Convegno Nazionale
Matematica senza Frontiere
Lecce, 5-8 marzo 2003

Gli Algoritmi genetici

Teresa Donateo
Dipartimento di Ingegneria dell'Innovazione - Università degli Studi di Lecce
teresa.donateo@unile.it

Sommario

Nel processo di evoluzione di una specie, gli individui più adatti a sopravvivere in un particolare ambiente hanno maggiore probabilità di riprodursi. In tal modo i loro caratteri genetici si trasmettono nelle generazioni successive migliorando le caratteristiche della specie. Questo processo è alla base degli algoritmi genetici, un metodo di ottimizzazione del tutto generale che si sta notevolmente diffondendo nel campo della scienza e della tecnica. Gli algoritmi genetici si basano su una semplice rappresentazione matematica del problema e si prestano ad ottimizzare anche sistemi molto complessi. L'obiettivo della presentazione è quello di illustrare la tecnica di ottimizzazione e presentare alcune applicazioni pratiche.

1 Introduzione

Sin dalle origini, l'informatica ha preso spunto dalla biologia cercando di costruire sistemi in grado di emulare il funzionamento del cervello umano (reti neurali), i suoi meccanismi di apprendimento (apprendimento automatico) e l'evoluzione biologica (calcolo evolutivo).
L'idea alla base del calcolo evolutivo, di cui gli algoritmi genetici sono solo un esempio, è far evolvere una popolazione di soluzioni candidate per un dato problema usando operatori che si ispirano alle leggi della genetica e alla selezione naturale della specie.
Gli algoritmi genetici sono stati sviluppati negli anni '60 da John Holland [1], dell'università del Michigan. I GA consentono di trovare la soluzione ad un problema tra un numero enorme di possibili alternative utilizzando un modello del problema da risolvere.
È noto che, nell'evoluzione delle specie biologiche, gli individui più "idonei" sono favoriti per la sopravvivenza e la riproduzione. In questo modo, il loro materiale genetico si trasmette alle generazioni future. L'idoneità di un organismo biologico dipende dalla sua capacità di sopportare le caratteristiche dell'ambiente e di competere o cooperare con gli altri organismi circostanti. Basandosi sull'analogia biologica, negli algoritmi genetici si assume che una possibile soluzione di un problema possa essere rappresentata come un set di parametri (detti geni), i quali, uniti, formano una stringa di valori denominata "cromosoma". Holland per primo ha dimostrato, ed è ancora accettato

da molti, che la codifica migliore è la rappresentazione della stringa in alfabeto binario. Ad esempio, se si vuole rappresentare una funzione di tre parametri $F(x, y, z)$, è possibile codificare ogni variabile con un numero binario di 10 bit. Il cromosoma associato alla soluzione sarà formato da tre geni e sarà composto da 30 cifre digitali.
L'analogia biologica negli algoritmi genetici si estende, infatti, anche alla terminologia utilizzata:

Individuo:	Possibile soluzione del problema
Popolazione:	Insieme delle soluzioni su cui ricercare contemporaneamente la soluzione
Cromosoma:	Codifica della soluzione nell'alfabeto scelto
Gene:	Codifica di un particolare elemento della soluzione candidata (es. un parametro)
Allele:	Possibile valore di ogni bit della stringa
Fitness o idoneità:	Capacità di risolvere in modo ottimale il problema

Gli operatori fondamentali utilizzati dall'algoritmo genetico sono: codifica, valutazione, selezione, *crossover* e mutazione.
Nella codifica, si associa ad ogni individuo un cromosoma in funzione del set di parametri che lo caratterizzano.
L'operatore di valutazione calcola l'idoneità di ciascun individuo utilizzando un modello del problema da risolvere.
Nella selezione si scelgono i cromosomi destinati alla riproduzione; più l'idoneità dell'individuo è alta, più spesso il suo cromosoma sarà scelto per la riproduzione.
La riproduzione è effettuata mediante il *crossover*, ovvero incrociando il cromosoma degli individui "genitore" per generare la prole; l'incrocio può essere effettuato in diversi modi, il più semplice è quello di scegliere casualmente una posizione nella stringa e scambiare le due parti in cui risulta diviso il cromosoma dei genitori.
Una volta generati gli individui "figlio", applicando l'operatore di mutazione si effettua la variazione di ogni bit di ogni cromosoma con una probabilità molto bassa; lo scopo di questo operatore è introdurre una certa variazione nel materiale genetico della popolazione durante la sua evoluzione.
Per migliorare l'efficienza della ricerca si può decidere di salvare nella nuova generazione l'individuo migliore della generazione precedente, preservandolo con un'operazione che prende il nome di elitismo.
Per illustrare più in dettaglio il funzionamento del GA, nel paragrafo successivo è riportato un esempio di applicazione ad un semplice problema singolo-obiettivo.
Nel corso dello sviluppo degli algoritmi genetici sono stati ideati diversi espedienti, es. codifiche non binarie, modalità alternative di *crossover*, tecniche di elitismo, rimappamento, ecc., per migliorare la convergenza dell'algoritmo genetico verso l'ottimo assoluto. Per una descrizione dettagliata di tali metodi si rimanda alla letteratura specializzata [2,3]. Per aumentare l'effi-

cienza dell'algoritmo genetico nell'individuare l'ottimo assoluto è necessario garantire un'ampia variabilità nella popolazione. Un modo per raggiungere questo scopo è usare un numero molto ampio di individui nella popolazione e utilizzare l'operatore di mutazione per esplorare nuove possibilità. In alternativa si può usare un approccio sviluppato da Krishakumar [4] e denominato micro-GA (μGA) che consiste nell'utilizzare un numero molto ridotto di individui (ad es. 5) e verificare la convergenza della micropopolazione. Se la differenza in bit tra tutti gli individui della generazione è minore di un valore prefissato (ad es. 5%) si può ritenere che la popolazione sia arrivata a convergenza. Quando ciò si verifica, l'algoritmo riparte con una nuova popolazione in cui si preserva l'individuo migliore della popolazione arrivata a convergenza mentre gli altri individui sono generati in modo casuale.

