

# Ricerca Operativa

Mascherpa Matteo

a.a. 2024-2025

## 1 Introduzione

Con lo scalare improvviso delle imprese della rivoluzione industriale è diventata preponderante la necessità di allocare al meglio le risorse disponibili. Spesso se un'operazione cresce velocemente è facile che ogni compartimento diventi un'entità a sé stante rendendo complessa la gestione. Da questa necessità è emersa la *Ricerca operativa* che è maturata ed è stata sfruttata per la prima volta nella seconda guerra mondiale. Servì per allocare nel miglior modo le poche risorse inglesi per evitare di soccombere ai tedeschi. Con l'avvenire dei computer poi questa materia ha avuto una spinta che la ha portata alla situazione attuale.

### 1.1 Attributi di una decisione

Una buona decisione è caratterizzata da:

1. ***Efficace***: Raggiunge lo scopo prefissato.
2. ***Efficiente***: Nel raggiungere lo scopo prefissato lo fa in modo da sprecare meno risorse possibile
3. ***Tempestiva***: Viene attuata in un arco di tempo coerente con l'orizzonte degli eventi prefissato(strategico(immediato), tattico(medio periodo), operativo(a lungo periodo)).
4. ***Robusta***: Rimane la una buona scelta anche in caso in caso di variazioni trascurabili.
5. ***Giustificabile***: Se si ha il compito di spiegare il motivo della decisione si può fare senza dover usare sotterfugi e depistaggi.

## 2 Panoramica all'approccio della modellazione

Nonostante la parte più preponderante della materia sia composta da metodi matematici non si deve dimenticare che una parte dello sforzo è la *raccolta dei dati*. Un riassunto delle varie fasi di uno studio sono:

1. Definire il problema e ottenere i dati.
2. Formulare il modello matematico che rappresenti il problema.
3. Sviluppare la procedura per derivare la soluzione del problema dal modello.

4. Test, miglioramento, rifinitura fino a quando si è soddisfatti.
5. Preparare l'applicazione operativa della soluzione.
6. Implementazione della soluzione trovata.

## 2.1 Definire il problema

Solitamente il problema da risolvere è descritto in termini vaghi, va quindi trovato un modo per formalizzarlo. Solitamente, non si deve dare la "soluzione" ma un insieme di diverse alternative dove il cliente sceglierà quella che più lo aggrada in base a quale soluzione lo attrae di più (Alcuni potrebbero volere una estrema rapidità altri una maggiore robustezza). Per prima cosa si deve discernere i **principi fondamentali**, che saranno quelli che guideranno la scelta finale. Generalmente per le imprese i punti salienti sono: *Proprietario, Impiegati, Clienti, Fornitori, Governo/nazione*.

## 2.2 Formulo il modello matematico

Il modello matematico sono versioni idealizzate della realtà per una più facile lavorazione. Esso è composto da **variabili decisionali**,  $(x_1, \dots, x_n)$  che rappresentano i valori come valori determinati. Le misure delle prestazioni con le **funzioni obiettivo** espresse come funzioni aritmetiche,  $(P = 3x_1 + 2x_2 + \dots + mx_n)$ . Ci sono anche modi per avere delle restrizioni, chiamate **vincoli**, espresse come equazioni,  $(3x_1 + 2x_2 + \dots + mx_n < 10)$ . Le costanti del modello vengono chiamate **parametro del modello**. Solitamente, per motivo di incertezza, i parametri non sono altro che grezze stime, concetto a cui ci si riferisce come **sensitività d'analisi**. Una volta modellato il problema esso non diventa altro che un problema di programmazione lineare. Bisogna evitare però di creare un modello insoddisfacibile essendo il modello astratto e idealizzato ma con un utilizzo concreto. Un criterio per giudicare un modello è la relativa correttezza in base alle scelte prese. Nello sviluppo del modello è buona pratica da cominciare da un nucleo ristretto e arricchirlo, questo processo si chiama **arricchimento del modello**. In caso gli obiettivi fossero molteplici essi vengono trasformati in una misura composita chiamata: **Misura globale delle prestazioni**.

