

Controllo dei sistemi energetici

Ottimizzazione statica e dinamica
Parte seconda

Ing. Alessandro Pisano
apisano@unica.it

Ora discutiamo una differente applicazione che coinvolge la **gestione ottima di un sistema di storage a batteria**

Torneremo successivamente al problema del “dispacciamento economico” ottimale includendo nella formulazione del problema alcune generalizzazioni (costo di produzione che dipenda quadraticamente dalla Potenza, e perdite di trasmissione) che faranno sì che da un problema LP si passi ad un problema più complesso con vincoli e/o funzione obiettivo quadratici nelle variabili di decisione

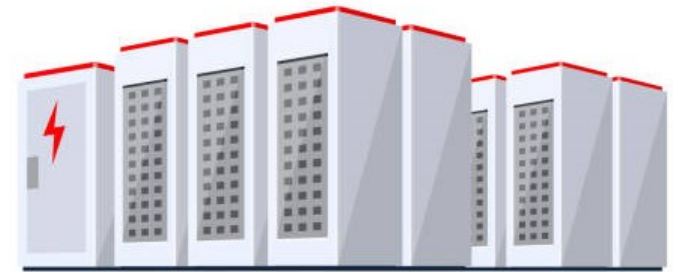
Esempio Gestione ottimale di una batteria

Consideriamo un edificio pubblico o industriale che assorbe dalla rete un carico prefissato $L(t)$ con una tariffazione differente sulla base della particolare fascia oraria.



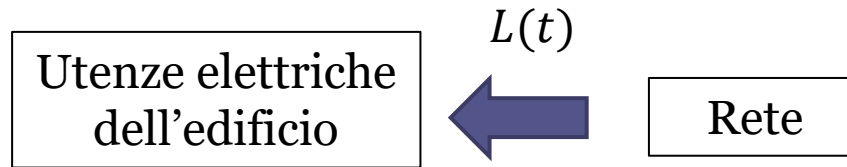
La determinazione del profilo ottimale per le fasi di carica e scarica della batteria può essere formulato come un problema di ottimizzazione dinamica vincolato.

L'impiego di un sistema di storage a batteria può consentire significativi risparmi sui costi energetici



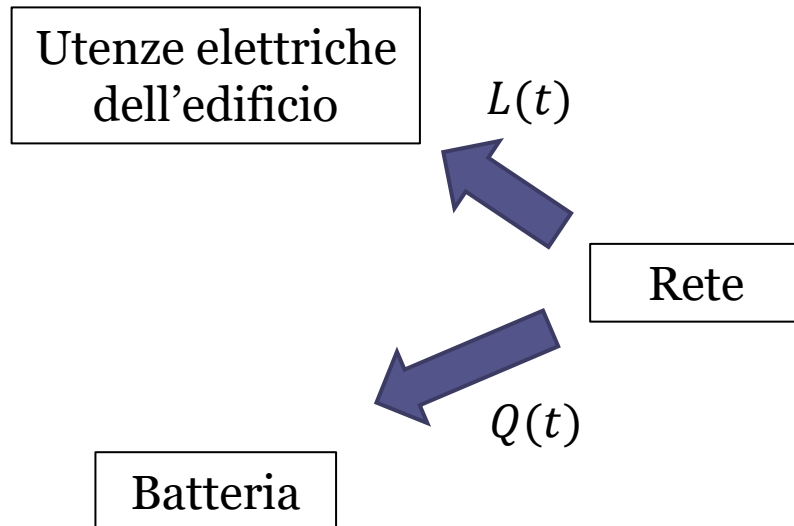
BATTERY  STORAGE

Senza batteria

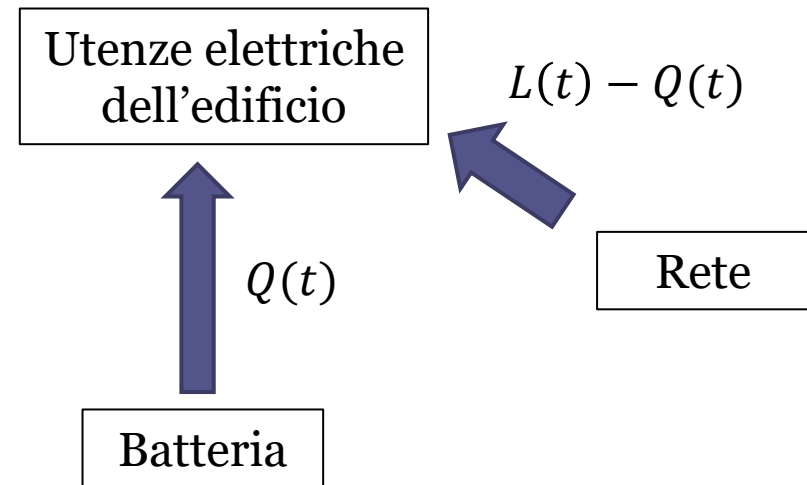


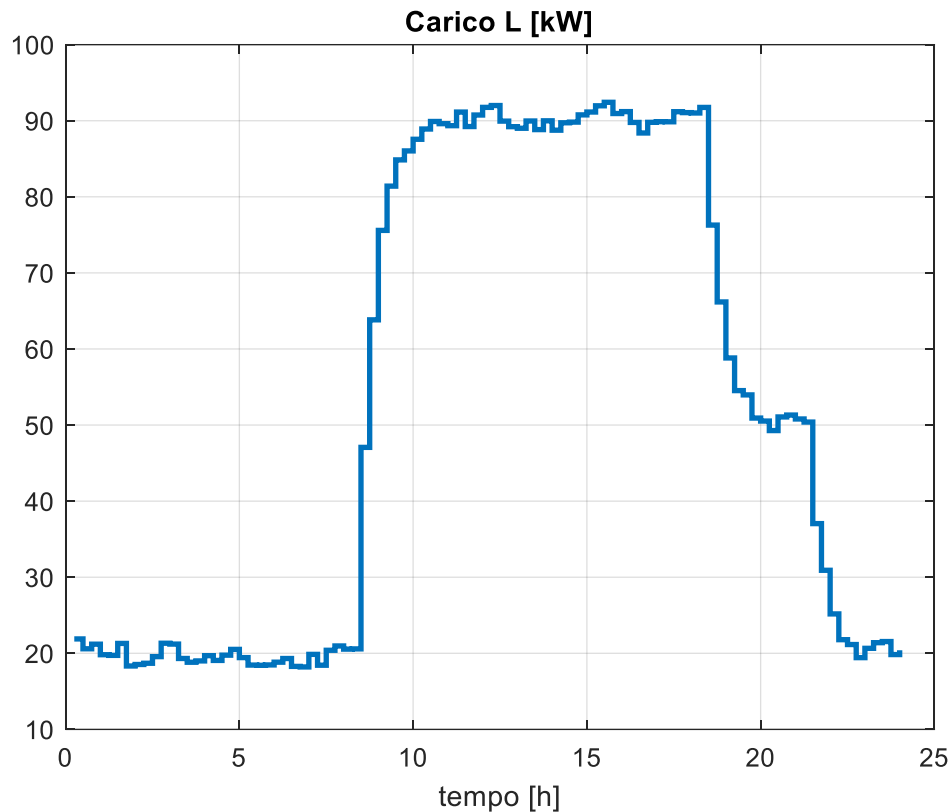
Con batteria

Fase di carica



Fase di scarica



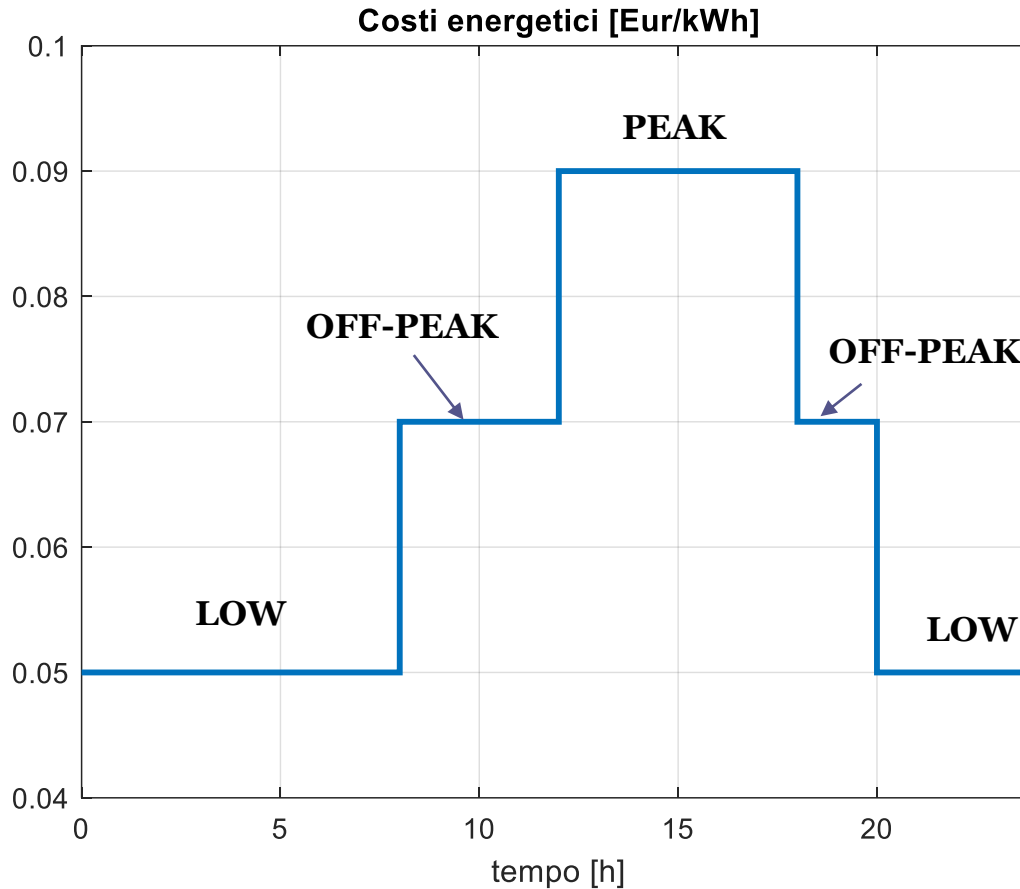


Profilo di carico dell'edificio
sulle 24 ore

Slot temporale: **15 minuti**

Inseriamo tali valori di carico in un vettore **L** avente 96 (24 x 4) elementi.

$$L = [L_1 \ L_2 \ L_3 \ \dots \ L_{95} \ L_{96}]$$



Costi energetici

Tariffa trioraria

Fasce «LOW», «OFF-PEAK», e «PEAK»

Inseriamo i costi energetici per kWh corrispondenti a ciascuno dei 96 slot temporali in un vettore **C** avente 96 elementi.

$$C = [C_1 \ C_2 \ C_3 \ \dots \ C_{95} \ C_{96}]$$



BATTERY ⚡ STORAGE

L'energia $S(t)$ immagazzinata nella batteria (State of Charge, [kWh]) varia secondo la seguente equazione differenziale

$$\dot{S}(t) = Q(t) \quad (1)$$

in cui $Q(t) > 0$ nella fase di carica e $Q(t) < 0$ nella fase di scarica della batteria.

