

UNIVERSITÀ DI PISA

Facoltà di Scienze Matematiche, Fisiche e  
Naturali  
Corso di Laurea in Informatica

Relazione sul tirocinio:

**Porting ed estensione di codice  
C++ per l'ottimizzazione non  
differenziabile**



**Tutore:**  
Antonio Frangioni

**Candidato:**  
Giacomo d'Antonio

13 Ottobre 2006

# Indice

<b>1</b>	<b>I metodi del subgradiente</b>	<b>6</b>
1.1	Rilassamenti Lagrangiani . . . . .	6
1.2	Algoritmo generico del subgradiente . . . . .	7
1.3	Algoritmo generico del Volume . . . . .	8
1.4	Valutazione approssimata della funzione obiettivo e del subgradiente . . . . .	9
1.5	Metodo del deflected subgradient . . . . .	10
1.6	Problema NDO vincolato e conditional subgradient . . . . .	11
1.7	Generazione di soluzioni primali . . . . .	15
1.8	Condizioni di terminazione . . . . .	16
1.9	Metodi del subgradiente incrementale . . . . .	18
<b>2</b>	<b>Il metodo condizionale deflesso</b>	<b>21</b>
2.1	Analisi di convergenza . . . . .	24
2.1.1	Lemmi Tecnici . . . . .	24
2.1.2	Polyak stepsize . . . . .	27
2.1.3	Diminishing stepsize . . . . .	29
2.1.4	Target Value stepsize . . . . .	36
<b>3</b>	<b>Un unico schema algoritmico</b>	<b>41</b>
3.1	Analisi di convergenza . . . . .	47
<b>4</b>	<b>Implementazione e test computazionali</b>	<b>51</b>
4.1	Implementazione . . . . .	51
4.2	Test computazionali . . . . .	55
4.2.1	Il politopo semimetrico . . . . .	55
4.2.2	Risultati numerici . . . . .	57
<b>A</b>	<b>Algoritmi proposti in letteratura</b>	<b>79</b>
A.1	Polyak stepsizes . . . . .	79
A.2	Target Value stepsizes . . . . .	80
A.3	Proprietà teoriche dell'algoritmo del volume . . . . .	81

# Introduzione

In questo documento ci occupiamo dello studio teorico e implementativo di algoritmi per l'ottimizzazione non differenziabile. Presentiamo qui il lavoro svolto durante un tirocinio tenutosi presso il Dipartimento di Informatica dell'Università di Pisa nell'ambito del corso di laurea di primo livello in Informatica. Possiamo brevemente schematizzare le attività svolte durante il tirocinio come segue:

- studio bibliografico,
- studio di convergenza,
- progetto implementativo,
- sviluppo del codice,
- test computazionali.

Queste attività sono state svolte pressappoco in sequenza e allo stesso modo è organizzata questa relazione.

I problemi di ottimizzazione non differenziabile si presentano frequentemente, specialmente nell'ambito della Ricerca Operativa. Una delle applicazioni più comuni consiste nella risoluzione di duali lagrangiani, una tecnica molto comune utilizzata sia nell'ambito della risoluzione esatta di problemi difficili (si pensi agli algoritmi *Branch and bound*) che nelle cosiddette *euristiche lagrangiane*. Infatti il caso lagrangiano è talmente importante che faremo spesso riferimento ad esso, concedendoci delle occasionali eccezioni rispetto allo sforzo di astrazione e sottolineando le proprietà specifiche della funzione lagrangiana rispetto al più generico caso non differenziabile.

Non dovrebbe stupire il fatto che i problemi di ottimizzazione non differenziabile si presentano come difficili e, infatti, gli algoritmi risolutivi per questa classe di problemi sono ancora oggi oggetto di ricerca. Quasi tutti gli algoritmi disponibili per questo problema si possono dividere in tre classi dai confini molto sfumati: algoritmi dei piani di taglio, algoritmi Bundle e algoritmi del subgradiente. Si può verificare come gli algoritmi dei piani di taglio e quelli Bundle siano più efficaci degli algoritmi del subgradiente ma a loro volta meno efficienti. Comunque molti algoritmi Bundle si possono interpretare come casi particolari dei metodi dei piani di taglio e molti algoritmi del subgradiente (in particolare i cosiddetti algoritmi del Volume) come versioni "economiche" dei metodi Bundle. La differenza nell'impiego delle risorse computazionali (in particolare il tempo necessario alla risoluzione del problema) delle tre classi sopra esposte è tale da rendere ancora popolari i metodi del subgradiente, nonostante essi siano poco

accurati. D'altra parte i metodi dei piani di taglio e i metodi Bundle, nel caso lagrangiano e più in generale quando si minimizza una funzione lineare a tratti, terminano finitamente e forniscono il valore ottimo esatto, mentre i metodi del subgradiente convergono solo asintoticamente. Spesso il Bundle è ritenuto un buon compromesso tra il difetto di accuratezza dei metodi del subgradiente e quello di efficienza dei piani di taglio; ad ogni modo queste valutazioni dipendono fortemente dall'ambito di applicazione. In questo documento ci occuperemo esclusivamente dei metodi del subgradiente ma avremo modo di confrontare i risultati computazionali ottenuti con un metodo di tipo Bundle.

Cerchiamo ora di discutere in maniera maggiormente formale gli aspetti fondamentali dell'ottimizzazione non differenziabile; consideriamo il problema

$$\min_{y \in Y} f(y)$$

dove  $f(y)$  è una funzione convessa possibilmente non differenziabile e  $Y$  è un insieme convesso. L'idea alla base degli algoritmi del subgradiente consiste nel generalizzare il metodo del gradiente per l'ottimizzazione di funzioni differenziabili al caso non differenziabile. In particolare il primo metodo storicamente proposto generalizza il cosiddetto metodo del gradiente ottimo (o *steepest descent*), ma altri metodi si riconducono a differenti metodi per il caso differenziabile, ad esempio metodi come quello deflesso di Camerini-Fratta sono molto simili ai metodi del gradiente coniugato. A prima vista potrebbe stupire il fatto che generalizziamo metodi comunemente utilizzati nella risoluzione di sistemi lineari ad un problema di ottimizzazione. D'altra parte è fatto ben noto che la soluzione di sistemi lineari è equivalente all'ottimizzazione di una funzione quadratica; l'aspetto più interessante è, invece, la possibilità di generalizzare quest'equivalenza al caso non differenziabile definendo una teoria della dualità per i programmi non differenziabili detta *dualità di Fenchel*. Comunque riguardo la dualità nell'ambito non differenziabile studieremo soltanto un caso particolare: la dualità lineare tra un duale lagrangiano e un rilassamento convessificato.

Per costruire un approccio simile a quello del gradiente nel caso non differenziabile dobbiamo generalizzare le proprietà e gli operatori differenziali, per questo definiamo i concetti di subgradiente e di subdifferenziale:

**Definizione 1** *Data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  convessa, possibilmente non differenziabile, si definisce il subdifferenziale di  $f$  in  $y_0 \in \mathbb{R}^n$  come:*

$$\partial f(y_0) = \{g \in \mathbb{R}^n : f(y) \geq f(y_0) + g(y - y_0), \quad \forall y \in \mathbb{R}^n\}; \quad (1)$$

*un elemento di  $\partial f(y_0)$  si dice subgradiente di  $f$  in  $y_0$ .*

Qui richiediamo esplicitamente la convessità della funzione  $f$  che, oltre a garantire che un ottimo locale sia anche un ottimo globale, assicura in ogni punto l'esistenza di iperpiani di supporto, in altre parole di subgradienti. Si nota subito come un subgradiente definisce una funzione lineare affine, che chiamiamo modello della funzione e indichiamo con  $\phi(y)$ , molto simile ad un'approssimazione di Taylor al prim'ordine. Tuttavia si perdono le proprietà differenziali, in particolare è difficile descrivere  $R(y) = f(y) - \phi(y)$ ; è anche difficile andare oltre il primo ordine, nonostante esistano tentativi di questo tipo, ad esempio i cosiddetti metodi *space dilation*. Inoltre un subgradiente non fornisce più una direzione di decrescita e questo condiziona sia la velocità che l'analisi di convergenza. In altri termini lo studio del caso differenziabile si basa su proprietà

topologiche (vale a dire la continuità) e geometriche (la convessità) della funzione anziché differenziali. In sostanza quello che rimane di un'approssimazione lineare di Taylor passando al non differenziabile è la sua proprietà di supporto al grafico della funzione, comunque vedremo che anche questa verrà rilassata ed utilizzeremo modelli che non solo non supportano la funzione ma la tagliano.

I metodi del subgradiente quindi considerano ad ogni iterazione una soluzione  $y \in Y$  e calcolano un subgradiente della funzione in  $y$ ,  $g = g(y) \in \partial f(y)$ , per poi spostarsi lungo una direzione che dipende da  $g$ . La scelta della direzione verso cui spostarsi e della lunghezza del passo in quella direzione è uno dei parametri che differenziano i diversi algoritmi del subgradiente. Tipicamente si considerano problemi vincolati (ovvero  $Y \subset \mathbb{R}^n$ ), per questo è necessario che gli algoritmi tengano conto dei vincoli nei passi iterativi. Questo di solito si ottiene proiettando subgradienti, direzioni e soluzioni sull'insieme ammissibile: il modo in cui vengono gestite le proiezioni è un'altra scelta algoritmica. Un'altra importante distinzione è quella tra algoritmi del subgradiente ordinario e algoritmi del Volume. I primi ad ogni iterazione calcolano una direzione e si spostano lungo quella, gli altri, invece, ad ogni iterazione calcolano una direzione e la relativa soluzione e si spostano solo se si ottiene un sufficiente miglioramento nella funzione obiettivo (*serious step*).

Durante lo studio bibliografico si è posta l'esigenza di sviluppare una teoria di convergenza per i metodi del subgradiente condizionale deflesso. Infatti in letteratura non sono presenti studi che descrivono l'integrazione tra il metodo del subgradiente deflesso, un metodo per affrontare uno dei problemi di lenta convergenza dei metodi del subgradiente, e il metodo condizionale per la gestione dei problemi di ottimizzazione vincolata. Inoltre molte delle analisi di convergenza del metodo deflesso presenti in letteratura utilizzano delle condizioni non facilmente verificabili. In questo documento sviluppiamo un'analisi di convergenza innovativa per diversi metodi del subgradiente condizionale deflesso; in alcuni casi estendendo dimostrazioni già esistenti per il metodo deflesso, in altri proponendo metodi innovativi, dimostrandone la convergenza. Queste analisi di convergenza, presentate nel capitolo 2, forniscono condizioni implementabili. Sono state quindi riformulate delle analisi di convergenza che prevedono condizioni non garantibili né da un punto di vista teorico, né da un punto di vista implementativo, per fornire delle condizioni che possono essere verificate semplicemente. Inoltre, ad eccezione dei recenti [6] e [24], i metodi deflessi vengono sempre analizzati con degli stepsize non implementabili; in particolare i cosiddetti stepsize di Polyak che richiedono la conoscenza a priori del valore ottimo obiettivo. Nel capitolo 2 presentiamo delle analisi di convergenza per le principali classi di stepsize di utilizzo comune.

Il capitolo 1 presenta un'introduzione ai metodi del subgradiente descrivendo i diversi schemi algoritmici presenti in letteratura. In particolare nella sezione 1.1 introduciamo brevemente la teoria dei rilassamenti lagrangiani; nelle sezioni 1.2 e 1.3 consideriamo rispettivamente la forma generica di un algoritmo del subgradiente e di un algoritmo del volume; nella sezione 1.4 introduciamo il metodo del subgradiente approssimato che permette di utilizzare questi algoritmi anche in quelle situazioni in cui non è possibile conoscere il valore esatto della funzione o calcolare un subgradiente esatto; nella sezione 1.5 discutiamo uno dei problemi di lenta convergenza dei metodi del subgradiente e proponiamo un comune approccio risolutivo; nella sezione 1.6 affrontiamo i problemi di ottimizzazione vincolati e introduciamo la soluzione proposta da [12] detta del

*conditional subgradient*; nella sezione 1.7 introduciamo dei metodi per calcolare delle soluzioni primali; nella sezione 1.8 discutiamo diverse condizioni di terminazione e nella sezione 1.9 presentiamo il metodo del subgradiente incrementale. Nel capitolo 2 presentiamo un'analisi approfondita di vari metodi del subgradiente condizionale deflesso concentrandoci sulle proprietà di convergenza. Nel capitolo 3 introduciamo uno schema algoritmico unico che ci guiderà verso l'implementazione e mostreremo come ricondurre ad esso diversi metodi comunemente utilizzati; lo schema presentato generalizza anche i metodi oggetto della trattazione al capitolo 2. Nel capitolo 4 descriviamo l'implementazione degli algoritmi discussi precedentemente, presentiamo una classe di problemi su cui testiamo gli algoritmi e discutiamo i risultati ottenuti.

# Capitolo 1

## I metodi del subgradiente

Come già accennato i metodi del subgradiente, sebbene computazionalmente efficienti, convergono molto lentamente se confrontati con gli altri algoritmi a disposizione per l'ottimizzazione non differenziabile. Questa considerazione ha generato numerosi sforzi volti a migliorare le prestazioni degli algoritmi del subgradiente producendo svariate estensioni teoriche ed accorgimenti empirici. Questo capitolo è essenzialmente uno studio bibliografico mirato a recensire le diverse varianti dei metodi del subgradiente in letteratura; i metodi qui presentati, insieme ai metodi innovativi discussi nel capitolo 2 sono stati implementati e testati come discusso nei capitoli 3 e 4. Prima di presentare i diversi metodi discutiamo il caso lagrangiano e l'algoritmo del subgradiente come originariamente proposto.