2 Un semplice algoritmo genetico

Supponiamo di voler massimizzare la seguente funzione nel campo dei numeri naturali:

$$f(n) = -n^2 + 256n \qquad 0 \leq n \leq 255 \tag{1}$$

Il primo passo da effettuare è la codifica dell'unico parametro n; ad esempio è possibile rappresentare n con una stringa di otto bit coincidente con la sua rappresentazione in alfabeto binario.

$$\begin{array}{lcl} 00000000 & \longrightarrow & 0 \\ 00000001 & \longrightarrow & 1 \\ 00000010 & \longrightarrow & 2 \\ \quad \ldots & & \\ \quad \ldots & & \\ 11111111 & \longrightarrow & 255 \end{array}$$

L'algoritmo genetico parte generando una popolazione casuale di N individui, ognuno dei quali è rappresentato da un diverso valore della variabile n. Ad esempio si può considerare una popolazione di 5 individui:

$$\begin{array}{lcl} 00000000 & \longrightarrow & 0 \\ 00001111 & \longrightarrow & 15 \\ 00000111 & \longrightarrow & 7 \\ 11111111 & \longrightarrow & 255 \\ 11000011 & \longrightarrow & 195 \end{array}$$

Per ciascuno degli individui generati, si calcola il fitness in base al modello del problema da risolvere; nel caso in esame il fitness può essere definito in modo molto semplice attraverso il valore assunto dalla funzione (1) in corrispondenza del valore di n associato all'individuo:

		n		$f(n)$
00000000	$\longrightarrow$	0	$\longrightarrow$	$-n^2 + 256n = 0$
00001111	$\longrightarrow$	15	$\longrightarrow$	$-n^2 + 256n = 3615$
00000111	$\longrightarrow$	7	$\longrightarrow$	$-n^2 + 256n = 1743$
11111111	$\longrightarrow$	255	$\longrightarrow$	$-n^2 + 256n = 255$
11000011	$\longrightarrow$	195	$\longrightarrow$	$-n^2 + 256n = 11895$

Il passo successivo è la scelta degli individui da riprodurre, scelta che deve essere effettuata in modo tale da favorire gli individui a più alto fitness, senza escludere però completamente gli altri per garantire uno sfruttamento ottimale del materiale genetico della popolazione. Un modo molto semplice per fare ciò è assegnare a ciascun individuo una probabilità di riproduzione pari al rapporto percentuale tra il suo fitness e la somma dei fitness di tutti gli individui della generazione:

		n		$f(n)$		*probabilità*	
00000000	$\longrightarrow$	0	$\longrightarrow$	$-n^2 + 256n = 0$	$\longrightarrow$		0%
00001111	$\longrightarrow$	15	$\longrightarrow$	$-n^2 + 256n = 3615$	$\longrightarrow$		21.5%
00000111	$\longrightarrow$	7	$\longrightarrow$	$-n^2 + 256n = 1743$	$\longrightarrow$		8.5%
11111111	$\longrightarrow$	255	$\longrightarrow$	$-n^2 + 256n = 255$	$\longrightarrow$		1.5%
11000011	$\longrightarrow$	195	$\longrightarrow$	$-n^2 + 256n = 11895$	$\longrightarrow$		68.5%
			somma	17238			100%

Nella fase di *crossover* si scambiano parti delle stringhe dei genitori per formare gli individui "figlio". Nel caso più semplice di "single-point" *crossover* si tagliano le stringhe in una posizione scelta a caso, per produrre due segmenti "testa" e due segmenti "coda". I segmenti testa sono poi scambiati per produrre due nuovi cromosomi di lunghezza completa. Il crossover non è abitualmente applicato a tutte le coppie di individui selezionati per l'accoppiamento, ma con una certa probabilità (tipicamente tra 0.6 e 1.0). Se il crossover non è applicato i figli sono generati semplicemente duplicando i genitori.
Si supponga di selezionare per la riproduzione il secondo e il quinto individuo e di incrociarli sul quarto bit. Per il secondo processo di riproduzione potrebbero essere selezionati il secondo e il terzo individuo, i cui cromosomi possono essere scambiati dal sesto bit:

genitori		*figli*
00001111	$\longrightarrow$	11001111
11000011	$\longrightarrow$	00000011
genitori		*figli*
00000111	$\longrightarrow$	11000111
11000011	$\longrightarrow$	00000011

