

Introduzione alle tecniche di ottimizzazione

Marco Cavazzuti

MilleChili Lab, DIMeC, Università degli Studi di Modena e Reggio Emilia
Aprile 2013

definizione di ottimizzazione

Che cos'è l'ottimizzazione? Da una semplice definizione trovata su di un dizionario si legge che

in un problema di ottimizzazione si cercano i valori delle variabili che portano ad un valore ottimale della funzione da ottimizzare.

Da questa definizione appaiono già alcuni degli elementi fondamentali di un processo di ottimizzazione: innanzitutto occorre definire chiaramente il *problema di ottimizzazione*, ovvero quello che si vuole ottimizzare. Occorrono poi delle *variabili*, ovvero una opportuna parametrizzazione del problema, ed una *funzione da ottimizzare*, vale a dire un obiettivo. Servirà infine un procedimento attraverso il quale per un dato valore delle variabili (*configurazione* o *campione*) si calcola il relativo obiettivo, tale procedimento può essere un *esperimento* o una *simulazione*. In un processo di ottimizzazione solitamente si procede iterativamente generando una configurazione e valutandone il valore della funzione obiettivo. Un *algoritmo di ottimizzazione* fondamentalmente è un generatore di configurazioni da testare che con diverse logiche genera nuove configurazioni sulla base delle configurazioni testate in precedenza, ovvero delle informazioni raccolte.

terminologia

La terminologia nel campo dell'ottimizzazione non è del tutto uniformata, comunque nel seguito si adotta la seguente terminologia.

- problema di ottimizzazione: definisce l'oggetto dell'ottimizzazione
- parametri di input: set di parametri che caratterizzano il problema
- variabili di input (x_i): sottoinsieme dei parametri di input che fattivamente vengono presi in considerazione in un processo di ottimizzazione (l'insieme delle variabili di input può essere un sottoinsieme dei parametri di input nel caso in cui, per esempio, alcuni parametri non siano controllabili o siano considerati relativamente influenti e quindi tralasciati al fine di semplificare il processo di ottimizzazione). Ad ogni variabile di input solitamente è associato un range di variabilità.
- campione o configurazione (\mathbf{x}): un set di variabili di ingresso
- spazio di progetto (X): spazio di tutte le possibili configurazioni del problema di ottimizzazione

- esperimento o simulazione: il processo attraverso il quale da un campione si estraggono le informazioni ad esso relative
- parametri di output: l'insieme dei risultati raccolti tramite una simulazione o un esperimento.
- funzione obiettivo ($f(\mathbf{x})$): il parametro di uscita o una funzione dei parametri di uscita che si vuole massimizzare o minimizzare. Nel caso ci sia una sola funzione obiettivo si parla di ottimizzazione a singolo obiettivo, altrimenti di ottimizzazione multi-obiettivo.
- vincoli ($\mathbf{c}(\mathbf{x}) \leq \mathbf{0}$): funzioni dei parametri di uscita che devono essere rispettate dalla soluzione ottimale
- spazio delle soluzioni (Y): il codominio dello spazio di progetto.

La formulazione tipica di un problema di ottimizzazione dal punto di vista matematico è la seguente

$$\begin{array}{ll} \underset{\mathbf{x} \in X \subseteq \mathbb{R}^k}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{c}(\mathbf{x}) \leq \mathbf{0} \end{array} \quad (1)$$

tecniche di ottimizzazione

Un problema di ottimizzazione può essere affrontato in molti modi e servendosi di diverse tecniche. Possiamo distinguere cinque tipologie di tecniche:

- DOE - design of experiments
- RSM - response surface modelling, o response surface methodology
- ottimizzazione deterministica
- ottimizzazione stocastica
- RDA - robust design analysis, anche detto RED - robust engineering design

Nel seguito si discuterà brevemente lo scopo di queste tipologie di tecniche e se ne vedranno alcuni esempi.

DOE

Il DOE nasce nell'ambito della statistica intorno al 1920. Inizialmente era chiamato statistical experimental design. Il suo scopo è quello di pianificare in modo opportuno la sequenza di esperimenti da effettuare nel corso di una serie di analisi sperimentali effettuate in laboratorio in modo che al termine della sperimentazione si potessero eseguire delle analisi statistiche sui risultati al fine di determinare l'influenza che su di essi hanno le variabili del problema. Il DOE è poi stato utilizzato nell'ambito dell'ottimizzazione al fine di reperire informazioni sullo spazio di progetto tramite opportuni campionamenti. Una definizione classica afferma che le tecniche di DOE hanno come scopo quello di pianificare gli esperimenti al fine di

raccogliere la maggior quantità di informazioni sul problema in esame utilizzando la minor quantità di risorse.

terminologia nel DOE

Nel DOE si è soliti trovare una terminologia leggermente differente da quella utilizzata nell'ambito dell'ottimizzazione, in particolare:

- le variabili di input vengono chiamate fattori
- lo spazio di progetto viene chiamato regione di interesse
- la funzione obiettivo viene chiamata variabile di risposta
- l'insieme degli esperimenti da effettuare prende il nome di spazio dei campioni
- le tecniche di DOE si possono differenziare in base al numero di livelli. Il numero di livelli è dato dal numero di valori discreti che ogni variabile assume nello spazio dei campioni

tecniche per il DOE

A titolo informativo si elencano alcune delle possibili tecniche per il DOE, di seguito se ne prenderanno in considerazione alcune:

- randomized complete block design,
- latin square,
- full factorial,
- fractional factorial,
- central composite,
- Box-Behnken,
- Plackett-Burman,
- Taguchi,
- random,
- Halton,
- Faure,
- Sobol,
- latin hypercube,
- optimal design.

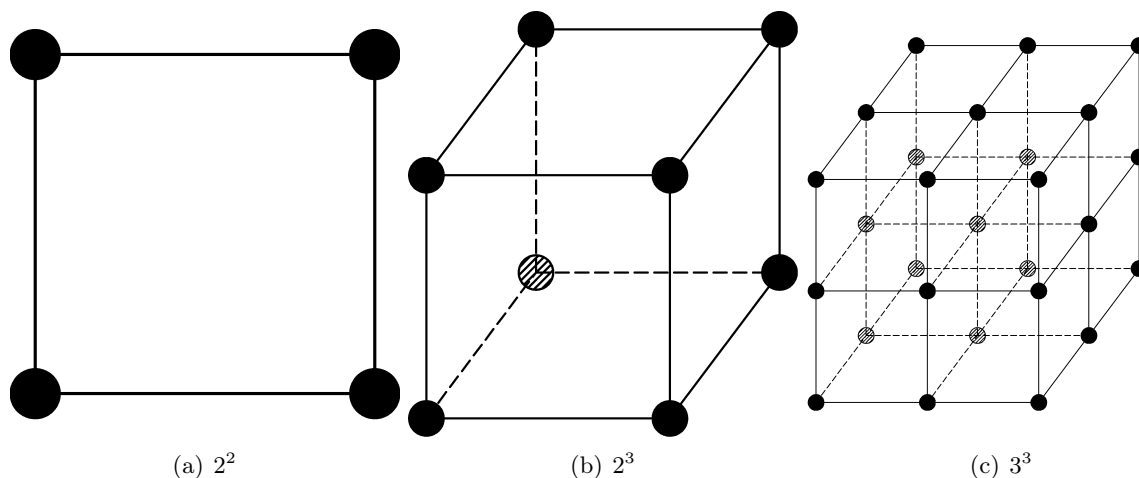


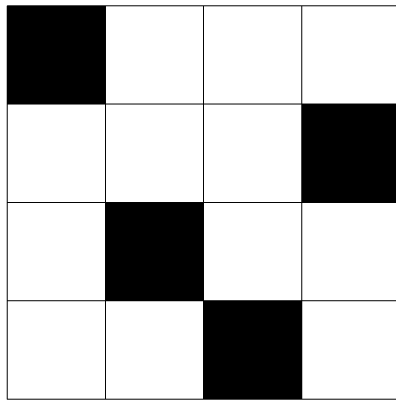
Figure 1: Esempi di campionamenti DOE full factorial.

