle lezioni del prof Ugo Dal Lago, si consiglia una lettura con accetto veneto
Questa introduzione è totalmente inutile, serve solo ad occupare byte. Ringraziate pure il bastianini per questo

Chapter 2

Problemi e Modelli

2.1 Problemi di ottimizzazione

Definition 2.1.1: Ricerca operativa

La **ricerca operativa** è un ramo della matematica applicata che si occupa dello studio, della modellizzazione e della risoluzione dei cosiddetti *problemi decisionali* complessi mediante strumenti matematici, algoritmici e computazionali, con l'obiettivo di ottimizzare processi e risorse

Per evitare qualsivoglia fraintendimento fornirò anche la definizione di **ottimizzazione Combinatoria**

Definition 2.1.2: Ottimizzazione Combinatoria

Si definisce **Ottimizzazione Combinatoria** una branca della Ricerca Operativa che nel modellare matematicamente e risolvere problemi complessi di natura discreta unisce tecniche di calcolo combinatorio alla teoria degli algoritmi e ai risultati teorici e metodologici della programmazione lineare

Pertanto ricerca operativa e ottimizzazione combinatoria sono due cose diverse, MA cito testualmente

> "Per tutti i nostri scopi ricerca operativa e ottimizzazione, sono sinonimi
>
> tuttavia non vedremo solo alcune tecniche di ottimizzazione combinatoria, ma anche altre tecniche che stanno nella ricerca operativa ma che trattano di valori non discreti"
>
> – Ugo

Adesso, sotterrato questo problema di carattere unicamente terminologico con cui io non posso fare a meno di strizzarmi il cervello perché c'ho l'autismo, possiamo tornare a parlare di ricerca operativa/ottimizzazione combinatoria (tanto so' sinonimi per noi)
I problemi di cui si occupa la ricerca operativa, quindi, riguardano situazioni in cui occorra massimizzare i ricavi o minimizzare i costi, in presenza di risorse limitate. Detto in termini più matematici, data una funzione **vincolata** l'obiettivo è trovare una soluzione ottimale che massimizzi o minimizzi tale funzione.
È pertanto vero, quindi, che questa disciplina ha forte contenuto economico
La ricerca operativa si inserisce all'interno del processo decisionale, il quale può essere suddiviso in diverse fasi

- **Individuazione problema**
- **Raccolta dati**
- **Costruzione modello**, ovvero la Traduzione del problema in un modello matematico che descriva il sistema e i vincoli in modo formale
- **Determinazione di piu' soluzioni**: applicazione di algoritmi e tecniche di ottimizzazione per individuare la soluzione migliore
- **Analisi dei risultati**

La ricerca operativa, quindi, si occupa delle fasi 3 e 4 del processo, dato che sono le fasi che richiedono l'impiego di modelli matematici, algoritmi di ottimizzazione e strumenti computazionali. Adesso andiamo a definire per benino che cosa intendiamo per "modello"

Definition 2.1.3: modello

un **modello** è una descrizione astratta e scritta in linguaggio matematico, della parte di realtà utile al processo decisionale

I modelli ci permettono di inquadrare i problemi in una determinata "cornice" che ci permette di determinare quale tipo di algoritmo risolutivo usare.
Esistono tre tipi di modelli:

- **Teoria dei giochi**: ricerca di un equilibrio fra le componenti coinvolte in un'interazione reciproca, spesso con obbiettivi contrastanti. (non ce ne occupiamo)
- **Simulazione**: il problema viene studiato simulando la situazione senza studiarne la natura in modo analitico tramite generazione di istanze casuali. (anche questi modelli non ci interessano)
- **Analitici**: dal problema si costruisce un modello matematico rigoroso (senza perdere informazione sul problema reale) e risolto mediante tecniche analitiche, senza ricorrere a simulazioni. La natura stessa dello spazio matematico in cui è inserito il problema è in grado di garantire la soluzione ottima. Questo tipo approccio è particolarmente vantaggioso in quanto assicura l'esattezza della soluzione supponendo che il modello sia formulato correttamente.

