

Modellizzazione dei Sistemi Logistici

I modelli di localizzazione

Corso di Studi in Ingegneria Gestionale

a.a. 2020-2021

giuseppe.bruno@unina.it

Docente: Prof. Giuseppe Bruno

- **Classificazione**
- **I modelli orientati ai costi**
 - Simple Plant Location
 - P-mediana
- **I modelli di copertura**
 - Set Covering
 - Maximal Covering

Definizione

Un problema di localizzazione consiste nella individuazione della posizione da assegnare ad un insieme di strutture (**servizi o facilities**), in uno spazio di localizzazione, in funzione della distribuzione di una **domanda**, reale o potenziale, relativa alla loro utilizzazione.

Spazio di localizzazione e tipo localizzazione

Rappresentazione della domanda/offerta

Metrica per la valutazione delle distanze

Interazione domanda/offerta

Criteri di localizzazione

Elementi di un problema di localizzazione

Criteri di localizzazione

La funzione obiettivo è in generale rappresentata da una funzione di costo da minimizzare o una valutazione di benefici da massimizzare.

I criteri più frequentemente considerati sono:

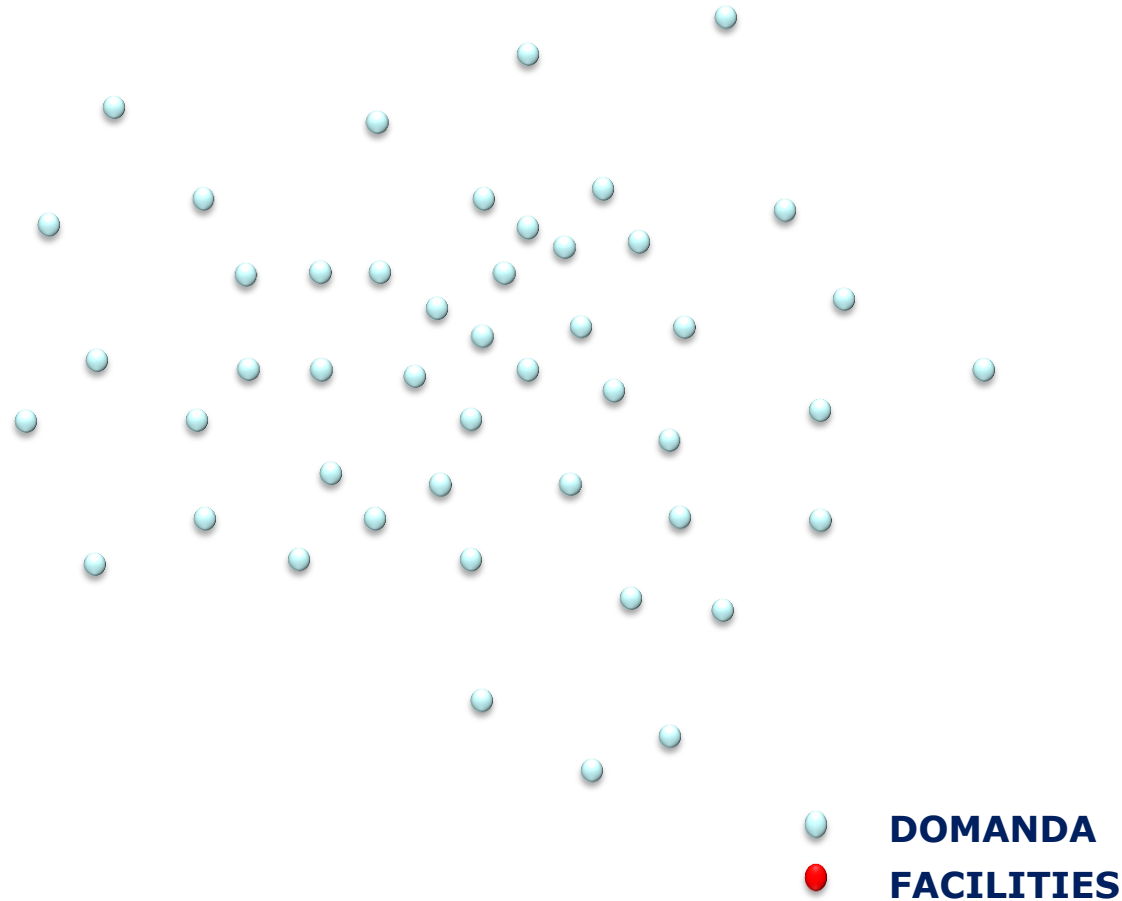
Costi di afferenza: rappresentano una funzione della distanza della domanda dai servizi disponibili

Costi di localizzazione: indicano il costo necessario all'apertura ed al mantenimento dei servizi

Copertura: rappresenta la domanda totale **coperta** dai servizi ovvero la domanda totale che si trova entro una certa distanza da uno dei servizio (raggio di copertura)

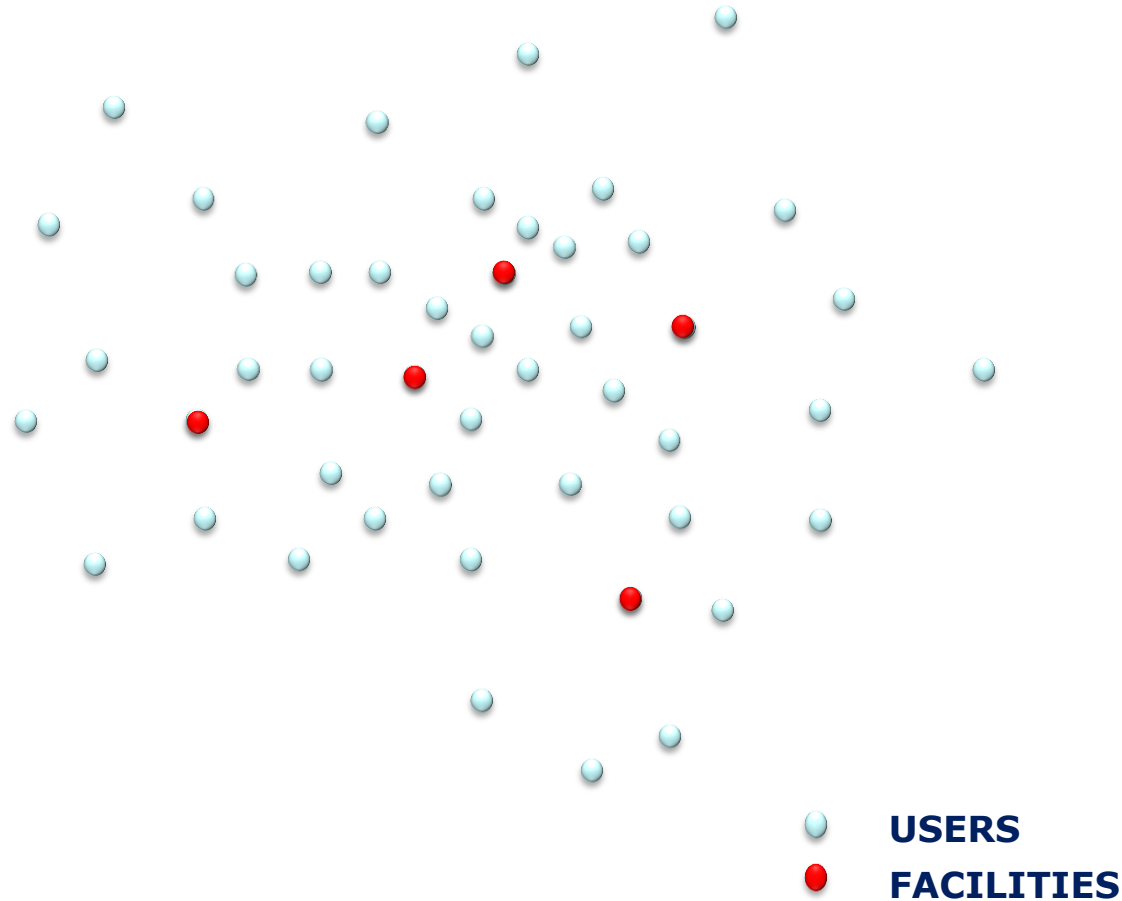
Elementi di un problema di localizzazione

Criteri di localizzazione



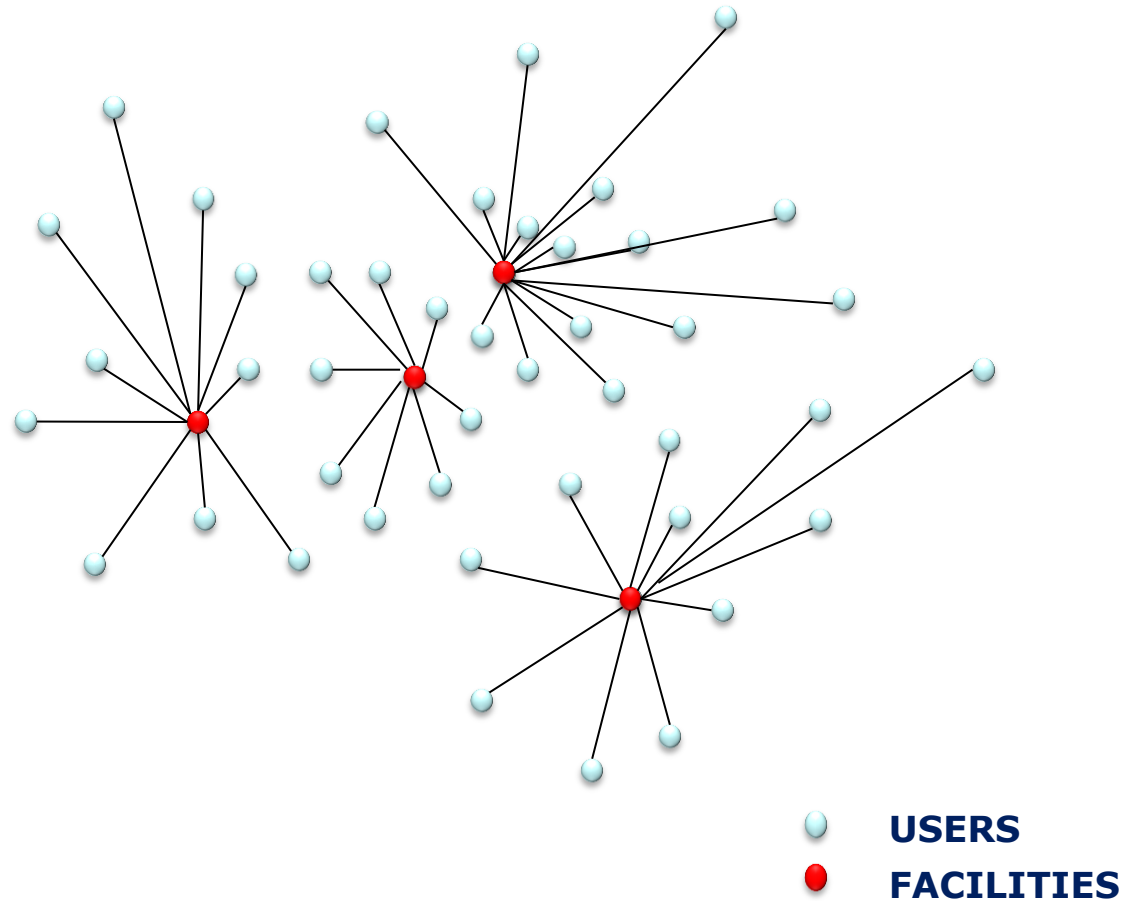
Elementi di un problema di localizzazione

