Ricerca operativa

Mario Petruccelli
Università degli studi di Milano

A.A. 2019/2020

Sommario

1 Introduzione

Ricerca operativa: disciplina che affronta la risoluzione di problemi decisionali complessi tramite modelli matematici e algoritmi. Si parte da un **sistema organizzato** e lo si formalizza in un **modello matematico** per poi risolverlo tramite **algoritmi**.

1.1 Tassonomia modelli

- **Descrittivi** → Modelli che cercano di descrivere o simulare sistemi complessi *(e.g. modellini, plastici, ...)*

- **Predittivi** → Modelli che cercano di predire dei dati *(e.g. andamento mercati finanziari, previsioni, ...)*

- **Prescrittivi** → Modelli che trovano la soluzione ottimale ad un problema *(sono quelli che studieremo in questo corso).*

  La descrizione del problema avverrà attraverso **vincoli**, **obiettivi**.

Esempio di problemi decisionali

- Finanza (investimenti)
- Produzione (dimensionamento, organizzazione, ...)
- Logistica (gestione scorte, quanta merce, ...)
- Gestione (pianificazione, turnistica personale, ...)
- Servizi (rotte, ...)

NB *Lo stesso modello può servire per risolvere problemi diversi.*

Set covering Problema per la gestione di un territorio. I problemi dei sismografi e dei ripetitori sono diversi ma si ragiona allo stesso modo.

1.2 Programmazione matematica

La programmazione matematica (intesa come *pianificazione* delle azioni necessarie per individuare la soluzione ottima) è ciò che rappresenta il processo risolutivo nella ricerca operativa:

- Analisi del problema e scrittura di un modello matematico.
- Definizione e applicazione di un metodo di soluzione.

In particolar modo, la programmazione matematica si occupa di ottimizzare una funzione di più variabili, spesso soggette a dei vincoli. A seconda del tipo di modello abbiamo:

- Programmazione lineare continua.
- Programmazione lineare intera.
- Programmazione booleana.

2 Modellazione di problemi

2.1 Problema dello zaino

Ci sono n oggetti di valore p_j e ingombro w_j per $j = 1, \ldots, n$ ed è data la capacità massima b di un contenitore.

Problema Quali oggetti inserire nel contenitore senza superare capacità.

Obiettivo Massimizzare il valore degli oggetti. Si tratta di un problema di **ottimizzazione** e va formalizzato in modello matematico. Ci sono 4 componenti fondamentali.

Dati I dati sono informazioni conosciute a priori, in questo caso sono:

- $p_j \rightarrow$ valore dell'oggetto j.
- $w_j \rightarrow$ ingombro dell'oggetto j.
- $b \rightarrow$ capacità massima del contenitore.

Variabili Le variabili sono elementi che rappresentano una decisione.

- $x_j = \begin{cases} 1 \text{ se il j-esimo oggetto viene inserito} \\ 0 \text{ altrimenti} \end{cases}$

Obiettivo L'obiettivo è la funzione che rappresenta il risultato da ottenere.

- $max \sum_{j=1}^{n} p_j x_j \rightarrow massimizzare\ il\ valore$

Vincoli I vincoli sono le limitazioni che abbiamo sui dati.

- $\sum_{j=1}^{n} w_j x_j \leq b \rightarrow$ la somma degli ingombri degli oggetti presi non può superare la capacità del contenitore
- $x_j \in \{0, 1\} \quad j = 1, \ldots, n$

2.2 Problema di trasporto e localizzazione di impianti

Ci sono n siti candidati ad ospitare unità produttive, ciascuno con capacità massima a_i con $i = 1, \ldots, n$. Vi sono m magazzini, ognuno con una domanda da soddisfare b_j con $j = 1, \ldots, m$. Indichiamo con c_{ij} il costo di trasporto di una unità di prodotto dal sito i al magazzino j. L'attivazione di una unità produttiva nel sito i ha un costo fisso f_i.

Problema Dove aprire le unità produttive e come trasportare il prodotto dalle unità produttive aperte ai magazini in modo da soddisfare la domanda.

Obiettivo Minimizzare i costi di apertura e trasporto.

Dati

- $a_i \rightarrow$ capacità di produzione del sito i
- $b_j \rightarrow$ domanda del magazzino j
- $c_{ij} \rightarrow$ costo del trasporto di un'unità dal sito i al magazzino j.
- $f_i \rightarrow$ costo di attivazione unità nel sito i.

Variabili

- $y_i = \begin{cases} 1 \text{ se il sito } i \text{ ospita un'unità produttiva} \\ 0 \text{ altrimenti} \end{cases}$
- $x_{ij} =$ numero di unità trasportata dal sito i al magazzino j.

Obiettivo

- $min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{n} f_i y_i \rightarrow$ minimizzare il costo di attivazione di un unità nei vari siti e il costo dei trasporti delle unità.

Vincoli

- $\sum_{j=1}^{m} x_{ij} \leq a_i y_i \quad i = 1, \ldots, n \rightarrow$ le unità trasportate da un sito i non possono superare la capacità a_i di quel sito i.
- $\sum_{i=1}^{n} x_{ij} \geq b_j \quad j = 1, \ldots, m \rightarrow$ Le unità inviate ad un magazzino j dai vari siti deve soddisfare la domanda di quel magazzino.
- $x_{ij} \geq 0 \quad i = 1, \ldots, n \quad j = 1, \ldots, m$
- $y_i \in \{0, 1\} \quad i = 1, \ldots, n$

2.3 Problema assegnamento

Ci sono n lavoratori e n attività. Indichiamo con t_{ij} il tempo impiegato dal lavoratore i per svolgere l'attività j.

Problema Assegnare a ciascun lavoratore una sola attività, così che tutte le attività siano svolte.

Obiettivo Minimizzare il tempo richiesto a svolgere l'attività j.