full factorial

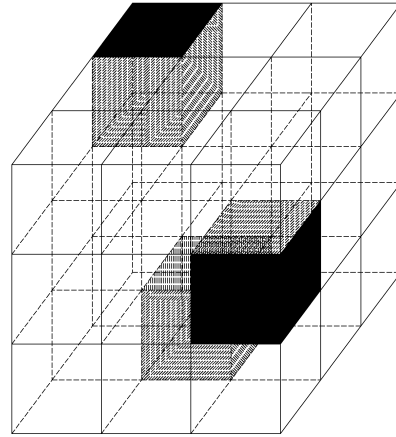
Consideriamo un problema a due variabili in cui ogni variabile possa variare tra -1 e $+1$. Lo spazio di progetto è quindi dato da un quadrato di lato due nel piano, centrato nell'origine e con lati paralleli agli assi coordinati. Secondo il full factorial a due livelli (individuato dalla dicitura 2^2) il numero di campionamenti da effettuare è pari a 4 ed è dato da tutte le permutazioni possibili in cui le due variabili assumono o il valore massimo o il valore minimo del loro range di variabilità. Analogamente questo concetto si può estendere a problemi a più dimensioni e a spazi dei campioni a più di due livelli. L'idea è molto intuitiva e consiste nel tracciare una griglia più o meno fitta nello spazio di progetto e campionare nei punti in cui le linee della griglia si intersecano. In Fig. 1 vengono visualizzati alcuni esempi. Nel caso il numero di livelli sia diverso per ogni variabile si parla di adjustable full factorial. Il vantaggio principale del full factorial è che dal punto di vista statistico permette una chiara stima dell'influenza dei fattori principali e delle interazioni tra fattori. Per contro, il numero di campioni necessari per completare un full factorial cresce molto rapidamente col numero di variabili e col numero di livelli e molto presto diventa impossibile a livello sperimentale e numerico effettuare un tal numero di esperimenti o simulazioni. D'altra parte, se il numero di livelli non è sufficientemente elevato, ci sono ampie zone dello spazio di progetto che sono sottocampionate, ovvero lontane da tutti i punti dello spazio dei campioni. Questo significa che qualora si vogliano interpolare i dati raccolti tramite DOE in queste zone l'affidabilità dell'interpolazione sarà scarsa.

latin hypercube

Con il LH il numero di campioni è scelto liberamente dall'utente e non, in un certo senso, imposto dalla tipologia di DOE scelta. Lo scopo del LH è quello di spargere il più possibile i campioni nello spazio di progetto allo scopo di limitare le zone sottocampionate ed aumentare l'affidabilità dell'interpolazione dei dati. Consideriamo uno spazio di progetto simile a quello descritto per il full factorial. Nel caso si decida di effettuare n campionamenti si traccia una griglia $n \times n$ nello spazio di progetto e si scelgono n quadrati tra gli $n \times n$ della griglia in modo che per ogni riga e per ogni colonna della griglia un solo quadrato sia scelto. Gli n campioni verranno quindi scelti casualmente all'interno degli n quadrati. La procedura è estendibile facilmente

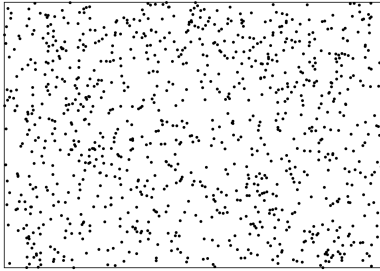


(a) $k = 2, N = 4$

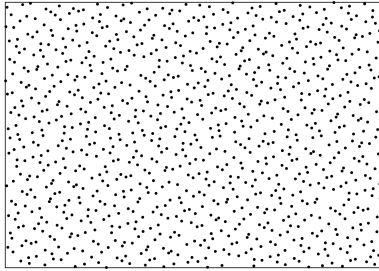


(b) $k = 3, N = 3$

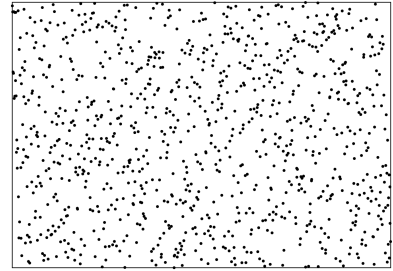
Figure 2: Esempi di campionamenti DOE LH.



(a) Random



(b) Sobol



(c) Latin hypercube

Figure 3: Confronto tra diverse tecniche di campionamento.

ad un caso multidimensionale. Occorre una scelta accorta degli n quadrati al fine di evitare effetti di correlazione spuri tra le variabili, esistono comunque tecniche per la riduzione della correlazione che modificano in modo opportuno un set di quadrati scelto casualmente. In Fig. 2 si esemplifica graficamente il campionamento LH.

Sobol

Il campionamento Sobol si basa su di un generatore di numeri pseudo-random che implementa la sequenza di Van der Corput. Non entriamo nel merito di come questa sequenza viene applicata allo scopo di generare lo spazio dei campioni. Basti sapere che la tecnica è in grado di spargere i campioni nello spazio di progetto in modo molto efficiente ed ordinato evitando eventuali accorpamenti (clustering) che si possono verificare con tecniche di campionamento random. La Fig. 3 mette a confronto tre campionamenti di mille campioni ciascuno su di uno spazio bidimensionale effettuati con campionamenti random, LH e Sobol.

RSM

Le tecniche di RSM hanno come scopo quello di interpolare o approssimare i dati raccolti tramite DOE creando un metamodello che poi può essere utilizzato nell'ottimizzazione. L'utilità nell'uso

combinato del DOE+RSM nell'ottimizzazione è che un processo di ottimizzazione può richiedere molte simulazioni e se le simulazioni sono particolarmente onerose questo significa dover aspettare molto tempo prima di avere i risultati. Campionando ed interpolando opportunamente i dati, con un numero inferiore di simulazioni si può ottenere un modello matematico che stima la risposta delle simulazioni. A questo punto effettuare delle ottimizzazioni basandosi sul metamodello è questione solitamente di pochi attimi. Questo approccio quindi può essere particolarmente vantaggioso nel caso di problemi di ottimizzazione molto onerosi. D'altra parte però appoggiarsi ad un metamodello è sempre un rischio: se il modello non è affidabile perché il numero di campioni è insufficiente o perché la risposta reale del sistema è particolarmente affetta da rumore i risultati che se ne ottengono possono essere pressoché inutili.

