ESERCIZI PRIMALE-DUALE

$$\min\ 2x_1 - x_2$$
s.t.
u_1: $x_1 + 2x_2 \geq 7$
u_2: $2x_1 - x_2 \geq -6$
u_3: $-3x_1 + 2x_2 \geq 8$
D: $x_1 \geq 0 \quad x_2 \geq 0$

(colonne K_1, K_2; termini noti = f.o. del duale)

$[x_1, x_2] = [0, 6]$ ottima

[con $x_1 + 2x_2 = 7$ aggiungo lo stesso un u_i nei vincoli primali/duali, scrivo "non ci sono da imporre condizioni di complementarietà con la relativa variabile u_i (nei primali) / x_i (nei duali)"]

1. Verifica ammissibilità — sostituisco valori dati
   - $0 + 12 \geq 7$ OK
   - $0 - 6 \geq -6$ OK
   - $0 + 12 \geq 8$ OK
   - $0 \geq 0$ OK $\quad 6 \geq 0$ OK

2. Passaggio al duale — f.o., K e dominio tramite tabella

$$\max\ 7u_1 - 6u_2 + 8u_3$$
D: $u_1 \geq 0 \quad u_2 \geq 0 \quad u_3 \geq 0$
K_1: $u_1 + 2u_2 - 3u_3 \leq 2$
K_2: $2u_1 - u_2 + 2u_3 \leq -1$

3. Applicazione delle condizioni — se ho "$\neq 0$" allora lo tengo come condizione ed elaboro, se "$= 0$" lo scarto

3.1 Vincoli primali

$u_1(x_1 + x_2 - 7) = 0 \Rightarrow u_1(\underbrace{0+6-7}_{\neq 0}) = 0 \Rightarrow u_1 = 0$ prima condizione

$u_2(2x_1 - x_2 + 6) = 0 \Rightarrow u_2(\underbrace{0-6+6}_{=0}) = 0 \Rightarrow$ non è possibile dedurre alcuna condizione

$u_3(-3x_1 + 2x_2 - 8) = 0 \Rightarrow u_3(0 + 12 - 8) = 0 \Rightarrow u_3 = 0$ seconda condizione

3.2 Vincoli duali

$x_1(u_1 + 2u_2 - 3u_3 - 2) = 0 \Rightarrow \underbrace{0}_{=0} \cdot (u_1 + 2u_2 - 3u_3 - 2) = 0 \Rightarrow$ non è possibile dedurre alcuna condizione

$\underbrace{x_2}_{\neq 0}(2u_1 - u_2 + 2u_3 + 1) = 0 \Rightarrow 2u_1 - u_2 + 2u_3 = -1$ terza condizione

4. Sistema di equazioni per l'impostazione delle condizioni di complementarietà

$$\begin{cases} u_1 = 0 \\ u_3 = 0 \\ 2u_1 - u_2 + 2u_3 = -1 \end{cases} \Rightarrow \begin{cases} u_1 = 0 \\ u_3 = 0 \\ u_2 = 1 \end{cases}$$

5. Verifica ammissibilità duale — riscrivere esattamente i sottopunti
   - soddisfa i vincoli di dominio $(u_1, u_2, u_3 \geq 0)$ — scrivere il dominio
   - soddisfa i due vincoli duali $\begin{pmatrix} u_1 + 2u_2 - 3u_3 = 2 \leq 2 \\ 2u_1 - u_2 + 2u_3 = -1 \leq -1 \end{pmatrix}$
   - è in scarti complementari con la soluzione primale data (per costruzione)

Pertanto, le due soluzioni sono ottime per i rispettivi problemi primale e duale.

[Per verifica, i valori delle funzioni obiettivo sono uguali: $2x_1 - x_2 = -6$ e $7u_1 - 6u_2 + 8u_3 = -6$, il che verifica il Corollario 2 della dualità (all'ottimo, i valori delle funzioni obiettivo primale e duale coincidono).]