A questo punto è possibile prendere entrambi i figli oppure uno solo. In ogni caso, il cromosoma dei figli è soggetto a mutazione ovvero ogni gene è modificato con una probabilità bassa (tipicamente 0.001). La mutazione serve ad inserire un po' di "casualità" nella ricerca per assicurare che nessun punto nello spazio abbia probabilità nulla di essere esaminato. Nell'esempio in esame, facendo riferimento alla prima coppia di figli generati, si supponga di mutare solo il primo bit del secondo figlio:

genitori		*figli*		*mutazione*
00000111	$\longrightarrow$	11000111	$\longrightarrow$	11000111
11000011	$\longrightarrow$	00000011	$\longrightarrow$	10000011

Il processo di generazione dei figli continua fino a quando il numero di individui generati uguaglia il numero iniziale di individui nella popolazione. In questo modo si completa la nuova generazione e il processo si ripete attraverso le fasi di valutazione, selezione, incrocio, mutazione, fino a che non si raggiunge il criterio di stop desiderato. Si può decidere di fermare l'algoritmo quando si ottiene la convergenza della popolazione, quando si raggiunge un determinato grado di idoneità oppure quando è stato effettuato un numero prefissato di generazioni.

Figura 1: Schema di funzionamento di un algoritmo genetico

Per convergenza della popolazione si intende la progressione verso la crescente uniformità. Assumendo che un gene converga quando il 95% della popolazione condivide lo stesso valore si può ritenere che la popolazione converga quando tutti i geni convergono. Per quanto riguarda il grado di idoneità raggiunto, è evidente che questo criterio di stop si può utilizzare esclusivamente quando è noto il valore massimo che può assumere il fitness. In alternativa si può interrompere il GA quando l'incremento di fitness tra una generazione e la successiva è inferiore ad una predeterminata quantità [5].

3 Confronto con altre tecniche di ottimizzazione

In applicazioni particolari alcune tecniche adatte a risolvere problemi di ricerca e ottimizzazione possono comportarsi meglio dei GA ovvero consentire una più veloce convergenza verso l'ottimo. Il vantaggio principale degli algoritmi genetici rispetto a tali tecniche è costituito dal fatto che il loro funzionamento prescinde completamente dal comportamento e dalle caratteristiche del problema da risolvere. Gli unici collegamenti tra l'algoritmo genetico e il sistema da ottimizzare, infatti, sono rappresentati dalla codifica dei parametri e dal calcolo del fitness attraverso il modello del problema.
Inoltre, gli algoritmi genetici sono dotati di un'implicita caratteristica di parallelismo, per il fatto che ricercano su una popolazione di individui, e non richiedono alcuna proprietà matematica sulla funzione di fitness.
In generale si può affermare che un algoritmo di ottimizzazione è efficiente se utilizza due tecniche per trovare il massimo globale: l'esplorazione, per esaminare nuove e sconosciute aree dello spazio di ricerca, e lo sfruttamento, ricavando informazioni dai punti precedentemente analizzati per indirizzare la ricerca. A differenza degli altri metodi, un GA combina insieme esplorazione e sfruttamento allo stesso tempo e in modo ottimale.
Di seguito sono riportati alcuni esempi di tecniche di ottimizzazione.

Ricerca casuale

L'approccio più banale per l'ottimizzazione di funzioni complicate è una ricerca casuale o enumerata. In questo caso i punti nello spazio di ricerca sono scelti a caso o in qualche maniera sistematica, e il loro valore calcolato attraverso la funzione stessa.

Metodi del gradiente

Sono stati inventati diversi metodi che sfruttano informazioni sul gradiente della funzione per guidare la direzione della ricerca. Questi metodi, chiamati *hillclimb* (scalata), funzionano bene con funzioni continue e dotate di un solo picco (unimodali). Infatti, nel caso di funzioni multimodali questi metodi possono convergere su ottimi localisenza mai raggiungere l'ottimo assoluto.
Un esempio di metodo basato sul calcolo del gradiente è quello utilizzato da Catalano e Dadone [6] e denominato "ottimizzazione progressiva". Il metodo si basa sul calcolo delle cosiddette "derivate di sensitività" ottenute perturbando uno ad uno i parametri di progetto, e calcolandone l'effetto sulla funzione obiettivo. Nell'approccio utilizzato da Catalano et al. [6] si impongono grosse variazioni per le variabili la cui derivata mantiene sempre lo stesso segno, in quanto tali soluzioni si ritengono lontane dal punto di ottimo. Quando, invece, la derivata cambia di segno, il parametro in esame è variato con un passo molto piccolo.

Ricerca iterata

I metodi della ricerca casuale e quello del gradiente possono essere combinati in modo che, una volta individuato un ottimo locale, la scalata inizia nuovamente da un altro punto scelto a caso. Questo metodo, chiamato ricerca iterata, garantisce l'esplorazione di tutto il dominio di ricerca ma non consente comunque di definirne la forma in quanto ogni prova è fatta isolatamente. Nel corso dell'applicazione del metodo, infatti, tutte le regioni del dominio di ricerca hanno la stessa probabilità di essere esplorate indipendentemente dal loro valore di fitness. Nei GA, invece, nella ricerca dell'ottimo si assegnano maggiori probabilità di ricerca alle regioni con più alto fitness. D'altra parte se il massimo si trova in una piccola regione circondata su tutti i lati da regioni con basso fitness, il metodo della ricerca iterata può avere le stesse probabilità di trovare l'ottimo rispetto ad un GA, e quindi è da preferire per la sua semplicità.

