Agenzia nazionale per le nuove tecnologie,
l'energia e lo sviluppo economico sostenibile

MINISTERO DELLO SVILUPPO ECONOMICO

Ricerca di Sistema elettrico

# Ottimizzazione "*day-ahead*" dei carichi energetici stagionali di uno Smart Building in uno scenario di dynamic pricing

Dario Masucci

ROMA TRE
UNIVERSITÀ DEGLI STUDI

Report RdS/PAR2018/118

# OTTIMIZZAZIONE “DAY-AHEAD” DEI CARICHI ENERGETICI STAGIONALI DI UNO SMART BUILDING IN UNO SCENARIO DI DYNAMIC PRICING

D. Masucci (Università Roma Tre)

Dicembre 2018

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA
Piano Annuale di Realizzazione 2017
Area: Efficienza energetica e risparmio di energia negli usi finali elettrici e interazione con altri vettori energetici
Progetto: D.1 Tecnologie per costruire gli edifici del futuro
Responsabile del Progetto: Giovanni Puglisi, ENEA

Il presente documento descrive le attività di ricerca svolte all’interno dell’Accordo di collaborazione “Algoritmi mirati all’ottimizzazione dei carichi energetici di uno smart building in uno scenario di dynamic pricing”
Responsabile scientifico ENEA: Stefano Pizzuti
Responsabile scientifico Università Roma Tre: prof. Stefano Panzieri

# Indice



# Indice delle figure

# 1 Introduzione

La ricerca condotta nell'ambito del presente accordo di collaborazione tra ENEA e Università Roma Tre ha proseguito alcuni studi iniziati nelle precedenti annualità inerenti la gestione energetica di edifici terziari.
In particolare, la linea di attività ripresa e ulteriormente sviluppata riguarda l'Ottimizzazione "day-ahead" dei carichi energetici in uno Smart Building applicata ad uno scenario di *dynamic pricing*.
Attualmente il consumo di energia necessario al raggiungimento del comfort termico degli ambienti e dell'acqua sanitaria in un edificio rappresenta circa il 30% del consumo energetico nazionale ed è responsabile del 25% delle emissioni di anidride carbonica.
Risulta quindi cruciale intervenire nei processi di gestione delle risorse considerando non solo l'aspetto etico-sociale legato all'impatto ambientale dovuto all'utilizzo dei combustibili fossili per soddisfare il fabbisogno energetico.
Ma è importante altresì sottolineare il risparmio economico derivante dall'utilizzo razionale delle risorse. Basti pensare che il costo annuale della bolletta energetica rappresenta oggi una delle voci più rilevanti del bilancio familiare.
In base a quanto detto, l'attività di ricerca e sviluppo descritta in questo documento ha l'obiettivo di definire e implementare uno strumento automatico per l'ottimizzazione delle prestazioni energetiche dei uno Smart Building.
Quindi, attraverso una procedura automatizzata, si vuole simulare il comportamento termico di un edificio nell'arco dell'intera stagione estiva e attuare una strategia di ottimizzazione, applicabile a edifici intelligenti, mirata alla riduzione del consumo energetico e alla minimizzazione della percentuale di occupanti insoddisfatti a causa del "discomfort" termico.
Partendo da uno strumento di simulazione realizzato durante le passate annualità, sono state identificate e realizzate le opportune modifiche da apportare per ottenere le funzionalità richieste.
La realizzazione di questa funzionalità permette di ridurre sensibilmente i consumi termici dell'edificio, mantenendo un accettabile livello di comfort termico per gli occupanti intervenendo sulle temperature di set point, di valvole termostatiche e mandata, con cui è attualmente controllato il sistema di climatizzazione elettrica dell'edificio.

# 2 Descrizione dell'attività svolta

La attività descritta di seguito è volte alla realizzazione di uno strumento per la pianificazione dei flussi energetici gestiti dal sistema centrale di supervisione e controllo attivo nell'edificio F40, sito all'interno del Centro Ricerche ENEA "La Casaccia".