CONDIZIONI DI COMPLEMENTARIETÀ

Data la coppia di problemi:
$\min\{c^T x : x \geq 0, Ax \geq b\}$
$\max\{u^T b : u \geq 0, u^T A \leq c^T\}$

x e u ottime $\Leftrightarrow$
- $Ax \geq b \ \wedge\ x \geq 0$ AMMISS. PRIMALE
- $u^T A \leq c^T \ \wedge\ u \geq 0$ AMMISS. DUALE
- $u^T(Ax - b) = 0$, $(c^T - u^T A)x = 0$ } ORTOGONALITÀ

DUALITÀ FORTE

Se un problema PL ha una sol. ottima finita, il corrispondente problema duale ha sol. ottima finita e i valori coincidono. Se chiede di risolvere il simplesso per problema di PL e poi chiede qualcosa sulla sol. duale, le sol. sono uguali per questo teorema.

DUALITÀ DEBOLE

Se ho una sol. x ammissibile per il problema primale e una sol. u ammissibile per il problema duale. Se $x = u$ allora x ed u sono ottime per il primale e il duale rispettivamente.

GRAFI Scelta dell'algoritmo

- **Bellman-Ford** quando si ha un massimo numero di archi/hop (anche se i costi sono tutti positivi). Posso applicare solo l'algoritmo di Bellman-Ford perché è l'unico che dà la possibilità di calcolare i cammini minimi con il massimo numero di archi. Infatti, è possibile dimostrare che, all'iterazione k dell'algoritmo, le etichette corrispondono ai cammini minimi che utilizzano al più k archi. Applicherò quindi Bellman Ford fermandomi alla k-esima iterazione, dopo l'inizializzazione.
- **Bellman-Ford** quando si hanno costi ridotti negativi (archi negativi). Si sceglie l'algoritmo di Bellman-Ford in quanto esistono archi con costo negativo. Bellman-Ford è l'unico algoritmo visto in grado di garantire convergenza alla soluzione ottima del problema dei cammini minimi in presenza di archi di costo negativo, sebbene mediamente meno efficiente dell'algoritmo di Dijkstra, che non garantisce di trovare la soluzione al problema dei cammini minimi se esistono archi di costo negativo.
- **Dijkstra** se i costi ridotti tutti positivi e non ci sono max hop, perché più efficiente computazionalmente. Essendo tutti i costi sugli archi non negativi, possiamo applicare l'algoritmo di Dijkstra, il più efficiente tra quelli visti.

---

if $x = 0$ then $y = 1 \Rightarrow y \geq 1 - x$
$\neg P_1 \Rightarrow y_1 = 0$
$\neg(P_1 \vee P_2) \Rightarrow y_1 = 0,\ y_2 = 0$
$P_1 \Rightarrow (P_2 \wedge P_3) \Rightarrow y_1 \leq y_2,\ y_1 \leq y_3$
$P_1 \Rightarrow (P_2 \vee P_3) \Rightarrow y_1 \leq y_2 + y_3$

$(P_1 \wedge P_2) \Rightarrow P_3 \Rightarrow y_1 + y_2 \leq y_3 + 1$
$(P_1 \vee P_2) \Rightarrow P_3 \Rightarrow y_1 \leq y_3,\ y_2 \leq y_3$
$P_1 \wedge (P_2 \vee P_3) \Rightarrow y_1 = 1,\ y_2 + y_3 \geq 1$
$P_1 \vee (P_2 \wedge P_3) \Rightarrow y_1 + y_2 \geq 1,\ y_1 + y_3 \geq 1$
$P_1 \Leftrightarrow P_2 \Rightarrow y_1 = y_2$