## 2.3 Sviluppare la procedura

Solitamente un passo semplice dove applicare al modello sviluppato precedentemente un *algoritmo standard*, la parte più complessa è l'*analisi postottimale*. Gli algoritmi infatti restituiscono la soluzione *ottima* per il modello dato, il problema è che il modello dato essendo un'astrazione può quasi sempre essere migliorato per ottenere una nuova soluzione migliore. **Herbert Simon**, nobel per l'economia, ha definito "*satisficing*" (*soddisfacente*) la tendenza dei manager a cercare una soluzione "*abbastanza buona*" e non per forza ottima. Bisogna, nella modifica del modello porre attenzione alla *sensitività delle analisi*, cioè determinare quale parametro modifica considerevolmente la soluzione e la sposta di più verso un obiettivo che più si confà alla realtà del problema.

## 2.4 Test, Miglioramento, e rifinitura

Essendo i casi della realtà, al contrario degli esempi didattici, enormi è delirante aspettarsi che la prima soluzione trovata sia quella che alla fine si andrà ad attuare. Per questo esiste questa fase dove il modello viene messo sotto stress fino ad arrivare ad un livello soddisfacente. Non ci si può

neanche aspettare di arrivare alla versione perfetta, si punta perciò alla minimizzazione dei bachi. Questo processo si chiama ***validazione del modello***. Visto l'ammontare di tempo speso nelle fasi precedenti è facile trascurare, per dimenticanza, vari dettagli. Per questo è spesso consigliato che a questo processo venga aggiunto nel team un elemento totalmente avulso dal progetto. Tra gli errori più frequenti ci sono gli errori di incosistenze nelle unità dei valori, *consistenza dimensionale*. Un approccio sistematico è il *test a retrospettiva*. Ciò consiste nell'utilizzo di dati passati per ricostruire vecchi scenari simili e determinare la risposta del modello. Lo svantaggio è che i dati usati per i test sono gli stessi usati per creare il modello quindi non i migliori per trovare le falle.

## 2.5 Preparazione dell'applicazione

Una volta testato il modello ed essere arrivati ad una versione da adottare allora si deve creare una documentazione estesa che spieghi ogni dettaglio del modello. Il sistema che verrà creato sarà quasi certamente informatizzato. In questa fase si creano le *basi di dati* e *sistemi di gestione delle informazioni*. In caso le decisioni da prendere siano interattive si crea un *sistema di supporto alle decisioni* in modo da avere un rapido accesso alle possibili migliori decisioni.

## 2.6 Implementazione

La fase critica in cui le scelte prese in precedenza vanno, finalmente, attuate nel modo più accurato possibile al modello ricavato dai mesi di lavoro. Tasselli importanti nell'applicazione sono la comunicazione tra reparti. se tutto viene svolto in cordinazione tra i vari capi reparto il modello potrà essere utilizzato per anni, per questo è importante un riscontro costante per potere avere aggiustamenti minori che vanno documentati e motivati. Grazie a questo si potrà andare avanti fino a quando non cambieranno le necessità che il modello soddisfaceva.

# 3 Programmazione lineare

In breve l'applicazione più comune per la programmazione lineare è l'allocazione di *risorse limitate*. La *programmazione lineare* usa modelli matematici, dove lineare indica che le funzioni utilizzate devono essere lineari. Un metodo efficiente per trovare la soluzione è l'***algoritmo del simplesso***.

## 3.1 Prototipo esempio

### 3.1.1 Il problema

Il problema è così formulato:

Schiopparelli & CO produce vetri di alta qualità e ha 3 stabilimenti. Lo scopo è massimizzare il guadagno bilanciando la quantità dei vari prodotti.

- **Stabilimento 1:** Crea gli infissi in alluminio.
- **Stabilimento 2/3:** Creano e assemblano le finestre.