Con l'RSM il parametro di output $f(\mathbf{x})$ viene approssimato tramite una funzione $\hat{f}(\mathbf{x})$

$$f(\mathbf{x}) = \hat{f}(\mathbf{x}) + \varepsilon(\mathbf{x}) \quad (2)$$

Se l'errore $\varepsilon(\mathbf{x})$ nei punti campionati \mathbf{x}_i è nullo ($\varepsilon(\mathbf{x}_i) = 0$) il metodo viene detto interpolante, altrimenti viene detto approssimante.

tecniche per il RSM

A titolo informativo si elencano alcune delle tecniche più comuni per il RSM, di seguito se ne prenderanno in considerazione alcune:

- minimi quadrati,
- optimal RSM,
- Shepard,
- K-nearest,
- Kriging,
- processi Gaussiani,
- funzioni di base radiali,
- reti neurali.

Minimi quadrati

La tecnica più nota e più immediata per il RSM è quella dei minimi quadrati in cui si cerca di trovare i coefficienti β che definiscono un certo tipo di funzione data in modo che gli errori quadratici di interpolazione nei punti di campionamento sia minimo.

$$\text{minimize } \sum_{i=1}^N \varepsilon(\mathbf{x}_i)^2 = \sum_{i=1}^N \left(f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i, \beta) \right)^2 \quad (3)$$

Nel caso più semplice, ipotizzando una funzione di risposta lineare l'equazione che ne deriva è risolvibile analiticamente in forma chiusa, altrimenti, per funzioni di risposta non lineari è necessaria una procedura iterativa.

Nel caso di funzione di risposta lineare la procedura è abbastanza semplice: si vogliono trovare i coefficienti $\boldsymbol{\beta}$ della funzione

$$\hat{f}(\mathbf{x}, \boldsymbol{\beta}) = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k = \boldsymbol{\beta}^T \mathbf{x} \quad (4)$$

che minimizzano lo scarto quadratico S . Sia \mathbf{X} la matrice dei vettori \mathbf{x} per i punti del DOE, si ha

$$\begin{aligned} S = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} &= (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \quad \Rightarrow \\ \frac{\partial S}{\partial \boldsymbol{\beta}} &= -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X} \boldsymbol{\beta} = \mathbf{0} \quad \Rightarrow \\ \boldsymbol{\beta} &= (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad \Rightarrow \quad \hat{f}(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{x} \end{aligned} \quad (5)$$

K-nearest

Molti metodi per il RSM si basano sullo stimare il valore di \hat{f} tramite una media pesata dei valori delle funzioni obiettivo nei punti campionati. I diversi metodi si differenziano nella logica con cui vengono scelti i pesi $\boldsymbol{\lambda}$.

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^N \lambda_i(\mathbf{x}) f(\mathbf{x}_i), \quad \sum_{i=1}^N \lambda_i = 1 \quad (6)$$

Secondo il K-nearest i pesi vengono scelti sulla base delle distanze d_i del punto \mathbf{x} dai punti campionati \mathbf{x}_i . Più un punto è vicino ad \mathbf{x} più il suo peso sarà maggiore.

$$\lambda_i = \frac{\frac{1}{d_i^p + c}}{\sum_{j=1}^N \frac{1}{d_j^p + c}} \quad (7)$$

Dove c è una costante piccola che serve a fare in modo che la formula non degeneri quando \mathbf{x} coincide con uno degli \mathbf{x}_i . Il problema relativo all'uso del K-nearest nell'ottimizzazione è che il metodo con cui vengono scelti i pesi fa sì che questi siano sempre compresi tra 0 e 1. In questo modo il valore stimato per un generico punto \mathbf{x} utilizzando il metodo K-nearest sarà sempre compreso tra $\min f(\mathbf{x}_i)$ e $\max f(\mathbf{x}_i) \forall i$: vale a dire, non ha senso applicare tecniche di ottimizzazione su di un metamodello costruito con il K-nearest.

Kriging

Servono quindi dei metodi per il calcolo dei pesi che permettano agli stessi di diventare anche negativi o superiori all'unità. Un metodo più evoluto, anche se concettualmente simile è adottato dal metodo Kriging nel quale si fa uso di mezzi statistici, quali i concetti di media, varianza e covarianza nel calcolo dei pesi.

$$\begin{aligned} \mu &= E[f(\mathbf{x})] = \sum_{i=1}^N \frac{f(\mathbf{x}_i)}{N} \\ \text{var}(f(\mathbf{x})) &= E[(f(\mathbf{x}) - \mu)^2] \\ \text{cov}(f(\mathbf{x}), f(\mathbf{y})) &= E[(f(\mathbf{x}) - \mu)(f(\mathbf{y}) - \nu)] \end{aligned} \quad (8)$$

Il metodo Kriging nasce nell'ambito della geostatistica allo scopo di interpolare i rilevamenti altimetrici del terreno o la presenza e la disposizione di certi materiali in un'area geologica. Fondamentalmente i pesi vengono calcolati dalla risoluzione di un sistema lineare del tipo

$$\mathbf{C}(\mathbf{x}_i, \mathbf{x}_j) \boldsymbol{\lambda}(\mathbf{x}) = \mathbf{c}(\mathbf{x}_i, \mathbf{x}) \quad (9)$$

Dove $\mathbf{C}(\mathbf{x}_i, \mathbf{x}_j)$ è la matrice delle covarianze dei punti campionati e $\mathbf{c}(\mathbf{x}_i, \mathbf{x})$ è il vettore delle covarianze tra il punto che si vuole interpolare \mathbf{x} ed i punti campionati. \mathbf{C} e \mathbf{c} vengono approssimate mediante l'uso di opportuni variogrammi che approssimano l'andamento della covarianza dei dati campionati al variare della distanza relativa tra di essi.

ottimizzazione deterministica

Conosciuta anche come mathematical programming, comprende tutti i metodi di ottimizzazione matematicamente più rigorosi e che non presentano elementi stocastici al loro interno. Si tratta prevalentemente di metodi basati sul calcolo del gradiente e dell'Hessiano delle funzioni dei parametri di output e che in buona parte traggono origine dal metodo di Newton-Raphson. Qualora i gradienti e l'Hessiano di una data funzione non siano disponibili direttamente dalla simulazione, come spesso accade, questi vanno approssimati alle differenze finite, il che può richiedere un elevato numero di simulazioni. Generalmente si tende a distinguere, all'interno dell'ottimizzazione deterministica, tra ottimizzazione non vincolata ed ottimizzazione vincolata viste le complicazioni che l'inclusione di vincoli apporta ai tradizionali algoritmi di ottimizzazione.