- SPECIFICARE I DOMINI
- ATTIVARE LE VAR. LOGICHE

SBAGLIATO formulazioni NON LINEARI!!!			Corretto Vincoli	Corretto Domini
if $x_1 > 0$ then $x_2 = 0$ and if $x_2 > 0$ then $x_1 = 0$	$nand(y_1, y_2)$	$x_1 x_2 = 0$ $y_1 y_2 = 0$	$x_1 \leq M y_1$ $x_2 \leq M y_2$ $y_1 + y_2 \leq 1$	$x_1, x_2 \geq 0$ $y_1, y_2 \in \{0,1\}$ $(M \to \infty)$
if $x > 0$ then $y = 1$	$(x > 0) \to (y = 1)$	$x(1 - y) = 0$	$x \leq My$	$x \geq 0, y \in \{0,1\}$
$y_1 = 1$ or $y_2 = 1$	$y_1 \vee y_2$	$(1 - y_1)(1 - y_2) = 0$	$y_1 + y_2 \geq 1$	$y_1, y_2 \in \{0,1\}$
$y_1 = 1$ and $y_2 = 1$	$y_1 \wedge y_2$	$y_1 y_2 = 1$	$y_1 + y_2 = 2$	$y_1, y_2 \in \{0,1\}$
$y_1 = 1$ only if $y_2 = 1$	$y_1 \to y_2$ $= (\neg P_1) \vee P_2$	$y_1(1 - y_2) = 0$	$y_1 \leq y_2$	$y_1, y_2 \in \{0,1\}$
$y_1 = 1$ only if $y_2 = 0$	$y_1 \to \overline{y_2}$ $= \neg(P_1 \wedge P_2)$	$y_1 y_2 = 0$	$y_1 \leq (1 - y_2)$	$y_1, y_2 \in \{0,1\}$
$y_1 = 1$ xor $y_2 = 1$	$y_1 \neq y_2$		$y_1 + y_2 = 1$	$y_1, y_2 \in \{0,1\}$

ESERCIZIO PL PIVOT

$$\max\ (+)2x_1 + 3x_2 - x_3 + 3x_4$$
s.t.
$x_1 - x_2 + 2x_3 + 2x_4 \geq -2$
$2x_1 + 2x_2 + x_3 - 4x_4 \leq 4$
$x_1 - x_2 + 2x_3 + \quad \leq 2$
$x_1, x_2, x_3 \geq 0$
$x_4 \leq 0$

calcoli di aggiustamento
es. $3x_p \leq 18 \Rightarrow x_p \leq 6$

1. FORMA STANDARD — max → min (cambio segni)

DA FARE TUTTI ASSIEME PER VELOCIZZARE

1.1 f.o. $\min\ -2x_1 - 3x_2 + x_3 - 3x_4$

1.2 vincoli di uguaglianza — variabili di slack $(\leq [+])(\geq [-])$

s.t.
$x_1 - x_2 + 2x_3 + 2x_4 - x_5 = -2$
$2x_1 + 2x_2 + x_3 - 4x_4 + x_6 = 4$
$x_1 - x_2 + 2x_3 + x_7 = 2$

1.3 variabili non negative $(\leq 0) \Rightarrow [\hat{x}_i = -x_i,\ \hat{x}_i \geq 0]$ e 1.4 Termini noti non negativi — se il termine noto è negativo cambio segno a tutta la disuguaglianza

$\hat{x}_4 = -x_4$ con $\hat{x}_4 \geq 0$ — mantengo il segno, aggiungo solo il cappello

$\min\ -2x_1 - 3x_2 + x_3 + 3\hat{x}_4$
s.t.
$-x_1 + x_2 - 2x_3 - 2\hat{x}_4 + x_5 = 2$
$2x_1 + 2x_2 + x_3 - 4\hat{x}_4 + x_6 = 4$
$x_1 - x_2 + 2x_3 + x_7 = 2$
$x_1, x_2, x_3, \hat{x}_4, x_5, x_6, x_7 \geq 0$

2. Inizializzazione base ammissibile — mettere in forma canonica rispetto alla base data/scelta

Avendo aggiunto le variabili di slack (x_5, x_6, x_7) disponiamo di una base ammissibile di partenza $B = \{x_5, x_6, x_7\}$.
Il problema è già in forma canonica rispetto alla base B.
Organizzo i dati in forma tableau:

per ordine posizionale

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	z	$\bar{b}$	
$-z$	-2	-3	1	3	0	0	0	-1	0	$R_0 \leftarrow R_0 + R_2$
x_5	-1	1	-2	-2	1	0	0	0	2	$R_1 \leftarrow R_1 + \frac{R_2}{2}$
x_6	2	2	1	-4	0	1	0	0	4	$R_2 \leftarrow \frac{R_2}{2}$
x_7	1	-1	2	0	0	0	1	0	2	$R_3 \leftarrow R_3 - \frac{R_2}{2}$

devono diventare zeri tranne l'incrocio (1)

in base: $x_B = \begin{bmatrix} x_5 \\ x_6 \\ x_7 \end{bmatrix}$ $\quad x_F = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \hat{x}_4 \end{bmatrix}$ fuori base