I prodotti sono:

- **Porta-finestra:** Alta  $2,4m$  con infissi in alluminio.  
Necessita *Stabilimento 1 e 3*
- **Doppia finestra:**  $1,2m \times 1,8m$  con infissi in legno.  
Necessita *Stabilimento 2 e 3*

I dati necessari:

- Numero di ore di lavoro disponibili a settimana in ogni stabilimento per ogni prodotto.
- Numero di ore necessarie per assemblare ogni prodotto.
- Profitto per ogni insieme di prodotti

**Modello risultante:**

Stabilimento	Tempi di produzione per lotto		Produzione a settimana ( $h$ )
	Prod. 1	Prod. 2	
1	1	0	4
2	0	2	12
3	3	2	18
<b>Profitto per lotto</b>	\$3,000	\$5,000	

### 3.1.2 Formulazione per la programmazione lineare

Il modello allora per il problema.

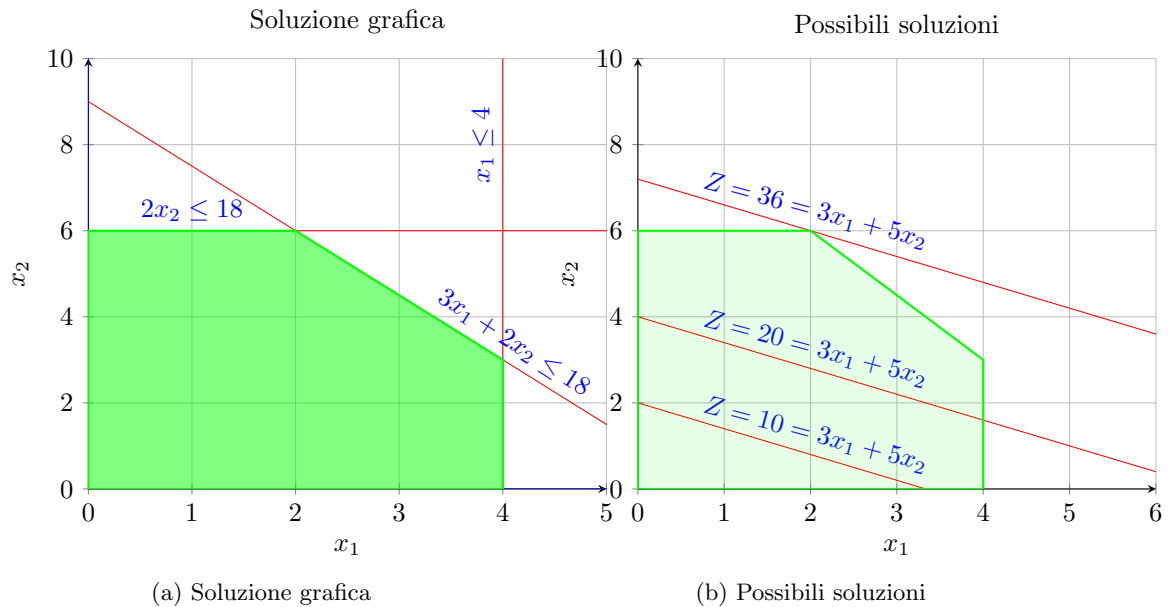
$$\begin{cases} x_1 = \text{N}^\circ \text{ lotti prod. 1 per settimana} \\ x_2 = \text{N}^\circ \text{ lotti prod. 2 per settimana} \\ Z = \text{Profitto totale per settimana (in migliaia di \$)} \end{cases}$$

Quindi  $x_1, x_2$  sono *variabili decisionali* del modello. Dalla tabella precedente si ottiene  $Z = 3x_1 + 5x_2$ . L'obiettivo è trovare i valori  $x_1, x_2$  tale che  $Z$  abbia il valore massimo. Introduciamo ora i vincoli degli orari dei vari stabilimenti. Lo stabilimento 1:  $x_1 \leq 4$ , stabilimento 2:  $2x_2 \leq 12$ , stabilimento 3:  $3x_1 + 2x_2 \leq 18$ , e infine i *limiti di positività*. Si arriva quindi al sistema:

$$\text{Modello} = \begin{cases} x_1 \leq 4 \\ 2x_2 \leq 12 \\ 3x_1 + 2x_2 \leq 18 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ Z = 3x_1 + 5x_2 \end{cases}$$