## *2.1 Contesto*

Partendo dal lavoro svolto nelle precedenti annualità (Report RdS/PAR2015/158 e Report RdS/PAR2016/…), è stato possibile definire una strategia previsionale per la riduzione del consumo di energia associabile al processo di climatizzazione di un edificio.
La strategia prevede l'utilizzo di un simulatore in ambiente MATLAB/Simulink in grado di predire, in base alle condizioni esterne fornite in automatico attraverso un file meteo (temperatura, umidità, velocità del vento, ecc…), il comportamento dell'edificio in termini di consumi e di condizioni termo-igrometriche con un giorno di anticipo. Il sistema di supervisione attivo nell'F40 mette a disposizione funzionalità di controllo sulla temperatura di set point delle valvole termostatiche e di mandata che rappresentano i parametri su cui poter agire per ottimizzare i consumi.
Infatti, l'edificio viene riscaldato mediante radiatori di dimensioni molto simili tra loro e lo scambio di calore con l'ambiente avviene per irraggiamento e convezione naturale. I consumi termici dipendono sia dalla temperatura dell'acqua di mandata, che proviene dalla centrale termica e raggiunge l'edificio, sia dalle temperature a cui vengono impostate le valvole termostatiche presenti su ogni dispositivo di climatizzazione

Oltre a ridurre i consumi termici, l'obiettivo principale è di mantenere il comfort degli occupanti sempre ad un livello accettabile. Vengono quindi valutate le percentuali di insoddisfatti che occupano lo stabile tramite l'indice di PPD (Predicted Percentage of Dissatisfied) presente nella norma vigente UNI EN ISO 7730.
La potenza di elaborazione del simulatore viene sfruttata da un sistema esperto implementato in MATLAB che, tramite una tecnica di ottimizzazione basata su algoritmi genetici, è in grado di formulare la strategia ottima da applicare.
Grazie ai risultati ottenuti nelle precedenti annualità si ha a disposizione uno strumento in grado di elaborare la combinazione ottima di set point di temperatura per le valvole termostatiche dei caloriferi e per l'acqua di mandata per ogni ora di un predeterminato giorno.

## 2.2 Metodologie e Obiettivi

L'obiettivo di questa progettualità è quindi quello di definire una strategia previsionale che possa tenere in considerazione il consumo sull'intera stagione estiva (90 giorni) e ampliare i risultati ottenuti nelle annualità precedenti. In particolare, la stagione estiva che si vuole simulare va dal 21 giugno al 20 settembre.
Prima di procedere alla definizione della strategia risolutiva, è stata effettuata un'approfondita analisi del codice riguardante l'ottimizzatore e il simulatore, a seguito della quale è stato possibile definire le caratteristiche principali dell'attuale strumento.

### 2.2.1 L'ottimizzatore

Per individuare la combinazione ottima di set-point viene utilizzato l'algoritmo evolutivo NSGA-II (Non-dominated Sorting Genetic Algorithm). Questo è un metodo di ottimizzazione multi-obiettivo in questione che basandosi sul principio di dominanza di Pareto, restituisce l'insieme di tutte le soluzioni non-dominate appartenenti al problema e individua la soluzione che realizza il miglior compromesso tra i due obiettivi da perseguire.

Viene quindi individuato il fronte di Pareto in cui ogni punto rappresenta una differente temperatura della valvola termostatica considerata, e a cui corrispondono due importanti informazioni:

- Valore di consumo energetico orario;
- Valore di indice PPD orario.