Dati

- $t_{ij} \rightarrow$ tempo impiegato dal lavoratore i per svolgere l'attività j.

Variabili

- $x_{ij} = \begin{cases} 1 \text{ se il lavoratore } i \text{ svolge l'attività } j \\ 0 \text{ altrimenti} \end{cases}$

Obiettivo

- $min \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} x_{ij} \rightarrow$ minimizzare il tempo speso per svolgere tutte le attività dei vari lavoratori.

Vincoli

- $\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \rightarrow$ a ogni lavoratore è associata una sola attività.
- $\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \rightarrow$ a ogni attività è associata nn solo lavoratore.
- $x_{ij} \in \{0, 1\} \quad \forall i, j$

2.4 Mix Produttivo

Si hanno m risorse produttive con disponibilità b_i. Si possono produrre n prodotti diversi. Per produrre una unità di un prodotto *j-esimo* si utilizzano a_{ij} unità della risorsa *i-esima*. Ciascun prodotto ha un profitto unitario c_j.

Dati

- $b_i \rightarrow$ disponibilità risorsa *i-esima*.
- $a_{ij} \rightarrow$ unità della risorsa *i-esima* usate per produrre un prodotto *j-esimo*.
- $c_j \rightarrow$ profitto di un unità del prodotto j.

Variabili

- x_j = unità prodotte del prodotto *j-esimo*.

Obiettivo

- $max \sum_{j=1}^{n} c_j x_j \rightarrow$ massimizzare il profitto tra i vari prodotti.

Vincoli

- $\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \forall i \rightarrow$ le risorse usate nella produzione non possono superare la disponibilità di ciascuna risorsa.
- $x_j \geq 0 \quad \forall j$

2.4.1 Vernici

L'azienda produce due tipi di vernici, una vernice per interni (`I`) e una venrice per esterni (`E`), usando due materie prime indicate con `A` e `B`. La disponibilità al giorno di materia prima `A` è pari a `6 ton`, mentre quella di materia prima `B` è di `8 ton`. La quantità di `A` e `B` consumata per produrre una `ton` di vernice `E` ed `I` è riportata nella seguente tabella.

		Vernici	
		E	I
Materie prime	A	1	2
	B	2	1

Si ipotizza che tutta la vernice prodotta venga venduta. Il prezzo di vendita per tonnellata è `3K$` per `E` e `2Ks` per `I`. L'azienda ha effettuato un'indagine di mercato con i seguenti esisti:

- La domanda giornaliera di vernice `I` non super mai di più di `1 ton` quella di vernice `E`.
- La domanda massima giornaliera di vernice `I` è di `2 ton.`

Dati

- `3k$` per `E`.
- `2k$` per `I`.
- Disponibilità `A` 6 tonnellate.
- Disponibilità `B` 8 tonnellate.

Variabili

- x_E Tonnellate vernice E.
- x_I Tonnellate vernice I.

Obiettivo

- $\max 3x_E + 2x_I$

Vincoli

- $x_E + 2x_I \leq 6$
- $2x_E + x_I \leq 8$
- $x_I - x_E \leq 1$
- $x_I \leq 2$
- $x_E, x_I \geq 0$

2.4.2 Problema della dieta

Un determinato mangime per animali deve contenere in ogni dose almeno `2hg` di proteine, `4hg` di carboidrati e `3hg` di grasso. Si possono miscelare 4 ingredienti con le seguenti caratteristiche (*in hg per ogni kg)*.

Ingrediente	Proteine	Carboidrati	Grasso	Costo euro/kg
1	1	4	3	3
2	3	4	2	6
3	2	3	3	5
4	2	2	4	6

Problema Determinare quali ingredienti ed in quale quantità miscelare in modo da minimizzare il costo del mangime.

Dati

- Ogni dose deve contenere *almeno* `2hg` di proteine, `4hg` di carboidrati, `3hg` di grasso.

Variabili

- x_j = quantità di ingredienti j in `kg`.

Obiettivo

- $min \sum_{j=1}^{4} c_j x_j \rightarrow$ minimizzare il costo.

Vincoli

- $\min 3x_1 + 6x_2 + 5x_3 + 6x_4$
- $x_1 + 3x_2 + 2x_3 + 2x_4 \geq 2hg$ proteine
- $4x_1 + 4x_2 + 3x_3 + 2x_4 \geq 4hg$ carboidrati
- $3x_1 + 2x_2 + 3x_3 + 4x_4 \geq 3hg$ grasso
- $x_j \geq 0 \quad \jmath = 1 \ldots 4$

2.5 Miscelazione

Due tipi di benzina si ottengono miscelando 3 tipi di materie grezze. Le due benzine sono vendute rispettivamente a 40 cent/l e a 30 cent/l. Le materie grezze sono vendute a 10 cent/l, 16 cent/l, 14 cent/l e sono disponibili in quantità giornaliere pari a 100000 l, 70000 l, 120000 l.

Problema Produrre benzina con le quantità di materie a disposizione.

Obiettivo Massimizzare il profitto tenendo conto del costo delle materie.

Dati

- $r_j \rightarrow$ ricavo benzina *j-esima*.
- $c_i \rightarrow$ costo petrolio *i-esimo*.
- $a_i \rightarrow$ quantità giornaliera di petrolio *i-esimo*.
- $\%_{ij}^{m/M} \rightarrow$ percentuale minima e massima di petrolio *i-esimo* da avere all'interno della benzina *j-esima*.

Variabili

- p_{ij} = percentuale di petrolio i in benzina j.
- b_j = litri di benzina *j-esima* prodotti.
- p_i = litri di petrolio *i-esimo* usati.