### 1.1 Rilassamenti Lagrangiani

Un utilizzo tipico degli algoritmi per l'ottimizzazione non differenziabile è la risoluzione di duali Lagrangiani. Consideriamo un problema di Programmazione Lineare Intera (PLI), che si presenta nella forma:

$$(P) \quad \max_x \{cx : Ax \leq b, Ex \leq d, x \in \mathbb{N}^n\} \quad (1.1)$$

Spesso problemi di questo tipo sono “difficili” da risolvere, mentre rilassando alcuni vincoli (detti *vincoli complicanti*, nel nostro esempio i vincoli  $Ax \leq b$ ) si ottiene un problema risolvibile in tempo ragionevolmente breve. Perciò si definisce il *rilassamento Lagrangiano* di parametro  $y$ :

$$(P_y) \quad \max \{cx + y(b - Ax) : Ex \leq d, x \in \mathbb{N}^n\}. \quad (1.2)$$

Il problema di determinare il miglior parametro  $y$  è conosciuto come *duale Lagrangiano* ed è risolvibile mediante un algoritmo del subgradiente dove  $f(y) = z(P_y)$  e  $z(P_y)$  indica il valore obiettivo ottimo del problema  $(P_y)$ . Pertanto il duale Lagrangiano si presenta nella forma:

$$\min_y \{z(P_y) : y \geq 0\}. \quad (1.3)$$

Il problema (1.2) può essere riscritto come:

$$yb + \max \{(c - yA)x : Ex \leq d, x \in \mathbb{N}^n\}.$$

Dunque risolvere un problema di rilassamento Lagrangiano vuol dire risolvere un rilassamento per eliminazione di vincoli con una diversa funzione costo (parametrica rispetto ad  $y$ ). Pertanto, se per qualche  $y$  il problema è illimitato lungo la direzione  $c - yA$ , la funzione  $f(y)$  assumerà valore  $+\infty$ . La funzione quindi è una cosiddetta *real extended function*, ovvero:

$$f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}.$$

Vedremo nella sezione 1.6 che questo tipo di problema è equivalente al problema di ottimizzazione vincolata di una funzione reale, vale a dire:

$$\min_{y \in Y} \{z(P_y)\}, \quad Y = \{y \in \mathbb{R}^n : y \geq 0, f(y) < +\infty\};$$

l'insieme  $Y$  è evidentemente convesso; si noti come secondo le nostre definizioni un duale lagrangiano è sempre un problema vincolato dove sono presenti dei vincoli espliciti di non negatività che derivano dalla formulazione dei vincoli primali come disequaglianze ed eventualmente dei vincoli impliciti dovuti alla limitatezza del rilassamento lagrangiano.

Si dimostra che il duale Lagrangiano è il duale lineare del cosiddetto *rilassamento convessificato* di  $(P)$ , vale a dire:

$$(\tilde{P}) \quad \max_x \{cx : Ax \leq b, x \in \text{Conv}(X)\}. \quad (1.4)$$

dove  $X = \{x \in \mathbb{N}^n : Ex \leq d\}$  e  $\text{Conv}(X)$  indica l'involuppo convesso di  $X$ . Nel seguito vengono presentati alcuni algoritmi che producono anche soluzioni primali, ossia soluzioni ( $\varepsilon$ -)ottime di  $(\tilde{P})$ .

Un problema di duale Lagrangiano quindi può essere interpretato e risolto come un problema di ottimizzazione non differenziabile. In questo caso la seguente proprietà ci fornisce un buon metodo per calcolare dei subgradienti della funzione  $z(P_y)$  in un punto.

**Proprietà 1** *sia  $\bar{x}$  una soluzione ottima di (1.2), allora  $b - A\bar{x}$  è un subgradiente di  $f(y) = z(P_y)$  in  $y$ .*

## 1.2 Algoritmo generico del subgradiente

Presentiamo, qui, la forma generica di un algoritmo del subgradiente, senza considerare le caratteristiche che “raffinano” il metodo; vale a dire le regole per la scelta della lunghezza del passo di spostamento  $\nu$ , i metodi per “deflettere” la direzione di spostamento, la gestione di problemi vincolati, l'introduzione di ulteriori e più efficaci criteri di terminazione e la possibilità di generare soluzioni primali di un duale lagrangiano. Questi aspetti verranno introdotti e discussi in seguito.

L'algoritmo del subgradiente ordinario quindi nella sua forma più semplice si presenta come segue:

**Inizializzazione** Si sceglie una soluzione iniziale  $y \in Y$ . Si inizializza  $\bar{f} = +\infty$ .

**Passo 1 (Object evaluation)** Si calcolano  $f(y)$  e  $g \in \partial f(y)$ . Se  $f(y) < \bar{f}$ , allora si aggiornano  $\bar{f} = f(y_k)$ ,  $\bar{y} = y$ .

**Passo 2** (*Stopping criterion*) Si valutano i criteri di terminazione e si termina se uno di essi vale. Un criterio valido per tutti gli algoritmi è il seguente:

$$\|g\| \leq \psi$$

per un fissato valore  $\psi$ .

**Passo 3** (*Stepping forward*) Si calcola il passo  $\nu$  e la nuova soluzione

$$y := P_Y(y - \nu g).$$

Si inizia una nuova iterazione.

Questo algoritmo utilizza le seguenti tecniche:

**Record point** Durante l'esecuzione dell'algoritmo teniamo traccia del miglior punto ottenuto (la variabile  $\bar{y}$ ) e del valore della funzione ad esso associato (la variabile  $\bar{f}$ ).

**Terminazione** In generale una condizione di ottimalità sufficiente (e necessaria se il problema non è vincolato) è  $0 \in \partial f(y)$  (cfr. sez. 1.8). Pertanto se vale  $\|g_k\| = 0$  la soluzione trovata è ottima, tuttavia i metodi del subgradiente garantiscono solo convergenza asintotica. Per questo si può scegliere di terminare l'algoritmo quando si trova una soluzione sufficientemente vicina all'ottimo (cioè quando il subgradiente è "sufficientemente piatto") introducendo l'errore  $\psi$  sulla norma del subgradiente.

### 1.3 Algoritmo generico del Volume

Gli algoritmi del volume mantengono l'informazione relativa al cosiddetto *centro di stabilità*: il punto generato dall'ultimo serious step,  $\tilde{y}$ . Ad ogni iterazione si valuta il valore della funzione obiettivo nel punto generato e secondo un criterio di aggiornamento specifico si decide se aggiornare il centro di stabilità eseguendo un *serious step* o eseguire un *null step*. Di fatto quindi gli algoritmi del volume, per ogni ciclo esterno di *serious step*, eseguono un ciclo interno di *null step* in cui provano delle soluzioni (dette *tentative solutions*) scartandole se non vengono ritenute adeguate, utilizzando comunque l'informazione generata dai null step per calcolare la direzione (detta *tentative direction*) e la lunghezza del passo all'iterazione successiva.

**Inizializzazione** Si sceglie una soluzione iniziale  $y \in Y$ . Si inizializzano  $\bar{y} = y$ ,  $\bar{f} = f(y)$ ,  $\tilde{y} = y$ ,  $\tilde{f} = f(y)$ . Si calcolano  $d = g \in \partial f(y)$ .

**Passo 1** (*Stepping forward*) Si calcola il passo  $\nu$  e la nuova soluzione

$$y = P_Y(\tilde{y} - \nu d).$$

**Passo 2** (*Object evaluation*) Si calcolano  $f(y)$  e  $g \in \partial f(y)$ . Se  $f(y) < \bar{f}$ , allora si aggiornano  $\bar{f} = f(y)$ ,  $\bar{y} = y$ .

**Passo 3** (*Sufficient descent detection*) Se  $f(y) < \tilde{f} - \delta$  (*serious step*), allora si aggiornano  $\tilde{y} = y$ ,  $\tilde{f} = f(y)$ , si calcola il nuovo valore di  $\delta$  secondo le regole dell'algoritmo specifico; altrimenti passa al passo 4 (in tal caso l'iterazione corrente si dice *null step*).

**Passo 4** (*Stopping criterion*) Si valutano tutti i criteri di terminazione e si termina se uno di essi vale. Un criterio valido per tutti gli algoritmi è il seguente:

$$\|g\| \leq \psi$$

per un fissato valore  $\psi$ .

**Passo 5** (*Aggregate solutions*) Si calcola il coefficiente  $\alpha$  e si aggiornano le medie, in particolare si aggiorna:

$$d := \alpha g + (1 - \alpha)d$$

Si inizia una nuova iterazione.

Si noti come nell'algoritmo proposto in origine in [4] si abbia  $\delta = 0$  ad ogni iterazione.

## 1.4 Valutazione approssimata della funzione obiettivo e del subgradiente

Spesso non si è in grado di calcolare il valore esatto della funzione e un suo subgradiente nel punto, oppure può risultare computazionalmente troppo costoso. Per questi motivi è utile estendere i metodi del subgradiente al caso in cui si valutano approssimativamente la funzione ed i suoi subgradienti. Definiamo, perciò, subgradiente e subdifferenziale approssimati:

**Definizione 2** Si definisce  $l'$ - $\varepsilon$ -subdifferenziale come:

$$\partial_\varepsilon f(y) = \{g \in \mathbb{R}^n : f(z) \geq f(y) + g^T(z - y) - \varepsilon, \quad \forall z \in \mathbb{R}^n\} \supseteq \partial f(y) \quad (1.5)$$

e si dice  $\varepsilon$ -subgradiente un elemento di  $\partial_\varepsilon f(y)$ .

Permettendo quindi di utilizzare un  $\varepsilon$ -subgradiente si ottiene il metodo del subgradiente approssimato; sia [1] che [9] dimostrano la convergenza di questo metodo sotto l'ipotesi che la successione degli errori tenda a zero e utilizzando le comuni regole per la scelta della lunghezza del passo.

Di particolare interesse è il caso lagrangiano. A volte anche calcolare  $f(y)$  (ad esempio risolvere un problema di PLI) può essere computazionalmente difficile. Tuttavia possiamo utilizzare degli algoritmi approssimati e trovare, così, soluzioni  $\varepsilon$ -ottime. In questo contesto vale la seguente proprietà:

**Teorema 2** Sia  $\bar{x}$  una soluzione  $\varepsilon$ -ottima del problema,:

$$f(y) = \max\{cx + y(b - Ax) : x \in X\},$$

ovvero  $\forall x \in X : \quad c\bar{x} + y(b - A\bar{x}) \geq cx + y(b - Ax) - \varepsilon$ , allora  $b - A\bar{x}$  è un  $\varepsilon$ -subgradiente di  $f(y)$ :  $(b - A\bar{x}) \in \partial_\varepsilon f(y)$ .

*Dimostrazione:* Sia  $\bar{x}$  una soluzione  $\varepsilon$ -ottima di  $(P_y)$ .  
 $\max_x (cx + z(b - Ax)) \geq c\bar{x} + z(b - A\bar{x}) \Rightarrow$   
 $f(z) \geq (\max_x ((c - yA)x) + yb) - ((c - yA)\bar{x} + yb) - \varepsilon + c\bar{x} + z(b - A\bar{x}) \Rightarrow$   
 $f(z) \geq f(y) - y(b - A\bar{x}) + z(b - A\bar{x}) - \varepsilon \Rightarrow$

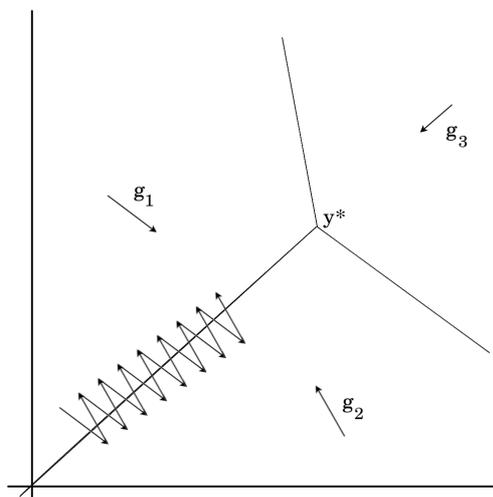


Figura 1.1: Zigzagging del metodo del subgradiente ordinario

$$f(z) \geq f(y) + (b - A\bar{x})(z - y) - \varepsilon \Rightarrow (b - A\bar{x}) \in \partial_\varepsilon f(y).$$

Si noti come la proprietà 1 sia un caso particolare di questo teorema in cui  $\varepsilon = 0$ .

Dunque una breve estensione teorica permette di utilizzare calcoli approssimati della funzione e del subgradiente. Tuttavia questi strumenti sono utili anche nello sviluppo di analisi teoriche; vedremo in seguito come molti dei metodi del subgradiente e del volume possano essere interpretati come metodi approssimati. In questi casi l'inesattezza dei subgradienti non è più uno strumento per facilitare il calcolo ma una proprietà teorica del metodo.

## 1.5 Metodo del deflected subgradient

Nell'applicazione del metodo del subgradiente ordinario spesso si riscontra un fenomeno di lenta convergenza (conosciuto anche come fenomeno di *zigzagging*) dovuto al fatto che il subgradiente scelto dall'algoritmo (la direzione verso cui ci si muove) può essere "quasi opposto" al subgradiente (la direzione) generato all'iterazione precedente. In altri termini, se accade  $d_k d_{k-1} < 0$  per più iterazioni l'algoritmo si muoverà verso l'ottimo ma molto lentamente "vagando" per lo spazio delle soluzioni ammissibili ( $d_k(y_k - y^*) \approx 0$ ).

La figura 1.1 mostra un esempio di *zigzagging* del metodo del subgradiente ordinario in  $\mathbb{R}^2$ : lo spazio delle soluzioni è diviso in tre regioni su ognuna delle quali la funzione è lineare (rispettivamente di gradiente  $g_1$ ,  $g_2$  e  $g_3$ ), l'algoritmo si muove attorno ai confini delle regioni (che sono anche i punti di non differenziabilità), si noti che  $g_1$  e  $g_2$  sono quasi opposti.

L'approccio generalmente utilizzato per affrontare questa situazione consiste nel "deflettere" la direzione; ovvero utilizzare come direzione una combinazione lineare del subgradiente e della direzione all'iterazione precedente. L'idea è di garantire  $d_k d_{k-1} \geq 0$ . Uno dei primi metodi presenti in letteratura (cfr. [21])

utilizza:

$$d := g + \eta d, \quad \eta = \begin{cases} -\tau \frac{gd}{\|d\|^2}, & \tau \in [1, 2) \text{ se } gd < 0 \\ 0 & \text{altrimenti} \end{cases} \quad (1.6)$$

Questo metodo, ideato da Camerini, Fratta e Maffioli, si muove lungo il subgradiente se non provoca zigzagging, altrimenti deflette il subgradiente fino a renderlo almeno perpendicolare alla direzione precedente.