Quindi la scelta di uno solo tra i punti del fronte è dettata dalla necessità di rispettare la normativa UNI EN ISO 7730 sul microclima in ambienti di lavoro secondo cui la percentuale di insoddisfatti non superi il 10% del personale presente nell'edificio. Perciò viene scelto il punto ottimo del fronte di Pareto per cui vale l'uguaglianza che segue:

$$PPD_{opt} = \max(PPD < 10)$$

In questo modo viene massimizzato l'indice di comfort, rimanendo sempre entro la soglia prevista, ma contestualmente viene minimizzato il consumo di energia necessaria per la climatizzazione elettrica dell'edificio.
Nel caso in esame, il consumo è inteso in termini di euro, sulla base del PUN (Prezzo Unico Nazionale), ottenuti secondo una precisa elaborazione effettuata all'interno del simulatore.

### 2.2.2 Il simulatore

Il comportamento termico dell'edificio F40 è modellato tramite il software open source HAMbase utilizzabile in ambiente di programmazione Matlab/Simulink. HAMbase, sviluppato dai ricercatori dell'università di Eindhoven, permette di simulare il comportamento di un edificio sfruttando le informazioni riguardanti la temperatura interna, l'umidità dell'aria e il consumo di energia necessario per la climatizzazione dell'edificio.
Schematizzando i parametri fondamentali della simulazione si ha:

- Parametri in input:
  - Set-point delle valvole termostatiche;
  - Set-point temperatura di mandata dell'acqua;
  - Condizioni meteo esterne (lette da un file meteo);

- Parametri in output:
  - Consumo termico;
  - Temperatura e umidità relativa.

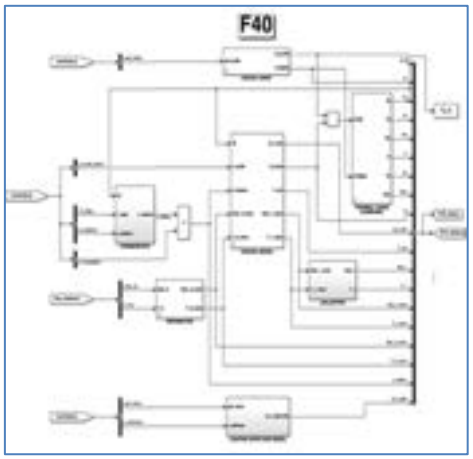


**Figura 1 - Snapshot Simulatore Simulink Edificio F40**

Le simulazioni condotte riguardano il giorno 21 giugno, per il quale sono disponibili file meteo contenenti le condizioni climatiche esterne necessarie per il corretto funzionamento del modello. In particolare, la simulazione ha riguardato tre differenti temperature interne iniziali
$T0 = 21; 23; 25\ [°C]$ per riprodurre tre diverse situazioni climatiche interne dovute ad un eccessivo raffrescamento o riscaldamento nel giorno precedente. Inoltre, un'apposita funzione, implementata in MATLAB, fornisce il valore orario medio dell'indice PPD dell'intero edificio.

## 2.3 Logica di simulazione proposta

Si è deciso di intervenire in modo poco invasivo sul lavoro prodotto durante le precedenti annualità, e di prolungare il periodo di simulazione, da un giorno a una stagione, agendo internamente al codice con l'aggiunta di cicli iterativi, opportunamente definiti, in grado di riprodurre il comportamento desiderato per i novanta giorni richiesti.
Oltre a questo, è stato notato che le risorse computazionali, in termini di memoria fissa e di RAM, che richiederebbe l'algoritmo per simulare l'intera stagione risultano molto grandi rivelando in questo modo una potenziale pericolosa inefficienza.
Per realizzare questa funzionalità, e ovviare ai problemi computazionali, sono stati definiti e implementati quattro fondamentali meccanismi elencati di seguito:

- processo iterativo per impostare i giorni e i mesi della simulazione;
- processo per il salvataggio delle informazioni di interesse al termine di ogni giorno;
- processo per il salvataggio delle informazioni necessarie per lo start-up della simulazione successiva;
- processo di pulizia del workspace per consentire lo start-up della simulazione successiva.

