

# 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.



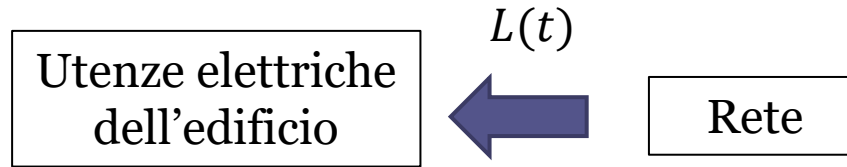
L'impiego di un sistema di storage a batteria può consentire significativi risparmi sui costi energetici

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



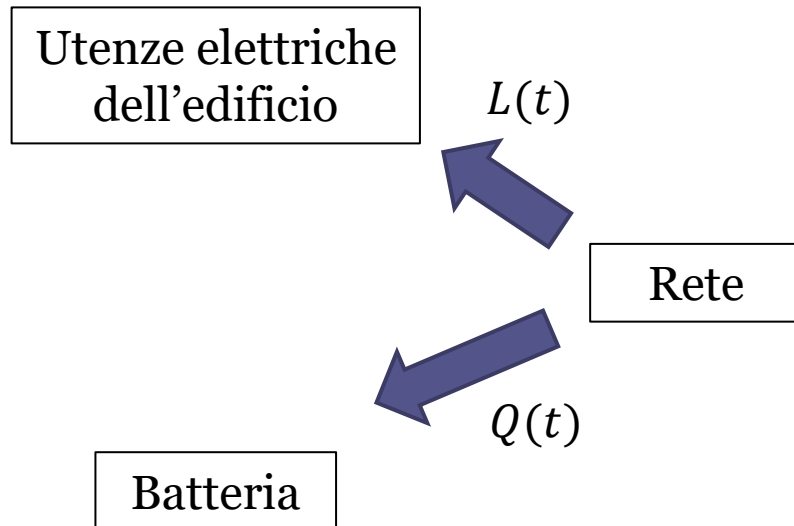
BATTERY  STORAGE

## Senza batteria

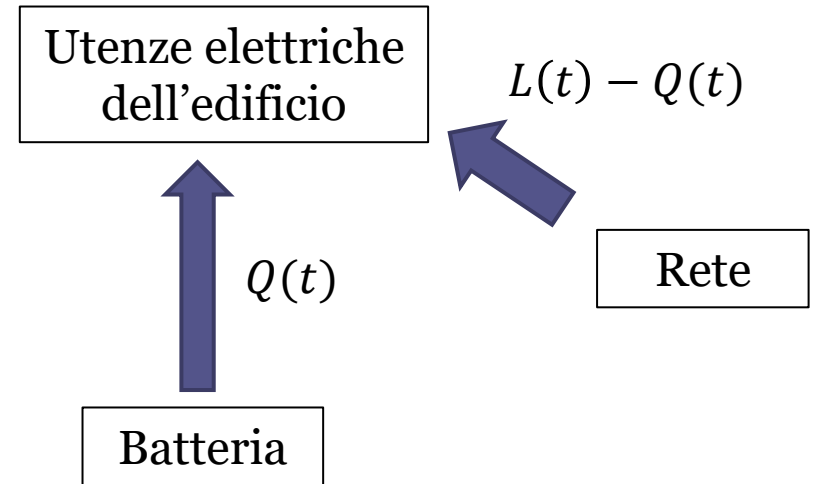


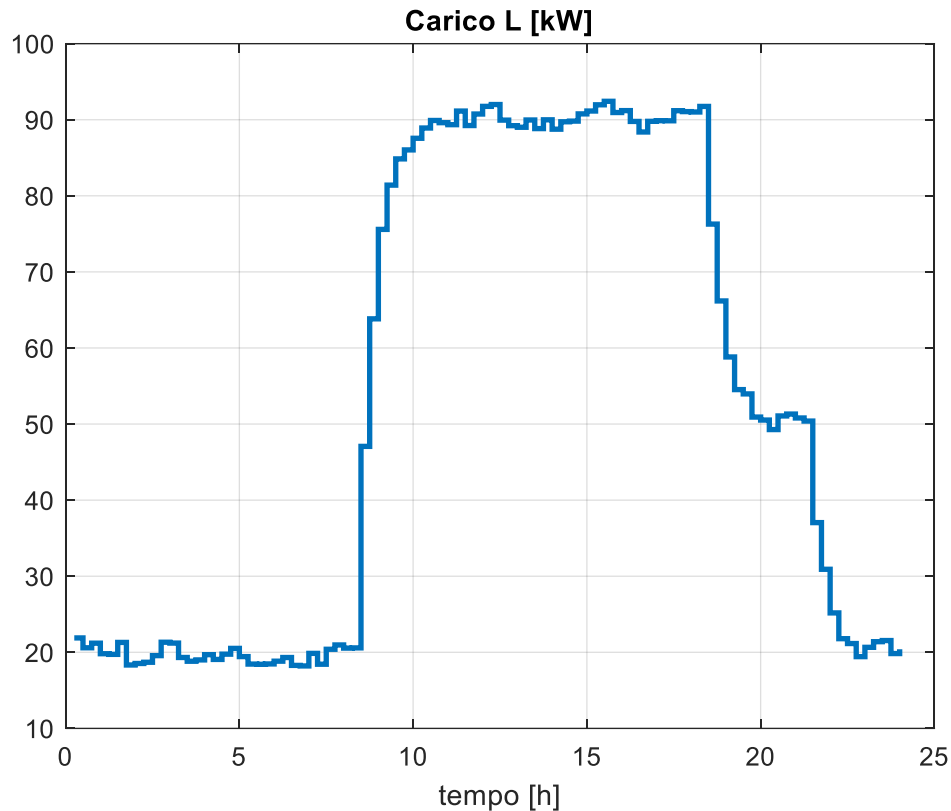
## Con batteria

### Fase di carica [ $Q > 0$ ]



### Fase di scarica [ $Q < 0$ ]



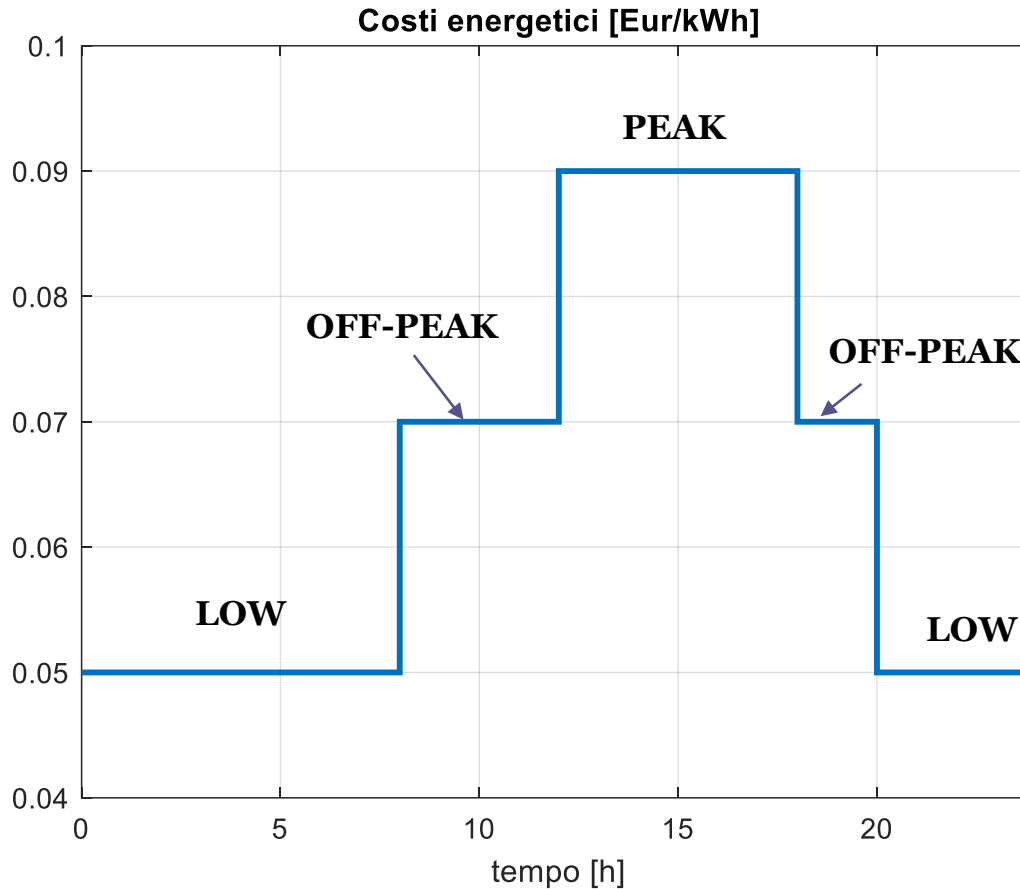


Profilo di carico dell'edificio  
sulle 24 ore

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  $k - \text{esimo}$  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

$$-Q_{min} \leq Q(k) \leq Q_{Max} \quad \begin{array}{l} Q_{min} = 50 \text{ kW} \\ Q_{Max} = 50 \text{ kW} \end{array}$$

Vincoli sulla rapidità di carica/scarica della batteria

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

La potenza prelevata istantaneamente dalla rete è non negativa (non è consentita l'immissione di energia nella rete)

$$\begin{array}{l} S(k) = S(k-1) + T_s Q(k) \\ S(0) = S_0 \quad S_0 = 0 \text{ kWh} \end{array}$$

Dinamica dello state of charge della batteria

$$\begin{array}{l} S_{min} \leq S(k) \leq S_{Max} \\ S_{min} = 0 \text{ kWh} \\ S_{max} = 390 \text{ kWh} \end{array}$$

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:

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

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

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

$$S(0) = S_0$$

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

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

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

### Ottimizzazione dinamica:

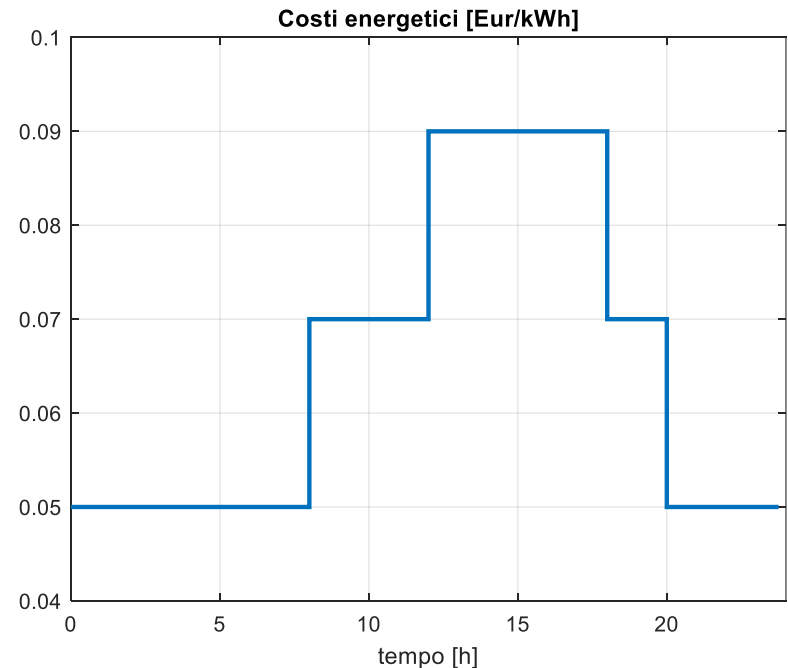
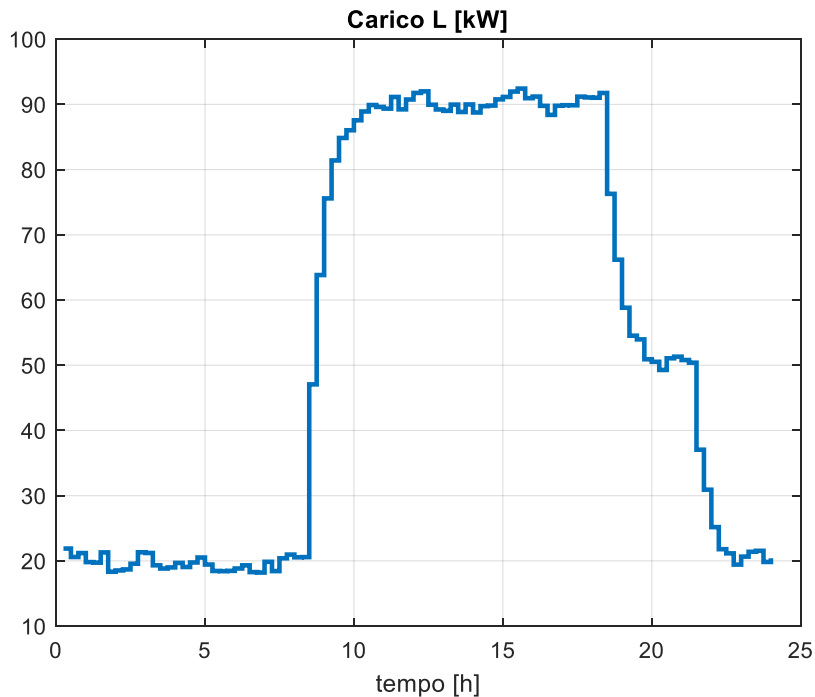
all'interno dei vincoli compaiono delle variabili addizionali  $S(k)$  governate da un sistema dinamico a tempo discreto del quale le variabili decisionali costituiscono gli ingressi.

# Codice Matlab

**File:** BatteryManagementBuilding.m

```
clear all, close all, clc
N=96; % numero slot temporali di durata 15 minuti

% questo script genera e plotta i vettori C ed L
BatteryManagementBuilding_CreaCL
```



# 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

**NB:** S non è propriamente una variabile di ottimizzazione, in quanto la sua evoluzione è interamente definita da quella di Q, ma onde poter imporre dei **vincoli** sulla evoluzione temporale di S è necessario definirla come tale

```
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);
```

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

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

## No immissione di energia nella rete

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

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

## Vincoli su state of charge finale della batteria

```
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

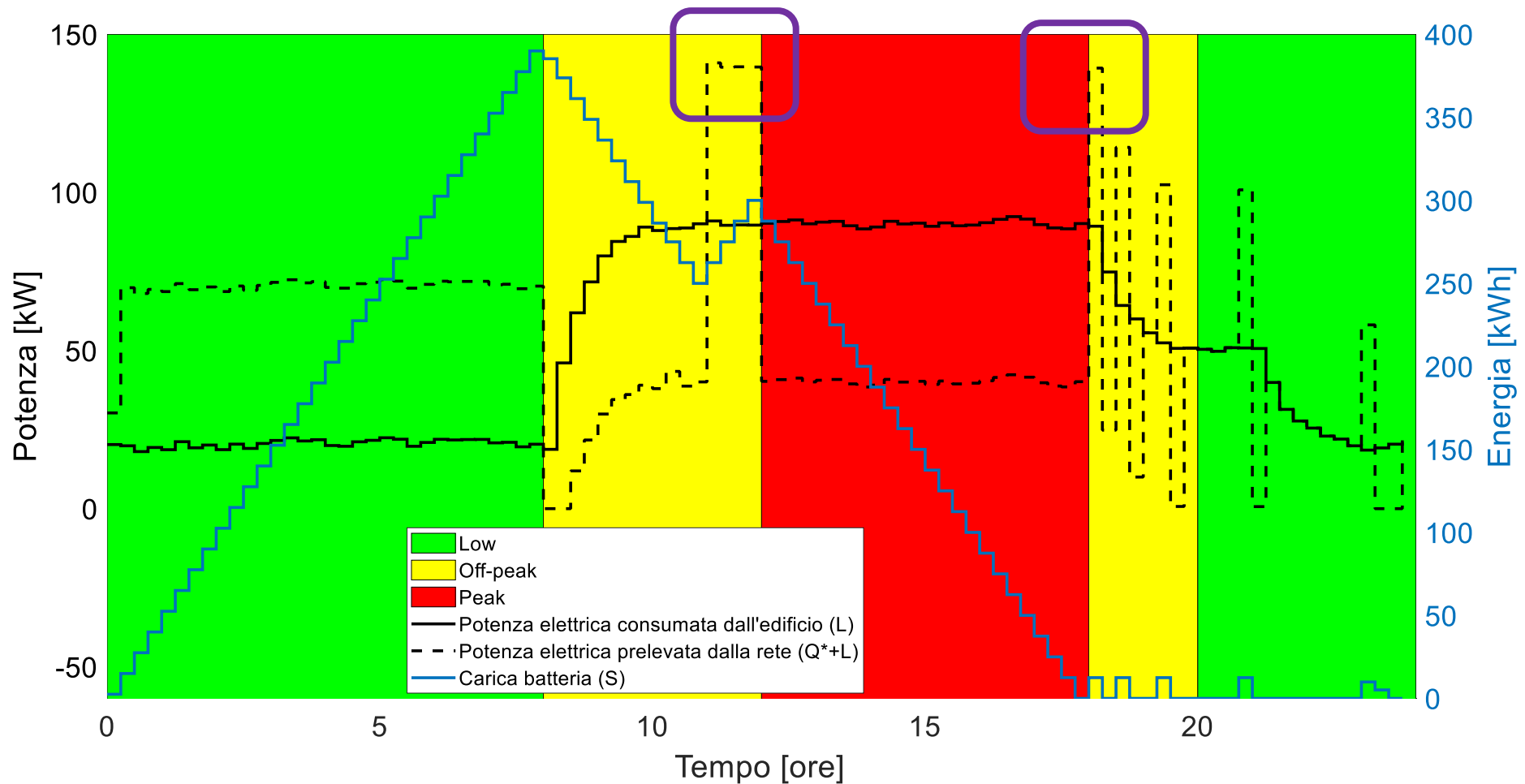
```
Qstar=optbatterySOL.Q;
```

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

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

# Risultati

Picco di circa 140 kW nella  
potenza assorbita dalla rete