Un'altra possibile scelta utilizza come direzione una combinazione convessa del subgradiente e della direzione all'iterazione precedente:

$$d := \alpha g + (1 - \alpha)d, \quad \alpha \in (0, 1]. \quad (1.7)$$

L'utilità di questa scelta è principalmente dovuta alla sua somiglianza con i metodi del volume, che calcolano la direzione come combinazione convessa di più subgradienti. In particolare noi utilizzeremo questo tipo di deflessione per poter unificare in un unico algoritmo i diversi metodi presentati in questo capitolo. Inoltre (1.7) garantisce  $d \in \partial_\varepsilon f(y)$  (si veda cap. 2).

Questo metodo elimina il fenomeno di *zigzagging* (formalmente garantisce  $\forall k \ d_k d_{k-1} \geq 0$ ) se e solo se vale la seguente condizione:

$$\forall k \quad 0 < \alpha_k < \frac{\|d_{k-1}\|^2}{\|d_{k-1}\|^2 - g_k d_{k-1}}. \quad (1.8)$$

Ad ogni modo tipicamente non si forza  $\alpha_k$  a rispettare (1.8) in quanto computazionalmente si verifica l'efficienza del metodo anche senza preoccuparsi di queste proprietà teoriche.

## 1.6 Problema NDO vincolato e conditional subgradient

Quando il problema di ottimizzazione non differenziabile è vincolato (cioè  $y \in Y \subset \mathbb{R}^n$ ) non è detto che il subgradiente della funzione obiettivo sia una direzione ammissibile; oppure, pur muovendosi lungo una direzione ammissibile, non abbiamo a disposizione una limitazione dello *stepsize* che garantisca l'ammissibilità della soluzione generata. Si noti (cfr. sez. 1.1) che questo caso può verificarsi anche nella soluzione di duali Lagrangiani.

Il metodo del subgradiente quindi potrebbe generare una soluzione non ammissibile (cioè  $y \notin Y$ ); per questo motivo i metodi del subgradiente proiettano ad ogni iterazione la soluzione generata sull'insieme ammissibile (*metodo di Polyak*). Tuttavia questo metodo può risultare molto lento in quanto il subgradiente può essere quasi normale in direzione rispetto alla frontiera dell'insieme ammissibile. Un esempio di questo comportamento è riportato nella figura 1.2 dove la proiezione della soluzione generata sull'insieme ammissibile annulla quasi completamente lo spostamento. Inoltre anche in vicinanza della soluzione ottima ammissibile il subgradiente potrebbe essere in norma molto alto, falsando così la condizione di terminazione.

Si vuole quindi estendere il metodo per tener conto dei vincoli del problema; ottenendo un vantaggio in termini di efficienza. L'idea alla base del metodo

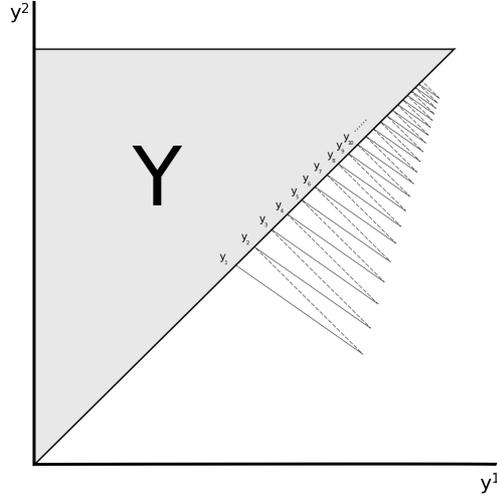


Figura 1.2: Un esempio di *slow convergence*.

condizionale -esposto in [9]- consiste nel minimizzare una funzione, detta *essential objective*, che “incorpora” l’informazione sui vincoli:  $f^Y(y) = f(y) + \phi_Y(y)$  dove  $\phi_Y(y)$  è la cosiddetta funzione indicatrice ed è definita come segue:

$$\phi_Y(y) = \begin{cases} 0 & \text{se } y \in Y \\ +\infty & \text{altrimenti} \end{cases} \quad (1.9)$$

$f^Y$  è ancora una funzione convessa e  $\forall y \in \mathbb{R}^n : f(y) \leq f^Y(y)$ ; pertanto un subgradiente di  $f$  sarà anche subgradiente di  $f^Y$ , ovvero  $\forall y \in \mathbb{R}^n : \partial f(y) \subseteq \partial f^Y(y)$ . L’originalità di [9] consiste in un’estensione della teoria che permette di calcolare agevolmente un miglior subgradiente di  $f^Y$  nei punti dove  $\partial f(y) \neq \partial f^Y(y)$ . È necessario, però, introdurre i concetti di *conditional subdifferential* e le cosiddette *conical approximations*.

**Definizione 3** *Definiamo il conditional subdifferential come:*

$$\partial^Y f(y) = \{g \in \mathbb{R}^n : f(z) \geq f(y) + g^T(z - y), \quad \forall z \in Y\} \supseteq \partial f(y). \quad (1.10)$$

*Diciamo conditional subgradient o subgradiente condizionale  $g \in \partial^Y f(y)$ .*

Il metodo del *conditional subgradient* si basa sull’idea di utilizzare come direzione un subgradiente condizionale anziché uno ordinario; in questo modo vale la condizione di ottimalità per il problema vincolato  $0 \in \partial^Y f(y)$ .

Introduciamo, ora, una prima approssimazione conica:

**Definizione 4** *Definiamo il normal cone di  $Y$  nel punto  $y$  come:*

$$N_Y(y) = \begin{cases} \{\nu \in \mathbb{R}^n : \nu^T(z - y) \leq 0, \quad \forall z \in Y\} & \text{se } y \in Y, \\ \emptyset & \text{altrimenti.} \end{cases} \quad (1.11)$$

Per calcolare agevolmente un subgradiente condizionale utilizziamo il seguente risultato.

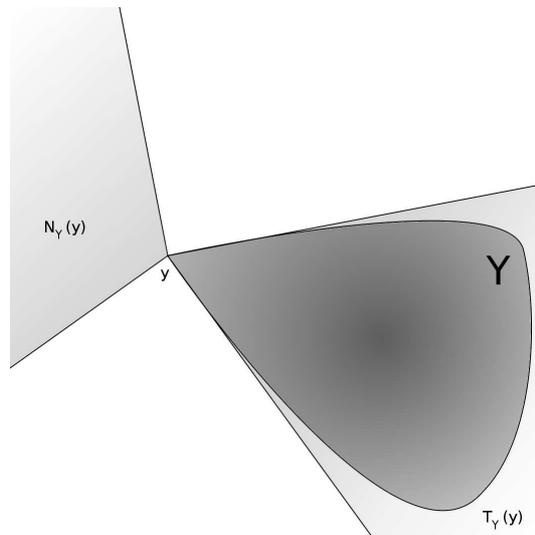


Figura 1.3: Un esempio di *tangent cone* e *normal cone*.

Vale la proprietà:

$$\partial^Y f(y) = \partial f(y) + N_Y(y), \quad y \in Y \quad (1.12)$$

e, quindi:

$$g^Y \in \partial^Y f(y) \Leftrightarrow g^Y = g + \nu \quad \wedge \quad g \in \partial f(y) \quad \wedge \quad \nu \in N_Y(y). \quad (1.13)$$

Dunque il *normal cone* in un punto è il subdifferenziale della funzione indicatrice ( $\partial\phi_Y(y) = N_Y(y)$ ): cioè, eliminando da un subgradiente di  $f$  una componente “normale” rispetto all’insieme ammissibile, otteniamo un miglior subgradiente di  $f^Y$ . La proprietà fondamentale è:  $\partial^Y f(y) = \partial f^Y(y)$ , ovvero il subdifferenziale condizionale è esattamente il subdifferenziale dell’*essential objective*.

Si noti che il *conditional subdifferential* così definito contiene anche i subgradienti della funzione obiettivo. Perciò anche il metodo del subgradiente ordinario può essere interpretato come metodo condizionale.

Fino ad ora quindi questa estensione teorica ci permette di utilizzare subgradienti condizionali, ma ci dice anche che non sempre questo migliora l’efficienza del metodo iterativo (banalmente il metodo non condizionale rientra in questa analisi); abbiamo bisogno di un metodo per calcolare “buoni” subgradienti condizionali.

Sappiamo che il subgradiente condizionale di norma minima è per certo una direzione tangente all’insieme ammissibile, oltre ad essere la direzione di maggior decrescita (cfr. [9]); in particolare se la regione ammissibile è poliedrale il subgradiente condizionale di norma minima è una direzione ammissibile. Tuttavia nel caso tipico (ad esempio in presenza di un oracolo che modella la funzione) si ha a disposizione un generico subgradiente della funzione obiettivo e non è possibile individuare il subgradiente condizionale di norma minima.

Per questi motivi si sviluppa il cosiddetto *subgradient projection method*, caso particolare del metodo condizionale. Introduciamo, preliminarmente, un’altra approssimazione conica:

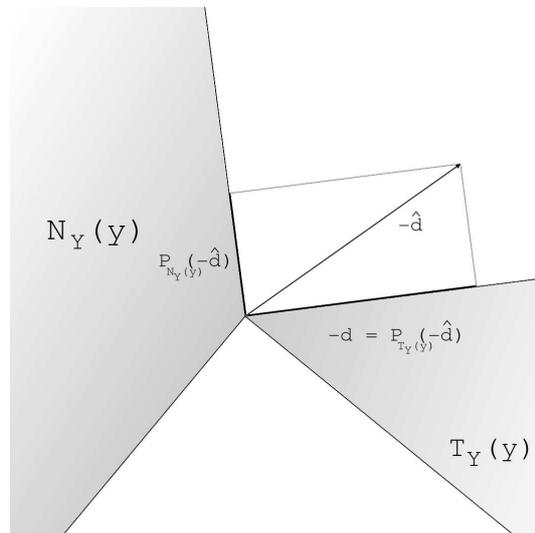


Figura 1.4: Proiezione della direzione sui coni  $T_Y(y)$  e  $N_Y(y)$ .

**Definizione 5** Definiamo il tangent cone dell'insieme  $Y$  nel punto  $y$  come:

$$T_Y(y) = \{\nu \in \mathbb{R}^n : \nu^T z \leq 0, \quad \forall z \in N_Y(y)\}; \quad (1.14)$$

ovvero il cono polare rispetto a  $N_Y(y)$ .

La figura 1.3 mostra un esempio di tangent cone e normal cone in un punto. La figura mostra inoltre come il tangent cone non sia, in generale, il cono delle direzioni ammissibili ma la sua chiusura; tuttavia nel caso in cui  $Y$  sia poliedrale il cono delle direzioni ammissibili è chiuso e, pertanto, coincide con il tangent cone. Si noti che le definizioni di normal cone e tangent cone fornite in questa sezione possono non valere per insiemi non convessi; in [13] si trovano definizioni più generiche. In particolare, se l'insieme  $Y$  non è convesso, tangent cone e normal cone possono non essere polari; [16] indica le condizioni sull'insieme  $Y$  per cui valgono le definizioni da noi utilizzate.

Abbiamo visto come la scelta della direzione migliore richiederebbe l'esplorazione del subdifferenziale, generalmente difficile. Possiamo, però, minimizzare in norma la componente "facile" della funzione, vale a dire la funzione indicatrice. Vogliamo, cioè, annullare la componente normale all'insieme ammissibile di un dato subgradiente di  $f$ . Dato il subgradiente della funzione obiettivo  $g(y)$  scegliamo, ad ogni iterazione, il subgradiente condizionale come soluzione del seguente problema:

$$g^Y(y) = \operatorname{argmin}\{\|g\|^2 : g \in (g(y) + N_Y(y))\} \quad (1.15)$$

e chiamiamo  $g^Y(y)$  *projected subgradient*. Il problema (1.15) può anche essere riformulato in spazio duale:

$$g^Y(y) = \operatorname{argmin}\{\|g - g(y)\|^2 : g \in -T_Y(y)\}. \quad (1.16)$$

La direzione così ottenuta è evidentemente la proiezione di  $-g(y)$  (la direzione non ammissibile) sul *tangent cone*  $T_Y(y)$  (figura 1.4). Inoltre questa direzione è ancora ammissibile sotto l'ipotesi che la regione  $Y$  sia poliedrale; tuttavia

poiché la lunghezza del passo  $\nu$  potrebbe non tener conto dei vincoli del problema è ancora necessario proiettare la soluzione ottenuta durante un'iterazione sull'insieme ammissibile.

Il metodo consiste quindi nel decomporre il subgradiente  $g(y)$  in una componente “normale” ( $P_{N_Y(y)}(g(y)) \in N_Y(y)$ ) ed una tangente ( $g_Y(y) \in T_Y(y)$ ). Si tratta della decomposizione di un vettore su componenti appartenenti a coni polari, questa è nota come *moreau decomposition* ed è mostrata in figura 1.4. Inoltre il teorema di Moreau assicura che  $P_{N_Y(y)}(g(y))$  e  $g_Y(y)$  così ottenuti sono ortogonali ( $P_{N_Y(y)}(g(y)) g_Y(y) = 0$ ).

A partire da queste premesse possiamo considerare il caso specifico in cui la regione ammissibile sia poliedrale e definita da vincoli di “box” (l'unico che consideriamo nel nostro progetto implementativo), vale a dire:

$$Y = \{y \in \mathbb{R}^n : Ay \leq b\}, \quad A = \begin{pmatrix} I_n \\ -I_n \end{pmatrix}, \quad b = \begin{pmatrix} V \\ 0 \end{pmatrix}; \quad (1.17)$$

dove  $I_n$  è la matrice identità di dimensione  $n$  e  $V$  è il vettore delle soglie delle varie componenti.

Nel caso poliedrale il *projected subgradient* si calcola come:

$$g^Y(y) = \operatorname{argmin}\{\|g - g(y)\| : A_i g \geq 0, \quad i \in \mathcal{I}(y)\}, \quad (1.18)$$

dove  $\mathcal{I}(y)$  indica l'insieme dei vincoli attivi in  $y$  ( $\mathcal{I}(y) = \{i : A_i y = b_i\}$ ).