Per ridurre ulteriormente il carico computazionale, e quindi il tempo di esecuzione dell'intera simulazione è stato effettuato uno studio dell'ottimizzatore per individuare i parametri della simulazione su cui agire, senza però intervenire su quelli riguardanti la configurazione dell'algoritmo genetico.

### 2.3.1 Procedura di simulazione stagionale

La logica di simulazione elaborata può essere facilmente ricondotta ad una struttura ibrida, che presenta aspetti sia iterativi che sequenziali. Infatti, è immediato notare che ogni processo sopracitato è implementato tramite un blocco di istruzioni sequenziali che viene richiamato iterativamente.

L'idea, quindi, è di modificare opportunamente il codice esistente e ampliare la struttura di simulazione già elaborata che risulta essere perfettamente funzionante è stato possibile implementare uno strumento in grado realizzare gli obiettivi richiesti.
Per ottenere che la simulazione compia un lasso temporale pari ad una stagione si è provato ad agire sul codice modificando il metodo *simopt.BAS.Period* all'interno del file *mm_getsimoptions.m*.
Il metodo prevede in ingresso un array di quattro elementi $[a, b, c, d]$ in cui:

- $a$ = stagione (1 estate, -1 inverno);
- $b$ = mese dell'anno;
- $c$ = giorno del mese;
- $d$ = durata, in giorni, del periodo di simulazione.

E' stato quindi modificato l'array di ingresso in $[1,6,21,2]$ per testare l'efficacia della modifica. Purtroppo, appena termina la simulazione del primo giorno, il processo genera un errore *index out of bound*.

Coerentemente con la prima modifica si è aumentato il periodo di simulazione del modello *simulink* e si è intervenuto sul codice della classe *f40_optimization_simulator2.mat* aggiungendo il giorno 22 al primo ciclo iterativo *for*.

All'interno della classe *f40_optimization_config2.mat* è stata modificata l'assegnazione *simopt.BAS.Period=[1, m, n, 2]*.
Coerentemente sono state apportate le seguenti correzioni:

- Classe *f40_optimization_simulator2.m*
  - Linea 27 → `for n = 21:22,`
- Classe *f40_optimization_config2.m*
  - Linea 183 → `simopt.BAS.Period=[1, 6, 21, 2];`
  - Linea 185 → `Prob.nsga2_simulator.parameters.simulationHours = 48;`
- Classe *mm_getoptions.m*
  - Linea 7 → `simopt.BAS.Period=[1, 6, 21, 2];`
  - Linea 10 → `simopt.Ta_start=47*ones(1,15);`
- Classe *Z15_conf_ver14_4_0.m*
  - Linea 141 → `BAS.Period = [1, 6, 21, 2];`

Ma anche in questo caso veniva generato lo stesso errore, ciò significa che durante l'elaborazione vengono definite delle strutture di dati fisse e non parametriche rispetto ai dati di ingresso.
Poiché tutti i tentativi di prolungare la simulazione agendo internamente al codice sono risultati vani, si è proceduto considerando la possibilità di ripetere la simulazione giornaliera per i novanta giorni che compongono la stagione estiva. Il tutto agendo esternamente grazie all'aggiunta di cicli iterativi, appropriatamente definiti, nella classe *f40_optimization_simulator2.mat*.
È stato elaborato lo script *f40_optimization_simulator3.m* che aumenta il periodo di simulazione coprendo l'intera stagione estiva, implementando il meccanismo di iterazione automatica dei giorni di simulazione.
E' possibile osservare che iterativamente viene modificato il giorno di riferimento della simulazione richiamando la classe *mm_simoption2.m* e intervenendo sui parametri richiamati dal simulatore tramite il metodo. In questo modo è possibile aggiornare in modo parametrico i giorni da simulare.
Il tempo computazionale che impiegherebbe l'algoritmo per simulare l'intera stagione è molto grande e non permette di testare in tempi brevi la strategia implementata. Per questo è stato elaborato lo script *f40_optimization_simulator4.m* di dimensioni ridotte, ma tramite il quale è possibile simulare un numero a piacere di giorni.