  È tuttavia richiesto un discreto livello di creatività

Definiamo, adesso, i problemi che andiamo a trattare

Definition 2.1.4: Problema

Definiamo **problema** una domanda, espressa in termini generali, la cui risposta dipende da *parametri* e *variabili*, sopratutto nei problemi analitici

Un problema $\mathcal{P}$ è descritto tramite:

- I suoi parametri e variabili
- Le caratteristiche che una soluzione deve avere

Quando fissiamo un'istanza di un problema, vengono fissati i parametri ma non le variabili, che sono le incognite che devono essere definite. Distinguiamo un problema dalla sua istanza per generalizzarlo. Si presti attenzione alla differenza tra parametri e variabili che molti si confondono

Example 2.1.1 (Problema con paramteri e variabili)

Sia $\mathcal{P}$ il seguente problema

$$ax^2 + bx + c = 0$$

Dove a, b e c sono i suoi parametri e x rappresenta le variabili, una possibile istanza di tale problema è:

$$5x^2 - 6x + 1 = 0$$

Un modo comune per descrivere un problema è dare l'insieme di soluzioni ammissibili $\mathbb{F}_{\mathcal{P}} \subseteq G$, dove G è un sovrainsieme generico noto, di solito contenente la collezione di tutte le possibili configurazioni o decisioni che si possono prendere, dando dei vincoli che un generico $g \in G$ deve soddisfare per far parte di $\mathbb{F}_{\mathcal{P}}$, avremo così che $G - \mathbb{F}_{\mathcal{P}}$ è l'insieme delle soluzioni non ammissibili

Example 2.1.2

Sia l'instanza di $\mathcal{P}$ definita precedentemente

$$5x^s - 6x + 1 = 0$$

si ha che

$$\mathbb{G} = \mathbb{R}$$
$$\mathbb{F}_{\mathcal{P}} = \{x \in \mathbb{R} | 5x^2 - 6x + 1 = 0\}$$

2.1.1 2.1.1 Problemi di ottimizzazione

Iniziamo con una definizione preliminare.

Definition 2.1.5: Problema di ottimizzazione

In matematica e in informatica, un problema di ottimizzazione è il problema di trovare la migliore soluzione fra tutte le soluzioni fattibili.

Un problema di ottimizzazione $\mathcal{P}$ viene descritto:

- Dando l'insieme $\mathbb{F}_{\mathcal{P}}$ delle soluzioni ammissibili
- Specificando una funzione obiettivo $c_{\mathcal{P}} : \mathbb{F}_{\mathcal{P}} \to \mathbb{R}$ che assegna ad ogni $g \in \mathbb{F}_{\mathcal{P}}$ un valore reale $c_{\mathcal{P}}(g)$, che rappresenta il costo o il beneficio.

Un problema (di ottimizzazione) di massimo $\mathcal{P}$ consiste nel determinare il valore

$$Z_{\mathcal{P}} = \max\{c_{\mathcal{P}}(g) \mid g \in \mathbb{F}_{\mathcal{P}}\},$$

mentre un problema (di ottimizzazione) di minimo $\mathcal{P}$ consiste nel determinare il valore

$$Z_{\mathcal{P}} = \min\{c_{\mathcal{P}}(g) \mid g \in \mathbb{F}_{\mathcal{P}}\}.$$

Ci si può trastullare con la definizione, infatti ad ogni problema di massimo $\mathcal{P}$ corrisponde un problema di minimo $\mathcal{P}'$ tale che $c_{\mathcal{P}'}(g) = -c_{\mathcal{P}}(g)$, ovvero:

$$Z_{\mathcal{P}} = -\min\{c_{\mathcal{P}'}(g) \mid g \in \mathbb{F}_{\mathcal{P}} = \mathbb{F}_{\mathcal{P}'}\}.$$

Definition 2.1.6: Valore ottimo e soluzione ottima

Dato un problema di ottimizzazione $\mathcal{P}$, il valore $Z_{\mathcal{P}}$ definito in precedenza è detto valore ottimo, mentre il $g^* \in \mathbb{F}_{\mathcal{P}}$ tale che $Z_{\mathcal{P}} = c_{\mathcal{P}}(g^*)$ è detto soluzione ottima.