Iterazione 1

base entrante: x_1
base uscente: $\operatorname{argmin}\{\cancel{-2}, 2, 2\} = x_6$ — no <0 or $\frac{k}{0}$
pivot sull'elemento in riga 1 e colonna 2

Iterazione 2

	x_1	x_2	x_3	$\hat{x}_4$	x_5	x_6	x_7	z	$\bar{b}$	
$-z$	0	-1	2	-1	0	1	0	-1	4	$R_0 \leftarrow R_0 + R_2$
x_5	0	2	-3/2	-4	1	1/2	0	0	4	$R_1 \leftarrow R_1 - 2R_2$
x_1	1	1	1/2	-2	0	1/2	0	0	2	$R_2 \leftarrow R_2$
x_7	0	-2	3/2	2	0	-1/2	1	0	0	$R_3 \leftarrow R_3 + 2R_2$

base entrante: x_2
base uscente: $\operatorname{argmin}\{2, 2, \cancel{\tfrac{0}{2}}\} = x_1$ — per indice posizionale minore

Iterazione 3

	x_1	x_2	x_3	$\hat{x}_4$	x_5	x_6	x_7	z	$\bar{b}$
$-z$	1	0	5/2	-3	0	3/2	0	-1	6
x_5	-2	0	-5/2	0	1	-1/2	0	0	0
x_2	1	1	1/2	-2	0	1/2	0	0	2
x_7	2	0	5/2	-2	0	1/2	1	0	4

$x_B = \begin{bmatrix} x_5 \\ x_2 \\ x_7 \end{bmatrix}$ $\quad x_F = \begin{bmatrix} x_1 \\ x_3 \\ \hat{x}_4 \\ x_6 \end{bmatrix}$

da riscrivere con $x_i = -\hat{x}_i$

scegliere se la sol soddisfa (o non) le condizioni di ottimalità e se il problema è illimitato.

3. Conclusioni
- soluzione $x = (x_1, x_2, x_3, \hat{x}_4, x_5, x_6, x_7) = (0, 2, 0, 0, 0, 0, 4)$ di valore $z = -6$ — attenzione alla $(-z)$
- Nel tableau è presente un costo ridotto negativo quindi la base B non soddisfa le condizioni di ottimalità.
- La colonna relativa al costo ridotto negativo è tutta non positiva, quindi il problema è illimitato.

[
- è in FORMA CANONICA rispetto alla base corrente? colonne, anche sparse/non in ordine, della matrice identità nel tableau rispetto agli elementi della base
- AMMISSIBILE? La colonna dei termini noti $\bar{b}$ deve essere positiva
- OTTIMA? Non ho costi ridotti nella prima riga
- ILLIMITATA? ho una colonna tutta negativa in corrispondenza di un costo negativo
- ENTRA in BASE → variabile con costo negativo e indice minore (anticiclo di Bland)
- ESCE dalla BASE → l'argmin del rapporto tra b e i coeff. della var che entra
]

[
- Se il simplesso risulta in soluzione ottima, allora il problema duale avrà una soluzione ottima identica secondo il corollario della dualità forte.
- Se il simplesso risulta in illimitatezza, allora il problema duale sarà inammissibile per il corollario della dualità debole
- Base degenere quando ci sono almeno due righe con lo stesso rapporto minimo, risulterà cioè che almeno una esce dalla base con valore 0, mentre le altre assumeranno valore 0 rimanendo in base.
- Il valore della f.o. al cambio base = (rapporto minimo)*(costo ridotto del pivot)+(b*z)
  $\Rightarrow z_{base} = \frac{b_{pivot}}{pivot} \cdot c_{pivot} + z_{minimo} \cdot b_{sol}$
]