- $x_{ij} := p_{ij}b_j \rightarrow$ litri di petrolio *i-esimo* usato per produrre i litri di benzina *j-esima (ciò permette di ottenere un modello lineare).*

Obiettivo

- $max[\underbrace{\sum_{j=1}^{n}(\sum_{i=1}^{m} x_{ij})r_j}_{\text{litri prodotti di benzina } j\text{-esima}} - \underbrace{\sum_{i=1}^{m}(\sum_{j=1}^{n} x_{ij})c_i}_{\text{litri di petrolio } i\text{-esimo utilizzati}}] \rightarrow$ massimizzare il ricavo netto della produzione.

Vincoli

- $\sum_{j=1}^{n} x_{ij} \leq a_i \quad \forall i \rightarrow$ i litri di petrolio *i-esimo* utilizzati non possono superare i litri disponibili giornalmente.
- $\%_{ij}^{m}(\sum_{k=1}^{m} x_{kj}) \leq x_{ij} \leq \%_{ij}^{M}(\sum_{k=1}^{m} x_{kj}) \quad \forall i,j \rightarrow$ i litri di petrolio *i-esimo* all'interno della miscela per benzina *j-esima* deve essere compresa tra gli estremi di percentuale dati dalla tabella.
- $x_{ij} \geq 0 \quad \forall i,j$

2.6 Turnazione personale

Ci sono 3 turni lavorativi (mattina, pomeriggio, notte). Sono presenti n lavoratori che svolgono 5 turni settimanali, dopo un turno lavorativo per un lavoratore ce ne devono essere almeno 2 di riposo. Ogni lavoratore propone 5 turni in ordine di preferenza.

Problema Organizzare turni in modo tale che ognuno si coperto.

Obiettivo Minimizzare il grado di soddisfacibilità globale.

Dati

- $p_{gt}^{j} \rightarrow$ grado di soddisfacibilità del lavoratore *j-esimo* a lavorare il giorno g nel turno t.
- $r_{gt} \rightarrow$ numero lavoratori necessari il giorno g al turno t.

Variabili

- $x_{gt}^{j} = \begin{cases} 1 \text{ se il lavoratore } j\text{-esimo lavora il giorno } g \text{ al turno } t \\ 0 \text{ altrimenti} \end{cases}$

Obiettivo

- $min \sum_{j=1}^{n} \sum_{g \in G} \sum_{t \in T} p_{gt}^{j} x_{gt}^{j}$

Vincoli

- $\sum_{j=1}^{n} x_{gt}^{j} \geq r_{gt} \quad \forall g, t \rightarrow$ per ogni turno giornaliero ci devono essere tanti lavoratori quanti sono richiesti.

- $\sum_{g \in G} \sum_{t \in T} x_{gt}^{j} \geq 5 \quad \forall j \rightarrow$ ogni lavoratore deve lavorare per almeno 5 turni.

- $$\left.\begin{aligned} x_{l,m}^{j} + x_{l,p}^{j} + x_{l,s}^{j} \leq 1 \\ x_{l,p}^{j} + x_{l,s}^{j} + x_{ma,m}^{j} \leq 1 \\ \dots \\ x_{d,m}^{j} + x_{d,p}^{j} + x_{d,s}^{j} \leq 1 \end{aligned}\right\} \forall j \rightarrow \text{ogni lavoratore deve avere almeno 2 turni di riposo dopo un turno di lavoro.}$$

2.7 Locazione di servizi

Abbiamo un insieme $N = \{1, \dots, n\}$ di potenziali localizzazioni di servizi ed un insieme $I = \{1, \dots, m\}$ di clienti. Ogni località j ha una capacità u_j e costo di attivazione c_j. Ogni cliente i ha una richiesta b_i. Il costo da sostenere per servire il cliente i dalla località j è h_{ij} e ogni cliente è servito da una sola località.

Problema Determinare la localizzazione di servizi così da soddisfare ogni cliente.

Obiettivo Minimizzare il costo di attivazione e di servizio complessivo.

Dati

- $u_j \rightarrow$ capacità località *j-esima.*

- $c_j \rightarrow$ costo attivazione località *j-esima.*

- $b_i \rightarrow$ richiesta cliente *i-esimo.*

- $h_{ij} \rightarrow$ costo per servire il cliente i dalla località j.

Variabili

- $y_j = \begin{cases} 1 \text{ se è attivo un servizio nella località } \textit{j-esima} \\ 0 \text{ altrimenti} \end{cases}$

- $x_{ij} = \begin{cases} 1 \text{ se il cliente } i \text{ viene servito dalla località } j \\ 0 \text{ altrimenti} \end{cases}$

Obiettivo

- $min[\sum_{j\in N}\sum_{i\in I} h_{ij}x_{ij} + \sum_{j\in N} c_j y_j] \rightarrow$ minimizzare il costo di attivazione di servizio tra i vari siti e clienti.

Vincoli

- $\sum_{j\in N} x_{ij} = 1 \quad \forall i \in I \rightarrow$ ogni cliente i è servito da una sola località j.
- $\sum_{i\in I} x_{ij} b_i \leq u_j \quad \forall j \in N \rightarrow$ ogni località j attiva deve soddisfare la domanda b_i del cliente i associato.
- $x_{ij} \leq y_j \quad \forall i \in I, j \in N$
- $x_{ij} \in \{0, 1\} \quad \forall i, j$
- $y_j \in \{0, 1\} \quad \forall j$

  Gli ultimi 3 sono vincoli opzionali (ma non troppo). Il simplesso risolve il modello nel continuo, da cui tira poi fuori la soluzione intera.

2.8 Bin packing

Ci sono n oggetti, ciascuno con ingombro w_j. Sono dati dei contenitori di capacità b.

Problema Assegnare gli oggetti ai contenitori rispettando le capacità.

Obiettivo Minimizzare il numero di contenitori usati.

Dati

- $w_j \rightarrow$ ingombro oggetto *j-esimo*.
- b capacità contenitori.