### 3.2 Soluzione dei grafici

Avendo questo problema due *variabili decisionali* su può usare un grafico bidimensionale. Analizziamo la soluzione passo passo. In *blu* ci sono i vincoli di positività  $x_1 \geq 0$ ,  $x_2 \geq 0$ . In *rosso* il inseriamo i vincoli  $x_1 \leq 4$  e  $2x_2 \leq 12$  e  $3x_1 + 2x_2 \leq 18$ . In *verde* l'area delle possibili soluzioni. Si arriva quindi al punto di scegliere un punto che massimizzi  $Z = 3x_1 + 5x_2$ . Da qui si avanza con il metodo *trial and error* prendendo un valore arbitrario.



Si noti nel grafico a sinistra che tutte le rette sono parallele, infatti sono tutte forme di una funzione chiamata **slope-intercept form**, in questo caso è  $x_2 = -\frac{3}{5}x_1 + \frac{1}{5}Z$ . Nel grafico a destra si può vedere che la soluzione massima sia  $(x_1 = 2, x_2 = 6)$  portando  $Z = 36$ .

**ATTENZIONE:** Nell'esempio fornito si può utilizzare un piano per la scaristà di parametri, ma con il crescere del numero di parametri cresce il numero di dimensioni del grafico. In caso la *regione ammissibile* diventa un poliedro che può essere: *vuoto* in caso non esistano soluzioni ammissibili, *illimitato* in caso in almeno una dimensione non esiste il *valore ottimo*. Esiste poi un almeno un vertice di tale poliedro che corrisponde ad un *valore ottimo*. In ogni caso il poliedro è convesso.

### 3.3 Modello di programmazione lineare

Abbiamo visto come trovare la soluzione graficamente, ora vediamo l'approccio di programmazione lineare. Cominciamo ora con termini necessari e notazione. *Risorse* e *attività*, dove  $m$  indica il numero di risorse utilizzabili e  $n$  il numero di attività considerate. Lo scopo in questo caso è trovare i *livelli* delle attività per avere la *migliore prestazione globale*. Di seguito la lista di simboli e il grafico generici per la programmazione lineare.

- $Z$ : Valore totale delle prestazioni.
- $x_j$ : Livello dell'attività  $j \in [1, n]$ .
- $c_j$ : Incremento in  $Z$  per ogni incremento unitario dell'attività  $j$ .
- $b_i$ : Ammontare delle risorse  $i$  disponibili per l'attività.  $i \in [1, m]$ .
- $a_{ij}$ : Valore della risorsa  $i$  consumata da ogni unità di attività  $j$ .

Risorse usate per Unità di attività					
Risorse	Attività				Quantità di risorse disponibili
	1	2	...	$n$	
1	$a_{11}$	$a_{12}$	...	$a_{1n}$	$b_1$
2	$a_{21}$	$a_{22}$	...	$a_{2n}$	$b_2$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$m$	$a_{m1}$	$a_{m2}$	...	$a_{mn}$	$b_m$
Contributo a $Z$ per unità					
	$c_1$	$c_2$	...	$c_n$	

### 3.4 Forma standard del modello

Dalla tabella precedente si deriva il modello generale (**forma standard**) dal quale si può derivare il modello per il problema che abbia già risolto graficamente.