Criteri di localizzazione



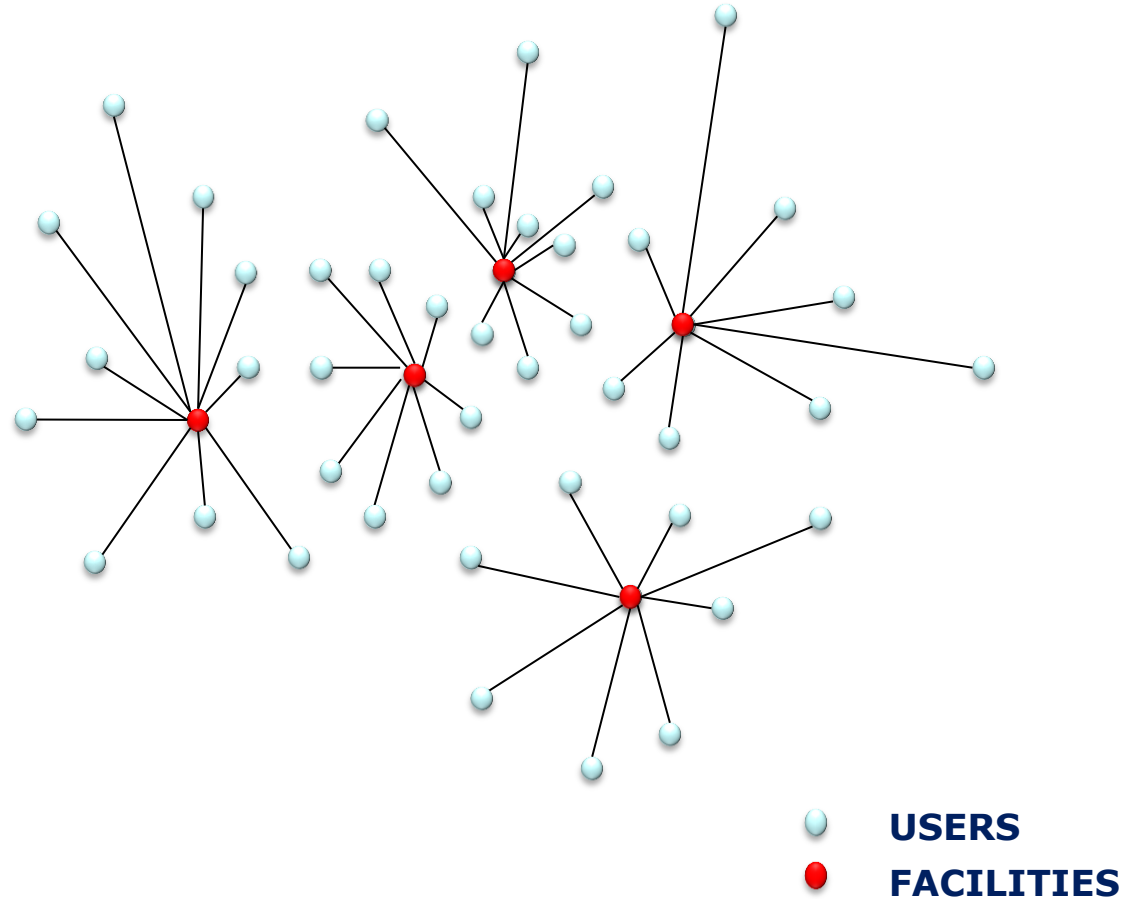
Elementi di un problema di localizzazione

Criteri di localizzazione



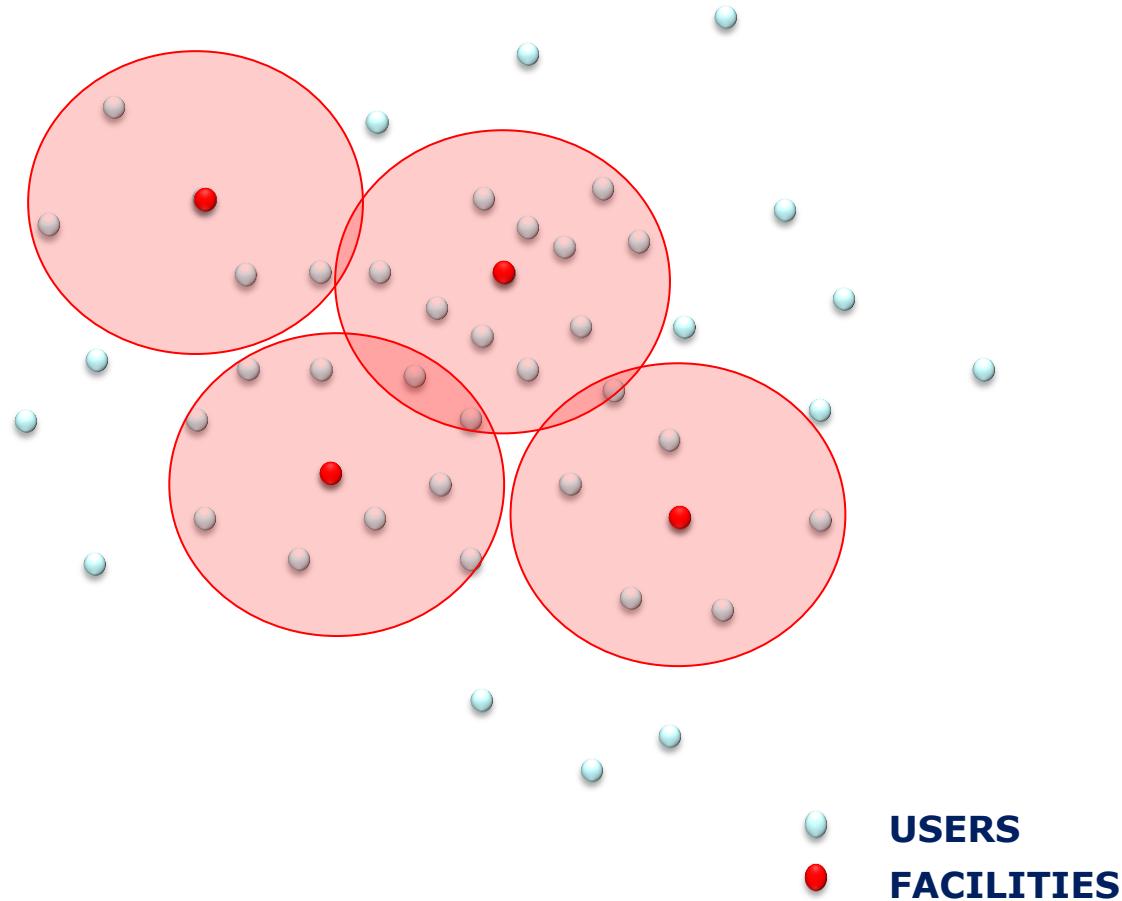
Elementi di un problema di localizzazione

Criteri di localizzazione



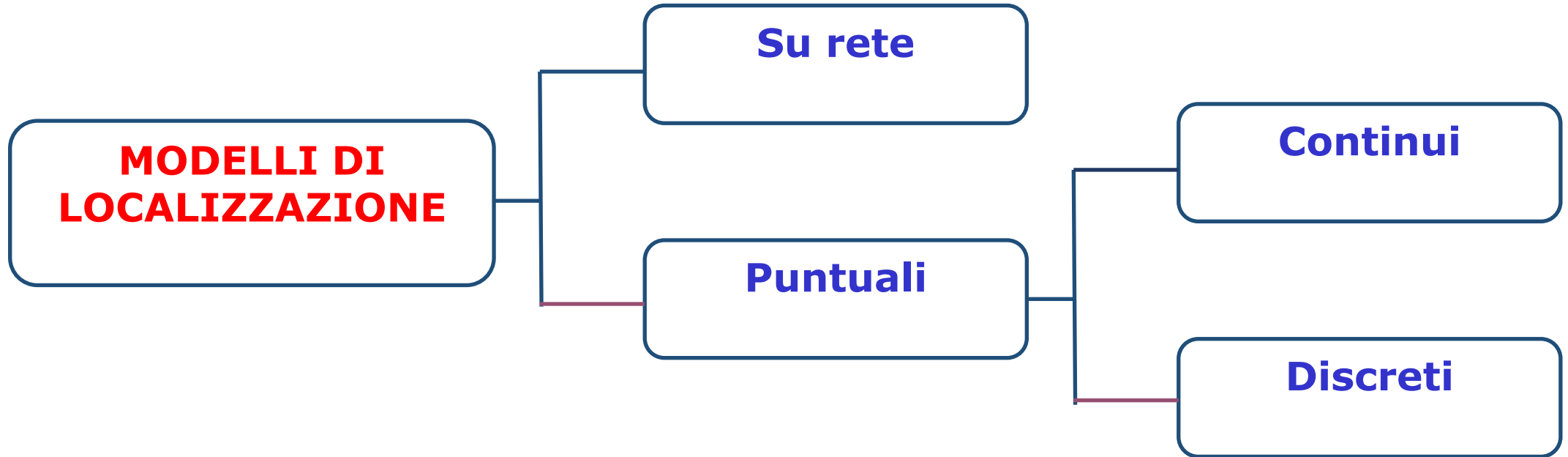
Elementi di un problema di localizzazione

Criteri di localizzazione



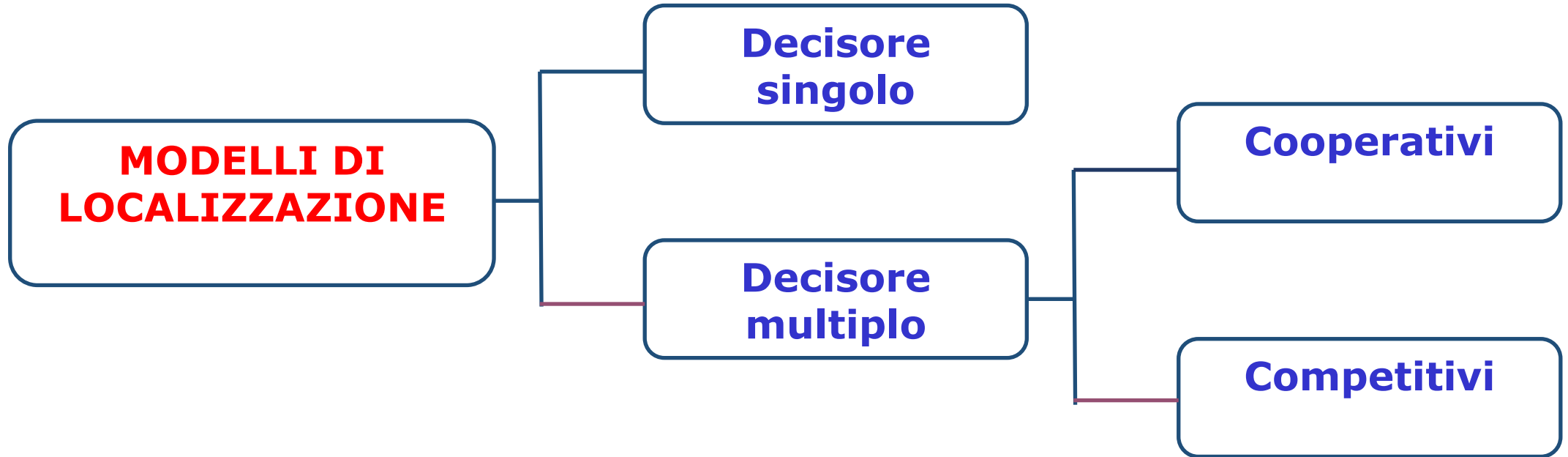
Classificazione dei modelli di localizzazione

Sulla base dello spazio e del tipo di localizzazione

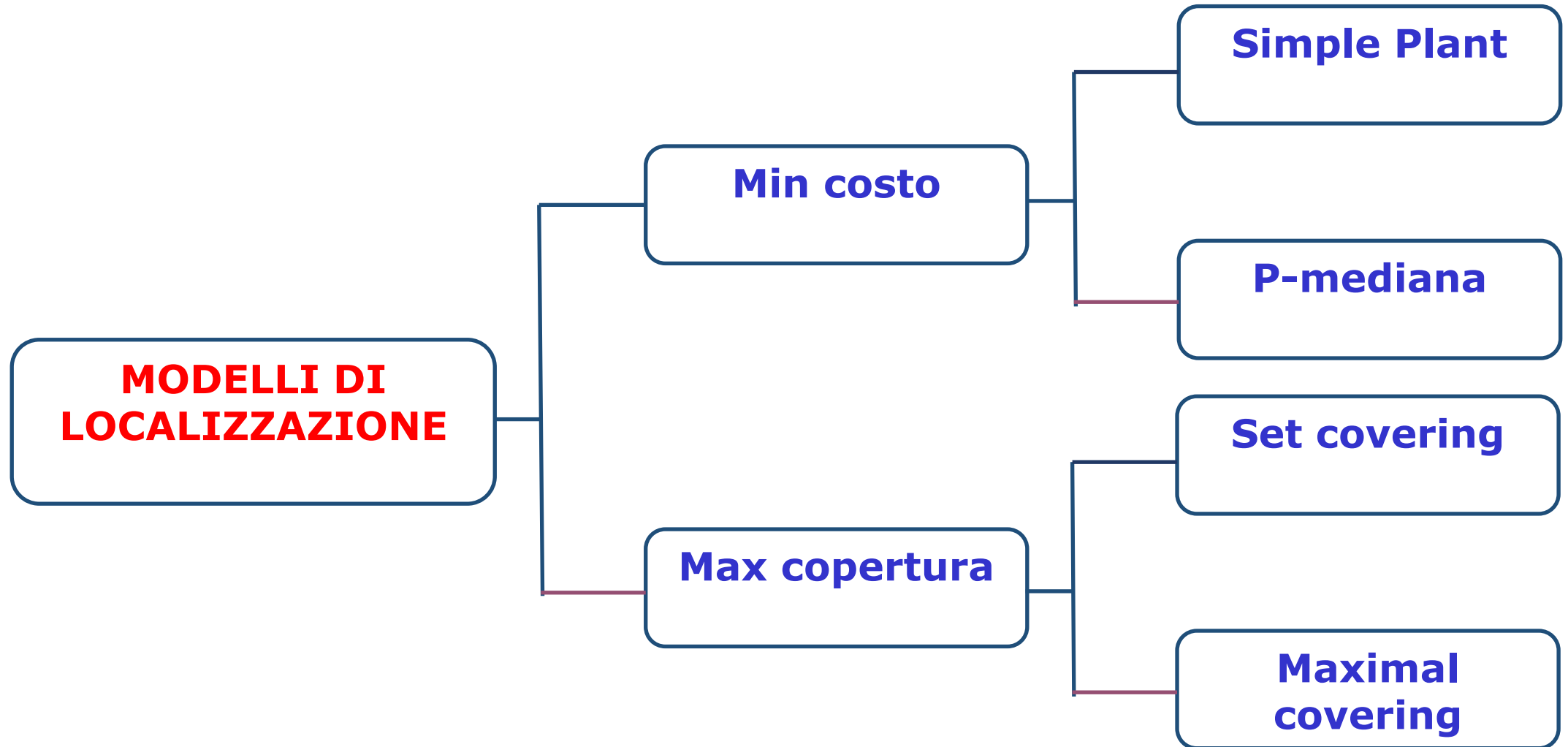


Classificazione dei modelli di localizzazione

Sulla base del numero di decisori



Classificazione dei modelli di localizzazione



Elementi di un problema di localizzazione

Altri vincoli possibili

Nella pratica possono essere presenti diversi vincoli aggiuntivi.

