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

MINISTERO DELLO SVILUPPO ECONOMICO

Metodologia di ottimizzazione multi-obiettivo della climatizzazione termica di edifici. Validazione su sistema di simulazione

F. Moretti, S.Panzieri

ROMA TRE
UNIVERSITÀ DEGLI STUDI

Report RdS/2013/122

METODOLOGIA DI OTTIMIZZAZIONE MULTI-OBIETTIVO DELLA CLIMATIZZAZIONE TERMICA DI EDIFICI. VALIDAZIONE SU SISTEMA DI SIMULAZIONE.

F. Moretti (ENEA), S. Panzieri (Univ. Roma Tre)

Settembre 2013

Report Ricerca di Sistema Elettrico
Accordo di Programma Ministero dello Sviluppo Economico – ENEA
Piano Annuale di Realizzazione 2012
Area: Razionalizzazione e risparmio nell'uso dell'energia elettrica
Progetto: Risparmio di energia elettrica nei settori: civile, industria e servizi
Obiettivo: Gestione ottimale delle reti di edifici

Responsabile del Progetto: Ilaria Bertini, ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione "Sviluppo di algoritmi per la gestione ottimale di edifici : applicazione su un caso pilota"

Responsabile scientifico ENEA: Stefano Pizzuti

Responsabile scientifico Università di Roma Tre: Stefano Panzieri

Indice

1 DESCRIZIONE ATTIVITÀ 4

2 OTTIMIZZAZIONE MULTIOBIETTIVO 5

3 ALGORITMI DI OTTIMIZZAZIONE 6

3.1 ALGORITMI GENETICI 6
3.2 ALGORITMI MUTLIOBIETTIVO 7
3.2.1 Weighted Sum 7
3.2.2 Vincolo ϵ 7
3.3 ALGORITMI GENETICI MULTIOBIETTIVO 8
3.3.1 Vector Evalued GA 9
3.3.2 NSGA-II 9

4 FITNESS APPROXIMATION 10

5 IMPLEMENTAZIONE E TESTING 12

5.1 IMPLEMENTAZIONE ALGORITMI DI OTTIMIZZAZIONE 12
5.2 BENCHMARKING OTTIMIZZAZIONE MULTIOBIETTIVO 14
5.3 BENCHMARKING FITNESS APPROXIMATION 16

6 OTTIMIZZAZIONE TERMICA DELL'EDIFICIO F40 17

6.1 DESCRIZIONE DELLO SCENARIO 18
6.1.1 Funzioni fitness 19
6.1.2 Setup del problema 20
6.1.3 Ottimizzazione con NSGA-II supportato da fitness approximation 21

7 CONCLUSIONI E SVILUPPI FUTURI 24

8 BIBLIOGRAFIA 25

9 CURRICULUM 26

FABIO MORETTI: 26
STEFANO PANZIERI: 26
GRUPPO DI LAVORO 26

1 Descrizione attività

Nell'ambito della realizzazione del dimostrativo dello "Smart Village" nel Centro Ricerche ENEA Casaccia sono stati sviluppati degli algoritmi di ottimizzazione multiobiettivo al fine di attuare politiche di controllo ottimizzate sugli edifici. Il rapporto tecnico fa riferimento PAR 2012 sulla Ricerca di Sistema Elettrico, in particolare al ***"Sviluppo di algoritmi per la gestione ottimale di edifici: applicazione su un caso pilota***" ed è realizzato dal Dipartimento di Informatica ed Automazione dell'Università degli Studi "Roma Tre".

La piattaforma integrata realizzata nelle precedenti annualità del PAR ci ha consentito di applicare metodologie di diagnostica avanzata al fine di monitorare il comportamento energetico degli edifici e di poter intervenire qualora fosse possibile al recupero del guasto o, in alternativa, alla segnalazione dell'anomalia. L'obiettivo dell'annualità corrente è quello di attuare delle logiche di controllo ottimizzate al fine di prevenire sprechi e cattive gestioni dell'utenza. A tale scopo è stato realizzato in collaborazione con l'Università Politecnica delle Marche un simulatore che approssima il comportamento energetico di un edificio all'interno del Centro Ricerche "Casaccia".

Il simulatore, dati come input i set-point di temperatura dei termostati che afferiscono a delle zone preimpostate dell'edificio e la temperatura di mandata dell'acqua in centrale termica, fornisce i valori elettrici, termici e ambientali dell'edificio. Tali valori costituiscono l'input delle funzioni obiettivo, realizzate in collaborazione con il Politecnico di Torino, da ottimizzare. In particolare le funzioni obiettivo sono due: una fornisce un valore adimensionale relativo al comfort dell'utente, l'altra un valore relativo alla quantificazione del consumo energetico.

Tali problemi rientrano nella categoria dell'ottimizzazione multiobiettivo, in quanto lo scopo è massimizzare il comfort dell'utente e contemporaneamente minimizzare i consumi termici. Le due funzioni sono in contrasto tra loro poiché la massimizzazione del comfort può comportare un aumento dei consumi, viceversa la minimizzazione dei consumi può comportare un decremento del comfort.In situazioni in cui le funzioni obiettivo sono totalmente o parzialmente in contrapposizione l'ottimo non è univoco, bensì un insieme di soluzioni non dominate definito come fronte di Pareto, che sono il risultato del compromesso tra comfort e risparmio energetico.

Sono stati effettuati studi su quale approccio fosse migliore per risolvere i problemi di ottimizzazione multi obiettivo, in particolare se utilizzare i metodi di ottimizzazione basati su gradiente e sulla direzione di discesa o gli algoritmi evolutivi, basati su principi ispirati alla natura.

In particolare gli algoritmi genetici si basano sul principio darwiniano della selezione naturale secondo la quale gli individui dominanti sopravvivono a discapito di quelli più deboli. Di conseguenza, la probabilità che gli individui dominanti procreino è maggiore, mentre quelli più deboli si vanno estinguendo con il passare delle generazioni. La qualità dell'individuo, la sua *fitness*, è l'equivalente della funzione obiettivo nel campo della ricerca operativa classica.

Per i problemi di ottimizzazione multiobiettivo gli algoritmi evolutivi sono considerati molto efficienti, in quanto, essendo basati su una popolazione iniziale, esplorano in maniera efficace lo spazio delle soluzioni. Sono stati eseguiti test su dei problemi di ottimizzazione multiobiettivo di benchmarking con l'algoritmo NSGA-II ed i risultati, misurati con delle metriche note in letteratura, sono stati soddisfacenti.

L'applicazione sulle funzioni obiettivo precedentemente descritta è risultata in prima istanza problematica, in quanto la valutazione del valore della fitness delle due funzioni dipende dall'output del simulatore, un processo che richiede un calcolo computazionale non trascurabile. Poiché gli algoritmi genetici si basano su un numero elevato di valutazioni della fitness, il processo di ottimizzazione in questo caso risente del collo di bottiglia dovuto all'elaborazione del simulatore, è stato ritenuto opportuno fare uso della fitness approximation, un approccio in cui in realtime viene costruito un modello che approssima le funzioni di fitness mappando gli input al simulatore e l'output delle funzioni.

2 Ottimizzazione multiobiettivo

L'ottimizzazione multi obiettivo, o **MOOP**(**M**ulti **O**bjective**OP**timimisation) è un processo che mira a trovare la soluzione ottima di un problema costituito da molteplici obiettivi.I primi studi su come risolvere i **MOP** (**M**ulti **O**bjective**P**roblem) sono iniziati nella seconda metà del '900 e sono aumentati notevolmente con il passare degli anni. Ciò è dovuto al fatto che quasi la totalità dei problemi reali non può essere ricondotta ad un **SOP** (Single ObjectiveProblem), in quanto ci sono spesso molteplici aspetti di cui tener conto, il più delle volte in contrasto tra loro. I primi approcci che furono applicati per la risoluzione dei MOP consistono nel ricondurre il problema ad un SOP, eventualmente sfruttando informazioni come l'ordine di preferenza degli obiettivi . Quando si affronta un MOP ottenere una soluzione univoca è possibile solo qualora gli obiettivi non sono in conflitto tra loro, ma nella maggior parte dei casi reali questo non accade. Quindi nella MOOP si ha a che fare con un insieme di soluzioni, e non più una soluzione univoca.
In letteratura la risoluzione di MOP è basata su due differenti approcci:

- Approccio basato sulle preferenze
- Approccio ideale

L'approccio basato sulle preferenze consiste nello sfruttare le eventuali informazioni preliminari, come l'ordine di preferenza degli obiettivi, per pesare l'importanza delle funzioni obiettivo e ricondurre così il MOP ad un SOP. L'efficacia di questo approccio è discutibile, in quanto è fortemente dipendente dai pesi. Lo stesso MOP infatti, risolto con lo stesso tipo di approccio, ma con pesi differenti, può dare risultati notevolmente differenti. Inoltre la scelta dei pesi risulta essere molto soggettiva e poco diretta. Infatti, pur conoscendo l'ordine di preferenza degli obiettivi non è immediato poter quantificare numericamente un fattore che indichi con precisione quanto un obiettivo sia più importante di un altro. L'approccio ideale invece mira a trovare in una prima fase un insieme di soluzioni, senza basarsi su alcun tipo di preferenza iniziale. Una volta ricavato l'insieme di soluzioni, nella seconda fase vengono sfruttate le eventuali preferenze per selezionare la migliore. E' un approccio molto più efficace del precedente, in quanto le soluzioni ricavate non sono indirizzate da preferenze personali. L'unico scopo della prima fase è quello di trovare un numero più alto possibile di soluzioni differenti.