Nel nostro caso quindi dato  $g \in \partial f(y)$  è sufficiente calcolare un subgradiente condizionale  $g^Y \in \partial_Y f(y)$  come

$$g_i^Y = \begin{cases} 0 & \text{se } y_i = V_i \wedge g_i > 0 \\ 0 & \text{se } y_i = 0 \wedge g_i < 0 \\ g_i & \text{altrimenti.} \end{cases} \quad (1.19)$$

In [12] viene dimostrata la convergenza di entrambi i metodi (*conditional subgradient* e *subgradient projection*) sotto ipotesi comuni per la scelta della lunghezza del passo  $\nu$ .

## 1.7 Generazione di soluzioni primali

L'algoritmo del volume è stato sviluppato con l'intento di fornire un metodo valido per generare soluzioni primali approssimate (cfr. sez. 1.1). Nel caso Lagrangiano quindi si calcola per ogni iterazione la soluzione primale corrispondente e si tiene traccia della media di tutte le soluzioni primali generate utilizzando come pesi i coefficienti convessi  $\alpha$ ,  $(1 - \alpha)$ ; vale a dire:

$$\bar{x} = \alpha x + (1 - \alpha)\bar{x}, \quad (1.20)$$

dove  $x$  è la soluzione primale generata all'iterazione corrente.

Barahona in [4] mostra come la soluzione primale aggregata approssimi asintoticamente una soluzione primale ottima, tuttavia non fornisce dimostrazioni teoriche di questo comportamento. Un metodo del volume convergente dal punto di vista primale, conosciuto come RVA, è presentato in [7] e discusso tra gli altri nel capitolo 3.

Ad ogni modo si può verificare computazionalmente che utilizzando una combinazione convessa delle soluzioni primali, nei metodi che deflettono utilizzando un’analoga combinazione convessa dei subgradienti, si produce una soluzione primale migliore di quella non aggregata all’ultima iterazione (cioè quella fornita dai metodi del subgradiente ordinario). In particolare qualche vantaggio da un punto di vista primale sembra essere offerto anche dai metodi del subgradiente deflesso considerati nella sezione 1.5.

Un metodo alternativo per generare soluzioni primali è descritto da Larsson et al. in [25] dove si utilizza il cosiddetto *metodo delle serie ergodiche*. Si considerano le seguenti quantità:

$$\nu_{sum}^k = \sum_{i=0}^k \nu_i, \quad \bar{x}_k = \sum_{i=0}^k \frac{\nu_i}{\nu_{sum}^k} x_i. \quad (1.21)$$

[25] dimostra la convergenza della soluzione così costruita ad una soluzione primale ottima per il metodo del subgradiente condizionale (cfr. sez. 1.6) e quindi anche per il metodo del subgradiente ordinario.

## 1.8 Condizioni di terminazione

Uno dei punti più deboli degli algoritmi del subgradiente è la mancanza di condizioni di terminazione efficienti. In questa sezione ne esaminiamo tre tipi; il nostro algoritmo terminerà se vale una di queste tre condizioni:

$$f(y) - f^* \leq \psi_1 \max\{|\bar{f}(y)|, \psi_{min}\} \quad (1.22)$$

$$\|g^Y\| \leq \psi_2 \max\{|\bar{f}(y)|, \psi_{min}\} \quad (1.23)$$

$$\varepsilon + t\|g^Y\|^2 \leq \psi_3 \max\{|\bar{f}(y)|, \psi_{min}\} \quad (1.24)$$

dove le tolleranze a destra delle disequazioni sono relative; in questo modo esse vengono rese omogenee sui diversi problemi in esame.

La condizione (1.22) è l’unica condizione efficiente qualunque sia la funzione da minimizzare; purtroppo richiede la conoscenza del valore ottimo della funzione e pertanto è raramente applicabile. Si noti che, volendo risolvere il problema all’ottimo, si dovrebbe avere  $\psi_1 = 0$ , ma, poiché la convergenza verso l’ottimo è soltanto asintotica,  $\psi_1 = 0$  non garantirebbe la terminazione finita dell’algoritmo; pertanto si limita lo scarto dal valore ottimo in funzione obiettivo.

Un’altra condizione di ottimalità necessaria e sufficiente è

$$0 \in \partial^Y f(y).$$

Infatti, applicando la proprietà di subgradiente si ha  $f^* - f(y) \geq 0(y^* - y) = 0$  e  $f^* - f(y) \leq 0$  per la minimalità di  $f^*$ .

Purtroppo anche questa condizione è di difficile implementazione: tipicamente non si dispone di un metodo per “esplorare” il subdifferenziale di  $f$  in un punto. Comunque questa proprietà suggerisce di utilizzare la condizione  $\|g\| = 0$ , ancora sufficiente ma non più necessaria. Possiamo, cioè, trovarci nell’ottimo ma “non accorgercene” perché abbiamo calcolato un subgradiente non nullo. Inoltre anche in questo caso la convergenza verso l’ottimo sarebbe soltanto asintotica e si introduce un errore di accuratezza  $\psi_2$ , ottenendo la condizione (1.23). Se  $g^Y$

è un subgradiente condizionale questa condizione si rivela in un certo modo più utile: se l'algoritmo individua una soluzione ottima sulla frontiera dell'insieme ammissibile e calcoliamo un subgradiente  $g \in -N_Y(y_k)$ , allora l'operazione di proiezione restituisce un subgradiente nullo e l'algoritmo termina. Si può pensare quindi che nonostante il subdifferenziale di  $f$  sia di difficile "esplorazione" quello della funzione indicatrice è immediato (e coincide con  $N_Y(y)$ ). Può essere interessante notare come la condizione di terminazione del caso lineare sia un caso particolare di questa; se la regione è poliedrale il *normal cone* è il cono generato dai vincoli attivi e il vettore dei costi  $c$  è il gradiente (e quindi l'unico subgradiente) costante della funzione obiettivo. Dunque esiste un subgradiente condizionale nullo se e solo se  $c \in \text{Cono}(A_{i_1}, A_{i_2}, \dots, A_{i_m})$   $I = \{i_1, i_2, \dots, i_m\}$ .

La condizione (1.23), però, non fornisce un metodo efficiente per terminare l'algoritmo. Se l'ottimo fornito dall'algoritmo è un punto in cui la funzione  $f$  è differenziabile, il subgradiente è unico (ed è il gradiente) e, pertanto, la condizione "funziona bene". Sfortunatamente spesso la funzione è ottima nei punti di non differenziabilità e quindi l'algoritmo potrebbe continuare ad iterare senza accorgersi di aver già risolto il problema. Inoltre questa condizione non garantisce alcuna proprietà sul valore obiettivo generato dall'algoritmo. Se arrestiamo l'algoritmo quando troviamo un subgradiente nullo sappiamo per certo che la soluzione generata è ottima, ma se limitiamo il subgradiente in norma potrebbe non esserci modo di limitare il valore della funzione obiettivo, o potrebbe essere difficile.

Se usiamo un metodo approssimato la condizione  $0 \in \partial_\varepsilon^Y f(y)$  garantisce soltanto la  $\varepsilon$ -ottimalità e, allo stesso modo, tutte le condizioni che da essa derivano approssimano soltanto un test di  $\varepsilon$ -ottimalità. Questo non è vero con la condizione (1.22) che continua a funzionare bene anche con il metodo approssimato.

Vogliamo costruire una condizione di terminazione che sia in qualche modo "correlata" al valore obiettivo della soluzione corrente. Dalla proprietà dei subgradienti, se  $g^Y \in \partial_\varepsilon^Y f(y)$ , abbiamo:  $f(y) - f(y^*) \leq (y - y^*)g^Y + \varepsilon$ . Pertanto la condizione

$$\varepsilon + (y - y^*)g^Y \leq \psi \quad (1.25)$$

garantirebbe  $f(y) - f(y^*) \leq \psi$ , ma conoscere  $y^*$  renderebbe inutile l'applicazione dell'algoritmo. Scriviamo quindi la condizione come:

$$t^*(y, g^Y) = \frac{\|y - y^*\|}{\|g^Y\|}, \quad \varepsilon + t^*(y, g^Y)\|g^Y\|^2 \leq \psi. \quad (1.26)$$

Dove (1.26)  $\Rightarrow$  (1.25), infatti  $|\varepsilon + (y - y^*)g^Y| \leq \varepsilon + \|y - y^*\|\|g^Y\| = \varepsilon + t^*\|g^Y\|^2 \leq \psi$ . Ovviamente neanche questa condizione è implementabile ma suggerisce l'approssimazione (1.24): se accade  $t \geq t^*$ , allora  $\varepsilon + t^*\|g^Y\|^2 \leq \varepsilon + t\|g^Y\|^2 \leq \psi$  e la condizione garantisce l' $\varepsilon$ -ottimalità. Scegliendo un parametro  $t$  troppo grande si rischia di avere una condizione poco efficiente, dunque  $t$  andrebbe modulato attentamente sul problema in esame.

Si noti che anche (1.24) soffre dello stesso problema di (1.23) dovuto all'incapacità di "esplorare" il subdifferenziale: non è detto che in un punto di ottimo (1.24) fornisca un riscontro positivo. Dunque nemmeno la condizione (1.24) può garantire la terminazione finita dell'algoritmo in condizioni di ottimalità. Si rende quindi necessario implementare alcune condizioni "brutali"; tipicamente si

limita il numero di iterazioni o il tempo impiegato dall'algoritmo. Ovviamente terminando in questo modo l'algoritmo non si ha alcun tipo di garanzia sul valore obiettivo generato.

A volte è possibile calcolare, durante l'esecuzione, un *lower bound* del valore ottimo obiettivo; ad esempio nel caso lagrangiano quando si dispone di un metodo efficiente per calcolare una soluzione primale ammissibile a partire da quella non ammissibile generata dall'algoritmo. In questo caso si può utilizzare la seguente condizione di terminazione:

$$\bar{f}(y) - LB \leq \psi_4 |LB| \quad (1.27)$$

Questa condizione è molto efficiente ma non sempre applicabile. Una condizione efficiente nella pratica può essere ottenuta terminando l'algoritmo quando il miglioramento dei valori obiettivi è quasi nullo; vale a dire:

$$\bar{f}_{k-\Delta} - \bar{f}_k < \psi_5 |\bar{f}_k|, \quad (1.28)$$

dove gli indici indicano le iterazioni e  $\Delta$  è un parametro. Un rallentamento dell'algoritmo, e quindi la terminazione secondo questo criterio, potrebbe indicare la vicinanza del valore ottimo, tuttavia è sempre possibile che l'algoritmo rallenti per altri motivi; in altre parole questo criterio sembra essere efficiente ma non garantisce proprietà sul valore obiettivo alla terminazione.

## 1.9 Metodi del subgradiente incrementale

Quando la funzione da minimizzare è composta da più funzioni convesse, vale a dire:

$$f(y) = \sum_{i=1}^m f_i(y),$$

si può ottenere un sostanziale miglioramento operando separatamente su ogni singola funzione. I metodi incrementali, infatti, migliorano l'efficienza dell'algoritmo evitando di calcolare la funzione totale (che generalmente è un'operazione computazionalmente costosa) e calcolando una migliore direzione di spostamento dividendo una singola iterazione in  $m$  iterazioni incrementali.

Ad ogni iterazione dei metodi del subgradiente ordinario si fa corrispondere un ciclo di iterazioni incrementali, in ognuna delle quali si considera il subgradiente di una sola delle componenti. Pertanto il metodo del subgradiente incrementale può essere descritto come segue:

$$\begin{aligned} y_{0,k} &= y_k \\ y_{i+1,k} &= P_Y(y_{i,k} - \nu_k g_{i,k}), \quad \text{con } g_{i,k} \in \partial f_i(y_{i,k}), \quad i = 1, \dots, m \\ y_{k+1} &= y_{m,k} \end{aligned}$$

Il metodo del subgradiente incrementale può essere visto come un metodo approssimato del subgradiente ordinario. Vale, infatti, la seguente proprietà (cfr. [11]):

**Teorema 3**

$$\widehat{g}_k = \sum_{i=1}^m g_{i,k} \in \partial_{\varepsilon_k} f(y_k)$$

$$\text{con } \varepsilon_k = \sum_{i=0}^m (\widetilde{g}_{i,k} - g_{i,k})(y_k - y_{i,k}), \quad \text{dove } \widetilde{g}_{i,k} \in \partial f_i(y_k) \quad i = 1, \dots, m$$

*Dimostrazione:*

$$\begin{aligned} \widehat{g}_k(y - y_k) &= \\ &= \sum_{i=1}^m g_{i,k}(y - y_{i,k}) + \sum_{i=1}^m g_{i,k}(y_{i,k} - y_k) \leq \\ &= \sum_{i=1}^m (f_i(y) - f_i(y_{i,k})) + \sum_{i=1}^m g_{i,k}(y_{i,k} - y_k) = \text{perché } g_{i,k} \in \partial f_i(y_{i,k}) \\ f(y) - f(y_k) + \sum_{i=1}^m (f_i(y_k) - f_i(y_{i,k}) + g_{i,k}(y_{i,k} - y_k)) &\leq \\ f(y) - f(y_k) + \sum_{i=1}^m (\widetilde{g}_{i,k}(y_k - y_{i,k}) + g_{i,k}(y_{i,k} - y_k)) &= \\ f(y) - f(y_k) + \sum_{i=1}^m ((\widetilde{g}_{i,k} - g_{i,k})(y_k - y_{i,k})) &. \end{aligned}$$

Questo teorema fornisce, quindi, una stima dell'errore sul subgradiente; è analogamente disponibile una stima dell'errore sulla funzione (cfr. [1]):

$$\widehat{f}_k = \sum_{i=0}^m f_i(y_{i,k})$$

$$f(y_k) - \widehat{f}_k \leq \sum_{i=0}^m \nu_k \|\widetilde{g}_{i,k}\| \|g_{i,k}\|. \quad (1.29)$$

Dunque, se i subgradienti sono limitati in norma e  $\nu_k \rightarrow 0$  (che è un'ipotesi molto comune),  $f(y_k) - \widehat{f}_k \rightarrow 0$ ; cioè  $\widehat{f}_k$  è una buona approssimazione di  $f(y_k)$ . Queste considerazioni suggeriscono l'opportunità di evitare il calcolo della funzione al termine dei cicli incrementali e utilizzare l'approssimazione  $\widehat{f}_k$  in luogo di  $f(y_k)$ . In questo modo possiamo approssimare molte condizioni di aggiornamento e regole per lo stepsize; comunque non tutti i metodi convergono quando si utilizzano queste approssimazioni. Lo studio delle approssimazioni utilizzabili e delle relative proprietà di convergenza è un aspetto interessante che meriterebbe di essere studiato più approfonditamente.