Dalla (1):

$$S(t) = S(t_i) + \int_{t_i}^t Q(\tau) d\tau$$

che ci interessa
valutare per:

$$t_i = (k - 1)T_s$$

$$t = k T_s$$

Discretizzazione esatta (sotto l'ipotesi che la potenza prelevata/immessa sia costante e pari a $Q(k)$ all'interno del k -esimo periodo di campionamento di 15 minuti):

$$S(k) = S(k - 1) + T_s Q(k)$$

$$T_s = 0.25 \text{ [h]}$$

$$k = 1, 2, \dots, 96$$

$$S(0) = S_0$$

Formalizziamo il problema di ottimizzazione in termini di minimizzazione dei costi energetici

La potenza prelevata dalla rete al generico istante t è pari a $L(t) + Q(t)$

Nel generico slot temporale (la cui durata è di 15 minuti, cioè $T_s = 0.25 h$) il costo sostenuto è pari a

$$T_s[L(k) + Q(k)] C(k)$$

Il costo totale è pertanto

$$J = \sum_{k=1}^{96} T_s[L(k) + Q(k)] C(k) = T_s[L + Q]^T C \quad \text{Funzione obiettivo}$$

NB. La potenza assorbita dalla batteria ($Q > 0$) o ceduta dalla stessa ($Q < 0$) in corrispondenza di ciascuno dei 96 slot temporali è inserita in un vettore Q , di 96 elementi, che costituisce la variabile di cui si deve determinare il profilo ottimale

$$-Q_{min} \leq Q(k) \leq Q_{Max}$$

$$Q_{min} = 50 \text{ kW}$$

$$Q_{Max} = 50 \text{ kW}$$

Vincoli sulla rapidità di carica/scarica della batteria

$$L(k) + Q(k) \geq 0$$

La potenza prelevata istantaneamente dalla rete è non negativa

$$S(k) = S(k-1) + T_s Q(k)$$

$$S(0) = S_0 \quad S_0 = 0 \text{ kWh}$$

Dinamica dello state of charge della batteria

$$S_{min} \leq S(k) \leq S_{Max}$$

$$S_{min} = 0 \text{ kWh}$$

$$S_{max} = 390 \text{ kWh}$$

Vincoli sulla capacità energetica della batteria

$$0.95 S(0) \leq S(96) \leq 1.05 S(0)$$

Ripristino a fine giornata di uno stato di carica della batteria che differisca dallo stato iniziale non più del 5%

$$k = 1, 2, \dots, 96$$

Funzione obiettivo

$$J = \sum_{k=1}^{96} T_s [L(k) + Q(k)] C(k) = T_s [L + Q]^T C$$

Vincoli:

$$-Q_{min} \leq Q(k) \leq Q_{Max}$$

$$L(k) + Q(k) \geq 0$$

$$S(k) = S(k - 1) + T_s Q(k)$$

$$S(0) = S_0$$

$$S_{min} \leq S(k) \leq S_{Max}$$

$$0.95 S(0) \leq S(96) \leq 1.05 S(0)$$

$$k = 1, 2, \dots, 96$$

Programmazione dinamica:

all'interno dei vincoli compaiono delle variabili addizionali $S(k)$ governate da un sistema dinamico a tempo discreto del quale le variabili decisionali costituiscono gli ingressi.

Codice Matlab

```
S0=0;
Smin=0;
Smax=390; % [kWh]
Qmax=50; % [kW]
Ts=0.25;
N=96;
```

File: BatteryManagementBuilding.m

```
Q = optimvar('Q',N,1,'LowerBound',-Qmax,'UpperBound',Qmax);
S = optimvar('S',N,1,'LowerBound',Smin,'UpperBound',Smax);
```

Definizione delle variabili di ottimizzazione (funzione `optimvar`)
con i relativi valori minimi e massimi

```
obj=Ts*((L+Q) '*C);
```

Funzione obiettivo

$$J = T_s[L + Q]^T C$$

Dinamica dello state of charge della batteria

```
zcons1 = S(1) == S0 + Ts * Q(1);
zcons2 = S(2:N) == S(1:N-1) + Ts * Q(2:N);
```

$$S(1) = S_0 + T_s Q(1)$$

$$S(k) = S(k-1) + T_s Q(k), \quad 2 \leq k \leq N$$

```
LplusQcons = L+Q >= 0;
```

$$L(k) + Q(k) \geq 0$$

```
ZNconslow = 0.95 * S0 <= S(N);
ZNconshigh = S(N) <= 1.05 * S0;
```

$$0.95S(0) \leq S(96) \leq 1.05S(0)$$

```
optbattery=optimproblem;
```

Definizione problema di ottimizzazione

```
optbattery.Objective = obj;
```

Definizione funzione obiettivo

```
optbattery.Constraints.zcons1 = zcons1;
optbattery.Constraints.zcons2 = zcons2;
optbattery.Constraints.LplusQcons = LplusQcons;
optbattery.Constraints.ZNconslow = ZNconslow;
optbattery.Constraints.ZNconshigh = ZNconshigh;
```

Definizione vincoli

```
options = optimoptions('linprog','Display','final');
```

Scelta del solutore

```
[optbatterySOL mincost]= solve(optbattery,'options',options);
```

```
disp('spese giornaliere: [Eur]')
mincost
```

Analisi della soluzione. Calcolo del risparmio percentuale dovuto all'impiego ottimale della batteria

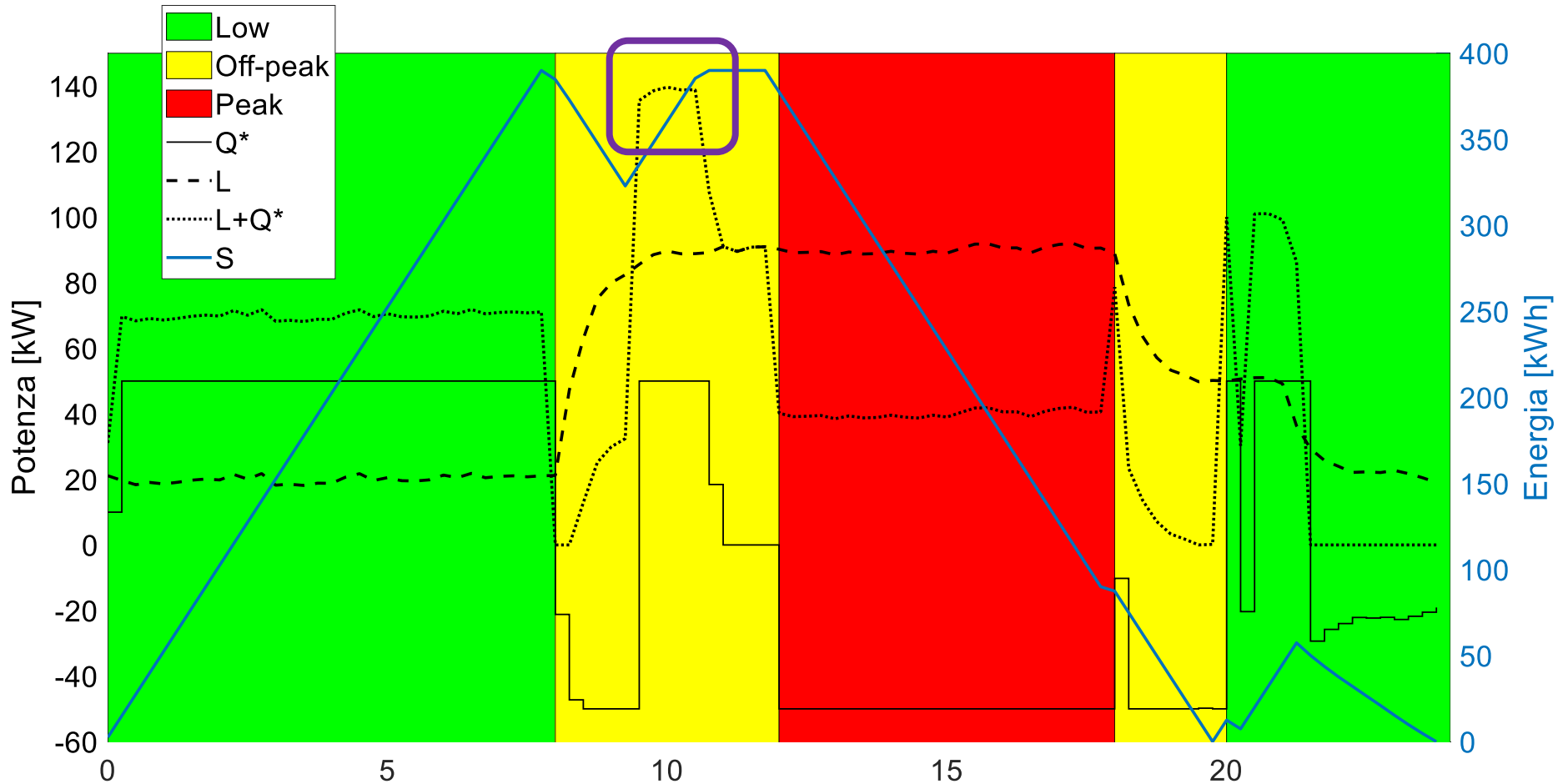
```
Qstar=optbatterySOL.Q;
```

```
disp('spese giornaliere senza batteria: [Eur]')
spesenobatt=0.25*L'*C
```

```
risparmio_percentuale=(spesenobatt-mincost)/spesenobatt*100
```

Risultati

Picco di circa 130 kW
nella potenza assorbita
dalla rete



```
Solving problem using linprog.
```

```
Optimal solution found.
```

```
spese giornaliera: [Eur]
```

```
mincost =
```

```
79.6329
```

```
spese giornaliera senza batteria: [Eur]
```

```
spesenobatt =
```

```
93.4329
```

```
risparmio_percentuale =
```

```
14.7700
```

L'impiego ottimizzato della batteria ha consentito un risparmio su base giornaliera del **15%** circa

File: BatteryManagementBuilding.m

Peak-shaving (livellamento dei picchi)

Nei problemi di gestione ottima dei profili di carica e scarica di batterie connesse alla rete (o più in generale nella pianificazione del prelievo di potenza dalla rete da parte di carichi arbitrari), diventa importante non solo «ottimizzare» il prelievo di energia dalla rete in modo che questo avvenga in prevalenza nelle fasce orarie più convenienti in termini di «energy rate», ma anche «livellare» il profilo di potenza assorbito dalla rete, che genera un contributo addizionale in bolletta (demand rate) proporzionale al valore massimo di potenza assorbito su base settimanale o mensile

Riformuliamo il problema di ottimizzazione appena trattato includendo nella funzione obiettivo un contributo proporzionale al picco di potenza assorbito nelle 24 ore.

$$\begin{aligned}
 J_2 &= \sum_{k=1}^{96} T_s [L(k) + Q(k)] C(k) + c_{maxpow} \max_{1 \leq k \leq 96} [L(k) + Q(k)] = \\
 &= T_s [L + Q]^T + c_{maxpow} \max[L + Q]
 \end{aligned}$$

Funzione obiettivo

Il risultante problema di ottimizzazione:

Minimizza: $J_2(Q) = T_s[L + Q]^T + c_{maxpow} \max [L + Q]$

Sotto i vincoli:

$$L(k) + Q(k) \geq 0$$

$$Q_{min} \leq Q(k) \leq Q_{Max}$$

$$S_{min} \leq S(k) \leq S_{Max} \quad k = 1, 2, \dots, 96$$

$$S(k) = S(k - 1) + T_s Q(k)$$

$$S(0) = S_0$$

$$0.95S(0) \leq S(96) \leq 1.05S(0)$$

non appartiene a nessuna delle classi precedentemente illustrate, e come tale richiede l'impiego di un algoritmo di ottimizzazione generico (ad esempio uno di quelli inglobati nella funzione Matlab `fmincon`).

Vi è però modo di **reformulare il problema in modo differente**, riconducendone la struttura a quella di un problema di tipo LP.