### 3.4.1 Modello generale

$$\left\{ \begin{array}{l} Z = c_1x_1 + \cdots + c_nx_n \\ a_{11}x_1 + a_{12} + \cdots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22} + \cdots + a_{2n}x_n \leq b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2} + \cdots + a_{mn}x_n \leq b_m \end{array} \right\} \begin{array}{l} \text{Funzione obiettivo, da massimizzare} \\ \\ \text{Vincoli funzionali} \end{array}$$

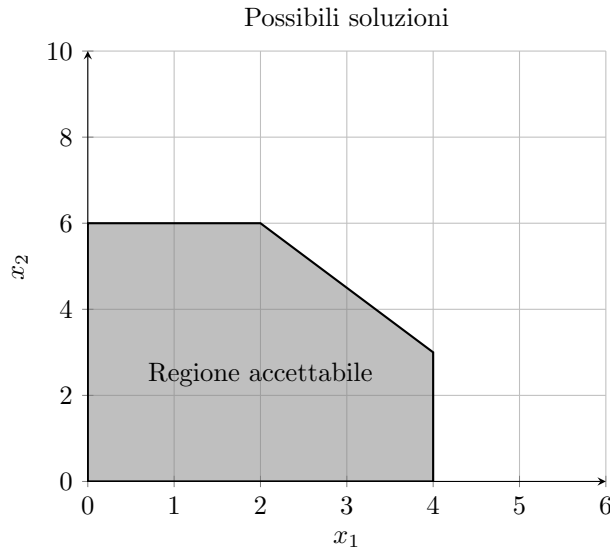
$$\left\{ \begin{array}{l} x_1 \geq 0 \\ \vdots \\ x_n \geq 0 \end{array} \right\} \text{Vincoli di non negatività}$$

alternative:

- Minimizzare la *funzione obiettivo*.
- Rivoltare i *vincoli funzionali* in modo da avere  $\geq$ .
- Porre un *vincolo funzionale* a un'uguaglianza.
- Eliminare *vincoli di non negatività*.

### 3.4.2 Soluzione del modello

Il termine è utilizzato per indicare il metodo di soluzione del modello. Una **soluzione accettabile** è una soluzione dove tutti vincoli sono soddisfatti. Una **soluzione inaccettabile** è una soluzione dove almeno un vincolo non è soddisfatto. La **regione accettabile** è l'insieme di tutte le possibili soluzioni. La **soluzione ottima** è la soluzione accettabile più favorevole alla funzione obiettivo. È possibile che non esista una soluzione. La maggior parte dei modelli ha una soluzione ottima, ma non è impossibile che ne esistano più di una. Potrebbe accadere che non esistano soluzioni ottimali, che succede in caso non esista una soluzione accettabile o se i limiti non prevengono la funzione obiettivo( $Z$ ), al secondo caso ci si riferisce come  $Z$  **svincolata** o **obiettivo svincolato**. Una soluzione che si trova a un angolo della regione accettabile si chiama **soluzione accettabile ad angolo**, (CPF).



### 3.5 Soluzione aritmetica del modello

Metodo per risolvere il modello evitando di utilizzare il grafico. Si parte quindi dal *modello lineare* per portarlo in forma standard e poi risolverlo. Per portare il modello in forma standard tutti i vincoli di *disuguaglianza* devono essere trasformati in *vincoli di uguaglianza* e la funzione obiettivo  $Z$  vengono poste in forma di minimizzazione.

Per poter portare ogni vincolo di *disuguaglianza* ad un vincolo di *uguaglianza* si aggiunge una variabile di *slack* per ogni vincolo in forma di disuguaglianza.

$$\begin{cases} Z = x_1 + 3x_2 - 3x_3 \\ -3x_1 + 2x_2 - 2x_3 \leq 9 \\ -4x_1 - 3x_2 + 3x_3 \leq 8 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases} \rightarrow \begin{cases} Z = x_1 + 3x_2 - 3x_3 \\ -3x_1 + 2x_2 - 2x_3 + x_4 = 9 \\ -4x_1 - 3x_2 + 3x_3 + x_5 = 8 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \\ x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Si passa così da un sistema da  $m$  vincoli e  $n$  variabili ad un modello con  $m$  vincoli e  $n + m$  variabili. Si rimane quindi con una matrice che, a meno di vincoli ridondanti, ha **Rango**  $m$ . Il sistema ha una soluzione univoca se eliminando gli  $n$  gradi di libertà usando  $n$  variabili.