In [11] vengono proposte tre classi di regole per la lunghezza del passo (*costante, diminishing, target value*) e per ognuna vengono studiate le proprietà di convergenza. Inoltre viene introdotto il metodo del subgradiente incrementale randomizzato che non assume un ordinamento fisso nella valutazione della funzione.

**Metodi del subgradiente incrementale randomizzati** Negli algoritmi del subgradiente incrementale la scelta dell'ordinamento delle funzioni è arbitraria. Esiste, quindi, per ogni iterazione “l'ordinamento migliore”, quello che, ad esempio, massimizza  $|f(y_{k+1}) - f(y_k)|$ ; tuttavia non siamo in grado di calcolarlo. Queste considerazioni suggeriscono che randomizzando l'ordine in cui vengono considerate le funzioni che compongono la funzione obiettivo si ottengano risultati migliori e, come mostra [11], i risultati computazionali supportano questa ipotesi.

Abbiamo a disposizione due principali tecniche di randomizzazione: (a) riordinare casualmente le funzioni ad ogni ciclo incrementale e (b) ad ogni iterazione scegliere casualmente (con distribuzione uniforme) una funzione  $f_i$ . La seconda tecnica che può essere descritta come segue:

$$y_{k+1} = P_Y(y_k - \nu_k g(\omega_k, y_k))$$

dove  $\omega_k \in \{1, \dots, m\}$  è una variabile aleatoria con distribuzione uniforme e  $g(\omega_k, y_k) \in \partial f_{\omega_k}(y_k)$ . Questo metodo è un caso particolare dei cosiddetti *stochastic subgradient methods*.

## Capitolo 2

# Il metodo condizionale deflesso

La trattazione del metodo condizionale alla sezione 1.6 si limita al caso in cui non siano previste strategie di *direction deflection*; analogamente la trattazione del metodo deflesso in 1.5 non prende in considerazione la possibilità di utilizzare subgradienti condizionali. Questa è grossomodo anche la situazione in letteratura.

Presenteremo, quindi, in questo capitolo un metodo innovativo che integra i metodi condizionale e deflesso in due varianti. Proponiamo inoltre diverse dimostrazioni di convergenza (sotto condizioni differenti). Poiché la trattazione di questo metodo è approfondita le dedichiamo un capitolo a parte.

Siamo interessati ad applicare il metodo condizionale sia in presenza di *direction deflection* che nei metodi del volume. A prima vista sembrerebbe opportuno applicare una strategia di *deflection* ai subgradienti condizionali generati nelle passate iterazioni. Vale a dire

$$d := \alpha g + (1 - \alpha)d, \quad g \in \partial^Y f(y). \quad (2.1)$$

Tuttavia il metodo così generato rischierebbe di non comportarsi in maniera efficiente sulla frontiera dell'insieme ammissibile, infatti non garantisce che la direzione generata sia tangente all'insieme ammissibile (ovvero che appartenga a  $T_Y(y)$ ). In [14] viene proposto un metodo che adotta una strategia di *deflection* nei punti interni dell'insieme ammissibile ed il metodo condizionale nei punti di frontiera; vale a dire:

$$\text{Sia } g \in \partial f(y) \quad d := \begin{cases} P_{T_Y(y)}(g) & \text{se } y \in bd(Y) \text{ e } g \notin T_Y(y) \\ (\alpha)g + (1 - \alpha)d & \text{altrimenti} \end{cases} \quad (2.2)$$

dove  $bd(Y)$  indica la frontiera dell'insieme  $Y$ .

Altre possibili soluzioni si ottengono proiettando ad ogni iterazione la direzione aggregata sul cono  $T_Y(y)$ . In questo modo otteniamo due metodi differenti se scegliamo di aggregare le direzioni proiettate o quelle non proiettate:

$$g \in \partial f(y) \quad \hat{d} := \alpha g + (1 - \alpha)d \quad d := P_{T_Y(y)}(\hat{d}). \quad (2.3)$$

$$g \in \partial f(y) \quad \hat{d} := \alpha g + (1 - \alpha)\hat{d} \quad d := P_{T_Y(y)}(\hat{d}). \quad (2.4)$$

Nel seguito analizzeremo la convergenza di queste ultime due soluzioni.

Tipicamente in letteratura il metodo deflesso viene considerato utilizzando stepsize di Polyak (cfr. [18]) o Target Value (cfr. [6]). In questo capitolo estendiamo le analisi di convergenza per questi tipi di stepsize e proponiamo un'analisi di convergenza per il metodo deflesso che utilizza stepsize di tipo square summable/diminishing.

I metodi deflessi possono essere interpretati come metodi del subgradiente approssimato (lemma 8). Per questo motivo si potrebbe pensare di avere già a disposizione un'analisi di convergenza e di potersi ricondurre ad essa. Tuttavia queste analisi presuppongono delle condizioni sull'errore di non facile dimostrazione in questo caso; di seguito, quindi, sarà presentata una teoria della convergenza studiata ad hoc per i metodi del subgradiente condizionale deflesso mettendo in evidenza il loro comportamento come metodi del subgradiente approssimato. In altre parole saremo in grado, a posteriori, di sapere come si comporta asintoticamente l'approssimazione della direzione deflessa.

Le proprietà di convergenza del metodo del subgradiente ordinario (cfr. sez. 1.2) si fondano sulle proprietà dei subgradienti (quindi in sostanza sulla convessità della funzione  $f$ ) e su proprietà degli stepsize. Quando si introduce la deflessione, però, alcune delle proprietà dei subgradienti vengono perse (ovviamente la direzione non è più un subgradiente esatto). In altre parole il metodo deflesso introduce delle complesse relazioni tra il parametro di deflessione (quanto deflettiamo) e la scelta dello stepsize (di quanto ci spostiamo) che condizionano la convergenza dell'algoritmo. Cercheremo, ora, di indagare queste relazioni indicando quali metodi permettono di scegliere liberamente la modalità di deflessione vincolando lo stepsize e viceversa.

Il metodo del subgradiente ordinario garantisce alcune proprietà fondamentali per dimostrare la convergenza e le proprietà dell'algoritmo. Cioè se  $d_k = g_k \in \partial f(y_k)$ :

$$(-d_k)(y_k - y^*) = d_k(y^* - y_k) \leq f(y^*) - f(y_k) \quad (2.5)$$

$$(-d_k)(y^* - y_k) \geq 0 \quad (2.6)$$

$$\|y_k - y^*\| \leq \|y_{k-1} - y^*\| \quad (2.7)$$

dove (2.5) viene da (1) e (2.6) segue da (2.5) per la minimalità di  $f(y^*)$ .

Di queste proprietà, quando si utilizza una direzione diversa da  $g_k$ , soltanto (2.6) è sempre necessaria per dimostrare la convergenza; (2.5) permette di derivare (2.7).

Quando si sostituisce  $g_k$  con (1.7) queste proprietà non dipendono più solamente dal subgradiente ma dallo stepsize  $\nu$  e dal coefficiente  $\alpha$ .

La figura 2.1 mostra le relazioni tra lo stepsize  $\nu$ , il parametro  $\alpha$  e le proprietà del vettore  $d$ . La proprietà (2.6) afferma che ogni punto di ottimalità appartiene al semispazio affine individuato da  $-d_k$ , cioè  $Y^* \subseteq \{y \in \mathbb{R}^n : (-d_k)(y - y_k) \geq 0\}$ .

(1.7) e

$$d_{k-1}(y^* - y_k) \leq 0 \quad (2.8)$$

implicano (2.5). In particolare (2.8) afferma che la direzione precedente individua comunque, nella soluzione corrente, un semispazio contenente l'insieme ottimo.

Dunque, nell'esempio in figura,  $y_{k+1}^{(a)} - y_k$  è il massimo spostamento che garantisce (2.8) e dunque (2.5), si rispettano questi limiti se si utilizzano opportuni stepsize, ad esempio (2.26).

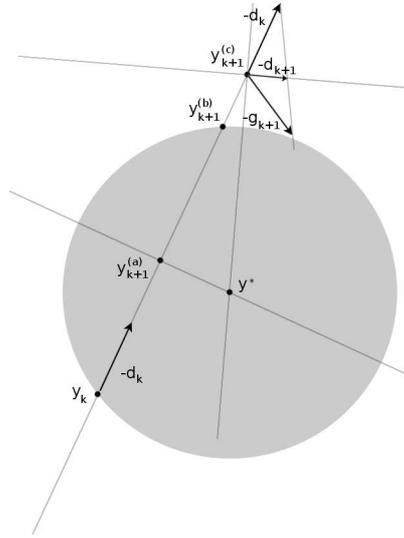


Figura 2.1: Relazioni tra lo stepsize  $\nu$ , il parametro  $\alpha$  e la proprietà (2.6) nel metodo (1.7).

(2.5) serve per dimostrare (2.7) ed è garantita da (2.8), ma in realtà il massimo spostamento che garantisce (2.7) è  $y_{k+1}^{(b)} - y_k$ . Se non imponiamo limitazione alcuna allo stepsize  $\nu$  (e otteniamo, ad esempio, la soluzione  $y_{k+1}^{(c)}$ ), possiamo ancora far valere (2.6), imponendo una limitazione inferiore al parametro  $\alpha$ . Di fatto limitiamo la capacità dell'algoritmo di deflettere quanto basta per far sì che una soluzione ottima sia nel semispazio verso cui ci si muove. La condizione sufficiente è:

$$\nu_{k-1} \frac{\|d_{k-1}\|^2}{(f(y_k) - f(y^*)) + \nu_{k-1}^2 \|d_{k-1}\|^2} \leq \alpha_k \leq 1. \quad (2.9)$$

Nel seguito presentiamo diversi metodi ognuno dei quali garantisce una di queste proprietà e per essi dimostriamo la convergenza. Il metodo 1 garantisce (2.5) imponendo la scelta dello stepsize: questo metodo, però, è di difficile implementazione in quanto richiede la conoscenza del valore ottimo  $f^*$ . Il metodo 5 è, o meglio può essere visto come, un'approssimazione implementabile del metodo 1. Il metodo 3, invece, non pone limitazioni sullo stepsize (eccetto delle comuni condizioni di convergenza) ma limita la possibilità di deflettere; questo metodo presenta lo stesso problema del metodo 1 e, analogamente ad esso, può essere approssimato utilizzando il metodo 4. Il metodo 2 è simile al metodo 1 ma presenta condizioni di convergenza differenti, è di scarsa utilità pratica ma viene utilizzato nel dimostrare la convergenza del metodo 5.

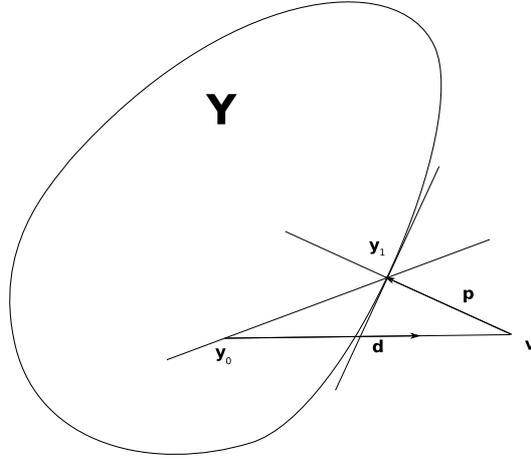


Figura 2.2: Lemma 4 in  $\mathbb{R}^2$

## 2.1 Analisi di convergenza

### 2.1.1 Lemmi Tecnici

**Lemma 4** Sia  $Y \subseteq \mathbb{R}^n$  un insieme chiuso e convesso,  $y_0 \in Y$  e  $v = y_0 + \lambda d$ , con  $d \in \mathbb{R}^n$  e  $\lambda > 0$ . Se  $y_1 = P_Y(v)$  e  $p = y_1 - v$ , allora

$$pd \leq 0 \quad (2.10)$$

$$\|y_1 - y_0\| \leq \|v - y_0\| \quad (2.11)$$

$$(y_0 - y_1)(v - y_1) \leq 0 \quad (2.12)$$

*Dimostrazione:* Per (2.10) e (2.11) si veda lemma 3.9 in [14]; per (2.12) si veda 2.2.1 in [16].

La figura 2.2 fornisce un esempio del lemma.

**Lemma 5** Sia

$$\hat{y}_{k+1} = y_k - \nu_k d_k, \quad y_{k+1} = P_Y(\hat{y}_{k+1}) \quad (2.13)$$

il metodo utilizzato, dove  $\nu_k$  e  $d_k$  sono calcolati secondo uno qualunque dei metodi possibili; allora:

$$\forall k \quad d_k(y_k - y_{k+1}) \leq \nu_k \|d_k\|^2 \quad (2.14)$$

*Dimostrazione:*

$$\begin{aligned} d_k(y_k - y_{k+1}) &= \\ d_k(y_k - \hat{y}_{k+1}) + d_k(\hat{y}_{k+1} - y_{k+1}) &= \\ \nu_k \|d_k\|^2 + d_k(\hat{y}_{k+1} - y_{k+1}) &\leq \nu_k \|d_k\|^2 \quad \text{per (2.10)}. \end{aligned}$$

**Lemma 6** Sia (2.13) il metodo utilizzato, dove  $\nu_k$  e  $\widehat{d}_k$  sono calcolati secondo uno dei modi possibili e  $d_k = -P_{T_Y(y_k)}(-\widehat{d}_k)$ ; allora:

$$\forall k \quad \widehat{d}_k(y_k - y_{k+1}) \leq \nu_k \|d_k\|^2 \quad (2.15)$$

*Dimostrazione:*

$$\begin{aligned} \widehat{d}_k(y_k - y_{k+1}) &= \\ d_k(y_k - y_{k+1}) - P_{N_Y(y_k)}(-\widehat{d}_k)(y_k - y_{k+1}) &\leq \text{perché } \widehat{d}_k = d_k - P_{N_Y(y_k)}(-\widehat{d}_k) \\ \nu_k \|d_k\|^2 + P_{N_Y(y_k)}(-\widehat{d}_k)(y_{k+1} - y_k) &\leq \text{per (2.14)} \\ \nu_k \|d_k\|^2 &\text{ perché } (y_{k+1} - y_k) \in T_Y(y_k). \end{aligned}$$

**Lemma 7** Sia  $\widehat{d}_k$  una direzione calcolata secondo uno dei metodi possibili, sia  $d_k = -P_{T_Y(y_k)}(-\widehat{d}_k)$ , sia  $y \in T_Y(y_k)$ , allora:

$$d_k y \leq \widehat{d}_k y. \quad (2.16)$$

*Dimostrazione:*

$$-\widehat{d}_k = P_{T_Y(y_k)}(-\widehat{d}_k) + P_{N_Y(y_k)}(-\widehat{d}_k) = (-d_k) + P_{N_Y(y_k)}(-\widehat{d}_k).$$

$$d_k y = (\widehat{d}_k + P_{N_Y(y_k)}(-\widehat{d}_k))y = \widehat{d}_k y + P_{N_Y(y_k)}(-\widehat{d}_k)y \leq \widehat{d}_k y.$$

Dove  $P_{N_Y(y_k)}(-\widehat{d}_k)y \leq 0$  poiché  $P_{N_Y(y_k)}(-\widehat{d}_k) \in N_Y(y_k)$  e  $y \in T_Y(y_k)$ .