Si può quindi constatare che la differenza reale tra i due è che il valore ottimo è inserito nel codominio della funzione obiettivo (cioè il mero valore reale "ottimizato"), mentre la soluzione ottima appartiene al dominio (ovvero è l'elemento in $\mathbb{F}_{\mathcal{P}}$ che ottimizza la funzione).

Example 2.1.3 (Esempio di problema di ottimizzazione)

Dati

$$G = \mathbb{R}, \quad \mathbb{F}_{\mathcal{P}} = \{x \in \mathbb{R} \mid 5x^2 - 6x + 1 = 0\}, \quad c_{\mathcal{P}} : \mathbb{R} \to \mathbb{R} \text{ con } c_{\mathcal{P}}(g) = g^2,$$

e si ponga

$$Z_{\mathcal{P}} = \max\{x^2 \mid x \in \mathbb{F}_{\mathcal{P}}\}.$$

Innanzitutto, calcolo l'insieme delle soluzioni ammissibili (hold my soluzione parabolica):

$$x = \frac{6 \pm \sqrt{(-6)^2 - 4 \cdot 5 \cdot 1}}{2 \cdot 5} = \frac{6 \pm \sqrt{36 - 20}}{10} = \frac{6 \pm \sqrt{16}}{10} = \frac{6 \pm 4}{10}.$$

Quindi, le soluzioni sono:

$$x_1 = \frac{6+4}{10} = 1 \quad \text{e} \quad x_2 = \frac{6-4}{10} = \frac{1}{5}.$$

Pertanto, l'insieme delle soluzioni ammissibili è:

$$\mathbb{F}_{\mathcal{P}} = \{1, \tfrac{1}{5}\}.$$

Si procede poi nel calcolare la funzione obiettivo per ogni elemento:

$$c_{\mathcal{P}}(1) = 1^2 = 1, \quad c_{\mathcal{P}}\left(\tfrac{1}{5}\right) = \left(\tfrac{1}{5}\right)^2 = \tfrac{1}{25}.$$

Il massimo tra questi valori è:

$$Z_{\mathcal{P}} = \max\{1, \tfrac{1}{25}\} = 1.$$

fin casi dei problemi di decisione
Si hanno quattro casi principali in cui sono inseriti i problemi di decisione:

- **Problema vuoto:** non esistono soluzioni ammissibili, ovvero $\mathbb{F}_{\mathcal{P}} = \emptyset$, per cui l'ottimizzazione è impossibile e si assume che $Z_{\mathcal{P}} = \infty$.
- **Problema illimitato:** si ha quando non esiste un limite inferiore/superiore per i valori della funzione obiettivo tra le soluzioni ammissibili, ovvero quando $Z_{\mathcal{P}} = \pm\infty$. Ad esempio, nel caso del massimo, per ogni $x \in \mathbb{R}$ esiste un $g \in \mathbb{F}_{\mathcal{P}}$ con $c_{\mathcal{P}}(g) \geq x$, e quindi $Z_{\mathcal{P}} = +\infty$.
- **Valore ottimo finito ma non soluzione ottima finita:** un problema di ottimizzazione può presentare un valore ottimo finito, pur non ammettendo alcuna soluzione $g \in \mathbb{F}_{\mathcal{P}}$ tale che $c_{\mathcal{P}}(g)$ sia esattamente uguale a $Z_{\mathcal{P}}$. (Es.: nell'insieme $\{x \mid x > 0\}$, dove l'estremo inferiore è 0, ma non esiste una soluzione che raggiunga tale valore.)
- **Valore ottimo finito e soluzione ottima finita:** esiste almeno un $g \in \mathbb{F}_{\mathcal{P}}$ tale che $c_{\mathcal{P}}(g) = Z_{\mathcal{P}}$, ed esso è ottimo. Notare che possono esistere più soluzioni ottime, ma un solo valore ottimo.

2.1.2 Problemi di ottimizzazione e di decisione

Verranno fornite due definizioni importanti:

Definition 2.1.7: Problema di decisione

Un problema di decisione consiste nel determinare una qualunque soluzione ammissibile $g \in \mathbb{F}_{\mathcal{P}}$.