Definita una direzione \mathbf{s} , definiamo il concetto di linea come

$$\mathbf{x}(\alpha) = \mathbf{x}' + \alpha \mathbf{s}, \quad \forall \alpha \in \mathbb{R} \quad (10)$$

Per la regola della derivazione a catena dalla definizione di linea si possono calcolare il gradiente e la curvatura lungo la linea come

$$\frac{df(\mathbf{x})}{d\alpha} = \mathbf{g}(\mathbf{x})^T \mathbf{s}, \quad \frac{d^2f(\mathbf{x})}{d\alpha^2} = \mathbf{s}^T \mathbf{G}(\mathbf{x}) \mathbf{s} \quad (11)$$

Le condizioni necessarie di primo e di secondo ordine perché un punto \mathbf{x}^* sia un punto di ottimo sono le seguenti

$$\begin{cases} \mathbf{s}^T \mathbf{g}(\mathbf{x}^*) = 0 & \forall \mathbf{s} \\ \mathbf{s}^T \mathbf{G}(\mathbf{x}^*) \mathbf{s} \geq 0 & \forall \mathbf{s} \end{cases} \Rightarrow \begin{cases} \mathbf{g}(\mathbf{x}^*) = \mathbf{0} \\ \mathbf{G}(\mathbf{x}^*) \text{ definita positiva} \end{cases} \quad (12)$$

I metodi di ottimizzazione a gradiente possono servirsi di due diversi approcci:

- line search: una volta che viene individuata la direzione in cui spostarsi viene effettuato un campionamento lungo la direzione scelta fino a quando certe condizioni vengono rispettate dal punto campionato. Queste si chiamano condizioni di Wolfe-Powell e servono ad evitare che il miglioramento della funzione obiettivo tra un passo e l'altro dell'ottimizzazione sia troppo piccolo e, di conseguenza, che la convergenza del metodo sia complessivamente troppo lenta.
- trust region: in questi metodi si procede iterativamente ad una locale approssimazione quadratica della funzione obiettivo tramite troncamento della serie di Taylor. Si va quindi a calcolare il punto di minimo di questa funzione all'interno di un intorno del punto. La dimensione dell'intorno è controllata in modo adattivo. Questa tecnica può essere vista come una procedura di rilassamento che cerca di evitare passi troppo lunghi tra un'iterazione e l'altra del processo di ottimizzazione al fine di favorirne la convergenza.

tecniche per l'ottimizzazione deterministica non vincolata

A titolo informativo si elencano alcune delle possibili tecniche per l'ottimizzazione deterministica non vincolata, di seguito se ne prenderanno in considerazione alcune:

- metodo di Newton,
- metodi quasi-Newtoniani,
- metodi delle direzioni coniugate,
- metodi di Levenberg-Marquardt,
- metodo del simplesso di Nelder e Mead.

metodo di Newton

Secondo il metodo di Newton si approssima la funzione nell'intorno del punto $\mathbf{x}^{(k)}$ raggiunto alla iterazione k mediante troncamento della serie di Taylor al secondo ordine

$$f(\mathbf{x}^{(k)} + \boldsymbol{\delta}) \approx q^{(k)}(\boldsymbol{\delta}) = f(\mathbf{x}^{(k)}) + \mathbf{g}(\mathbf{x}^{(k)})^T \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^T \mathbf{G}(\mathbf{x}^{(k)}) \boldsymbol{\delta} \quad (13)$$

Derivando si ottiene

$$\mathbf{G}(\mathbf{x}^{(k)}) \boldsymbol{\delta} = -\mathbf{g}(\mathbf{x}^{(k)}) \quad (14)$$

Risolvendo in $\boldsymbol{\delta}$ si ottiene il passo all'iterazione k ed il punto all'iterazione successiva è dato da

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \boldsymbol{\delta}^{(k)} \quad (15)$$

Il metodo di Newton però può divergere nel caso in cui la matrice Hessiana \mathbf{G} non sia definita positiva. Per ovviare a questo inconveniente si sono sviluppati diversi altri metodi e stratagemmi. Il più semplice consiste nel far diventare la matrice definita positiva aggiungendo ad essa un multiplo della matrice identità.

$$(\mathbf{G}(\mathbf{x}^{(k)}) + \nu \mathbf{I}) \mathbf{s}^{(k)} = -\mathbf{g}(\mathbf{x}^{(k)}) \quad (16)$$

Per poi andare ad effettuare un line search lungo la direzione \mathbf{s} . In questo modo la direzione del passo sta tra la direzione del passo di Newton (per $\nu = 0$) e quella del minimo gradiente (per $\nu = \infty$).

metodi quasi-Newtoniani

Il principale inconveniente nell'utilizzo del metodo di Newton e delle sue varianti sta nella necessità di conoscere il gradiente e la matrice Hessiana ad ogni passo dell'ottimizzazione. Qualora la valutazione di queste grandezze debba essere fatta alle differenze finite il numero di simulazioni richiesto dall'ottimizzazione cresce molto in fretta col numero di variabili del problema. I metodi quasi-Newtoniani sono in grado di limitare l'onere computazionale e dare maggiore stabilità al metodo di Newton tradizionale. In essi il gradiente viene calcolato, mentre invece la matrice Hessiana (o meglio la sua inversa) viene approssimata iterativamente tramite delle opportune formule che tengono in considerazione il gradiente ed il passo fatto dall'ottimizzatore nelle precedenti iterazioni. Tramite queste formule di aggiornamento iterativo dell'inversa della

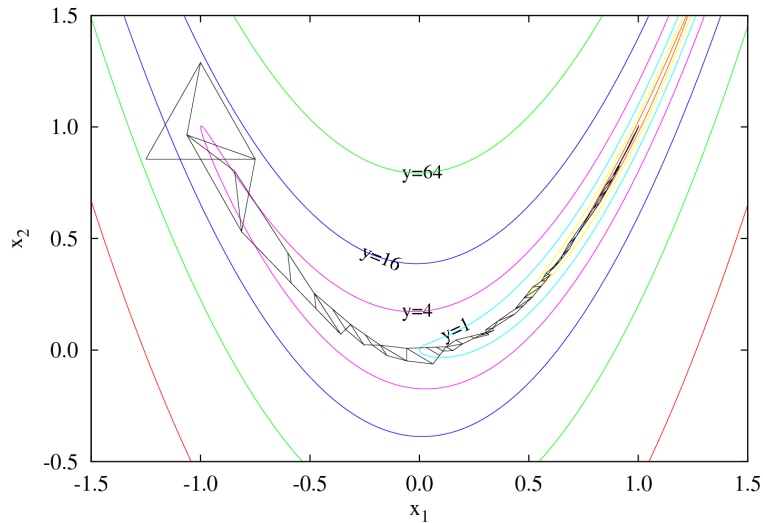


Figure 4: Esempio di ottimizzazione tramite metodo del semplice di Nelder e Mead.

matrice Hessiana \mathbf{H} si ha che partendo da $\mathbf{H}^{(1)} = \mathbf{I}$ alla prima iterazione, detta matrice rimane sempre definita positiva durante l'ottimizzazione (di fatto evitando che la soluzione possa divergere), e al tendere dell'ottimizzazione verso un punto di ottimo \mathbf{x}^* , $\mathbf{H}^{(k)}$ tende a $\mathbf{G}(\mathbf{x}^*)^{-1}$. Nonostante l'utilizzo di Hessiani approssimati questi metodi riescono a convergere molto velocemente alla soluzione.