**Lemma 8** Sia (2.13),

$$\widehat{d}_{k+1} = \alpha_{k+1} g_{k+1} + (1 - \alpha_{k+1}) d_k, \quad g_k \in \partial^Y f(y_k), \alpha_k \in (0, 1], \alpha_0 = 1 \quad (2.17)$$

$$d_{k+1} = -P_{T_Y(y_{k+1})}(-\widehat{d}_{k+1}), \quad (2.18)$$

Il metodo utilizzato, allora:

$$\forall k : \quad d_k \in \partial_{\varepsilon_k}^Y f(y_k);$$

con:

$$\varepsilon_k = \begin{cases} 0 & \text{se } k = 0. \\ (1 - \alpha_k)(f(y_k) - f(y_{k-1}) - d_{k-1}(y_k - y_{k-1}) + \varepsilon_{k-1}) & \text{altrimenti} \end{cases} \quad (2.19)$$

*Dimostrazione:* Per induzione  $d_0 = -P_{T(y_0)}(-g_0) \in \partial^Y f(y_0)$ .

$\forall y \in Y :$

$$\begin{aligned} d_k(y - y_k) &\leq \widehat{d}_k(y - y_k) = && \text{per (2.16) utilizzando} \\ & && (y - y_k) \in T_Y(y_k) \\ & \alpha_k g_k(y - y_k) + (1 - \alpha_k) d_{k-1}(y - y_k) = && \text{per (2.17)} \\ \alpha_k (f(y) - f(y_k)) + (1 - \alpha_k) (d_{k-1}(y - y_{k-1}) - d_{k-1}(y_k - y_{k-1})) &\leq \\ \alpha_k (f(y) - f(y_k)) + (1 - \alpha_k) (f(y) - f(y_{k-1}) + \varepsilon_{k-1} - d_{k-1}(y_k - y_{k-1})) &= && \text{per ip.ind. e } g_k \in \partial^Y f(y_k) \\ f(y) - f(y_k) + (1 - \alpha_k) (f(y_k) - f(y_{k-1}) - d_{k-1}(y_k - y_{k-1}) + \varepsilon_{k-1}). \end{aligned}$$

**Lemma 9** Sia (2.13), (2.18),

$$\widehat{d}_{k+1} = \alpha_{k+1}g_{k+1} + (1 - \alpha_{k+1})\widehat{d}_k, \quad g_k \in \partial^Y f(y_k), \alpha_k \in (0, 1], \alpha_0 = 1 \quad (2.20)$$

Il metodo utilizzato, allora:

$$\forall k : \widehat{d}_k \in \partial_{\varepsilon_k}^Y f(y_k);$$

con:

$$\varepsilon_k = \begin{cases} 0 & \text{se } k = 0. \\ (1 - \alpha_k)(f(y_k) - f(y_{k-1}) - \widehat{d}_{k-1}(y_k - y_{k-1}) + \varepsilon_{k-1}) & \text{altrimenti} \end{cases} \quad (2.21)$$

*Dimostrazione:* Per induzione  $\widehat{d}_0 = g_0 \in \partial^Y f(y_0)$ .

$\forall y \in Y :$

$$\begin{aligned} \widehat{d}_k(y - y_k) &= \\ \alpha_k g_k(y - y_k) + (1 - \alpha_k)\widehat{d}_{k-1}(y - y_k) &= \quad \text{per (2.20)} \\ \alpha_k g_k(y - y_k) + (1 - \alpha_k)(\widehat{d}_{k-1}(y - y_{k-1}) - \widehat{d}_{k-1}(y_k - y_{k-1})) &\leq \\ \alpha_k(f(y) - f(y_k)) + (1 - \alpha_k)(f(y) - f(y_{k-1}) + \varepsilon_{k-1} - \widehat{d}_{k-1}(y_k - y_{k-1})) &= \quad \text{per ip.ind. e } g_k \in \partial^Y f(y_k) \\ f(y) - f(y_k) + (1 - \alpha_k)(f(y_k) - f(y_{k-1}) - \widehat{d}_{k-1}(y_k - y_{k-1}) + \varepsilon_{k-1}). & \end{aligned}$$

**Lemma 10** Per il metodo (2.13), (2.17), (2.18); se vale:

$$\forall k \quad \nu_k \leq \alpha_k \frac{f(y_k) - f^*}{\|d_k\|^2}; \quad (2.22)$$

allora:

$$\varepsilon_k \leq (1 - \alpha_k)(f(y_k) - f^*). \quad (2.23)$$

*Dimostrazione:* Per induzione su  $k$ :  $\varepsilon_0 = 0 = (1 - \alpha_0)(f(y_0) - f^*)$ .

$$\begin{aligned} \varepsilon_k &= (1 - \alpha_k)(f(y_k) - f(y_{k-1}) - d_{k-1}(y_k - y_{k-1}) + \varepsilon_{k-1}) \leq \\ &\quad (1 - \alpha_k)(f(y_k) - f(y_{k-1}) + \nu_{k-1}\|d_{k-1}\|^2 + \varepsilon_{k-1}) \leq \quad \text{per (2.14)} \\ &\quad (1 - \alpha_k)(f(y_k) - f(y_{k-1}) + \alpha_{k-1}(f(y_{k-1}) - f^*) + \varepsilon_{k-1}) \leq \quad \text{per (2.22)} \\ (1 - \alpha_k)(f(y_k) - f(y_{k-1}) + \alpha_{k-1}(f(y_{k-1}) - f^*) + (1 - \alpha_{k-1})(f(y_{k-1}) - f^*)) &= \quad \text{per ip. ind.} \\ &\quad (1 - \alpha_k)(f(y_k) - f^*). \end{aligned}$$

**Lemma 11** Per il metodo (2.13), (2.20), (2.18); se vale (2.22); allora vale (2.23).

*Dimostrazione:* Analoga al lemma 10 utilizzando (2.15) anziché (2.14).

**Corollario 12** Per il metodo (2.13), (2.17), (2.18), (2.22), vale:

$$y^* \in Y^* \Rightarrow d_k(y^* - y_k) \leq \alpha_k(f^* - f(y_k)). \quad (2.24)$$

*Dimostrazione:* Corollario del lemma 10.

$$\begin{aligned} d_k(y^* - y_k) &\leq f^* - f(y_k) + \varepsilon_k \leq \\ f^* - f(y_k) + (1 - \alpha_k)(f(y_k) - f^*) &= \quad \text{per (2.23)} \\ \alpha_k(f^* - f(y_k)). & \end{aligned}$$

**Osservazione** Il corollario vale anche per il metodo (2.13), (2.20), (2.18), (2.22) se utilizziamo nella dimostrazione il lemma 11 anzichè il lemma 10.

Abbiamo visto come utilizzando (2.20), continuano a valere (2.23) e (2.24); poiché le dimostrazioni di convergenza si basano sostanzialmente su queste proprietà possono essere facilmente estese al metodo (2.20). In seguito, comunque, sottolineeremo come modificare le dimostrazione per adattarle a questo caso.

**Lemma 13** *Utilizzando il metodo (2.13); sia  $y \in Y$ , si ha:*

$$\|y_{k+1} - y\|^2 \leq \|y_k - y\|^2 - 2\nu_k(y_k - y)d_k + \nu_k^2\|d_k\|^2. \quad (2.25)$$

*Dimostrazione:*

$$\begin{aligned} \|y_{k+1} - y\|^2 &\leq \\ \|\widehat{y}_{k+1} - y\|^2 &= \quad \text{teorema 2.2.1 in [16]} \\ \|y_k - \nu_k d_k - y\|^2 &= \\ \|y_k - y\|^2 - 2\nu_k(y_k - y)d_k + \nu_k^2\|d_k\|^2 & \quad \text{svolgendo il quadrato} \end{aligned}$$

## 2.1.2 Polyak stepsize

**Metodo 1** *Sia (2.13), (2.17), (2.18) con*

$$\nu_k = \beta_k \frac{f(y_k) - f^*}{\|d_k\|^2}, \quad 0 < \beta_k \leq \alpha_k \leq 1 \quad (2.26)$$

*il metodo iterativo utilizzato per ottimizzare  $f$ .*

Si noti come (2.26)  $\Rightarrow$  (2.22) e quindi valgono (2.23) e (2.24).

**Teorema 14** *Sotto le condizioni del Metodo 1, se  $c = \sup\{\|d_k\|\} < \infty$ ,  $\inf_k\{\beta_k\} > 0$  e  $Y^* \neq \emptyset$ , allora  $\lim_{k \rightarrow \infty} f(y_k) = f^*$ .*

*Dimostrazione:* Sia  $\varepsilon : \forall k \ 0 < \varepsilon \leq \beta_k \leq \alpha_k$ ; sia  $y^* \in Y^*$  fissato:

$$\begin{aligned} \|y_{k+1} - y^*\|^2 &\leq \\ \|y_k - y^*\|^2 - 2\nu_k(y_k - y^*)d_k + \nu_k^2\|d_k\|^2 &= \quad \text{per (2.25)} \\ \|y_k - y^*\|^2 - 2\beta_k \frac{f(y_k) - f(y^*)}{\|d_k\|^2} (y_k - y^*)d_k + \beta_k^2 \frac{(f(y_k) - f(y^*))^2}{\|d_k\|^2} &\leq \quad \text{esplicitando } \nu_k. \\ \|y_k - y^*\|^2 - 2\alpha_k \beta_k \frac{(f(y_k) - f(y^*))^2}{\|d_k\|^2} + \beta_k^2 \frac{(f(y_k) - f(y^*))^2}{\|d_k\|^2} &= \quad \text{per (2.24).} \\ \|y_k - y^*\|^2 - \beta_k(2\alpha_k - \beta_k) \frac{(f(y_k) - f(y^*))^2}{\|d_k\|^2} &\leq \\ \|y_k - y^*\|^2 - \frac{\varepsilon^2}{c^2} (f(y_k) - f(y^*))^2 &\leq \quad (2.27) \\ \|y_k - y^*\|^2 & \quad (2.28) \end{aligned}$$

Dove (2.27) vale perché  $2\alpha_k - \beta_k \geq \alpha_k \geq \varepsilon$ ,  $\beta_k \geq \varepsilon$  e  $\forall k \|d_k\| \leq c$ .

Da (2.27) abbiamo  $\|y_{k+1} - y^*\|^2 < \|y_k - y^*\|^2$ . Dunque  $\{\|y_k - y^*\|\}$  è monotona decrescente ed ha limite finito. In particolare  $\{y_k\}$  è limitata e, poiché  $Y$  è chiuso ed  $f$  è continua,  $\{f(y_k)\}$  è una successione a valori in un insieme compatto e, quindi, esistono sottosuccessioni convergenti. Supponiamo che non esistano sottosuccessioni  $f(y_{k_i}) - f^*$  convergenti a 0. Si avrebbe:

$$\exists \delta > 0 : \exists K : \forall k > K : 0 < \delta < f(y_k) - f(y^*).$$

Da (2.27) otteniamo:  $\|y_{k+1} - y^*\|^2 \leq \|y_k - y^*\|^2 - \xi$  dove  $\xi = \frac{\varepsilon^2 \delta^2}{c^2} > 0$  e, applicando ricorsivamente questa disuguaglianza abbiamo  $\|y_{k+1} - y^*\|^2 \leq \|y_K - y^*\|^2 - \xi(k - K) \rightarrow -\infty$  per  $k \rightarrow \infty$ : una contraddizione.

Pertanto esiste una sottosuccessione  $f(y_{k_i}) \rightarrow f(y^*)$ .

Ora  $\{y_k\}$  è una successione a valori in un compatto, quindi lo è anche  $\{y_{k_i}\}$  ed esiste una sottosuccessione  $\{y_{k_j}\} \subseteq \{y_{k_i}\}$  convergente ad un valore  $y^\infty \in Y$  ( $Y$  è chiuso).

Poiché  $f$  è continua  $f(y^\infty) = \lim_{j \rightarrow \infty} f(y_{k_j}) = \lim_{i \rightarrow \infty} f(y_{k_i}) = f^*$ . Dunque  $y^\infty \in Y^*$ .

Consideriamo, ora, la successione  $\|y_k - y^\infty\|$ . Da  $y^\infty \in Y^*$  e (2.28) sappiamo che è monotona decrescente ed ha limite finito. Dunque

$$\lim_{k \rightarrow \infty} \|y_k - y^\infty\| = \lim_{j \rightarrow \infty} \|y_{k_j} - y^\infty\| = 0.$$

E quindi  $\lim_{k \rightarrow \infty} (y_k - y^\infty) = 0 \Rightarrow \lim_{k \rightarrow \infty} y_k = y^\infty \in Y^*$ .

**Osservazione** Si può ottenere un metodo alternativo sostituendo (2.17) con (2.20). La dimostrazione di convergenza rimane inalterata.