Esempi tipici sono vincoli di

budget sui costi di localizzazione

capacità massima dei servizi

distanza (minima o massima) tra servizi aperti

Bilanciamento tra le facilities

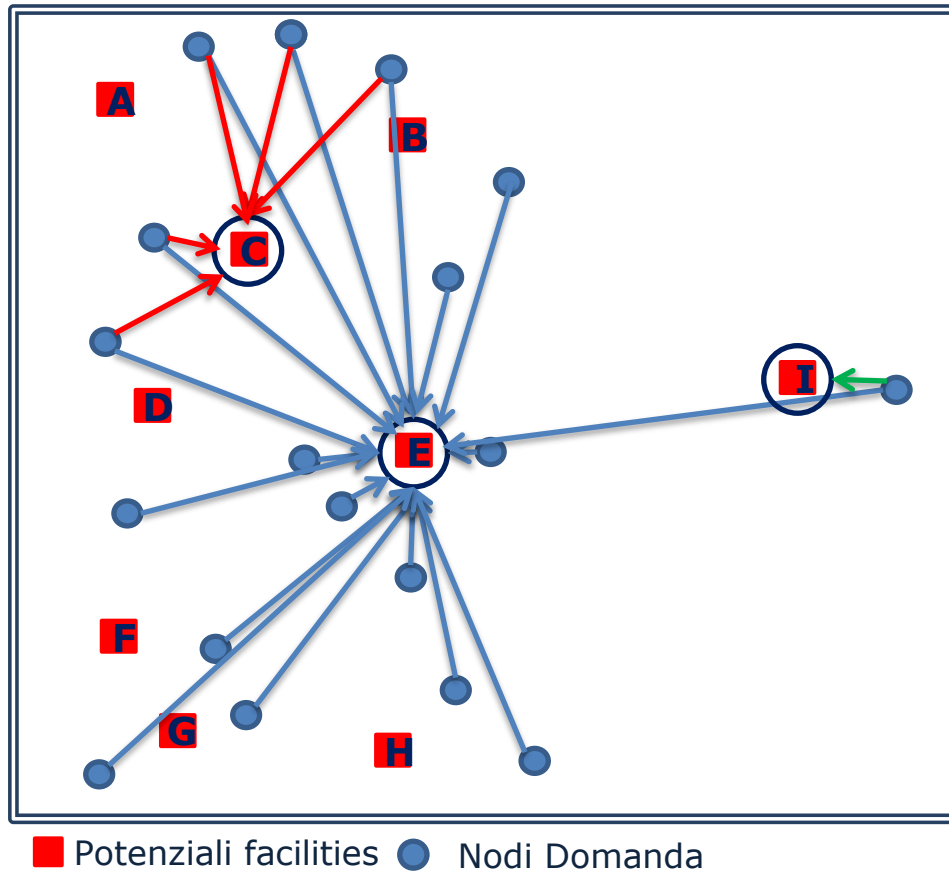
Il Simple Plant Location

Consiste nella ricerca del **numero e della posizione** delle facilities da attivare con l'obiettivo di minimizzare i costi totali (costi di afferenza + costi di localizzazione).

L'intera domanda di ciascun nodo afferisce a servizi attivi.

Problema di localizzazione puntuale, discreto, singolo decisore, orientato alla minimizzazione dei costi

II Simple Plant Location



Il Simple Plant Location

I insieme dei nodi domanda $|I| = n$;
 J insieme delle possibili localizzazioni $|J| = m$;
 c_{ij} costo di afferenza della domanda in i al servizio in j ;
 r_j costo di localizzazione del servizio nel nodo j ;

Variabili decisionali:

x_{ij} percentuale della domanda in i che afferisce al servizio in j ;
 y_j pari ad 1 se e solo se nel nodo j viene attivato un servizio.

$$z = \sum_{i \in I, j \in J} c_{ij} x_{ij} + \sum_{j \in J} r_j y_j \quad \text{Min!}$$

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I$$

Tutta la domanda in i viene soddisfatta da servizi

$$x_{ij} \leq y_j \quad x_{ij} - y_j \leq 0 \quad \forall i \in I, j \in J$$

La domanda in i può essere soddisfatta solo da servizi attivi

$$x_{ij} \geq 0 \quad \forall i \in I, j \in J$$

$$y_j = 0/1 \quad \forall j \in J$$

Numero variabili: $m + nm$

Numero vincoli: $n + nm$

Il Simple Plant Location

I	insieme dei nodi domanda $ I = n$;
J	insieme delle possibili localizzazioni $ J = m$;
c_{ij}	costo di afferenza della domanda in i al servizio in j;
r_j	costo di localizzazione del servizio nel nodo j;

Formulazione più compatta ma
meno efficiente
computazionalmente

Variabili decisionali:

x_{ij}	percentuale della domanda in i che afferisce coperto dal servizio in j;
y_j	pari ad 1 se e solo se nel nodo j viene attivato un servizio.

$$z = \sum_{i \in I, j \in J} c_{ij} x_{ij} + \sum_{j \in J} r_j y_j \quad \text{Min!}$$

$$\begin{aligned} \sum_{j \in J} x_{ij} &= 1 & \forall i \in I \\ \sum_{i \in I} x_{ij} &\leq n y_j & \forall j \in J \\ x_{ij} &\geq 0 & \forall i \in I, j \in J \\ y_j &= 0/1 & \forall j \in J \end{aligned}$$

Tutta la domanda in i viene soddisfatta da servizi

La domanda complessiva può essere soddisfatta solo da servizi attivi

Numero variabili: $m + nm$

Numero vincoli: $n + m$

Il Simple Plant Location

Esempio di calcolo della funzione obiettivo

Si consideri una soluzione $S=(2,4)$ con servizi aperti nei nodi 2 e 4
Ciascun nodo di domanda afferisce al nodo più vicino

Domanda

Servizi

↓

↓

	1	2	3	4	5	6
1	0	8	10	7	12	9
2	8	0	7	9	11	10
3	10	7	0	10	9	12
4	7	9	10	0	8	13
5	12	11	9	8	0	12
6	9	10	12	13	12	0

1

2

3

4

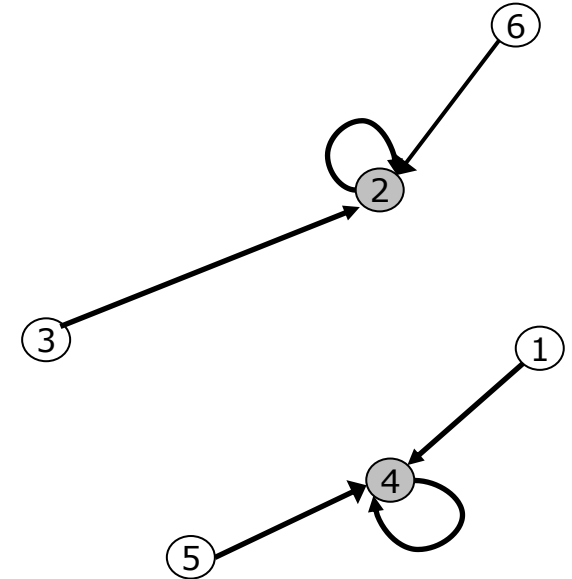
5

6

r_j	15	11	9	13	8	10
-------	----	----	---	----	---	----

Variabili x_{ij}

						
	1	2	3	4	5	6
1	0	0	0	1	0	0
2	0	1	0	0	0	0
3	0	1	0	0	0	0
4	0	0	0	1	0	0
5	0	0	0	1	0	0
6	0	1	0	0	0	0
	1	2	3	4	5	6
y_j	0	1	0	1	0	0



Rappresentazione delle afferenze

Matrice costi di afferenza e
vettore costi di localizzazione

$$z = (7+0+7+0+8+10) + (0+11+0+13+0+0) = 56$$

Costi di afferenza

Costi di localizzazione

Il Simple Plant Location

Il Simple Plant Location appartiene alla classe dei problemi **NP-hard**

Una soluzione ammissibile del problema può essere rappresentata da una stringa di valori binari corrispondente al vettore delle variabili y_j

	1	2	..	..	n-1	n
y_j	0	1	0	1	1	0

Il numero di soluzioni ammissibili è quindi pari al numero di totale di possibili stringhe binarie ovvero è pari a $2^n - 1$

(meno 1 perché la stringa identicamente nulla non è associata ad una soluzione ammissibile)

Per risolvere il problema è possibile ricorrere, oltre ad euristiche costruttive, a metodi di ricerca locale o a metaeuristiche basate su mosse che alterano in qualche maniera i valori della stringa rappresentativa della soluzione corrente

Il Simple Plant Location

Algoritmo migliorativo

In generale l'apertura di un nuovo servizio, produce un aumento dei costi di localizzazione (relativi al servizio che si apre) e una riduzione dei costi di afferenza

All'iterazione generica si considera (mossa) la possibilità di attivare un nuovo servizio. A ciascun potenziale nuovo servizio si associa un risparmio (saving) pari alla differenza tra il risparmio nei costi di afferenza e l'incremento del costo di localizzazione.

Si sceglie di aprire il servizio con il saving (positivo) maggiore. Se tutti i savings dovessero risultare negativi, l'algoritmo si arresta.