Definition 2.1.8: Problema di certificato

Il problema di certificato consiste nel verificare se, per un dato $g \in G$, risulti che $g \in \mathbb{F}_{\mathcal{P}}$.

I problemi di decisione sono generalmente più semplici dei problemi di ottimizzazione, in quanto non richiedono la definizione di una funzione obiettivo $c_{\mathcal{P}}()$. Tuttavia, è possibile trasformarli in problemi di ottimizzazione fissando una funzione obiettivo triviale (ad esempio, costante), per cui tutte le soluzioni ammissibili risultano ottime.
In alternativa, possiamo considerare un problema decisionale R definito da

$$F_R = \{g \in \mathbb{F}_{\mathcal{P}} \mid c_{\mathcal{P}}(g) = Z_{\mathcal{P}}\},$$

oppure, per un dato $k \in \mathbb{R}$, il problema decisionale R_k con

$$F_{R_k} = \{g \in \mathbb{F}_{\mathcal{P}} \mid c_{\mathcal{P}}(g) \leq k\},$$

nel caso in cui $\mathcal{P}$ sia un problema di minimo.

2.1.3 Aspetto algoritmico

- Algoritmi esatti: e' un algoritmo ch epreso un'istanza di P (P e' un modello di un problema), fornisce in output una soluzione ottima g^* di P (se esiste). Spesso pero' i problemi sono troppo complessi ed e' impossibile costruire algoritmi efficenti.
- Algoritmi euristici: non ci danno garanzia sulla soluzione trovata (e' sicuramente ammissibile), ma un'approssimazione.

Come possiamo valutare la correttezza di una soluzione euristica? Possiamo misurare l'errore (assoluto o relativo) fra il valore ottimo euristico e quello esatto.
Gli algoritmi euristici vengono detti anche greedy.

2.2 Modelli

Al posto di dare un'algoritmo per ogni specifico problema, possiamo definire classi di problemi che possono essere risolti con lo stesso algoritmo.

2.2.1 Programmazione Lineare

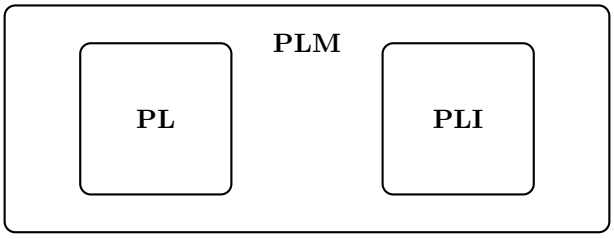
Fornisco Innanzitutto la def. di problema di problema di Programmazione Lineare

Definition 2.2.1: Problema di Programmazione Lineare

Si definiscono **Problemi di programmazione lineare (PL)** tutti quei problemi di ottimizzazione in cui la funzione obbiettivo $c_{\mathcal{P}}$ è *lineare* e i vincoli sono *tutti espressi da disequazioni lineari* ed anche, eventualmente, equazioni lineari (quest'ultime possono mancare le prime no).
In particolare in un problema $\mathcal{P}$ di programmazione lineare si ha:

- $\mathbb{G} = \mathbb{R}$, definendo poi un numero finito $n \in \mathbb{N}$ di variabili reali, che nella realtà rappresentano delle *quantità*
$$x = (x_1, \dots, x_n) \in \mathbb{RN}$$
- Una *funzione obiettivo* $c_{\mathcal{P}}$ definita $f : \mathbb{R}^n \to \mathbb{R}$ nella forma:
$$f(x) = cx$$
Dove c è un vettore riga e x è un vettore colonna. Si noti che c non è una variabile, bensì un *parametro* (definizione della funzione obiettivo)
- Un insieme di m vincoli lineari, tutti in una delle forme seguenti:
$$ax = b \quad ax \leqslant b \quad ax \geqslant b$$
Dove $a \in \mathbb{R}^n$ e $b \in \mathbb{R}$

f e' lineare. c_p e' l'insieme di vettori di numeri reali. $\mathbb{G} = \mathbb{R}^n$. (definizione di G)
È talvolta utile assumere che $x \in \mathbb{Z}^n$, ovvero che le soluzioni ammissibili siano vettori di numeri interi. In questo caso si parla di *programmazione lineare intera*. Facendo così, stiamo restringendo il campo di ricerca (ed il dominio delle soluzioni possibili), ma si perdono alcune proprietà (geometriche) che in realtà possono rendere più difficile la ricerca della soluzione.
Esiste inoltre la *programmazione lineare mista* (PLM), dove si hanno variabili di natura mista (alcune variabili in $\mathbb{R}$ ed alcune in $\mathbb{Z}$)
Si noti questo diagramma riassuntivo:



Un problema PL può sempre essere espresso nella seguente forma matriciale:

$$\max\{cx \mid Ax \leqslant b\}$$

dove $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$. Grazie ad alcuni accorgimenti è possibile, inoltre, scrivere tutti i vincoli possibili di un problema di PL $\mathcal{P}$ in un'unico sistema di disequazioni lineari. Infatti:

- Se $\mathcal{P}$è un problema di minimo, occorre considerare semplicemente la funzione $f(x) = (-c)x$

- Ogni vincolo $ax = b$diventa la coppia di vincoli $ax \leqslant b$ e $ax \geqslant b$
- Ogni vincolo $ax \geqslant b$ è equivalente a $(-a)x \leqslant (-b)$.

2.2.2 Programmazione lineare intera

Se nella *PL* le variabili rappresentano quantità nella PLI le variabili possono essere

- **Quantitative**: ovvero rappresentare quantità.
- **Logiche**: rappresentare valori booleani, ovvero scelte basate sulla possibilità di scegliere o meno una determinata opzione

Qui la definizione di variabile logica

Definition 2.2.2: variabile logica

Una **variabile** x si definisce **logica** se:

$$x \in \mathbb{N}(o\mathbb{Z}) \quad 0 \leqslant x \quad x \leqslant 1$$

Un esempio tipico è rappresentato dalle variabili che associano una determinata risorsa a un compito specifico, oppure che determinano l'utilizzo di un particolare processo.

> Questo e' il bello dei problemi lineari
>
> – Il Basta

Un giga esempietto bellino è il problema dello zaino:

Example 2.2.1 (Problema dello zaino)

Il problema dello zaino è un esempio classico di problema di ottimizzazione combinatoria piuttosto complesso.
TODO: inserire il testo del problema

Parametri:

$$\begin{aligned} E &= \{1, \dots, n\} \\ a_i &= \text{peso dell'oggetto } i \\ c_i &= \text{costo dell'oggetto } i \\ b &= \text{peso massimo dello zaino} \end{aligned}$$

Variabili:

$$x_i = \begin{cases} 1 & \text{se l'oggetto } i \text{ è nello zaino} \\ 0 & \text{altrimenti} \end{cases}$$
$$x_i \in \{0, 1\} \quad \forall i \in \{1, \dots, n\}$$

Vincoli:

$$\sum_{i=1}^{n} x_i a_i \leqslant b$$

Funzione obiettivo:

$$\max \sum_{i=1}^{n} x_i c_i$$

Relazioni logiche

Avendo introdotto le variabili logiche, è opportuno chiedersi se sia possibile esprimere, mediante programmazione lineare (PL), le regole di inferenza logica per le relazioni che intercorrono tra di esse. Grazie all'uso di opportuni vincoli lineari, la risposta è affermativa:

- **Negazione** ($y = \neg x$):
$$x = 1 - y$$
- **Congiunzione** ($z = x \wedge y$):
$$\begin{aligned} z &\leqslant x \\ z &\leqslant y \\ z &\geqslant x + y - 1 \end{aligned}$$
- **Disgiunzione** ($z = x \vee y$):
$$\begin{aligned} z &\geqslant x \\ z &\geqslant y \\ z &\leqslant x + y \end{aligned}$$
- **Implicazione** ($z = x \implies y$):
$$\begin{aligned} x + z &\geqslant 1 \\ z &\geqslant y \\ x + z &\leqslant 1 + y \end{aligned}$$

È possibile, tuttavia, dimostrare che il problema di ottimizzazione lineare è, in generale, *NP-hard*, dato che il problema di soddisfacibilità di una formula logica rientra nella classe NP-hard