Primale min (x)	Duale max (u)	Primale max (x)	Duale min (u)
$x_i \geq 0$	condizione $\leq c_i$	$x_i \geq 0$	cond $\geq c_i$
$x_i \leq 0$	cond $\geq c_i$	$x_i \leq 0$	cond $\leq c_i$
x_i libera	cond $= c_i$	x_i libera	cond $= c_i$
cond $\geq b_j$	$u_i \geq 0$	cond $\geq b_j$	$u_i \leq 0$
cond $\leq b_j$	$u_i \leq 0$	cond $\leq b_j$	$u_i \geq 0$
cond $= b_j$	u_i libera	cond $= b_j$	u_i libera

BELLMAN-FORD

max # iterazioni = n
comp. comp. $O(m \cdot n)$
[n = # nodi, m = # archi]

- Per le **iterazioni successive** controllo il nodo modificato e a quali nodi è collegato e verifico se il peso migliora (se migliora cambio, altrimenti lascio il valore precedente).
- Si **interrompe** o quando raggiungo il **max # iterazioni** o quando compare 2 volte di fila la **stessa riga** nella tabella.

Iterazione	A	B	C	D	Aggiornati
0	0_A	$+\infty$	$+\infty$	$+\infty$	A
1	0_A	1_A	$+\infty$	$+\infty$	A, B
–	–	–	–	–	–
4	0_A	-1_D	2_B	3_C	A, B, C, D

Prendere nodi tornando indietro e salendo di riga ogni volta
N.B. Fare affidamento ai numeri della riga precedente

GRAFO, ALBERO, CICLO NEGATIVO

- Se ho un **ciclo negativo** non è possibile fare **né albero né grafo**.
- Per **trovare il ciclo negativo** si osserva quale variabile viene modificata nell'ultima iterazione e si prosegue a ritroso, seguendo le indicazioni da dove arriva finché non si chiude il ciclo, oppure semplicemente quando, all'ultima iterazione di controllo, almeno un nodo viene aggiornato.
- Per individuare l'**albero** si prendono i valori dell'ultima riga della tabella
- Per identificare il **grafo** si ricopia il grafo iniziale con anche i cammini minimi dei nodi, quindi si osservano i percorsi che sommati risultano lo stesso peso del nodo

- non posso parlare di albero dei cammini minimi se si parla di max # archi.
- se **lista nodi aggiornati è vuota**, allora le etichette sono stabili ed è possibile determinare albero/grafo dei cammini minimi. Altrimenti no

Indicazioni teoriche (da scrivere)

Si riporta una tabella che riporta una riga per ogni iterazione dell'algoritmo.
Ogni colonna della tabella è dedicata ad un nodo e riporta, iterazione dopo iterazione, l'evoluzione delle rispettive etichette. L'ultima colonna riporta i nodi aggiornati nel corso dell'iterazione: all'iterazione successiva è sufficiente controllare solo gli archi uscenti da questi nodi (vincolo di Bellman).
La tabella riporta la riga 0 (inizializzazione) e una riga per ogni iterazione. All'iterazione h si controllano gli archi (i,j) uscenti da ciascun nodo della colonna Aggiornati alla riga h-1, e si aggiornano i costi e i predecessori del nodo j all'iterazione h qualora l'etichetta del nodo i all'iterazione h-1 più il costo dell'arco (i,j) sia strettamente minore dell'etichetta corrente del nodo j

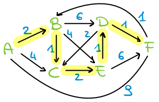
Dijkstra

comp. comp. $O(n^2)$ oppure $O(m \cdot n\log)$, con coda di priorità: $O(m + n\log n)$

Nella tabella di sviluppo:
- S: sono i nodi da verificare
- $\hat{v}$: è il nodo che si sta verificando.
- $*$: messo dopo che si è verificato un nodo
- : se non è controllato si lascia vuoto
- $-$: se non migliora
- $\times$: se il nodo è già chiuso e viene ricontrollato