```
Solving problem using linprog.
```

```
Optimal solution found.
```

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

```
mincost =
```

```
79.8357
```

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

```
spesenobatt =
```

```
93.6357
```

```
risparmio_percentuale =
```

```
14.7380
```

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.

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

$$= T_s [L + Q]^T + c_{maxpow} \max[L + Q]$$

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 <=S(N);
ZNconshigh = S(N)<=1.05*S0 ;
maxpowconstr = L+Q<=Pmax;

optbattery=optimproblem;
optbattery.Objective = obj;

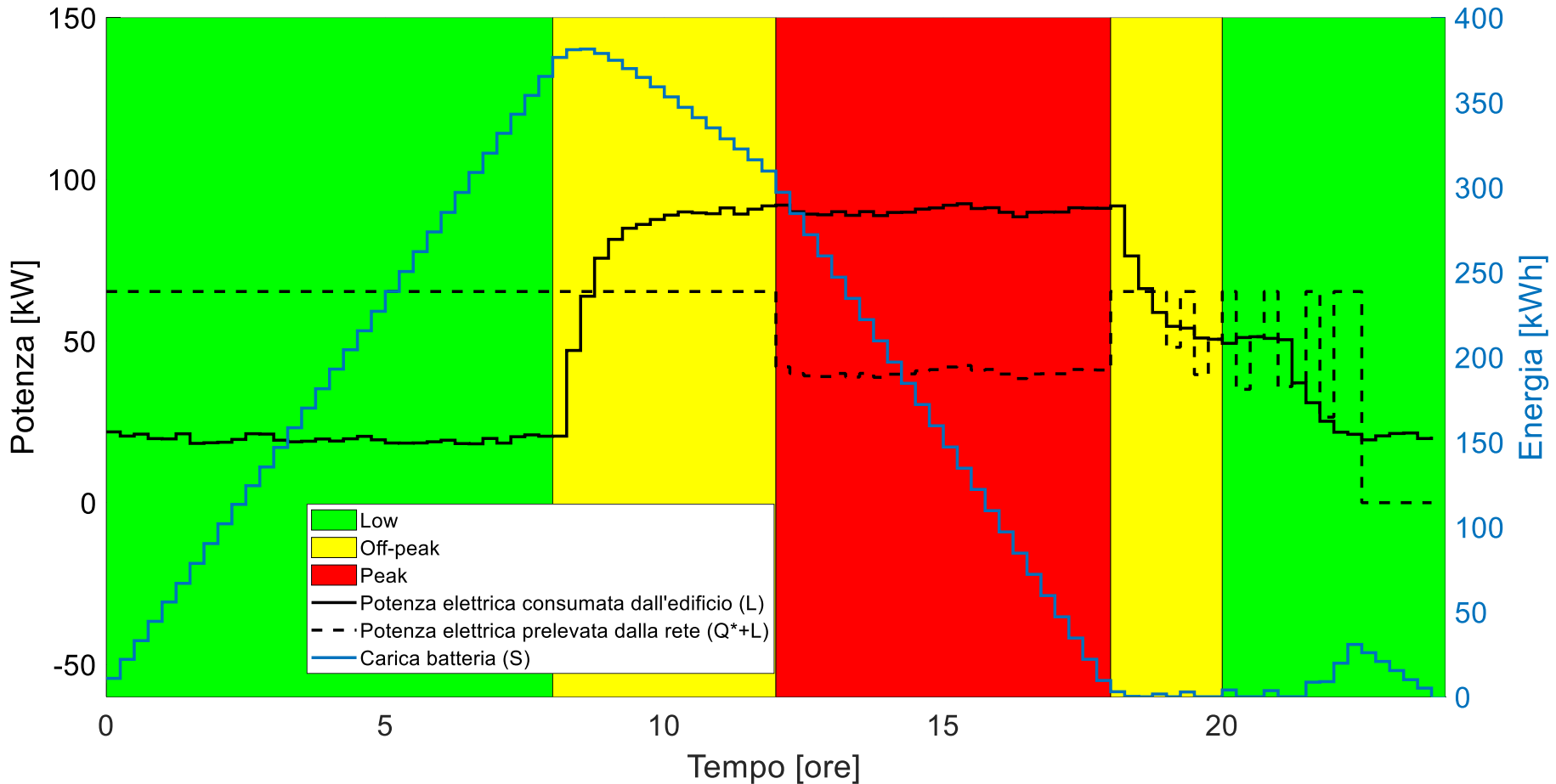
optbattery.Constraints.zcons1 = zcons1;
optbattery.Constraints.zcons2 = zcons2;
optbattery.Constraints.LplusQcons = LplusQcons;
optbattery.Constraints.ZNconslow = ZNconslow;
optbattery.Constraints.ZNconshigh = ZNconshigh;
optbattery.Constraints.maxpowconstr = maxpowconstr;

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

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

