UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Enrico Lacchin

Enrico Lacchin

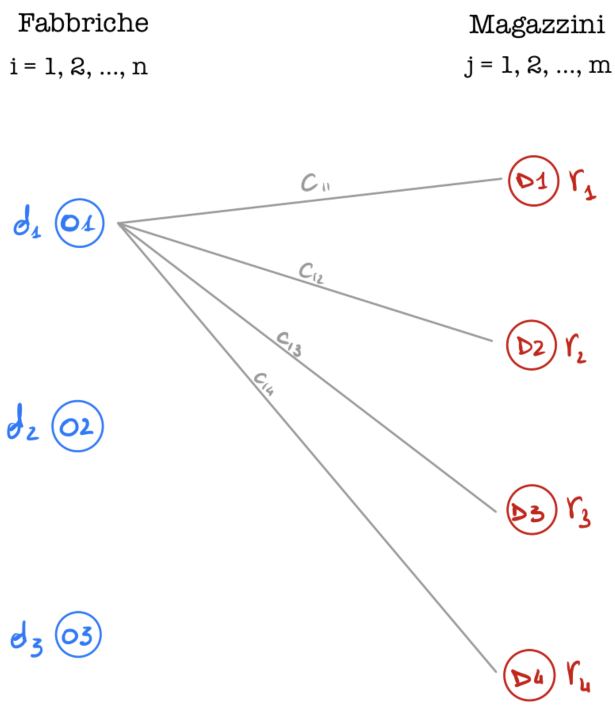
Logistica

Appunti

Materia: Logistica
Docente: Luca Coslovich

Indice

1 Problema del Trasporto



1.1 Formulazione Matematica

Variabili decisionali

$$x_{ij} = \text{quantità trasportata da } O_i \text{ a } D_j$$

Funzione Obiettivo

$$min\ z = \sum_{i}^{n} \sum_{j}^{m} x_{ij} \cdot c_{ij}$$

Vincoli

- $\sum_j x_{ij} \leq d_i, \quad \forall i = 1, \ldots, n$
- $\sum_i x_{ij} \geq r_j, \quad \forall j = 1, \ldots, n$
- $x_{ij} \geq 0, \quad \forall i, j$

Nota Bene:

- Se $\sum_i d_i \leq \sum_j r_j$ il problema NON è ammissibile
- Se $\sum_i d_i \geq \sum_j r_j$ il problema è ammissibile

- Se $\sum_i d_i > \sum_j r_j$ il problema è ammissibile, ma aggiungo un magazzino D_{m+1} nel quale la sua richiesta $r_{m+1} = \sum_i d_i - \sum_j r_j$ e il suo costo $c_{ij} = 0$

Un problema del trasporto è costituito da $N \times M$ variabili, $N + M$ vincoli funzionali (di cui $N + M - 1$ indipendenti tra loro) e $N \times M - (N + M - 1) = (N - 1)(M - 1)$ variabili fuori base.

1.2 Proprietà: Totale Unimodularità

Siano tutte le disponibilità intere e tutte le richiesta intere
Tesi: Esiste una soluzione ottima intera.

$$\left.\begin{array}{l} d_i \ intero, \quad \forall i \\ r_j \ intero, \quad \forall j \end{array}\right\} \Rightarrow \exists \text{ soluzione ottima intera}$$

$\Rightarrow$ La regione ammissibile è tale per cui tutti i vertici stanno su punti interi.

Osservazione: Se il problema è equilibrato posso scrivere i vincoli con l'uguaglianza (=)

1.3 Matrice dei trasporti (con esempio)

Sia dato il seguente problema:

- Le disponibilità i delle fabbriche sono le seguenti: $O_1 = 40,\ O_2 = 60,\ O_3 = 90,\ O_4 = 50$
- Le richieste j dei magazzini sono le seguenti: $D_1 = 30,\ D_2 = 40,\ D_3 = 70,\ D_4 = 40,\ D_5 = 60$

Prima del riempimento la matrice dei trasporti sarà:

	D1	D2	D3	D4	D5	DISP. i
O1						40
O2						60
O3						90
O4						50
RICH. j	30	40	70	40	60	EQ

Vado a riempire la matrice dei trasporti seguendo il metodo "**Angolo di N-W**" e ottengo:

	D1	D2	D3	D4	D5	DISP. i
O1	30	10				~~40~~ ~~10~~
O2		30	30			~~60~~ ~~30~~
O3			40	40	10	~~90~~ ~~50~~ ~~10~~
O4					50	~~50~~
RICH. j	~~30~~	~~40~~ ~~30~~	~~70~~ ~~40~~	~~40~~ 0	~~60~~ ~~50~~	EQ

Se volessi trasportare 30 in O_4/D_1 e farlo entrare in base cosa dovrei fare?

	D1	D2	D3	D4	D5	DISP. i
O1	30 $-$	10 $+$				~~40~~ ~~10~~
O2		30 $-$	30 $+$			~~60~~ ~~30~~
O3			40 $-$	40	10 $+$	~~90~~ ~~50~~ ~~10~~
O4	? $+$				50 $-$	~~50~~
RICH. j	~~30~~	~~40~~ ~~30~~	~~70~~ ~~40~~	~~40~~ 0	~~60~~ ~~50~~	EQ

1. **Perché ho trasportato 30?** Ho trasportato 30 perché il $min\{30, 30, 40, 50\}$ che sono i valori che poi andrò a sottrarre.

2. **Perché questo percorso?** è l'unico percorso disponibile, per trovarlo devo:

   - Mettere una "x" sulle variabili in base ■
   - Mettere una "x" sulla variabile fuori base che voglio far entrare in base ■
   - Cancellare le righe / colonne con una sola "x"
   - Vedere qual è il percorso (muovendomi in verticale e orizzontale) che mi permette di chiudere il ciclo. ■

	D1	D2	D3	~~D4~~	D5
O1	×	×			
O2		×	×		
O3			×	~~×~~	×
O4	×				×

3. **Conviene o non conviene?** Per dire se conviene o non conviene fare questo spostamento bisogna introdurre il concetto di "**Costo Marginale Unitario**[C_M]" che equivale alla somma/sottrazione dei costi unitari lungo il percorso scelto. Nel nostro caso

$$C_M = c_{41} - c_{11} + c_{12} - c_{22} + c_{23} - c_{33} + c_{35} - c_{45}$$

   - Se $C_M < 0$ conviene
   - Se $C_M > 0$ NON conviene
   - Se $C_M = 0$ è indifferente

1.4 Esempio pratico

	D1	D2	D3	D4	D5
O1	4	3	5	5	0
O2	2	3	2	3	0
O3	4	3	2	5	0

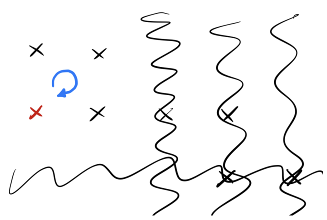
Tabella Costi

	D1	D2	D3	D4	D5	DISP. i
O1	11	11				22
O2		4	12	2		18
O3				10	10	20
RICH. j	11	15	12	12	10	EQ

Tabella Domanda e Richiesta (riempita con metodo "Angolo di N-W")

Costo $= 11 \cdot 4 + 11 \cdot 3 + 4 \cdot 3 + 12 \cdot 2 + 2 \cdot 3 + 10 \cdot 5 + 10 \cdot 0 = 169$
Possiamo spendere di meno?

- **O2/D1** $\rightarrow$ Quanto trasporto: $4 = min\{11, 4\}$
  $C_M = 2 - 4 + 3 - 4 = -2 \Rightarrow CONVIENE$ ■
  $Nuovo\ costo = costo - C_M \cdot quantita = 161$



- **O3/D2** $\rightarrow$ $C_M = 3 - 3 + 4 - 2 + 3 - 5 = 0 \Rightarrow Indifferente$
- **O1/D3** $\rightarrow$ $C_M = 5 - 2 + 2 - 4 = 1 \Rightarrow NON\ CONVIENE$
- **O1/D4** $\rightarrow$ $C_M = 0 \Rightarrow Indifferente$
- **O1/D5** $\rightarrow$ $C_M = 0 \Rightarrow Indifferente$
- **O2/D5** $\rightarrow$ $C_M = 2 \Rightarrow NON\ CONVIENE$
- **O2/D2** $\rightarrow$ $C_M = 2 \Rightarrow NON\ CONVIENE$
- **O3/D1** $\rightarrow$ $C_M = 0 \Rightarrow Indifferente$
- **O3/D3** $\rightarrow$ Quanto trasporto: 10
  $C_M = -2 \Rightarrow CONVIENE$
  $Nuovo\ costo = 161 - 10 \cdot 2 = 141$
- **O1/D5** $\rightarrow$ Quanto trasporto: 2
  $C_M = -2 \Rightarrow CONVIENE$
  $Nuovo\ costo = 141 - 2 \cdot 2 = 137$

Arrivo alla soluzione ottima:

	D1	D2	D3	D4	D5	DISP. i
O1	5	15			2	22
O2	6	0	0	12		18
O3			12	0	8	20
RICH. j	11	15	12	12	10	EQ

1.5 Tecnica del "Big M"

$$C_{N+1,j} = M >> \dots$$

dove:

- $C_{N+1,j}$ = Costo unitario elevato
- $\dots$ = Altri parametri in gioco

In questo caso il costo finale del trasporto = $costo' - D_{N+1} \cdot M$ $\rightarrow$ cosi posso studiare un problema inizialmente non ammissibile.

D_{N+1} mi rappresenta quella che è una disponibilità fittizia.

1.6 Problema Degenere

Supponiamo di trovare un problema in cui:

D / R

100	200
?	100

Costi

3	4
1	5

$C_M = 1 - 3 + 4 - 5 = -3 < 0$
$min\{100, 100\} = 100$

Iterando l'algoritmo di Stepping Stone ottengo:

	200
100	eps

eps è in base ma con valore nullo → è utilizzabile in successive iterazioni di stepping stone ⇒ tale soluzione si dice degenere.

1.7 Più soluzioni ottime

Una soluzione è ottima quando tutti i costi marginali alternativi provati con altre interazioni di stepping stone sono > 0 (non ce ne sono altri < 0)
⇒ se ho due soluzioni ottime → ce ne sono infinite (non è detto che siano tutte intere)

1.8 Metodo del minimo costo

D / R

/	/	100	100
	200	100	300
300	/	/	300
300	200	200	

Costi

5	4	3
8	4	3
9	7	5

Noto che se applicassi il metodo "Angolo di N-W" il costo sarebbe pari a $4'200$, invece partendo dalle caselle con costo più basso il costo si abbassa a $4'100$

1.9 Metodo di Vogel [VAM]

Costi di opportunità: differenza tra i due costi più bassi.

Algoritmo:

1. Per ogni riga/colonna determino il costo di opportunità
2. Scelgo la riga/colonna con il costo di opportunità maggiore
3. Per la riga/colonna scelta trasporto il più possibile
4. Elimino righe/colonne "sature"
5. Itero ⇒ ricalcolo i costi di opportunità
6. Vai al punto 2

Applicando il "Metodo di Vogel" ottengo la tabella seguente

D / R

100	/	/	100
/	200	100	300
200	/	100	300
300	200	200	

			1		2	3^ iteraz
			1	3	2	2^ iteraz
			3	0	0	1^ iteraz
		1	5	4	3	
5	1	1	8	4	3	
4	2	2	9	7	5	
3^ iteraz	2^ iteraz	1^ iteraz				

con un costo pari a $3'900$

1.10 Metodo Modified Distribution [MODI]

É un metodo che modifica l'algoritmo di stepping stone sfruttando la teoria della dualità. Introduco R_i, K_j

Algoritmo:

1. Per ogni cella occupata scrivo: $R_i + K_j = C_{ij}$
2. Scelgo $R_1 = 0$
3. Per sostituzione ricavo tutte le altre R_i, K_i (a cascata)
4. Definisco I_{ij} per ogni cella vuota, dove $I_{ij} = c_{ij} - R_i - K_j$
5. Seleziono la cella con $I_{ij} < 0$ e più piccolo di tutti gli altri
6. Itero finché trovo tutti gli $I_{ij} > 0$ (soluzione ottima)

1.10.1 Esempio

Parto con N-W $\rightarrow$ $costo = 4'200$

D / R

100			100
200	100		300
	100	200	300
300	200	200	

Costi

5	4	3
8	4	3
9	7	5

Per applicare MODI vado a calcolarmi gli R_i, i K_j e i rispettivi I_{ij}:

1. $$\begin{aligned} R_1 + K_1 &= 5 \\ R_2 + K_1 &= 8 \\ R_2 + K_2 &= 4 \\ R_3 + K_2 &= 7 \\ R_3 + K_3 &= 5 \end{aligned}$$

2-3. Pongo $R_1 = 0 \Rightarrow \begin{aligned} K_1 &= 5 \\ R_2 &= 3 \\ K_2 &= 1 \\ R_3 &= 6 \\ K_3 &= -1 \end{aligned}$

4. Mi ricavo gli $I_{ij} \Rightarrow \begin{aligned} I_{12} &= c_{12} - K_1 - K_2 = 3 \\ I_{13} &= 3 - 0 + 1 = 4 \\ I_{23} &= 3 - 3 + 1 = 1 \\ I_{31} &= 9 - 6 - 5 = -2 \end{aligned}$

5. Scelgo l'I_{ij} con valore minore quindi x_{31} va in base

6. Applico Stepping Stone classico ($min = 100$) (MODI serve per scegliere da dove partire) $\rightarrow costo = 4'000$

D / R

100		
100	200	
100		200

Costi

5	4	3
8	4	3
9	7	5

1. $$\begin{aligned} R_1 + K_1 &= 5 \\ R_2 + K_1 &= 8 \\ R_2 + K_2 &= 4 \\ R_3 + K_1 &= 9 \\ R_3 + K_3 &= 5 \end{aligned}$$

2-3. Pongo $R_1 = 0 \Rightarrow \begin{array}{l} K_1 = 5 \\ R_2 = 3 \\ K_2 = 1 \\ R_3 = 4 \\ K_3 = 1 \end{array}$

4. Mi ricavo gli $I_{ij} \Rightarrow \begin{array}{l} I_{12} = 4 - 0 - 1 = 3 \\ I_{13} = 3 - 0 - 1 = 2 \\ I_{23} = 3 - 3 - 1 = -1 \\ I_{32} = 7 - 4 - 1 = 2 = -2 \end{array}$

5. Scelgo l'I_{ij} con valore minore quindi x_{23} va in base

6. Applico Stepping Stone classico $\rightarrow costo = 3'900$

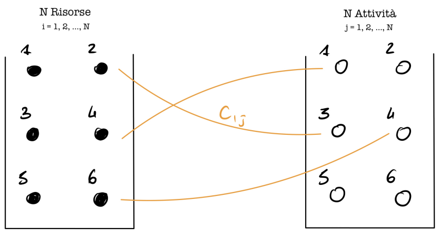
D / R

100		
	200	100
200		100

Costi

5	4	3
8	4	3
9	7	5

2 Problema dell'assegnazione



2.1 Formulazione Matematica

Problema di programmazione lineare intera [PLI].

Variabili decisionali

$$x_{ij} = \begin{cases} 0 & \text{risorsa } i \text{ non assegnata a attività } j \\ 1 & \text{risorsa } i \text{ assegnata a attività } j \end{cases} .$$

Funzione Obiettivo

$$min\ z = \sum_{i}^{n} \sum_{j}^{m} x_{ij} \cdot c_{ij}$$

Vincoli

- $\sum_j x_{ij} = 1, \quad \forall i = 1, \dots, n$
- $\sum_i x_{ij} = 1, \quad \forall j = 1, \dots, n$
- $x_{ij} \in \{0; 1\} \quad \forall i, j \rightsquigarrow x_{ij} \geq 0 \quad RILASSATI$

2.2 Similitudini col problema di trasporto

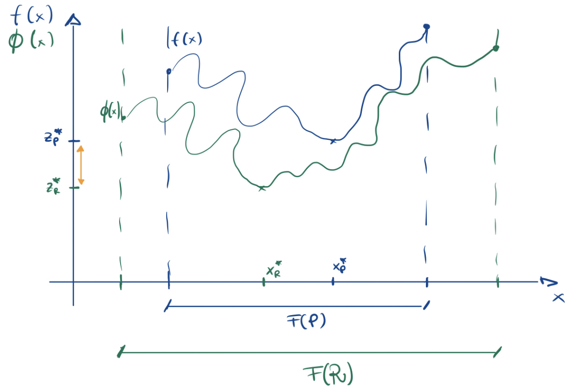
$$\left.\begin{array}{l} N = M \\ d_i = 1 \quad \forall i \\ r_i = 1 \quad \forall j \end{array}\right\} \Rightarrow Assegnazione?$$

2.3 Parentesi: Rilassamenti di problemi di ottimizzazione

Siano P e R due problemi:
P: $min\ f(x), \quad x \in F(P)$ dove con F intendo la regione ammissibile (feasable)
R: $min\ \phi(x), \quad x \in F(R)$

R è detto rilassamento di P se e solo se:

1. $F(P) \subseteq F(R)$
2. $\phi(x) \leq (\geq$ *se di massimo*$)\ f(x), \quad \forall x \in F(P)$



2.3.1 Quando risolvere un problema rilassato

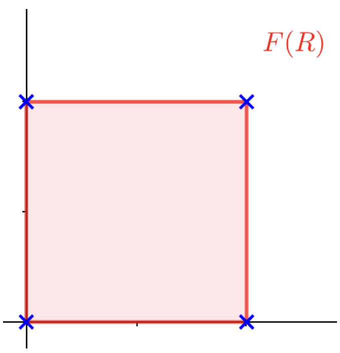
1. R più semplice di P
2. I valori ottimi z_P^* e z_R^* sono prossimi (il più simili possibile)
3. Almeno una soluzione ottima di R dev'essere ammissibile per P
4. Se almeno una soluzione ottima di R è ottima anche per P

2.3.2 Rilassamento per eliminazione

Eliminare dei vincoli "alleggerisce" il problema. Si andranno a verificare tali vincoli a posteriori

2.3.3 Rilassamento lineare (continuo)

Variabili decisionali intere $\rightsquigarrow$ Variabili decisionali continue

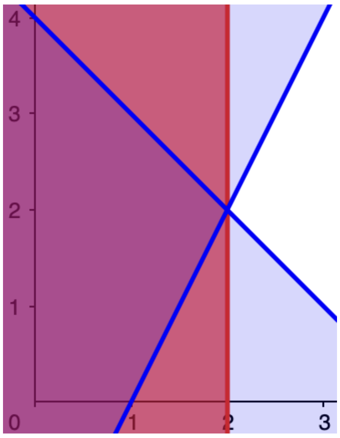


Esempio: $x_{ij} \in \{0; 1\} \rightsquigarrow 0 \leq x_{ij} \leq 1, \quad \forall i, j$

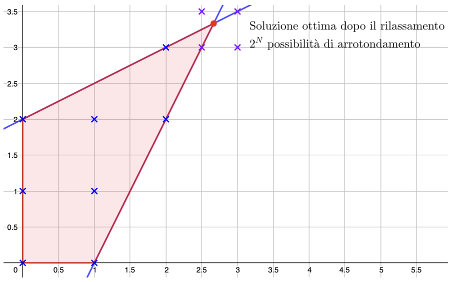
2.3.4 Rilassamento surrogato

$$max(x = 2x_1 + 3x_2)$$

$$\begin{cases} x_1 + x_2 \leq 4 \\ 2x_1 - x_2 \leq 2 \\ x_1, x_2 \geq 0 \end{cases} \Rightarrow \begin{cases} 3x_1 \leq 6 \\ x_1, x_2 \geq 0 \end{cases}$$



2.4 Altra parentesi



2.5 Algoritmo ungherese

Ogni problema di assegnazione ha $N!$ soluzioni ammissibili. Sia dato il problema con tabella dei costi c_{ij} seguente:

	A1	A2	A3	A4	A5	A6
R1	23	13	3	9	5	32
R2	7	25	10	12	11	13
R3	26	20	13	33	24	12
R4	2	17	5	8	4	20
R5	18	5	32	2	4	1
R6	2	1	23	3	19	14

1. Tolgo il valore minimo ad ogni riga
2. Cerco una soluzione a costo 0
3. Tolgo il valore minimo ad ogni colonna
4. Cerco una soluzione a costo 0

20	10	0	5	0	29
0	18	3	4	2	6
14	8	1	20	10	0
0	15	3	5	0	18
17	4	31	0	1	0
1	0	22	1	16	13

Caso particolare per procedere con l'algoritmo:
Soluzione non ottima, $costo = 86$

32	18	12	22	41	13
16	21	19	34	22	20
35	42	22	29	21	33
11	17	14	26	29	12
28	25	40	13	9	10
8	12	32	10	23	26

→

20	6	0	10	29	1
0	5	3	18	6	4
14	21	1	8	0	12
0	6	3	15	18	1
17	16	31	4	0	1
0	4	24	2	15	18

→

20	2	0	8	29	0
0	1	3	16	6	3
14	17	1	6	0	11
0	2	3	13	18	0
17	12	31	2	0	0
0	0	24	0	15	17

5. Ricoprimento degli zeri:
   (a) Segno le righe senza assegnazione
   (b) Segno che colonne che hanno "zero" nelle righe segnate
   (c) Segno le righe che hanno assegnazioni nelle colonne segnate
   (d) Se al punto 5.c ho aggiunto righe, vai al punto 5.b
   (e) Le righe coprenti sono le righe non segnate. Le colonne coprenti solo le colonne non segnate: Cinque ⬅/⬇ tante quante le assegnazioni obbligate.
      - Trovo un riquadro di "elementi scoperti"
      - Tolgo il valore minimo degli elementi scoperti e lo sottraggo a tutti gli elementi scoperti
      - Sommo tale valore delle righe/colonne dove c'è sovrapposizione

⬇				⬇	⬇	
20	2	0	8	29	0	⬅
0	1	3	16	6	3	✅
14	17	1	6	0	11	✅
0	2	3	13	18	0	✅
17	12	31	2	0	0	✅
0	0	24	0	15	17	⬅
✅				✅	✅	

Ottenendo così un $costo = 85$, che nel nostro caso è una soluzione ottima.

21	2	0	8	30	1
0	0	2	15	6	3
14	16	0	5	0	11
0	1	2	12	18	0
17	11	30	1	0	0
1	0	24	0	16	18

E qualora avessi un problema di massimizzazione? [Non è un problema di assegnazione!]
Vado a considerare p_{ij} profitti

$$max\ z = \sum_i \sum_j p_{ij} \cdot x_{ij} \qquad 0 \leq p_{ij} \leq \hat{p}_{\bar{i}\bar{j}} = max_{ij}\{p_{ij}\}$$

$\rightarrow$ costruisco un problema di assegnazione:
Pongo $c_{ij} := \hat{p}_{\bar{i}\bar{j}} - p_{ij}, \forall i, j$ (complemento dei profitti rispetto al profitto massimo)
$z' = \sum_i \sum_j (\hat{p}_{\bar{i}\bar{j}} - p_{ij}) x_{ij} = \sum_i \sum_j \hat{p}_{\bar{i}\bar{j}} \cdot x_{ij} - \sum_i \sum_j p_{ij} \cdot x_{ij} = min\ \alpha = max\ (-\alpha) \rightarrow min\ z' = max\ z$

Esempio:

9	15	14	2	8	11	12
18	16	13	18	11	8	14
14	12	17	8	8	0	1
6	9	9	12	12	9	6
16	18	8	6	7	14	10
4	7	9	13	11	7	8
12	13	7	13	6	9	14

$\rightarrow$

6	9	10	0	5	9	9
9	4	3	10	2	0	5
13	8	15	8	7	0	0
0!	0	2	7	6	4	0
9	8	0!	0	0	8	3
0	0	4	10	7	4	4
6	4	0	8	0	4	8

Dove ho messo dei **!** vuol dire che ho fatto delle scelte. Le soluzioni ottime possibili sono 2^k, dove $k = \#$ scelte (non obbligate) fatte

Più nello specifico:

- Stallo dopo il ■, non ho più scelte obbligate
- Inizio a operare in maniera arbitraria ■
- Continuando vedo se sono obbligato in delle scelte ■

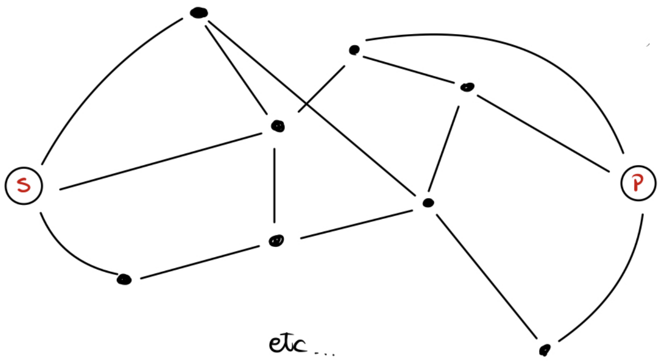
6	9	10	0	5	9	9
9	4	3	10	2	0	5
13	8	15	8	7	0	0
0	0	2	7	6	4	0
9	8	0	0	0	8	3
0	0	4	10	7	4	4
6	4	0	8	0	4	8

Caso ancora più patologico
Le soluzioni ottime sono tante quante le soluzioni ammissibili $\Rightarrow$ 4!