Variabili

- $x_{ij} = \begin{cases} 1 \text{ se il contenitore } \textit{i-esimo} \text{ accetta l'oggetto } \textit{j-esimo} \\ 0 \text{ altrimenti} \end{cases}$

- $y_i = \begin{cases} 1 \text{ se uso il contenitore } \textit{i-esimo} \\ 0 \text{ altrimenti} \end{cases}$

Obiettivo

- $min \sum_{i=1}^{n} y_i \rightarrow$ minimizzare il numero di contenitori usati.

Vincoli

- $\sum_{j=1}^{n} x_{ij} w_j \leq b y_i \quad \forall i \rightarrow$ tutti gli oggetti contenuti in ogni contenitore *i-esimo* non devono superare la capacità b.

- $\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \rightarrow$ ogni oggetto può essere messo in un unico contenitore.

- $x_{ij} \in \{0, 1\} \quad \forall i, j$

- $y_i \in \{0, 1\} \quad \forall i$

2.9 Problema di assegnamento

Ci sono m macchine identiche e n lavorazioni. Ogni lavorazione j richiede di essere processata da una qualsiasi delle m macchine per una durata ininterrotta p_j. Ogni macchina processa una sola lavorazione alla volta.

Problema Come assegnare le lavorazioni alle macchine.

Obiettivo Minimizzare l'istante di completamento della macchina che termina per ultima.

Dati

- $p_j \rightarrow$ durata della lavorazione *j-esima*.

Variabili

- $x_{ij} = \begin{cases} 1 \text{ se la macchina } \textit{i-esima} \text{ effettua la lavorazione } j \\ 0 \text{ altrimenti} \end{cases}$

- $T \geq 0 \rightarrow$ istante di completamento della macchina che termina per ultima.

Obiettivo

- $min\ max \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij}p_j \rightarrow$ minimizzare l'istante di completamento della macchina che termina per ultima.

Vincoli

- $\sum_{j=1}^{m} x_{ij}p_j \leq T \quad \forall i \rightarrow$ ogni macchina termina le lavorazioni al più in contemporanea con la macchna che termina per ultima.

- $\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \rightarrow$ ogni lavorazione *i-esima* è effettuata da na e una sola macchina.

- $x_{ij} \in \{0, 1\} \quad \forall i, j$

2.10 Problema di sequenziamento monoprocessore

C'è una macchina e ci sono n lavorazioni. Ogni lavorazione j ha un tempo di processamento p_j, è disponibile a partire dall'istante r_j e deve essere completata entro la data d_j. La macchina può processare una sola lavorazione alla volta.

Problema In quale ordine processare le lavorazioni sulla macchina.

Obiettivo Minimizzare la somma degli istanti di completamento di tutte le lavorazioni.

Dati

- $p_j \rightarrow$ tempo di processamento della lavorazione *j-esima*.

- $r_j \rightarrow$ istante minimo di inizio della lavorazione *j-esima*.

- $d_j \rightarrow$ istante massimo di completamento della lavorazione *j-esima*.

Variabili

- $x_{ij} = \begin{cases} 1 \text{ se la lavorazione } i \text{ precede la lavorazione } j \\ 0 \text{ altrimenti} \end{cases}$

- $c_j =$ istante di completamento del lavoro *j-esimo*.

Obiettivo

- $min \sum_{j=1}^{n} c_j \rightarrow$ minimizzare la somma degli istanti di completamento delle lavorazioni.

Vincoli

- $c_j \leq c_k - p_k + M(1 - x_{jk}) \quad 1 \leq j < k \leq n \quad M \in R \rightarrow$ se j precede k, il suo istante di completamento al più coincide con quello di inizio di k.
- $c_k \leq c_j - p_j + Mx_{jk} \quad 1 \leq j < k \leq n \quad M \in R \rightarrow$ vincolo ridondante, poichè tiene conto del caso opposto (k precede j).
- $0 \leq p_j + r_j \leq c_j \leq d_j \quad \forall j = 1, \ldots, n \rightarrow$ l'istante minimo di fine processo j ($p_j + r_j$) al più coincide con il suo istante di completamento effettivo, ed al più coincide con l'istante massimo di completamento.
- $x_{jk} \in \{0, 1\} \quad 1 \leq j < k \leq n$

2.11 Problema di pianificazione della produzione

Determiniamo un piano di produzione di un prodotto specifico nell'arco di n periodi. Per ciascun periodo conosciamo la domanda da soddisfare d_t, il costo di produzione c_t e il costo di magazzino i_t per unità di prodotto. La capacità massima di produzione è c.

Problema Pianificare la produzione così da soddisfare la domanda per ogni periodo.

Obiettivo minimizzare i costi.

Dati

- $d_t \rightarrow$ domanda del periodo *t-esimo*.
- $c_t \rightarrow$ costo di produzione di un'unità nel periodo *t-esimo*.
- $i_t \rightarrow$ costo di magazzino di un unità nel periodo *t-esimo*.
- $c \rightarrow$ capacità massima di produzione.

Variabili

- $x_t =$ unità del prodotto nel periodo *t-esimo*.
- $m_t =$ unità del prodotto immagazzinate al termine del periodo *t-esimo*

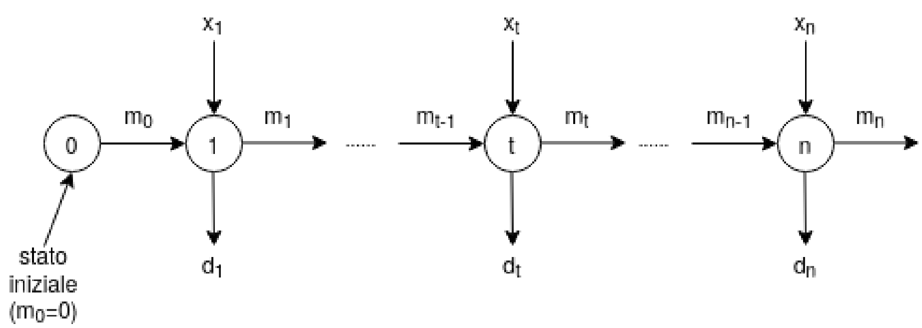
Obiettivo

- $min \sum_{t=1}^{n} c_t x_t + \sum_{t=1}^{n} i_t m_t \rightarrow$ minimizzare il costo di produzione e di magazzino.

