

6. CONCLUSIONI

L'obiettivo principale della tesi era l'analisi di un nuovo algoritmo di ottimizzazione matematica e l'applicazione di tale metodologia per la calibrazione di differenti modelli che rappresentano un giacimento: il problema di *History Matching*.

Durante l'esposizione della tesi, abbiamo verificato che i problemi di *History Matching* sono a priori mal posti, cioè la calibrazione dei vari modelli che rappresentano un giacimento porta ad identificare più minimi locali della funzione obiettivo. Data la natura e la complessità del problema inverso che abbiamo affrontato, non è a prima vista possibile concludere quale modello calibrato potrà fornire previsioni di produzione attendibili. Da un lato, la valutazione della funzione obiettivo durante la simulazione è un processo caratterizzato da un elevato costo computazionale, che ci obbliga a ridurre il numero di simulazioni per ogni calibrazione e quindi diminuisce le probabilità di localizzare minimi locali più promettenti; da un altro lato, gli algoritmi normalmente implementati nei software di simulazione commerciali sono di tipo locale e quindi non offrono la possibilità di analizzare più punti stazionari della funzione oggetto.

L'ultimo di questi due problemi è stato in parte risolto con l'introduzione della metodologia di ottimizzazione di *Tunneling*, che, come abbiamo visto, è in grado di calcolare più minimi locali caratterizzati da un valore decrescente della funzione obiettivo. Il fatto di avere a disposizione più modelli calibrati per una stessa parametrizzazione ci ha consentito di generare diverse previsioni di produzione per ogni modello. Prendendo in considerazione tutti i punti stazionari per cui la funzione oggetto ha un valore relativamente piccolo, abbiamo avuto a disposizione più set di parametri e quindi diversi scenari di produzione possibili. Ciò è estremamente importante, dato che è possibile interpretare e predire il comportamento futuro del giacimento da più angolazioni, nonché fornire un quadro più completo ed esaustivo sulle possibilità di produzione future del *Reservoir*.

Durante l'analisi dei casi test relativi al giacimento PunqS3, è stato possibile verificare che il valore della funzione obiettivo decresce aumentando il numero di parametri che descrivono il giacimento. Per parametrizzazioni ricche, infatti, questo

comportamento è spiegabile in base all'aumento dei gradi di libertà aggiustabili nel corso della minimizzazione.

Considerando sempre la parametrizzazione Punq30, abbiamo notato come le fasi di *Tunneling* fossero divenute dispendiose rispetto ai casi test rappresentati da un numero inferiore di parametri. Non è facile ricercare il motivo di tale comportamento del metodo: da un lato, come nel caso di molte funzioni analitiche tratte dalla letteratura matematica, è possibile che il numero di minimi locali isolati della funzione obiettivo aumenti proporzionalmente (e a volte esponenzialmente) all'aumentare della dimensionalità del problema, e quindi l'ottimizzazione della funzione ausiliaria di *Tunneling* risulti molto ardua. Da un altro lato, incrementando il numero di parametri lo spazio delle variabili assume dimensioni sempre maggiori ed è quindi sempre più complesso riuscire ad esplorarlo in modo esaustivo in modo da localizzare nuovi punti di partenza per una successiva fase di minimizzazione.

Dall'analisi del caso test Punq30, è possibile notare come i quattro minimi locali calcolati con il metodo di *Tunneling* forniscano delle previsioni di produzione sostanzialmente identiche. Da questo punto di vista, dunque, una parametrizzazione più ricca non permette di generare degli scenari di produzione differenti, nonostante lo sforzo computazionale applicato per la calibrazione del modello mediante l'algoritmo di *Tunneling*.

Generalizzando dunque questi commenti a tutti i casi test presi in considerazione, è possibile notare che le fasi di *Tunneling* risultano sempre particolarmente dispendiose, con un costo computazionale crescente man mano che la parametrizzazione del giacimento si arricchisce. Tralasciando una possibile rigorosa spiegazione matematica di ciò, dall'analisi delle prestazioni dei vari metodi di ottimizzazione testati risulta chiaro che una semplice tecnica, basata essenzialmente sulla forza bruta, quale l'algoritmo *Multistart* è in grado di fornire dei minimi locali migliori utilizzando un tempo macchina inferiore. Inoltre, se consideriamo come riferimento per la calibrazione di uno qualunque dei tre modelli il valore della funzione oggetto ottenuto dall'applicazione del metodo di Levenberg – Marquardt, l'algoritmo di *Tunneling* nella sua versione originale (BFGS + BFGS) richiede più fasi di minimizzazione e successivo *Tunneling* se si desidera ottenere un valore migliore della funzione obiettivo. In altre parole, per raggiungere lo stesso valore della funzione oggetto ottenuto applicando l'algoritmo di Levenberg – Marquardt

il metodo di *Tunneling* richiede da 23 (caso Punq5) a 5 (caso Punq30) volte lo sforzo computazionale.