L'idea è quella di introdurre una nuova variabile di ottimizzazione scalare P_{max}

$$\text{Minimizza: } J_2(Q, P_{max}) = T_s[L + Q]^T + c_{maxpow} P_{max}$$

e ricercare il valore minimo della funzione obiettivo sotto il **vincolo addizionale**

$$P_{max} \geq L(k) + Q(k) \quad 1 \leq k \leq 96$$

che si unisce ai vincoli precedentemente considerati

$$L(k) + Q(k) \geq 0$$

$$Q_{min} \leq Q(k) \leq Q_{Max}$$

$$S_{min} \leq S(k) \leq S_{Max} \quad 1 \leq k \leq 96$$

$$S(k) = S(k - 1) + T_s Q(k)$$

$$S(0) = S_0$$

$$0.95S(0) \leq S(96) \leq 1.05S(0)$$

Abbiamo ottenuto un problema LP nelle 97 incognite

$$Q(1), Q(2), \dots, \\ Q(95), Q(96), P_{max}$$

```

S0=0;
Smin=0;
Smax=390; % [kWh]
Qmax=50; % [kW]
Ts=0.25;
cpowpeak=0.3; % [€/kW]
File: BatteryManagementBuilding_PeakShavingLP.m

Costo giornaliero ricavato normalizzando di un fattore 30 l'importo su base mensile

Q = optimvar('Q',N,1,'LowerBound',-Qmax,'UpperBound',Qmax);
S = optimvar('S',N,1,'LowerBound',Smin,'UpperBound',Smax);
Pmax = optimvar('Pmax',1,1,'LowerBound',0);

obj=Ts*((L+Q)*C)+cpowpeak*Pmax;

zcons1 = S(1)==S0+Ts*Q(1);
zcons2 = S(2:N)==S(1:N-1)+Ts*Q(2:N);
LplusQcons = L+Q>=0;
ZNconslow = 0.95*S0 <=1.05*S(N);
ZNconshigh = S(N)<=1.05*S0 ;
maxpowconstr = L+Q<=Pmax;

optbattery=optimproblem;
optbattery.Objective = obj;

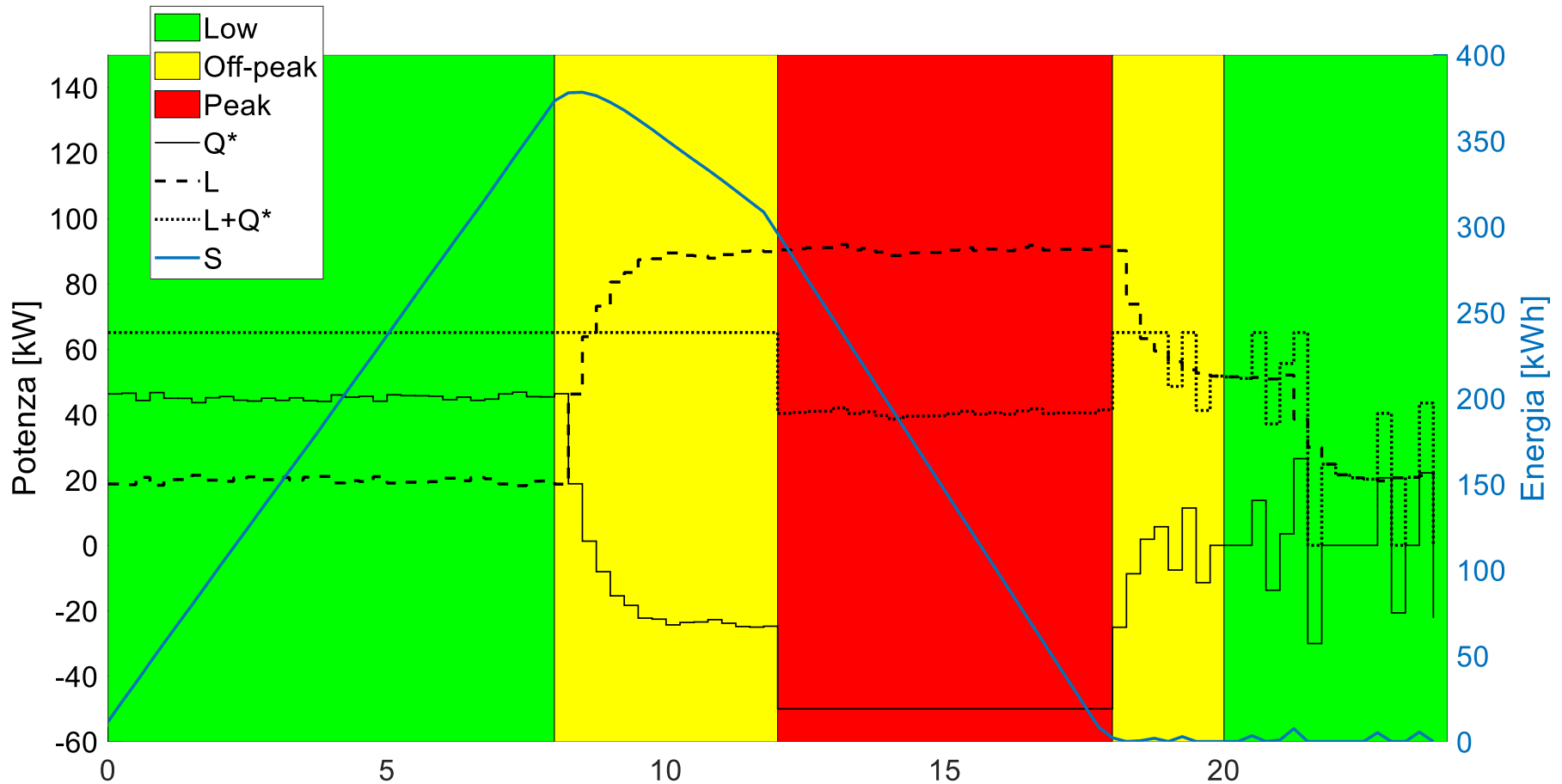
optbattery.Constraints.zcons1 = zcons1;
optbattery.Constraints.zcons2 = zcons2;
optbattery.Constraints.LplusQcons = LplusQcons;
optbattery.Constraints.ZNconslow = ZNconslow;
optbattery.Constraints.ZNconshigh = ZNconshigh;
optbattery.Constraints.maxpowconstr = maxpowconstr;

options = optimoptions('linprog','Display','final');

[optbatterySOL mincost]= solve(optbattery,'options',options);

```

Risultati



Si focalizzi l'attenzione sulla curva $L+Q^*$, che presenta un andamento molto più regolare ed in particolare un **picco molto meno marcato** rispetto al caso precedente

```
Solving problem using linprog.
```

```
Optimal solution found.
```

```
spese sostenute: [Eur]
```

```
mincost =
```

```
100.0218
```

```
spese senza batteria: [Eur]
```

```
spesenobatt =
```

```
121.1984
```

```
risparmio_percentuale =
```

```
17.4727
```

L'impiego ottimizzato della batteria ha consentito un risparmio su base giornaliera superiore al **17%**

Spunti di approfondimento

Il problema potrebbe essere approfondito in diverse direzioni

Sia la capacità S_{Max} della batteria che il massimo rate di carica Q_{Max} (dipendente dalla taglia dell'inverter connesso alla batteria) potrebbero essere considerate variabili da ottimizzare sulla base di un bilancio economico più articolato che tenga in considerazione i relativi costi di investimento e gli anni attesi di impiego della batteria.

I processi di carica e scarica della batteria potrebbero essere modellati in maniera più accurata, includendo le relative **perdite** e/o penalizzando in aggiunta i cicli di carica e scarica, che velocizzano l'invecchiamento della batteria ed il decadimento delle prestazioni.

Economic dispatching con costi di produzione quadratici

Torniamo al problema del dispacciamento economico della produzione giornaliera di un gruppo di Ng generatori (step temporale 1 ora) affrontato in precedenza.

Una modellazione più accurata dei costi di produzione include una parte costante, indipendente dal livello di potenza, ed una parte variabile caratterizzata da un termine lineare e da un termine **quadratico**

$$F_i(p_i(k)) = a_i p_i^2(k) + b_i p_i(k) + \rho_i$$

La funzione obiettivo da minimizzare sarà quindi di tipo **quadratico**

$$J = \sum_{k=1}^{24} \sum_{i=1}^{Ng} F_i(p_i(k)) = \sum_{k=1}^{24} \sum_{i=1}^{Ng} [a_i p_i^2(k) + b_i p_i(k)] = \frac{1}{2} x^T H x + c^T x$$

I termini costanti possono essere rimossi dalla funzione obiettivo

La matrice H ed il vettore c si determinano una volta che sia stata definita la struttura di un vettore x che contenga, in un qualche ordine arbitrario, tutte le variabili di decisione $p_i(k)$ ($i = 1, \dots, Ng$, $k = 1, 2, \dots, 24$). Grazie al linguaggio di modellazione dell'Optimization Toolbox di Matlab non sarà comunque necessaria la determinazione esplicita della matrice H e del vettore c .

Il problema di ottimizzazione esula dalla programmazione lineare LP. **Analizziamo i vincoli per valutare se si abbia a che fare con un problema di tipo QP (che implica vincoli di tipo lineare) o con un problema di natura più complessa.**

Oltre ai vincoli sulla massima e minima potenza erogabile ed al vincolo di uguaglianza fra la potenza complessiva prodotta e la richiesta di carico $D(k)$, includiamo dei **vincoli aggiuntivi** sulla **massima variazione di potenza** fra due fasce orarie adiacenti $p_i(k) - p_i(k - 1)$.

Vincoli

$$P_i^{min} \leq p_i(k) \leq P_i^{max}$$

Limiti su massima e minima potenza erogabile

$$\sum_{i=1}^{Ng} p_i(k) = D(k)$$

potenza totale uguale al carico richiesto

$$-\zeta_i^{down} \leq p_i(k) - p_i(k - 1) \leq \zeta_i^{up}$$

Limiti su massima variazione di potenza consentita fra due intervalli temporali adiacenti (ramping rate limits)

Vincoli lineari e funzione obiettivo quadratica: problema QP

Funzione Matlab: **quadprog**

Risolviamo il problema QP considerando $Ng = 6$ generatori ed i seguenti dati

TABLE I. THE 6 UNIT TEST SYSTEM CHARACTERISTICS

Unit n°	P_i^{min}	P_i^{max}	a_i	b_i	ρ_i
1	100	500	0.007	7	240
2	50	200	0.0095	10	200
3	80	300	0.009	8	220
4	50	150	0.009	11	200
5	50	200	0.008	10.5	220
6	50	120	0.0075	12	190

$$J = \sum_{k=1}^{24} \sum_{i=1}^{Ng} [a_i p_i^2(k) + b_i p_i(k)]$$

$$P_i^{min} \leq p_i(k) \leq P_i^{max}$$

TABLE IV. RAMPING RATE LIMITS OF THE 6-UNIT TEST SYSTEM

Unit N°	ζ_i^{up}	ζ_i^{down}
1	80	120
2	50	90
3	65	100
4	50	90
5	50	90
6	50	90

$$-\zeta_i^{down} \leq p_i(k) - p_i(k-1) \leq \zeta_i^{up}$$

Carico

TABLE V. THE HOURLY LOAD DATA OF 24 TIME ONTERVED OF THE 6 UNIT TEST SYSTEM

	Time(h)	Load(MW)	Time(h)	Load(MW)
$D(k)$ $1 \leq k \leq 24$	1	955	13	1190
	2	942	14	1251
	3	953	15	1263
	4	930	16	1250
	5	935	17	1221
	6	963	18	1202
	7	989	19	1159
	8	1023	20	1092
	9	1126	21	1023
	10	1150	22	984
	11	1201	23	975
	12	1235	24	960

$$\sum_{i=1}^{Ng} p_i(k) = D(k)$$

$$D(k) = [955, \quad 942, \quad 953, \quad \dots, \quad 975, \quad 960] \in R^{24}$$

Funzione Matlab per i problemi QP:

quadprog

Script che risolve il problema:

QP_dispatching24h.m

```
clear all, close all, clc
```

```
numh=24; % numero slot orari
numgen=6; % numero generatori
```

```
a1=0.007; b1=7;
a2=0.0095; b2=10;
a3=0.009; b3=8;
a4=0.009; b4=11;
a5=0.008; b5=10.5;
a6=0.0075; b6=12;
```

```
Pm1=100; PM1=500;
Pm2=50; PM2=200;
Pm3=80; PM3=300;
Pm4=50; PM4=150;
Pm5=50; PM5=200;
Pm6=50; PM6=120;
```

```
D=[955 942 953 930 935 963 989 1023 1126 1150 1201 1235 1190 1251 1263 1250 1221 ...
1202 1159 1092 1023 984 975 960];
```

```
xi1U=80; xi1D=120;
xi2U=50; xi2D=90;
xi3U=65; xi3D=110;
xi4U=50; xi4D=90;
xi5U=50; xi5D=90;
xi6U=50; xi6D=90;
```