Gli aspetti salienti della logica di simulazione proposta si possono riassumere come segue:

- Periodo di simulazione

  Giugno $\begin{cases} m = 6 \\ n = 21{:}30 \end{cases}$ Luglio $\begin{cases} m = 7 \\ n = 1{:}31 \end{cases}$ Agosto $\begin{cases} m = 8 \\ n = 1{:}31 \end{cases}$ Settembre $\begin{cases} m = 9 \\ n = 1{:}20 \end{cases}$

- Strategia in pseudocodice

```
[blocco di inizializzazione]
for m = [6,7,8,9],
      if m == 6
            for n = 21:30,
                  [blocco di elaborazione singola]
                  for h = 1:23
                        [blocco iterativo di elaborazione]
                  end
                  [blocco di aggiornamento e salvataggio]
            end
      end

      if m == 7
            for n = 1:31,
                  [blocco di elaborazione singola]
                  for h = 1:23
                        [blocco iterativo di elaborazione]
                  end
                  [blocco di aggiornamento e salvataggio]
            end
      end

      if m == 8
            for n = 1:31,
                  [blocco di elaborazione singola]
                  for h = 1:23
                        [blocco iterativo di elaborazione]
                  end
                  [blocco di aggiornamento e salvataggio]
            end
      end

      if m == 9
            for n = 1:20,
                  [blocco di elaborazione singola]
                  for h = 1:23
                        [blocco iterativo di elaborazione]
                  end
                  [blocco di aggiornamento e salvataggio]
            end
      end
end
```

Nell'allegato A è riportato il codice Matlab dei blocchi sopracitati.
In questo modo la struttura dati del simulatore viene rielaborata di giorno in giorno per permettere lo startup della simulazione riguardante il giorno successivo.

### 2.3.2 Complessità computazionale

Dopo un'approfondita analisi del codice riguardante l'ottimizzatore e il simulatore è emerso che per simulare una giornata l'algoritmo impiega circa quattro o cinque ore.
Per ridurre la complessità computazionale, e quindi il tempo di esecuzione dell'intera simulazione, è stato possibile agire modificando i parametri concernenti l'algoritmo genetico. In particolare, sono stati identificati e modificati:

- All'interno di *f40_optimization_config2.mat*
    - *Prob.runs = 1;* linea 23, richiamato in: linea 100, 211, 259
    - *Prob.nsga2.runs = 1;* linea 102
    - *Prob.nsga2_approx.runs = 1;* linea 156
    - *Prob.wsea.runs = 1;* linea 213
    - *Prob.rand.runs = 1;* linea 261

Oltre a questi, sono stati identificati anche:

- All'interno di *f40_optimization_config2.mat*
    - *Prob.nsga.pop = 4;* linea 91
    - *Prob.nsga.gen = 8;* linea 93
    - *Options.PopulationSize = 10;* linea 240
    - *Options.Generations = 5;* linea 241
- All'interno di *nsga_cofig.mat*
    - *Prob.pop = 10;* linea 8
    - *Prob.gen = 6;* linea 10

Per evitare di modificare l'evoluzione dell'algoritmo di ottimizzazione non sono stati modificati i parametri riguardanti la popolazione e il numero di generazioni ma solo i parametri riguardanti il numero di run.

Durante la fase di test dello strumento elaborato sono state simulate diverse situazioni:

- Due giorni consecutivi;
- Due giorni non consecutivi;
- Un intero mese;
- Tre mesi consecutivi;
- L'intera stagione estiva;

E si è tenuto traccia:

- Del comportamento dell'ottimizzatore;
- Del tempo di simulazione e delle risorse computazionali impiegate;
- Dei dati e delle figure salvate;
- Dei risultati ottenuti.