Simulated Annealing

Si tratta di una versione modificata del metodo di hillclimbing, sviluppata da Kirkpatrick [7], nella quale si parte da un punto scelto a caso nel dominio, e successivamente si effettua un movimento casuale. Se il movimento porta a un incremento del fitness esso è accettato, in caso contrario è accettato con probabilità p(t), dove t è il tempo. All'inizio p(t) è vicino al valore 1, ma gradualmente tende a zero, ovvero la probabilità di accettare un movimento negativo diminuisce. A volte movimenti negativi sono necessari per evitare massimi locali, ma se sono eccessivi possono allontanare dal massimo la direzione della ricerca. Comunque come la tecnica della ricerca casuale, lavora con solo una soluzione candidata per volta e perciò non costruisce una figura complessiva dello spazio di ricerca e non consente di sfruttare le informazioni ottenute ai precedenti movimenti per guidare la ricerca verso la soluzione.

Metodi statistici (DOE)

I metodi statistici si basano sull'effettuazione di una serie di prove in cui si variano le variabili di input di un problema in modo da osservare e quantificare le corrispondenti variazioni della risposta del sistema.
Nel caso di ricerca di una soluzione ottimale di un problema funzione di n variabili, si suddivide il campo di ricerca su un numero di livelli m e si analizzano tutte le possibili combinazioni delle variabili (piano fattoriale completo) o un numero ridotto di prove ottenute variando più parametri alla volta (piano fattoriale ridotto). I risultati ottenuti con le prove consentono di individuare la significatività di ciascuna variabile sul problema in esame e quindi indicano la direzione in cui muoversi per raggiungere l'ottimo. Inoltre, utilizzando il DOE è possibile individuare una relazione analitica tra l'obiettivo in esame e i parametri considerati, tale che interpoli i risultati

delle prove e consenta l'applicazione di altri metodi di ottimizzazione [8]. Tale relazione è detta *superficie di risposta* [9].

Reti neurali

In alternativa al DOE è possibile utilizzare tecniche basate sulle reti neurali per costruire un modello complesso del sistema che consenta di studiarne il funzionamento al variare dei parametri di controllo. Applicando questo metodo, Carlucci et al [10] hanno costruito un modello in grado di prevedere il livello di rumore di un motore in funzione dei parametri di iniezione su un ampio range di condizioni operative. Altri esempi di applicazione delle reti neurali all'ottimizzazione dei motori a combustione interna possono essere trovati in [11,12].
Il modello ottenuto con la rete neurale è più complesso e realistico rispetto a quello sviluppabile con un metodo DOE ma richiede, per l'addestramento della rete, un numero di esperimenti che aumenta notevolmente al crescere delle variabili di input e di output. Una volta ottenuto il modello del sistema l'applicazione alla ricerca dell'ottimo è analoga al caso della superficie di risposta. Mentre un algoritmo genetico può essere applicato ad ogni problema senza alcuna conoscenza aggiuntiva e senza modificarne la struttura, la validità di una rete neurale dipende da alcune variabili come il numero di *hidden layers*, neuroni e funzioni di attivazione usate durante l'addestramento. Pertanto, ogni applicazione richiede una accurata analisi del problema in esame per costruire una rete neurale appropriata alla sua risoluzione.

Metodo della Superficie di Risposta (RSM)

Questo metodo, basato sulla combinazione di *hillclimbing* e DOE, cerca da punto a punto lungo una ipersuperficie definita dalla funzione obiettivo in relazione ai parametri di controllo [13-14]. Se un problema è modellabile matematicamente, la superficie di risposta è nota in quanto è rappresentata dal modello matematico stesso. In caso contrario si può utilizzare un metodo DOE con risoluzione ridotta per adattare una ipersuperficie modello ai dati sperimentali acquisiti con la sperimentazione. Successivamente si applica il metodo del gradiente per determinare la direzione di ricerca identificando i punti nella direzione di maggiore miglioramento, fino a quando la funzione obiettivo subisce un cambiamento di segno della derivata, indicando che si è raggiunto un ottimo locale. Una volta individuato l'ottimo locale si applica un nuovo DOE per studiare il dominio intorno ad esso e si ricalcola l'ipersuperficie. Se tale superficie mostra gradienti elevati, l'ottimizzazione continua altrimenti si assume che esista nelle vicinanze un ottimo locale e si effettuano ulteriori esperimenti per ricercare il valore esatto dell'ottimo.
In uno studio di Thiel et al. [15] è stato effettuato un confronto diretto tra questa tecnica e un algoritmo genetico micro-GA nell'ottimizzazione dei

parametri di controllo di un motore diesel. I risultati del confronto hanno dimostrato che, anche se la tecnica RSM ha consentito di individuare l'ottimo in un tempo di calcolo ridotto rispetto al GA, l'uso del RSM richiede l'intervento continuo dell'utente per analizzare i dati ottenuti da ogni applicazione del metodo del gradiente e pianificare il nuovo DOE per continuare la ricerca. Tenendo conto di tale tempo aggiuntivo, Thiel et al concludono che il micro-GA è vantaggioso in termini di efficienza e semplicità di applicazione. Bisogna ricordare che, come tutti i metodi basati sul gradiente, il metodo della superficie di risposta non è in grado di distinguere tra un ottimo locale e un ottimo globale per cui si ha maggiore probabilità di rimanere vincolati su ottimi locali.