Il Simple Plant Location

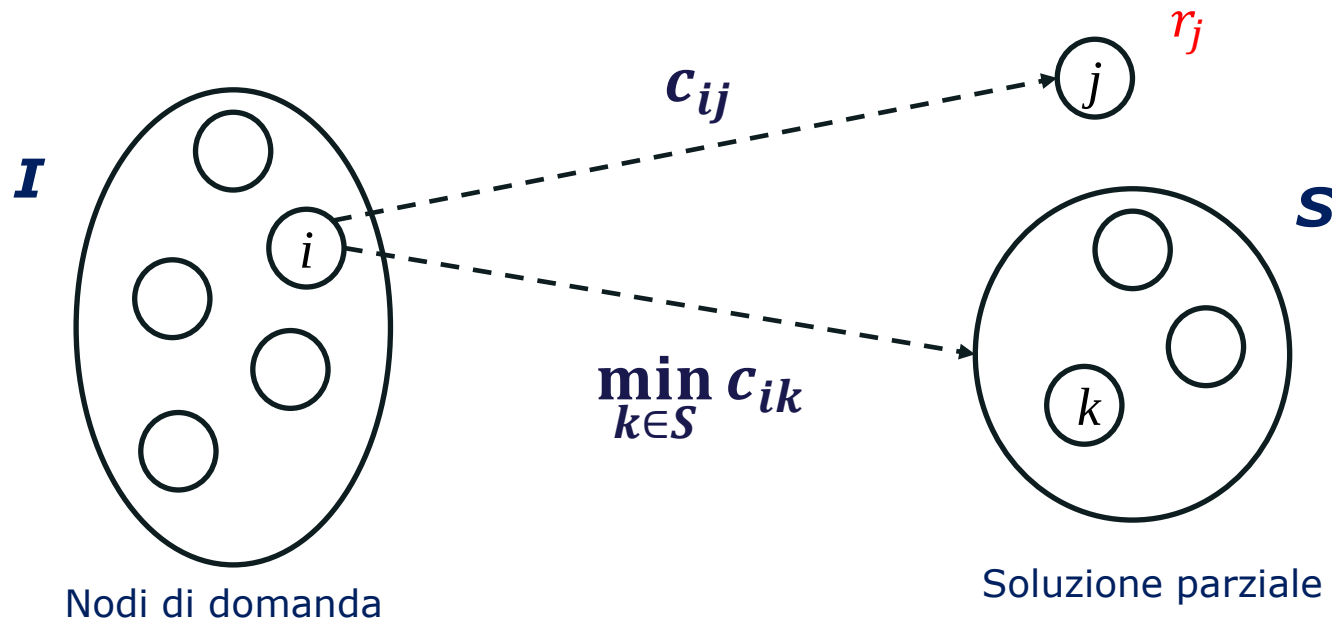
Algoritmo migliorativo

Calcolo dei savings

Per ogni $j \in S'$ si calcolano i savings $s_j \geq 0$

$$s_j = \sum_{i \in I} \max(\min_{k \in S} c_{ik} - c_{ij}, 0) - r_j$$

$\forall j: s_j \geq 0$ si pone $l(j) = +1$



$$c_{ij} < \min_{k \in S} c_{ik}$$



Riduzione costo di afferenza

$$\min_{k \in S} c_{ik} - c_{ij}$$

$$c_{ij} \geq \min_{k \in S} c_{ik}$$



Riduzione costo di afferenza

0

Il Simple Plant Location

Algoritmo migliorativo

Inizializzazione

Per ogni $j \in J$ si calcola

$$w_j = r_j + \sum_{i \in I} c_{ij}$$

Si apre il primo servizio in $j^* : w_{j^*} \leq w_j$.

Si pone $S = \{j^*\}$, $S' = J - \{j^*\}$, $l(j) = -1$ per ogni $j \in S'$; $z(S) = w_{j^*}$

Calcolo dei savings

Per ogni $j \in S'$ si calcolano i savings $s_j \geq 0$ si pone $l(j) = +1$

$$s_j = \sum_{i \in I} \max(\min_{k \in S} c_{ik} - c_{ij}, 0) - r_j$$

$\forall j: s_j \geq 0$ si pone $l(j) = +1$

Criterio d'arresto

Se $s_j \leq 0$ per ogni $j \in S'$, l'algoritmo si arresta; altrimenti, si individua $j^* : \max s_j > 0$ con $j \in S'$. Si pone $S = S \cup \{j^*\}$, $z(S) = z(S) - s_{j^*}$. Se S' è vuoto, l'algoritmo si arresta. Altrimenti si ritorna allo step 2.

Il Simple Plant Location

Esempio di applicazione algoritmo migliorativo

Inizializzazione

Per ogni $j \in J$ si calcola

$$w_j = r_j + \sum_{i \in I} c_{ij}$$

Si localizza il primo servizio in $j^* : w_{j^*} \leq w_j$. Si pone $S = \{j^*\}$, $S' = J - \{j^*\}$, $l(j) = -1$ per ogni $j \in S'$; $z(S) = w_{j^*}$

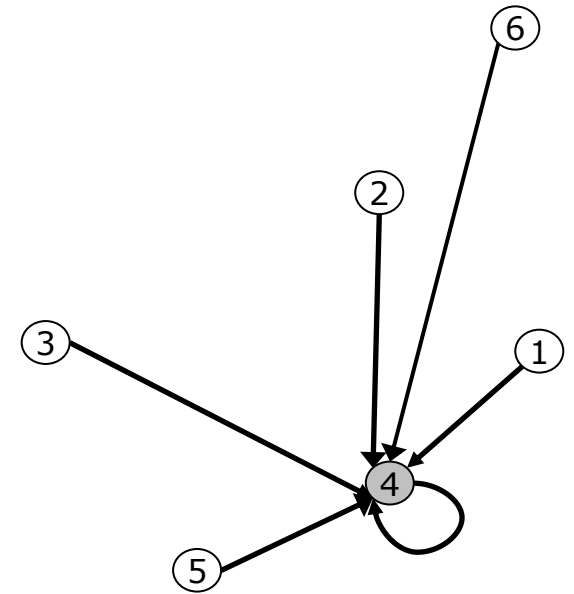
	1	2	3	4	5	6
1	0	5	7	3	4	9
2	5	0	10	6	8	5
3	7	10	0	4	7	8
4	3	6	4	0	2	9
5	4	8	7	2	0	6
6	9	5	8	9	6	0

	1	2	3	4	5	6
r_j	7	3	8	5	10	6

	1	2	3	4	5	6
w_j	35	38	44	29	37	43

$$S = \{4\}$$

	1	2	3	4	5	6
l_j	-1	-1	-1		-1	-1



Il Simple Plant Location

Esempio di applicazione algoritmo migliorativo

Calcolo dei savings

Per ogni $j \in S'$ si calcolano i savings $s_j \geq 0$ si pone $l(j) = +1$

$$s_j = \sum_{i \in I} \max(\min_{k \in S} c_{ik} - c_{ij}, 0) - r_j \quad \forall j: s_j \geq 0 \text{ si pone } l(j) = +1$$

	↓	↓	↓	↓	↓	
	1	2	3	4	5	6
1	0	5	7	3	4	9
2	5	0	10	6	8	5
3	7	10	0	4	7	8
4	3	6	4	0	2	9
5	4	8	7	2	0	6
6	9	5	8	9	6	0
	1	2	3	4	5	6
r_j	7	3	8	5	10	6

$$s_1 = (3+1+0+0+0+0) - 7 = -3$$

$$s_2 = (0+6+0+0+0+4) - 3 = +7$$

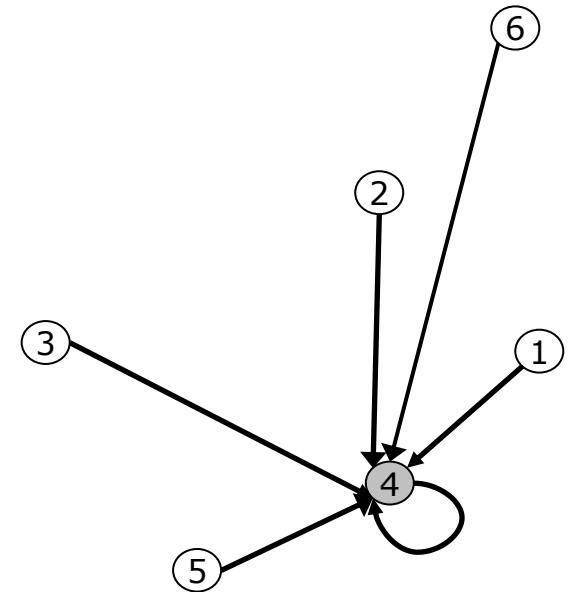
$$s_3 = (0+0+4+0+0+1) - 8 = -3$$

$$s_5 = (0+0+0+0+2+3) - 10 = -5$$

$$s_6 = (0+1+0+0+0+9) - 6 = +4$$

$$S = \{4\}$$

	1	2	3	4	5	6
l_j	-1	+1	-1		-1	+1



Il Simple Plant Location

Esempio di applicazione algoritmo migliorativo

Criterio d'arresto

Se $s_j \leq 0$ per ogni $j \in S'$, l'algoritmo si arresta; altrimenti, si individua j^* : $\max s_j > 0$ con $j \in S'$. Si pone $S = S \cup \{j^*\}$, $z(S) = z(S) - s_{j^*}$. Se S' è vuoto, l'algoritmo si arresta. Altrimenti si ritorna allo step 2.

	1	2	3	4	5	6
1	0	5	7	3	4	9
2	5	0	10	6	8	5
3	7	10	0	4	7	8
4	3	6	4	0	2	9
5	4	8	7	2	0	6
6	9	5	8	9	6	0

	1	2	3	4	5	6
r_j	7	3	8	5	10	6

$$s_1 = (3+1+0+0+0+0) - 7 = -3$$

$$s_2 = (0+6+0+0+0+4) - 3 = +7$$

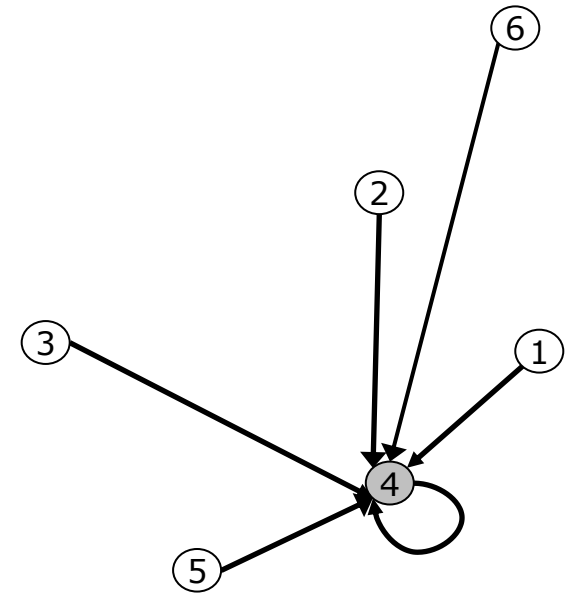
$$s_3 = (0+0+4+0+0+1) - 8 = -3$$

$$s_5 = (0+0+0+0+2+3) - 10 = -5$$

$$s_6 = (0+1+0+0+0+9) - 6 = +4$$

$$S = \{4\}$$

	1	2	3	4	5	6
l_j	-1	+1	-1		-1	+1



Il Simple Plant Location

Esempio di applicazione algoritmo migliorativo

Criterio d'arresto

Se $s_j \leq 0$ per ogni $j \in S'$, l'algoritmo si arresta; altrimenti, si individua j^* : $\max s_j > 0$ con $j \in S'$. Si pone $S = S \cup \{j^*\}$, $z(S) = z(S) - s_{j^*}$. Se S' è vuoto, l'algoritmo si arresta. Altrimenti si ritorna allo step 2.

	1	2	3	4	5	6
1	0	5	7	3	4	9
2	5	0	10	6	8	5
3	7	10	0	4	7	8
4	3	6	4	0	2	9
5	4	8	7	2	0	6
6	9	5	8	9	6	0
r_j	7	3	8	5	10	6

$$s_1 = (3+1+0+0+0+0) - 7 = -3$$

$$s_2 = (0+6+0+0+0+4) - 3 = +7$$

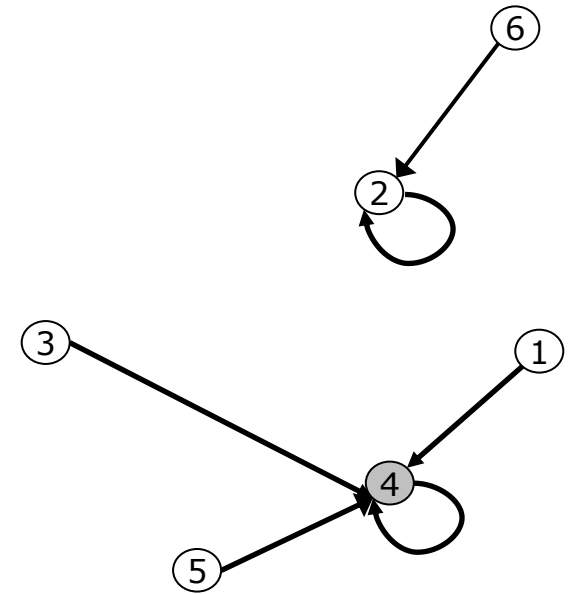
$$s_3 = (0+0+4+0+0+1) - 8 = -3$$

$$s_5 = (0+0+0+0+2+3) - 10 = -5$$

$$s_6 = (0+1+0+0+0+9) - 6 = +4$$

$$S = \{2, 4\}$$

	1	2	3	4	5	6
l_j	-1		-1		-1	+1



Il Simple Plant Location

Esempio di applicazione algoritmo migliorativo

Calcolo dei savings

Per ogni $j \in S'$ si calcolano i savings $s_j \geq 0$ si pone $l(j) = +1$

$$s_j = \sum_{i \in I} \max(\min_{k \in S} c_{ik} - c_{ij}, 0) - r_j \quad \forall j: s_j \geq 0 \text{ si pone } l(j) = +1$$

↓

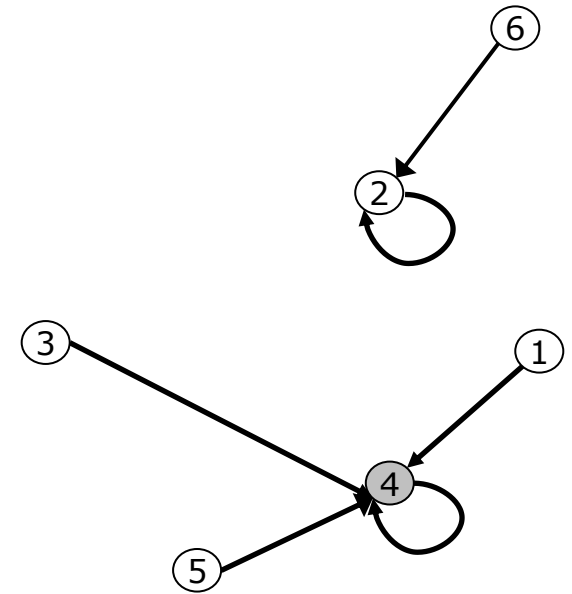
	1	2	3	4	5	6
1	0	5	7	3	4	9
2	5	0	10	6	8	5
3	7	10	0	4	7	8
4	3	6	4	0	2	9
5	4	8	7	2	0	6
6	9	5	8	9	6	0

	1	2	3	4	5	6
r_j	7	3	8	5	10	6

$$s_6 = (0 + 0 + 0 + 0 + 0 + 5) - 6 = -1$$

$$S = \{2, 4\}$$

	1	2	3	4	5	6
l_j	-1		-1		-1	+1



Il Simple Plant Location

Esempio di applicazione algoritmo migliorativo

Criterio d'arresto

Se $s_j \leq 0$ per ogni $j \in S'$, l'algoritmo si arresta; altrimenti, si individua j^* : $\max s_j > 0$ con $j \in S'$. Si pone $S = S \cup \{j^*\}$, $z(S) = z(S) - s_{j^*}$. Se S' è vuoto, l'algoritmo si arresta. Altrimenti si ritorna allo step 2.