Vincoli

- $m_{t-1} + x_t = d_t + m_t \quad t = 1, \dots, n \rightarrow$ le unità immagazzinate dal periodo precedente insieme alle unità prodotte nel periodo t attuale devono coincidere con la domanda nel periodo t sommati ai prodotti rimanenti immagazzinati.
- $x_t \leq c \quad t = 1, \dots, n \rightarrow$ le unità prodotte non possono superare la capacità produttiva.

Questo problema è rappresentabile tramite un **modello di flusso**:



Modello di flusso

A ogni periodo la produzione deve soddisfare la domanda (e non superare la capacità) e le unità rimaste vanno in magazzino.

2.11.1 Variante lotto minimo

Ogni periodo il lotto minimo è pari a L.

Variabili $\quad y_t = \begin{cases} 1 \text{ se produco un lotto al periodo } t\text{-}esimo \\ 0 \text{ altrimenti} \end{cases}$

Vincoli

- $x_t \geq L y_t \quad t = 1, \dots, n$
- $x_t \leq c y_t \quad t = 1, \dots, n$

2.11.2 Variante produzione con costi fissi

Se in un periodo è stata prodotta almeno una unità di prodotto si aggiunge un costo fisso k.

Obiettivo $min \sum_{t=1}^{n} c_t x_t + \sum_{t=1}^{n} i_t m_t + \sum_{t=1}^{n} k y_c \rightarrow$ aggiungo il costo fisso nel caso in cui produco qualcosa.

2.11.3 Variante multiprodotto

Possono essere fabbricati più prodotti durante gli n periodi.

Variabili

- $x_t^j \rightarrow$ unità di prodotto j fabbricate nel peridodo t.
- $m_t^j \rightarrow$ unità di prodotto j immagazzinate nel periodo t.

Obiettivo $min \sum_j (\sum_{t=1}^{n} c_t^j x_t^j + \sum_{t=1}^{n} i_t^j m_t^j + \sum_{t=1}^{n} k y_t^j)$

2.12 Set Covering

È dato un insieme $M = \{1, 2, \ldots, m\}$ ed una famiglia di n suoi sottoinsiemi $S_j \subseteq M$. Ogni sottoinsieme ha un costo c_j

Problema Trovare un insieme $T \subseteq N = \{1, \ldots, n\}$ tale che l'unione degli S_j, con $j \in T$ sia uguale a M.

Obiettivo Minimizzare i costi dei sottoinsiemi scelti.

Dati

- $M \rightarrow$ insieme di partenza.
- $S_j \rightarrow$ sottoinsieme di M.
- $c_j \rightarrow$ costo del sottoinsieme *j-esimo*.

Variabili

- $y_j = \begin{cases} 1 \text{ se scelgo il sottoinsieme } \textit{j-esimo } S_j. \\ 0 \text{ altrimenti.} \end{cases}$

Obiettivo

- $min \sum_{j \in N} c_j y_j$ $\rightarrow$ minimizzare il costo dei sottoinsiemi scelti per coprire M.

Vincoli

- $\bigcup_{j \in T} S_j = M \rightarrow$ l'unione dei sottoinsiemi scelti coincide con M.
  Tuttavia S_j è rappresentabile come un vettore di m elementi:

- $S_j = \begin{pmatrix} a_{1,j} \\ a_{2,j} \\ \dots \\ a_{m,j} \end{pmatrix} \rightarrow a_{ij} = \begin{cases} 1 \text{ se l'elemento } \textit{i-esimo} \text{ appartiene al sottoinsieme } S_j \\ 0 \text{ altrimenti.} \end{cases}$

  Inoltre, il sottoinsieme S_j viene viene scelto se la rispettiva variabile y_j è uguale a 1, quindi utilizziamo un vettore di n elementi, costituito da y_j.

  $\underline{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix}$

  Ora rappresentiamo anche a_i, j sotto forma matriciale:

  $A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}$

  Se effettuiamo il prodotto matriciale tra A e $\underline{Y}$ otteniamo un sistema, in cui ogni riga rappresenta quante volte il valore $i \in M$ compare tra tutti i sottoinsiemi scelti.

  $A\underline{Y} \geq 1 \rightarrow \begin{cases} a_{1,1}y_1 + a_{1,2}y_2 + \dots + a_{1,n}y_n \geq 1 \\ \dots \\ a_{m,1}y_1 + a_{m,2}y_2 + \dots + a_{m,n}y_n \geq 1 \end{cases}$

- $y_j \in \{0,1\} \quad \forall j \rightarrow \underline{Y} \in \{0,1\}^n$

2.13 Modellare vincoli logici utilizzando variabili binarie

In primo luogo associamo una variabile binaria a ciascuna variabile logica. Ad esempio alla variabile logica $X =$"attivare l'impianto di produzione" associamo la variabile binaria x nel seguente modo: $x = \begin{cases} 1 \text{ "attivazione dell'impianto di produzione"} \\ 0 \text{ "non attivazione dell'impianto di produzione"} \end{cases}$

Rappresentiamo	con
$\neg X$	$(1 - x)$
$X \vee Y$	$(x + y)$
$X \rightarrow Y$	$x \leq y$
$X_1 \vee X_2 \vee \cdots \vee X_n \rightarrow Y$	$\sum_1^n x_i \leq ny$
$X \rightarrow Y_1 \vee Y_2 \vee \cdots \vee Y_n$	$x \leq \sum_1^n y_i$
$X \rightarrow Y_1 \wedge Y_2 \wedge \cdots \wedge Y_n$	$nx \leq \sum_1^n y_i$