La MOOP, come la **SOOP** (**S**ingle **O**bjective**OP**timimisation), è caratterizzata da uno spazio delle decisioni multidimensionale, ovvero lo spazio che caratterizza tutti i valori che le variabili di decisione (gli input) possono assumere. Se l'ottimizzazione non è vincolata tale spazio è illimitato. La differenza sostanziale con la SOOP è che anche lo spazio degli obiettivi è multidimensionale, ovvero lo spazio che rappresenta il valore che le funzioni obiettivo possono assumere. La corrispondenza tra i due spazi non è biunivoca e la relazione che li lega è spesso non lineare, per cui il mapping non è banale. Formalmente la MOOP è definita come:

Equazione 1. Forma generalizzata dei MOOP

$$\begin{cases} Min/Max & f_m(x) & m = 1,2,...,M \\ & g_j(x) \geq 0 & j = 1,2,...,J \\ & h_k(x) = 0 & k = 1,2,...,K \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1,2,...,n \end{cases}$$

Dove:

- M numero di obiettivi
- J numero di vincoli
- K numero di vincoli di uguaglianza
- n numero di variabili di decisione
- $x^{(L)}$ e $x^{(U)}$ bound delle variabili di decisione

Poiché nel caso della MOOP si devono confrontare più obiettivi, definire quale soluzione sia migliore di un'altra non è immediato come nella SOOP, per cui viene introdotto il concetto di dominanza di Pareto secondo cui una soluzione x_1 domina ($\triangleleft$) una soluzione x_2 se:

Equazione 2. Criterio di dominanza

1. $f_i(x^{(1)}) \not\triangleleft f_i(x^{(2)}) \quad per \quad i = 1 \dots M$
2. $\exists\, i \mid f_i(x^{(1)}) \triangleleft f_i(x^{(2)}) \quad per \quad i = 1 \dots M$

3 Algoritmi di ottimizzazione

3.1 Algoritmi genetici

Gli algoritmi genetici o GA (**G**enetic **A**lgorithm) appartengono alla categoria degli algoritmi evolutivi e si basano sui principi della genetica secondo cui, al momento della nascita di ogni individuo, il DNA a questi associato è influenzato in parte dal DNA dei genitori ed in parte dal processo di mutazione. L'associazione di ogni soluzione con un DNA è l'aspetto peculiare dei GA e significa identificare il suo cromosoma (o genotipo), ovvero la combinazione genetica che contraddistingue univocamente tale soluzione. Il processo inverso, ovvero associare al genotipo la soluzione e le relative variabili di decisione ad essa associata significa ricavare il fenotipo. Facendo riferimento alla genetica umana, una serie di cromosomi, ossia il nostro genotipo, determinano il colore dei nostri occhi, ossia il fenotipo. Tali genotipi costituiscono la popolazione dell'algoritmo genetico, ovvero l' insieme di individui, o soluzioni e sono rappresentati con delle stringhe binarie.

Sulla popolazione vengono applicati i principi della genetica e dell'evoluzione darwiniana: selezione, crossover e mutazione.

Il ruolo dell'operatore di selezione è quello di dare enfasi alle soluzioni migliori a discapito delle peggiori. L'accostamento alla selezione naturale in cui il più forte prevale sul più debole della sua stessa specie è evidente. La selezione può essere ottenuta in modi differenti, ma ciò che accomuna tutti gli approcci è che a partire da una popolazione di individui, ne viene creata una nuova con un numero più alto di individui migliori e minore di individui peggiori. Il processo di selezione più comune è per torneo, in cui avviene, come in natura, un combattimento per la sopravvivenza. Vengono estratti casualmente dalla popolazione due o più individui, l'individuo con il valore della funzione obiettivo (fitness nella computazione evolutiva) migliore sopravvive e viene inserito nella nuova popolazione.

Il crossover consente la creazione di nuove soluzioni precedentemente non esistenti combinando i bit di due soluzioni e generando cosi una nuova soluzione. Mantenendo l'accostamento con la genetica i cromosomi di due individui, i genitori, si combinano dando vita ad una progenie (generalmente due figli) con caratteristiche comuni ad entrambi i genitori.

La mutazione è il processo che consente agli algoritmi genetici di generare nuove soluzioni e quindi esplorare in maniera più completa lo spazio di ricerca. Il metodo utilizzato per gli algoritmi genetici è la mutazione Bit-a-Bit. Una volta applicata la selezione ed il crossover ogni bit di ogni stringa della popolazione ha una probabilità (solitamente molto bassa) che venga invertito. La conversione di un bit viene concepita come una piccola modifica al cromosoma di un individuo al momento della nascita che gli conferisce caratteristiche non appartenenti ai genitori. La Figura 1 mostra il diagramma di flusso del processo dell'algoritmo genetico.

Figura 1. Diagramma di flusso del GA

3.2 Algoritmi mutliobiettivo

Gli algoritmi Multiobiettivo o MOA (**M**ulti **O**bjective **A**lgorithm) sono una categoria di algoritmi in grado di ottimizzare i MOP, e si suddividono in due tipologie dipendenti dall'approccio all'ottimizzazione cui è stato fatto riferimento nel paragrafo 2: approccio basato sulle preferenze o approccio multiobiettivo.

3.2.1 Weighted Sum

Box [1] nel 1957 propose il metodo della somma pesata, basato sulle preferenze, in cui viene assegnato un peso ad ogni obiettivo e l'obiettivo è la minimizzazione della funzione che ingloba tutti gli obiettivi, il MOP viene ricondotto alla forma mostrata nell' Equazione 3.

Equazione 3. Metodo della Somma Pesata

$$\begin{cases} Min/Max & F(x) = \sum_{m=1}^{M} w_m f_m(x) & m = 1, 2, ..., M \\ & g_j(x) \geq 0 & j = 1, 2, ..., J \\ & h_k(x) = 0 & k = 1, 2, ..., K \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1, 2, ..., n \end{cases}$$

Tale approccio è fortemente dipendente dalla scelta iniziale dei pesi assegnati alle funzioni obiettivo, inoltre non è efficace per problemi non convessi, in quanto è stato dimostrato che esistono soluzioni che tale algoritmo non può trovare [2]. Inoltre per poter assegnare dei pesi significativi alle funzioni è necessario effettuare uno scaling che non sempre è possibile attuare.

3.2.2 Vincolo ϵ

Haimes et al. [3] proposero un metodo basato sui vincoli. L'idea è quella di ottimizzare una delle funzioni obiettivo, mentre le altre sono convertite in vincoli del problema come mostrato nell'Equazione 4. Anche questo approccio è influenzato dalla scelta del vincolo,ma riesce a risolvere anche problemi di

ottimizzazione non convessa. La Figura 2 mostra come la scelta del vincolo condizioni l'ottimizzazione multiobiettivo: scegliendo un vincolo troppo restrittivo ϵ_a non viene trovata nessuna soluzione, con il vincolo ϵ_c viene trovata una soluzione intermedia (C) mentre con un vincolo poco restrittivo ϵ_d viene trovata la soluzione che minimizza $f1$ (D)

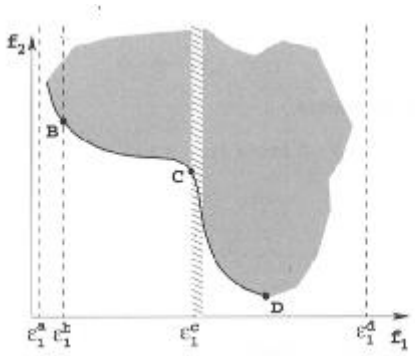

Figura 2. Esempio di vincolo ϵ

Equazione 4. Metodo del vincolo ϵ

$$\begin{cases} Min/Max & f_\mu(x) \\ & f_m(x) \leq \epsilon_m \qquad m = 1,2,...,M \; m \neq \mu \\ & g_j(x) \geq 0 \qquad j = 1,2,...,J \\ & h_k(x) = 0 \qquad k = 1,2,...,K \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} \qquad i = 1,2,...,n \end{cases}$$

3.3 Algoritmi genetici multiobiettivo

Gli algoritmi genetici multiobiettivo si basano sull'approccio ideale, per la risoluzione di problemi multiobiettivo sono più efficienti rispetto agli algoritmi basati sulla direzione di discesa e sul gradiente. Gli algoritmi gradient-based infatti presentano dei problemi parzialmente risolvibili con l'uso degli algoritmi genetici:

- La convergenza dell'algoritmo dipende dalla scelta del punto di partenza. Soprattutto in problemi con molti minimi locali la scelta del punto di partenza può influenzare enormemente l'esito dell'ottimizzazione.
- Alto rischio di convergenza in un sub-ottimo. Un algoritmo classico è diffcile che riesca ad uscire da un ottimo locale.
- Scarsa adattabilità di un algoritmo ad una tipologia di problema differente. Ad esempio il metodo del gradiente coniugato è molto efficiente nel risolvere problemi con funzioni obiettivo quadratiche, ma è inefficiente in problemi con funzioni obiettivo multi-modali.
- Poca attitudine al calcolo parallelo. Poiché i metodi classici prevedono l'uso di una sola soluzione, aggiornata ad ogni iterazione, il calcolo parallelo è difficilmente applicabile.

Gli algoritmi evolutivi in parte risolvono i problemi precedentemente descritti. Essendo population based e la popolazione iniziale solitamente viene creata in maniera casuale, l'esito dell'ottimizzazione dipende in maniera minore dall'inizializzazione. Infatti durante la creazione della popolazione iniziale in tutto lo spazio di ricerca c'è la stessa probabilità che sia creata una soluzione. Inoltre grazie alla rappresentazione delle soluzioni tramite il genotipo è facile far assumere alle variabili valori discreti e quindi gli EA possono essere applicati senza problemi a questa tipologia di problemi. Inoltre, grazie alla loro natura stocastica, e non basata su metodi ben precisi come gli algoritmi classici, gli EA sono maggiormente robusti e possono adattarsi a problemi differenti.