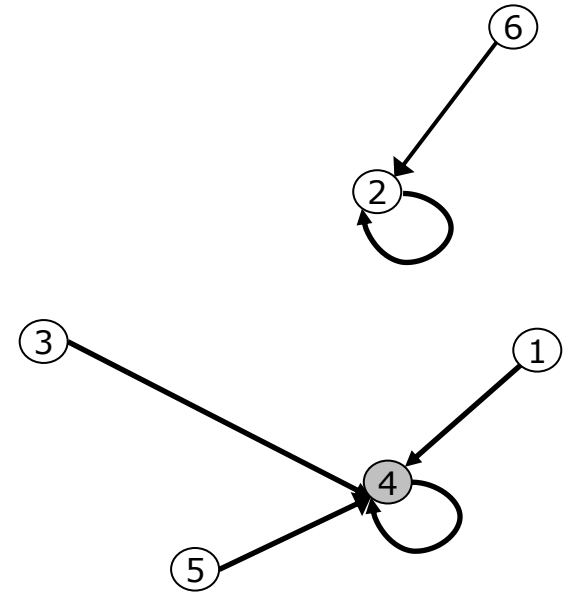
	1	2	3	4	5	6
1	0	5	7	3	4	9
2	5	0	10	6	8	5
3	7	10	0	4	7	8
4	3	6	4	0	2	9
5	4	8	7	2	0	6
6	9	5	8	9	6	0

	1	2	3	4	5	6
r_j	7	3	8	5	10	6

$$s_6 = (0+0+0+0+0+5) - 6 = -1$$

$$S = \{2, 4\}$$

	1	2	3	4	5	6
l_j	-1		-1		-1	+1



Il Simple Plant Location

Esempio di applicazione algoritmo migliorativo

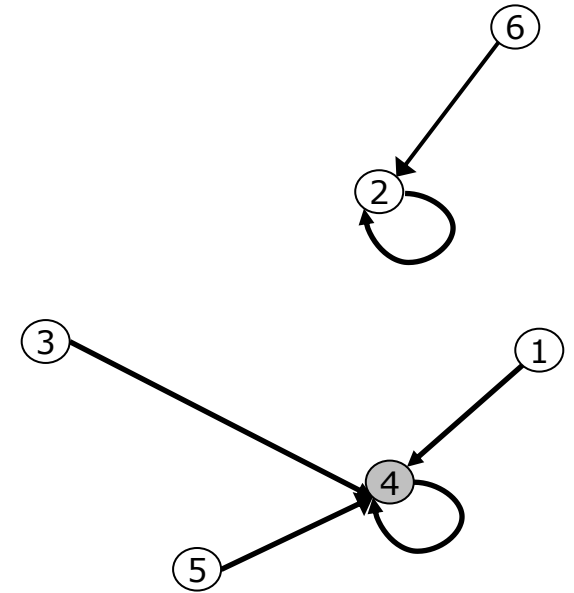
Criterio d'arresto

Se $s_j \leq 0$ per ogni $j \in S'$, l'algoritmo si arresta; altrimenti, si individua j^* : $\max s_j > 0$ con $j \in S'$. Si pone $S = S \cup \{j^*\}$, $z(S) = z(S) - s_{j^*}$. Se S' è vuoto, l'algoritmo si arresta. Altrimenti si ritorna allo step 2.

	1	2	3	4	5	6
1	0	5	7	3	4	9
2	5	0	10	6	8	5
3	7	10	0	4	7	8
4	3	6	4	0	2	9
5	4	8	7	2	0	6
6	9	5	8	9	6	0
r_j	7	3	8	5	10	6

Variabili x_{ij}

	1	2	3	4	5	6
1	0	0	0	1	0	0
2	0	1	0	0	0	0
3	0	0	0	1	0	0
4	0	0	0	1	0	0
5	0	0	0	1	0	0
6	0	1	0	0	0	0
y_j	0	1	0	1	0	0



$$z = (3+0+4+0+2+5) + (0+3+0+5+0+0) = 20$$

Costi di afferenza Costi di localizzazione

Capacitated Simple Plant Location

I	insieme dei nodi domanda $ I = n$;
J	insieme delle possibili localizzazioni $ J = m$;
d_i	domanda associata al punto i;
c_{ij}	costo di afferenza della domanda in i al servizio in j;
r_j	costo di localizzazione del servizio nel nodo j
C_j	capacità associata al servizio j;

Variabili decisionali:

x_{ij}	percentuale della domanda in i che afferisce coperto dal servizio in j;
y_j	pari ad 1 se e solo se nel nodo j viene attivato un servizio.

$$z = \sum_{j \in J, j \in J} c_{ij} x_{jj} + \sum_{j \in J} r_j y_j \quad \text{Min!}$$

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I$$

Tutta la domanda in i viene soddisfatta da servizi

$$\Rightarrow \sum_{i \in J} d_i x_{ij} \leq C_j y_j \quad \forall i \in I$$

La domanda complessiva afferente al servizio j (se attivo) non può eccedere la sua capacità

$$x_{ij} \geq 0 \quad \forall i \in I, j \in J$$

$$y_j = 0/1 \quad \forall j \in J$$

Numero variabili: $m + nm$

Numero vincoli: $n + m$

Capacitated Simple Plant Location

Nel modello senza vincoli di capacità **il meccanismo di afferenza della domanda ai servizi è di tipo tutto-niente** dal momento che, dovendo minimizzare i costi di afferenza, la domanda sarà soddisfatta dal servizio attivo con minor costo di afferenza

Nella versione con vincoli di capacità le variabili decisionali di afferenza potrebbero assumere **valori non necessariamente binari**, per la necessità di dover soddisfare i vincoli di capacità

Capacitated Simple Plant Location

Esempio di calcolo della funzione obiettivo

Si consideri una soluzione $S=(2,4)$ con servizi aperti nei nodi 2 e 4

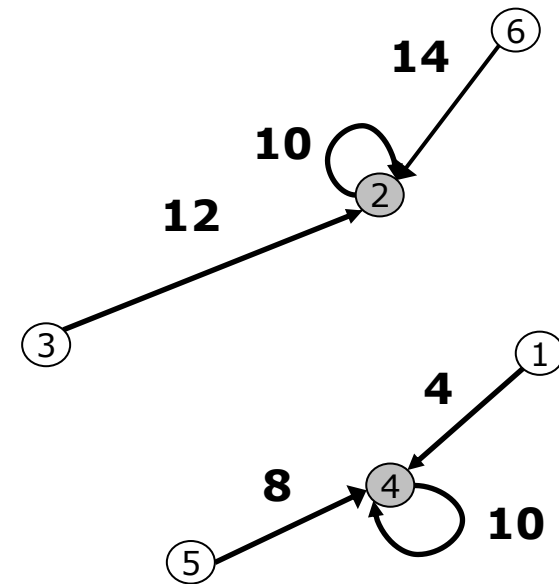
Domanda

Servizi

		Servizi					
		1	2	3	4	5	6
d_j	1	0	8	10	7	12	9
	2	8	0	7	9	11	10
	3	10	7	0	10	9	12
	4	7	9	10	0	8	13
	5	12	11	9	8	0	12
	6	9	10	12	13	12	0
		1	2	3	4	5	6
r_j		15	11	9	13	8	10

Variabili x_{ij}

		1	2	3	4	5	6
	1	0	0	0	1	0	0
	2	0	1	0	0	0	0
	3	0	1	0	0	0	0
	4	0	0	0	1	0	0
	5	0	0	0	1	0	0
	6	0	1	0	0	0	0
		1	2	3	4	5	6
y_j		0	1	0	1	0	0
		1	2	3	4	5	6
$\sum d_i x_{ij}$		0	36	0	22	0	0



Rappresentazione delle afferenze

Capacitated Simple Plant Location

Esempio di calcolo della funzione obiettivo

Si consideri una soluzione $S=(2,4)$ con servizi aperti nei nodi 2 e 4

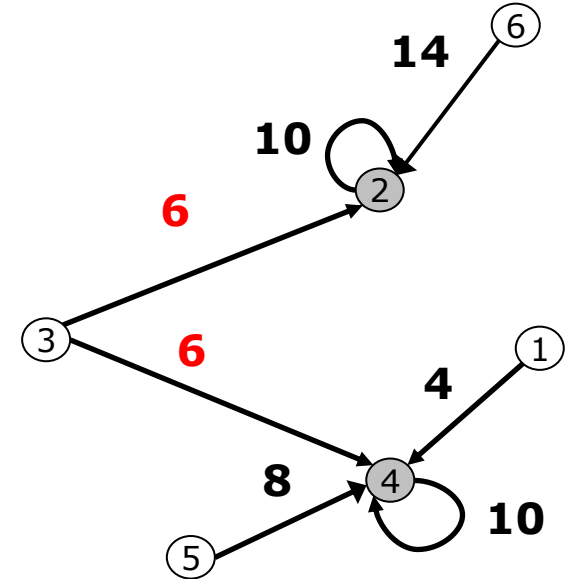
Domanda

Servizi

		Servizi					
		1	2	3	4	5	6
d_j							
4	1	0	8	10	7	12	9
10	2	8	0	7	9	11	10
12	3	10	7	0	10	9	12
10	4	7	9	10	0	8	13
8	5	12	11	9	8	0	12
14	6	9	10	12	13	12	0
		1	2	3	4	5	6
r_j		15	11	9	13	8	10

Variabili x_{ij}

		1	2	3	4	5	6
1		0	0	0	1	0	0
2		0	1	0	0	0	0
3		0	0,5	0	0,5	0	0
4		0	0	0	1	0	0
5		0	0	0	1	0	0
6		0	1	0	0	0	0
		1	2	3	4	5	6
y_j		0	1	0	1	0	0
		1	2	3	4	5	6
$\sum d_i x_{ij}$		0	30	0	28	0	0



Rappresentazione delle afferenze

p-mediana

Consiste nella ricerca **della posizione di un numero prefissato p di facilities** da attivare con l'obiettivo di minimizzare i costi totali (costi di afferenza + costi di localizzazione).

L'intera domanda di ciascun nodo afferisce a servizi attivi.

Nella versione più classica, si assume che i costi di localizzazione siano uguale per ciascuna facility.