■ $\rightarrow$ *arbitraria*
■ $\rightarrow$ *obbligata*

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

3 Problema del flusso massimo [MAX FLOW]



3.1 Formulazione Matematica

ll problema del flusso massimo viene descritto dalla funzione $G = (\aleph,\ A)$ dove

$$\begin{array}{ll} \aleph & \text{insieme dei nodi} \\ A & \text{insieme degli archi (orientati)} \end{array}$$

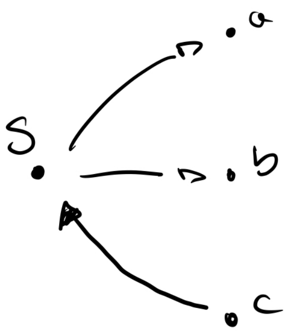
Variabili decisionali :

$$x_{ij} = \text{ quantità del flusso lungo l'arco } (i,j) \in A$$

Funzione obiettivo :

$$max\ z = \sum_{(S,j)\in\delta^+(S)} x_{S,j} - \sum (i,S) \in \delta^-(S) x_{i,S}$$

Nota Bene: $\delta^+(S)$ e $\delta^-(S)$ mi rappresentano gli insiemi di archi uscenti ed entranti nel punto S. Nell'esempio seguente: $\begin{array}{l} \delta^+(S) = \{(S,a);(S,b)\} \\ \delta^-(S) = \{(c,S)\} \end{array}$



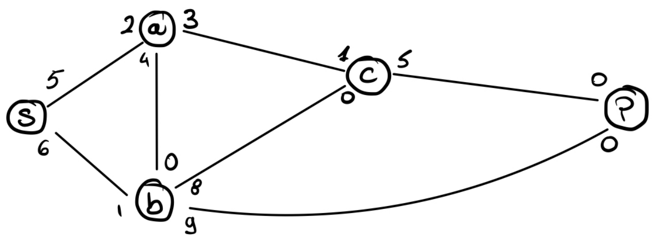
Vincoli

$$\sum_{(i,j)\in\delta^+(i)} x_{ij} - \sum_{(i,j)\in\delta^-(i)} = 0, \quad \forall i, \in \aleph \setminus \{S; P\}$$

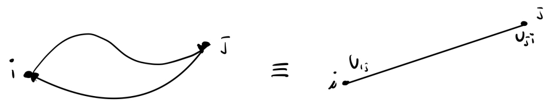
$$0 \le x_{ij} \le u_{ij}, \quad \forall (i,j) \in A$$

Condizione sui dati: $u_{ij} \ge 0, \ \forall (i,j) \in A$

3.2 Vediamo un istanza



Nota Notazionale:



$\aleph = \{S,\ a,\ b,\ c,\ P\}$; $\quad A = \{(S,a),(S,b),(a,S),(a,b),(b,S),(a,c),(b,a),(c,a),(b,c),(b,P),$ $(c,b),(P,b),(c,P),(P,c)\}$
Vincoli:

a) $x_{Sa} + x_{ba} + x_{ca} - x_{aS} - x_{ab} - x_{ac}$

b) $x_{Sb} + x_{ab} + x_{cb} + x_{Pb} - x_{bS} - x_{ba} - x_{bc} - x_{bP}$

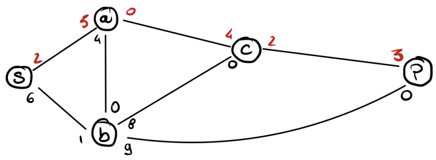
c) $x_{ac} + x_{bc} + x_{Pc} - x_{ca} - x_{cb} - x_{cP}$

$0 \le x_{Sa} \le 5$	$0 \le x_{as} \le 2$
$0 \le x_{Sb} \le 6$	$0 \le x_{bs} \le 1$
$0 \le x_{ab} \le 4$	$0 \le x_{ba} \le 0$
$0 \le x_{ac} \le 3$	$0 \le x_{ca} \le 1$
$0 \le x_{bc} \le 8$	$0 \le x_{cb} \le 0$
$0 \le x_{bP} \le 5$	$0 \le x_{Pb} \le 0$
$0 \le x_{cP} \le 9$	$0 \le x_{Pc} \le 0$

3.3 Algoritmo di Ford - Fulkerson

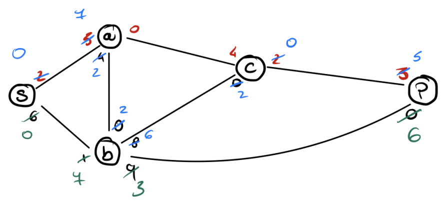
1. Individuo un cammino: $S - a - c - P : 3$ (trasporto tre perché è il collo di bottiglia del percorso)

2. Aggiorno le capacità in entrambi i sensi ■

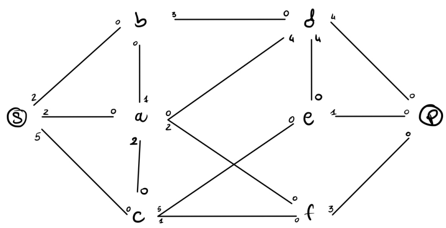


3. Trovo un cammino aumentante
4. Vai al punto 2

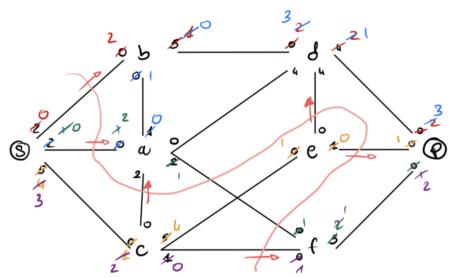
Dopo un paio di passaggi, nella nostra istanza, arrivo a un flusso risultante di 11 trovando due percorsi aumentanti $S-a-b-c-P:2$ ■ e $S-b-P:6$ ■



3.4 Istanza più complicata



Cammino: $S-b-d-P:2$ ■; $S-a-b-d-P:1$ ■; $S-a-f-P:1$ ■; $S-c-e-P:1$ ■; $S-c-f-P:2$ ■



La riga ■ mi rappresenta il taglio S-P dell'istanza e $\rightarrow$ gli archi tagliati.

Definizione [Taglio S-P in una rete]:

$$\aleph = \aleph_1 \cup \aleph_2 \quad e \quad \aleph_1 \cap \aleph_2 = \emptyset$$
$$S \in \aleph_1 \quad e \quad P \in \aleph_2$$

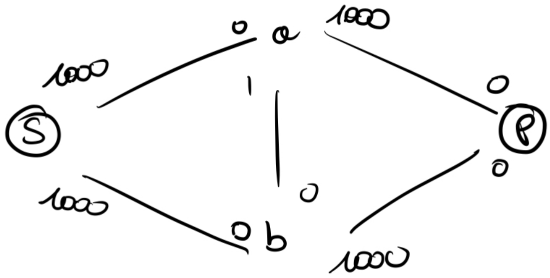
Definizione [Flusso attraverso il taglio S-P]:

$$\sum_{(i,j)\in\delta^+(\aleph_1)} x_{ij} - \sum_{(i,j)\in\delta^-(\aleph_1)} x_{ji}$$

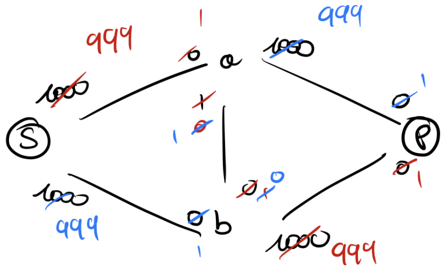
Definizione [Capacità del taglio S-P]:

$$\sum_{(i,j)\in\delta^+(\aleph_1)} u_{ij}$$

3.5 Istanza patologica



Poterebbe capitare un istanza dove prendendo dei specifici cammini $S-a-b-P$: 1 ■; $S-b-a-P:1$ ■ ci si impiegherebbero un sacco di passaggi inutili.



Per questo motivo nell'algoritmo può essere introdotta la **regola di Edmonds & Karp** ossia quella di preferire i cammini con meno archi coinvolti

3.6 Problemino

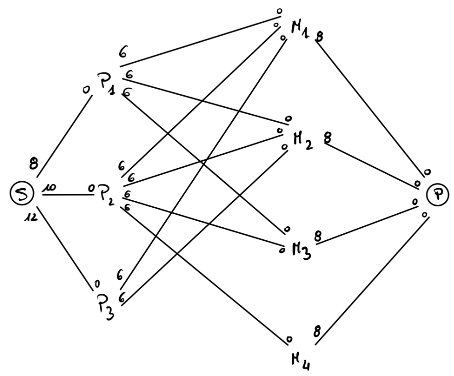
Specifiche:
P_1 deve essere completati entro i prossimi 3 *mesi*
P_2 deve essere completati entro i prossimi 4 *mesi*
P_3 deve essere completati entro i prossimi 2 *mesi*

P_1 impiega una forza di lavoro pari a 8 $mesi \times uomo$
P_2 impiega una forza di lavoro pari a 10 $mesi \times uomo$

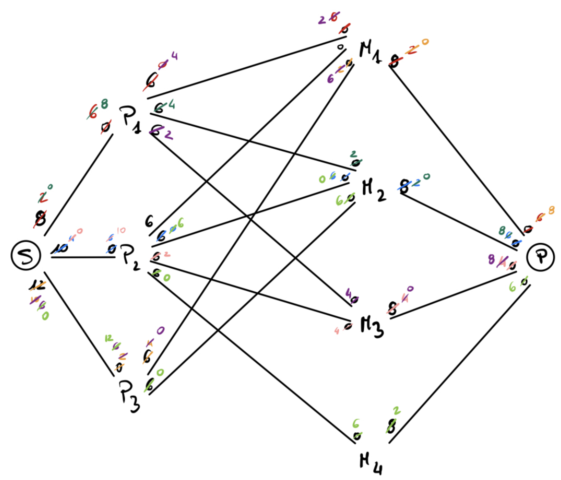
P_3 impiega una forza di lavoro pari a 12 $mesi \times uomo$

Non si possono utilizzare più di 8 *risorse* in un mese e possono lavorare allo stesso progetto un massimo di 6 *persone* in un certo mese.

Risoluzione: Imposto il problema come un problema di flusso massimo



Inizio coi cammini: $S - P_1 - M_1 - P : 6$ ■; $S - P_2 - M_2 - P : 6$ ■; $S - P_3 - M_1 - P : 2$ ■; $S - P_1 - M_2 - P : 2$ ■; $S - P_2 - M_3 - P : 4$ ■; $S - P_3 - M_1 - P_1 - M_3 - P : 4$ ■; $S - P_3 - M_2 - P_2 - M_4 - P : 6$ ■

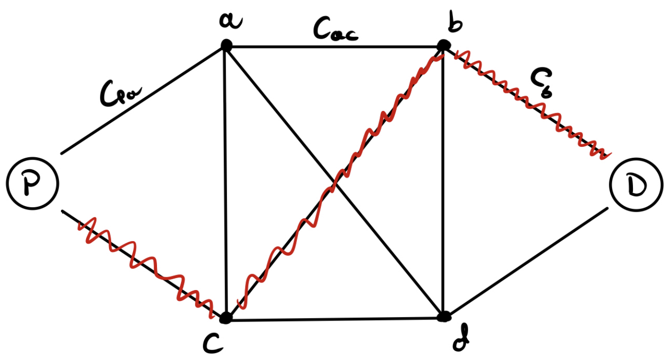


$\aleph_1 = \{S\}, \quad \aleph_2 = \aleph \setminus \aleph_1$
Soluzione:

	M1	M2	M3	M4	
P1	2	2	4	/	**8**
P2	/	/	4	6	**10**
P3	6	6	/	/	**12**
	8	**8**	**8**	**6**	

4 Problema del cammino minimo [SPP]

SPP: Shortest Path Problem



■: Soluzione indicativa

4.1 Formulazione Matematica

ll problema del cammino minimo è un problema su reti e viene descritto dalla funzione $G = (\aleph,\ A)$ dove

$$\begin{array}{ll} \aleph & \text{insieme dei nodi} \\ A & \text{insieme degli archi (orientati)} \end{array}$$

c_{ij} mi rappresentano quelli che sono i costi di percorrenza degli archi.