I più noti metodi di ottimizzazione quasi-Newtoniani sono il DFP ed il BFGS. Il primo è generalmente un po' più lento a convergere e non garantisce che la stima della matrice Hessiana inversa rimanga sempre definita positiva.

metodo del semplice di Nelder e Mead

Il metodo del semplice di Nelder e Mead per l'ottimizzazione non lineare è un procedimento molto intuitivo ed efficace per affrontare problemi di ottimizzazione. In un problema ad n dimensioni si considerano $(n + 1)$ campioni iniziali. Questi campioni rappresentano i vertici del semplice, il semplice è infatti definito come una figura a $n + 1$ vertici in uno spazio n dimensionale. I vertici del semplice vengono valutati e quello la cui funzione obiettivo è peggiore viene riflesso rispetto al centroide dei rimanenti campioni. Ripetendo questo procedimento e permettendo al semplice di deformarsi sotto certe condizioni il semplice viene a spostarsi e a restringersi verso il punto di ottimo della funzione. I procedimenti attraverso i quali il semplice si muove nello spazio sono quelli della riflessione, contrazione, espansione e restringimento. In Fig. 4 si mostra con un esempio il moto del semplice nel piano nel caso di un problema di ottimizzazione risolto tramite metodo del semplice. La funzione test che si vuole minimizzare nell'esempio è la funzione di Rosenbrock.

ottimizzazione vincolata

Il tema dell'ottimizzazione vincolata è matematicamente molto più complesso. Le condizioni di minimalità coinvolgono i moltiplicatori di Lagrange e ci si deve avvalere di diverse tecniche per

tener conto dei vincoli. Un generico problema di ottimizzazione vincolata si scrive nella forma

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f(\mathbf{x}) \\ & \text{subject to} && c_i(\mathbf{x}) = 0 \quad i \in E \\ & && c_i(\mathbf{x}) \leq 0 \quad i \in I \end{aligned} \tag{17}$$

Dove E e I sono gli insiemi dei vincoli di uguaglianza e di disuguaglianza rispettivamente. Tali problemi di ottimizzazione si possono risolvere tramite sostituzione diretta, tramite metodo dei moltiplicatori di Lagrange, o tramite metodi basati sulle penalty functions. Qualsiasi metodo venga applicato occorre integrarlo con un metodo active set che ha lo scopo di individuare ad ogni passo quali sono i vincoli di disuguaglianza attivi.

I metodi principali per la risoluzione di problemi di ottimizzazione deterministica vincolati sono i metodi SQP (sequential quadratic programming) in cui la funzione obiettivo viene iterativamente approssimata ad una funzione quadratica e risolta tramite metodo dei moltiplicatori di Lagrange. Il metodo più avanzato è un metodo SQP chiamato NLPQLP, il cui acronimo sta per non linear programming (NLP) con approssimazioni quadratiche successive della funzione obiettivo (Q), linearizzazione delle funzioni di vincolo (L) e possibilità di risoluzione tramite calcolo parallelo (P).

ottimizzazione stocastica

L'ottimizzazione stocastica comprende tutte quelle tecniche di ottimizzazione in cui sono presenti a qualche livello dei generatori di numeri random o pseudo-random. La forza di questi algoritmi sta nel fatto che i campioni testati possono spostarsi nello spazio di progetto più liberamente rispetto a quanto succede per l'ottimizzazione deterministica. Per questo motivo non soffrono del fatto di poter incappare in minimi locali senza poterne uscire. I metodi deterministici, infatti, essendo basati sul calcolo del gradiente, sono metodi locali e non sono in grado di uscire dal bacino di attrazione del punto di minimo all'interno del quale il punto da cui viene fatta partire l'ottimizzazione si trova. I metodi stocastici, essendo in grado di spaziare piuttosto liberamente nello spazio di progetto, possono esplorarlo riducendo la probabilità di fermarsi a soluzioni solo localmente ottime. D'altra parte questa loro natura implica necessariamente un processo di ottimizzazione che solitamente ha una velocità di convergenza verso il punto di ottimo molto più lenta.

I metodi stocastici inoltre sono in grado di gestire problemi di ottimizzazione multi-obiettivo, cosa impossibile ad un metodo deterministico. Mentre il risultato di un processo di ottimizzazione a singolo obiettivo consta di una soluzione ottimale, nel caso di ottimizzazione multi-obiettivo non è così. Un processo di ottimizzazione multi-obiettivo infatti ha come risultato un set di soluzioni che vengono dette non dominate. Consideriamo un problema di ottimizzazione a due funzioni obiettivo f_1 e f_2 (vedi Fig. 5) ove si voglia minimizzare sia f_1 che f_2 . Se si considerano due campioni \mathbf{x}_1 e \mathbf{x}_2 tali per cui

$$f_1(\mathbf{x}_1) \leq f_1(\mathbf{x}_2) \quad \text{e} \quad f_2(\mathbf{x}_1) \leq f_2(\mathbf{x}_2) \tag{18}$$

è facile stabilire che il campione \mathbf{x}_1 è migliore del campione \mathbf{x}_2 in quanto entrambe le funzioni obiettivo relative al punto \mathbf{x}_1 sono migliori di quelle del punto \mathbf{x}_2 . In questo caso si dice che il campione \mathbf{x}_1 domina il campione \mathbf{x}_2 . Nel caso in cui, per esempio, sia invece

$$f_1(\mathbf{x}_1) \geq f_1(\mathbf{x}_2) \quad \text{e} \quad f_2(\mathbf{x}_1) \leq f_2(\mathbf{x}_2) \tag{19}$$

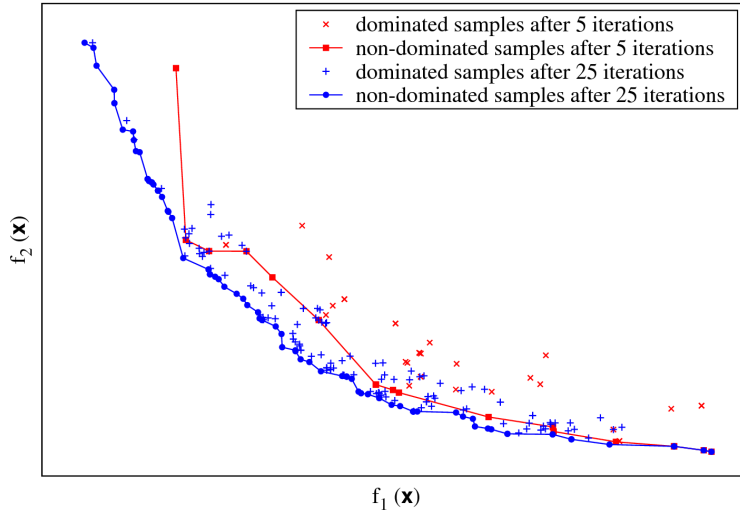


Figure 5: Rappresentazione del fronte di Pareto nello spazio delle soluzioni. La figura si riferisce ad un problema di ottimizzazione che mira alla minimizzazione della funzione f_1 ed alla minimizzazione della funzione f_2 .