Problema di localizzazione puntuale, discreto, singolo decisore, orientato alla minimizzazione dei costi

p-mediana

- numero di servizi da localizzare noto a priori e pari a p
- costi di localizzazione uguali per ciascun servizio pari a r

Queste due ipotesi consentono di trascurare i costi di localizzazione nella funzione obiettivo essendo essi costanti.

$$z = \sum_{i \in I, j \in J} c_{ij} x_{ij} + r \sum_{j \in J} y_j = \sum_{i \in I, j \in J} c_{ij} x_{ij} + rp$$

Inoltre bisogna aggiungere il vincolo sul numero prefissato di servizi da attivare:

$$\sum_{j \in J} y_j = p$$

p-mediana

I	insieme dei nodi domanda $ I = n$;
J	insieme delle possibili localizzazioni $ J = m$;
c_{ij}	costo di afferenza della domanda in i al servizio in j ;
r_j	costo di localizzazione del servizio nel nodo j ;

Variabili decisionali:

x_{ij}	percentuale della domanda in i che afferisce coperto dal servizio in j ;
y_j	pari ad 1 se e solo se nel nodo j viene attivato un servizio.

$$z = \sum_{j \in J, i \in I} c_{ij} x_{ij} + \sum_{j \in J} r_j y_j \quad \text{Min!}$$

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I$$

Tutta la domanda in i viene soddisfatta da servizi

$$x_{ij} - y_j \leq 0 \quad \forall i \in I, j \in J$$

La domanda in i può essere soddisfatta solo da servizi attivi

$$\sum_{j \in J} y_j = p$$

Il numero di servizi attivi deve essere pari a p

$$x_{ij} \geq 0 \quad \forall i \in I, j \in J$$

$$y_j = 0/1 \quad \forall j \in J$$

Numero variabili: $m + nm$

Numero vincoli: $n + nm$

p-mediana

Il problema della p-mediana è di tipo **NP-hard**

Una soluzione ammissibile del problema può essere rappresentata attraverso una stringa binaria caratterizzata da p valori unitari corrispondente al vettore delle variabili y_j

	1	2	..	..	n-1	n
y_j	0	1	0	1	1	0

$$\sum_{j \in J} y_j = p$$

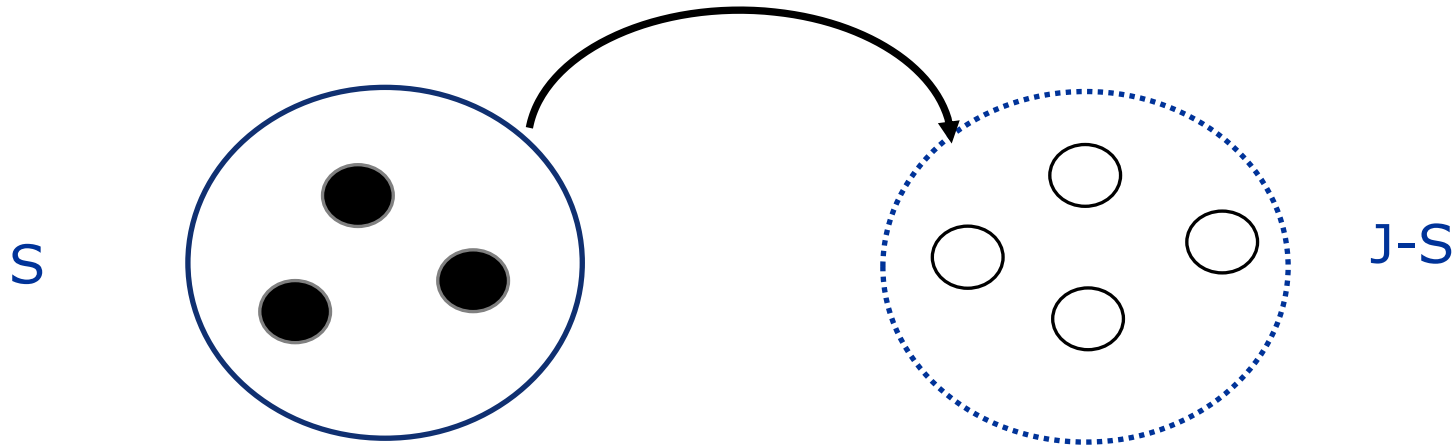
Il numero di soluzioni ammissibili è quindi pari al numero di possibili combinazioni di p elementi su n posizioni che è pari a

$$\binom{n}{p} = \frac{n!}{(n-p)! p!}$$

Per risolvere il problema è possibile ricorrere ad opportune euristiche

p-mediana

Algoritmo migliorativo di Teitz e Bart



Si scambia ogni $i \in S$ con ciascun $j \in S' = J - S$

Si calcola il saving $z_{ij} = z(S) - z(S - \{i\} \cup \{j\})$

Numero scambi: $p(n-p)$

p-mediana

Algoritmo migliorativo di Teitz e Bart

$$s_{ij} = z(S) - z(S - \{i\} \cup \{j\})$$

	1	2	3	4	5	6
1	0	8	10	7	12	9
2	8	0	7	9	11	10
3	10	7	0	10	9	12
4	7	9	10	0	8	13
5	12	11	9	8	0	12
6	9	10	12	13	12	0

↑ ↑

$$S=(1,2); z(S)=34$$

$$S=(4,2); z(S)=34-s_{14}=34-2=32$$

$$s_{13} = z(1,2) - z(3,2) = 34 - (8+9+9+10) = -2$$

$$s_{14} = z(1,2) - z(4,2) = 34 - (7+7+8+10) = +2 \quad \leftarrow$$

$$s_{15} = z(1,2) - z(5,2) = 34 - (8+7+8+10) = +1$$

$$s_{16} = z(1,2) - z(6,2) = 34 - (8+7+9+11) = -1$$

$$s_{23} = z(1,2) - z(1,3) = 34 - (7+7+9+9) = +2$$

$$s_{24} = z(1,2) - z(1,4) = 34 - (8+10+8+9) = -1$$

$$s_{25} = z(1,2) - z(1,5) = 34 - (8+9+7+9) = +1$$

$$s_{26} = z(1,2) - z(1,6) = 34 - (8+10+7+12) = -3$$

p-mediana

Algoritmo migliorativo di Teitz e Bart

$$s_{ij} = z(S) - z(S - \{i\} \cup \{j\})$$

	1	2	3	4	5	6
1	0	8	10	7	12	9
2	8	0	7	9	11	10
3	10	7	0	10	9	12
4	7	9	10	0	8	13
5	12	11	9	8	0	12
6	9	10	12	13	12	0



$S=(2,4)$; $z(S)=32$

$$s_{21} = z(2,4) - z(1,4) = 32 - (8 + 10 + 8 + 9) = -3$$

$$s_{23} = z(2,4) - z(3,4) = 32 - (7 + 7 + 9 + 12) = -3$$

$$s_{25} = z(2,4) - z(5,4) = 32 - (7 + 9 + 9 + 12) = -6$$

$$s_{26} = z(2,4) - z(6,4) = 32 - (7 + 9 + 12 + 8) = -4$$

$$s_{41} = z(2,4) - z(2,1) = 32 - (7 + 7 + 11 + 9) = -2$$

$$s_{43} = z(2,4) - z(2,3) = 32 - (8 + 9 + 9 + 10) = -4$$

$$s_{45} = z(2,4) - z(2,5) = 32 - (8 + 7 + 8 + 10) = -1$$

$$s_{46} = z(2,4) - z(2,6) = 32 - (8 + 7 + 8 + 11) = -2$$

p-mediana

Algoritmo migliorativo di Teitz e Bart

1. Inizializzazione

Si individui una prima soluzione S costituita da p nodi (la scelta può essere anche casuale); si calcola $z(S)$ e si pone $S' = J - S$.

2. Calcolo dei savings

Per ciascun nodo $i \in S'$ e $j \in S'$ si calcola il saving $s_{ij} = z(S) - z(S - \{i\} \cup \{j\})$. Si individua i^*, j^* tale che $s_{i^*j^*} = \max s_{ij}$.

3. Criterio di arresto

Se $s_{i^*j^*} \leq 0$ l'algoritmo si arresta; altrimenti si pone $S = S \cup \{j^*\} - \{i^*\}$, $S' = S' \cup \{i^*\} - \{j^*\}$, e si torna al passo 2.

Modellizzazione Sistemi logistici

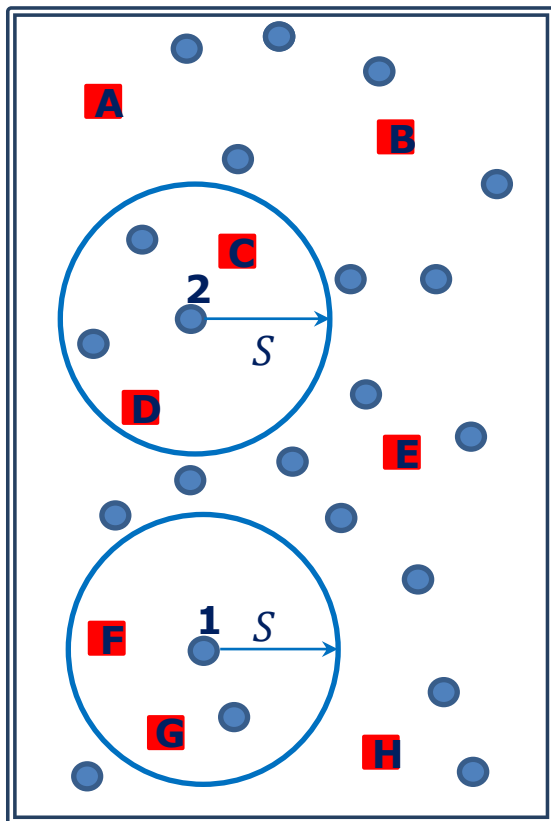
I modelli di Localizzazione

giuseppe.bruno@unina.it

Docente: Prof. Giuseppe Bruno

Modelli di copertura

Il concetto di copertura



■ Potenziali facilities ● Nodi Domanda

Si dice che un punto di domanda i è **coperto** da un servizio j se la distanza che li divide è minore o uguale ad un valore prefissato S , detto **distanza** di copertura.

L'insieme di copertura N_i del generico nodo i è costituito da tutte le potenziali localizzazioni in grado di coprire la sua domanda.

$$N_i = \{j \in J: d_{ij} \leq S\}$$

Es. $N_1 = \{F, G\}$

Modelli di copertura

La matrice di copertura

MATRICE DELLE DISTANZE

$$D = \{d_{ij}\}$$

$$D = \begin{pmatrix} d_{11} & \dots & d_{1j} & \dots & d_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ d_{i1} & \dots & d_{ij} & \dots & d_{im} \\ \dots & \dots & \dots & \dots & \dots \\ d_{n1} & \dots & d_{nj} & \dots & d_{nm} \end{pmatrix}$$

MATRICE DI COPERTURA

$$A = \{a_{ij}\}$$

$$A = \begin{pmatrix} a_{11} & \dots & a_{1j} & \dots & a_{1m} \\ \dots & \dots & \dots & \dots & \dots \\ a_{i1} & \dots & a_{ij} & \dots & a_{im} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{nj} & \dots & a_{nm} \end{pmatrix}$$

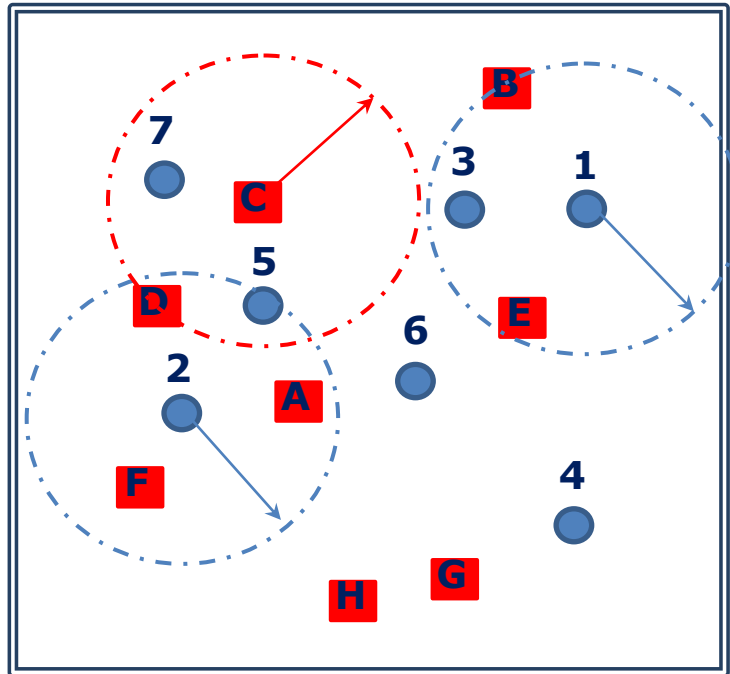
$$\text{con: } a_{ij} = \begin{cases} 1 & \text{se } d_{ij} \leq S \\ 0 & \text{se } d_{ij} > S \end{cases}$$

$$N_i = \{j \in J: a_{ij} = 1\}$$

Modelli di copertura

La matrice di copertura

$S = 2$



■ Potenziali facilities ● Nodi Domanda

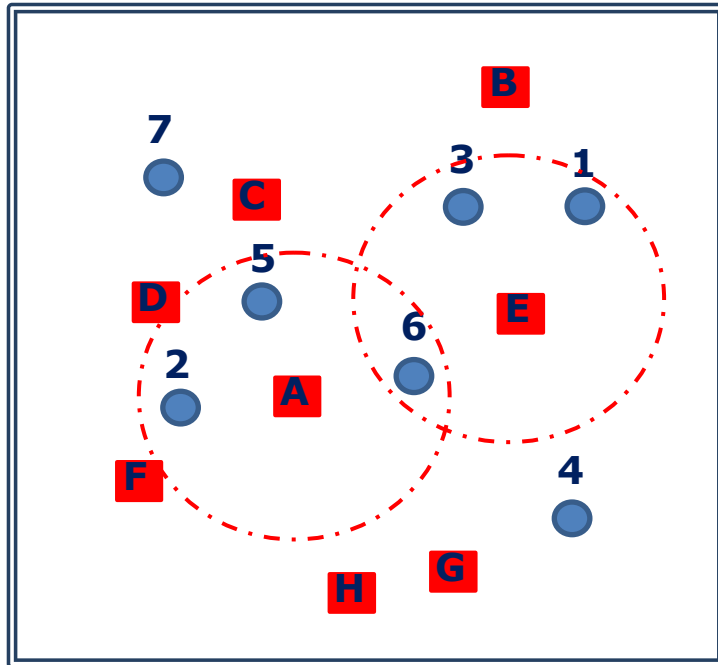
$D =$

	A	B	C	D	E	F	G	H
1	5	2	4.5	6	2	7	6	7
2	1.5	6	3	1.5	4.5	1	4	3.5
3	3	2	2.5	4	2	5	5	6
4	4	6	6	6.5	2.5	6	2	2.5
5	1.5	4	1.5	1.5	3	3	4.5	4
6	1.5	4	3	3	1.5	4	2.5	3
7	3	4	1	2	5	4	6.5	6