Variabili decisionali

$$x_{ij} = \begin{cases} 0 & \text{se } (i,j) \notin \text{percorso a costo minimo} \\ 1 & \text{se } (i,j) \in \text{percorso a costo minimo} \end{cases}$$

Funzione Obiettivo

$$min\ z = \sum_{i}^{n} \sum_{j}^{m} x_{ij} \cdot c_{ij}$$

Vincoli

P)

$$\sum_{(P,j) \in \delta^+(P)} x_{Pj} = 1$$

D)

$$\sum_{(i,D) \in \delta^-(D)} x_{iD} = 1$$

Altri vincoli funzionali:

$$\sum_{(h,j)\in\delta^+(h)} x_{hj} - \sum_{(i,h)\in\delta^-(h)} x_{ih} = 0 \quad \forall h \in \aleph \setminus \{P;\ D\}$$

Vincoli di dominio: $x_{ij} \in \{0; 1\}; \quad \forall(i,j) \in A$
Condizione sui parametri: $c_{ij} \geq 0, \quad \forall(i,j) \in A$

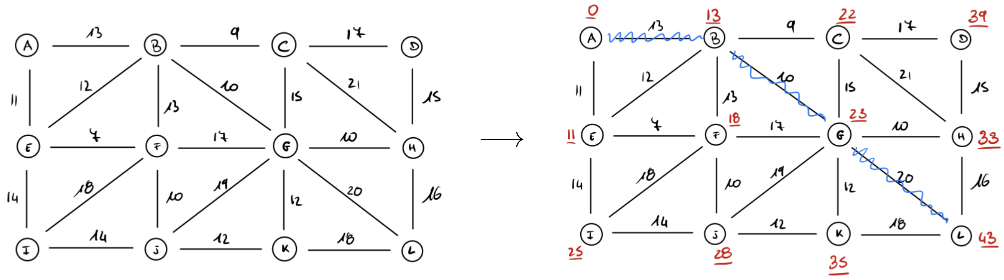
4.2 Algoritmo risolutivo [Dijkstra]

1. **Inizializzazione**
   - $d[P] = 0$ distanza nodo iniziale $= 0$
   - $d[i] = \infty$ distanza iniziale di ogni nodo diverso da P
   - $pred[i] = i$ ogni nodo ha predecessore se stesso
   - $L = \{P\}$ la lista dei nodi da analizzare contiene il nodo P
   - $S = \{\}$ la lista dei nodi già analizzati/etichettati
2. **Iterazione**
   Finché la lista dei nodi da visitare L non è vuota:
   - Estrarre dalla lista L il nodo x con distanza più bassa
   - Eliminare il nodo x dalla lista L (nodi da analizzare)
   - Aggiungere il nodo x alla lista S (nodi analizzati)
   - Analizzare ogni nodo j adiacente di x se j non è in S (nodi analizzati)
     - Calcolare la distanza dei nodi adiacenti del nodo x $d'[j] = d[x] + c(x, j)$
     - Se ll costo $d'[j] < d[j]$
       * Aggiornare il vettore $d[j] = d'[j]$
       * Aggiornare il predecessore $pred[j] = x$
       * Aggiungere alla lista L il nodo adiacente j (se non presente)

   Al termine dell'esecuzione l'algoritmo indica il costo minimo $d[i]$ del percorso per raggiungere l'i-esimo nodo a partire dal nodo iniziale P.
   Se un valore $d[i]$ è infinito, vuol dire che l'i-esimo nodo non è raggiungibile da P.

`https://it.wikipedia.org/wiki/Algoritmo_di_Dijkstra`

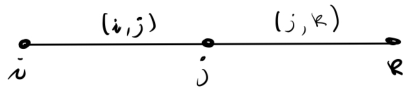
4.3 Esempio



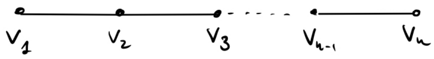
5 Alberi Minimi

5.1 Richiamo & Definizioni

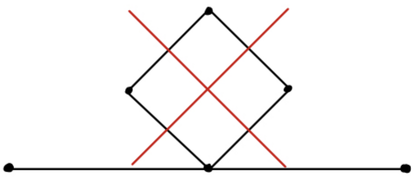
Archi adiacenti: Archi che condividono un nodo



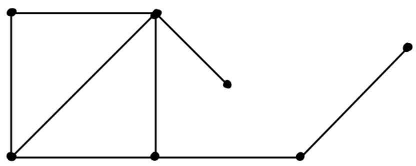
Cammino / Percorso: Sequenza di archi a due a due adiacenti



Cammino semplice: Cammino che non contiene cicli

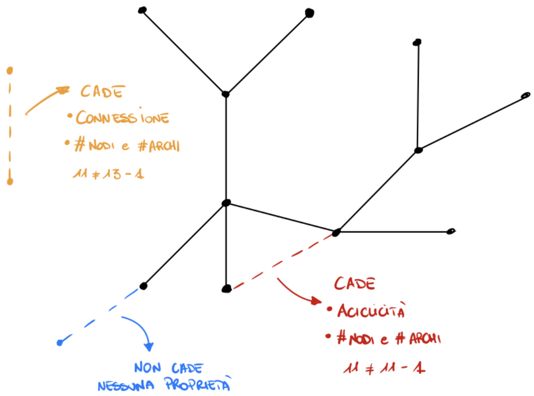


Ciclo / Circuito: Cammino con $v_1 = v_n$
Grafo Connesso: Esiste (almeno) un cammino per ogni coppia possibile di nodi

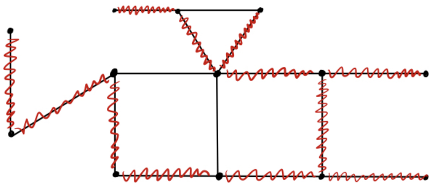


Albero: Grafo con 3 proprietà:

- É connesso
- É aciclico
- Ha un numero di nodi = numero di archi + 1 $\Rightarrow$ $\#\ Archi = \#\ Nodi - 1$

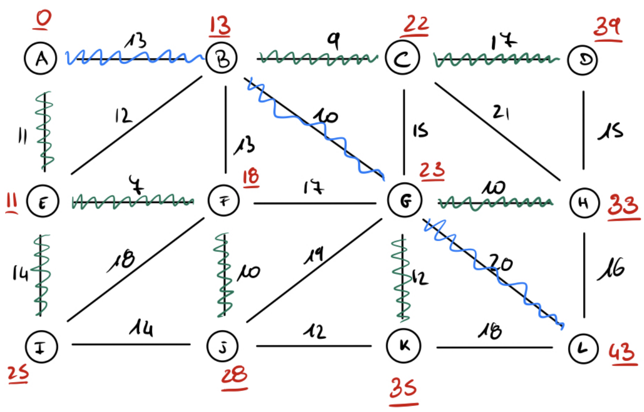


Albero di supporto / ricoprente / di copertura: É un sottografo ad albero che tocca tutti i nodi del grafo che lo contiene



5.2 Problema dell'albero dei percorsi minimi

Dopo aver risolto il problema del percorso minimo, vado a ritroso e vedo quello che è l'albero dei percorsi minimi sottraendo le varie etichette date ai nodi con quelli che sono i percorsi.



Nota Bene: se ho diverse possibilità vado a sceglierne una in maniera tale da non generare cicli e mantenere le 3 proprietà di un albero.

5.3 Problema del minimo albero ricoprente [MST]

MST: Minimum spending tree

5.3.1 Formulazione Matematica

ll problema del minimo albero ricoprente è un problema su reti e viene descritto dalla funzione $G = (\aleph,\ A)$ dove

$$\begin{array}{ll} \aleph & \text{insieme dei nodi} \\ A & \text{insieme degli archi (non orientati)} \end{array}$$

c_{ij} mi rappresentano quelli che sono i costi di percorrenza degli archi.

Variabili decisionali

$$x_{ij} = \begin{cases} 0 & \text{se } (i,j) \notin \text{albero} \\ 1 & \text{se } (i,j) \in \text{albero} \end{cases}$$

Funzione Obiettivo

$$min\ z = \sum_{(i,j)\in A} x_{ij} \cdot c_{ij}$$

Vincoli

$$\sum_{(i,j)\in A} x_{ij} = |N| - 1$$

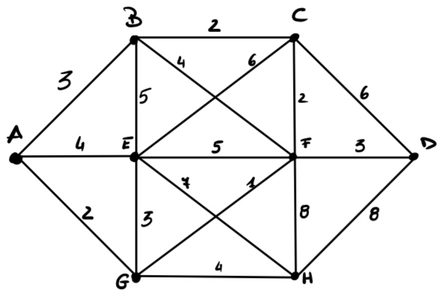
$$\sum_{(i,j)\in S} x_{ij} \leq |S|, \quad \forall S \subseteq \aleph$$

Vincoli di dominio: $x_{ij} \in \{0; 1\}; \quad \forall (i,j) \in A$

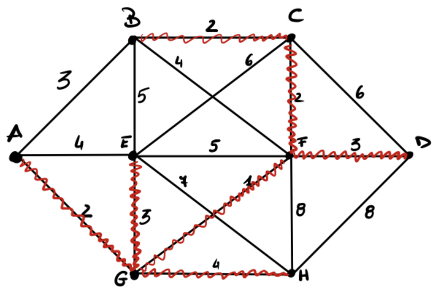
5.3.2 Algoritmo di Kruskal

L'algoritmo di Kruskal è un algoritmo miope (Greedy)

1. Ordino in c_{ij} non decrescenti
2. Seleziono l'arco più economico (non ancora selezionato è purché non generi un ciclo)
3. Continuo col passo 2 fino ad ottenere un albero coprente



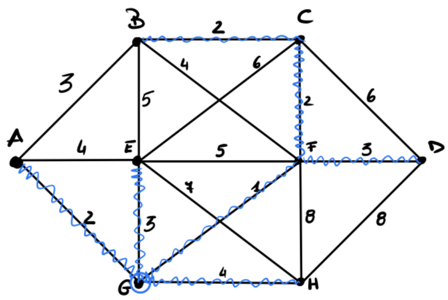
1	2	3	4	5	6	7	8
~~FG~~	~~BC~~	AB^{no}	BF^{no}	BE^{no}	CE^{no}	EH^{no}	DH^{no}
	~~AG~~	~~EG~~	AE^{no}	EF^{no}	CD^{no}		FH^{no}
	~~CF~~	~~DF~~	~~GH~~				



5.3.3 Algoritmo di Prim

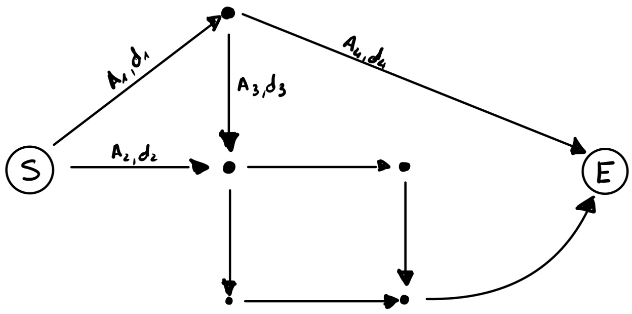
L'algoritmo di Prim è un algoritmo miope (Greedy)

1. Scegli un nodo casuale
2. Connetti un nodo (adiacente) sconnesso che sia a costo minimo rispetto all'albero che stai creando.



6 Problema del cammino critico [CPM]

CPM: Critical Path Method
Questo problema rientra nella categoria del project management, gli archi rappresentano le attività di un progetto dove P_j rappresenta l'insieme di attività (task) da svolgere. Le attività sono legate a vincoli di precedenza [$\prec$], ed avranno delle durate.