**Teorema di Rouché Capelli** Data una matrice in forma  $A' = [A|b]$  dove  $A$  è una matrice  $m \times n$  e  $b$  un vettore di  $n$  elementi allora:

1. Se il sistema è risolubile la soluzione è unica se, e solo se,  $r(A') = k$  dove  $k$  è il numero di incognite del sistema.
2. Se il sistema è risolubile si hanno  $\infty^j$  soluzioni ( $j$  gradi di infinito) se, e solo se,  $j = k - r(A')$ .



La soluzione poi si trova estrapolando la base( $B$ ) dal modello dove il resto del modello lo chiamiamo  $N$ . Il modello allora lo si può riscrivere in forma  $[B|N]$ . Riscrivo quindi il modello nella forma  $Bx_B + N_{x_N} = b$  dove  $x_B$  e  $x_N$  sono le soluzioni del sistema. La soluzione si ottiene dopo aver posto a 0 tutte le  $n$  variabili che non appartengono a  $B$ . Per farlo si inverte la matrice  $B$ , quindi si ottiene:  $x_B = B^{-1}b - B^{-1}N_{x_N}$  da qui si comprende il motivo di porre  $x_N$  a 0 visto che la soluzione è:  $x_N = 0, x_B = B^{-1}b$ .

### 3.5.1 Degenerazione del modello

La degenerazione si ottiene quando si pone a 0 un parametro interno alla base  $B$ . questo porta ad avere più soluzioni nella base.

### 3.5.2 Teorema fondamentale della PL

Dato il problema lineare in forma standard

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

con  $A \in \mathbb{R}^m \times \mathbb{R}^n$

1. Se esiste una *soluzione ammissibile* esiste anche una soluzione di base
2. Se esiste una *soluzione ottima* esiste una soluzione ottima di base.

## 3.6 Assunzioni sulla programmazione lineare

### 3.6.1 Proporzionalità

Il contributo di ogni attività al *valore della funzione obiettivo*  $Z$  è proporzionale al livello dell'attività  $x_j$  rappresentata da  $c_j x_j$  in  $Z$ . Come il contributo nei vincoli funzionali è proporzionale al livello di  $x_j$  rappresentato da prodotto  $a_{ij} x_j$ . L'assunzione dice dunque esclude ogni esponente diverso da quello neutro(1) per ogni variabile nel modello lineare.

### 3.6.2 Additività

Ogni funzione nel modello lineare è la somma dei contributi individuali delle rispettive attività.

### 3.6.3 Divisibilità

Le variabili di decisione in un modello di programmazione lineare possono avere un qualsiasi valore, (anche non interi) per soddisfare i vincoli. Quindi i valori non sono obbligatoriamente interi, quindi un'attività può essere svolta a *livelli frazionati*.

### 3.6.4 Certezza

Il valore assegnato ad ogni parametro si assume sia una costante conosciuta.

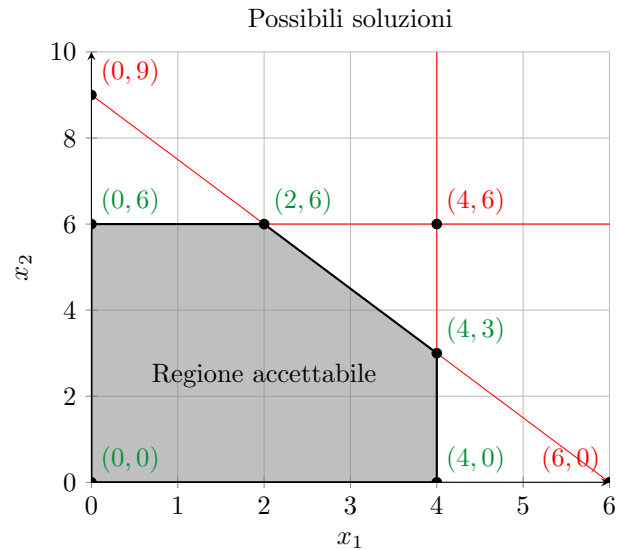
## 4 Risolvere i problemi di PL con l'algoritmo del simplesso:

Arrivati a questo punto si deve introdurre l'*algoritmo del simplesso* una procedura generale per risolvere problemi di programmazione lineare sviluppato da *George Dantzig* nel 1947.