4 Il fondamento matematico

La spiegazione matematica del funzionamento degli algoritmi genetici è fornita dal teorema dello schema di Holland [1] il cui enunciato è: *"short, low order, above average schemata receive exponentially increasing trials in subsequent generations of genetic algoritm"*. In pratica si suppone che un individuo abbia un alto valore di fitness quando possiede buoni schemi nel suo cromosoma. Nel funzionamento del GA il numero di opportunità che ogni individuo riceve è in proporzione al suo fitness; quindi, i migliori individui forniscono gli schemi migliori ai geni della generazione successiva, aumentando la probabilità di trovare soluzioni ottime. Per spiegare meglio l'enunciato del teorema è necessario introdurre alcune definizioni.
Per schema si intende un modello dei valori possibili assunti da un gene che consenta di quantificare il grado di somiglianza tra diversi geni.
Uno schema può essere costruito introducendo il simbolo ♯ nell'alfabeto usato. Nel caso di codifica binaria, i valori possibili di un gene possono essere rappresentati da una stringa di caratteri dell'alfabeto $\{0, 1, \sharp\}$, dove simbolo jolly ♯ indica 0 o 1. In pratica, un cromosoma contiene gli schemi ottenuti sostituendo col simbolo "♯" uno o più dei suoi bit. Ad esempio, lo schema ♯1100101 è contenuto nei due cromosomi, 01100101 e 11100101, mentre lo schema ♯11001♯1 può essere contenuto in quattro cromosomi. In pratica un cromosoma di lunghezza L può contenere 2^L schemi; viceversa, se uno schema possiede N volte il carattere jolly esistono 2^N cromosomi che soddisfano lo schema. Si definisce ordine, O, di uno schema il numero di simboli diversi da "♯" in esso contenuti, ad es. $O\,(\sharp 11001\sharp 1) = 6$.
Si chiama lunghezza definita, δ , di uno schema la distanza tra i simboli diversi da "♯" più esterni, ad esempio $\delta\,(\sharp 11001\sharp 1) = 6$.
Il fitness di uno schema alla generazione t è definito dalla seguente relazione:

$$Fitness(schema, t) = \sum_{i=1}^{Nm(t)} Fitness(crom_i, t)/Nm(t) \qquad (2)$$

dove $crom_i$ rappresenta l'i-simo degli N_m individui che soddisfano lo schema alla generazione t.

Il numero di cromosomi che soddisfano lo schema nella generazione successiva può essere calcolato come:

$$N_m(t+1) = N_m(t) \cdot Fitness(schema, t)/Fitness(popolazione, t) \qquad (3)$$

dove $Fitness(popolazione, t)$ rappresenta il fitness medio della generazione t.
Questa regola stabilisce che gli schemi con fitness al di sopra della media hanno maggiore probabilità di essere riprodotti nella generazione successiva; al contrario il numero di riproduzioni dello schema diminuisce quando questo ha fitness basso. Più in dettaglio, se uno schema ha un fitness che supera di $\epsilon\%$ il valore medio, allora il numero di riproduzioni alla generazione t può essere calcolato a partire dal suo valore alla generazione 0 in base alla seguente relazione:

$$N_m(t) = N_m(0) \cdot (1+\epsilon)^t \qquad (4)$$

La relazione (4) mostra che, man mano che evolvono le generazioni, gli schemi con fitness al di sopra della media subiscono un aumento esponenziale del numero di cromosomi che li contengono. Tale osservazione è alla base della cosiddetta "*building block hypothesis*" e spiega il teorema di Holland: l'algoritmo genetico ricerca la soluzione ottimale attraverso il confronto di schemi corti, a basso ordine e con alto fitness, denominati blocchi costruttivi della soluzione.

5 Ottimizzazione multiobiettivo

In campo automobilistico la possibilità di effettuare una ottimizzazione multi-obiettivo è molto importante in quanto gli obiettivi da raggiungere (minimizzazione dei consumi, riduzione delle emissioni, ottimizzazione delle prestazioni) sono tra di loro competitivi.
Per chiarire questo concetto, si consideri il caso di un motore Diesel ad iniezione diretta e si ponga come obiettivo quello di ridurre le emissioni di particolato e di NO_x, agendo sull'anticipo iniezione. È noto che ritardando l'anticipo di iniezione si ottiene una riduzione degli ossidi di azoto e un aumento del particolato per cui l'identificazione del valore ottimale dell'anticipo di iniezione è un tipico esempio di problema multi-obiettivo. Poiché lo scopo dell'ottimizzazione multiobiettivo è quello di trovare un buon compromesso tra i diversi obiettivi da raggiungere, un possibile metodo di ottimizzazione potrebbe essere quello di combinare le varie funzioni obiettivo in un'unica funzione di fitness e trovare la condizione di funzionamento "ottimale". Un esempio di applicazione di questo metodo è riportato nei numerosi lavori effettuati da Reitz, Senecal e i loro collaboratori [17-18] nell'ottimizzazione dei motori a combustione interna.