2.3 Template per risolvere problemi

2.3.1 Vincoli di assegnamento

Un tipo di vincolo che la programmazione lineare intera (PLI) gestisce in modo molto efficace è rappresentato dai cosiddetti *vincoli di assegnamento*. Questi vincoli sono utilizzati per modellare problemi che riguardano l'assegnazione di "oggetti a luoghi".
Si considerano:

- Un insieme $N = \{1, \ldots, n\}$ di **oggetti** (che possono rappresentare automezzi, persone, ecc.)
- Un insieme $V = \{1, \ldots, m\}$ di **luoghi**

Per rappresentare le diverse condizioni di assegnazione degli oggetti ai luoghi, si introduce la variabile $x_{ij} \in \{0, 1\}$ (dove $1 \leqslant i \leqslant n$ e $1 \leqslant j \leqslant m$). Questa variabile indica se l'i-esimo oggetto è stato assegnato al j-esimo luogo ($x_{ij} = 1$) o meno ($x_{ij} = 0$), si può così procedere alla formale

Definition 2.3.1: Vincoli di assegnamento

Sia $N = \{1, \ldots, n\}$ un insieme di oggetti e $V = \{1, \ldots, m\}$ di luoghi, si definiscono **vincoli di assegnamento** quei vincoli che impongono che *ogni oggetto sia assegnato a un solo luogo e ad ogni luogo è assegnato esattamente un oggetto.* Questi sono espressi mediante queste sommatorie:

$$\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i \in \{1, \ldots, n\} \qquad \sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \in \{1, \ldots, m\}$$

Dove:

- x_{ij} è una variabile binaria che vale 1 se l'oggetto i è assegnato al luogo j, e 0 altrimenti.
- La prima equazione assicura che ogni oggetto i sia assegnato esattamente a un luogo.
- La seconda equazione assicura che ogni luogo j riceva esattamente un oggetto.

il luogo può anche essere visto come uno slot temporale, quindi imponiamo un certo pipeline

Vincoli di semi-assegnamento

I vincoli di semi-assegnamento sono una variante dei vincoli di assegnamento in cui non è necessario che ogni luogo riceva esattamente un oggetto, in particole si noti la seguente definizione

Definition 2.3.2: Vincoli di semi-assegnamento

Sia $N = \{1, \ldots, n\}$ un insieme di oggetti e $V = \{1, \ldots, m\}$ un insieme di luoghi. Si definiscono **vincoli di semi-assegnamento** quei vincoli che impongono che *ogni oggetto sia assegnato ad al massimo un luogo.* Questi vincoli sono espressi mediante la seguente sommatoria:

$$\sum_{j=1}^{m} x_{ij} \leqslant 1 \quad \forall i \in \{1, \ldots, n\}$$

Dove x_{ij} è una variabile binaria che vale 1 se l'oggetto i è assegnato al luogo j, e 0 altrimenti.

Insiemi ammissibili:
Vincoli di Assegnamento: ogni oggetto e' assegnato ad un luogo e ogni luogo ha un oggetto

2.3.2 Selezione di Sottoinsiemi

Sia $N = \{1, ..., n\}$ un insieme finito di elementi e sia $F = \{F_1, ..., F_n\}$ una famiglia di sottoinsiemi (non necessariamente disgiunti) dove $F_i \subseteq N$. Vogliamo decidere qual'e' la scelta migliore di $D \subseteq F$ di costo minimo. Per aiutare nella formulatzione del problema, possiamo rappresentare l'appartenenza degli ielementi di N in un sottoinsieme F_j come una grande matrice che ha come righe gli elementi e come colonne i sottoinsiemi. La scelta e' catturatanon piu' da x_{ij} ma solo da x_j dato che basta dire se il sottoinsieme j lo volgiamo o no.

$$x_j = \begin{cases} 1 & F_j \in D \\ 0 & \text{altrimenti} \end{cases}$$

La funzione obbiettivo e' sempre:

$$\sum_{j=1}^{n} x_j c_j$$

I vincoli dipendono dal problema:

- Problema di copertura: ognuno degli eventi di N atanno in almeno uno degli elementi di D:

$$\sum_{j=1}^{m} a_{ij} x_j \geqslant$$

Numero di sottoinsiemi di D che appartengono a i (?). I sottoinsiemi F di N possono essere dei curriculum dove c'e' scritto quale dei linguaggi di programmazione $n \in N$ conosce un candidato. Ogni candidato ha anche uno stipendio. Dobbiamo scegliere quali canditati assumere per coprire tutti i linguaggi in N minimizzando il costo.