Definizione dei parametri

```

for i=1:numh-1
xiUext(:,i)=[xi1U;xi2U;xi3U;xi4U;xi5U;xi6U];
xiDext(:,i)=[xi1D;xi2D;xi3D;xi4D;xi5D;xi6D];
end

for i=1:numh
aext(:,i)=[a1;a2;a3;a4;a5;a6];
bext(:,i)=[b1;b2;b3;b4;b5;b6];
lbext(:,i)=[Pm1;Pm2;Pm3;Pm4;Pm5;Pm6];
ubext(:,i)=[PM1;PM2;PM3;PM4;PM5;PM6];
end

```

Matrici estese che replicano la medesima colonna

Definizione della decision variable (numgen = 6 righe e numh=24 colonne) con i relativi limiti di saturazione

```
p = optimvar('p', numgen, numh, 'LowerBound', lbext, 'UpperBound', ubext);
```

```
F=aext.*p.^2+bext.*p;
obj=sum(sum(F));
```

Definizione della funzione obiettivo

$$J = \sum_{k=1}^{24} \sum_{i=1}^6 [a_i p_i^2(k) + b_i p_i(k)]$$

La matrice F ha $N_g = 6$ righe ed $\text{numh}=24$ colonne, ed il suo generico elemento con indice di riga i ed indice di colonna k è: $a_i p_i^2(k) + b_i p_i(k)$

La doppia applicazione ad una matrice della funzione `sum` restituisce la somma fra tutti gli elementi della matrice, definendo quindi la funzione obiettivo

Definizione dei vincoli

```
powercons = sum(p) == D;
```

```
rateconsU=p(:,2:numh)-p(:,1:numh-1)<=xiUext;
```

```
rateconsD=p(:,2:numh)-p(:,1:numh-1)>=-xiDext;
```

```
QPdispatch=optimproblem;
```

```
QPdispatch.Objective = obj;
```

```
QPdispatch.Constraints.powercons = powercons;
```

```
QPdispatch.Constraints.rateconsU = rateconsU;
```

```
QPdispatch.Constraints.rateconsD = rateconsD;
```

```
options = optimoptions('quadprog', 'Display', 'final');
```

```
QPdispatchsol= solve(QPdispatch, 'options', options);
```

```
Pstar=QPdispatchsol.p;
```

$$\sum_{i=1}^{Ng} p_i(k) = D(k)$$

$$p_i(k) - p_i(k - 1) \leq \zeta_i^{up}$$

$$p_i(k) - p_i(k - 1) \geq -\zeta_i^{down}$$

Definizione del problema di ottimizzazione.

Assegnazione della funzione obiettivo e dei vincoli. Scelta del solutore **quadprog** e risoluzione del problema.

La soluzione ottima viene memorizzata nella variabile `Pstar`

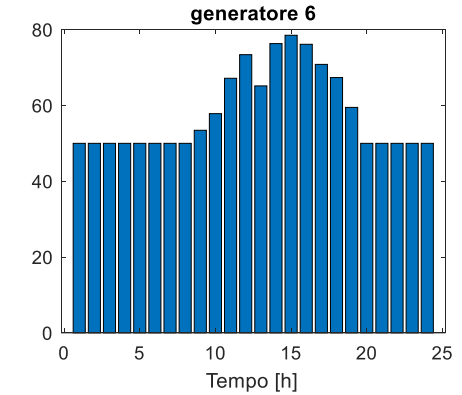
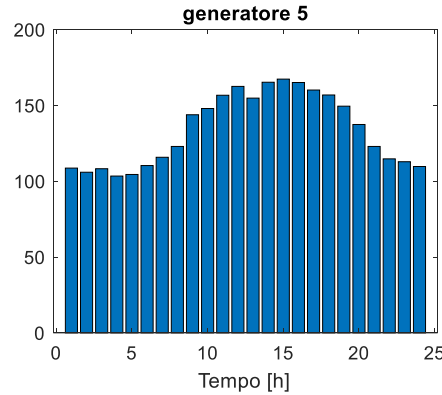
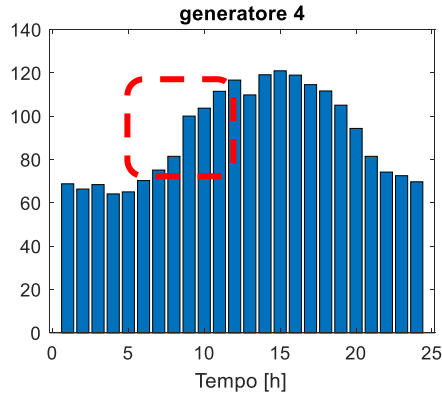
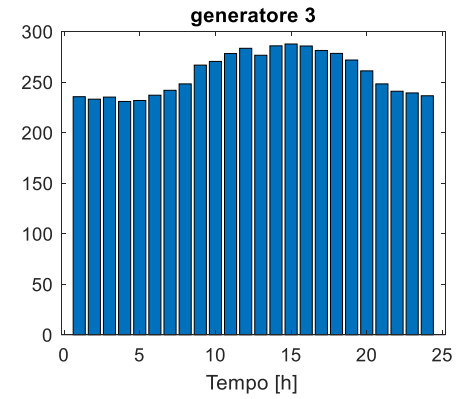
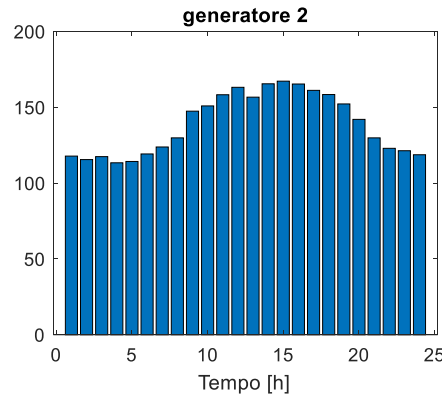
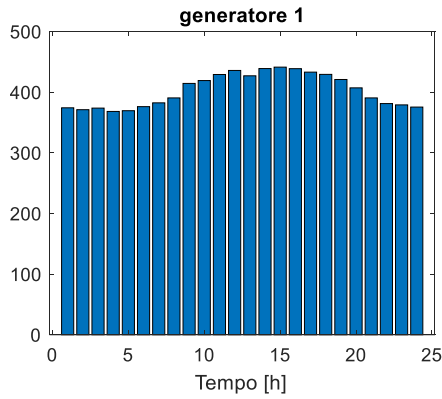
Solving problem using quadprog.

Minimum found that satisfies the constraints.

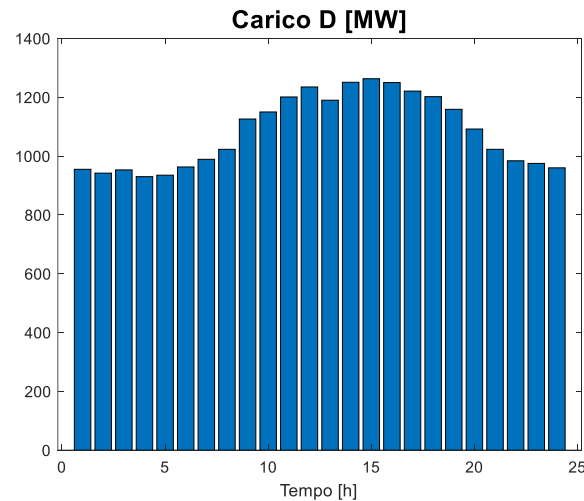
Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied to within the value of the constraint tolerance.

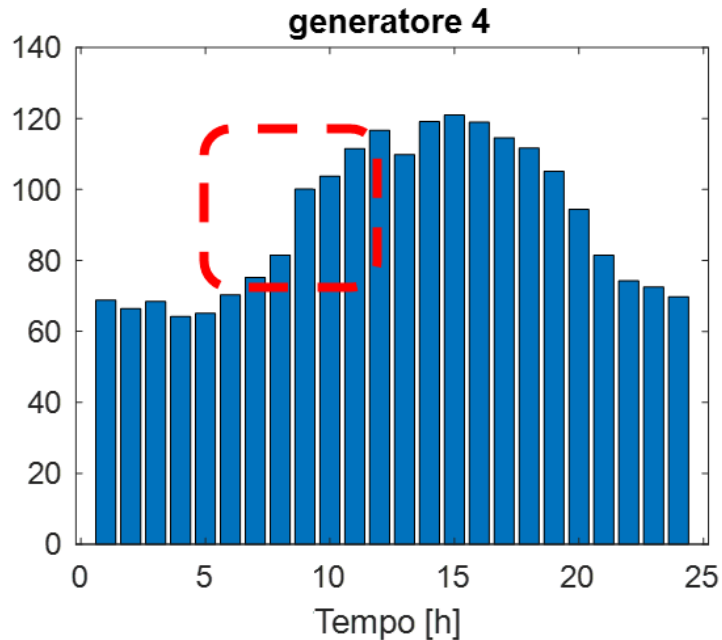
<stopping criteria details>

|



Carico





Per ridurre il picco nella variazione di potenza in salita del **quarto generatore** imponiamo dei vincoli più stringenti sul ramping rate, diminuendo il valore del parametro ζ_4^{up}

Valore
precedentemente
utilizzato

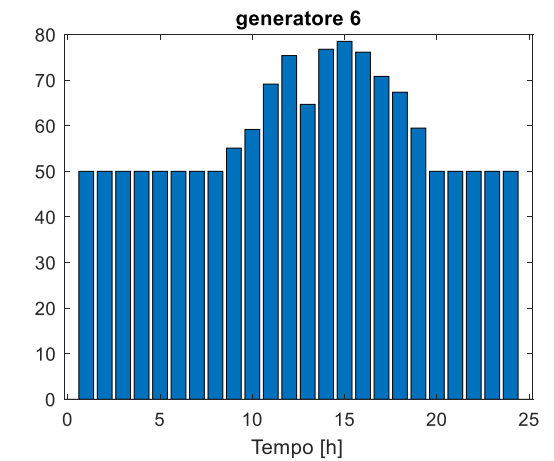
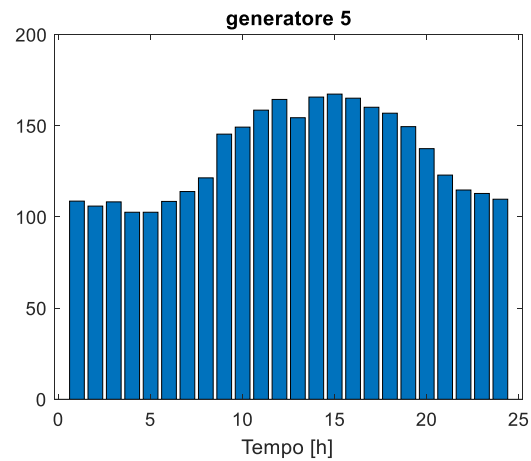
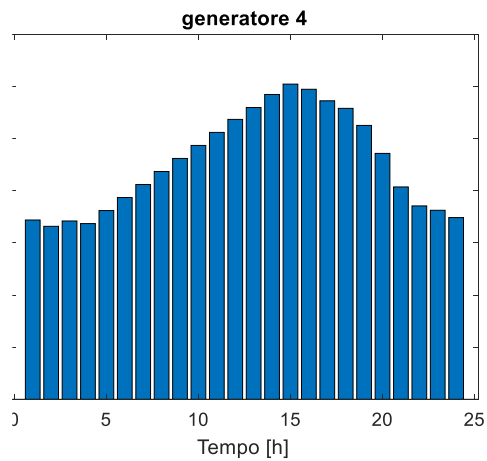
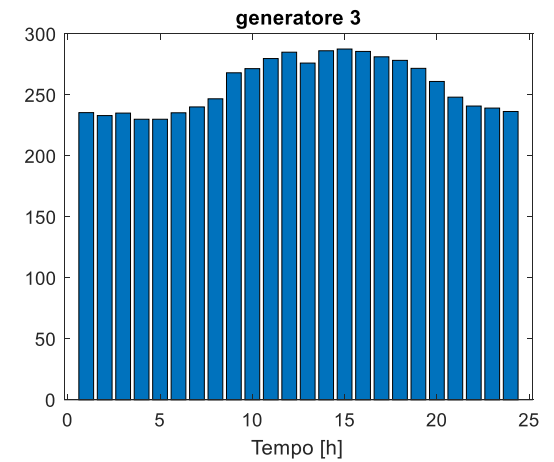
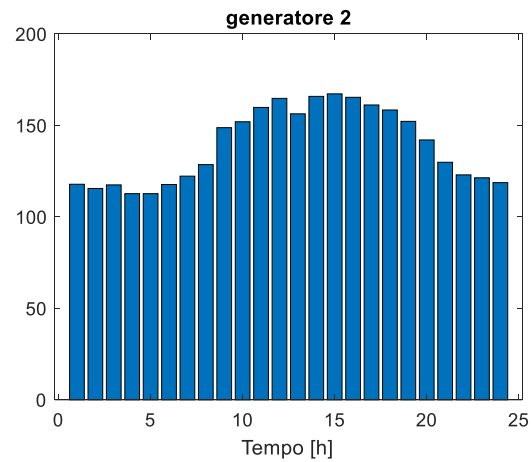
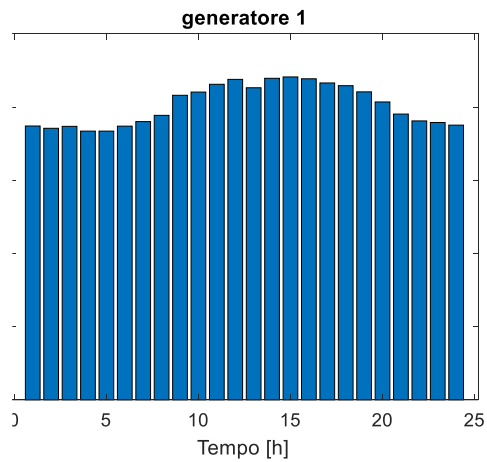
$$\zeta_4^{up} = 50.$$