non c'è modo di stabilire quale dei due campioni sia migliore in quanto uno ha una performance migliore relativamente ad un obiettivo, l'altro ha una performance migliore rispetto all'altro obiettivo. In questo caso si dice che i punti sono reciprocamente non dominati. Al termine di un processo di ottimizzazione tutti i campioni valutati che non sono dominati da nessun altro campione formano il fronte di Pareto. I punti che appartengono al fronte di Pareto sono quindi tutti punti ottimali. Alcuni propenderanno più verso una funzione obiettivo, altri verso un'altra, altri ancora staranno in mezzo, ma comunque sono tutti punti non dominati. A questo punto il progettista ha la possibilità di scegliere la configurazione preferita tra quelle di Pareto. Implementando quindi opportuni algoritmi in grado di individuare il fronte di Pareto in un set di campioni è quindi possibile effettuare ottimizzazioni multi-obiettivo.

tecniche per l'ottimizzazione stocastica

A titolo informativo si elencano alcune delle possibili tecniche per l'ottimizzazione stocastica:

- simulated annealing (SA),
- particle swarm optimization (PSO),
- game theory optimization (GT),
- evolutionary algorithms (EA),
- genetic algorithms (GA).

Tutte queste tecniche si basano su concetti diversi ma che comunque mirano a dare al processo di ottimizzazione la possibilità di spaziare inizialmente con una certa libertà all'interno dello spazio di progetto. Man mano che il processo di ottimizzazione continua questa libertà viene via via ad essere ristretta in modo da permettere all'algoritmo di concentrarsi sui campioni migliori

trovati fino a quel momento.

Le idee alla base di questi algoritmi sono a volte piuttosto stravaganti e fantasiose. Per esempio, il SA vuole simulare il comportamento degli atomi all'interno di un metallo che si raffredda. Nel corso dell'ottimizzazione un parametro chiamato temperatura cala. Man mano che la temperatura cala l'algoritmo riduce la possibilità che un nuovo campione venga "accettato" nel caso in cui presenti una diminuzione delle performance del sistema. Ovvero, quando la temperatura è alta l'elettrone ha più mobilità e può spostarsi su stadi a maggiore energia, quando il metallo si raffredda però questa possibilità viene meno. Quindi l'algoritmo inizialmente può spaziare nello spazio di progetto ed uscire da minimi locali avendo la possibilità di muoversi anche in direzioni sconvenienti in termini di performance. In una seconda fase questo non è più possibile e l'algoritmo si concentra sulla soluzione migliore trovata fino a quel momento allo scopo di migliorarla spostandosi localmente.

Il PSO vuole invece simulare il comportamento degli stormi di uccelli in cerca di cibo. Qui cibo sta per valore migliore della funzione obiettivo. Si parte da un set di campioni che rappresentano gli uccelli dello stormo. Ad ogni iterazione gli uccelli si muovono con una certa velocità ed in una certa direzione dettate da un effetto random (per esplorare lo spazio di progetto), dalla memoria della posizione migliore trovata fino a quel punto dall'individuo (per esplorare le zone di minimo locale) e dalla posizione dell'uccello che si trova nella posizione migliore rispetto ai compagni, vale a dire gli uccelli tendono a seguire il leader dello stormo (per esplorare le zone intorno al miglior minimo trovato fino a quel momento, che potrebbe essere il minimo globale).

Il GT vuole simulare l'evoluzione di un gioco in cui diversi giocatori nel proprio turno di gioco tentano di vincere la partita perseguendo il proprio scopo (cioè massimizzando o minimizzando una certa funzione obiettivo). Tale tecnica si basa sulla teoria dei giochi di Nash.

I metodi EA e GA vogliono invece simulare l'evoluzione di una popolazione seguendo la teoria evoluzionistica di Darwin. Si parte da una popolazione di individui (un set di campioni) che vengono valutati relativamente alla loro performance. Secondo un processo di selezione naturale gli individui più prestanti hanno una maggiore probabilità di generare figli (ovvero altri campioni da valutare). Secondo i metodi EA l'evoluzione avviene principalmente per mutazione genetica, ovvero per mutazione di alcune caratteristiche (variabili) degli individui scelti. I metodi GA invece si basano su di una codifica binaria delle variabili che li caratterizzano. La stringa binaria che ne deriva viene chiamata DNA dell'individuo. Secondo un meccanismo di selezione gli individui vengono scelti due alla volta allo scopo di generare due figli tramite un cross-over del proprio patrimonio genetico.

I metodi più comunemente utilizzati per l'ottimizzazione stocastica sono gli EA nel caso si voglia dare particolare rilevanza all'esplorazione dello spazio di progetto, e i GA.

RDA

Per quanto possa sembrare un controsenso non è sempre una buona norma posizionarsi in sede di progetto sulla configurazione globalmente ottima; è buona norma verificarne prima la robustezza. Infatti, possono esserci parametri che non possono essere controllati e che fanno sì che il reale funzionamento dell'oggetto del problema di ottimizzazione si scosti dalle condizioni di progetto, influenzando significativamente la sua performance. Per esempio, questi fattori di disturbo possono essere dovuti a

- piccoli errori in produzione o incertezze dovute alle tolleranze,
- deterioramento dell'oggetto con l'uso,

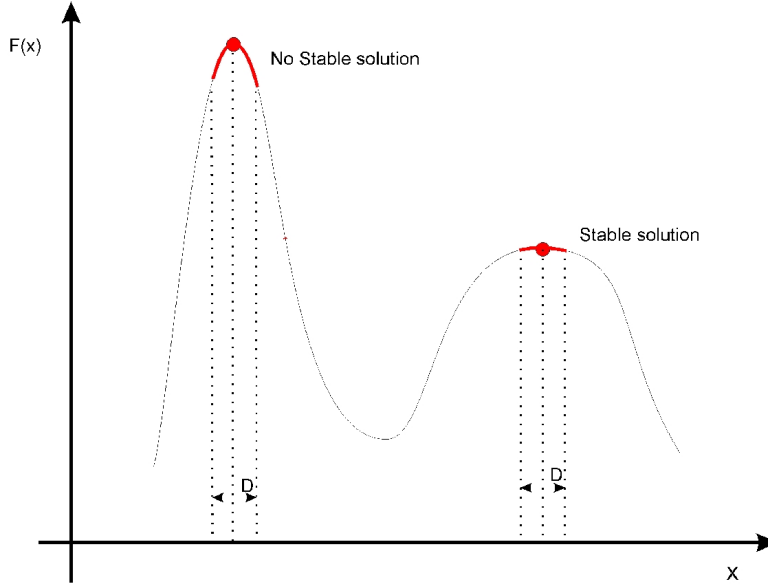


Figure 6: Esempio di robustezza della soluzione.

- condizioni operative variabili o comunque diverse da quelle di progetto,
- altri parametri non controllabili.

Se la performance della configurazione di ottimo risente in maniera forte delle incertezze può essere una scelta più conservativa posizionarsi su di una configurazione subottimale che però risenta meno delle incertezze. Una tale configurazione viene detta robusta. Per robustezza si intende appunto la capacità di non perdere troppo in termini di prestazioni a causa delle incertezze. Lo scopo del RDA è appunto quella di valutare la robustezza delle configurazioni. La Fig. 6 esemplifica la situazione descritta sopra.