```

# Risultati



Si focalizzi l'attenzione sulla curva  $L+Q^*$  (potenza elettrica prelevata dalla rete), 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 =
```

```
100.5725
```

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

```
spesenobatt =
```

```
121.6358
```

```
risparmio_percentuale =
```

```
17.3167
```

L'impiego ottimizzato della batteria ha consentito un risparmio su base giornaliera pari al **17%** circa anche includendo nel profilo tariffario il demand rate proporzionale al picco di potenza assorbito dalla rete

## Spunti di approfondimento

Il problema potrebbe essere approfondito in diverse direzioni

Sia la capacità  $S_{Max}$  della batteria che il massimo rate di carica e scarica  $Q_{Max}$  (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  $N_g$  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**

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

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  $p_i(k)$  ( $i = 1, \dots, N_g$ ,  $k = 1, 2, \dots, 24$ ). 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°	$P_i^{min}$	$P_i^{max}$	$a_i$	$b_i$	$\rho_i$
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°	$\zeta_i^{up}$	$\zeta_i^{down}$
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

$D(k)$   
 $1 \leq k \leq 24$

Time(h)	Load(MW)	Time(h)	Load(MW)
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

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

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

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 Ng = 6 righe ed 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`

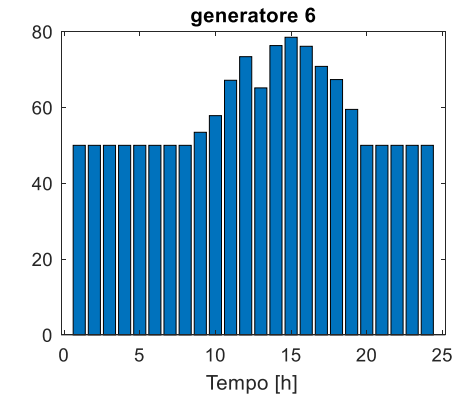
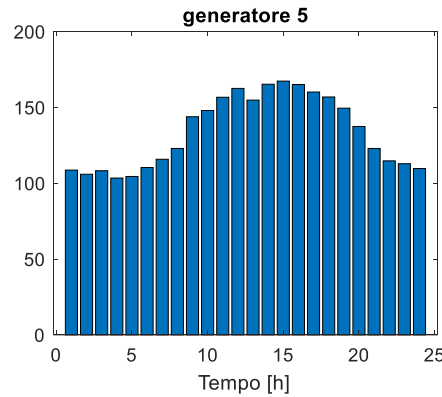
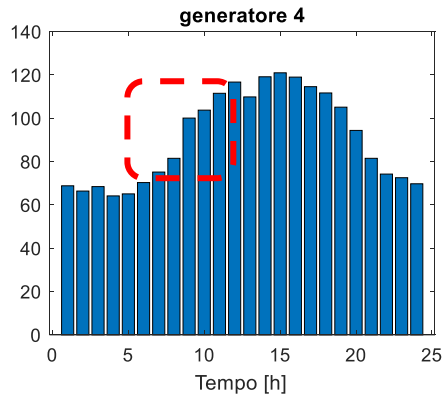
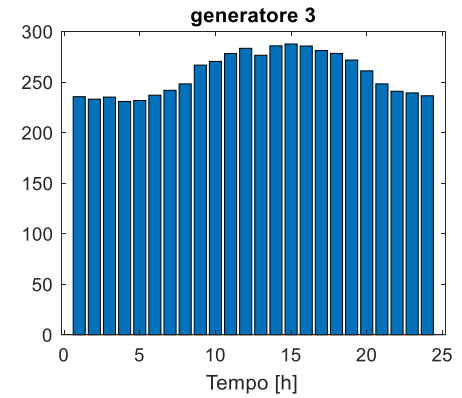
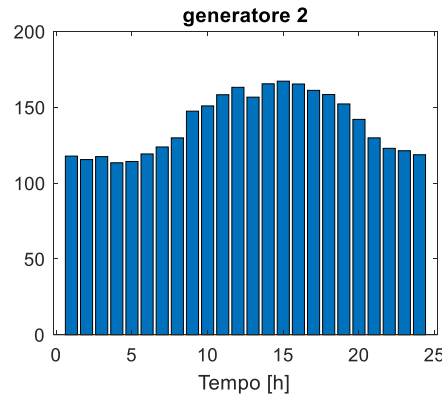
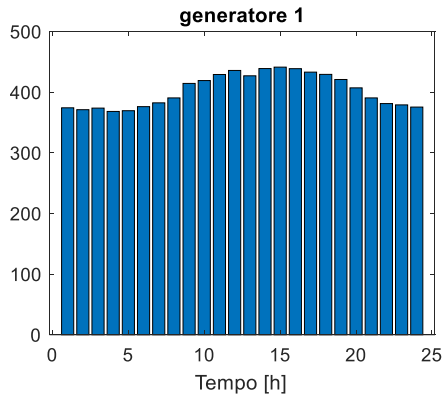
Solving problem using quadprog.