Gli operatori genetici degli EA (crossover e mutazione) consentono la creazione di nuove soluzioni precedentemente inesistenti, e quindi di esplorare nuove aree dello spazio di ricerca, limitando quindi il pericolo di convergenza ad un ottimo locale.
Infine, la loro natura population based si adatta perfettamente alle esigenze del parallelismo, consentendo quindi la valutazione di più soluzioni contemporaneamente da parte di più calcolatori.

3.3.1 Vector Evalued GA

Il Vector Evalued GA (VEGA) è stato il primo algoritmo basato sull'approccio ideale, ideato da Shaffer nel 1984 [4][5]. Dati M obiettivi, suddivide la popolazione in altrettante sottopopolazioni, ciascuna delle quali viene valutata in base ad uno degli obiettivi. Gli operatori genetici di selezione, crossover e mutazione sono applicati ad ogni sottopopolazione indipendentemente, quindi sono evitati problemi di scaling tra le diverse funzioni obiettivo. Una volta applicati gli operatori genetici, le sottopopolazioni vengono unificate per formare la nuova popolazione. La Figura 3 mostra un esempio di popolazione di 6 individui in un problema di massimizzazione di due funzioni obiettivo le soluzioni 1,2 e 5 alla funzione f_1 mentre le soluzioni 3,4 e 6 vengono associate alla funzione f_2 .

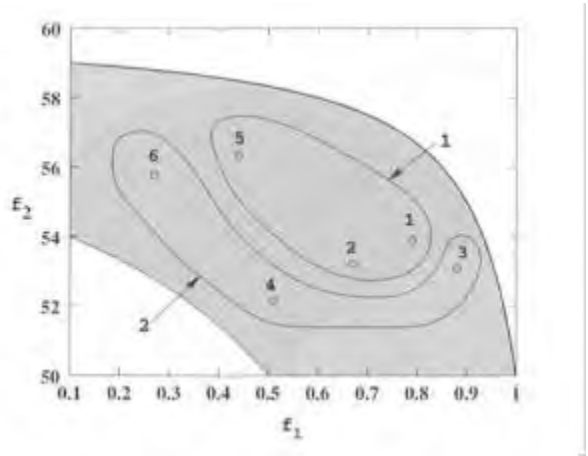

Figura 3. Esempio di VEGA

Il principale problema del VEGA è che in ogni sottopopolazione viene favorita la soluzione che ha miglior fitness nel rispettivo obiettivo, quindi VEGA tende a favorire le soluzioni campioni in un obiettivo piuttosto che quelle buone in tutti gli obiettivi. Nell'esempio le soluzioni ottime saranno quindi la 1 e la 6.

3.3.2 NSGA-II

NSGA-II (Elitist Non-Dominated Sorting Genetic Algorithm) è stato introdotto da Deb nel 2000 [6]. E' una revisione evoluta di NSGA [7] in cui viene utilizzato l'elitismo ed una nuova metodologia di mantenimento della diversità, la crowding distance.
In NSGA-II viene utilizzata una classificazione delle soluzioni secondo un ordine di dominanza. Il principio su cui si basa è quello di assegnare un livello o un fronte di dominanza a tutte le soluzioni di una popolazione. Le soluzioni non-dominate all'interno di tutta la popolazione avranno rank 1. Le soluzioni non-dominate tra tutta la popolazione tranne le soluzioni di rank 1 avranno rank 2 e così via. La Figura 4 mostra un esempio di ordinamento per livelli di una popolazione di 10 individui in un problema di minimizzazione di due funzioni obiettivo.

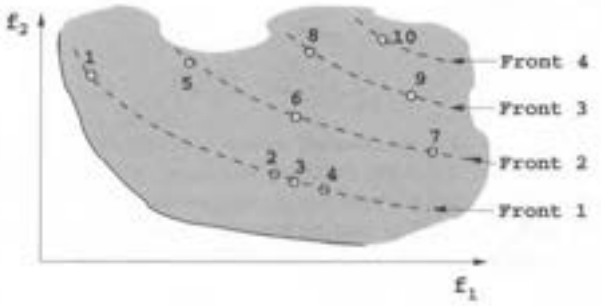

Figura 4. Esempio di ordinamento per livelli

Una volta assegnato un rank a tutte le soluzioni viene assegnata la crowding distance. La crowding distance è un valore che come il niche count, indica quanto una soluzione è isolata. La crowding distance viene applicata in maniera indipendente ad ogni insieme di soluzioni aventi lo stesso rank. Le soluzioni vengono ordinate iterativamente in ordine peggiorativo rispetto ad ogni funzione obiettivo. Alle soluzioni estreme, aventi la migliore e la peggior fitness in un determinato obiettivo, viene assegnata crowding distance pari a infinito, per le altre un valore proporzionale alla distanza tra la i-esima soluzione e le soluzioni precedente e successiva, come mostrato nella Figura 5 .Quindi più alto è il valore di crowding distance, più è isolata la soluzione. Nel confrontare due soluzioni viene comparato prima il rank tra le due soluzioni, se il rank è lo stesso allora viene comparata la crowding distance.

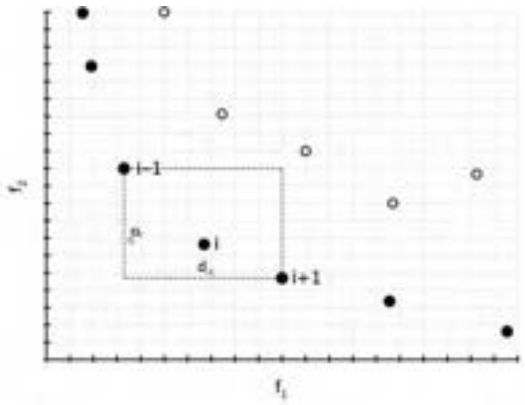

Figura 5. Esempio di assegnazione della crowding distance

4 Fitness Approximation

La Fitness Approximation (FA) è la stima della funzione di costo di un problema di ottimizzazione. La funzione di costo, o funzione obiettivo è definita anche come funzione di fitness in quanto rappresenta , nel campo degli algoritmi evolutivi, l'indice di qualità di una soluzione così come la funzione di costo. Le motivazioni per cui si ricorre all'uso della fitness approximation sono molteplici, tipicamente nei casi in cui la valutazione della funzione di fitness è computazionalmente dispendiosa, come ad esempio nei problemi di design ingegneristico riguardanti la dinamica dei fluidi [8] o l'iniezione del motore di un razzo [9]. Campi come quello delle dinamiche dei fluidi (Computational Fluid Dynamics), dell'elettromagnetismo (Computational Electro-magnetics) o della meccanica strutturale (Computational Structural Mechanics) fanno largamente uso di simulatori nella fase di design e di analisi. In tali casi ogniqualvolta durante il processo di ottimizzazione viene effettuata la valutazione di una soluzione (una richiesta di performance), possono occorrere ore, od anche giorni, per produrre il risultato a seguito di una simulazione. L'uso della fitness approximation consente un risparmio considerevole di tempo. Inoltre può rilevarsi efficace nei casi in cui la funzione di fitness è soggetta a molto rumore [10] o non è disponibile un modello per la sua valutazione [11].

L'obiettivo dell'approssimazione nel caso in cui viene usata per problemi di ottimizzazione computazionalmente od economicamente costosi consiste nel ridurre il più possibile il numero di richieste di performance. Tale obiettivo viene conseguito alternando richieste di performance al modello originario e delle richieste di performance al modello approssimato.

La Figura 6 mostra un esempio di processo di ottimizzazione supportato dalla fitness approximation: i campioni in uscita dal problema vengono memorizzati ed utilizzati per calcolare un modello che approssima le dipendenze ingresso/uscita del problema. Una volta creato tale modello, il processo di ottimizzazione alterna richieste di performance al modello approssimato ed al problema reale.

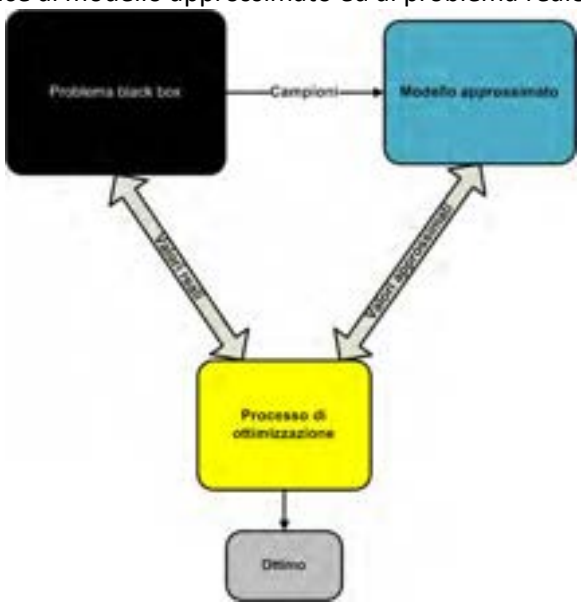

Figura 6. Schema del processo di ottimizzazione supportato dalla fitness approximation

Il rapporto con cui si alternano richieste di performance al problema originario ed al modello approssimato è importante. Chiaramente nella fase iniziale del processo di ottimizzazione è necessario un numero maggiore di richieste al problema originario rispetto alle fasi avanzate, in quanto si presume che il modello approssimato modella il problema in modo migliore all'aumentare del numero di campioni calcolati. Il modello approssimato può essere ottenuto con diverse metodologie: modelli polinomiali, di Kriging, reti neurali artificiali o tramite le Support Vector Machines (SVM). Nella nostra attività abbiamo testato l'approccio con le reti neurali. La Figura 7 mostra un esempio di approssimazione della funzione di Rastrigin con una rete neurale a percettroni multistrato con un addestramento di 40 campioni.