Esistono due diversi approcci al RDA: il MORDO e il RA. Entrambi i metodi si basano su di un campionamento Montecarlo nell'intorno delle configurazioni da testare in termini di robustezza. Per far questo ovviamente occorre definire una distribuzione per ogni variabile affetta da rumore. Il punto debole di queste analisi è che l'entità dell'incertezza che ci si può aspettare per ogni variabile non è sempre nota, e quindi non è facile definire una corretta dispersione dei dati.

Il MORDO trasforma il problema di ottimizzazione a p obiettivi in un problema a $2p$ obiettivi in cui si mira a massimizzare o minimizzare la media μ delle funzioni obiettivo, e a minimizzarne la deviazione standard σ^2 . La determinazione della robustezza tramite MORDO richiede quindi l'uso di algoritmi di ottimizzazione multi-obiettivo (l'acronimo difatti sta per multi-objective robust design optimization). Nel caso di $p = 1$ il problema di ottimizzazione viene modificato come segue

$$\begin{array}{ll}
 \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) \\
 \text{subject to} & \mathbf{c}(\mathbf{x}) \leq \mathbf{0}
 \end{array}
 \quad \Longrightarrow \quad
 \begin{array}{ll}
 \underset{\mathbf{x}}{\text{minimize}} & \mu(\mathbf{x}) \\
 \underset{\mathbf{x}}{\text{minimize}} & \sigma^2(\mathbf{x}) \\
 \text{subject to} & \mathbf{c}(\mathbf{x}) \leq \mathbf{0}
 \end{array}
 \quad (20)$$

Il RA (reliability analysis) definisce invece per ogni obiettivo un target di performance al di

sotto del quale non si vuole scendere. Tramite campionamenti Montecarlo viene poi stimata la probabilità che le configurazioni testate non rispettino i target imposti. Qualora questa probabilità sia molto bassa (come è sperabile che sia) il campionamento Montecarlo è però particolarmente inefficiente. Difatti, il coefficiente di variazione è definito come

$$\nu = \frac{\sigma}{P_f} = \sqrt{\frac{1 - P_f}{nP_f}} \quad (21)$$

ed individua una stima dell'errore che si ha sulla valutazione dell'incertezza. P_f sta per la probabilità di fallimento e n per il numero di campioni. Se si vuole avere una stima il cui errore sia entro il $\pm 10\%$ ($\nu = 0.1$) e la probabilità di fallimento è $P_f = 3 \times 10^{-3}$, si vede infatti che il numero di campionamenti da effettuare è di almeno $n = 33234$. Esistono diverse tecniche per incrementare l'efficienza del campionamento le quali non vengono trattate in questa sede. Alcune di esse sono elencate di seguito:

- FORM,
- SORM,
- SILHS,
- TILHS,
- AILHS,
- AISMC.

il processo di ottimizzazione

Si è visto che esistono diverse tecniche di cui ci si può servire allo scopo di risolvere un problema di ottimizzazione. Ognuna di esse ha le proprie caratteristiche, i propri punti di forza e le proprie debolezze. È importante considerare che l'ottimizzazione non è mai un processo one-shot in cui si sceglie un dato algoritmo e basta. A seconda del problema è buona norma servirsi di più tecniche, anche se questo può essere dispendioso in termini di numero di simulazioni necessarie per completare il processo di ottimizzazione.

Per esempio, ci si può servire di un campionamento DOE allo scopo di creare una superficie di risposta tramite tecniche RSM. Sul metamodello che ne risulta si può applicare un metodo di ottimizzazione stocastico allo scopo di esplorare lo spazio di progetto, per poi raffinare la soluzione ottima trovata tramite un metodo di ottimizzazione deterministico. Infine si può valutare la robustezza della soluzione trovata tramite una analisi di affidabilità (RA). Un simile approccio definisce quello che si può chiamare un processo di ottimizzazione completo, in cui tutte le tecniche a disposizione vengono utilizzate all'interno dello stesso processo.

I singoli algoritmi che possono essere scelti ad ogni passo del processo poi possono essere più adatti per una tipologia di problema piuttosto che per un'altra. L'impostazione di un processo di ottimizzazione non è quindi sempre immediata e richiede conoscenza delle varie tecniche da parte del progettista ed esperienza.

L'ottimizzazione in ambito strutturale

In ambito strutturale esistono diverse tecniche di ottimizzazione ad hoc quali

- ottimizzazione topologica (topology),
- ottimizzazione topometrica (topometry),
- ottimizzazione topografica (topography),
- ottimizzazione di forma (shape),
- ottimizzazione di dimensione (size).

Quando si vuole ottimizzare la forma di un oggetto si ricorre generalmente ad una parametrizzazione della forma ed alla risoluzione del problema di ottimizzazione che ne deriva mediante le tecniche discusse in precedenza (ottimizzazione di shape). Una parametrizzazione della forma di un oggetto fatta in questo modo non consente tuttavia di definirne la topologia ottimale. Vale a dire, non consente di individuare la distribuzione ottimale del materiale in una struttura: dove è meglio mettere i buchi e dove invece mettere il materiale.

Per poter risolvere un problema di ottimizzazione topologica occorre impostare il problema in modo diverso. Si consideri un modello FEM in cui la seguente parametrizzazione viene applicata: ad ogni elemento del modello corrisponde una variabile del problema di ottimizzazione x_i . Questa variabile varia con continuità nel range $(0, 1]$ e rappresenta la densità relativa dell'elemento. Il problema viene risolto in un dominio continuo per ragioni di semplicità, in quanto una ottimizzazione discreta presenta maggiori difficoltà di risoluzione, anche se un elemento la cui densità relativa non sia o zero o uno non ha alcun significato fisico. La densità di materiale degli elementi viene quindi fatta variare e con qualche stratagemma si può far in modo di ottenere soluzioni con un numero contenuto di elementi a densità intermedia.

Il problema principale nella risoluzione di un problema di ottimizzazione di questo tipo è che il numero di variabili è estremamente elevato e quindi non è agevole affrontarlo. Per esempio, la costruzione di un metamodello è impensabile, come è improponibile anche l'utilizzo di tecniche di ottimizzazione stocastica o una analisi di robustezza. Le tecniche di ottimizzazione deterministica sono applicabili ma solo qualora i gradienti non vadano stimati alle differenze finite e siano direttamente disponibili dal calcolo FEM.