**Remark** Il teorema 14 e (2.24) implicano  $\lim_{k \rightarrow \infty} \varepsilon_k = 0$ , dunque il metodo soddisfa le condizioni del teorema 7.17 in [1]. Inoltre tale teorema fornisce le seguenti condizioni di convergenza alternative a quelle del teorema 14:

$$\alpha_k \rightarrow 1, \beta_k \in (0, 2), \inf\{\beta_k\} > 0 \quad \text{in modo da avere } \varepsilon_k \rightarrow 0 \quad (2.29)$$

$$\{d_k\}, \{y_k\} \text{ e } \{f(y_k)\} \text{ limitati}$$

oppure:

$$\beta_k \in (0, 2), \inf\{\beta_k\} > 0, Y^* \neq \emptyset, \quad (2.30)$$

$$\exists \xi \in [0, 1) : \forall k \varepsilon_k \leq \frac{1}{2} \xi (2 - \beta_k) (f(y_k) - f^*),$$

Le condizioni (2.30) e (2.26) sembrano essere indipendenti, in particolare:

$$(0 < \zeta < \beta_k \leq \alpha_k \leq 1) \wedge (\exists \xi \in [0, 1) : \forall k \alpha_k \geq 1 - (1 - \frac{\beta_k}{2}) \xi) \Rightarrow (2.30),$$

Infatti:

$$\varepsilon_k \leq (1 - \alpha_k)(f(y_k) - f^*) \leq \text{lemma 10}$$

$$(1 - \frac{\beta_k}{2}) \xi (f(y_k) - f^*) =$$

$$\frac{1}{2} \xi (2 - \beta_k) (f(y_k) - f^*)$$

Le nostre condizioni coprono anche il caso in cui  $\lim_{k \rightarrow \infty} \alpha_k \neq 1$ , o il caso in cui tale limite non esista e pertanto sono più generali di (2.29). Inoltre, a differenza di (2.30) le nostre condizioni sono *implementabili*: ovvero forniamo un metodo per cui esse sono soddisfatte, mentre, senza un'ulteriore analisi, non si conosce un metodo non banale per cui (2.30) sia soddisfatta.

Le nostre condizioni sono simili a quelle richieste da [3], [6] e [24] per i metodi target value, tuttavia l'analisi di convergenza è diversa.

### 2.1.3 Diminishing stepsize

**Metodo 2** Sia (2.13), (2.17), (2.18) con le condizioni sullo stepsize (2.22) e:

$$\forall k \quad \nu_k > 0, \quad \lim_{k \rightarrow \infty} \nu_k = 0, \quad \sum_{k=0}^{\infty} \nu_k = \infty, \quad \sum_{k=0}^{\infty} \nu_k^2 < \infty. \quad (2.31)$$

Il metodo iterativo utilizzato per ottimizzare  $f$ .

**Teorema 15** Sotto le condizioni del metodo 2, se  $c = \sup\{\|d_k\|\} < \infty$ ,  $\inf_k\{\alpha_k\} > 0$  e  $Y^* \neq \emptyset$ , allora  $\lim_{k \rightarrow \infty} f(y_k) = f^*$ .

*Dimostrazione:*

$$\begin{aligned} \|y_{k+1} - y^*\|^2 &\leq \\ \|y_k - y^*\|^2 - 2\nu_k(y_k - y^*)d_k + \nu_k^2\|d_k\|^2 &\leq \quad \text{per (2.25)} \\ \|y_k - y^*\|^2 - \nu_k(2\alpha_k(f(y_k) - f(y^*)) - \nu_k\|d_k\|^2) &\leq \quad \text{per (2.24)} \\ \|y_k - y^*\|^2 &\leq \end{aligned} \quad (2.32)$$

Dove (2.32) viene da  $\nu_k > 0$  e:

$$\begin{aligned} 2\alpha_k(f(y_k) - f(y^*)) - \nu_k\|d_k\|^2 &\geq \\ 2\alpha_k(f(y_k) - f(y^*)) - \alpha_k(f(y_k) - f(y^*)) &\geq 0. \quad \text{per (2.22)}. \end{aligned}$$

Quindi  $\{\|y_k - y^*\|\}$  è non negativa monotona decrescente e quindi ha limite finito. In particolare  $\{y_k\}$  è limitata e, poiché  $Y$  è chiuso ed  $f$  è continua,  $\{f(y_k)\}$  è una successione a valori in un insieme compatto e, quindi, esistono sottosuccessioni convergenti. Supponiamo che non esistano sottosuccessioni  $f(y_{k_i}) - f(y^*)$  convergenti a 0. Si avrebbe:

$$\exists \varepsilon > 0 : \exists K : \forall k > K : \quad 0 < \varepsilon < f(y_k) - f(y^*).$$

E, quindi:

$$\begin{aligned} \|y_k - y^*\|^2 &\leq \\ \|y_0 - y^*\|^2 - 2 \sum_{j=0}^{k-1} \nu_j d_j (y_j - y^*) + \sum_{j=0}^{k-1} \nu_j^2 \|d_j\|^2 &\leq \\ \|y_0 - y^*\|^2 - 2 \sum_{j=0}^{k-1} \nu_j \alpha_j (f(y_j) - f(y^*)) + \sum_{j=0}^{k-1} \nu_j^2 \|d_j\|^2 &\leq \\ \|y_0 - y^*\|^2 - 2\xi \varepsilon \sum_{j=0}^{k-1} \nu_j + \sum_{j=0}^{k-1} \nu_j^2 \|d_j\|^2 &\rightarrow -\infty. \end{aligned}$$

Una contraddizione, pertanto esiste una sottosuccessione  $f(y_{k_i}) \rightarrow f(y^*)$  e, analogamente a quanto visto nel teorema 14, esiste una sottosuccessione  $\{y_{k_j}\} \subseteq \{y_{k_i}\}$  tale che:  $y_{k_j} \rightarrow y^\infty \in Y^*$ , e poiché  $\|y_k - y^\infty\|$  ha limite finito  $\lim_{k \rightarrow \infty} \|y_k - y^\infty\| = \lim_{k \rightarrow \infty} \|y_{k_j} - y^\infty\| = 0$ .

**Osservazione** Si può ottenere un metodo alternativo sostituendo (2.17) con (2.20). La dimostrazione di convergenza rimane inalterata.

**Osservazione** Si noti che si può sostituire  $\sum_{k=0}^{\infty} \nu_k^2 < \infty$  con  $\sum_{k=0}^{\infty} \nu_k^2 \|d_k\|^2 < \infty$ , mantenendo inalterata l'analisi di convergenza.

Inoltre questo non è un caso particolare del metodo 1 in quanto qui non vale l'ipotesi  $\inf \beta_k > 0$ , dove  $\beta_k = \frac{\nu_k \|d_k\|^2}{f(y_k) - f^*}$ . Le condizioni di convergenza, tuttavia, sono molto restrittive; in particolare non sappiamo se esiste una successione  $\nu_k$  che rispetti (2.31) e (2.22); rimane comunque utile questa analisi di convergenza perché la utilizzeremo nel caso target value (metodo 5).

**Remark** Ancora una volta sappiamo, a posteriori, che  $\lim_{k \rightarrow \infty} \varepsilon_k = 0$ . Analisi di convergenza alternative sono fornite da [1]; così come non possiamo applicare qui il teorema 14, non possiamo applicare il teorema 7.17 in [1] (discusso per il metodo 1). Perciò deriviamo una condizione di convergenza dal teorema 3.4 (in [1]), vale a dire:

$$\sum_{k=0}^{\infty} \nu_k \varepsilon_k < \infty. \quad (2.33)$$

In particolare il metodo 2 converge se eliminiamo la condizione  $\beta_k \leq \alpha_k$  e aggiungiamo (2.33). Per garantire che questa condizione sia soddisfatta dovremmo dimostrare  $\varepsilon_k = o(\nu_k)$ . Ad ogni modo, come già detto, non sappiamo se esista una successione che rispetti (2.22) e (2.31), pertanto qui le nostre condizioni non sono molto utili da un punto di vista implementativo.

**Metodo 3** Sia (2.13), (2.17), (2.18), (2.31), con la seguente condizione su  $\alpha_k$ :

$$\nu_{k-1} \frac{\|d_{k-1}\|^2}{(f(y_k) - f(y^*)) + \nu_{k-1} \|d_{k-1}\|^2} \leq \alpha_k \leq 1; \quad (2.34)$$

il metodo iterativo utilizzato per ottimizzare  $f$ .

Si noti che

$$\nu_{k-1} \frac{\|d_{k-1}\|^2}{(f(y_k) - f(y^*)) + \nu_{k-1} \|d_{k-1}\|^2} = \frac{1}{\frac{f(y_k) - f(y^*)}{\nu_{k-1} \|d_{k-1}\|^2} + 1} < 1.$$

Pertanto possiamo deflettere ad ogni iterazione.

**Lemma 16** Per il metodo (2.13), (2.17), (2.18), (2.34) vale:

$$\varepsilon_k \leq f(y_k) - f^*. \quad (2.35)$$

*Dimostrazione:* Per induzione su  $k$ .  $\varepsilon_0 = 0 \leq f(y_0) - f^*$ .

$$\begin{aligned}
\varepsilon_k &= (1 - \alpha_k)(f(y_k) - f(y_{k-1}) - d_{k-1}(y_k - y_{k-1}) + \varepsilon_{k-1}) \leq \\
(1 - \alpha_k)(f(y_k) - f(y_{k-1}) - d_{k-1}(y_k - y_{k-1}) + f(y_{k-1}) - f^*) &= \text{per ip. ind.} \\
&\quad (1 - \alpha_k)(f(y_k) - f^* - d_{k-1}(y_k - y_{k-1})) \leq \\
&\quad (1 - \alpha_k)(f(y_k) - f^* + \nu_{k-1}\|d_{k-1}\|^2) \leq \text{per (2.14)} \\
\left(1 - \frac{\nu_{k-1}\|d_{k-1}\|^2}{(f(y_k) - f^*) + \nu_{k-1}\|d_{k-1}\|^2}\right) (f(y_k) - f^* + \nu_{k-1}\|d_{k-1}\|^2) &= \text{per (2.34)} \\
\frac{f(y_k) - f^*}{(f(y_k) - f^*) + \nu_{k-1}\|d_{k-1}\|^2} (f(y_k) - f^* + \nu_{k-1}\|d_{k-1}\|^2) &= \\
&= f(y_k) - f^*.
\end{aligned}$$

**Osservazione** Il lemma continua a valere se si sostituisce (2.17) con (2.20), utilizzando (2.21) anziché (2.19) e (2.15) al posto di (2.14).

**Corollario 17** *Sotto le condizioni del lemma 16 vale:*

$$y^* \in Y^* \Rightarrow \forall k \quad d_k(y^* - y_k) \leq 0. \quad (2.36)$$

*Dimostrazione:*

$$\begin{aligned}
d_k(y^* - y_k) &\leq \\
f^* - f(y_k) + \varepsilon_k &\leq \text{per (2.19)} \\
f^* - f(y_k) + f(y_k) - f^* &= 0. \text{ per (2.35)}
\end{aligned}$$

**Teorema 18** *Sotto le condizioni del metodo 3, se  $c = \sup\{\|d_k\|\} < \infty$ ,  $\inf_k\{\alpha_k\} > 0$  e  $Y^* \neq \emptyset$ , allora  $\lim_{k \rightarrow \infty} f(y_k) = f^*$ .*

*Dimostrazione:*

Applicando ricorsivamente (2.25) si ottiene:

$$\begin{aligned}
&\|y_k - y^*\|^2 \leq \\
\|y_0 - y^*\|^2 - 2 \sum_{j=0}^{k-1} \nu_j d_j(y_j - y^*) + \sum_{j=0}^{k-1} \nu_j^2 \|d_j\|^2 &\leq \quad (2.37)
\end{aligned}$$

$$\|y_0 - y^*\|^2 + \sum_{j=0}^{k-1} \nu_j^2 \|d_j\|^2 \leq \text{per (2.36)} \quad (2.38)$$

$$\|y_0 - y^*\|^2 + c \sum_{j=0}^{k-1} \nu_j^2 \rightarrow L < \infty \text{ per (2.31)} \quad (2.39)$$

Dunque  $\{y_k\}$  è limitata.

Inoltre si ha:

$$|d_k(y_k - y^*)| \leq \|d_k\| \|y_k - y^*\|.$$

Cioè questa successione è limitata in quanto sia  $\{d_k\}$  che  $\{y_k\}$  sono limitate. Pertanto esiste una sottosuccessione convergente, supponiamo che nessuna sottosuccessione converga a 0:

$$\exists \varepsilon > 0 : \exists K : \forall k > K : 0 < \varepsilon < d_k(y_k - y^*).$$

E, da (2.37) e (2.31), si avrebbe  $\|y_k - y^*\|^2 \rightarrow -\infty$ , una contraddizione.

Pertanto esiste una sottosuccessione convergente a 0, sia essa  $d_{k_i}(y_{k_i} - y^*) \rightarrow 0$ .

Consideriamo, ora, la successione  $d_k(y_k - y_{k+1})$ , si ha:

$$\begin{aligned} |d_k(y_k - y_{k+1})| &\leq \\ \|d_k\| \|y_k - y_{k+1}\| &\leq \\ c \|y_k - y_{k+1}\| &\leq \\ c \|y_k - \widehat{y}_{k+1}\| &\leq \text{teorema 2.2.1 in [16]} \\ c \nu_k \|d_k\| &\leq c^2 \nu_k \rightarrow 0. \end{aligned}$$

Pertanto  $d_k(y_k - y_{k+1}) \rightarrow 0$ .