### 4.1 Il concetto

Il metodo del simplesso è un metodo algebrico che si basa su un concetto geometrico. Quindi una prefazione geometrica aiuta a comprendere il concetto sotto. Per evitare di definire un nuovo esempio riprenderemo quello del capitolo precedente di "Schiopparelli & CO".

$$\text{Modello} = \begin{cases} x_1 \leq 4 \\ 2x_2 \leq 12 \\ 3x_1 + 2x_2 \leq 18 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ Z = 3x_1 + 5x_2 \end{cases}$$



(a) Modello del problema "Schiopparelli & CO"

(b) Soluzioni grafiche

Si noti come i punti evidenziati (CPF) non siano altro che l'intersezione di due *vincoli*, alcuni sono al limitare e altri no, rispettivamente quelli *verdi* e quelli *rossi*, se ora questa affermazione lascia il tempo che trova più tardi sarà importante.

**Definizione:**

Per ogni problema lineare con  $n$  variabili di decisione due CPF sono *adiacenti* se condividono  $n - 1$  vincoli. Due CPF adiacenti sono uniti da un segmento sulle dimensioni dei vincoli condivisi. Ci si riferisce a questo segmento come: **Spigolo** della regione accettabile.

Per la soluzione si usa il **test di ottimalità**:

**Test di ottimalità:** Considerando un qualsiasi problema di programmazione lineare che possiede almeno una soluzione ottimale. Se una soluzione CPF non ha *adiacenze* migliori allora quella soluzione è la soluzione migliore.

## 4.2 Risolvere il problema d'esempio

Dato che nel nostro problema  $n = 2$  i CPF adiacenti devono condividere un vincolo.

CPF	Adiacenze
(0, 0)	(0, 6), (4, 0)
(0, 6)	(2, 6), (0, 0)
(2, 6)	(4, 3), (0, 6)
(4, 3)	(4, 0), (2, 6)
(4, 0)	(0, 0), (4, 3)

### Inizializzazione:

Prendo (0, 0) come soluzione CPF da esaminare, non c'è un vero e proprio criterio per scegliere in questo caso era solo il più semplice.

#### ① Test di ottimalità:

**Fallito**, non è ottimale, esiste una soluzione migliore adiacente.

**Iterazione 1:** Mi muovo verso (0, 6) usando i seguenti step.

1. Considero(nella tabella) i due CPF adiacenti (0, 6) e (4, 0), da cui vien fuori: (0, 6) :  $Z = 3 \cdot 0 + 6 \cdot 5 = 30$  e (4, 0) :  $Z = 3 \cdot 4 + 5 \cdot 0 = 12$ . Ora prendo il valore che massimizza  $Z$ , quindi (0, 6) mi muoverò nella direzione di  $x_2$  fino a trovare il primo CPF.
2. Mi fermo all'incontrare primo vincolo incontrato:  $2x_2 \leq 12$  per evitare di uscire dalla *regione accettabile*.
3. Risolvo l'intersezione del nuovo insieme di vincoli.

#### ② Test di ottimalità:

**Fallito**, non è ottimale, esiste una soluzione migliore adiacente.

**Iterazione 2:** Mi muovo verso (2, 6) usando i seguenti step.

1. Considero(nella tabella) i due CPF adiacenti (2, 6) e (0, 0), da cui vien fuori: (0, 0) :  $Z = 3 \cdot 0 + 6 \cdot 0 = 0$  e (2, 6) :  $Z = 3 \cdot 2 + 5 \cdot 6 = 36$ . Ora prendo il valore che massimizza  $Z$ , quindi (2, 6) mi muoverò nella direzione di  $x_1$  fino a trovare il primo CPF.
2. Mi fermo all'incontrare primo vincolo incontrato:  $3x_1 + 2x_2 \leq 18$  per evitare di uscire dalla *regione accettabile*.
3. Risolvo l'intersezione del nuovo insieme di vincoli.