$A =$

	A	B	C	D	E	F	G	H	Tot. Riga
1	0	1	0	0	1	0	0	0	2
2	1	0	0	1	0	1	0	0	3
3	0	1	0	0	1	0	0	0	2
4	0	0	0	0	0	0	1	0	1
5	1	0	1	1	0	0	0	0	3
6	1	0	0	0	1	0	0	0	2
7	0	0	1	1	0	0	0	0	2
Tot. Colonna	3	2	2	2	3	1	1	0	

Modelli di copertura



■ Potenziali facilities ● Nodi Domanda

Set Covering Problem:

individuare il numero e la posizione di servizi da attivare in modo da coprire tutti la domanda e minimizzare il costo di localizzazione.

Maximal Covering Problem:

individuare la posizione da assegnare ad un prefissato numero di servizi in modo da massimizzare la domanda totale coperta.

	Numero Servizi	Copertura domanda
<i>Set Covering</i>	Non Fissato	Vincolata
<i>Maximal Covering</i>	Fissato	Non Vincolata

Il problema del Set Covering

Notazione:

I	insieme dei nodi domanda;
J	insieme delle possibili localizzazioni;
d_{ij}	distanza tra i nodi i e j ;
S	distanza di copertura;
a_{ij}	pari ad 1 se e solo se il nodo domanda i è coperto dal servizio in j ;
N_i	insieme di copertura del nodo domanda i ;
r_j	costo di localizzazione del servizio nel nodo j ;
y_j	pari ad 1 se e solo se nel nodo j viene localizzato un nuovo servizio.

Formulazione:

$$\begin{aligned} z &= \sum_{j \in J} r_j y_j \quad \min! \\ \sum_{j \in J} a_{ij} y_j &\geq 1 \quad \forall i \in I \\ y_j &= 0/1 \quad \forall j \in J \end{aligned} \quad \xrightarrow{\text{red arrow}} \quad \begin{aligned} \sum_{j \in N_i} y_j &\geq 1 \quad \forall i \in I \end{aligned}$$

N.b. Se tutti i costi di localizzazione sono uguali ($r_j = r \forall j \in J$) il modello punta a minimizzare il numero totale di servizi da attivare ($z = \sum_{j \in J} y_j$).

Il problema del Set Covering

$$A = \begin{matrix} & \begin{matrix} A & B & C & D & E & F & G & H \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$z = \sum_{j \in J} r_j y_j \quad \text{Min!}$$

$$\sum_{j \in N_i} y_j \geq 1 \quad \forall i \in I$$

$$y_j = 0/1 \quad \forall j \in J$$

$$z = r_A y_A + r_B y_B + r_C y_C + r_D y_D + r_E y_E + r_F y_F + r_G y_G + r_H y_H \quad \text{min!}$$

$$1 \rightarrow y_B + y_E \geq 1$$

$$2 \rightarrow y_A + y_D + y_F \geq 1$$

$$3 \rightarrow y_B + y_E \geq 1$$

$$4 \rightarrow y_G \geq 1$$

$$5 \rightarrow y_A + y_C + y_D \geq 1$$

$$6 \rightarrow y_A + y_E \geq 1$$

$$7 \rightarrow y_C + y_D \geq 1$$

$$y_A, y_B, y_C, y_D, y_E, y_F, y_G = 0/1$$

Il problema del Set Covering

Algoritmo costruttivo per la risoluzione

S insieme dei servizi già localizzati;
 S' insieme dei servizi candidati ad essere aperti;
 $C(S)$ insieme di punti coperti da S ($C(S) = \{i \in I: \exists j \in S: a_{ij} = 1\}$);
 $\underline{C}(S) = I - C(S)$ insieme di punti non ancora coperti;

Criterio di inserimento di un nuovo nodo in S

per ciascun nodo $j \in S'$, si calcola il rapporto tra:

- il costo di localizzazione associato al nuovo servizio aperto
- il numero di **ulteriori** nodi che il servizio riuscirebbe a coprire ($\sum_{i \in C(S)} a_{ij}$, $\forall j \in S'$);

$$\left(\frac{r_j}{\sum_{i \in C(S)} a_{ij}}, \forall j \in S' \right)$$

Si sceglie di inserire il nodo j^* cui è associato il minimo costo rapporto

Criterio di arresto

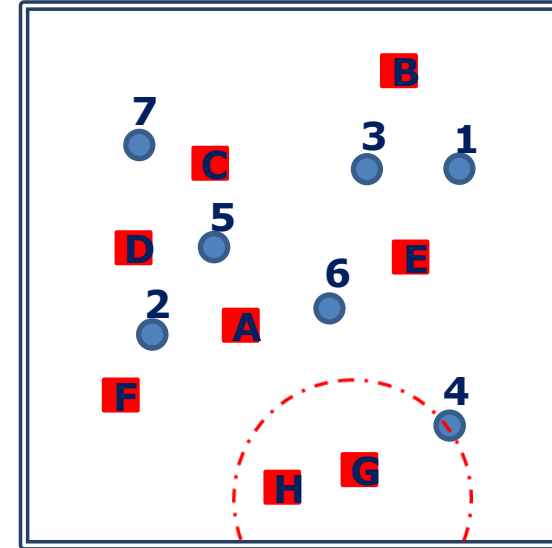
L 'algoritmo si arresta quando tutti i nodi sono stati coperti ($\underline{C}(S) = \emptyset$)

Il problema del Set Covering

Algoritmo costruttivo per la risoluzione

Inizializzazione: $S = \emptyset$ $S' = J$ $C(S) = \emptyset$ $\underline{C}(S) = I$ $z(S) = 0$

	A	B	C	D	E	F	G	H	
	9	12	5	9	11	5	2	7	r_j
1	0	1	0	0	1	0	0	0	
2	1	0	0	1	0	1	0	0	
3	0	1	0	0	1	0	0	0	
4	0	0	0	0	0	0	1	0	
5	1	0	1	1	0	0	0	0	
6	1	0	0	0	1	0	0	0	
7	0	0	1	1	0	0	0	0	
	3	2	2	3	3	1	1	0	$\sum_{i \in I} a_{ij}$
A	B	C	D	E	F	G	H		
3	6	2.5	3	3.7	5	2	∞	$\frac{r_j}{\sum_{i \in I} a_{ij}}$	



$$S = \{G\}$$

$$C(S) = \{4\}$$

$$S' = J - \{G\}$$

$$\underline{C}(S) = \{1, 2, 3, 5, 6, 7\}$$

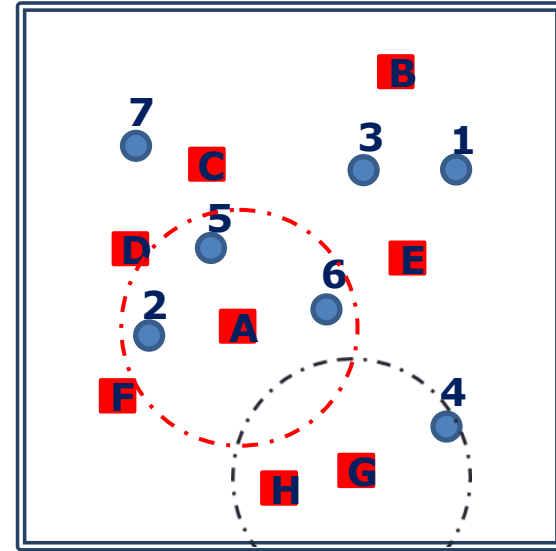
$$z(S) = r_G = 2$$

Il problema del Set Covering

Algoritmo costruttivo per la risoluzione

Iterazione 1: $S = \{G\}$ $S' = J - \{G\}$ $\mathcal{C}(S) = \{4\}$ $\underline{\mathcal{C}}(S) = \{1, 2, 3, 5, 6, 7\}$ $z(S) = 2$

	A	B	C	D	E	F	G	H	
	9	12	5	9	11	5	7		$\leftarrow r_j$
1	0	1	0	0	1	0	0	0	
2	1	0	0	1	0	1	0	0	
3	0	1	0	0	1	0	0	0	
4	0	0	0	0	0	0	1	0	
5	1	0	1	1	0	0	0	0	
6	1	0	0	0	1	0	0	0	
7	0	0	1	1	0	0	0	0	
	3	2	2	3	3	1	0		$\leftarrow \sum_{i \in \underline{\mathcal{C}}(S)} a_{ij}$
A	3	6	2.5	3	3.7	5	∞		$\leftarrow \frac{r_j}{\sum_{i \in \underline{\mathcal{C}}(S)} a_{ij}}$



$$S = \{A, G\}$$

$$\mathcal{C}(S) = \{4\} \cup \{2, 5, 6\}$$

$$S' = J - \{A, G\}$$

$$\underline{\mathcal{C}}(S) = \{1, 3, 7\}$$

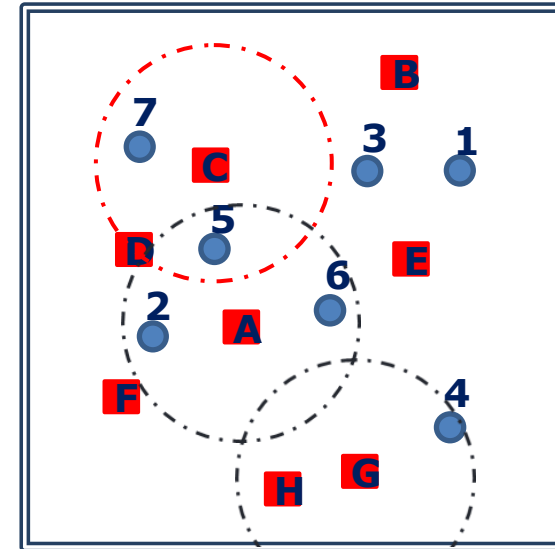
$$z(S) = 2 + 9 = 11$$

Il problema del Set Covering

Algoritmo costruttivo per la risoluzione

Iterazione 2: $S = \{A, G\}$ $S' = \{B, C, D, E, H\}$ $C(S) = \{2, 4, 5, 6\}$ $\underline{C}(S) = \{1, 3, 7\}$ $z(S) = 11$