Nota Bene: Per costruzione non posso avere loop

6.1 Formulazione Matematica

Variabili decisionali

$$x_{ij} = \begin{cases} 0 & \text{se } (i,j) \text{ non è critica} \\ 1 & \text{se } (i,j) \text{ è critica} \end{cases}$$

Funzione Obiettivo

$$max\ z = \sum_{(i,j) \in A} x_{ij} \cdot d_{ij}$$

Vincoli

$$\sum_{(S,j) \in \delta^+(S)} x_{Sj} = 1$$

$$\sum_{(i,E) \in \delta^-(E)} x_{iE} = 1$$

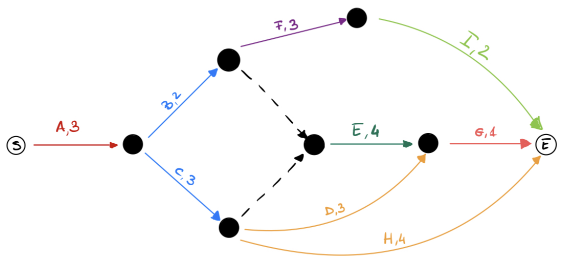
$$\sum_{(i,h) \in \delta^+(i)} x_{ih} - \sum_{(h,i) \in \delta^-(i)} x_{hi} = 0, \quad \forall i \in \aleph \setminus \{S; E\}$$

6.2 Costruzione di un grafo

Sia data la seguente tabella:

Attività	Durata (GG)	Precedenze	Risorse
A	3	/	5
B	2	A	3
C	3	A	4
D	3	C	2
E	4	B, C	1
F	3	B	3
G	1	E, D	5
H	4	C	4
I	2	F	1

1. Inizio segnando la attività senza precedenze
2. segno TUTTE le attività che hanno come unica precedenza l'attività appena tracciata
3. Se un'attività ha due precedenze (come nel caso di E) posso andare a tracciare delle attività fittizie a costo 0 (■) per rappresentarla graficamente
4. Itero il procedimento per tutte le attività
5. Quando ho aggiunto tutte le attività, controllo andando a ritroso le propedeuticità e che non si siano generati loop



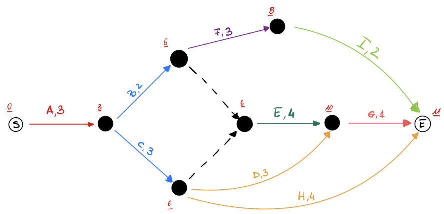
Nel grafo ho rappresentato le propedeuticità coi seguenti colori:
Nessuna propedeuticità: ■ / A: ■ / B: ■ / C: ■ / B & C: ■ / E & D: ■ / F: ■

6.3 Algoritmo Risolutivo

Molto simile all'algoritmo di Dijkstra

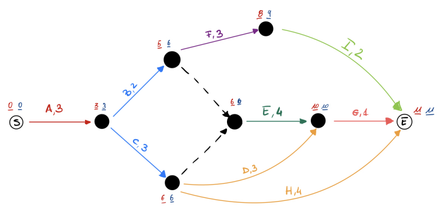
I Fase di etichettatura dei nodi

1. Il progetto inizia a potenziale 0
2. Trovo i potenziali delle attività vicine
3. Diventa definitivo il potenziale che ha precedenti potenziali definitivi
4. **Itero** tenendo a mente che i potenziali provvisori vengono aggiornati al rialzo



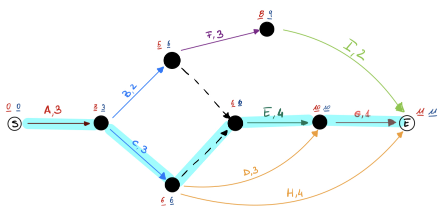
II Fase di etichettatura dei nodi

0. Andando a ritroso:
1. Il progetto inizia a potenziale 0
2. Trovo i potenziali delle attività vicine
3. Diventa definitivo il potenziale che ha tutti i successivi potenziali definitivi
4. **Itero** tenendo a mente che i potenziali provvisori vengono aggiornati al ribasso



Determinazione delle attività critiche Sia $[\underline{a}, \underline{b}] \longrightarrow^{x,l} [\underline{c}, \underline{d}]$ un arco, si definisce cammino critico se

$$\begin{cases} a = b \\ c = d \\ d - l = a \end{cases}$$



Determinazione dei tempi e degli slittamenti

- **Slittamento Libero** $[S_{ij}]$:

$$S_{ij} \doteq T_{max}(j) - T_{min}(i) - d_{ij}, \quad \forall \ attivita'$$

  Per le attività critiche $S_{ij} = 0$
  Quantità di tempo per la quale l'attività può essere ritardata senza causare un ritardo nel progetto.

- **Early Start** $[ES_{ij}]$: $ES_{ij} = T_{min}(i)$

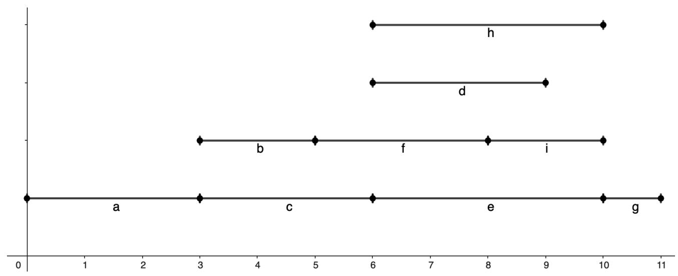
- **Early Finish** $[EF_{ij}]$: $EF_{ij} = ES_{ij} + d_{ij}$

- **Late Start** $[LS_{ij}]$: $LS_{ij} = T_{max}(j) - d_{ij}$

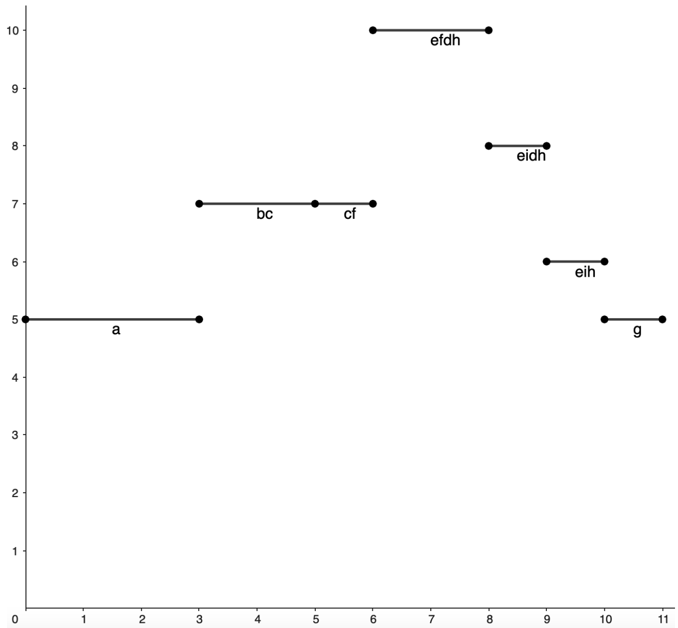
- **Late FInish** $[LF_{ij}]$: $LF_{ij} = T_{max}(j)$

N.B.: Se un'attività è critica logicamente $ES_{ij} = LS_{ij}$ e $EF_{ij} = LF_{ij}$

6.4 Diagramma di Gantt



6.5 Diagramma dei carichi di risorse



7 Program Evaluation and Review Technique [PERT]

Andiamo a definire subito:

- **a**: Durata ottimistica
- **b**: Durata pessimistica
- **M**: Durata più probabile

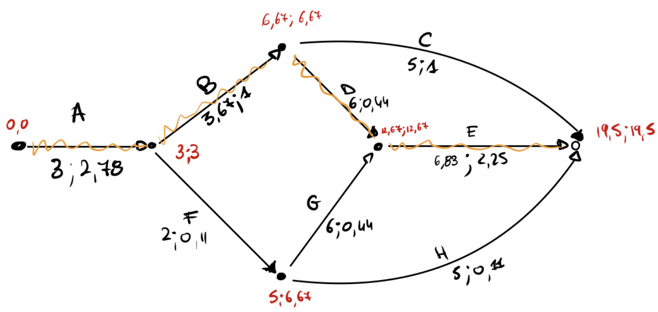
Definiamo anche la media m e la varianza σ come:

$$m = \frac{a + 4M + b}{6} \quad e \quad \sigma^2 = \frac{(b-a)^2}{36}$$

Esempio:

Attività	a	M	b	Precedenze	m	sigma^2
	Durate					
A	1	1,5	11	/	3,00	2,78
B	2	3	8	A	3,67	1,00
C	2	5	8	B	5,00	1,00
D	4	6	8	B	6,00	0,44
E	2	7	11	D, G	6,83	2,25
F	1	2	3	A	2,00	0,11
G	4	6	8	F	6,00	0,44
H	4	5	6	F	5,00	0,11

Il sistema di etichettatura è uguale a quello visto per il problema del CPM tenendo il considerazione il parametro a



Ipotesi

1. Numero di attività dei progetti $Pj \geq 10$
2. Le variabili aleatorie delle durate devono essere indipendenti e identicamente distribuite (i.i.d)

Durata Minima : La durata minima del progetto è una V.A. che ha

$$m = \sum medie$$

$$\sigma = \sum varianze$$

Qual è la probabilità che il progetto duri almeno 22 giorni ($P(t \geq 22)$)?

$$\begin{array}{l} M_p = 19,5 \\ \sigma_p^2 = 6,47 \end{array} \Rightarrow z = \frac{t - M_p}{\sigma} \Rightarrow P(t \geq 22) = 1 - P\left(z \geq \frac{22 - 19,5}{\sqrt{6,47}}\right) \simeq 16,35\%$$

8 Travelling Salesman Problem [TSP]

Il TSP è un problema di routing su nodi. Gli obiettivi del TSP sono due:

- Visitare tutti i nodi
- Non passare più volte per un nodo già visitato

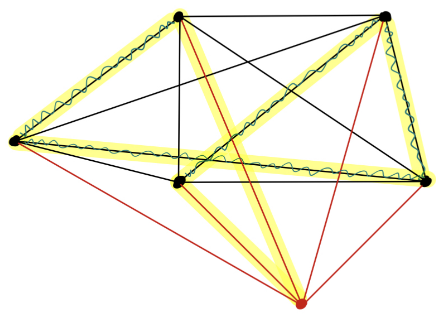
8.1 Definizioni

Circuito Hamiltoniano [CH]: Circuito che tocca tutti i nodi di una rete una ed una sola volta.
L'obiettivo del TSP è quello di andare a trovare un CH a costo minimo.

8.2 Sottoproblemi del TSP

8.2.1 PB: Percorso Hamiltoniano minimo tra 2 nodi qualunque

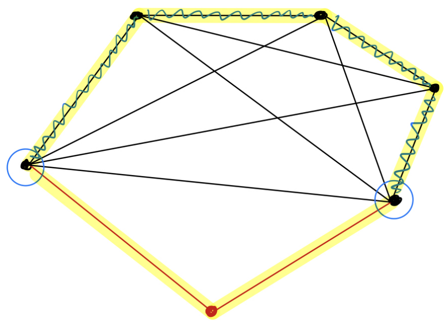
Ipotesi: Sia data una rete completa



1. Aggiungo un nodo fittizio e lo connetto a tutta la rete a costo 0 ■
2. Trovo un CH ■
3. Elimino gli archi che collegano il nodo fittizio e mi ricavo il PH ■

8.2.2 PB: Percorso Hamiltoniano minimo tra 2 nodi scelti

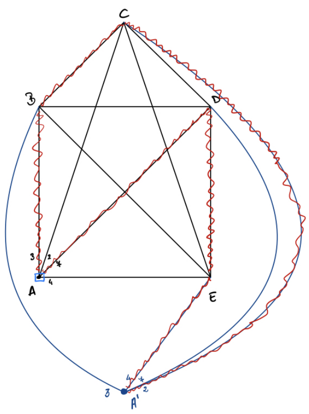
Ipotesi: Sia data una rete completa

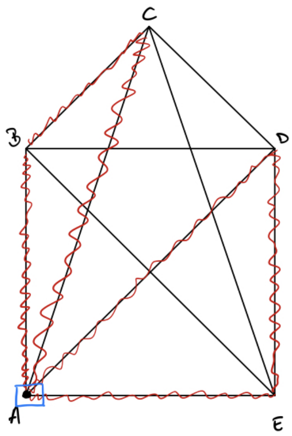


1. Aggiungo un nodo fittizio e lo connetto **solo ai due nodi scelti** a costo 0
2. Trovo un CH
3. Elimino gli archi che collegano il nodo fittizio e mi ricavo il PH

8.3 M-TSP

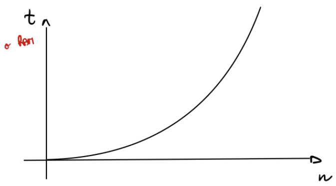
$(M-1)$ copie del nodo di partenza. Se $M = 2$ abbiamo un $2 - TSP$.
$M = 2 \Rightarrow M - 1 = 2 - 1 = 1$ copia



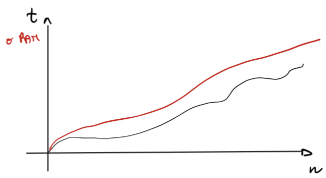


8.4 Complessità computazionale

Algoritmo Efficace: Algoritmo che, ammesso che esista, è in rado di trovare una soluzione ottima.
Il tempo potrebbe crescere un modo esponenziale.