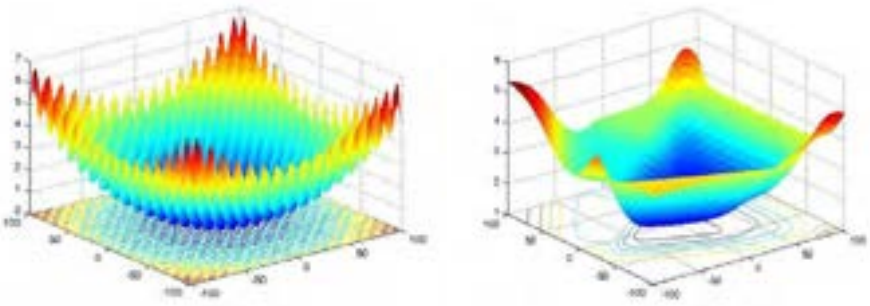

Figura 7. Funzione di Rastrigin (a sinistra) e modello approssimato (a destra)

5 Implementazione e testing

In questa sezione vengono presentati gli algoritmi di ottimizzazione multiobiettivo e di fitness approximation implementati e testati, con l'obiettivo di stabilire quale potesse essere applicabile al nostro caso specifico.

5.1 Implementazione algoritmi di ottimizzazione

Per testare l'efficacia degli algoritmi di ottimizzazione e per avere un quadro su quale potesse essere più promettente per il nostro caso di studio sono stati implementati il WSGA ed NSGA-II. Essendo algoritmi stocastici sono stati confrontati anche con un algoritmo di generazione randomico di soluzioni.

L'Algoritmo 1 mostra la pseudocodifica dell'ordinamento delle soluzioni secondo il ranking e la crowding distance presentati nel sottoparagrafo 3.3.2. Tale algoritmo viene poi utilizzato dall'NSGA-II per ottenere il fronte di Pareto, come mostrato nell'Algoritmo 2.

Algoritmo 1. NonDominatedSorting

Require: P \\ Popolazione avente lo stesso rank
Require: M \\ Numero di obiettivi
1: **for** $1 \leq m \leq M$ **do**
2: AscendingOrderSort(P, f_m) \\ Ordinamento rispetto ai valori di f_m
3: $d_{(1,m)} \leftarrow d_{(|P|,m)} \leftarrow \infty$
4: **for** $2 \leq j \leq |P| - 1$ **do**
5: $d_{(j,m)} \leftarrow d_{(j,m)} + \frac{f_{(j+1,m)} - f_{(j-1,m)}}{f_m^{max} - f_m^{min}}$
6: **end for**
7: **end for**
8: **return** P \\ Popolazione con la Crowding Distance assegnata

Algoritmo 2. NSGA-II

Require: N \\ Numero di individui
1: $P \leftarrow InitalizePopulation(N)$
2: $P \leftarrow NonDominatedSorting(P)$
3: **for all** rank levels **do**
4: $P_i \leftarrow CrowdingDistanceSorting(P_i)$
5: $P \leftarrow P \cup P_i$
6: **end for**
7: **while** *notStopCondition* **do**
8: $P_{selected} \leftarrow Selection(P)$ \\ Basata sul rank e la Crowding Distance
9: $P_{offspring} \leftarrow Crossover(P_{selected})$
10: $P_{mutated} \leftarrow Mutation(P_{offspring})$
11: $P \leftarrow P \cup P_{mutated}$ \\ Applicazione dell'elitismo
12: $P \leftarrow RankingLevelSorting(P)$
13: **for all** rank levels **do**
14: $P_i \leftarrow CrowdingDistanceSorting(P_i)$
15: $P \leftarrow P \cup P_i$
16: **end for**
17: $ReplacePopulation(P, N)$ \\ Prendo i milgiori N di P
18: **end while**
19: **return** P_1 \\ Ritorno gli individui appartenenti al primo rank

L'Algoritmo 3 mostra la pseudocodifica del Weighted Sum Genetic Algorithm.

Algoritmo 3. WSGA

Require: N \\ Numero di individui
Require: M \\ Numero di obiettivi
$P \leftarrow InitalizePopulation(N)$
$W_{tot} = 1$ \\ Inizializzazione dei pesi
for all $i \in M$ **do**
  $W_i \leftarrow RAND(0, W_{tot})$ \\ Numero random compreso tra i due valori
  $W_{tot} = W_{tot} - W_i$
end for
while $notStopCondition$ **do**
  $P_{selected} \leftarrow Selection(P)$ \\ Basata sui pesi W
  $P_{offspring} \leftarrow Crossover(P_{selected})$
  $P_{mutated} \leftarrow Mutation(P_{offspring})$
  $P_{mutated} \leftarrow best(P)$ \\ Inserimento campione generazione precedente
end while
return P \\ Ritorno della popolazione finale

5.2 Benchmarking ottimizzazione multiobiettivo

Tali algoritmi sono stati applicati su problemi multi obiettivo di benchmark ideati da Zitzler, Deb e Thiele da cui il nome ZDT, in cui l'obiettivo è la minimizzazione di entrambe le funzioni e caratterizzati da differenti fronti di Pareto:

- Continuo e convesso (ZDT1)
- Continuo non convesso (ZDT2)
- Discreto (ZDT3)

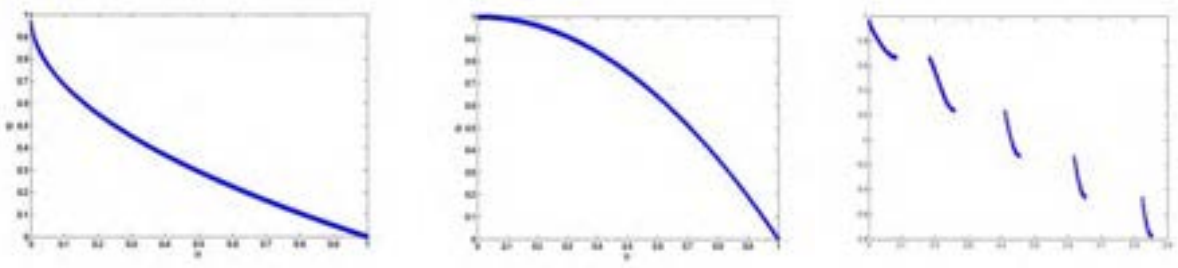

Figura 8. ZDT1, ZDT2, ZDT3

Lo spazio delle decisioni dei test è molto elevato (n = 30) per cui sono state valutate 14000 valutazioni richieste di performance (calcolo della fitness). Sono stati eseguiti 10run di NSGA-II e di WSEA, e viene mostrata l'unione dei fronti di Pareto ottenuti ad ogni run. I parametri dell'algoritmo genetico utilizzato sono mostrati nella Tabella 1. E' importate sottolineare che l'efficacia dell'ottimizzazione dipende non solo dal numero di soluzioni non dominate, ma anche di come esse sono distribuite nello spazio delle decisioni. I risultati dei test (figure 3-5) mostrano che NSGA-II risulta migliore sia come numero di soluzioni sia come distribuzione, ed ha una conformazione del fronte molto simile a quella reale.

Tabella 1. Parametri algoritmo genetico

Popolazione	**70**
Generazioni	20
Selezione	BinaryTournament
Crossover	SBX
Probabilità Crossover	0.8
Mutation	Polinomiale
Probabilità Mutation	0.03
Elitismo	Generazionale
Tipologia Popolazione	Parametri Reali

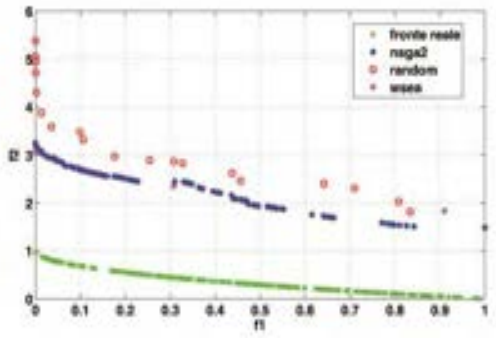

Figura 9. Risultati ZDT1

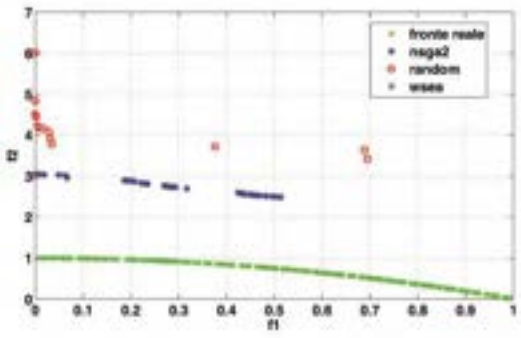

Figura 10. Risultati ZDT2

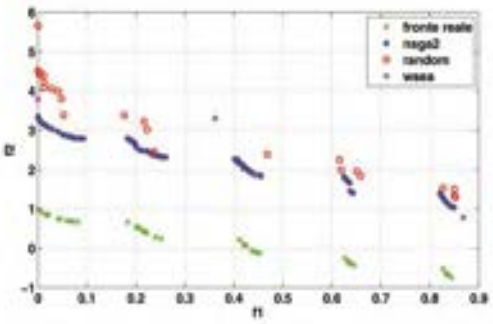

Figura 11. Risultati ZDT3

Tabella 2. Dominance Ratio ZDT1

	NSGA-II	RAND	WSGA
NSGA-II	x	0	0,17
RAND	1	x	0,73
WSGA	0,33	0,33	X

Tabella 3. Spacing ZDT1

NSGA-II	RAND	WSGA
0,01	0,09	0,05

Tabella 4. Ipervolume ZDT1

NSGA-II	RAND	WSGA
0,61	0,51	0,53

Tabella 5. Dominance Ratio ZDT2

	NSGA-II	RAND	WSGA
NSGA-II	x	0	0,11
RAND	1	x	1
WSGA	0	0	X

Tabella 6. Spacing ZDT2

NSGA-II	RAND	WSGA
0,12	0,28	0,19

Tabella 7. Ipervolume ZDT2

NSGA-II	RAND	WSGA
0,58	0,49	0,5

Tabella 8. Dominance Ratio ZDT3

	NSGA-II	RAND	WSGA
NSGA-II	x	0	0,01
RAND	1	x	0,58
WSGA	0,5	0,25	X

Tabella 9. Spacing ZDT3

NSGA-II	RAND	WSGA
0,01	0,22	0,32

Tabella 10. Ipervolume ZDT3

NSGA-II	RAND	WSGA
0,64	0,54	0,51

I risultati confermano che sia NSGA-II che WSGA hanno un rendimento migliore del RANDOM e che NSGA-II ha il rendimento migliore, sia in terminidi soluzioni non dominate, sia in termini di spreading delle soluzioni.