E'possibile notare che si è ridotto il tempo computazionale circa del 15-20%, passando quindi da quattro ore e mezza a tre ore e mezza, mantenendo pressoché costanti le risorse computazionali impiegate, a meno di qualche picco che risulta comunque molto limitato in ampiezza. L'allegato B riporta in modo tabellare una parte dei tempi di simulazione registrati durante la fase di test.

### 2.3.3 Output della simulazione

Al termine di ogni giorno di simulazione vengono salvate e quindi rese disponibili le seguenti informazioni:

- Figure
  - Pareto-*mese-giorno*-h*ora*

    in cui sono riportati le soluzioni appartenenti al fronte di Pareto;

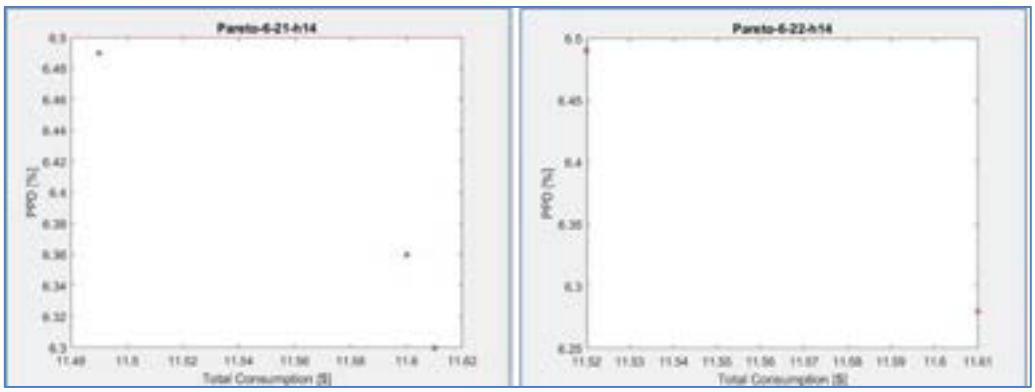


**Figura 2 - punti appartenenti al fronte di Pareto**

  - ConfrontoPPD-Consumo-*mese-giorno*

    che riporta l'andamento giornaliero previsto dei consumi e dell'indice PPD relativo alle sole ore lavorative (8-19).

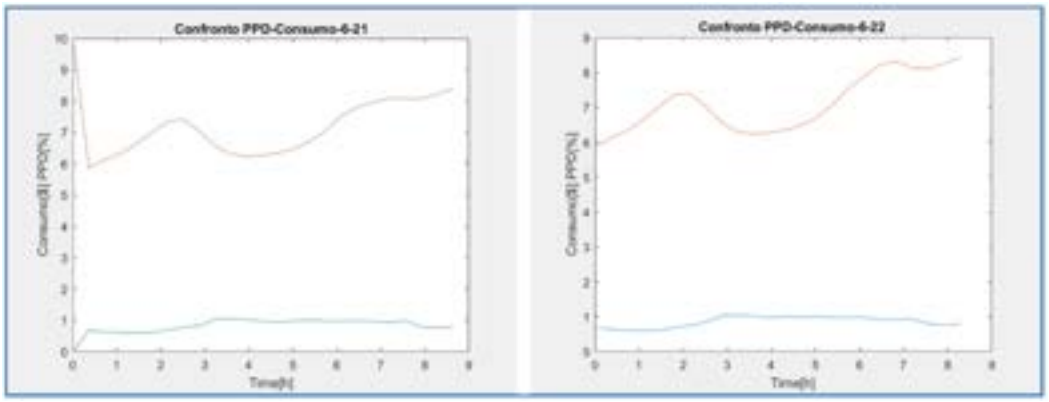


**Figura 3 - Confronto giornaliero tra indice PPD e Consumo**

- Dati
  - Result-*mese-giorno*.mat

    che contiene la struttura *out_tot* in cui è memorizzato il fronte di Pareto orario con i valori assunti dalle funzioni obiettivo e delle variabili decisionali (set point temperatura della stanza, set point temperatura di mandata, consumo in euro, indice PPD, numero di run);
  - ConsumiPPD-*mese-giorno*.mat

contiene due strutture dati: *Y1* in cui sono mantenuti gli indici PPD orari, e *Y2* in cui sono mantenuti i consumi orari;