È appunto il verificarsi di questa condizione che rende l'ottimizzazione topologica possibile. L'algoritmo di ottimizzazione alla base del processo è solitamente il MMA (metodo degli asintoti mobili) che è un algoritmo di ottimizzazione deterministica a gradiente. Ad ogni elemento del dominio si associa una densità ed una rigidezza funzione del valore della variabile x_i .

$$\rho_i(x_i) = x_i \rho^*, \quad E_i(x_i) = x_i^p E^* \quad (22)$$

Dove ρ^* e E^* sono la densità e la rigidezza dell'elemento a densità relativa unitaria. Grazie a questa scelta si può calcolare agevolmente la derivata

$$\frac{\partial E_i(x_i)}{\partial x_i} = p x_i^{p-1} E^* \quad (23)$$

La scelta di un legame di questo tipo tra variabile x_i e rigidezza dell'elemento permette quindi il calcolo dei gradienti delle funzioni obiettivo per ogni variabile all'interno dell'analisi FEM scongiurando il ricorso alle differenze finite. Si consideri un problema di minimizzazione della

compliance c di una struttura (vale a dire dell'energia esterna della struttura). Sia E_i la rigidezza dell'elemento, $\mathbf{K}_i(x_i) = x_i^p \mathbf{K}_i^*$ la matrice di rigidezza dell'elemento, $\mathbf{K} = \sum_i \mathbf{K}_i(x_i)$ la matrice di rigidezza della struttura, \mathbf{f} il vettore delle forze applicate ed \mathbf{u} il vettore degli spostamenti. Il problema di ottimizzazione considerato può essere scritto come

$$\begin{aligned} & \underset{\mathbf{u}, \mathbf{x}_i}{\text{minimize}} && \mathbf{f}^T \mathbf{u} \\ & \text{subject to} && \mathbf{K} \mathbf{u} = \mathbf{f} \end{aligned} \quad (24)$$

dove il vincolo applicato sta ad indicare il soddisfacimento dell'equilibrio. Ulteriori vincoli come ad esempio la massa relativa della struttura vengono solitamente applicati accanto a questo per dare un senso all'ottimizzazione ed evitare che la soluzione degeneri. Applicando l'adjoint method per il calcolo del gradiente la compliance c può essere riscritta come

$$c(x_i) = \mathbf{f}^T \mathbf{u} = \mathbf{f}^T \mathbf{u} - \tilde{\mathbf{u}}^T (\mathbf{K} \mathbf{u} - \mathbf{f}) \quad (25)$$

Derivando si ottiene

$$\frac{\partial c(x_i)}{\partial x_i} = (\mathbf{f}^T - \tilde{\mathbf{u}}^T \mathbf{K}) \frac{\partial \mathbf{u}}{\partial x_i} - \tilde{\mathbf{u}}^T \frac{\partial \mathbf{K}}{\partial x_i} \mathbf{u} \quad (26)$$

Scegliendo $\tilde{\mathbf{u}} = \mathbf{u}$, con alcuni passaggi algebrici si ricava

$$\frac{\partial c(x_i)}{\partial x_i} = -\mathbf{u}^T \frac{\partial}{\partial x_i} \left(\sum_j x_j^p \mathbf{K}_j^* \right) \mathbf{u} = -p x_i^{p-1} \mathbf{u}^T \mathbf{K}_i^* \mathbf{u} \quad (27)$$

In questo modo si vede come, nel caso di minimizzazione della compliance, le derivate della funzione obiettivo rispetto a tutte le variabili del problema vengano calcolate direttamente all'interno di una singola simulazione FEM. Un ragionamento analogo vale per altri tipi di funzione obiettivo.

Due parametri governano il comportamento del solutore topologico: il parametro di penalità p ed il filtro di sensitività r . L'effetto di questi parametri è nullo se essi sono posti uguali ad uno, mentre diventa sempre maggiore mano a mano che il valore del parametro cresce. Il ruolo di p è quello di rendere gli elementi a densità intermedia svantaggiosi. Infatti, per esempio, per $p = 3$ un elemento a densità relativa $x_i = 0.5$ pesa relativamente molto ($0.5^3 V$) rispetto al contributo che dà alla rigidezza della struttura ($0.5^3 E^* = 0.125 E^*$). Applicando un valore di $p = 1$ infatti questo effetto viene meno ed il risultato dell'ottimizzazione è dato da una nuvola di grigi, ovvero di elementi a densità intermedia. Al crescere di p le soluzioni diventano sempre più "bianco e nero". Si instaura però un fenomeno noto come checkerboarding per il quale il solutore si sposta verso configurazioni "a scacchiera" in cui elementi a densità bassa ed elevata sono alternativamente uno accanto all'altro. Queste configurazioni danno infatti alla struttura una rigidezza spuria, che non ha significato fisico. Lo scopo del filtro di sensitività è appunto quello di evitare il checkerboarding. Il filtro agisce mediando, ad ogni iterazione, la densità di ogni elemento con quelle degli elementi vicini entro un raggio di azione di r volte la dimensione media dell'elemento. In questo modo il solutore forza la configurazione finale a creare strutture trabeiformi di diametro pari ad almeno $2r$ volte la dimensione media degli elementi. L'effetto dei due parametri sul risultato dell'ottimizzazione topologica è esemplificato in Fig. 7 in cui è rappresentato il risultato nel caso di una struttura bidimensionale tipo ponte caricata in mezzeria ed incernierata alle estremità.

Gli altri tipi di ottimizzazione strutturale citati all'inizio del paragrafo, eccezion fatta










$p \downarrow r \rightarrow$	1.0	1.2	1.5
1.0			
1.5			
2.5			

Figure 7: Effetto dei parametri p ed r sul risultato dell'ottimizzazione topologica.

per l'ottimizzazione di shape, sono concettualmente molto simili all'ottimizzazione topologica. L'ottimizzazione topometrica si differenzia per il fatto che le variabili del problema non sono più date dalle densità relative degli elementi ma dagli spessori degli elementi shell. Stessa cosa per l'ottimizzazione di size in cui però gli spessori non vengono variati elemento per elemento ma componente per componente. L'ottimizzazione topografica considera come variabili degli offset degli elementi shell dal piano medio della piastra, in questo modo, con opportuni accorgimenti, si vuole simulare l'effetto dell'aggiunta di nervature su di una piastra. Si noti come l'ottimizzazione topologica può essere applicata sia a mesh tridimensionali che di tipo shell, il concetto di densità infatti è valido per entrambi i tipi di elementi. Le ottimizzazioni topometriche, topografiche e di size sono invece applicabili esclusivamente ad elementi di tipo shell.

L'ottimizzazione di shape, come già accennato all'inizio del paragrafo, è invece una cosa differente. Tutto quel che serve è una parametrizzazione della forma dell'oggetto che può essere fatta, per esempio, mediante curve di Bézier o tramite l'utilizzo di procedure di morphing applicate alla struttura. Il numero di variabili è quindi relativamente basso e generalmente il problema può essere risolto mediante procedure di ottimizzazione classiche.

Per familiarizzare con l'ottimizzazione topologica esiste uno strumento didattico molto istruttivo sviluppato dall'università tecnica della Danimarca. Si tratta di uno script matlab che in sole 99 righe di codice contiene un solutore agli elementi finiti con integrato un ottimizzatore topologico. Lo script funziona solamente per strutture bidimensionali rettangolari con mesh quadrangolari regolari e per problemi di minimizzazione della compliance dato un peso della struttura come target. Lo script ed il relativo manuale sono reperibili su internet agli indirizzi

<http://www.topopt.dtu.dk/files/top.m>
<http://www.topopt.dtu.dk/files/matlab.pdf>