5.3 *Benchmarking fitness approximation*

Per testare l'efficacia dell'algoritmo è stato confrontato NSGA-II con e senza approssimazione sui problemi multiobiettivo ZDT. I grafici e le tabelle di seguito riportate mostrano che a parità di richieste di performance (300) le soluzioni di NSGA-II con approssimazione dominano quasi totalmente quelle ottenute con NSGA-II senza approssimazione.

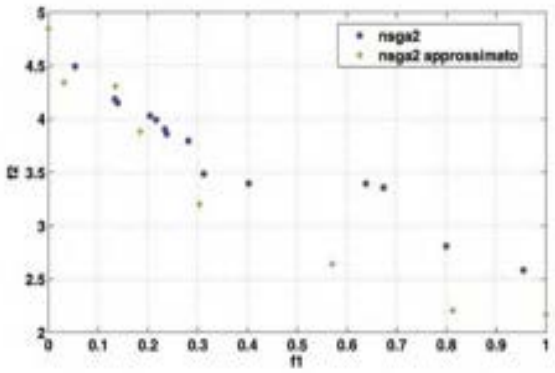

Figura 12. ZDT1

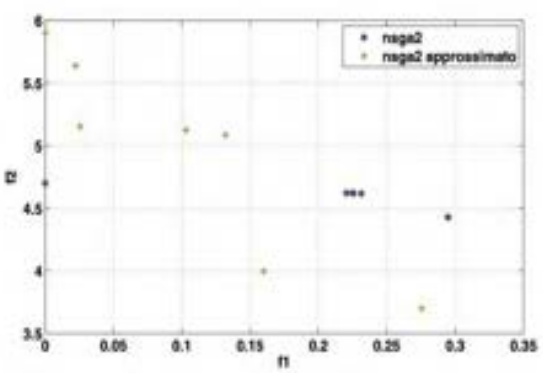

Figura 13. ZDT2

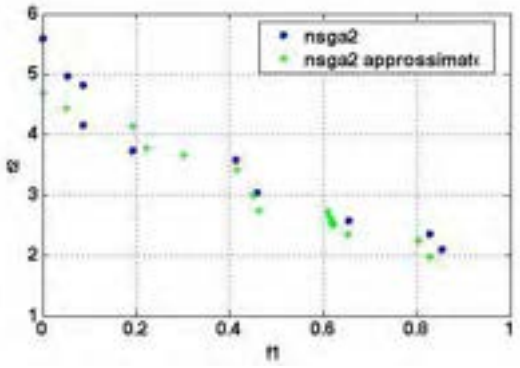

Figura 14. ZDT3

Tabella 11. Dominance Ratio ZDT1 con approssimazione

	NSGA-II	NSGA-II approx
NSGA-II	x	0,71
NSGA-II approx	0,18	x

Tabella 12. Spacing ZDT1 con approssimazione

NSGA-II	NSGA-II approx
0,26	0,34

Tabella 13. Ipervolume ZDT1 con approssimazione

NSGA-II	NSGA-II approx
0,24	0,35

Tabella 14. Dominance Ratio ZDT2 con approssimazione

	NSGA-II	NSGA-II approx
NSGA-II	x	0,89
NSGA-II approx	0,67	x

Tabella 15. Spacing ZDT2 con approssimazione

NSGA-II	NSGA-II approx
0,09	0,08

Tabella 16. Ipervolume ZDT2 con approssimazione

NSGA-II	NSGA-II approx
0,21	0,21

Tabella 17. Dominance Ratio ZDT3 con approssimazione

	NSGA-II	NSGA-II approx
NSGA-II	x	0,76
NSGA-II approx	0,2	x

Tabella 18. Spacing ZDT3 con approssimazione

NSGA-II	RAND
0,21	0,09

Tabella 19. Ipervolume ZDT3 con approssimazione

NSGA-II	NSGA-II approx
0,37	0,41

6 Ottimizzazione termica dell'edificio F40

L'attività di ottimizzazione termica dell'edificio F40 è focalizzata sulla gestione dei set-point di temperatura degli uffici. L'obiettivo è minimizzare il consumo termico relativo al riscaldamento delle stanze ed al tempo stesso massimizzare il comfort degli utenti, quindi è necessario valutare il miglior compromesso tra i due obiettivi. Il risultato aspettato dell'ottimizzazione è un insieme di soluzioni da proporre al decision maker, il quale avrà il compito di valutare in maniera automatizzata o meno, quale delle soluzioni ottime ottenute attuare nel sistema di controllo. E' fondamentale quindi che la distribuzione delle soluzioni sia molto uniforme, in modo tale da offrire al decision maker una gamma d'impostazioni ottime che non propenda solamente verso uno dei due obiettivi.

6.1 Descrizione dello scenario

L'edificio F40 è stato suddiviso in 15 zone, 5 zone per ogni piano di cui una rappresenta i corridoi mentre le altre 4 corrispondono alle zone degli uffici suddivise secondo l'esposizione alla radiazione solare, come mostra la Figura 15.

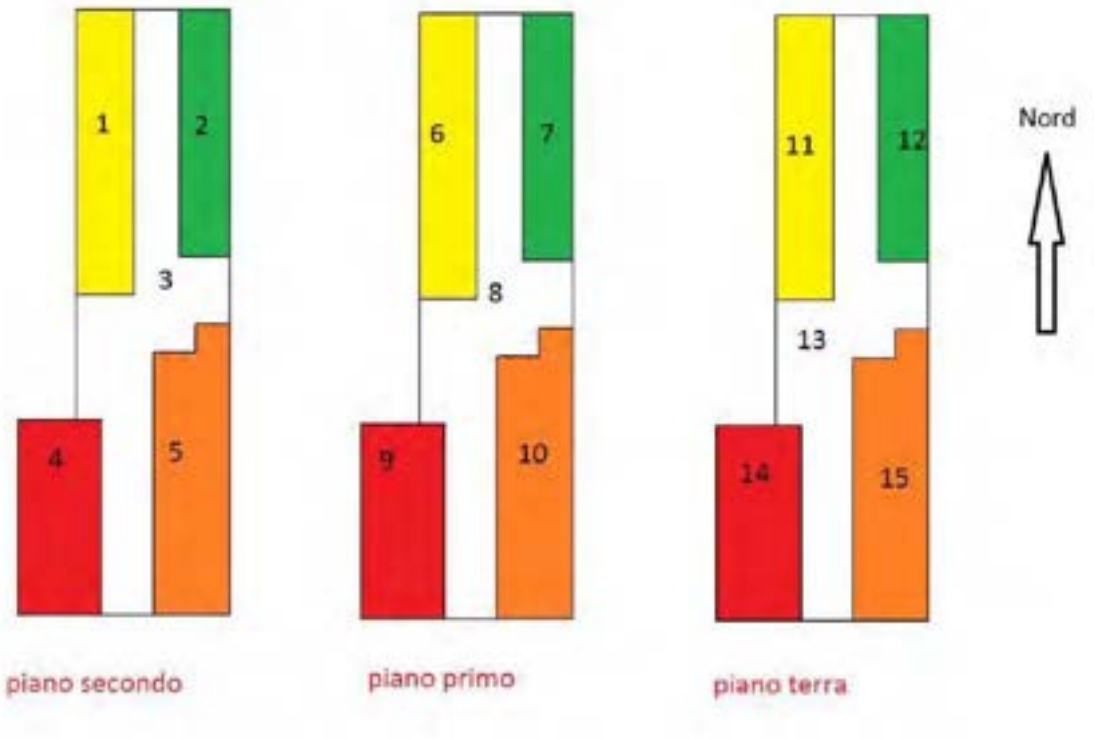

Figura 15. Suddivisione in zone termiche dell'edificio F40

Gli input del simulatore sono costituiti da:

- Set-point delle 15 zone
- Set-point della temperatura di mandata

Gli output del simulatore sono mostrati nella Tabella 20.

Tabella 20. Output del simulatore dell'edificio F40

Grandezza	Collocazione	Unità di Misura
Temperatura Interna Aria	Ogni zona	[˚C]
Temperatura Radiante	Ogni zona	[˚C]
Umidità Relativa Interna	Ogni zona	Adim
Pressione Interna	Ogni zona	[Pa]
Temperatura Esterna	Ogni zona	[˚C]
Umidità Relativa Esterna	Ogni zona	Adim
Potenza Termica Fornita	Ogni zona	[W]
Temp. Radiante Fluido	Ogni zona	[˚C]
Temp.Aria uscita Fancoil	Ogni zona	[˚C]
Temp. Ritorno Collettore	Centrale Termica	[˚C]
Potenza Termica Fornita	Centrale Termica	[W]
Portata Volumetrica Gas	Centrale Termica	[m^3/h]

Potenza Elettrica	Centrale Termica	[W]
Energia Termica Fornita	Centrale Termica	[kWh]
Volume Gas Consumato	Centrale Termica	[m^3]
Energia Elettrica Assorbita	Centrale Termica	[kWh]

L'approccio utilizzato per ottimizzare i set-point di temperatura delle 15 zone e della temperatura di mandata della centrale termica è stato quello di eseguire 1000 simulazioni di una giornata tipo a condizioni a contorno costanti durante la giornata. Le forzanti esterne tenute costanti sono mostrate nellaTabella 21.