Esempi				
$X \rightarrow (\neg Y \vee \neg Z)$	diviene	$x \leq (1 - y) + (1 - z)$	cioè	$x + y + z \leq 2$
$(X \vee Y) \rightarrow (\neg Z)$	diviene	$x + y \leq 2(1 - z)$		
$(X \vee Y) \wedge (\neg Z \vee Y)$	diviene	$x + y \geq 1, (1 - z) + y \geq 1$		
Almeno due fra X, Y, Z	diviene	$x + y + z \geq 2$		
Al più k fra $X_1, X_2, \ldots, X_n$	diviene	$\sum_1^n x_i \leq k$		

3 Parte formale della ricerca operativa

3.1 Tecnica di soluzione lineare

Prendiamo un esempio che abbiamo già visto, il problema delle vernici.

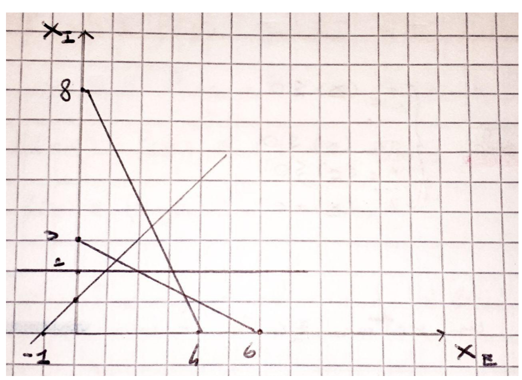
Obiettivo

- $\max 3x_E + 2x_I \leq 6$

Vincoli

- $x_E + 2x_I \leq 6$
- $2x_E + x_I \leq 8$
- $x_I - x_E \leq 1$
- $x_I \leq 2$
- $x_E, x_I \geq 0$

1. Disegnamo lo **spazio delle soluzioni** di x_E e x_I che soddisfano tutti i vincoli.
   (a) Consideriamo nel primo vincolo solamente l'uguaglianza $x_E + 2x_I = b$
   (b) Inseriamo i punti sul piano in base alle soluzioni del vincolo per una delle 2 variabili fissate.
      i. $x_E = 0 \rightarrow x_I = 3$
      ii. $x_I = 0 \rightarrow x_E = 6$
   (c) Traccia la retta tra i 2 punti.
   (d) Ripeti per ogni vincolo.

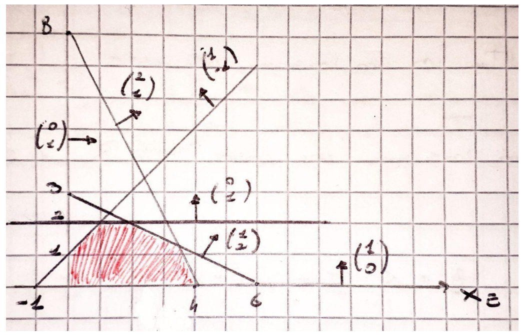


2. Ora cerchiamo di evidenzialre la **regione ammissibile**

    (a) Rappresentiamo i vettori **gradienti** per ogni vincolo, ciascuno rappresentato dai coefficienti del vincolo. $x_E + 2x_I \leq 6 \rightarrow \binom{1}{2}$

    (b) Nel caso di vincoli con il $\geq$ il **gradiente** indica la parte dei punti soddisfatta dal vincolo, altrimenti l'opposto.

    (c) L'intersezione tra tutti gli spazi da la regione ammissibile



3. Ora troviamo il punto che da la **soluzione ottima**.

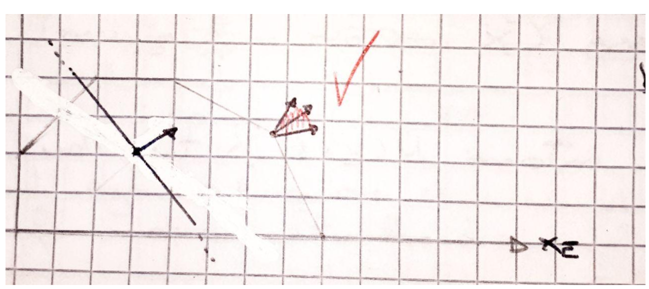
    (a) Si genera un punto generico nella regione ammissibile.

    (b) Sul punto si disegna il gradiente della funzione obiettivo.

    (c) Poi la retta ortogonale al gradiente (**retta di Isocasto**) in cui tutti i punti hanno lo stesso costo.

    (d) Il gradiente indica la direzione e il verso da seguire per aumentare il valore della funzione, quindi, dovendo massimizzare la funzione, spostiamo la retta in quella direzione fino a che la retta rimane nella regione ammissibile.

    (e) Per verificarlo, dobbiamo essere sicuri che il **gradiente** della funzione obiettivo sia nel cono tra il gradiente del primo vincolo incontrato e del secondo.



3.2 Tecnica di programmazione matematica

Formalizziamo ora un problema come **problema di programmazione matematica**: Problema (f, X) con

- $f : R^n \to R$ funzione obiettivo.
- $X \subseteq R^n$ regione ammissibile

Quindi un problema per noi diventa $minf(\underline{x}) \quad \underline{x} \in X$.