Algoritmo Efficiente: Algoritmo che ci da una soluzione ammissibile in tempo polinomiale.$\Rightarrow$ può essere maggiorato con qualcosa



AD OGGI NON ESISTONO ALGORITMI EFFICACI ED EFFICIENTI

Il TSP è un problema NP-hard e NP-completo.
Quante sono le soluzioni ammissibili per un problema TSP? $(N-1)!$

Disuguaglianza Triangolare:

$$c_{ij} \le c_{ik} + c_{kj}, \quad \forall i, j, k$$

- Il TSP si dirà metrico se l'istanza gode della disuguaglianza triangolare
- Un TSP metrico è euclideo se le distanze sono proporzionali

Ipotesi che faremo:

- $c_{ij} \ge 0$
- TSP Metrico
- Distanze simmetriche

8.5 Euristiche per il TSP

8.5.1 Euristiche costruttive

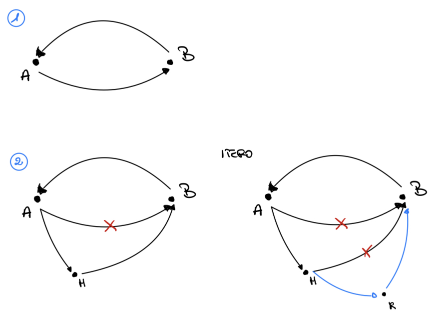
Nota Bene: Se il TSP è euclideo e c'è un intersezione $\Rightarrow$ può essere migliorato!

Euristiche di inserimento

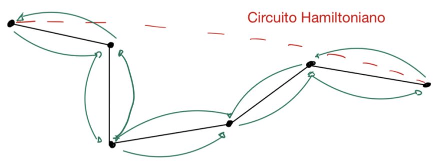
- Euristica del nodo più vicino (Eff. $\simeq 20\%$ in più rispetto all'ottimo)
- Euristica del nodo più lontano (Eff. $\simeq 10\%$ in più rispetto all'ottimo)
- Euristica del nodo più economico (Eff. $\simeq 17\%$ in più rispetto all'ottimo)
- Euristica di nodo casuale (Eff. $\simeq 11\%$ in più rispetto all'ottimo)

Spiegazione - Nodo più vicino:

1. Creo un circuito parziale con due nodi della rete
2. Inserisco via via dei nodi non considerati



Euristica del doppio albero ricoprente



$$c(MST) \leq c(PH) \leq c(CH)$$

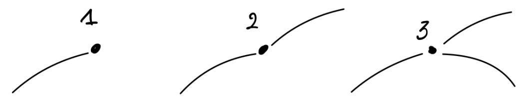
ma

$$c(MST) \leq c(TSP) \leq 2 \cdot c(MST)$$

Se vale la disuguaglianza triangolare $\Rightarrow$ $c(MST) \leq c(TSP) \leq c(CH) \leq 2 \cdot c(MST)$

1. Costruisco un MST
2. Ottengo un circuito non hamiltoniano raddoppiando gli archi
3. Sfrutto la disuguaglianza triangolare per ottenere un circuito hamiltoniano

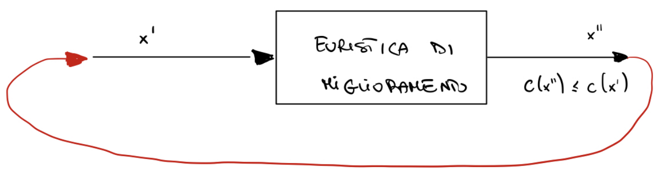
Euristica di Christofides
Ipotesi: TSP Metrico
Ordine di un nodo: # di archi incidenti al nodo



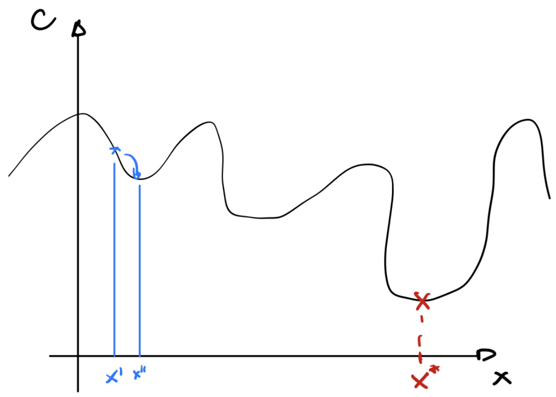
Proprietà: Il # di nodi che ha ordine dispari sono SEMPRE pari.
Nel TSP metrico non euclideo:

- $d(i,j) = d(j,i)$
- $d(i,j) = 0 \Leftrightarrow i = j$
- $d(i,j) \geq 0, \quad \forall i,j$
- $d(i,j) \leq d(i,k) + d(k,j)$

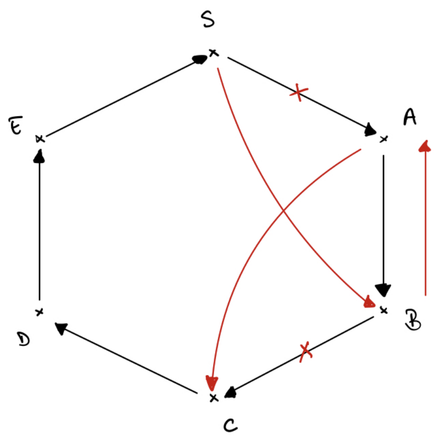
8.5.2 Euristiche di miglioramento



Un euristica di miglioramento trova un minimo locale, non è detto che quella sia la soluzione ottima



Euristica 2-OPT



Si basa sullo scambio di archi. Non ci possono essere scambi tra archi adiacenti.
Prestazioni: $\simeq 8\%$ in media più del minimo

Euristica di Lin-Kernighan Arrivato a un minimo locale $k - opt$ bisogna chiedere se si vuole e si può applicare una $(k+1) - opt$

Formulazione Matematica

Variabili decisionali:

$$x_{ij} = \begin{cases} 1 & se\ (i,j) \in CH \\ 0 & altrimenti \end{cases}$$

Funzione obiettivo:

$$min\ z = \sum_{(i,j)\in A} c_{ij} \cdot x_{ij}$$

Vincoli:

$$\sum_{(i,j)\in\delta(i)} x_{ij} = 2, \quad \forall i \in \aleph$$

$$I) \qquad \sum_{(i,j)\in\delta(w)} x_{ij} \geq 2, \quad \forall w \in \aleph : 3 \leq |w| \leq \frac{|w|}{2}$$

$$II) \qquad \sum_{(i,j):i,j\in w} x_{ij} \leq |w| - 1, \quad \forall w \in \aleph : 2 \leq |w| \leq |\aleph| - 2$$

$$Dominio: \quad x_{ij} \in \{0;1\}, \ \forall (i,j) \in A$$

9 Arch Routing

Circuito Euleriano [CE]: Circuito che passa una ed una sola volta per ogni arco della rete.

Data una rete $G(\aleph, A)$, G è euleriana se esiste in essa un Circuito Euleriano.

9.1 Condizioni Necessarie e Sufficienti

9.1.1 Rete non orientata

- Connessione
- Parità $\Rightarrow$ il # di archi incidenti in un nodo deve essere pari

9.1.2 Rete orientata

- Forte connessione
- Simmetrica

9.1.3 Rete mista

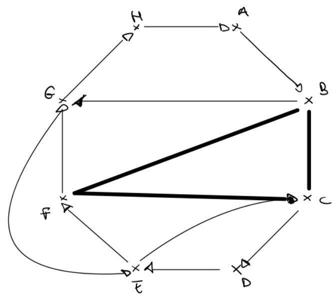
- Forte connessione
- Parità $\Rightarrow$ il # di archi incidenti in un nodo deve essere pari
- Di Bilanciamento $\rightarrow$ $\forall$ taglio, la differenza tra archi orientati in un senso e archi orientati nell'altro verso è $\leq$ degli archi non orientati

9.2 Esempio: Creazione di una rete mista

Requisiti:

- Fortemente connessa
- # nodi ≥ 8
- Pari
- Simmetrica

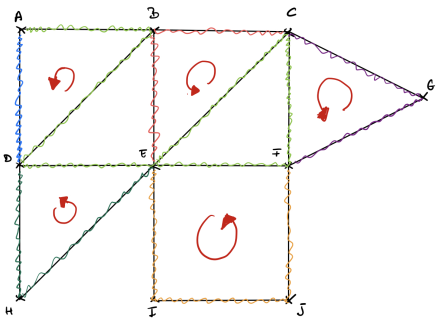
$\Rightarrow$ Rete euleriana non banale



9.3 Algoritmo Risolutivo

9.3.1 Algoritmo di End-Pairing

1. Crea dei circuiti euleriani che non condividono archi ■
2. Partendo da un punto scelto (o a caso a seconda del problema) inizio a percorrere il circuito euleriano che lo include fino a quando non incontro un altro circuito euleriano. ■
3. Se incontro un altro circuito euleriano inizio a percorrere quello
4. Itero i punti 2. e 3. fino al completamento dei circuiti a disposizione ■ → ■ → ■ → ■
5. Per concludere il CE vado a ritroso partendo dall'ultimo circuito percorso ■



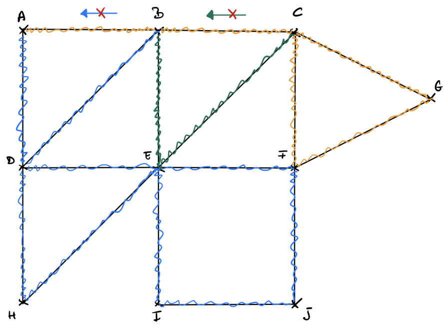
Nel nostro caso avevamo i sottocircuiti

- $A - \underline{D} - B - A$
- $D - H - \underline{E} - D$
- $E - I - J - \underline{F} - E$
- $F - G - \underline{C} - F$
- $C - B - E - C$

Circuito Finale: $A - D - H - E - I - J - F - G - C - B - E - C - F - E - D - B - A$

9.3.2 Algoritmo di Fleury

1. Parto da un arco qualunque
2. Scelgo un arco adiacente a caso controllando solo che l'arco non sia un bridge (ossia un arco che sconnette tra loro gli archi che devono ancora essere visitati) ■ → ■ → ■
3. Continuo la visita su archi adiacenti



9.4 Problema del Postino Cinese [CPP]

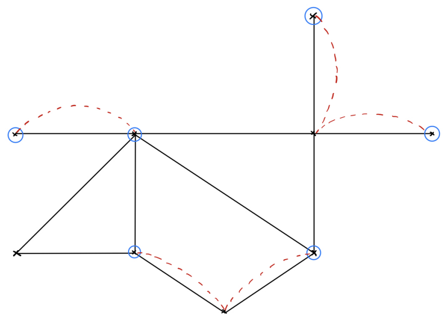
CPP: Chinese Postman Problem
É un problema di Arch routing che si basa su una rete $G(\aleph, A)$ con $c_{ij} \geq 0 \quad \forall (i,j) \in A$. L'obiettivo del CPP è quello di trovare un circuito che passi per ogni arco della rete una o più volte a costo minimo.

Come risolvere un CPP?