Tabella 21. Forzanti esterne

Grandezza	Unità di misura
Radiazione Solare Diretta	[W/m^2]
Radiazione Solare Diffusa	[W/m^2]
Umidità relativa esterna	Adim
Temperatura esterna	[˚C]
Nuvolosità	Adim
Velocità del vento	[m/s]
Direzione del vento	Gradi Angolari rispetto al nord

6.1.1 Funzioni fitness

Gli output del simulatore sono stati utilizzati per la definizione delle due funzioni di fitness da ottimizzare.

Funzione di fitness del comfort

Il comfort termico esprime il grado di soddisfazione di un individuo all'interno di un ambiente. E' un valore soggettivo influenzato da una serie di fattori:

- Metabolismo
- Isolamento termico del vestiario
- Temperatura dell'aria
- Velocità dell'aria
- Temperatura media radiante
- Umidità relativa

La funzione di fitness del comfort si basa sul PredictedMean Vote (PMV) definito dal modello di Fanger, un indice adimensionale valutatp su una scala semantica [-3,3] in cui 0 è lo stato ottimale di comfort dell'utente e da Ndis, un indice del numero di ore in cui il valore assoluto del PMV > 0,5.
La funzione obiettivo del comfort è la PPD (PredictedPercentage of Dissatisfied) definita dal PMV e da una funzione di penalità basata su Ndis.

Funzione di fitness dell'energia termica

La funzione di fitness relativa ai consumi energetici si basa sull'energia termicarichiesta per fornire calore durante la stagione invernale e raffrescamento nella stagione estiva. L'energia termica calcolata è relativa all'intero edificio.

La funzione dell'indice di comfort è stata normalizzata [0,1], in cui 0 esprime il massimo comfort dell'utente. Mentre il consumo di energia termica è espresso in kW/m^3 per rendere tale risultato confrontabile.

Per ulteriori informazioni sulle funzioni obiettivo si rimanda al rapporto tecnico del Politecnico di Torino "*Monitoraggio e diagnostica remota di una rete di edifici*".

6.1.2 Setup del problema

Per ottenere 1000 combinazioni random di input da immettere nel simulatore sono stati impostati dei rangeinput dei set-point il range dei set-point delle zone varia da 19 a 21 gradi, con step di un grado, il range della temperatura di mandata della centrale termica varia da 45 a 55 gradi, sempre con step di un grado.Le possibili combinazioni ottenibili sono quindi 3^15 + 10 = 14348917. Le 1000 combinazioni ottenute costituiscono l'analisi di una piccola parte del possibile spazio delle decisioni.
Tali simulazioni sono state eseguite tenendo costanti le forzanti esterne e mantenendo lo stesso set-point su ogni zona per 24 ore, l'output del PMV e del consumo termico orario è quindi mediato sulle 24 ore.

Nella fase di setup del problema è stato calcolato il fronte di Pareto ottimo, ottenuto applicando il criterio di dominanza a tutte le 1000 combinazioni ottenute con la simulazione. Sono state trovate 55 soluzioni appartenenti al fronte ottimo, mostrate nella Tabella 22.

Tabella 22. Soluzioni del fronte di Pareto ottimo

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Tm	En.Termica	PMV
(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(kW/m^3)	Adim
21	21	20	21	21	21	19	21	20	21	20	19	21	20	19	49	0,123860796	0,206557333
21	21	21	21	21	20	19	20	19	21	20	21	19	21	20	50	0,141784927	0,194774222
21	21	20	20	21	20	21	19	21	21	21	21	20	19	21	53	0,147576859	0,188135111
21	20	20	21	21	21	20	21	20	19	21	19	21	19	19	47	0,121684694	0,20906
21	21	20	20	21	19	19	21	21	19	19	21	21	20	19	48	0,117483268	0,213972667
21	21	20	19	21	21	20	21	21	19	19	21	21	20	19	47	0,121635493	0,209235778
21	21	21	19	20	21	19	20	19	19	19	19	20	20	19	46	0,104431017	0,229732889
20	19	19	21	19	20	21	19	19	19	19	19	19	20	20	49	0,100445597	0,234091333
21	20	19	21	21	20	20	19	20	21	19	20	21	19	19	55	0,119949049	0,211757556
19	20	19	19	20	19	20	21	20	20	19	19	19	19	19	49	0,093145384	0,245161556
21	20	19	21	21	20	19	21	21	19	19	20	21	19	19	46	0,114063778	0,217680444
21	21	20	20	19	21	20	19	19	19	19	20	20	20	19	47	0,106112884	0,226796222
21	19	19	21	20	20	19	19	20	20	19	20	19	19	19	47	0,103797337	0,229902222
21	21	21	21	21	19	20	20	21	21	20	21	21	20	21	49	0,142643623	0,192830667
20	21	21	21	21	21	21	19	21	21	20	21	19	20	19	54	0,136615271	0,195205778
21	21	20	21	19	20	20	21	20	20	21	19	21	20	20	46	0,111036846	0,220802
21	21	21	19	21	21	21	20	19	21	20	21	19	20	20	47	0,127821255	0,203551778
20	19	21	19	19	20	20	21	19	20	19	19	19	20	20	52	0,096134452	0,240872222
21	21	19	21	21	19	21	19	19	21	20	21	19	20	20	53	0,132014177	0,198781333
21	21	20	21	21	21	20	19	21	21	19	21	19	21	21	46	0,14749617	0,188297111
21	21	20	21	20	21	20	20	19	20	21	20	19	19	19	47	0,122456935	0,208453111
21	21	21	21	19	21	21	20	20	20	19	21	19	20	19	53	0,12181876	0,208488444
21	21	19	20	21	21	21	19	20	20	20	20	19	19	19	48	0,123232705	0,207758222
21	20	19	21	21	21	20	20	19	21	20	21	20	20	19	50	0,129504487	0,201429111
21	19	21	21	19	21	19	20	19	20	20	21	20	20	20	46	0,112151559	0,220623111
20	21	19	20	19	19	19	20	20	19	20	19	21	19	19	50	0,101857395	0,233412
20	20	21	21	21	21	21	19	20	21	19	19	21	20	20	51	0,118818879	0,213218889
21	21	20	21	21	19	20	20	19	20	20	21	19	19	19	50	0,124421865	0,206281111
21	21	21	19	21	19	21	20	19	19	19	20	20	20	20	49	0,11226723	0,220339333
20	21	20	19	21	19	19	19	19	19	19	19	21	19	19	46	0,097760712	0,236786222
20	21	21	19	19	20	20	20	19	20	19	20	20	20	20	49	0,098965513	0,236595333
21	21	21	21	21	20	21	21	21	19	21	19	19	21	20	51	0,144847792	0,190919333
21	20	20	21	21	21	20	20	20	21	21	19	21	19	20	50	0,127271623	0,204174
21	21	21	20	19	21	19	21	21	20	20	19	21	19	20	47	0,110884939	0,22097
21	21	19	21	21	21	20	19	19	19	21	19	19	21	19	53	0,144703597	0,192098444
21	21	19	21	19	21	20	19	20	21	21	21	20	20	21	54	0,142391433	0,193054
21	21	21	21	19	20	20	21	20	20	21	19	20	19	19	50	0,114750781	0,216815111
21	21	21	20	21	21	21	19	20	21	19	21	20	20	19	53	0,128263008	0,202259333
19	20	20	20	20	19	19	20	20	20	19	20	20	20	19	47	0,095250084	0,244298889
21	20	20	19	19	21	21	19	20	20	19	19	19	19	20	50	0,105504589	0,227567556
21	21	21	20	20	21	21	19	21	21	21	20	21	20	20	55	0,133430178	0,198647778
21	19	20	21	21	21	19	21	21	21	21	21	19	19	19	50	0,135177526	0,197040222
21	21	21	21	19	21	21	21	19	21	21	19	21	20	19	47	0,122740355	0,208070222
21	21	19	21	19	20	20	19	20	20	20	20	20	19	19	46	0,108705254	0,222457333
21	21	19	19	19	19	20	19	20	19	19	21	20	20	20	47	0,102604298	0,231243778

21	19	21	20	19	19	21	19	19	19	19	19	21	20	20	48	0,09754296	0,237105778
21	21	20	21	19	20	21	19	20	19	21	20	21	19	19	49	0,119142552	0,212182667
21	20	20	19	19	19	20	20	19	19	19	20	21	19	20	48	0,096309852	0,238774889
21	21	20	19	21	21	20	21	19	21	19	21	21	19	20	48	0,120904608	0,21021
21	21	21	21	19	19	19	21	21	19	21	21	20	20	19	47	0,120488496	0,210980889
21	21	20	21	20	20	20	19	21	20	20	19	21	19	19	47	0,115162241	0,215608444
20	21	19	21	19	19	19	20	19	19	19	20	20	19	19	52	0,105222606	0,228826444
21	21	19	20	19	20	19	19	20	20	19	19	19	19	19	54	0,10708095	0,225798889
21	20	21	21	19	20	19	19	20	20	19	20	19	19	19	48	0,102704012	0,230300222
21	21	19	21	19	19	21	19	21	20	20	19	19	20	19	47	0,112701826	0,218153778

Il grafico del fronte di Pareto è mostrato nella Figura 16. La distribuzione delle soluzioni non è molto uniforme in quanto è stato preso in considerazione solo un sottoinsieme delle possibili combinazioni di set-point.