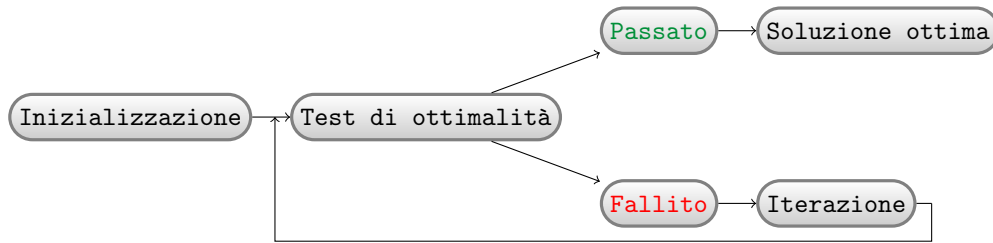
#### ③ Test di ottimalità:

**Passato**, è ottimale, non esiste una soluzione migliore adiacente. Ho trovato la soluzione migliore.

### 4.2.1 Concetti della soluzione:

1. Il simpleso si concentra esclusivamente sui CPF perché se esiste una soluzione migliore ne esiste una che coincide con un CPF.

2. Il simplesso è algoritmo *iterativo*.



3. Se possibile per l'inizializzazione si sceglie l'origine in modo da eliminare entropia iniziale.
4. Computazionalmente è più efficiente confrontare il CPF adiacenti.
5. Dopo aver trovato un CPF controlla se ne esiste uno adiacente migliore, cerca per ogni parametro quale deve essere modificato per incrementare di più la funzione obiettivo. O per meglio dire quale incrementa di più il *rateo di incremento di Z*

### 4.3 Il metodo del simplesso

Per quanto graficamente sia più facile da comprendere questo metodo non si può tradurre computazionalmente, per poterlo rendere computabile lo si deve formalizzare aritmeticamente. Come nel capitolo precedente per la soluzione aritmetica si pone il modello in forma aumentata.

$$\begin{cases} Z = 3x_1 + 5x_2 \\ x_1 \leq 4 \\ 2x_2 \leq 12 \\ 3x_1 + 2x_2 \leq 18 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \rightarrow \begin{cases} Z = 3x_1 + 5x_2 \\ x_1 + x_3 = 4 \\ 2x_2 + x_4 = 12 \\ 3x_1 + 2x_2 + x_5 = 18 \\ -3x_1 + 2x_2 - 2x_3 + x_4 = 9 \\ -4x_1 - 3x_2 + 3x_3 + x_5 = 8 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Si può notare dal nuovo modello che se una delle variabili di slack (*in rosso*) sono poste a zero allora la soluzione si sul limite di un vincolo. Un valore maggiore di 0 invece lo porta nella *regione accettabile* e un valore negativo lo porta nella *regione inaccettabile*. Per meglio comprendere viene comodo un esempio, ma prima si deve introdurre un po di terminologia.

- **Soluzione aumentata:**

Soluzione per le variabili originali, *di decisione*, che devono essere aumentate dal valore corrispondente della *variabile di slack*.

#### Esempio

Una soluzione normale sarebbe:  $(x_1 : 3, x_2 : 2) \rightarrow (x_1 : 3, x_2 : 2, x_3 : 1, x_4 : 8, x_5 : 5)$

- **Soluzione basica:**

È una soluzione CPF aumentata che può trovarsi all'interno o all'esterno della regione accettabile, che porta al concetto di *soluzione basica accettabile* una soluzione basica ha le seguenti proprietà:

- 1.
- 2.
- 3.
- 4.
- 5.

- **Soluzione basica accettabile:**

Come da nome una *soluzione basica* ma con il vincolo di trovarsi nella regione accettabile.

- **Gradi di libertà:**

Differenza tra il numero di variabili, (*variabili di decisione* + *variabili di slack*) e il numero di *vincoli funzionali*. Indica il numero di variabili che possono essere poste ad un valore arbitrario, 0, per poter ottenere una soluzione per le variabili rimanenti.