- Partizione: D deve essere una partizione di N. N possono essere dei task da svolgere e gli iniemi di F sono offerte da fornitori rispetto alla risoluzione di alcuni dei task. Quindi gli inisiemi di D devono essere disgiunti perche' non vogliamo che due societa' risolvano lo stesso task. Tutto deve essere coperto una sola volta.
- Riempimento: si usa solo quando si vuole massimizzare. N non sono piu' incombenze o task, ma risorse da usare una volta per costruire un prodotto (elemento di F). Al piu' perche' possiamo scartare qualche componente. Possiamo usare ogni elemento di N al massimo una volta.

Example 2.3.1 (Problema delle Commesse)

Un'agenzia deve decidere come impiegare i soi n dipendenti.
L'azienda, nell'intervallo di tempo considerato, deve evadere m commese.
Ciascuna commessa j deve essere svolta dal sottoinsieme $F_j \subseteq \{1, ..., n\}$ dei dipendenti dell'azienda.
Ogni commesssa, se evase, darebbe luogo ad un ricavo pari a r_j.
Ogni dipendente puo' lavorare ad una singola commessa nell'unita' di tempo.
SVOLGIMENTO:
e' un problema di selezione di sottoinsiemi:

$$N - \{1, ..., n\}\text{elementi/dipendenti}$$

$$F = \{F_1, ..., F_m\} commesse$$

VARIABILI:

$$x_j = \begin{cases} 1 & F_j \in D_j \\ 0 & \text{altrimenti} \end{cases}$$

VINCOLI: $0 \leqslant x_j \leqslant 1 \forall j$

$$\sum_{j=1}^{m} x_j a_{ij} \leqslant 1 \forall i$$

FO:

$$\sum_{j=1}^{m} x_j r_j$$

Ha fatto poi un esempio di quando c'e' una penalita' quando non fai una commessa (mi sa)

2.3.3 Ricondurci a problemi linari

Variabili a valori discreti possono prendere un insieme finito di valori diversi che non sono valori in un intervallo. Se questo insieme ha n valori ci servono n variabili diverse.

Example 2.3.2 (Progetto di reti)

alcune slide non le ho fatte
valore assoluto:

$$max\{f(x)|x \in X\} \quad max\{-f(x)|x \in X\}$$

Abbiamo ridotto il problema che ha un valore assoluto a uno che non ce l'ha. Se ci fosse stata un operazione fra valori assoluti, la situa si complica
Funzioni lineari a tratti

in due dimensioni hanno la forma:

$$f(x) = \begin{cases} b_1 + c_1 x & x \in [a_1, a_2] \\ b_2 + c_2 x & x \in (a_2, a_3] \end{cases}$$

nel punto a_2 ci puo' essere un punto di discontinuita'.
Come per il carico fisso vengono introdotte due variabili logiche asusiliarie y_1, y_2 co nil seguente significato: Inoltre ci servono variabili ausiliarie che ci dicono lo spostamento rispetto a un estremo dell'intervallo a cui appartiene x: Usiamo gli intervalli dinamici. Non c'e' niente che ci da la funzione da ottimizzare, abbiamo solo delle definizioni che sono dei semplici commenti, i vincoli lineari sono solo gli ultimi, ma non bastano.
Rappresentiamo la funzione f con una funzione g che prende le variabili logiche e quantitative. Il valore di f e' rappresentato univocamente dalla quadrupla, eccetto nel punto di non continuita'. Solo il primo valore e' accettabile perche' l'intervallo li e' chiuso.
GNU MathProf

Chapter 3

Esercitazioni

3.1 Problemi e modelli

3.1.1 Problema dei data center

Testo

Abbiamo n server $1, \dots, n$, ogni server può lavorare in $m_{i \in \mathbb{N}}$ modalità operative diverse. Nella modalità $j \in \{1, \dots, m\}$, il server 1 riesce ad eseguire un numero di istruzioni..