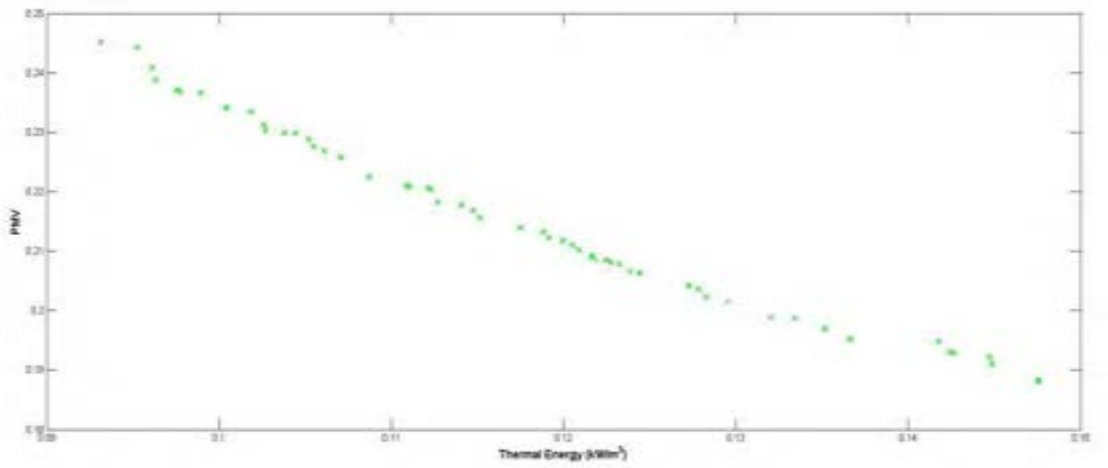

Figura 16. Fronte di Pareto ottimo

6.1.3 Ottimizzazione con NSGA-II supportato da fitness approximation

Per questo problema è stata utilizzata un'ottimizzazione binaria, in quanto è più adatta alla risoluzione di problemi il cui spazio delle decisioni è discreto. I 16 input sono quindi convertiti in una stringa binaria di 10 bit che esprime tutte le 1000 combinazioni di cui abbiamo calcolato le funzioni di fitness. Sull'insieme di stringhe binarie sono applicati i principi genetici precedentemente descritti e quindi riconvertite nei parametri di input iniziali.Per rendere confrontabile l'incremento di efficienza apportato dall'uso dell'approssimazione sono messi a confronto i fronti di Pareto ottenuti con lo stesso numero di richieste di fitness alla funzione reale. Sono stati effettuati test con 120 e 192 chiamate alla fitness.
La Tabella 23 mostra i parametri di set-up dell'algoritmo utilizzati.

Tabella 23. Parametri NSGA-II

Richieste di performance	120
Numero di input	16
Numero di output	2
Tipo di selezione	Torneo
Tipo di crossover	Single Point Binary Crossover
Probabilità di crossover	0,8
Tipo di mutazione	Bitwise
Probabilità di mutazione	0,0625
Popolazione iniziale	10
Generazioni	5

Iterazioni algoritmo	2

Le soluzioni del fronte ottenute sono mostrate nella Tabella 24 e nella Tabella 25.

Tabella 24. Fronte di Pareto NSGA-II con 120 chiamate alla fitness

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Tm	En.Termica	PMV
(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(kW/m^3)	Adim
21	20	20	21	19	21	20	19	21	19	21	21	21	20	21	49	0,138506633	0,198086222
20	20	20	20	19	19	20	20	20	21	21	19	19	20	20	47	0,105464641	0,232003333
21	20	20	21	19	21	20	19	21	19	21	21	21	20	21	49	0,138506633	0,198086222
20	20	20	20	19	19	20	20	20	21	21	19	19	20	20	47	0,105464641	0,232003333
21	20	21	20	20	21	21	21	20	21	19	21	21	19	20	55	0,123256339	0,209109333
21	21	19	20	21	21	19	21	21	19	19	19	20	20	19	47	0,115994544	0,215750667
20	21	20	21	21	21	19	21	21	21	21	19	20	19	19	49	0,129204671	0,203217556
21	20	19	19	19	19	19	19	20	19	21	21	19	20	20	48	0,110084009	0,225342889
20	20	21	21	19	20	19	20	20	19	21	20	19	20	20	46	0,108321745	0,227945333
21	20	21	21	20	20	21	20	21	20	21	20	21	20	21	53	0,139719573	0,197506444
20	19	19	20	20	20	20	19	19	20	19	20	19	20	20	46	0,098559506	0,239803778
20	19	19	20	20	20	20	19	19	20	19	20	19	20	20	46	0,098559506	0,239803778
21	20	20	21	21	20	21	21	21	21	19	19	21	19	21	49	0,135926038	0,200960889
21	21	20	21	21	21	19	21	20	21	20	19	21	20	19	49	0,123860796	0,206557333
21	21	20	21	19	20	20	21	20	20	21	19	21	20	20	46	0,111036846	0,220802
19	19	19	19	19	21	19	20	19	19	20	21	19	20	20	46	0,100526954	0,238324
19	21	19	19	21	21	21	20	20	21	19	21	19	19	20	53	0,121861661	0,212073556
20	19	19	20	19	21	21	20	19	20	19	19	21	20	19	53	0,10562427	0,229856889
20	19	19	20	19	21	21	20	19	20	19	19	21	20	19	53	0,10562427	0,229856889
19	21	21	20	21	21	20	19	19	19	19	20	21	20	20	55	0,116091502	0,218454222
19	20	20	21	21	19	21	21	20	19	20	19	21	19	20	54	0,117318526	0,218204667

Tabella 25. Fronte di Pareto NSGA-II con approssimazione con 120 chiamate alla fitness

Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Tm	En.Termica	PMV
(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(°C)	(kW/m^3)	Adim
21	21	20	21	21	21	19	21	20	21	20	19	21	20	19	49	0,123860796	0,206557333
19	21	19	20	20	21	20	21	20	20	21	21	21	19	20	52	0,122364797	0,213622
21	19	19	21	20	20	19	19	20	20	19	20	19	19	19	47	0,103797337	0,229902222
19	21	19	20	20	21	21	21	19	20	19	20	19	19	19	48	0,107825233	0,227183111
21	19	19	21	21	20	19	21	20	19	21	21	19	21	21	49	0,140846052	0,198000444
20	21	20	19	19	20	19	20	21	20	19	21	21	19	20	48	0,10641995	0,228299111
20	20	21	20	20	21	20	21	20	20	21	19	20	20	19	48	0,112203248	0,223415778
21	20	19	20	21	19	21	21	19	19	21	21	19	19	20	55	0,132010862	0,201594444
21	21	21	20	21	21	21	19	20	21	19	21	20	20	19	53	0,128263008	0,202259333
20	20	19	20	19	20	21	21	19	20	20	20	21	20	20	54	0,110925967	0,224743333
21	21	20	21	20	21	19	19	19	19	19	20	21	20	19	50	0,115405912	0,215689556
21	21	21	19	21	20	21	19	20	21	19	21	21	19	21	55	0,13736955	0,198472
21	21	19	21	19	19	21	19	21	20	20	19	19	20	19	47	0,112701826	0,218153778

La Figura 17 mostra la comparazione tra i fronti ottenuti. Mentre risulta evidente che il best Pareto è ovviamente un fronte migliore di quelli ottenuti con l'ottimizzazione, la differenza tra il fronte NSGA-II con e senza approssimazione non è cosi immediata da individuare graficamente. Da una prima analisi qualitativa si può notare che la distribuzione delle soluzioni in blu (NSGA-II) sono meglio distribuite nello spazio, mentre quelle rosse (NSGA-II con approssimazione) ricadono in numero maggiore sul best Pareto.
La Dominance Ratio nella Tabella 26 mostra che la percentuale di soluzioni di NSGA-II dominate da NSGA-II_approx è maggiore di quelle di NSGA-II_approx dominate da NSGA-II. Come ci si poteva aspettare la percentuale disoluzioni del best Pareto dominare è pari a zero. Lo spacing nellaTabella 27 mostra che la distribuzione di NSGA-II_approx è migliore, anche se di poco, rispetto a NSGA-II. In questo caso da un'analisi qualitativa ci si poteva aspettare che NSGA-II avesse una distribuzione migliore, ma tale metrica tiene conto solo della distribuzione relativa al singolo fronte, quindi fronti più ampi possono avere una distribuzione meno uniforme e quindi un valore di spacing maggiore, rispetto a fronti più ristretti ma con una distribuzione più omogenea. L'ipervolume nella Tabella 28 mostra una leggera predominanza di NSGA-II rispetto a NSGA-II_approx.