È importante sottolineare che, nell'approccio di Reitz, si definisce a priori l'importanza relativa di un obiettivo rispetto all'altro, ad es. delle emissioni di NO_x rispetto al particolato, e quindi si definisce un percorso preferenziale vincolato nella ricerca dell'ottimo. Questo può essere un limite in quanto nuovi sviluppi nella tecnica, o diversi tipi di applicazione, potrebbero modificare completamente l'importanza degli obiettivi. Se, ad esempio, si riuscisse a realizzare una trappola per il particolato a basso costo ed elevata efficienza, le emissioni di NO_x diventerebbero il parametro critico da ottimizzare e tutto il processo di ottimizzazione precedentemente svolto, andrebbe rifatto.
Per eliminare questa limitazione, è possibile utilizzare altri sistemi basati sulla combinazione dei fitness mediante pesi variabili nel corso del run di ottimizzazione. In questo modo, l'importanza di un obiettivo rispetto all'altro cambia continuamente durante il processo di ottimizzazione e la direzione di ricerca risulta meno vincolata [19].
Un'altra alternativa è confrontare le diverse funzioni obiettivo mediante il criterio di Pareto per il confronto di vettori. Questo metodo, da Fonseca [20] è ampiamente utilizzato nella risoluzione di problemi multiobiettivo [21,22] ed è illustrato nei paragrafi successivi.

Il criterio di Pareto

Nel caso di ottimizzazione multiobiettivo, ad ogni individuo può essere associato un vettore di fitness le cui componenti sono i fitness calcolati rispetto ai singoli obiettivi da ottimizzare. Per confrontare i vettori di fitness degli individui si può utilizzare il criterio di confronto di Pareto, in base al quale un vettore x si dice parzialmente minore di $(< p)y$ se sussiste la relazione:

$$x < py \iff \forall i(x_i \leq y_i) \wedge \exists i(x_i < y_i) \qquad (5)$$

In queste condizioni si dice che il vettore y domina il vettore x.
Si può applicare questa definizione agli individui della popolazione generata dall'algoritmo genetico. Se un individuo produce un vettore di fitness che non è dominato da nessun altro, l'individuo si dice *non dominato* o *non inferiore*.
Si supponga di voler massimizzare contemporaneamente due funzioni F_1 e F_2 e di avere sei possibili soluzioni del problema, A, B, C, D, E ed F.
È possibile rappresentare nel piano $F_1 - F_2$ il valore delle funzioni in corrispondenza di tali soluzioni, come riportato in Figura 2; In base alle definizioni date, si può affermare che i punti E, C and D sono non dominati, mentre A, F e B sono dominati.

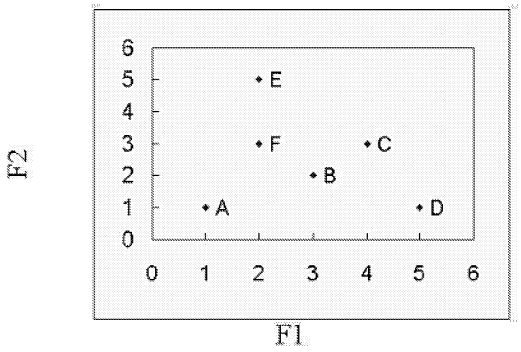

Figura 2: Esempio di ottimizzazione multiobiettivo

Utilizzando questo approccio, lo scopo dell'algoritmo genetico diventa quello di trovare tutti i punti non dominati nello spazio definito dalle singole funzioni obiettivo. Questi punti costituiscono il cosiddetto fronte di Pareto. Una volta individuato il fronte di Pareto è possibile scegliere tra tutti i punti quello che dà il migliore compromesso. In questo modo, l'importanza da assegnare ad un obiettivo rispetto ad un altro può essere stabilita a posteriori e può essere modificata in qualsiasi momento senza dover ripetere il processo di ottimizzazione.
Zitzler et al [23], nel confrontare diverse tecniche per realizzare l'ottimizzazione multiobiettivo ha evidenziato l'importanza dell'operatore elitismo. In un problema singolo-obiettivo, è sufficiente salvare un unico individuo, il migliore. Nel caso di problemi multiobiettivo, invece, l'applicazione dell'elitismo diventa critica perché è necessario salvare un set di individui, ad esempio tutti quelli non dominati. In questi casi, l'elitismo si effettua, in genere, utilizzando una popolazione esterna, costituita dalle soluzioni da preservare, e scambiando gli individui tra la popolazione esterna e la popolazione che evolve nel GA [23,24]. L'elemento chiave di tale approccio è definire quali soluzioni devono essere salvate nella popolazione esterna e per quanto tempo, oltre che stabilire quando e come gli elementi della popolazione esterna devono essere inseriti nella popolazione che evolve. Questo problema è particolarmente critico nel caso di un micro-GA a causa del numero limitato di individui nella popolazione [24].

6 Esempi di applicazione

Il metodo degli algoritmi genetici è stato utilizzato per definire la forma ottimale della camera di combustione di un motore diesel ad iniezione diretta di tipo common rail in modo da ridurre le emissioni inquinanti e migliorare le prestazioni. La valutazione delle camere analizzate è stata effettuata con una versione modificata del codice fluidodinamico KIVA3V arricchita

rispetto alla versione standard di modelli più accurati per i fenomeni di combustione e spray [25]. Il codice è in grado di descrivere i complessi fenomeni termo-fluidodinamici che avvengono nei moderni motori diesel ad iniezione diretta e valutare i livelli di emissione e le prestazioni del motore al variare della geometria e dei parametri di controllo.
La forma della camera di combustione, ed in particolare la parte ricavata all'interno del pistone, è stata schematizzata in funzione di cinque parametri geometrici riportati in Figura 3.
L'abbinamento tra il GA e il codice CFD è stato effettuato in base al seguente diagramma di flusso:

1. L'algoritmo genera una serie di camere di combustione assegnando valori casuali ai cinque parametri geometrici.
2. Per ogni combinazione di parametri geometrici utilizzati si realizza una griglia di calcolo che rappresenta una particolare forma della camera di combustione.
3. Su ogni griglia sono effettuati tanti run con il codice KIVA3V quante sono le condizioni operative sulle quali si vuole ottimizzare il motore.
4. Gli output delle simulazioni (pressione interno cilindro ed emissioni inquinanti) sono trasmessi all'algoritmo genetico.
5. In base agli output della simulazione, si definisce la bontà (rango) di ciascuna camera di combustione utilizzando il criterio di Pareto.
6. Le camere migliori sono selezionate per la fase di riproduzione e si genera una nuova popolazione.
7. Si ripetono tutti i passi a partire dal 2) fino a che non si raggiunge un numero prefissato di generazioni.

Come in tutti i problemi multiobiettivo, i risultati dell'ottimizzazione sono rappresentati dalle soluzioni non dominate cioè dalle camere che giacciono sul fronte di Pareto nell'iperspazio definito dalle funzioni obiettivo scelte.
Nel caso dell'ottimizzazione dei motori diesel, gli obiettivi più importanti da raggiungere sono l'abbattimento degli ossidi di azoto e la riduzione del fumo. Infatti, i meccanismi di formazione di tali inquinanti sono tra loro competitivi per cui risulta molto complesso ridurre contemporaneamente NO_x e particolato. In tali i risultati ottenuti con l'algoritmo genetico possono essere rappresentati nel piano definito dalle funzioni obiettivo F_1 e F_2 (vedi Figura 4). Tali funzioni obiettivo quantificano la capacità di ciascuna camera di ridurre le emissioni di NO_x e particolato.
In Figura 4 sono anche riportate cinque geometrie ottimizzate della camera di combustione e la camera baseline usata come riferimento per l'ottimizzazione.

Una delle camere individuate dall'algoritmo genetico è stata realizzata e testata sperimentalmente. I risultati del test hanno consentito di verificare la bontà della camera, e di validare la procedura di ottimizzazione.
Altre applicazioni del metodo ai motori diesel ad iniezione diretta hanno riguardato l'ottimizzazione della strategia di iniezione e del profilo di iniezione [26-28].

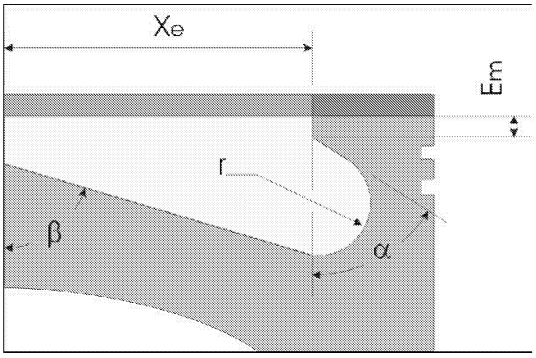

Figura 3: Schematizzazione della camera

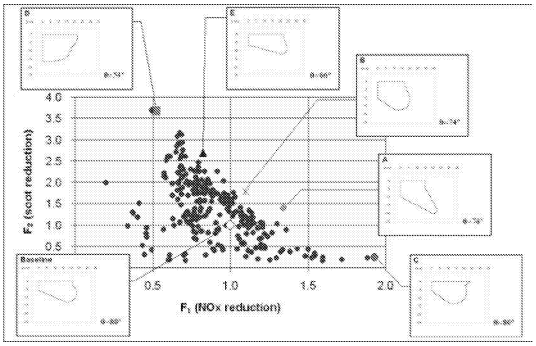

Figura 4: Camere ottimizzate individuate dall'algoritmo genetico

7 Conclusioni

Gli algoritmi genetici sono metodi di ottimizzazione in grado di risolvere problemi complessi in modo semplice. Gli operatori utilizzati dai GA sono basati sull'analogia biologica con i processi naturali di riproduzione e selezione della specie. Rispetto agli altri metodi di ottimizzazione hanno il vantaggio di funzionare indipendentemente dalla natura del problema e di

possedere caratteristiche intrinseche di parallelismo (esplorazione) oltre alla capacità di sfruttare le informazioni ricavate dalle soluzioni già individuate per guidare la ricerca dell'ottimo(sfruttamento).
Nelle applicazioni più immediate gli algoritmi genetici consentono trovare la soluzione ottimale di un problema singolo-obiettivo ma possono essere anche applicati a sistemi complessi che richiedono l'ottimizzazione contemporanea di più obiettivi. In tale caso i risultati migliori si ottengono applicando alle funzioni da ottimizzare il criterio di confronto di Pareto.
Esempi di applicazione degli algoritmi genetici si possono ritrovare in numerosi i campi della tecnica e della scienza, oltre all'economica, le scienze sociali, ecc.. Nel caso di motori diesel ad iniezione diretta gli algoritmi genetici sono stati utilizzati per individuare forme ottimali per la camera di combustione, ottimizzare il profilo di iniezione e individuare strategie di controllo del motore in grado di ridurre le emissioni e migliorare il consumo specifico. L'efficacia dell'utilizzo degli algoritmi genetici nell'ottimizzazione dei motori è stata verificata testando sperimentalmente le soluzioni individuate.