L'algoritmo **termina** quando non ci sono più nodi in $\bar{S}$

iter.	A	B	C	D	E	F	$\bar{S}$	u
h=0	0	$+\infty_A$	$+\infty_A$	$+\infty_A$	$+\infty_A$	$+\infty_A$	A,B,C,D,E,F	–
h=1	*	2_A	4_A			9_A	B,C,D,E,F	A
h=2		*	3_B	8_B	6_B	–	C,D,E,F	B
h=3			*	–	5_C	–	D,E,F	C
h=4				6_E	*	–	D,F	E
h=5				*		7_D	F	D
h=6						*	–	F



ATLPL

$$\min \sum_{i\in I, j\in J} C_{ij}x_{ij} - \sum_{i\in I} F_i y_i$$
$$\text{s.t.} \sum_{i\in I} x_{ij} \geq D_j \quad , \forall j\in J$$
$$\sum_{j\in J} x_{ij} \leq O_i y_j \quad , \forall i\in I$$
$$\sum_{i\in I} y_i \leq N$$
$$x_{ij}\in \mathbb{Z}_+ ,\ y_i \in\{0,1\} \ , \forall i\in I,\ j\in J$$

.mod

```
set I;            set J;
param O{I};   param D{J};  param C{I,J};  param F{I};  param N;
var x{I,J} >=0 integer;    var y{I} binary;
minimize/maximize fo: sum{i IN I, j IN J} C[i,j]* x[i,j] -
                      sum{i IN I}F[i]* y[i];

s.t. d{j IN J}: sum{i IN I} x[i,j] >= D[j];
s.t. o{i IN I}: sum{j IN J} x[i,j] <= O[i]*y[i];
s.t. n: sum{i IN I} y[i] <= N;
```

.dat

```
set I := Croazia Svezia Bretagna Canada;
set J := Italia Francia Germania;
param:    F    O  :=  Croazia  1000 1000  Svezia  1000 2000
                      Bretagna 1000 3000  Canada  1000 4000;
param D := Italia 5000 Francia 3000 Germania 2000;
param N := 4;
param C: Italia Francia Germania := Croazia   10  20  30
                                    Svezia    40  50  60
                                    Bretagna  70  80  90
                                    Canada   100 110 120;
```

accanto ciò so cui lavorano le variabili: F(I) e O(I)
J •
I • •
singolo elemento

.run

```
reset;
model ampl.mod;
data ampl.dat;
option solver cplexamp;
solve;
display fo,x,y;
```

può essere di vari tipi (cplex, ...)

1. Variabili decisionali (1 o 2 indici)
   "indipendente da" significa indice singolo
2. F.O.
3. Vincoli - almeno uno fra i primi è importante
4. Resto dei vincoli
   ↳ Logici (binari)
   - legarli alle variabili normali (attivazione) solitamente con bigM: $x_i \leq M y_i$

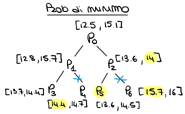
   ↳ casi notevoli:
   - COSTI FISSI ⇒ variabile apposita per il costo fisso
   - ALMENO ⇒ insieme var logiche ≥ numero; ⇒ $x_i \geq M \cdot y_i$ oppure var decis. ≥ numero
   - AL MASSIMO ⇒ insieme var logiche ≤ numero; ⇒ $x_i \geq M \cdot y_i$ oppure var decis. ≤ numero
   - ESATTAMENTE ⇒ insieme var logiche = numero; ⇒ $x_i \geq$ numero $\cdot y_i$
   - BUDGET ⇒ var. decisionali ≤ budget
   - CAPACITÀ ⇒ var decisionali ≤ cap. minima; var decisionali ≥ cap. massima

BRANCH & BOUND

Tabella da seguire per rispondere ad ogni domanda:

	min [LB; SA]	max [SA; UB]
min o max	LB sempre crescente	UB sempre decrescente
Chiusura nodi	LB ≥ SA più bassa	UB ≤ SA più alta
Intervallo ottimo	[min LB figli; min SA* generale]	[max SA* generale; max UB figli]
Nuovo nodo	LB = SA	SA = UB
Valori ottimi	LB padre ≤ SA ≤ LB min**	UB max** ≤ SA ≤ UB padre

Bab di minimo



* non è detto che la SA sia ottima se ci sono nodi con LB più basso (min) o UB più alto (max)
** nodi aperti senza considerare il padre

	min	max
nodo svilupp. per primo:	LB minimo	UB massimo

Scelta del nodo da sviluppare: Best Bound First (BBF) prende il nodo aperto più promettente (LB più basso in min, UB più alto in max), mentre il Depth First prende il nodo aperto al livello più profondo.