Nuovo valore

$$\zeta_4^{up} = 5$$

Risultati

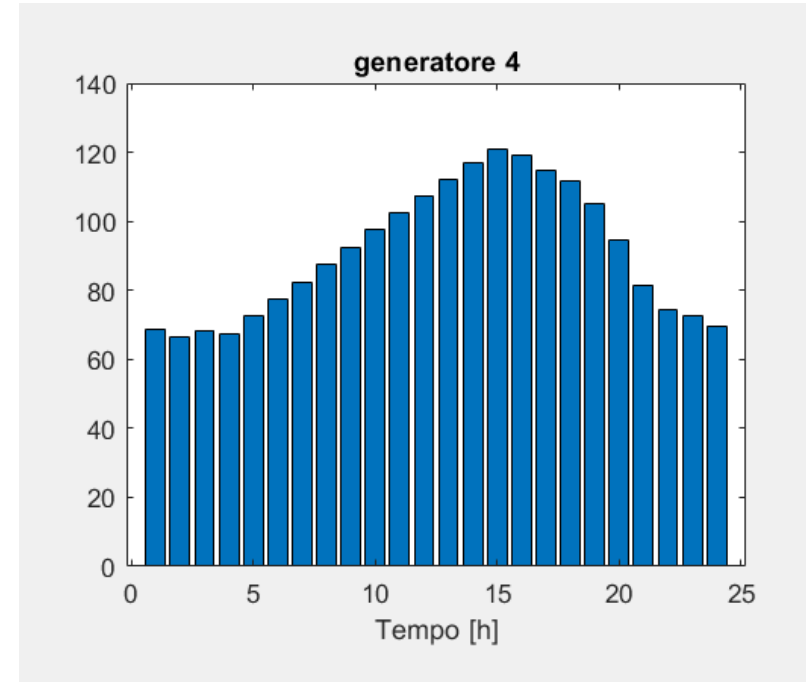
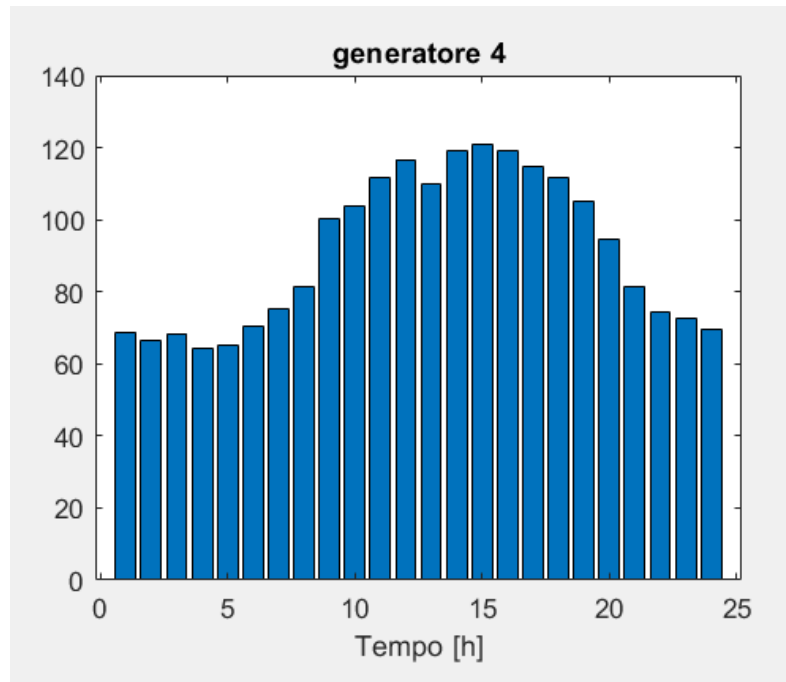
$$\zeta_4^{up} = 5$$



Confronto fra i due casi

$$\zeta_i^{up} = 50$$

$$\zeta_i^{up} = 5$$



Ora includiamo anche le perdite di trasmissione

$$P_T(k) = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p_i(k) B_{ij} p_j(k) + \sum_{i=1}^{Ng} B_{0i} p_i(k) + B_{00} \approx \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p_i(k) B_{ij} p_j(k)$$

Il vincolo $\sum_{i=1}^{Ng} p_i(k) = D(k)$ sulla produzione complessiva va **modificato** in quanto deve includere le perdite di trasmissione, e diventa quindi un **vincolo quadratico di uguaglianza** (il quale, accoppiato alla funzione obiettivo quadratica, dà luogo ad un **problema QCQP**)



$$\sum_{i=1}^{Ng} p_i(k) - P_T(k) = D(k)$$

Coefficienti di perdita («**B-coefficients**»)

TABLE II. B-COEFFICIENTS OF IEEE 30-BUS 6-UNIT SYSTEM

$$\sum_{i=1}^{Ng} p_i(k) - \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p_i(k) B_{ij} p_j(k) = D(k)$$

	2.231	1.162	-0.122	-0.017	0.113	0.39
	1.162	1.89	-0.077	-0.048	0.069	0.28
	-0.122	-0.077	2.004	-0.74	-0.724	-0.599
	-0.017	-0.048	-0.74	1.479	0.538	0.342
	0.113	0.069	-0.724	0.538	1.185	0.053
	0.39	0.28	-0.599	0.342	0.053	2.34
$B = 10^{-4} \times$						
	0.38	1.79	-5.32	1.52	2.33	1.26
$B_0 = 10^{-5} \times$						
$B_{00} =$	0.00154					

Problema complessivo (di tipo **QCQP**)

Funzione obiettivo da minimizzare

$$J = \sum_{k=1}^{24} \sum_{i=1}^{Ng} [a_i p_i^2(k) + b_i p_i(k)] \quad \text{Costo di produzione}$$

Vincoli

$$P_i^{min} \leq p_i(k) \leq P_i^{max} \quad \text{Limiti su massima e minima potenza erogabile}$$

$$\sum_{i=1}^{Ng} p_i(k) - \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p_i(k) B_{ij} p_j(k) = D(k) \quad \text{Potenza totale (al netto delle perdite di trasmissione) uguale al carico richiesto}$$

$$-\zeta_i^{down} \leq p_i(k) - p_i(k-1) \leq \zeta_i^{up} \quad \text{Limiti su massima variazione di potenza consentita fra due intervalli temporali adiacenti}$$
$$1 \leq k \leq 24$$

Poiché Matlab non ha una funzione dedicata ai problemi QCQP impieghiamo la funzione **fmincon** (che richiede che sia precisato un **punto iniziale** per la ricerca della soluzione)

Script che risolve il problema in esame: **QCQP_dispatching24h.m**

Modifiche e aggiunte al codice

File: **QCQP_dispatching24h.m**

```
B=0.0001*[2.231 1.162 -0.122 -0.017 0.113 0.39;
          1.162 1.89 -0.077 -0.048 0.069 0.28;
          -0.122 -0.077 2.004 -0.74 -0.724 -0.529;
          -0.017 -0.048 -0.74 -1.479 0.538 0.342;
          0.113 0.069 -0.724 0.538 1.185 0.053;
          0.39 0.28 -0.599 0.342 0.053 2.34]
```

Definizione matrice B dei coefficienti di perdita

```
for k=1:numh
    aux=0;
    for i=1:numgen
        for j=1:numgen
            aux=aux+p(i,k)*B(i,j)*p(j,k);
        end
    end
    PT(k)=aux;
end
```

Costruzione di $P_T(k) = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} p_i(k) B_{ij} p_j(k)$

```
powercons = sum(p) - PT == D;
```

Modifica del vincolo: $\sum_{i=1}^{Ng} p_i(k) - P_T(k) = D(k)$

Inclusione delle perdite $P_T(k)$ nel vincolo

Modifiche e aggiunte al codice (cont.)

```
p0.p = zeros(numgen, numh);
```

Definizione del **punto iniziale**.

```
options = optimoptions('fmincon', 'Display', 'final');
```