Svolgimento

Variabili:

$$x_{ij} = \begin{cases} 1 & \text{se il server } i \text{ è utilizzato in modalità } j \in \{1, \dots, m_i\} \\ 0 & \text{altrimenti} \end{cases}$$

$$x_{ij} \in \mathbb{N}, \quad \forall i \, \forall j$$

Vincoli: $\forall i \in \{1, \dots, n\}, \quad \forall j \in \{1, \dots, m_i\}$ con $0 \leqslant x_{ij} \leqslant 1$ si ha:

$$\forall i \in \{1, \dots, n\} \quad \sum_{j=1}^{m_i} x_{ij} = 1$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{ij} s_{ij} \geqslant k$$

funzione obbiettivo

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{ij} w_{ij}$$

3.1.2 Problema del docente di informatica

Testo

Si hanno dei progetti $(t_1, t_2, \dots, t_n)$, si hanno m PC, dove ogni pc può compilare qualunque progetto in modalità sequenziale. Le prestazioni del PC sono identiche
Vogliamo assegnare i progetti ai pc in modo da minimizzare il tempo complessivo parallelo di compilazione

Svolgimento

Variabili:

$$x_{ij} = \begin{cases} 1 & \text{se il progetto } i \text{ è compilato al pc } j \\ 0 & \text{altrimenti} \end{cases}$$

$$x_{ij} \in \mathbb{N}, \quad \forall i \in \{1, \dots, n\}\, \forall j \in \{1, \dots, m\}$$

Vincoli:

$$\forall i \in \{1, \dots, n\} \quad \sum_{j=1}^{m} x_{ij} = 1 \text{ (normale vincolo di semi-assegnamento!!)}$$

$$\forall j \in \{1, \dots, m\} \quad y \geqslant \sum_{i=1}^{n} x_{ij} t_i$$

Funzione obbiettivo:

$$\min\Big(\underbrace{\max_j \sum_{i=1}^{n} \underbrace{x_{ij} t_i}_{\text{tempo di compilazione del progetto } i \text{ al pc } j}}_{\text{nonostante questa espressione matematica ha perfettamente senso, non è lineare!}} \Big)$$

Funzione obbiettivo sbagliata! Pertanto prenderemo y per "simulare" il massimo, di modo da rendere la funzione obbiettivo lineare:

$$\min y$$

3.1.3 Problema delle commesse

Testo

Un'azienda deve decidere come impiegare i suoi n dipendenti $1, \dots, n$.
L'azienda, nell'intervallo di tempo desiderato deve evadere m commesse $1, \dots, m$
Ciascuna commessa j deve essere svolta dal sottoinsieme $D_j \subseteq \{1, \dots, n\}$ dei dipendenti dall'azienda.
Ogni commessa, se evasa darebbe luogo ad un ricavo pari a r_j euro.
Ogni dipendente può lavorare ad una singola commessa nell'unità di tempo

Svolgimento

Si consideri che questo è un Problema di selezione di sottoinsiemi, infatti: $N = \{1, \dots, n\}$ elementi/dipendenti
$F = \{F_1, \dots, F_m\}$ dove F_j è la j-esima commessa

$$a_{ij} = \begin{cases} 1 & \text{se} i \in F_j \\ 0 & \text{altrimenti} \end{cases}$$

Variabili:

$$x_j = \begin{cases} 1 & \text{se la commessa } j \text{ è evasa} \\ 0 & \text{altrimenti} \end{cases}$$

$$x_j \in \mathbb{N} \quad y_j \in \mathbb{N}$$

Vincoli:

$$0 \leqslant x_j \leqslant 1 \quad \forall j \in \{1, \dots, m\} \quad 0 \leqslant y_j \leqslant 1$$

$$\forall i \in \{1, \dots, n\} \quad \sum_{j=1}^{m} a_{ij} x_j \leqslant 1 \quad y_j = 1 - x_j$$

Funzione obbiettivo:

$$\max \sum_{j=1}^{m} r_j x_j - \sum_{j=1}^{m} y_i p_j$$