Con $\underline{x}$ definita: $\underline{x} \in X \iff \underline{x}\ soddisfa \begin{cases} g_1(\underline{x}) \leq 0 \\ \dots \\ g_m(\underline{x}) \leq 0 \\ h_1(\underline{x}) = 0 \\ \dots \\ h_k(\underline{x}) = 0 \end{cases}$

3.2.1 Convessità

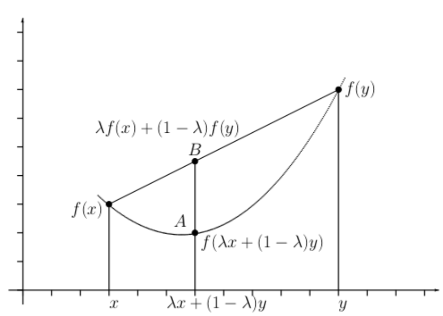
Dati $\underline{x}$ e $\underline{y} \in R^n$ e lo scalare $\lambda \in [0, 1]$, un vettore $\underline{z} \in R^n$ è una combinazione convessa di $\underline{x}$ e $\underline{y}$ se:

$$\underline{z} = \lambda\underline{x} + (1 - \lambda)\underline{y}$$

Insieme convesso Un insieme $S \subseteq R^n$ è convesso se ogni combinazione convessa di una qualunque coppia $\underline{x}, \underline{y} \in S$ appartiene ad S stesso. L'intersezione di un qualunque numero di insiemi convessi è un insieme convesso.

Funzione convessa Una funzione $f : X \to R$ definita su di un insieme convesso $X \subseteq R^n$ si dice **convessa** se $\forall \underline{x}, \underline{y} \in X$ e $\forall \lambda \in [0, 1]$ si ha che

$$f(\underline{z}) \leq \lambda f(\underline{x}) + (1 - \lambda)f(\underline{y}) \text{ con } \underline{z} = \lambda\underline{x} + (1 - \lambda)\underline{y}$$



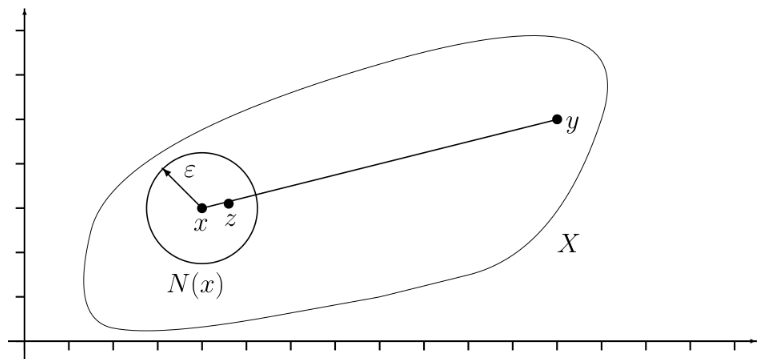
Funzione concava La funzione g è concava (sull'insieme convesso $X \subseteq R^n$) se $-g$ è convessa in X:

$$g(\underline{z}) = g(\lambda\underline{x} + (1-\lambda)\underline{y}) \geq \lambda g(\underline{x}) + (1-\lambda)g(\underline{y}) \quad \forall \underline{x}, \underline{y} \in X$$

Minimo locale $\underline{x} \in X$ è un minimo locale se esiste un intorno $N(\underline{x}) \subseteq X$ tale che $f(\underline{z}) \geq f(\underline{x})$ per ogni $\underline{z} \in N(\underline{x})$.

$$N(\underline{x}) = \{\underline{z} : \underline{z} \in X \text{ e } ||\underline{x} - \underline{z}|| \leq \epsilon\}$$

Teorema Dato un problema di ottimizzazione convessa (X, f) ogni **minimo locale** è anche **minimo globale.**



Dimostrazione La tesi da dimostrare è: $\forall \underline{y} \in X$ risulta $f(\underline{y}) \geq f(\underline{x})$.

- Il teorema vale se $\underline{y} \equiv \underline{z} \in N(\underline{x})$.
- Metto in relazione $\underline{x}, \underline{z}, \underline{y}$ e i corrispondenti valori della f.o.:

1. Per trovare un controesempio, basta prendere un vettore $\underline{y}$ non appartenente all'interno di $N(\underline{x})$, che sia minore di $\underline{x}$ *(minimo locale).*

$$\underline{y} \notin N(x) \quad f(\underline{x}) \leq f(\underline{y})$$

2. Prendiamo qundi uno $\underline{z}$ che sia combinazione convessa di $\underline{x}$ e $\underline{y}$ e che appartenga all'interno.
$$f(\underline{z}) = f(\lambda\underline{x} + (1-\lambda)\underline{y}) \leq \lambda f(\underline{x}) + (1-\lambda)f(\underline{y})$$

3. $f(\underline{x}) \leq f(\underline{z})$ perchè $\underline{x}$ è minimo locale.

4. - $f(\underline{x}) \leq \lambda f(\underline{x}) + (1-\lambda)f(\underline{y})$

- $f(\underline{x}) - \lambda f(\underline{x}) \leq (1-\lambda)$
- $(1-\lambda)f(\underline{x}) \leq (1-\lambda)f(\underline{y})$ ma, essendo $\underline{x}$ e $\underline{y}$ diversi, $\lambda \neq 1$ e $\lambda \neq 0$, quindi:
- $f(\underline{x}) \leq f(\underline{y}) \quad \square$

Noi ci interesseremo al mondo della **programmazione lineare**. (f, X) è detto problema di programmazione lineare se e solo se la funzione obiettivo f e tutte le funzioni che definiscono la regione ammissibile X $(g_1(\underline{x}), \dots, g_m(\underline{x}), h_1(\underline{x}), \dots, h_k(\underline{x})$ sono **lineari**, ossia sono concave e convesse contemporaneamente. Di conseguenza X è convesso perchè intersezioni di insiemi (disuguaglianze) convessi.
Perchè X è definito da un insieme di disuguaglianze, il cui sistema genera una intersezione convessa, dimostriamo che un insieme definito in questo modo è a sua volta convesso.