	A	B	C	D	E	F	G	H	
	0	12	5	9	11	5	2	7	$\leftarrow r_j$
1	0	1	0	0	1	0	0	0	
2	0	0	0	1	0	1	0	0	
3	0	1	0	0	1	0	0	0	
4	0	0	0	0	0	0	1	0	
5	0	0	1	1	0	0	0	0	
6	0	0	0	0	1	0	0	0	
7	0	0	1	1	0	0	0	0	
		2	1	1	2	0		0	$\leftarrow \sum_{i \in \underline{C}(S)} a_{ij}$
	A	B	C	D	E	F	G	H	
	6	5	9	5.5	∞	∞	∞	∞	$\leftarrow \frac{r_j}{\sum_{i \in \underline{C}(S)} a_{ij}}$



$$S = \{A, C, G\}$$

$$C(S) = \{2, 4, 5, 6\} \cup \{7\}$$

$$S' = \{B, D, E, F, H\}$$

$$\underline{C}(S) = \{1, 3\}$$

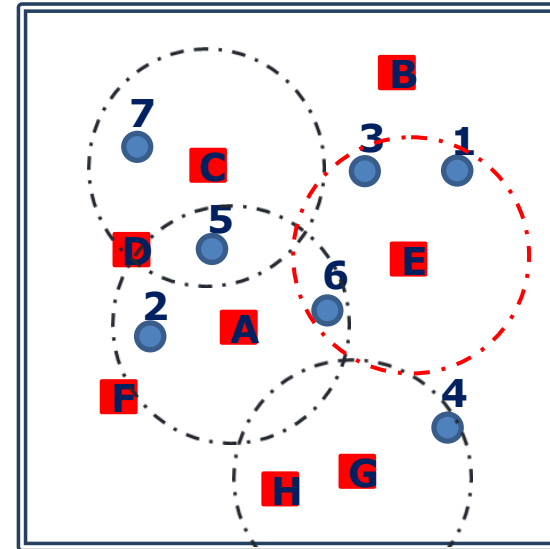
$$z(S) = 11 + 5 = 16$$

Il problema del Set Covering

Algoritmo costruttivo per la risoluzione

Iterazione 3: $S = \{A, G, C\}$ $S' = \{B, D, E, H\}$ $C(S) = \{2, 4, 5, 6, 7\}$ $\underline{C}(S) = \{1, 3\}$ $z(S) = 16$

	A	B	C	D	E	F	G	H	
	9	12	5	9	11	5	2	7	$\leftarrow r_j$
1	0	1	0	0	1	0	0	0	
2	1	0	0	1	0	1	0	0	
3	0	1	0	0	1	0	0	0	
4	0	0	0	0	0	0	1	0	
5	0	0	1	1	0	0	0	0	
6	0	0	0	0	1	0	0	0	
7	0	0	1	1	0	0	0	0	
		2		0	2	0		0	$\leftarrow \sum_{i \in \underline{C}(S)} a_{ij}$
	A	B	C	D	E	F	G	H	
		6		∞	5.5	∞		∞	$\leftarrow \frac{r_j}{\sum_{i \in \underline{C}(S)} a_{ij}}$



$S = \{A, C, E, G\}$

$C(S) = I$

$S' = \{B, D, F, H\}$

$\underline{C}(S) = \emptyset$

$z(S) =$
16+11=27

Maximal Covering

Consiste nella ricerca **della posizione di un numero prefissato p di facilities** da attivare con l'obiettivo di massimizzare la domanda totale coperta

A ciascun punto di domanda è associato un valore (domanda o popolazione)

Problema di localizzazione puntuale, discreto, singolo decisore, orientato alla copertura

Maximal Covering

Il problema è di tipo **NP-hard**

Una soluzione ammissibile del problema può essere rappresentata attraverso una stringa binaria caratterizzata da p valori unitari corrispondente al vettore delle variabili y_j

	1	2	..	..	n-1	n
y_j	0	1	0	1	1	0

$$\sum_{j \in J} y_j = p$$

Il numero di soluzioni ammissibili è quindi pari al numero di possibili combinazioni di p elementi su n posizioni che è pari a

$$\binom{n}{p} = \frac{n!}{(n-p)! p!}$$

Per risolvere il problema è possibile ricorrere ad opportune euristiche

Il problema del Maximal Covering

Notazione:

I	insieme dei nodi domanda;
J	insieme delle possibili localizzazioni;
a_{ij}	etichetta binaria pari ad 1 se e solo se il nodo domanda i è coperto dal servizio in j ;
N_i	insieme di copertura del nodo domanda i ;
d_i	domanda generata dal nodo i ;
p	numero di servizi da attivare;

Variabili decisionali:

y_j	variabile binaria pari ad 1 se e solo se nel nodo j viene localizzato un nuovo servizio.
x_i	variabile binaria pari ad 1 se e solo se il nodo i è coperto da un servizio.

Formulazione:

$$z = \sum_{i \in I} d_i x_i \quad \max!$$

$$\boxed{\sum_{j \in J} a_{ij} y_j \geq x_i \quad \forall i \in I} \longrightarrow \sum_{j \in N_i} y_j \geq x_i \quad \forall i \in I$$

$$\sum_{j \in J} y_j = p$$

$$x_i = 0/1 \quad \forall i \in I$$

$$y_j = 0/1 \quad \forall j \in J$$

Il problema del Maximal Covering

$$\begin{array}{c}
 \mathbf{d} = \{d_i\} \\
 \begin{array}{c} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \\ \mathbf{6} \\ \mathbf{7} \end{array}
 \end{array}
 \begin{pmatrix} 1 \\ 4 \\ 5 \\ 8 \\ 3 \\ 7 \\ 2 \end{pmatrix}
 \begin{array}{c}
 \mathbf{A} = \{a_{ij}\} \\
 \begin{array}{c} \mathbf{A} \quad \mathbf{B} \quad \mathbf{C} \quad \mathbf{D} \quad \mathbf{E} \quad \mathbf{F} \quad \mathbf{G} \quad \mathbf{H} \end{array} \\
 \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}
 \end{array}
 \end{array}
 \begin{array}{c}
 \mathbf{d}^T \cdot \mathbf{A} \\
 \begin{array}{c} \mathbf{A} \quad \mathbf{B} \quad \mathbf{C} \quad \mathbf{D} \quad \mathbf{E} \quad \mathbf{F} \quad \mathbf{G} \quad \mathbf{H} \end{array} \\
 \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 4 & 0 & 4 & 0 & 0 \\ 0 & 5 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\ 3 & 0 & 3 & 3 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 \end{pmatrix}
 \begin{array}{c} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \\ \mathbf{6} \\ \mathbf{7} \end{array}
 \end{array}
 \end{array}$$

$$\sum_{i \in I} a_{ij} d_i \quad \mathbf{14} \quad \mathbf{6} \quad \mathbf{5} \quad \mathbf{9} \quad \mathbf{13} \quad \mathbf{4} \quad \mathbf{8} \quad \mathbf{0}$$

Il problema del Maximal Covering

Algoritmo costruttivo per la risoluzione

S insieme dei servizi già localizzati;
 S' insieme dei servizi candidati ad essere aperti;
 $C(S)$ insieme di punti coperti da S ($C(S) = \{i \in I: \exists j \in S: a_{ij} = 1\}$);
 $\underline{C}(S) = I - C(S)$ insieme di punti non ancora coperti;

Criterio di inserimento di un nuovo nodo in S

per ciascun nodo $j \in S'$, si calcola:

➤ la domanda **ulteriore** che il servizio riuscirebbe a coprire

$$(\sum_{i \in \underline{C}(S)} d_i a_{ij}, \quad \forall j \in S')$$

Si sceglie di inserire il nodo j^* cui è associato il massimo valore

Criterio di arresto

L 'algoritmo si arresta quando si sono aperti p servizi

Il problema del Maximal Covering

Algoritmo costruttivo per la risoluzione ($p=3$)

Inizializzazione: $S = \emptyset$ $S' = J$ $C(S) = \emptyset$ $\underline{C}(S) = I$ $z(S) = 0$

$d = \{d_i\}$ $A = \{a_{ij}\}$

		A	B	C	D	E	F	G	H
1	(1)	0	1	0	0	1	0	0	0
2	4	1	0	0	1	0	1	0	0
3	5	0	1	0	0	1	0	0	0
4	8	0	0	0	0	0	0	1	0
5	3	1	0	1	1	0	0	0	0
6	7	1	0	0	0	0	0	0	0
7	2	0	0	1	1	0	0	0	0

$d^T \cdot A$

	A	B	C	D	E	F	G	H	
	0	1	0	0	1	0	0	0	1
	4	0	0	4	0	4	0	0	2
	0	5	0	0	5	0	0	0	3
	0	0	0	0	0	0	8	0	4
	3	0	3	3	0	0	0	0	5
	7	0	0	0	7	0	0	0	6
	0	0	2	2	0	0	0	0	7
$\sum_{i \in I} a_{ij} d_i$	14	6	5	9	13	4	8	0	

$$S = \{A\} \quad C(S) = \{2, 5, 6\}$$

$$S' = J - \{A\} \quad \underline{C}(S) = \{1, 3, 4, 7\}$$

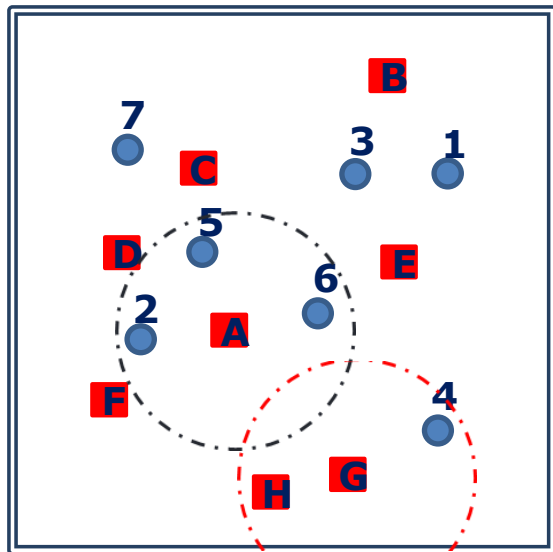
$$z(S) = \sum_{i \in I} a_{iA} d_A = 14$$

$$\sum_{j \in J} y_j = 1 < p$$

Il problema del Maximal Covering

Algoritmo costruttivo per la risoluzione ($p=3$)

Iterazione 1: $S = \{A\}$ $S' = J - \{A\}$ $C(S) = \{2, 5, 6\}$ $\underline{C}(S) = \{1, 3, 4, 7\}$ $z(S) = 14$



$d^T \cdot A$								
	A	B	C	D	E	F	G	H
1	0	1	0	0	1	0	0	0
2	0	0	0	1	0	1	0	0
3	0	5	0	0	5	0	0	0
4	0	0	0	0	0	0	8	0
5	0	0	3	3	0	0	0	0
6	0	0	0	0	7	0	0	0
7	0	0	2	2	0	0	0	0
	6	2	2	6	0	8	0	0

$$\sum_{i \in \underline{C}(S)} a_{ij} d_i$$

$$S = \{A, G\} \quad C(S) = \{2, 4, 5, 6\}$$

$$S' = J - \{A, G\} \quad \underline{C}(S) = \{1, 3, 7\}$$

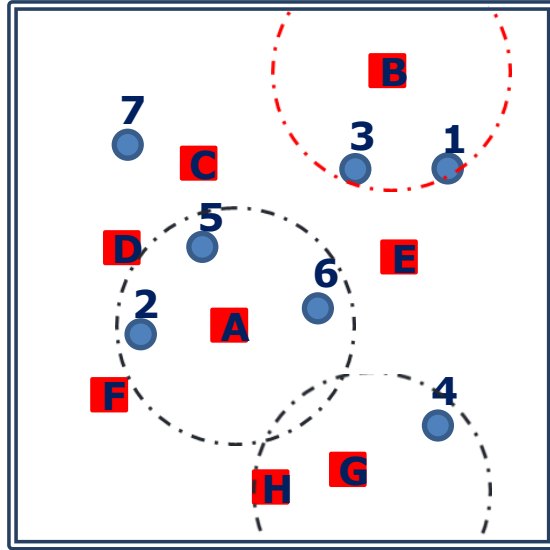
$$z(S) = 14 + \sum_{i \in \underline{C}(S)} a_{iG} d_G = 22$$

$$\sum_{j \in J} y_j = 2 < p$$

Il problema del Maximal Covering

Algoritmo costruttivo per la risoluzione ($p=3$)

Iterazione 2: $S = \{A, G\}$ $S' = J - \{A, G\}$ $C(S) = \{2, 4, 5, 6\}$ $\underline{C}(S) = \{1, 3, 7\}$ $z(S) = 22$



$$\sum_{i \in \underline{C}(S)} a_{ij} d_i$$

$d^T \cdot A$							
	A	B	C	D	E	F	H
1	0	1	0	0	1	0	0
2	0	0	0	1	0	1	0
3	0	5	0	0	5	0	0
4	0	0	0	0	0	0	0
5	0	0	3	3	0	0	0
6	0	0	0	0	7	0	0
7	0	0	2	2	0	0	0

$$S = \{A, G, B\} \quad C(S) = \{1, 2, 3, 4, 5, 6\}$$

$$S' = \{C, D, E, F\} \quad \underline{C}(S) = \{7\}$$

$$z(S) = 22 + \sum_{i \in \underline{C}(S)} a_{iB} d_B = 28$$

$$\sum_{j \in J} y_j = 3 = p \rightarrow \text{STOP}$$