Minimum found that satisfies the constraints.

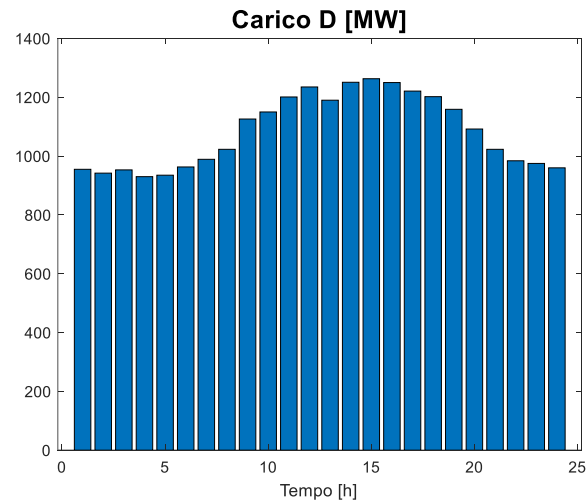
Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied to within the value of the constraint tolerance.

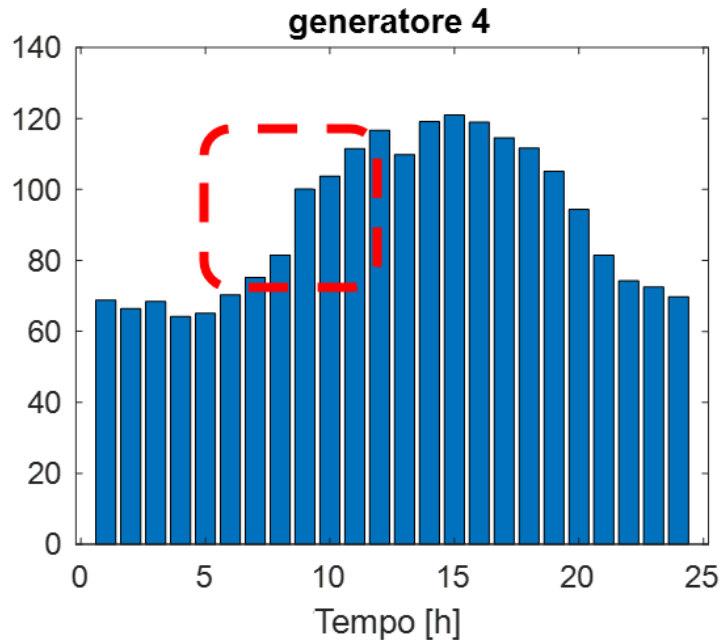
<stopping criteria details>

|



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  $\zeta_4^{up}$

Valore  
precedentemente  
utilizzato

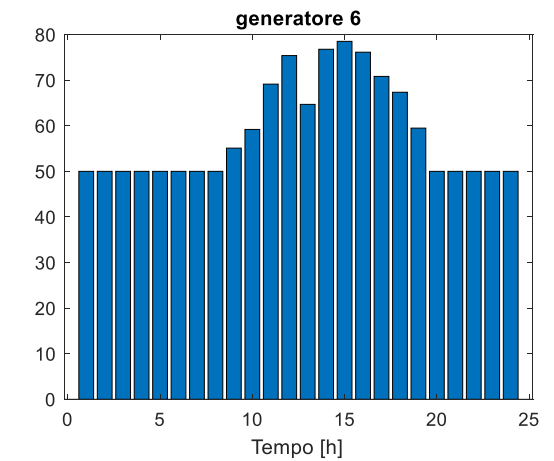
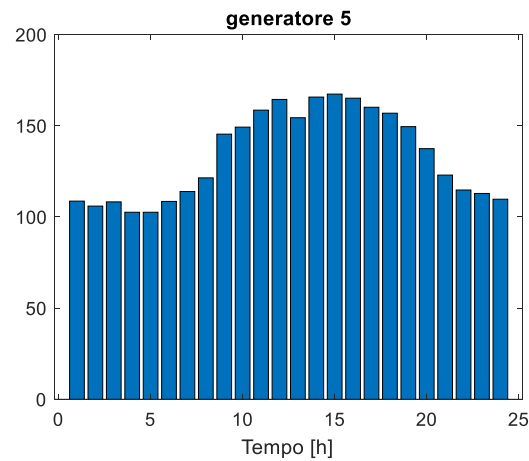
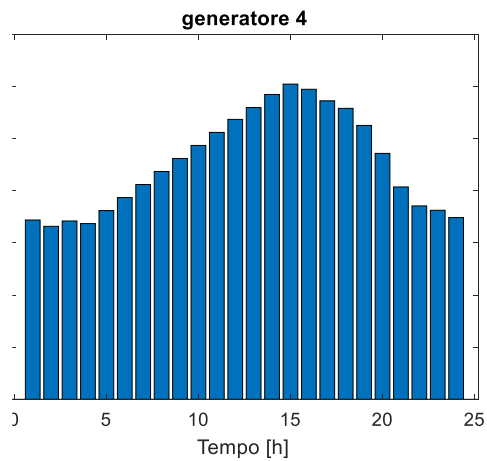
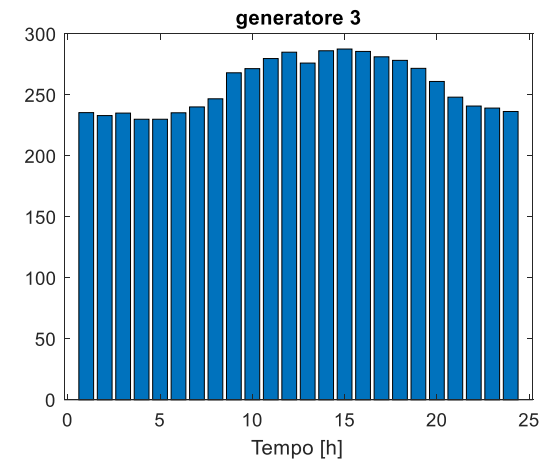
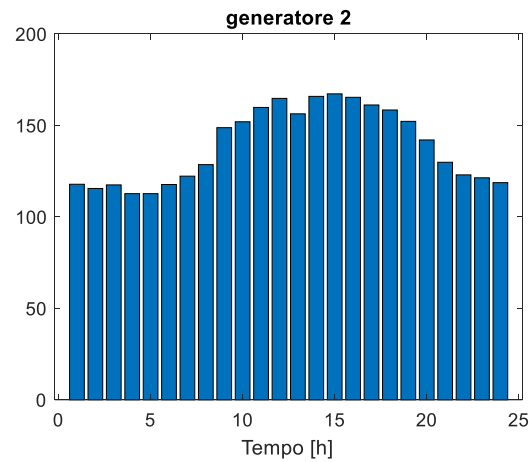
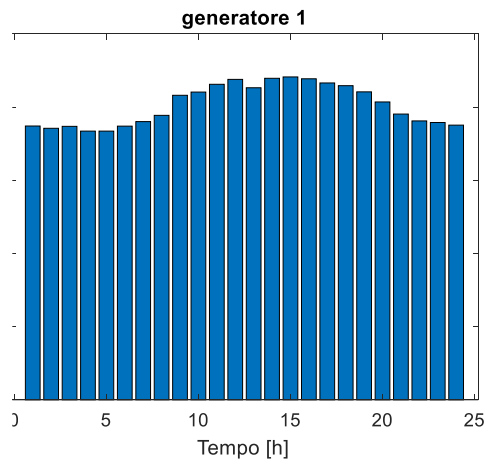
$$\zeta_4^{up} = 50.$$

Nuovo valore

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

# Risultati

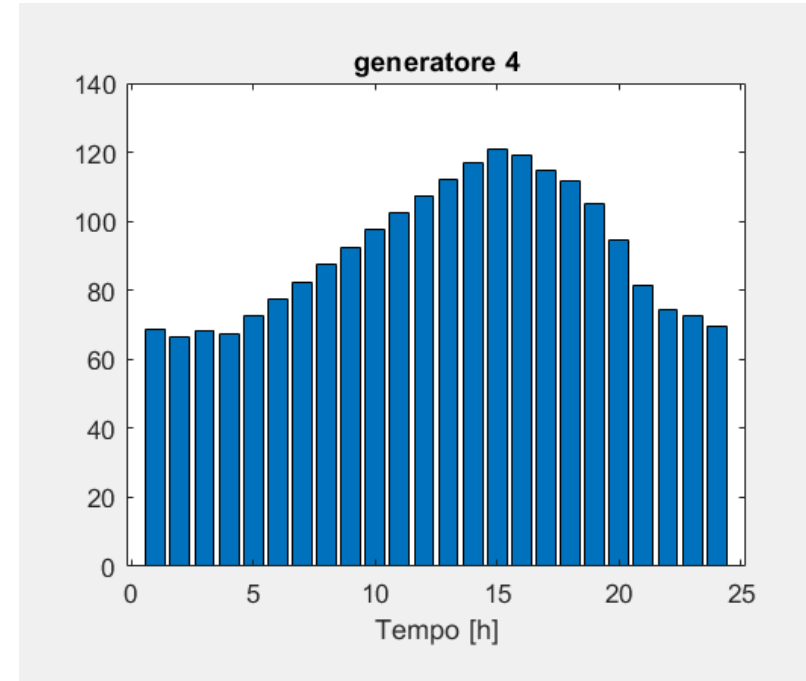
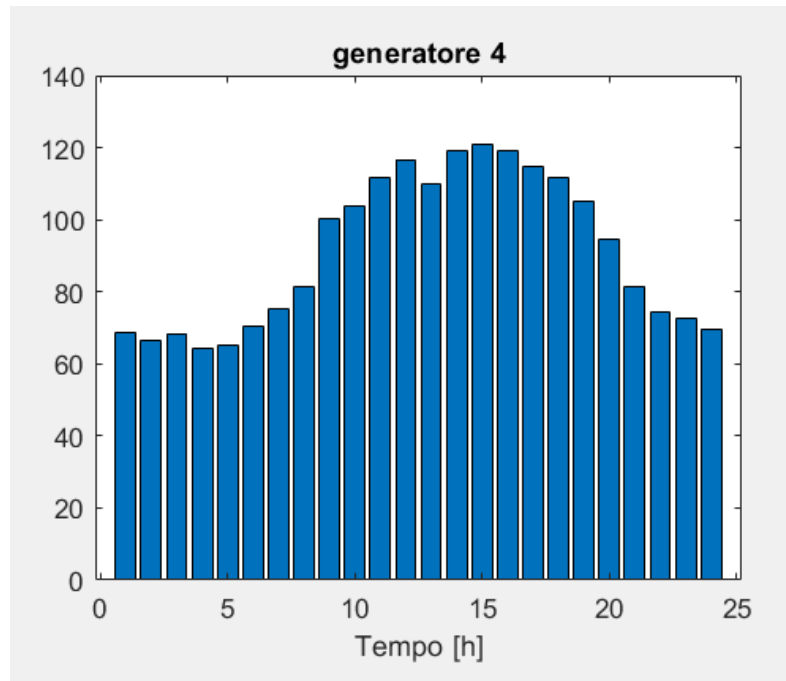
$$\zeta_4^{up} = 5$$



## 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

	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]					