		DUALE		
		finito	illimitato	impossibile
PRIMALE	finito	si	no	no
	illimitato	no	no	si
	impossibile	no	si	si

MODELLI DI TRASPORTO

$$\min \sum_{i\in I}\sum_{j\in J} C_{ij}x_{ij}$$
$$\text{s.t.} \sum_{j\in J} x_{ij} \leq O_i \quad \forall i\in I$$
$$\sum_{i\in I} x_{ij} \geq D_j \quad \forall j\in J$$
$$x_{ij}\in \mathbb{R}_+ [\mathbb{Z}_+ \mid \{0,1\}] \quad \forall i\in I,\ j\in J$$

- I: insieme dei centri di offerta
- J: insieme dei centri di domanda
- C_{ij}: costo (unitario) per il trasporto da $i\in I$ a $j\in J$
- O_i: ammontare dell'offerta in $i\in I$
- D_j: ammontare della domanda in $j\in J$

es. trasporto di frigoriferi

MIX OTTIMO DI PRODUZIONE

$$\max \sum_{i\in I} P_i x_i$$
$$\text{s.t.} \sum_{i\in I} A_{ij}x_i \leq Q_j \quad \forall j\in J$$
$$x_i \in \mathbb{R}_+ [\mathbb{Z}_+ \mid \{0,1\}] \ \forall i\in I$$

- I: insieme dei beni che possono essere prodotti
- J: insieme delle risorse disponibili
- P_i: profitto (unitario) per il bene $i\in I$
- Q_j: q.tà disponibile della risorsa $j\in J$
- A_{ij}: q.tà di risorsa j necessaria per la produzione di una unità del bene i

es. assemblaggio telefonini e contadino

MODELLO MULTIPERIODALE

L'effetto delle variabili in vincoli precedenti "ritorna" in quelli successivi.
es. piani di investimento

- anno 1: $x_{A1} + x_{B1} \leq 10000$
- anno 2: $x_{A2} + x_{B2} + x_{C2} \leq 10000 - x_{A1} - x_{B1}$
- anno 3: $x_{A3} + x_{B3} \leq 10000 + 0.4x_{A1} - x_{B1} - x_{A2} - x_{B2} - x_{C2}$
- anno 4: $x_{A4} \leq 10000 + 0.4x_{A1} + 0.7x_{B1} + 0.4x_{A2} - x_{B2} - x_{A3} - x_{C2} - x_{B3}$
- anno 5: $x_{D5} \leq 10000 + 0.4x_{A1} + 0.7x_{B1} + 0.4x_{A2} + 0.7x_{B2} + 0.4x_{A3} - x_{C2} - x_{B3} - x_{A4}$

Dominio: $x_{ij}\in \mathbb{R}_+, \forall i\in\{A,B,C,D\}, \forall j\in\{1,2,3,4\}$

COPERTURA DI COSTO MINIMO

$$\min \sum_{i\in I} C_i x_i$$
$$\text{s.t.} \sum_{i\in I} A_{ij}x_i \geq D_j \quad \forall j\in J$$
$$x_i\in \mathbb{R}_+ [\mathbb{Z}_+ \mid \{0,1\}] \quad \forall i\in I$$

- I: insieme delle risorse da acquistare
- J: insieme delle domande da coprire
- C_i: costo (unitario) per l'utilizzo della risorsa $i\in I$
- D_j: ammontare della domanda di $j\in J$
- A_{ij}: capacità (unitaria) della risorsa i di soddisfare la domanda j

es. dieta economica, indagine di mercato, turni in ospedale, localizzazione dei servizi