- Se la rete è euleriana utilizzo o l'algoritmo di end-pairing o quello di Fleury e il costo sarà $= \sum c_{ij}$
- Per le reti non euleriane lo vediamo di seguito

9.4.1 Rete non euleriana e non orientata

1. Individuo i nodi dispari ■
2. Collego i nodi dispari in maniera tale da renderli pari e ottenere cosi una rete euleriana ■
3. Applico l'algoritmo di Fleury o end-pairing



9.4.2 Rete non euleriana e orientata

Sia data una rete $G(\aleph, A)$ e sia $\delta_i = \textit{archi orientati entranti} - \textit{archi orientati uscenti}$ in ogni nodo.
Definiamo:

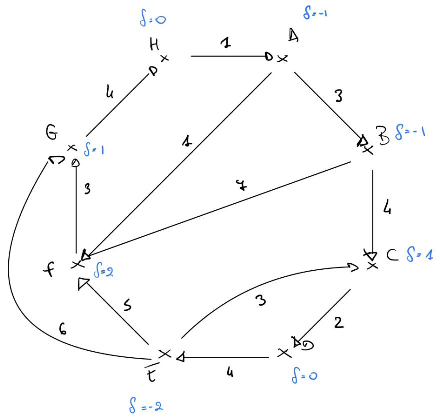
- D^+ come l'insieme dei nodi per i quali $\delta > 0$
- D^- come l'insieme dei nodi per i quali $\delta < 0$

$\forall i, j$ con $i \in D^+$ e $j \in D^-$ calcolo il costo w_{ij} come il costo di un percorso a costo minimo tra i e j

Esempio

Requisiti:

- Rete orientata
- 8 nodi
- Fortemente connessa
- 12 archi
- Non euleriana



$$D^+ = \{C;\ F;\ G\} \qquad D^- = \{A;\ B;\ E\}$$

$$w_{CE} = 6$$

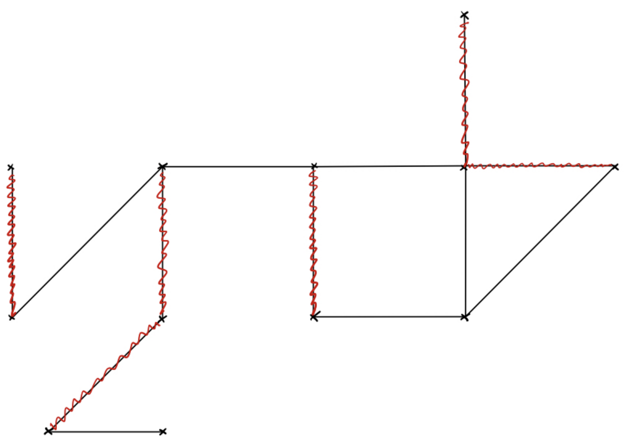
F.O. $min\ z = 17x_{CA} + 20x_{CB} + 6x_{CE} + 8x_{FA} + 11x_{FB} + 21x_{FE} + 5x_{GA} + 8x_{GB} + 18x_{GE}$

Vincoli

$$\begin{array}{ll}
x_{CA} + x_{CB} + x_{CE} \leq 1 & (\delta_C = 1) \\
x_{FA} + x_{FB} + x_{FE} \leq 2 & (\delta_F = 2) \\
x_{GA} + x_{GB} + x_{GE} \leq 1 & (\delta_G = 1) \\
 & \\
x_{CA} + x_{FA} + x_{GA} \geq 1 & (-\delta_A = 1) \\
x_{CB} + x_{FB} + x_{GB} \geq 1 & (-\delta_B = 1) \\
x_{CE} + x_{FE} + x_{GE} \geq 2 & (-\delta_E = 1) \\
 & \\
x_{ij} \geq 0 & \forall i \in D^+;\ \forall j \in D^-
\end{array}$$

9.5 Problema del Postino Rurale [RPP]

RPP: Rural Postman Problem
É un problema di Arch routing che si basa su una rete $G(\aleph, A)$ con $c_{ij} \geq 0 \ \ \forall (i,j) \in A$. All'interno del RPP però definiamo un insieme $R \subset A$ dove il numero di archi non è la totalità ma sono sconnessi in più parti.

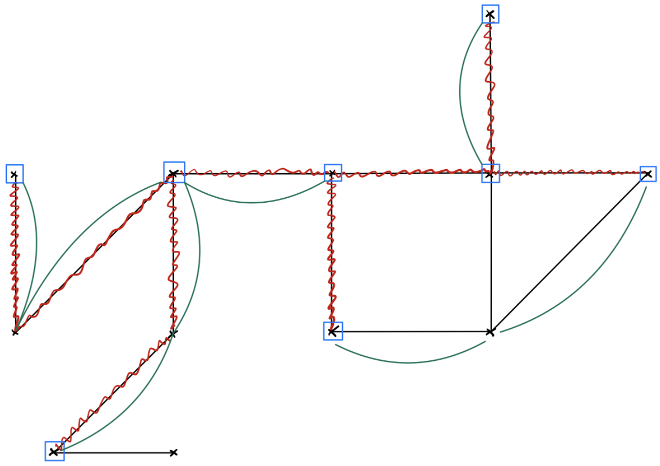


Soluzione: Circuito che passa su tutti gli archi di R almeno 1 volta
Obiettivo: A costo minimo

Definiamo G_R come rete indotta dagli archi R in G. G_R può essere $\begin{cases} Connessa \\ Sconnessa \end{cases}$

9.5.1 G_R connessa

:



Dove G definisce tutta la rete, G_R definisce la rete in rosso (■), con il ■ evidenzio i nodi di ordine dispari della rete G_R e con il ■ evidenzio un matching ottimo ipotetico.

Se la rete è euleriana ci ritroviamo nel caso super semplice → applico Fleury o end-pairing.

Se la rete NON è euleriana:

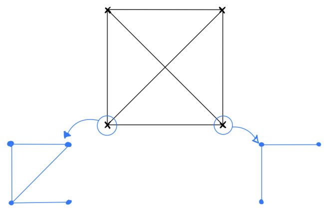
1. Matching ottimo considerando tutta la rete G
2. Dopo il matching ottimo ottengo una rete $\overline{G_R}$ ■
3. Applico l'algoritmo di Fleury o end-pairing

9.5.2 G_R NON connessa

Se G_R non è connessa ⇒ devo usare un approccio euristico poiché è un problema NP-hard.

Euristica Balance & Connect

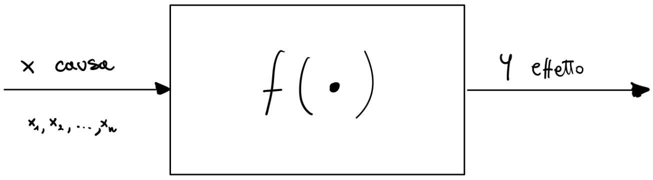
1. Rendere pari G_R
2. Creo una rete ausiliaria completa dei percorsi minimi con nodi le componenti connesse di G_R



3. Risolvo un MST sulla rete ausiliaria
4. Dopo il MST la "nuova" G_R risulterà connessa
5. Mi sono ricondotto al caso G_R connessa

10 Tecniche di Previsione

10.1 Modelli Causali



10.1.1 Covarianza

$Cov(X,Y) = E((X-\mu_x)(Y-\mu_y)) = E(XY - \mu_x Y - \mu_y X + \mu_x\mu_y) = E(XY) - \mu_x E(Y) - \mu_y E(X) + \mu_x\mu_y = E(XY) - \mu_x\mu_y$

Definiamo $Y = f(x,e)$ con e variabile aleatoria a media $= 0$

$$Y_i = a + bX_i + e_i$$
$$e_i = Y_i - (a + bX_i)$$

Dimostriamo che questo vale: $min(\sum_i e_i^2) = min\sum_i(Y_i - (a+bX_i))^2 = \sum_i(Y_i^2 + (a + bX_i)^2 - 2Y_i(a+bX_i)) = \sum_i(Y_i^2 + a^2 + b^2X_i^2 - 2abX_i - 2aY_i - 2bY_iX_i)$

$\begin{cases} \frac{\partial}{\partial a} = \sum_i(2a + 2bX_i - 2Y_i) = 0 \\ \frac{\partial}{\partial b} = \sum_i(2bX_i^2 + 2aX_i - 2X_iY_i) = 0 \end{cases}$

$\begin{cases} 2aN + 2b\sum X_i - 2\sum Y_i = 0 \\ 2b\sum X_i^2 + 2a\sum X_i - 2\sum X_iY_i \end{cases}$

Se denoto come $A = \sum X_i,\ B = \sum Y_i,\ C = \sum X_i^2,\ D = \sum X_iY_i$

Se isolo a nella I equazione ottengo $a = \frac{B-bA}{N}$ nella II ottengo $bC + \left(\frac{B-bA}{N}\right)A - D = 0 \Rightarrow bC + \frac{AB}{N} - \frac{bA^2}{N} - D = 0 \Rightarrow b(C - \frac{A^2}{N}) = \frac{CN-AB}{N} \Rightarrow b = \frac{DN-AB}{CN-A^2}$

Esplondendo A, B, C, D si ottiene: $\begin{cases} \hat{a} = \frac{\sum Y_i}{N} - \hat{b}\frac{\sum X_i}{N} \Rightarrow \hat{a} = \mu_y - \hat{b}\mu_x \\ \hat{b} = \frac{N\sum X_iY_i - \sum X_i \sum Y_i}{N\sum X_i^2 - (\sum X_i)^2} \end{cases}$

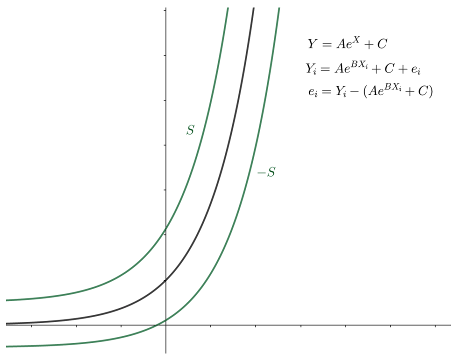
$\hat{b} = \frac{\sum X_iY_i - \mu_x\sum Y_i}{\sum X_i^2 - \frac{(\sum X_i)^2}{N}} = \frac{\sum Y_i(X_i + \mu_x)}{\sum (X_i - \mu_x)^2}$

Dimostriamo che $\sum(X_i - \mu_x)^2$ equivale al denominatore $\sum X_i^2 - \frac{(\sum X_i)^2}{N}$:

$\sum(X_i - \mu_x)^2 = \sum(X_i^2 + \mu_x^2 - 2X_i\mu_x) = \sum X_i + \mu_x^2 N - 2\mu_x\sum X_i = \sum X_i + \frac{(\sum X_i)^2 N^2}{N^2} - 2\mu_x\sum X_i = \sum X_i^2 + \mu_x\sum X_i - 2\mu_x\sum X_i = \sum X_i - \mu_x\sum X_i = \sum X_i^2 - \frac{(\sum X_i)^2}{N}$

10.1.2 Coefficiente di correlazione lineare $[P_{x,y}]$

$$P_{x,y} = \frac{Cov(X,Y)}{\sigma_x \cdot \sigma_y} \qquad -1 \leq P_{x,y} \leq 1$$



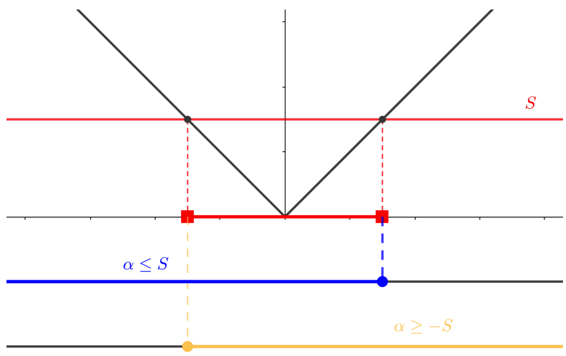
Siano (x_i, y_i) $\;i = 1, \dots, N$ e $|e_i| \leq S \quad \forall i = 1, \dots, N$

Variabili Decisionali $: A,\ B,\ C,\ S$

Funzione obiettivo $: min\ z = S$

Vincoli $: |e_i| \leq S, \quad i = 1, \dots, N$
Dominio: $S \geq 0$

10.1.3 Parentesi $|\alpha| \leq S$



Se metto a sistema $\begin{cases} \alpha \leq S \\ \alpha \geq -S \end{cases}$ ottengo il tratto in ■

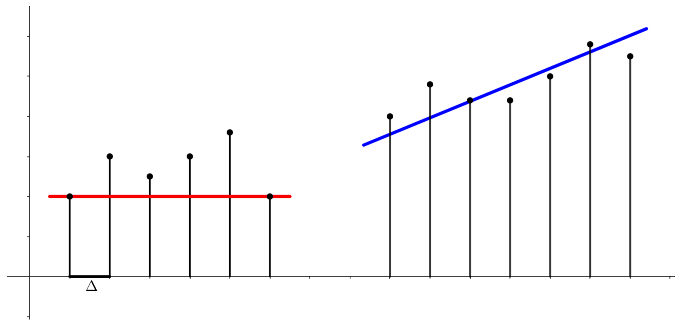
I vincoli diventano: $\begin{cases} Y_i - Ae^{BX_i} - C \leq S & i = 1, \dots, N \\ Y_i - Ae^{BX_i} - C \geq -S & i = 1, \dots, N \\ S \geq 0 \end{cases}$ ■ ■

Se $B = 1$ (per togliere un grado di liberà ottengo un problema di programmazione lineare)

10.2 Serie Temporali

$$(t, Y_t)$$

Se ritengo che il fenomeno sia autocorrelato ha senso sviluppare tecniche di previsione.