- Pareto-*mese-giorno*.mat

  che contiene due strutture dati: *out* in cui è mantenuto l'ultimo fronte di Pareto calcolato in termini temporali, e *outStructOpt* in cui sono memorizzate le informazioni della simulazione.

# 3 Conclusioni

L'applicazione definita in questa progettualità rappresenta un tassello all'interno del più grande progetto Smart Buildings Network. L'obiettivo finale è di realizzare una rete di smart buildings, dotati di sistemi sensoriali, i cui parametri vengono comunicati in tempo reale al sistema di supervisione e controllo dello Smart Village presente nel C.R. ENEA "La Casaccia".


**Figura 4 - Smart Village C.R. ENEA "La Casaccia"**

Gli edifici considerati presentano sensori di consumo, elettrico e termico, e sono collegati ad un server dove risiedono programmi di diagnostica remota in grado di segnalare di guasti e malfunzionamenti.
Il sistema di supervisione, quindi, provvede alla diagnostica avanzata ed alla ottimizzazione della gestione. I risultati ottenuti vengono inviati agli attuatori, per implementare le strategie di controllo, e all'energy manager della rete.
Le funzionalità implementate durante questa progettualità risultano particolarmente significative per i processi di monitoraggio e controllo dei flussi elettrici all'interno di un edificio e rappresentano un valore aggiunto tramite il quale ottenere una gestione ottimale delle risorse energetiche impiegate per la sussistenza di uno smart building.
E'stata definita una routine in grado di ottimizzare i consumi elettrici stagionali, dovuti alla climatizzazione, pur mantenendo un livello di comfort termico accettabile. Il periodo di simulazione è stato aumentato fino a coprire l'intera stagione estiva.
Il simulatore implementato è stato testato ripetutamente per verificarne il corretto comportamento. Durante la campagna di test si è tenuto traccia del tempo di simulazione e delle risorse computazionali utilizzate, dei dati e delle figure salvate.
Al termine di ogni giorno di simulazione vengono salvate e quindi rese disponibili le seguenti informazioni:

- le soluzioni appartenenti al fronte di Pareto;
- l'andamento giornaliero previsto dei consumi e dell'indice PPD.

E' auspicabile che in una prossima progettualità i dati ottenuti dalle simulazioni possano essere analizzati per

definire correttamente quali set-point impostare della temperatura di mandata e delle valvole termostatiche. In questo modo sarà possibile definire quali azioni effettuare per ottenere il giusto compromesso tra consumi elettrici impiegati per la climatizzazione e il mantenimento di un livello accettabile di confort all'interno dell'edificio.

# Allegato A

**Blocco di inizializzazione:**

```
addpath .\BuildingF40\datafiles
addpath .\meteofiles\
addpath .\libraries\
addpath .\MOOP\
addpath .\NSGA-II\
addpath .\hambase09\
addpath .\f40\hambase09adds\
addpath .\hambase09adds\
global oOpt outStructOpt cont xFinalall;
cont=0;
m=6;
n=21;
h=1;
```

**Blocco di elaborazione singola:**

```
global oOpt outStructOpt cont xFinalall;
cont=0;
Prob = f40_optimization_config2(m,n,struct(),0);
Prob.nsga2_simulator.contatore=0;
out = moop(Prob);
out_tot(1)=out;
Te_allday(1).Te(:,1)=oOpt.Te.Time;
Te_allday(1).Te(:,2)=oOpt.Te.Data;
indexTotalOSOpt = 2;
figure(1)
plot(out.nsga2_simulator.tot_pareto(:,3),out.nsga2_simulator.tot_pareto(:,4),'*r',...
             'LineWidth',1);
title('Pareto-',num2str(m),'-',num2str(n),'-h1');
ylabel('PPD [%]');
xlabel('Total Consumption [$]');
namefig1=strcat('Pareto-',num2str(m),'-',num2str(n),'-h1');
savefig(namefig1);
```