Dimostrazione $\quad X = \{\underline{x} \in R^n : f(\underline{x}) \leq 0\} \quad f$ convessa

1. Consideriamo 2 punti $\underline{x}$ e $\underline{y}$ tali che: $f(\underline{x}) \leq 0$ e $f(\underline{y}) \leq 0$
2. Preso $\underline{z}$ come **combinazione convessa** di $\underline{x}$ e $\underline{y}$ ($\underline{z} = \lambda\underline{x} + (1-\lambda)\underline{y}$) vale che $f(\underline{z}) \leq 0$:

$$f(\underline{z}) \leq \underbrace{\underbrace{\lambda}_{\geq 0}\underbrace{f(\underline{x})}_{\leq 0}}_{\leq 0} + \underbrace{\underbrace{(1-\lambda)}_{\geq 0}\underbrace{f(\underline{y})}_{\leq 0}}_{\leq 0} \rightarrow \text{poichè } f \text{ è funzione convessa}$$

Quindi ogni combinazione convessa di 2 punti che soddisfano la disuguaglianza, soddisfa a sua volta la disuguaglianza $\rightarrow X$ insieme convesso. $\quad \square$

Ovviamente la programmazione lineare è solo un caso specifico di quella convessa.

3.3 Geometria della programmazione lineare

Iperpiano *(di supporto di un vincolo)* $\{\underline{x} \in R^n : \underline{\alpha}^T\underline{x} = \alpha_0\}$

Semispazio $\{\underline{x} \in R^n : \underline{a}^T\underline{x} \leq \alpha_0\}$

Con $\underline{a}$ vettore dei coefficienti, $\underline{x}$ vettore delle variabili e α_0 valore reale $\rightarrow \underline{a}^T\underline{x}$ prodotto scalare tra $\underline{a}$ e $\underline{x}$ ($\underline{a}^T$ trasposta di $\underline{a}$).

Iperpiano e semispazio **insiemi convessi** $\rightarrow$ la loro intersezione genera un insieme convesso.

Poliedro intersezione di un numero finito di **iperpiani** e **semispazi**.

Politopo poliedro P limitato, ossia: $\exists M > 0 : ||\underline{x}|| \leq M \quad \forall \underline{x} \in P$

Vertice punto x di un poliedro P che non può essere espresso come **combinazione convessa stretta** ($\lambda \neq 0, 1$) di altri 2 punti del poliedro

$$\nexists \underline{y}, \underline{z} \in P, \underline{y} \neq \underline{z}, \lambda \in (0,1) : \underline{x} = \lambda \underline{y} + (1-\lambda)\underline{z}$$

Ogni politopo ha un **numero finito di vertici**.

3.3.1 Forma matriciale del modello

Obiettivo $max\ 3x_1 + 2x_2$

Vincoli

- $8x_1 + 4x_2 \leq 64$
- $4x_1 + 6x_2 \leq 54$
- $x_1 + x_2 \leq 10$
- $x_{1,2} \geq 0$

Vettore dei coefficienti dei costi $\underline{c}^T = (3, 2)$ vettore che contiene i coeeficienti della funzione obiettivo

Vettore dei termini noti $\underline{b} = \begin{pmatrix} 64 \\ 54 \\ 10 \end{pmatrix}$ vettore che contiene i termini noti posti dopo le disuguaglianze nei vincoli.

Matrice dei coefficienti tecnologici $A = \begin{pmatrix} 8 & 4 \\ 4 & 6 \\ 1 & 1 \end{pmatrix}$ matrice che contiene i coefficienti moltiplicativi delle variabili nei vincoli.

Con A_i indichiamo la colonna *i-esima* di A, con $\underline{a}_i^T$ indichiamo la riga *i-esima* di A.

Vettore delle variabili $\underline{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ vettore contenente tutte le variabili del modello.

Quindi il problema è riformulabile in questi termini:

Funzione obiettivo $max\ \underline{c}^T \underline{x}$

Vincoli

- $A\underline{x} \leq \underline{b}$
- $\underline{x} \geq 0$

$\underline{P} = \{\underline{x} \in R^n | A|vx \leq b \wedge \underline{x} \geq 0\} \rightarrow$ insieme dei punti che soddisfano i vincoli, ossia la regione ammissibile (politopo o poliedro).

Inoltre, in funzione della regione ammissibile $\underline{P}$, distinguiamo 3 tipi di problemi:

- **Soluzione ottima finita** la cui regione ammissibile è un **politopo** (ossia una regione di spazio limitata), in cui quindi il numero di souzioni ottime è finito ($\underline{P} \neq \varnothing$)
- **Problema illimitato** la cui regione ammissibile è un **poliedro** (ossia una regione di spazio illimitata)
- **Problema inammissibile** la cui regione ammissibile è vuota, non ammettendo quindi soluzioni ($\underline{P} = \varnothing$).

Nel primo caso, si possono verificare 3 diverse situazioni:

- **Soluzione unica** $\rightarrow$ esiste un unico punto *(vertice)* che è soluzione ottima.
- **Ottimo multiplo** $\rightarrow$ l'insieme delle soluzioni ottime non è finito, poichè corrisponde una faccia del politopo.
- **Soluzione degenere** $\rightarrow$ la soluzione del problema è un vertice definito da più di 2 iperpiani (per definire un vertice è necessaria l'intersezione di soli 2 iperpiani) quindi quel punto nasconde più vertici (a causa dell'intersezione di tutte le coppie di iperpiani possibili).

3.3.2 Teorema di Minkowski-Weil

Ogni punto di un politopo si può ottenere come combinazione convessa dei suoi vertici.

Teorema Se $\underline{P} = \{\underline{x} \in R^n | A\underline{x} \leq b \wedge \underline{x} \geq 0\}$ è un politopo allora esiste almeno un vertice di $\underline{P}$ ottimo per il problema $min\ \{c^T\underline{x} | \underline{x} \in \underline{P}\}$