Riferimenti bibliografici

[1] Holland, J.H., Adaptation in natural and artificial systems, Ann Arbor: the University of Michigan Press, 1975;

[2] Goldberg, D.E., Genetic Algorithms in Search, Optimization & Machine Learning, Addison-Wesley, 1989;

[3] Mitchell, M., Introduzione agli algoritmi genetici, Apogeo Scientifica, 1998;

[4] Krishakumar, K., "Micro-genetic Algoritms for Stationary and Non-stationary Function Optimization", SPIE Intelligent Control and Adaptative System, 1196, Philadelphia, 1989, pp. 289-296;

[5] Kim., B.M., Kim, Y.B., Oh, C.H., "A Study of the Convergence of Genetic Algorithms", Computers ind. Eng Vol. 33, No 3-4, 1997, pp. 581-588;

[6] Catalano, L.A., Dadone, A., "Ottimizzazione progressiva per un efficiente progetto automatico di profili palari", Atti del 56 Congresso Nazionale ATI, Napoli, 10-14 settembre 2001;

[7] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., Optimization by simulated annealing, Science 220, 1983, pp 671-680;

[8] Edwards, S. P. , Pilley, A.D. , Michon, S. , Fournier, G., "The Optimization of Common Rail FIE Equipped Engines Through the Use of

Statistical Experimental Design, Mathematical Modeling and Genetic Algorithms", SAE Paper 970346, 1997;

[9] Montgomery, D.C., Introduction to Statistical Quality Control, Wiley & Sons, 2001;

[10] Carlucci, P. , Ficarella, A., Laforgia, D., "Study of the Influence of the Injector Parameters on Combustion Noise in a Common Rail Diesel Engine Using ANOVA and Neural Networks" SAE Paper 2001-01-2011, 2001;

[11] Desantes, J.M., Lòpez, J.J., Garcìa, J.M., Hernàndez, L., "Application of Neural Networks for Prediction and Optimization of Exhausted Emissions in a H.D. Diesel Engine", Sae Paper 2002-01-1144;

[12] Hafner, M., Isermann, R., "The Use of Stationary and Dynamic Emission Models for an Improved Engine Performance in Legal Test Cycles", Modeling , Emissions and Control in Automotive Engines, Proceedings of the MeCA International Workshop September, 9-10, 2001;

[13] Box, G.E.P., Draper, N.R., Empirical Model-Building and Response Surfaces, Wiley, 1986;

[14] Box, G.E.P., Liu, P.Y.T., "Statistics as a Catalyst to Learning by Scientific Method, Part I- An example", J. of Quality Technology, Vol. 31, 1999;

[15] Thiel, M.P., Klingbeil, A.E., Reitz, R.D., "Experimental Optimization of a Heavy-Duty Diesel Engine Using Automated Genetic Algorithms", SAE paper 2002-01-0960, 2002;

[16] Carroll, D.L., "Genetic Algorithms and Optimizing Chemical Oxygen-Iodine Lasers", Developments in Theoretical and Applied Mechanics, 18, 411, 1996;

[17] Senecal, P.K., Reitz, R.D., "Simultaneous Reduction of Engine Emissions and Fuel Consumption using Genetic Algorithms and Multidimensional Spray and Combustion Modeling", SAE Paper 2000-01-1890, 2000;

[18] Senecal, P.K., Pomraning, E., Richards, K., "Multi-Mode Genetic Algorithm Optimization of Combustion Chamber Geometry for Low Emissions", SAE Paper 2002-01-0958.

[19] Murata, T., Ishibuchi, H., Tanaka, H., "Multi-objective Genetic Algorithm and its Application to Flowshop Scheduling", Computers ind. Eng., Vol 30, No 4, 1996, pp. 957-968;

[20] Fonseca, C.M., Fleming, J., "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", Proceedings of the Fifth International Conference on Genetic Algorithms, 1993, pp. 416-423.

[21] Santos, A., Dourado, A., "Global Optimization of Energy and Production in Process Industries: a Genetic Algorithm Application", Control engineering practice, 7, 1999, pp. 549-554;

[22] Oyama, A., Liou, M., Multiobjective Optimization of Rocket Engine Pumps using Evolutionary Algorithm, Journal of Propulsion and Power, Vol 18, No. 3, 2002, pp. 528-535;

[23] Zitler, E., Deb, K., Thiele, L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", Proceedings of the 1999 Genetic and Evolutionary Computation Conference, 2000;

[24] Coello, C., "Multiobjective Optimization using a Micro-Genetic Algorithm", Proceedings of the Genetic and Evolutionary Computation Conference GECCO, 2001;

[25] De Risi, A., Donateo, T., Laforgia, D., "Optimization of the Combustion Chamber of Direct Injection Diesel Engines", SAE paper 2003-01-1064;

[26] De Risi, A., Donateo T., Laforgia, D. "An Innovative Methodology to Improve the Design and the Performances of Direct Injection Diesel Engines", submitted to Internationa Journal of Engine Research, 2003;

[27] Donateo, T., de Risi, A., Laforgia, D., "Optimization of High Pressure Common Rail Electro-injector using Genetic Algorithms", SAE paper 2001-01-1980, 2001;

[28] Donateo, T., de Risi, A., Laforgia, D., "An Application of Multi-Criteria Genetic Algorithms to the Optimization of a Common-Rail Injector", ASME-ICE, Vol. 38, pp. 251-258, 2002;