**Blocco iterativo di elaborazione:**

```
oOpt.out=out.nsga2_simulator.tot_pareto;
Prob = f40_optimization_config2(m,n,oOpt,1);
Prob.nsga2_simulator.contatore=1;
out = moop(Prob);
out_tot(h+1)=out;
solutionbest(h,:)=Prob.nsga2_simulator.paretobest;
totalOutStructOpt(indexTotalOSOpt,1:length(outStructOpt)) = outStructOpt;
indexTotalOSOpt = indexTotalOSOpt + 1;
Te_allday(h+1).Te(:,1)=oOpt.Te.Time;
Te_allday(h+1).Te(:,2)=oOpt.Te.Data;
name=strcat('Pareto-',num2str(m),'-',num2str(n));
save(name,'outStructOpt','out')
f=h+1;
figure(f)
plot(out.nsga2_simulator.tot_pareto(:,3),out.nsga2_simulator.tot_pareto(:,4),'*r','LineWi
dth',1);
title(['Pareto-',num2str(m),'-',num2str(n),'-h',num2str(f)]);
ylabel('PPD [%]');
xlabel('Total Consumption [$]');
namefig=strcat('Pareto-',num2str(m),'-',num2str(n),'-h',num2str(f));
savefig(namefig);
```

### Blocco di aggiornamento e salvataggio

```
solutionbest(h+1,:)=paretobest(out.nsga2_simulator.tot_pareto);
save('totalOutStructOpt')
figure(f+1)
X=0:3600:86400+1;
Y1=vertcat(0,solutionbest(1,3),solutionbest(2:end,3)-solutionbest(1:end-1,3));
Y2=vertcat(10,solutionbest(:,4));
plot(X,Y1,X,Y2);
thisFigName = strcat('Confronto PPD-Consumo-',num2str(m),'-',num2str(n));
title(thisFigName);
ylabel('Consumo[$] PPD[%]');
xlabel('Time[h]');
savefig(thisFigName);
nameYY = strcat('ConsumiPDD-',num2str(m),'-',num2str(n));
nameResult = strcat('Result-',num2str(m),'-',num2str(n));
save(nameYY,'Y1','Y2');
save(nameResult,'out_tot');
clearvars -except m n;
```

# Allegato B

| | | Giorno 1 | | Giorno 2 | | Giorno 3 | | |
|---|---|---|---|---|---|---|---|---|
| # | Giorni di simulazione | Tempo iniziale | Tempo finale | Tempo iniziale | Tempo finale | Tempo iniziale | Tempo finale | Durata |
| 1 | 1 | 10:00 | 14:31 | / | / | | | 4:30 ore |
| 2 | 1 | 15:11 | 19:06 | / | / | | | 4 ore |
| 3 | 1 | 5:14 | 8:57 | / | / | | | 3:30 ore |
| 4 | 1 | 8:40 | 11:47 | / | / | | | 3 ore |
| 5 | 1 | 1:53 | 4:57 | / | / | | | 3 ore |
| 6 | 2 | 23:22 | 2:51 | 3:02 | 6:32 | | | 7 ore |
| 7 | 2 | 13:03 | 17:25 | 17:35 | 21:00 | | | 8 ore |
| 8 | 2 | 23:30 | 3:29 | 3:39 | 7:32 | | | 8 ore |
| 9 | 3 | 14:00 | 17:35 | 17:35 | 20:38 | 20:48 | 23:50 | 10 ore |