 Scelta del solutore

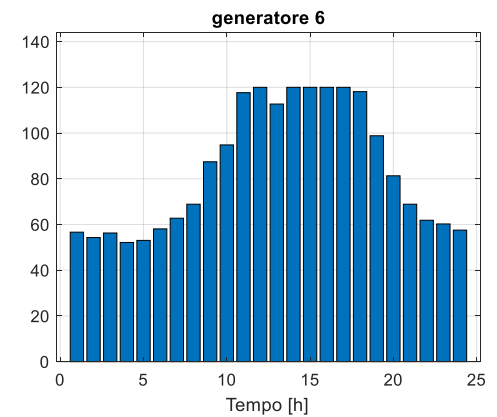
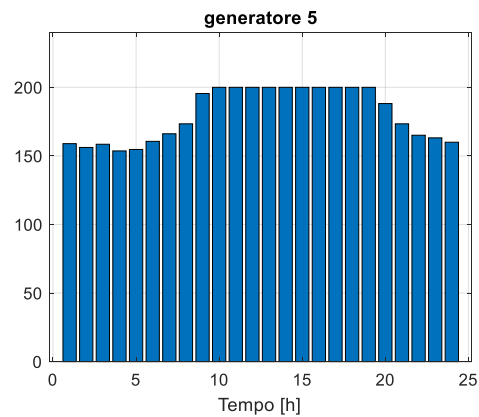
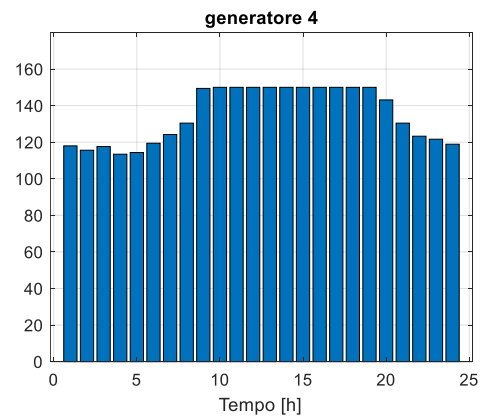
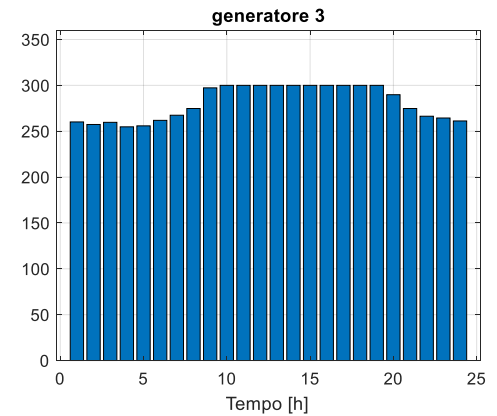
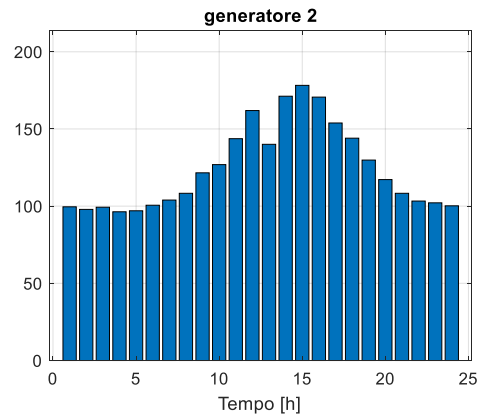
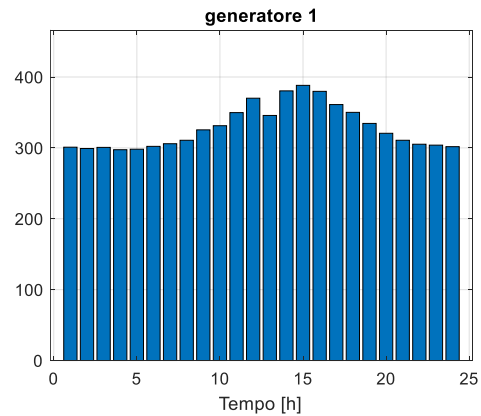
```
QPdispatchsol=solve(QPdispatch, p0, 'options', options);
```

 Risoluzione

```
Pstar=QPdispatchsol.p;
```

La soluzione ottima viene memorizzata nella
variabile Pstar

Risultati



Ottimizzazione in presenza di incertezze (ottimizzazione **robusta**)

E' stata menzionata in precedenza la classe di **problemi SOCP** (Second-Order Cone Programming)

Tale classe, che sarà definita a breve, costituisce una particolare tipologia di problemi con funzione obiettivo lineare e **vincoli non lineari** che trova applicazione nella cosiddetta **ottimizzazione robusta**, ove all'interno del problema sono presenti una o più variabili il cui valore è affetto da incertezza.

Mostriamo quindi come poter applicare tali formalismi e tecniche nell'ambito di un problema di dispacciamento in cui alcuni generatori sono da fonte rinnovabile, e come tali la produzione associata è intermittente e soggetta a pesanti fattori di incertezza

Second-order cone programming

SOCP

Problema generale SOCP

$$\min_x c^T x \quad \text{Funzione obiettivo}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Decision variables

$$x_i \in \mathbb{R}$$

Vincoli $Ax \leq b$

$$A_{eq}x = b_{eq}$$

$$lb \leq x \leq ub$$

$$\|A_{ci}x + b_{ci}\|_2 \leq d_{ci}x - \gamma_{ci} \quad i = 1, 2, \dots, N_{sc}$$

Second-order cone constraints

I second-order cone constraints sono dei vincoli non lineari di disuguaglianza che individuano particolari domini convessi denominati «coni del secondo ordine»

A_{ci} rappresenta una matrice quadrata (indicizzata per $i = 1, 2, \dots, N_{sc}$) mentre b_{ci}, d_{ci} sono vettori, e γ_{ci} degli scalari

I second-order cone constraints sono espressi in funzione della «norma 2» $\|A_{ci}x + b_{ci}\|_2$ del vettore $A_{ci}x + b_{ci}$

La «norma 2» coincide con la norma Euclidea. Sia z un generico vettore colonna:

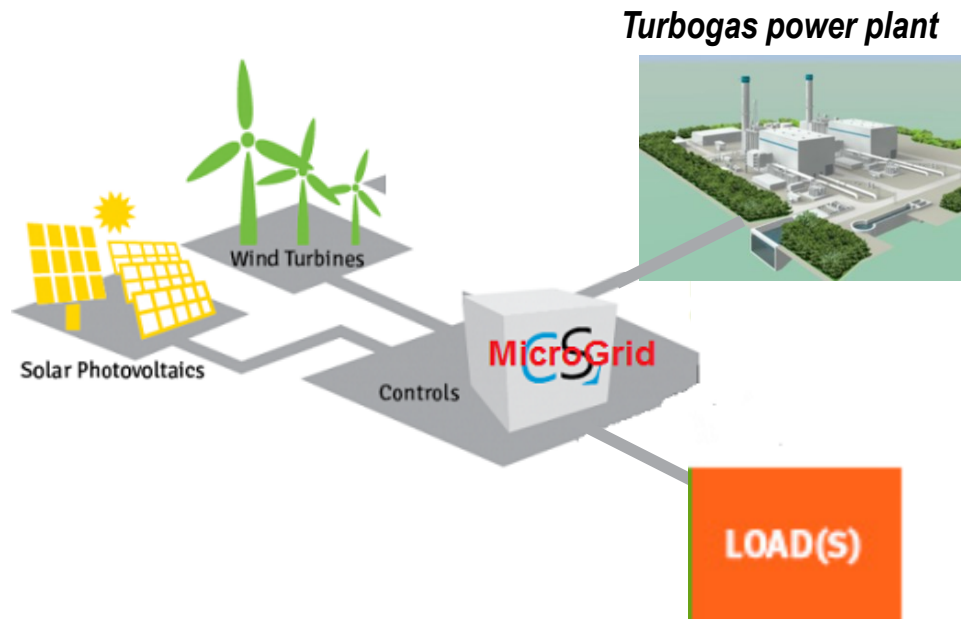
$$\|z\|_2 = \sqrt{z^T z}$$

Vediamo per mezzo di un esempio come un problema di dispacciamento soggetto ad incertezze conduca naturalmente ad una sua formalizzazione mediante un problema SOCP

Consideriamo una micro-rete ad elevata penetrazione di FER, con tre generatori due dei quali sono da FER, ed uno da fonte fossile. Formalizziamo un problema di dispacciamento includendo le inevitabili incertezze sulla produzione oraria e giornaliera dei generatori da FER

Esempio

Dispacciamento dei generatori in una microrete ad elevata penetrazione di FER che contiene un parco eolico, un parco solare fotovoltaico, ed un impianto turbogas.



Il parco eolico ha una potenza nominale $P_{nom}^W = 20 \text{ MW}$. Ciò significa che in relazione alle particolari condizioni climatiche potrà erogare una potenza p_W compresa fra 0 e 20 MW . Tale valore non potrà essere pianificato con esattezza.

Similarmente, il parco solare PV ha una potenza nominale $P_{nom}^{PV} = 10 \text{ MW}$, ed in corrispondenza delle particolari condizioni di insolazione potrà erogare una potenza p_{PV} compresa fra 0 e 10 MW .

L'impianto turbogas ha una potenza nominale $P_{nom}^{TG} = 50 \text{ MW}$ e la sua produzione p_{TG} può considerarsi «certa».

Per modellare tale scenario di incertezza, definiamo le tre variabili di decisione normalizzate (espresse come una frazione del livello di potenza nominale)

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_W \\ x_{PV} \\ x_{TG} \end{bmatrix} \quad x_W = \frac{p_W}{P_{nom}^W} \quad x_{PV} = \frac{p_{PV}}{P_{nom}^{PV}} \quad x_{TG} = \frac{p_{TG}}{P_{nom}^{TG}}$$

Ovviamente si avrà: $0 \leq x_i \leq 1$, $i = 1, 2, 3$.

N.B. La normalizzazione delle variabili rispetto al loro valore massimo è una operazione sempre consigliabile nell'ambito dell'ottimizzazione, anche in contesti che non coinvolgono la presenza di incertezza, per motivi di stabilità numerica dell'algoritmo di risoluzione del problema di ottimizzazione.

Siano c_1, c_2, c_3 i costi marginali di produzione **normalizzati**, espressi in euro

La funzione obiettivo da minimizzare è pertanto:

$$J = c_1 x_1 + c_2 x_2 + c_3 x_3 = c^T x \quad c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

La **potenza complessiva erogata** dai tre generatori è

$$p_{TOT} = p_W + p_{PV} + p_{TG} = r_W x_W + r_{PV} x_{PV} + r_{TG} x_{TG}$$

in cui i coefficienti r_W ed r_{PV} sono **incerti** ed appartenenti ai seguenti intervalli

$$0 \leq r_W \leq P_{nom}^W = 20MW \quad 0 \leq r_{PV} \leq P_{nom}^{PV} = 10 MW$$

mentre il coefficiente r_{TG} è perfettamente noto e vale $r_{TG} = P_{nom}^{TG}$

Essendo la potenza complessiva p_{TOT} erogata dai tre generatori **incerta**, il vincolo relazionato alla domanda di potenza D dovrà essere formulato in termini di una **disuguaglianza**:

$$\begin{aligned}
 p_{TOT} &= r_W x_W + r_{PV} x_{PV} + r_{TG} x_{TG} \\
 &= r_1 x_1 + r_2 x_2 + r_3 x_3 \geq D
 \end{aligned}
 \qquad
 r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} r_W \\ r_{PV} \\ r_{TG} \end{bmatrix}$$

$$\text{Minimizzare:} \quad c_1 x_1 + c_2 x_2 + c_3 x_3 = c^T x \qquad c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

$$\begin{aligned}
 \text{Sotto i vincoli:} \quad & r_1 x_1 + r_2 x_2 + r_3 x_3 \geq D \\
 & 0 \leq x_i \leq 1 \qquad i = 1, 2, 3.
 \end{aligned}$$

In forma compatta:

$$\begin{aligned}
 \text{Minimizzare:} \quad & c^T x \\
 \text{Sotto i vincoli:} \quad & r^T x \geq D \\
 & 0 \leq x \leq 1
 \end{aligned}$$

N.B. Il vettore r è incerto

Minimizzare: $c^T x$

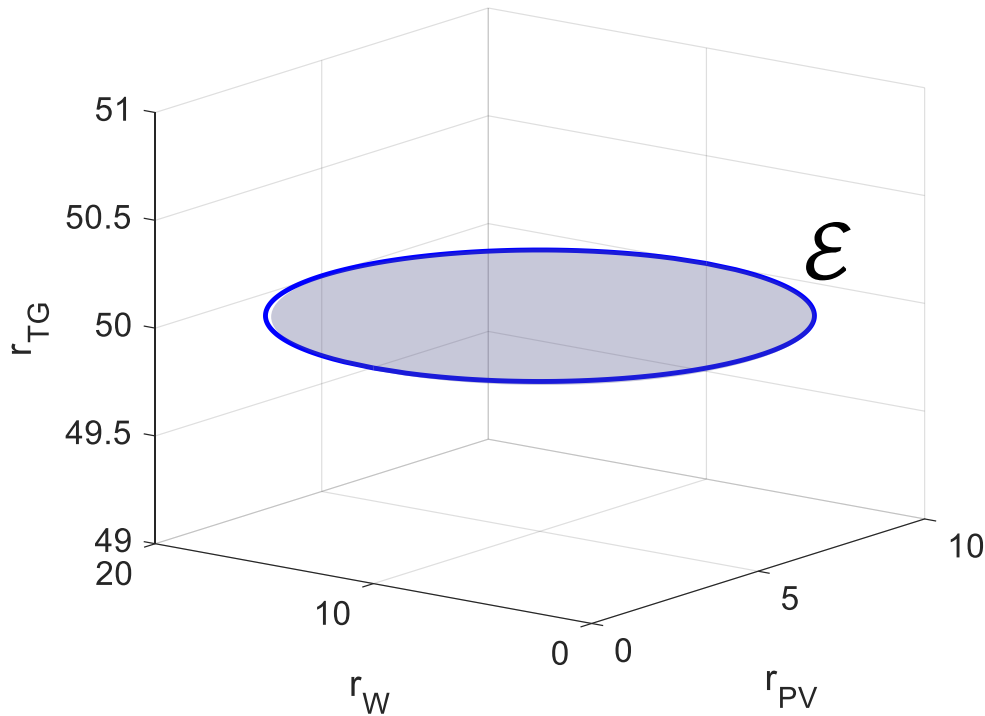
Sotto i vincoli: $r^T x \geq D$

$$0 \leq x \leq 1$$

$$r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} r_W \\ r_{PV} \\ r_{TG} \end{bmatrix}$$

Il vettore r è incerto

Ipotizziamo che il vettore incerto r appartenga ad un **ellissoide** \mathcal{E}



Rappresentazione nello spazio

E' un ellissoide degenere «schiacciato» sul piano $r_{TG} = 50$ poichè non c'è incertezza sul valore di r_{TG} .