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

## 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

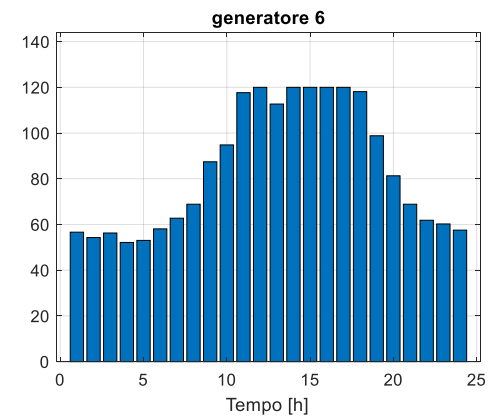
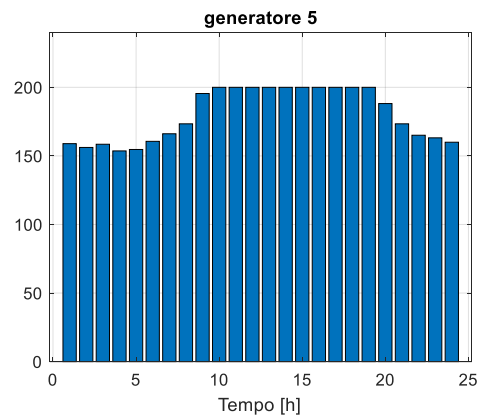
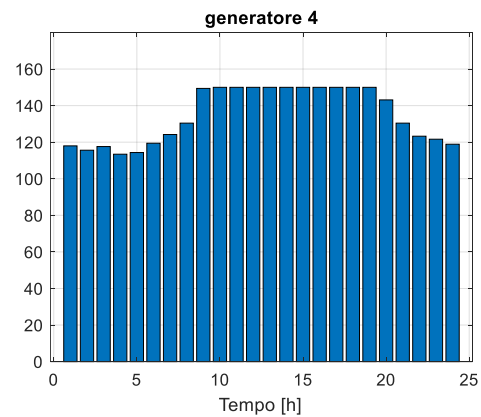
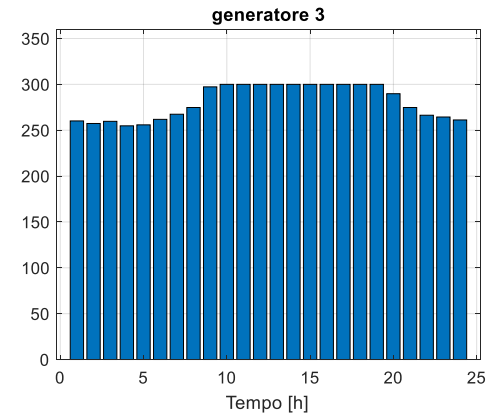
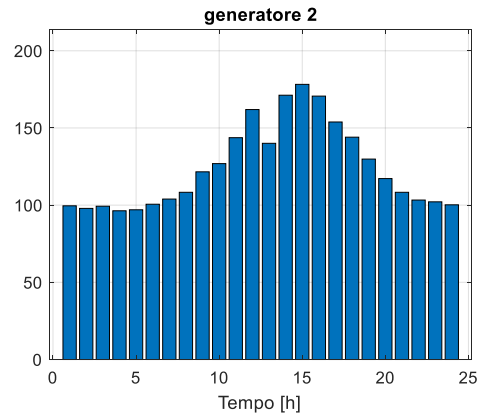
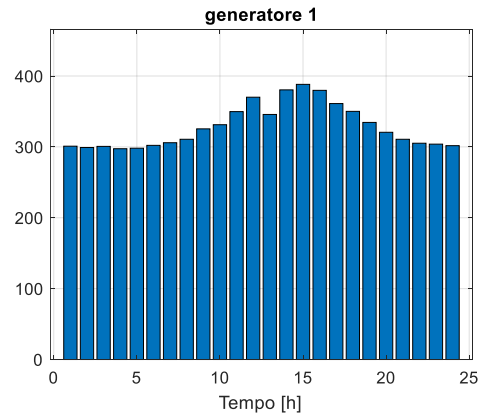
```
QPdispatchsol=solve(QPdispatch, p0, 'options', options);
```

 Risoluzione

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

La soluzione ottima viene memorizzata nella  
variabile Pstar

# Risultati



## Alcuni ambienti di ottimizzazione



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



**Gurobi: Always Free for Academics**

We make it easy for students, faculty, and researchers to work with mathematical optimization.



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



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



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