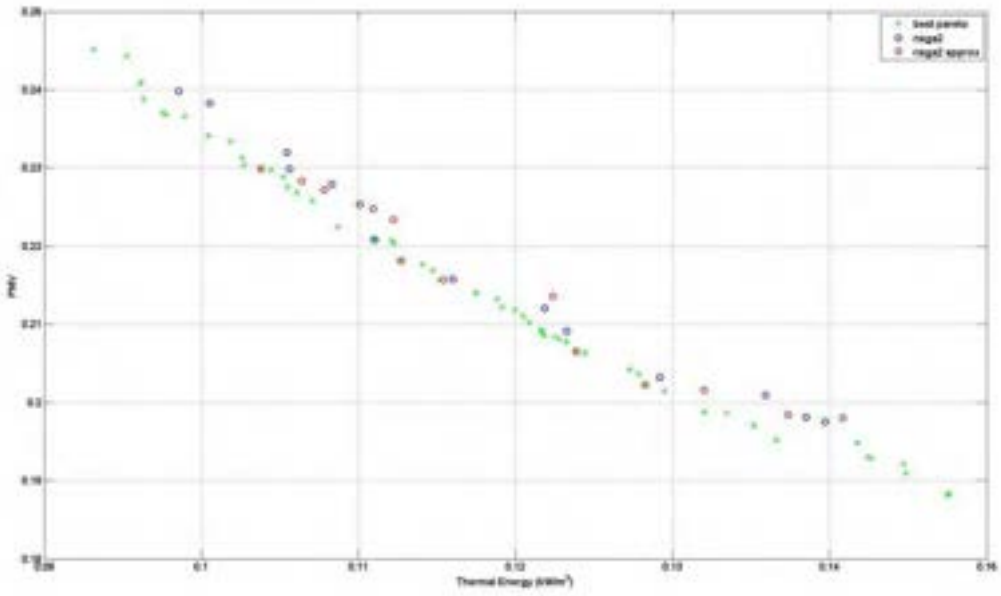

Figura 17. Comparazione dei fronti di Pareto con 120 chiamate alla fitness

Tabella 26. Dominance Ratio 120 richieste di fitness

	Best Pareto	NSGA-II	NSGA-II approx
Best Pareto	x	0	0
NSGA-II	0,8947	x	0,2632
NSGA-II approx	0,6923	0,2308	X

Tabella 27. Spacing 120 richieste di fitness

Best Pareto	NSGA-II	NSGA-II approx
0,0008	0,003	0,002

Tabella 28. Ipervolume120 richieste di fitness

Best Pareto	NSGA-II	NSGA-II approx
0,7349	0,7226	0,7182

I test sono stati effettuati aumentando anche il numero di richieste di fitness, il risultato ottenuto è mostrato nella Figura 18.

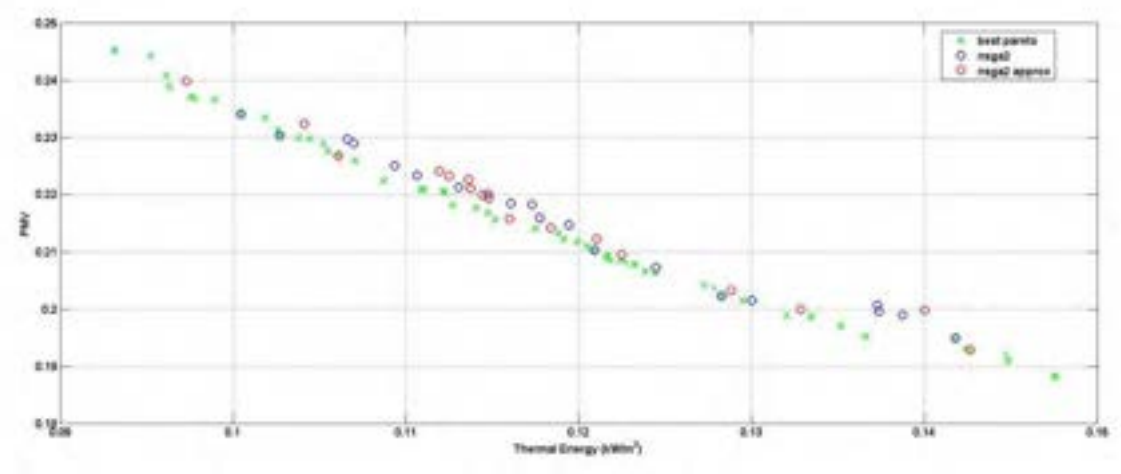

Figura 18. Comparazione dei fronti di Pareto con 192 chiamate di fitness.

Anche in questo caso la metrica di Dominance Ratio nella Tabella 29. Dominance Ratio 192 richieste di fitness conferma che la percentuale di soluzioni di NSGA-II_approx dominate da NSGA-II è minore rispetto al caso reciproco, mentre, a differenza del caso precedente, benché il fronte di NSGA-II_approx sia più ampio di NSGA-II la sua distribuzione di soluzioni è meno uniforme, come ci conferma la metrica dello spacing nella Tabella 30. La metrica dell'ipervolume nella Tabella 31 conferma una lieve predominanza di NSGA-II_approx. Complessivamente si può affermare che il miglioramento è stato ottenuto, ma con margini d'incremento.

Tabella 29. Dominance Ratio 192 richieste di fitness

	Best Pareto	NSGA-II	NSGA-II approx
Best Pareto	x	0	0
NSGA-II	0,8571	x	0,381
NSGA-II approx	0,8824	0,3529	x

Tabella 30. Spacing 192 richieste di fitness

Best Pareto	NSGA-II	NSGA-II approx
0,0018	0,0019	0,0036

Tabella 31. Ipervolume 192 richieste di fitness

Best Pareto	NSGA-II	NSGA-II approx
0,7349	0,7234	0,7275

7 Conclusioni e sviluppi futuri

L'attività si è focalizzata sullo studio l'implementazione il testing e l'applicazione di metodologie di ottimizzazione multiobiettivo supportate dalla fitness approximation a problemi di gestione ottimale degli edifici. In particolare è stata focalizzata l'attenzione sulla gestione del riscaldamento termico dell'edificio F40. L'obiettivo è quello di trovare i set-point delle temperature dei termostati delle zone dell'edificio F40 e della centrale termica per cui viene massimizzato il comfort e minimizzato il consumo di energia termica.
L'approccio dell'ottimizzazione multiobiettivo per tali problemi è molto efficace ed i risultati mostrano che il supporto dell'approssimazione può incrementare la velocità di convergenza all'ottimo degli algoritmi.
Tali risultati sono stati ottenuti sono un'insieme di impostazioni di set-point ottime tra le 1000 combinazioni di set-point calcolate con il simulatore. Di conseguenza l'ottimizzazione non è stata eseguita su un sottoinsieme dello spazio delle variabili di decisione. In prospettiva futura l'obiettivo è quello di interfacciare gli algoritmi di ottimizzazione direttamente col simulatore, senza vincolare l'ottimizzazione ad un limitato spazio di ricerca, in modo tale da esplorare un vasto spazio delle variabili di decisione.

Per quanto riguarda le funzioni obiettivo, nell'attività annuale è stata presa in considerazione la somma delle energia termiche giornaliere relative ad ogni zona, per ottenere un unica funzione di fitness relativa all'intero edificio F40, così come per la funzione di comfort è stato calcolato il valore medio giornaliero del PVM. Tale semplificazione è stata attuata per ridurre il numero di funzioni obiettivo, per poter verificare l'efficacia della fitness approximation in problemi di dimensione non troppo elevata. In prospettiva quindi l'obiettivo è di considerare l'energia primaria, includendo anche quella relativa al consumo elettrico dei fancoils ed arricchire la funzione di fitness del comfort utilizzando anche una funzione di penalità basata sul numero di ore di discomfort.

8 Bibliografia

1 Box, G.E.P. Evolutionary Operation: A Method for Increasing Industrial Productivity. *Journal of the Royal Statistical Society* (1957), 81-101.

2 Miettinen, K.M. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publisher, 1990.

3 Haimes, Y.Y., Lasdon, L.S, and Wismer, D.A. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimisation. *IEEE Transactions on Systems* (1971), 296-297.

4 Schaffer, J. D. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD Thesis, Vanderbilt University, 1984.

5 Schaffer, J. D. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms* (1985), 93-100.

6 Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. *Lecture Notes in Computer Science* (2000), 849-858.

7 Srinivas, N. and Deb, K. Multiobjective optimization using Non-Dominated sorting in genetic algorithms. *Evolutionary Computation* (1994), 221-248.

8 Olhofer, M., Jin, Y, and Sendhoff, B. Adaptive encoding for aerodynamic shape optimization using evolutionary stratgies. *Proceedings of IEEE Congress on Evolutionary Computation* (2001).

9 Shyy, W., Tucker, P. K., and R., Vaidyanathan. *Response surface and neural network techniques for rocket engine injector optimization*. AIAA, 1999.

10 Branke, J., Schmidt, C., and Schmeck, H. Efficient fitness estimation in noisy environment. *Proceedings of Genetic and Evolutionary Computation* (2001), 243-250.

11 Takagi, H. Interactive evolutionary computation. *Proceedings of the 5th International Conference on Soft Computing and Intelligent Systems* (1998), 41-50.

9 Curriculum

Fabio Moretti:

Fabio Moretti è nato a Viterbo il 24/09/1983, ha conseguito il diploma di maturità scientifica nel 2001. Nel 2009 riceve la Laurea Specialistica in Ingegneria Gestionale e dell'Automazione presso l'Università degli Studi "Roma Tre". Sta svolgendo l'attività di Dottorato di ricerca presso la medesima Università in collaborazione con ENEA. I suoi interessi di ricerca riguardano le tecniche di elaborazione delle immagini, metodologie di data fusion per la diagnostica ed il controllo ottimizzato di edifici.

Stefano Panzieri:

Stefano Panzieri è professore associato presso l'Università degli Studi Roma Tre e svolge la sua attività didattica presso le facoltà di Ingegneria Elettronica, Meccanica ed Informatica dove tiene i corsi di Controllo Digitale, Controllo dei Processi e Controlli Automatici.E' il coordinatore del laboratorio di Automatica e del laboratorio di Robotica Autonoma e Fusione Sensoriale. E' un membro di IEEE e del gruppo di lavoro sulle Infrastrutture Critiche. I suoi interessi di ricerca sono nel campo dei sistemi di Controllo Industriale, Teoria dei Sistemi, Sistemi Complessi e CIP. E' autore di molti articoli sperimentali che riguardano i robot mobili, i robot industriali, il controllo con apprendimento iterativo, logica Fuzzy, stima bayesiana, teoria di Dempster-Shafer.

Gruppo di lavoro

Il laboratorio di Modelli per la Protezione di Infrastrutture Critiche (MCIP) è composto da 3 dottorandi ed alcuni tesisti. Le attività principali riguardano la Data Fusion, lo Smart Energy Management, Interdipendency Modeling, sistemi SCADA, modellazione e simulazione di sistemi complessi, algoritmi distribuiti in reti di sensori, controllo decentralizzato del traffico.