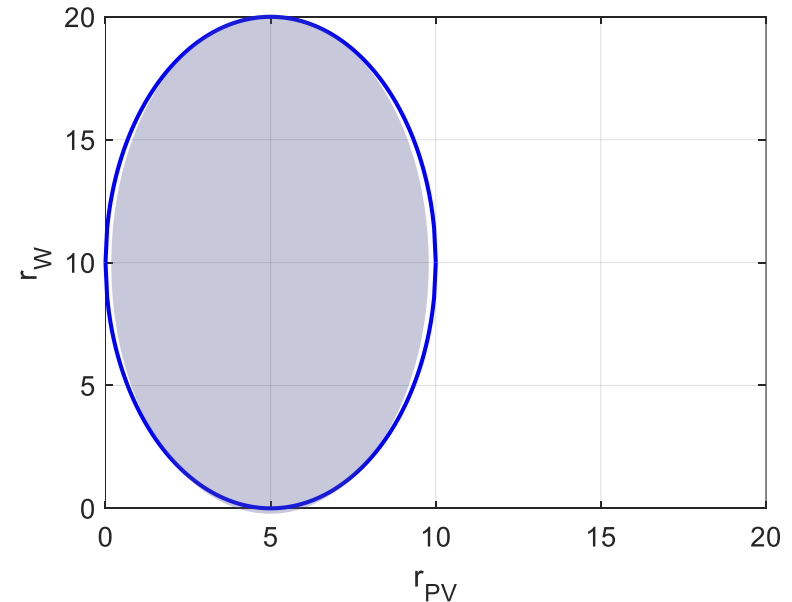
$$0 \leq r_W \leq P_{nom}^W = 20MW$$

$$0 \leq r_{PV} \leq P_{nom}^{PV} = 10 MW$$

$$r_{TG} = P_{nom}^{TG} = 50 MW$$

Assumere l'appartenenza del vettore a tale ellissoide implica assumere, ad esempio, che se il parco eolico ha una produzione nulla ($r_W = 0$) il parco PV produce (almeno) al 50% della sua capacità

Proiezione sul piano orizzontale



```
close all

u1=-1:0.01:1;
x1=5+5*u1;
x2=10+10*sqrt(1-u1.^2);
x2neg=10-10*sqrt(1-u1.^2)
x3=50*ones(1,length(u1));

figure(1)
plot3(x1,x2,x3,'b','LineWidth',2)
hold on
plot3(x1,x2neg,x3,'b','LineWidth',2)
grid
xlabel('r_{PV}','FontSize',14)
ylabel('r_W','FontSize',14)
zlabel('r_{TG}','FontSize',14)
set(gca,'FontSize',14)

figure(2)
plot(x1,x2,'b','LineWidth',2)
hold on
plot(x1,x2neg,'b','LineWidth',2)
grid
ylabel('r_W','FontSize',14)
xlabel('r_{PV}','FontSize',14)
xlim([0 20])
ylim([0 20])
set(gca,'FontSize',14)
```

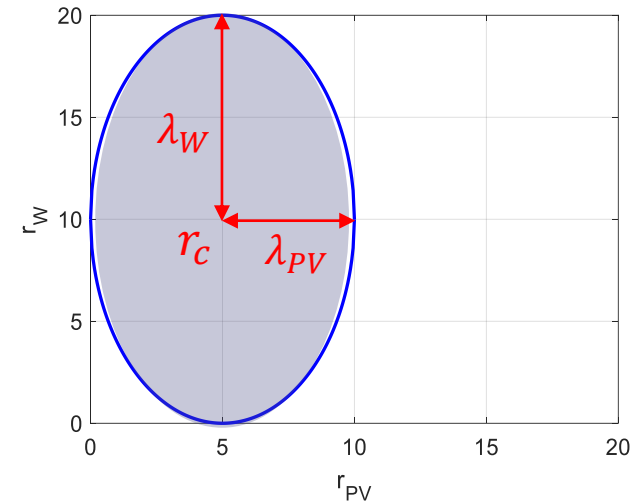
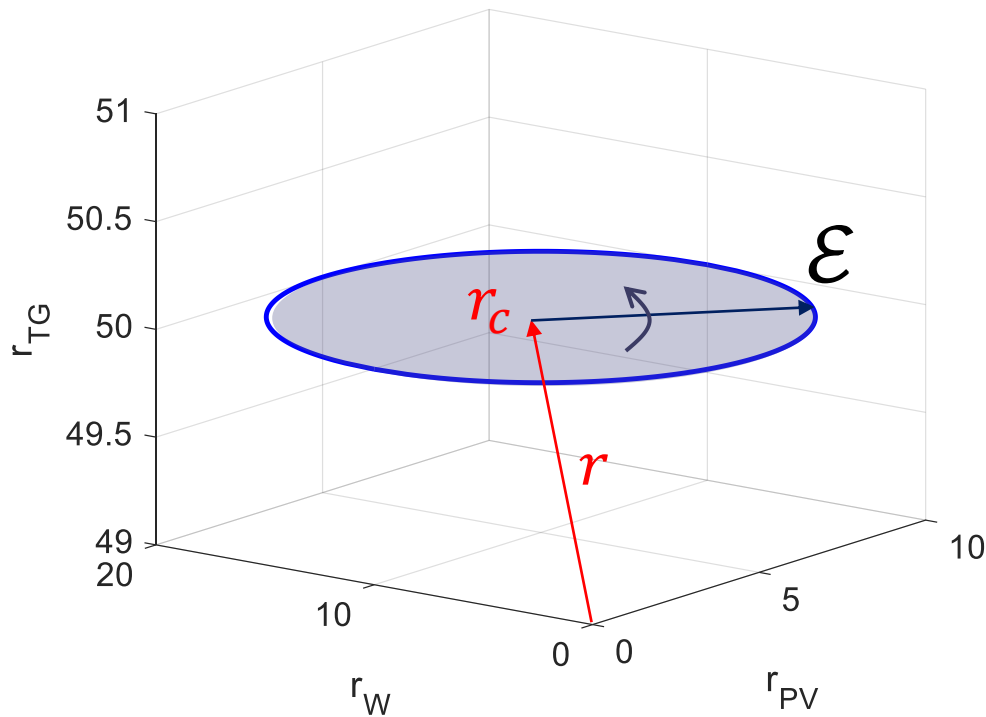
Codice Matlab per realizzare
le figure della slide precedente

Desideriamo risolvere il seguente problema di ottimizzazione robusta.

$$\begin{array}{ll} \text{Minimizzare:} & c^T x \\ \text{Sotto i vincoli:} & r^T x \geq D \quad \forall r \in \mathcal{E} \\ & 0 \leq x \leq 1 \end{array}$$

$$r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} r_W \\ r_{PV} \\ r_{TG} \end{bmatrix}$$

Il primo passo è caratterizzare in forma analitica (e conveniente per le analisi che dovremo effettuare) il vincolo di appartenenza di r all'ellissoide \mathcal{E}



$$\mathcal{E} = \{r \in R^3: r = r_c + Pu, \|u\|_2 \leq 1\} \quad r_c = \begin{bmatrix} \lambda_W \\ \lambda_{PV} \\ r_{TG} \end{bmatrix} \quad P = \begin{bmatrix} \lambda_W & 0 & 0 \\ 0 & \lambda_{PV} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

La matrice **diagonale** P contiene sulla diagonale principale i semiassi λ_W e PV dell'ellissoide (che valgono rispettivamente $\lambda_W = 10$ e $\lambda_{PV} = 5$). L'ultimo elemento diagonale è nullo in quanto l'ellisse è «schiacciato» sul piano $r_{TG} = 50$. Se anche il terzo generatore fosse stato da FER allora avremmo avuto un ellissoide «standard» in 3D e la matrice P avrebbe avuto anche il terzo elemento diagonale diverso da zero (e sarebbe quindi stata definita positiva).

Il problema di ottimizzazione robusta originario

$$\begin{array}{ll}
 \text{Minimizzare:} & c^T x \\
 \text{Sotto i vincoli:} & r^T x \geq D \quad \forall r \in \mathcal{E} \\
 & 0 \leq x \leq 1
 \end{array}$$

$$r = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} r_W \\ r_{PV} \\ r_{TG} \end{bmatrix}$$

viene espresso nella forma equivalente («worst-case scenario»)

$$\begin{array}{ll}
 \text{Minimizzare:} & c^T x \\
 \text{Sotto i vincoli:} & \min_{r \in \mathcal{E}} r^T x \geq D \\
 & 0 \leq x \leq 1
 \end{array}$$

La quantità $\min_{r \in \mathcal{E}} r^T x$ che compare nel vincolo può essere espressa facilmente in funzione dei parametri r_c e P che definiscono l'ellissoide \mathcal{E}

$$\mathcal{E} = \{r = r_c + Pu, \|u\|_2 \leq 1\}$$

Valutiamo la quantità $\min_{r \in \mathcal{E}} r^T x$

Sostituiamo all'interno di $r^T x$ l'espressione del punto generico r dell'ellissoide $\mathcal{E} = \{r = r_c + Pu, \|u\|_2 \leq 1\}$, e manipoliamo i vari termini

$$r^T x = (r_c + Pu)^T x = r_c^T x + (Pu)^T x = r_c^T x + u^T P^T x \quad \|u\|_2 \leq 1$$

$$r^T x = r_c^T x + u^T P^T x \quad \Rightarrow \quad \min_{r \in \mathcal{E}} r^T x = r_c^T x - \max_{u: \|u\|_2 \leq 1} u^T P^T x \quad \text{«Caso peggiore»}$$

Da proprietà note della norma euclidea si ricava che $\max_{u: \|u\|_2 \leq 1} u^T P^T x = \|Px\|_2$

$$\min_{r \in \mathcal{E}} r^T x = r_c^T x - \|Px\|_2$$

Ora che abbiamo determinato l'espressione della quantità $\min_{r \in \mathcal{E}} r^T x$, possiamo sostituirla all'interno del vincolo $\min_{r \in \mathcal{E}} r^T x \geq D$. Andremo ad ottenere un vincolo del tipo «**second-order cone constraint**»

$$\min_{r \in \mathcal{E}} r^T x \geq D \quad \longrightarrow \quad r_c^T x - \|Px\|_2 \geq D$$

Invertiamo il senso della disuguaglianza ed isoliamo alla sinistra dell'uguale il termine $\|Px\|_2$:

$$\|Px\|_2 \leq r_c^T x - D$$

Se la confrontiamo con l'espressione generale di un second-order cone constraint

$$\|A_c x + b_c\|_2 \leq d_c^T x - \gamma_c$$

Si osserva la corrispondenza se

$$\begin{aligned} A_c &= P & d_c &= r_c^T \\ b_c &= 0 & \gamma_c &= D \end{aligned}$$