Il problema potrebbe risiedere sia nella parametrizzazione adottata per i tre modelli, sia nel tipo di algoritmo locale implementato all'interno del codice di *Tunneling*. Come è noto, infatti, il metodo locale BFGS è troppo facilmente attratto da minimi locali non significativi, e ciò può portare ad un generale rallentamento dell'ottimizzazione della funzione obiettivo. Oppure, considerando le caratteristiche di convergenza globali dell'algoritmo di *Tunneling*, è possibile che esso non sia in grado di esplorare in modo esaustivo lo spazio delle variabili per localizzare un nuovo favorevole punto per inizializzare nuovamente la minimizzazione. È probabile che un'estesa ed esaustiva applicazione del metodo *Multistart* ai casi test considerati possa portare all'individuazione di più minimi locali caratterizzati da un piccolo valore della funzione obiettivo.

Lo studio del caso test PunqS3 ha portato alla luce le potenzialità del metodo di Levenberg – Marquardt come algoritmo di ottimizzazione locale. Per questo motivo è nata l'idea di implementare un algoritmo ibrido LM + BFGS ed applicarlo al caso test PunqS3. Questa scelta si è rivelata proficua, dato che le prestazioni di tale ibrido si sono rivelate molto soddisfacenti. Infatti, da un punto di vista computazionale, ci ha permesso di risparmiare all'incirca il 50% del tempo di calcolo rispetto al metodo di *Tunneling* tradizionale localizzando allo stesso tempo un ugual numero di minimi locali per il caso test Punq30. Tuttavia, il costo delle fasi di *Tunneling* è rimasto sostanzialmente invariato, e da ciò possiamo dedurre che il vero collo di bottiglia per questo algoritmo di ottimizzazione è rappresentato dall'utilizzo del metodo BFGS applicato alla funzione ausiliaria di *Tunneling*. Un ulteriore aspetto negativo, comune ad entrambe le implementazioni testate sul caso Punq30, è dovuto all'impossibilità pratica di migliorare il minimo locale ottenuto mediante l'applicazione di un semplice algoritmo quale il *Multistart*. Indipendentemente dunque dal metodo locale implementato, la fase di *Tunneling* non permette l'individuazione della regione che contiene il minimo calcolato con la forza bruta né di minimi locali ancora migliori.

L'idea fondamentale dell'algoritmo di *Tunneling* è comunque rimasta alla base di questa nuova implementazione: in effetti, questa metodologia si è dimostrata attraente da due punti di vista. In primo luogo, possiede caratteristiche di convergenza globali che ci

hanno permesso di localizzare più di una soluzione possibile al problema di *History Matching*. In secondo luogo, questa metodologia è molto flessibile poiché consente di implementare differenti algoritmi locali all'interno del codice sorgente e testarne le proprietà, scegliendo poi accuratamente quello che più si confà alle nostre esigenze.

Per concludere, possibili sviluppi del presente lavoro possono prendere in considerazione l'applicazione della metodologia di *Tunneling* a dei giacimenti reali. Ciò rappresenta il banco di prova più importante per tale algoritmo di ottimizzazione; in questo modo, infatti, è possibile verificare se le prestazioni del metodo e i risultati ottenuti dipendano oppure no dalla struttura dello spazio dei parametri tipica del caso test PunqS3.

Inoltre, è possibile considerare una completa riparametrizzazione del problema già esaminato, eliminando o accorpendo i parametri in base allo studio della matrice hessiana della funzione obiettivo. Ciò conduce ad una nuova zonazione del caso test e quindi ad un modello completamente diverso da calibrare. Le prestazioni del metodo di *Tunneling* possono dipendere anche dalla qualità del modello di giacimento considerato, e sarebbe dunque possibile analizzare eventuali differenze rispetto ai casi già esaminati in questo lavoro.

Infine, annotando il decremento del costo computazionale del processo di *History Matching* dovuto all'implementazione del metodo LM + BFGS, sarebbe interessante studiare una possibile metodologia ibrida di ottimizzazione in cui l'algoritmo di *Tunneling* venisse ridefinito mediante una funzione ai minimi quadrati, alla quale poi applicare il metodo di Levenberg – Marquardt. Il nuovo ibrido, LM + LM, potrebbe ulteriormente migliorare le prestazioni degli algoritmi testati nel presente lavoro.