■: **Stazionarietà**
■: **Trend** → Andamento di lungo periodo crescita/decrescita
Stagionalità: Pattern che tende a ripetersi con una certa stagionalità
Ciclicità: Pattern che tende a ripetersi con archi di tempo superiori alle *stagioni/anno*
Mera casualità

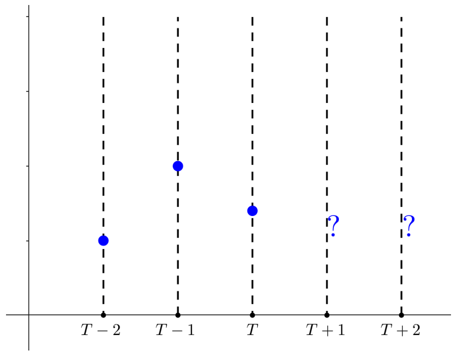
10.2.1 Definizioni

d_T = Dato osservato nel periodo T
P_{T+1} = Previsione per il periodo successivo
e_T = Errore tra previsione e dato osservato $e_T = P_T - d_T$

10.2.2 Misure di accuratezza della previsione

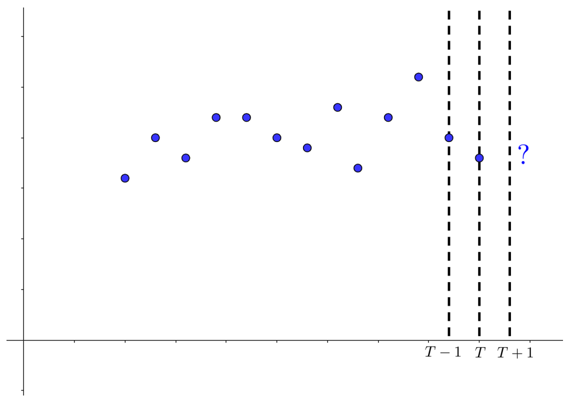
MAD (Mean Absolute Deviation) $= E(|e_T|)$
MSE (Mean Square Error) $= E\left(e_T^2\right)$
MAPE (Mean Absolute Percentage Error) $= E\left(\left|\frac{e_T}{d_T}\right|\right)$ [%]

Supponiamo di usare una tecnica di previsione; confrontiamo i dati raccolti con i dati della previsione e guardo gli errori; trovo un parametro da aggiustare nella mia previsione e lo aggiusto minimizzando l'errore da esso commesso.
⇒ **NON SI FÁ**! Non è detto che ciò migliori le previsioni future.



10.3 Serie Temporali Stazionarie

Andamento di lungo periodo costante



- **Tecnica elementare**: $P_{T+1} = d_T$ (per $T+1$ prevedo l'ultimo valore assegnato)
- **Media Mobile**:
$$P_{T+1} = \sum_{k=0}^{r-1} \frac{d_T - k}{r} \qquad n \geq 1$$
Se $r = 1$ rientro nella tecnica elementare.
La media mobile è la media aritmetica delle ultime n osservazioni
$n = 2 \;\; P_{T+1} = \frac{d_T}{2} + \frac{d_{T-1}}{2}$
Se $T < 3 \rightarrow P_{T+1} = \sum \frac{d_T - k}{T}$
Outlier: Dato che si disallinea sul trend è meglio ignorarlo
Variabili aleatorie indipendenti (in realtà sono autocorrelate ma faremo una semplificazione analitica), identicamente distribuite con una media μ e una deviazione standard σ
$$\mu_{T+1} = E\left(\sum_{k=0}^{r-1} \frac{d_t - k}{r}\right) = \sum_{k=0}^{r-1} \frac{1}{n} E(d_T - k) = \frac{1}{r} \sum_{k=0}^{r-1} \mu = \frac{1}{r} \cdot r \cdot \mu = \mu$$

$$Var(P_{T+1}) = Var\left(\sum_{k=0}^{n-1} \frac{d_T - k}{r}\right) = \sum_{k=0}^{r-1} Var\left(\frac{d_T - k}{r}\right) = \sum_{k=0}^{r-1} \frac{1}{r^2} Var(d_T - k) =$$

$$= \frac{1}{r}\sum_{k=0}^{r-1} \sigma^2 = \frac{1}{r^2} \cdot r \cdot \sigma^2 = \frac{\sigma^2}{r}$$

Richiamo:

$$Var(x) = E((X - E(X))^2) = E(X^2 + E(X)^2 - 2XE(X)) =$$

$$= E(X^2) + E(X)^2 - 2E(X)^2 = E(X^2) - E(X)^2$$

In definitiva

$$Var(X) = Cov(X, X)$$

inoltre

$$Var(\alpha X) = E(\alpha^2 X^2) - E(\alpha X)^2 = \alpha^2 E(X^2) - (\alpha E(X))^2 = \alpha^2 E(X^2) - \alpha^2 E(X)^2$$

Quindi

$$Var(\alpha X) = \alpha^2 Var(X)$$

In ultima battuta

$$\sigma_{P_{T+1}} = \frac{1}{\sqrt{n}} \cdot \sigma \Rightarrow \text{più è grande n, minore è la dimensione complessiva}$$

Formula Alternativa Media Mobile:

$$P_{T+2} = \sum_{k=0}^{r-1} \frac{d_{T+1} - k}{r} = \frac{d_{T+1}}{r} + \sum_{k=1}^{r-1} \frac{d_{T+1} - k}{r} = \frac{d_{T+1}}{r} + \sum_{k=0}^{r-2} \frac{d_{T+1} - k}{r} =$$

$$= \frac{d_{T+1}}{r} + \sum_{k=0}^{r-1} \frac{d_T - k}{r} - \frac{d_{T+1} - r}{r} = P_{T+1} + \frac{d_{T+1}}{r} + \frac{d_{T+1-r}}{r}$$

$\Rightarrow$ Se ho una previsione per d_{T+1} e ne voglio una per d_{T+2}, prendo P_{T+1}, sommo d_{T+1} (dato nuovo) e sottraggo d_{T+1-r} (dato più vecchio) $\rightarrow$ Media Mobile.

Media Mobile

Vantaggi	Svantaggi
Facile da implementare Bassa complessità	Sforza molto le variazioni $\rightarrow$ Non anticipa efficacemente il fenomeno

- **Media (livellamento) Esponenziale**:

$$P_{T+1} = \alpha d_T + (1 - \alpha) P_T \qquad 0 < \alpha < 1$$

- É ricorsiva
- Tiene conto di tutti i dati passati però con peso decrescente man mano che ”invecchiano”

P_{T+1} è una combinazione convessa

$$P_{T+1} = \alpha d_T + (1-\alpha)P_T = \alpha d_T + P_T - \alpha P_T = P_T + \alpha(d_T - P_T) = P_T + \alpha e_T$$

$P_{T+1} = \alpha d_T + (1-\alpha)P_T$
$P_T = \alpha d_{T-1} + (1-\alpha)P_{T-1}$

$$P_{T+1} = \alpha \sum_{k=0}^{T-2} (1-\alpha)^k d_{T-k} + (1-\alpha)^{T-1} d \qquad \text{(forma non ricorsiva)}$$

Ne consegue che:

- α Grande: molto peso alla storia recente
- α Piccolo: Molto peso alla storia passata

$P_T(\tau)$: sono in T, conosco d_T, voglio fare una previsione per τ periodi più avanti ($P_{T+\tau}$ non va bene, predispone la conoscenza di $P_{T+\tau-1}$, cosa che non conosco)
N.B.: $P_{T+i} \neq P_T(i)$ Non avendo informazioni tra P_T e $P_T(\tau)$, allora $P_T(\tau) = P_T$. In più il fenomeno è stazionario.

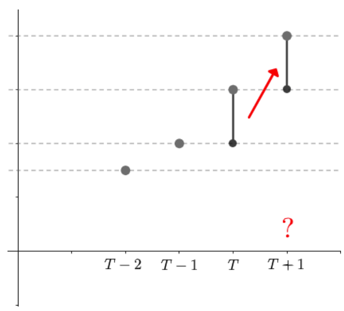
10.3.1 Serie Temporali con Trend (lineare)

$$P_T(\tau) = a_T + b_T \tau$$

$P_{T+1} = P_T(1) = a_T + b_T$
$P_T(2) = a_T + 2b_T$
etc.

- **Tecnica elementare**: Prendo il gap tra d_{T-1} e d_T e lo ripropongo per d_{T+1}

$$a_T = d_T \quad b_T = d_T - d_{T-1}$$



- **Regressione Lineare**:
$$\hat{a} = M_y - \hat{b}M_x$$
$$\hat{b} = \frac{\sum_i Y_i(X_i - M_x)}{\sum_i (X_i - M_x)^2}$$
Vedi 10.1.1

- **Doppia Media Mobile**:
$$a_T = 2\gamma_T - \eta_T \qquad b_T = \frac{2}{r-1}(\gamma_T - \eta_T)$$
ove
$$\gamma_T = \sum_{k=0}^{r-1} \frac{d_T - k}{r} \quad e \quad \eta_T = \sum_{k=0}^{r-1} \frac{\gamma_T - k}{r}$$

- **Metodo di Holt** (evoluzione del livellamento esponenziale):
$$a_T = \alpha d_T + (1-\alpha)(a_{T-1} + b_{T-1})$$
$$b_T = \beta(a_T - a_{T-1}) + (1-\beta)b_{T-1}$$

$$\begin{matrix} a_1 = d_1 \\ b_1 = 0 \end{matrix} \Rightarrow P_2 = P_1(1) = a_1 + b_1 \cdot 1 = a_1 + b_1 = d_1$$

10.3.2 Serie Temporali con Stagionalità (stazionarie)

M è il numero di periodi del ciclo stagionale

- **Tecnica elementare**: $P_T(\tau) = d_{T+\tau} - M$ (Prendi il dato più recente del periodo stagionale omologo passato).
$P(kM + \tau) = d_{T+\tau-M}$ con $k = 0, 1, \ldots$ e $\tau = 1, 2, \ldots, M$

- **Metodo della Media esponenziale revisionato**:
Schema computazionale: $P_T(\tau) = a_T \cdot S_{T+\tau}$ con $\tau = 1, \ldots, M$.
Vale anche $P_T(kM + \tau) = a_T \cdot S_{T+\tau}$.

Ipotesi non restrittiva: i cicli stagionali abbracciano periodi interi!
Definiamo quindi $K = \frac{T}{M},\ K \in \mathbb{N}$
$M = 12,\ T = 24 \rightarrow K = 2$ dove T sono i dati storici
$$a_T = \alpha\left(\frac{d_T}{S_T}\right) + (1-\alpha)a_{T-1}$$
$S_{T+\tau} = S_{kM+\tau} = \beta\left(\frac{d_{(K-1)M+\tau}}{a_{(K-1)M+\tau}}\right) + (1-\beta)s_{(k-1)M+\tau}$ con $\tau = 1, \ldots, M$ e $k \in \mathbb{N}$

Condizioni iniziali:
$a_0 = \overline{d_{(1)}} = \frac{d_1 + d_2 + \cdots + d_M}{M} \Rightarrow$ media aritmetica sui primi M periodi dei dati osservati.

$s_t = \frac{\frac{d_t}{\overline{d_{(1)}}} + \frac{d_{t+M}}{\overline{d_{(2)}}} + \cdots + \frac{d_{t+(K-1)M}}{\overline{d_{(K)}}}}{K}$
Si può dimostrare che

$$\sum_{t=1}^{M} s_t = \frac{T}{K} = M$$

- **Metodo di Winters**:
  Schema computazionale: $P_T(\tau) = (a_T + b_T \cdot \tau) \cdot s_{T+\tau}$ con $\tau = 1, 2, \ldots, M$
  Generalizzando: $P_T(kM + \tau) = (a_T + b_T \cdot \tau) \cdot s_{T+\tau}$ con $\tau = 1, 2, \ldots, M$ e $k = 0, 1, 2, \ldots$
  $K = \frac{T}{M} \in \mathbb{N}$ per ipotesi

$$a_T = \alpha \left(\frac{d_T}{s_T} \right) + (1 - \alpha)(a_{T-1} + b_{T-1})$$

$$b_T = \eta(a_T - a_{T-1}) + (1 - \eta)b_{t-1}$$

$$s_{T+\tau} = s_{kM+\tau} = \beta \left(\frac{d_{(K-1)M+\tau}}{a_{(K-1)M+\tau}} \right) + (1 - \beta)s_{(K-1)M+\tau}$$

con $\tau = 1, 2, \ldots, M$

N.B.: $0 < \alpha, \eta, \beta < 1$

Condizioni iniziali:
$b_0 = \frac{\overline{d_{(K)}} - \overline{d_{(1)}}}{T-M}$
$a_0 = \overline{d_{(1)}} - b_0 \left[\frac{(M+1)}{2} \right]$
$s_t = \frac{d_t}{(a_0 + b_0 \cdot t)}$ con $t = 1, 2, \ldots, M$

Esempio: $M = 4$; $T = 12$; $K = \frac{T}{M} = 3$