Siamo giunti al seguente problema SOCP

Minimizzare: $c^T x$

Sotto i vincoli: $\|Px\|_2 \leq r_c^T x - D$

$$0 \leq x \leq 1$$

Risolviamolo mediante Matlab con riferimento ad un test di dispacciamento

Potenze nominali

$$P_{nom}^W = 20 \text{ MW}$$

$$P_{nom}^{PV} = 10 \text{ MW}$$

$$P_{nom}^{TG} = 50 \text{ MW}$$

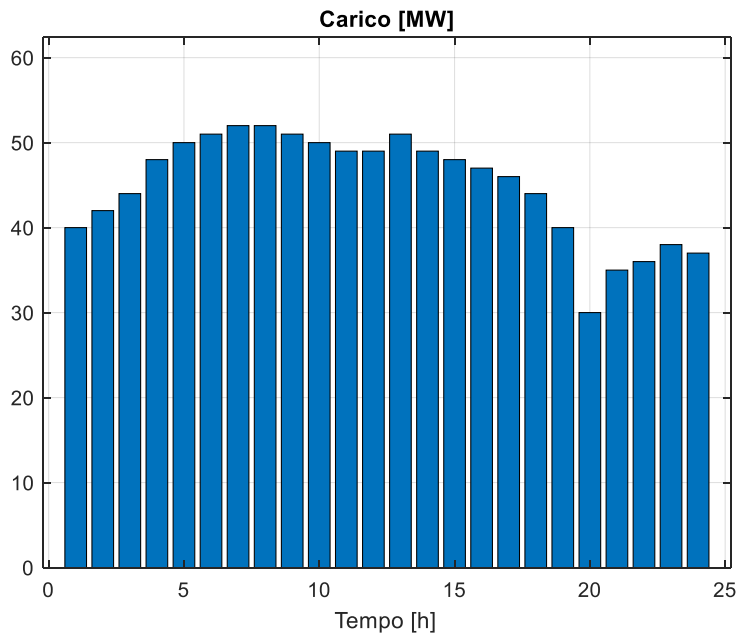
$$\lambda_W = \frac{P_{nom}^W}{2} = 10 \text{ MW}$$

$$\lambda_{PV} = \frac{P_{nom}^{PV}}{2} = 5 \text{ MW}$$

$$r_{TG} = P_{nom}^{TG} = 50 \text{ MW}$$

$$r_c = \begin{bmatrix} \lambda_W \\ \lambda_{PV} \\ r_{TG} \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \\ 50 \end{bmatrix}$$

$$P = \begin{bmatrix} \lambda_W & 0 & 0 \\ 0 & \lambda_{PV} & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



Carico

Costi marginali di produzione

$$c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} c_W \\ c_{PV} \\ c_{TG} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 25 \end{bmatrix}$$

Problema in formulazione originaria

Minimizzare: $c^T x$

$$c = \begin{bmatrix} c_W \\ c_{PV} \\ c_{TG} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 25 \end{bmatrix} \quad r_c = \begin{bmatrix} \lambda_W \\ \lambda_{PV} \\ r_{TG} \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \\ 50 \end{bmatrix}$$

Sotto i vincoli: $\|Px\|_2 \leq r_c^T x - D$

$$0 \leq x \leq 1$$

$$P = \begin{bmatrix} \lambda_W & 0 & 0 \\ 0 & \lambda_{PV} & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Problema in formulazione standard SOCP

Minimizzare: $c^T x$

Sotto i vincoli: $\|A_c x + b_c\|_2 \leq d_c x - \gamma_c$

$$0 \leq x \leq 1$$

$$A_c = P$$

$$b_c = 0$$

$$d_c = r_c^T$$

$$\gamma_c = D$$

Script: ConeProgrammingDispatch24h.m

```
clear all
close all
clc

lambdaW=10; % metà della taglia nominale
lambdaPV=5; % metà della taglia nominale
rTG=50;      % taglia nominale

% costi marginali di produzione
c1=1;
c2=1;
c3=25;

P=diag([lambdaW, lambdaPV, 0]);
rc=[lambdaW lambdaPV rTG];

A=P;
b = zeros(3,1);
d = rc';
```

Dati del
problema

Ciclo for che risolve il problema SOCP per ogni valore della richiesta oraria di potenza D

```
for D=[40 42 44 48 50 51 52 52 51 50 49 49 51 49 48 47 46 44 40 30 35 36 38 37]
gamma=D;

socConstraints = secondordercone(A,b,d,gamma);

c = [c1,c2,c3];
Aineq = [];
bineq = [];
Aeq = [];
beq = [];
lb = [0,0,0];
ub = [1,1,1];

[x(:,i),fval] = coneprog(c,socConstraints,Aineq,bineq,Aeq,beq,lb,ub);

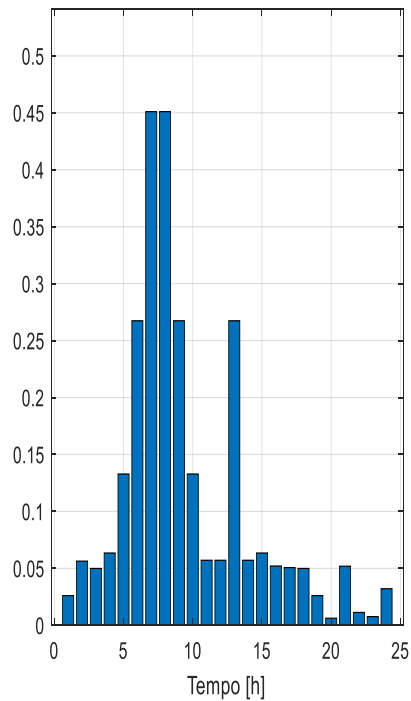
i=i+1;
end
```

Definizione del second order cone constraint

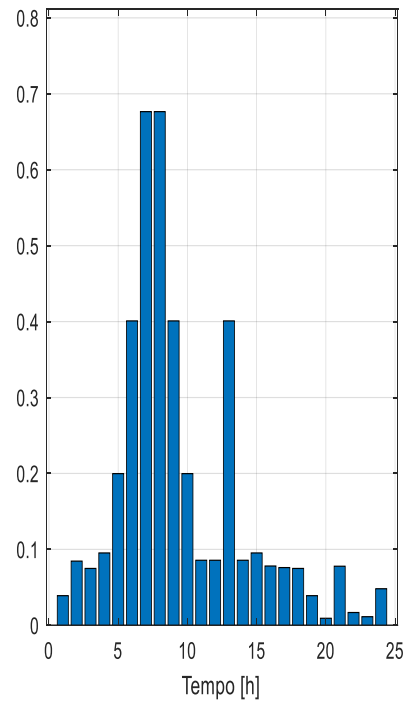
Funzione matlab coneprog.m per risolvere problemi SOCP

Risultati

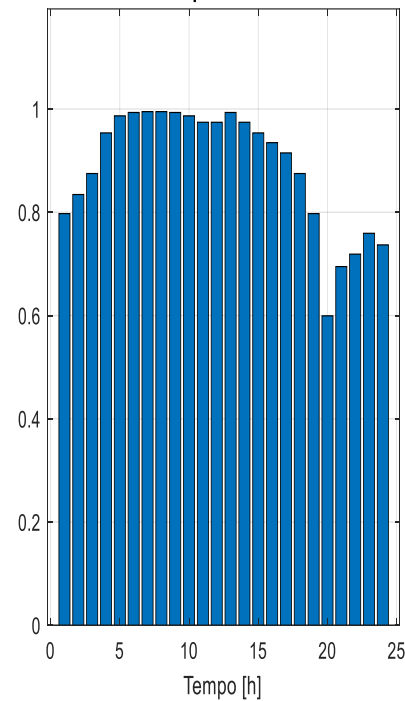
Parco eolico



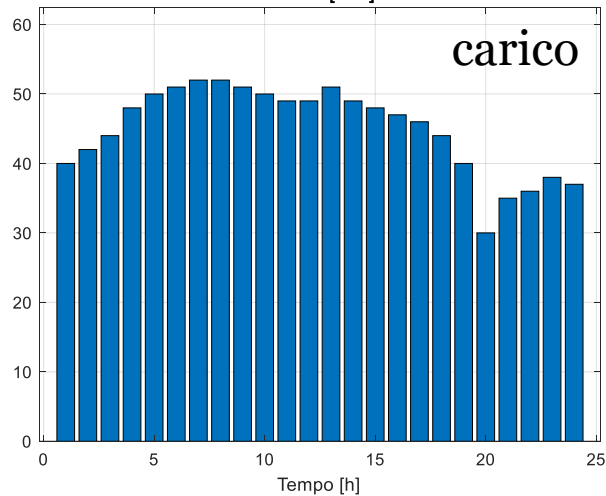
Parco PV



Impianto TG



Carico [MW]

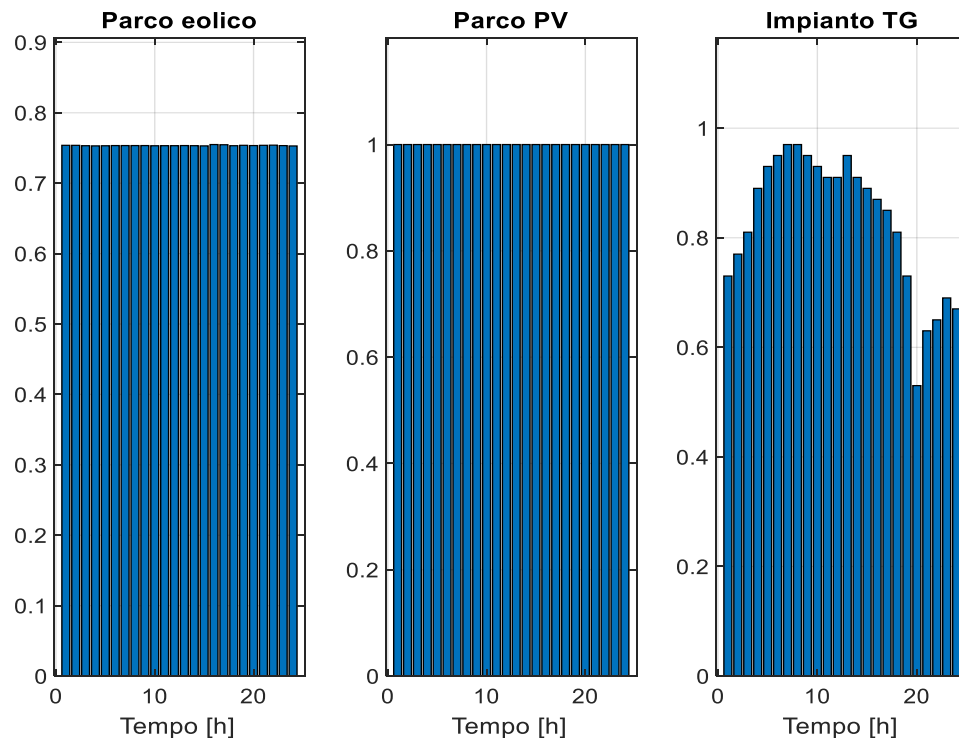


I generatori da FER vengono impiegati in modo più consistente nelle fasce orarie in cui si hanno le massime richieste di carico.

Risultati

Incrementiamo da 25 € a 30 € il costo marginale di produzione per MW dell'impianto TG.

$$c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} c_W \\ c_{PV} \\ c_{TG} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 30 \end{bmatrix}$$



L'incremento dei costi marginali del generatore TG spinge l'algoritmo verso un uso maggiore dei generatori da FER

L'impianto TG lavora sempre al di sotto della sua taglia nominale, ed il livello di potenza è pianificato in modo da garantire il soddisfacimento della domanda anche a fronte di fluttuazioni imprevedibili della produzione da FER

Alcuni ambienti di ottimizzazione



<https://www.gurobi.com/>



<http://cvxr.com/cvx/>



<https://yalmip.github.io/>



<https://www.mosek.com/>