Sia  $\xi : 0 < \xi \leq \alpha_k \leq 1 \quad \forall k$ ; fissiamo  $\varepsilon > 0$  e siano  $\varepsilon_1, \varepsilon_2 > 0 : \frac{\varepsilon_1 + \varepsilon_2}{\xi} = \varepsilon$ . Per i due limiti sopra dimostrati abbiamo:

$$\exists I_1 : \forall i \geq I_1 \quad -\varepsilon_1 < d_{k_i}(y^* - y_{k_i}) < 0 < \varepsilon_1 \quad (2.40)$$

$$\exists K : \forall k \geq K \quad -\varepsilon_2 < d_k(y_k - y_{k+1}) < \varepsilon_2. \quad (2.41)$$

Sia  $I_2 : k_{I_2} \geq K$  abbiamo  $\forall i > \max\{I_1, I_2\}$ :

$$\begin{aligned} 0 < d_{k_i}(y^* - y_{k_i}) + \varepsilon_1 &\leq && \text{per (2.40)} \\ \widehat{d}_{k_i}(y^* - y_{k_i}) + \varepsilon_1 &= && \text{per (2.16)} \\ \alpha_{k_i} g_{k_i}(y^* - y_{k_i}) + (1 - \alpha_{k_i}) d_{k_i-1}(y^* - y_{k_i}) + \varepsilon_1 &\leq && \text{per (2.17)} \\ \alpha_{k_i} (f(y^*) - f(y_{k_i})) + (1 - \alpha_{k_i}) d_{k_i-1}(y^* - y_{k_i}) + \varepsilon_1 &= && \text{perché } g_{k_i} \in \partial^Y f(y_{k_i}) \\ \alpha_{k_i} (f(y^*) - f(y_{k_i})) + (1 - \alpha_{k_i}) d_{k_i-1}(y^* - y_{k_i-1}) + &&& \\ (1 - \alpha_{k_i}) d_{k_i-1}(y_{k_i-1} - y_{k_i}) + \varepsilon_1 &\leq && \\ \alpha_{k_i} (f(y^*) - f(y_{k_i})) + (1 - \alpha_{k_i}) d_{k_i-1}(y_{k_i-1} - y_{k_i}) + \varepsilon_1 &< && \text{per (2.36)} \\ \alpha_{k_i} (f(y^*) - f(y_{k_i})) + (1 - \alpha_{k_i}) \varepsilon_2 + \varepsilon_1 &\leq && \text{per (2.41)} \\ \alpha_{k_i} (f(y^*) - f(y_{k_i})) + \varepsilon_2 + \varepsilon_1 &&& \end{aligned}$$

$$\forall i \geq \max\{I_1, I_2\} \quad 0 < \xi(f(y_{k_i}) - f(y^*)) < \alpha_{k_i}(f(y_{k_i}) - f(y^*)) < \varepsilon_2 + \varepsilon_1.$$

Cioè:

$$\forall i \geq \max\{I_1, I_2\} \quad 0 < f(y_{k_i}) - f(y^*) < \frac{\varepsilon_2 + \varepsilon_1}{\xi} = \varepsilon.$$

Che vuol dire esattamente  $(f(y_{k_i}) - f(y^*)) \rightarrow 0$  e, analogamente a quanto visto nel teorema 14, esiste una sottosuccessione  $\{y_{k_j}\} \subseteq \{y_{k_i}\}$  tale che :  $y_{k_j} \rightarrow y^\infty \in Y^*$ .

Ora si ha  $\lim_{k \rightarrow \infty} \sum_{j=k}^{\infty} \nu_j^2 = 0$ . Fissato  $\varepsilon > 0$  sia:

$$K_1 : \forall k > K_1 : \sum_{j=k}^{\infty} \nu_j^2 < \frac{\varepsilon}{2c^2}$$

E poichè  $\|y_{k_j} - y^\infty\| \rightarrow 0$  esiste  $K_2 \geq K_1$  tale che  $\|y_{K_2} - y^\infty\|^2 < \frac{\varepsilon}{2}$ .  
 Si ha, quindi,  $\forall k > K_2$ :

$$\begin{aligned} \|y_k - y^\infty\|^2 &\leq \\ \|y_{K_2} - y^\infty\|^2 + \sum_{j=K_2}^{k-1} \nu_j^2 \|d_j\|^2 &\leq \quad \text{per (2.38)} \\ \|y_{K_2} - y^\infty\|^2 + c^2 \sum_{j=K_2}^{k-1} \nu_j^2 &\leq \\ \frac{\varepsilon}{2} + \frac{\varepsilon}{2} &= \varepsilon. \end{aligned}$$

Cioè  $\|y_k - y^\infty\|^2 \rightarrow 0$  e quindi  $y_k \rightarrow y^\infty \in Y^*$ ; inoltre, per continuità,  $f(y_k) \rightarrow f(y^*)$ .

**Osservazione** Si può ottenere un metodo alternativo sostituendo (2.17) con (2.20), modificando leggermente la dimostrazione di convergenza.

**Remark** Anche qui possiamo affermare a posteriori  $\lim_{k \rightarrow \infty} \varepsilon_k = 0$ . Inoltre, come per il metodo 2, una condizione di convergenza alternativa a (2.34), fornita dal teorema 3.4 in [1], è (2.33). Come prima per assicurare la convergenza dovremmo dimostrare  $\varepsilon_k = o(\nu_k)$ . Secondo le nostre ricerche bibliografiche non esistono in letteratura metodi del subgradiente deflesso *implementabili*, cioè che non prevedano condizioni ulteriori non dimostrate per il metodo in considerazione, che utilizzino stepsize *square summable*; l'analisi sarebbe, quindi, innovativa. Inoltre, come per gli altri metodi, noi prendiamo in considerazione un metodo di proiezione delle direzioni non considerato nelle analisi di convergenza in letteratura.

#### Metodo 4

Il metodo 3 permette di eliminare qualunque vincolo sullo stepsize introducendo una condizione sul parametro di deflessione. Questo ha senz'altro degli aspetti positivi: la scelta dello stepsize, infatti, è spesso critica per quanto riguarda, da un punto di vista pratico, la velocità di convergenza. Tuttavia la condizione (2.34) è di difficile implementazione poiché richiede la conoscenza del valore ottimo di funzione obiettivo  $f^*$ . Proponiamo, quindi, un metodo che approssima il valore obiettivo come nei metodi target level (vedi sezione successiva) e ne dimostriamo la convergenza. Diventa necessaria una trattazione separata in quanto l'approssimazione sul valore obiettivo non è sempre un *upper bound* e quindi la condizione (2.34) non è sempre verificata.

La convergenza di questo metodo sfrutta le proprietà del metodo 3; pertanto possiamo ancora sostituire (2.17) con (2.20) ed ottenere un metodo convergente.

**Metodo 4** Siano  $\delta_0 > 0$  e  $R > 0$  fissati;  $\sigma_0 = 0$ .

Sia (2.13), (2.17), (2.18), (2.31),

$$f_{rec}^k = \min\{f_{rec}^{k-1}, f(y_k)\} \quad (2.42)$$

$$\sigma_k = \begin{cases} \sigma_{k-1} + \nu_{k-1} & \text{se } \sigma_{k-1} < R \text{ e } f(y_{k-1}) > \dot{f}_{k-1} - \frac{1}{2}\delta_{k-1} \\ 0 & \text{altrimenti} \end{cases} \quad (2.43)$$

$$\delta_{k+1} = \begin{cases} \delta_k & \text{se } \sigma_k < R \\ \frac{1}{2}\delta_k & \text{altrimenti} \end{cases} \quad (2.44)$$

$$\dot{f}_k = \begin{cases} f(y_k) & \text{se } f(y_k) \leq \dot{f}_k - \frac{1}{2}\delta_k \\ \dot{f}_{k-1} & \text{altrimenti} \end{cases} \quad (2.45)$$

$$f_{lev}^k = \dot{f}_k - \delta_k \quad (2.46)$$

E le seguenti condizioni su  $\alpha_k$ :

$$\nu_{k-1} \frac{\|d_{k-1}\|^2}{(f(y_k) - f_{lev}^k) + \nu_{k-1}\|d_{k-1}\|^2} \leq \alpha_k \leq 1. \quad (2.47)$$

$$\delta_k = \frac{1}{2}\delta_{k-1} \Rightarrow \alpha_k = 1 \quad (2.48)$$

Il metodo iterativo utilizzato per ottimizzare  $f$ .

Si utilizza, qui, la seguente notazione:

- $\dot{f}_k$  è il valore obiettivo all'ultimo aggiornamento del *target level*
- $\sigma_k$  è l'indicatore del percorso.
- $R$  è una limitazione sull'indicatore del percorso: intuitivamente il livello viene considerato *unfeasible* se spostandosi si eccede il raggio  $R$  senza un miglioramento (sufficiente) nei valori della funzione.
- $\delta_k$  controlla il livello; inoltre  $\frac{1}{2}\delta_k$  è il miglioramento obiettivo atteso.

**Lemma 19** Sotto le condizioni del metodo 4 vale:

$$\varepsilon_k \leq f(y_k) - f_{lev}^k \quad (2.49)$$

*Dimostrazione:* Notiamo preliminarmente che:

$$\forall k : \alpha_k < 1 \Rightarrow f_{lev}^k \leq f_{lev}^{k-1}. \quad (2.50)$$

Se il livello viene aggiornato perché *unfeasible* (i.e.  $\delta_k = \frac{1}{2}\delta_{k-1}$ ) allora  $\alpha_k = 1$ , dunque se  $\alpha_k < 1$  allora  $\delta_k = \delta_{k-1}$ . Dunque:

$$f_{lev}^k = \dot{f}_k - \delta_k = \dot{f}_k - \delta_{k-1} \leq \dot{f}_{k-1} - \delta_{k-1} = f_{lev}^{k-1}.$$

Procediamo, ora, per induzione su  $k$ ;  $\varepsilon_0 = 0 \leq f(y_0) - f_{lev}^0$ . Consideriamo l'iterazione  $k$ : se  $\alpha_k = 1$  allora  $\varepsilon_k = 0 \leq f(y_k) - f_{lev}^k$ ; supponiamo dunque  $\alpha_k < 1$ .

$$\begin{aligned}
\varepsilon_k &= (1 - \alpha_k)(f(y_k) - f(y_{k-1}) - d_k(y_k - y_{k-1}) + \varepsilon_{k-1}) \leq && \text{per (2.19)} \\
(1 - \alpha_k)(f(y_k) - f(y_{k-1}) - d_k(y_k - y_{k-1}) + f(y_{k-1}) - f_{lev}^{k-1}) &= && \text{per ip. ind.} \\
(1 - \alpha_k)(f(y_k) - f_{lev}^{k-1} - d_k(y_k - y_{k-1})) &\leq && \\
(1 - \alpha_k)(f(y_k) - f_{lev}^k - d_k(y_k - y_{k-1})) &\leq && \text{per (2.50)} \\
(1 - \alpha_k)(f(y_k) - f_{lev}^k + \nu_{k-1}\|d_{k-1}\|^2) &\leq && \text{per (2.14)} \\
\left(1 - \frac{\nu_{k-1}\|d_{k-1}\|^2}{(f(y_k) - f_{lev}^k) + \nu_{k-1}\|d_{k-1}\|^2}\right) (f(y_k) - f_{lev}^k + \nu_{k-1}\|d_{k-1}\|^2) &= && \text{per (2.47)} \\
\frac{f(y_k) - f_{lev}^k}{(f(y_k) - f_{lev}^k) + \nu_{k-1}\|d_{k-1}\|^2} (f(y_k) - f_{lev}^k + \nu_{k-1}\|d_{k-1}\|^2) &= && \\
&= && f(y_k) - f_{lev}^k.
\end{aligned}$$

**Osservazione** Il lemma continua a valere se si sostituisce (2.17) con (2.20), utilizzando (2.21) anziché (2.19) e (2.15) al posto di (2.14).

**Lemma 20** Sotto le condizioni del metodo 4, se  $Y^* \neq \emptyset$ ; detta, ad ogni  $k$ ,  $\dot{k}(k) \leq k$  l'iterazione all'ultimo aggiornamento di  $\delta_k$ , si ha:

$$\dot{k}(k) \rightarrow \infty \quad e \quad \delta_k \rightarrow 0.$$

*Dimostrazione:* Supponiamo  $\dot{k}(k) \rightarrow \bar{k} < \infty$ , si avrebbe:

$$\forall k > \bar{k} : \quad \sigma_k = \sum_{j=\bar{k}}^k \nu_k < R$$

Ma, per (2.31),  $\sum_{j=\bar{k}}^k \nu_k \rightarrow \infty$ , contro l'ipotesi. Pertanto  $\bar{k} = \infty$ .

Supponiamo ora  $\delta_k \rightarrow \delta > 0$ .

Poiché  $\delta_k = \frac{1}{2^m} \delta_0$  per qualche  $m$ , si ha:  $\delta_k \rightarrow \delta \Leftrightarrow \exists K : \forall k > K \delta_k = \delta$ . E, poiché  $\dot{k}(k) \rightarrow \infty$ :

$$\forall k > K \quad \dot{k}(k)_{+1} - \dot{k}(k) \leq -\frac{1}{2}\delta.$$

Si avrebbe, quindi,  $\inf_k \{f(y_k)\} = -\infty$ , impossibile poiché  $Y^* \neq \emptyset$ .

**Teorema 21** Utilizzando il metodo 4, se  $Y^* \neq \emptyset$ , vale:

$$\inf_k \{f(y_k)\} = f^*$$

*Dimostrazione:* La dimostrazione segue lo stesso schema di quella del metodo target value. Supponiamo, per assurdo,  $\inf_k \{f(y_k)\} > f^*$ , si avrebbe:

$$\inf_k \{f(y_k)\} - \varepsilon > f^*; \tag{2.51}$$

per qualche  $\varepsilon$ .

Pertanto si avrebbe per un qualche  $K$ ,  $\forall k > K$ :

$$\begin{aligned}
f_{lev}^k &= \dot{k} - \delta_k \geq \\
\inf_k \{f(y_k)\} - \delta_k &\geq && \text{perché } \dot{k} = f(y_j) \text{ per qualche } j \leq k \\
\inf_k \{f(y_k)\} - \varepsilon &> f^* && \text{perché } \delta_k \rightarrow 0
\end{aligned}$$

e, dunque, (2.47)  $\Rightarrow$  (2.34), inoltre (2.49)  $\Rightarrow$  (2.35). Pertanto possiamo applicare il teorema 18 ottenendo  $f(y_k) \rightarrow f^*$ , contro le ipotesi. Quindi  $\inf_k \{f(y_k)\} = f^*$ .

**Remark** Eliminando la condizione (2.47) possiamo fare le stesse considerazioni esposte riguardo al metodo 3 in merito alle ipotesi di convergenza. D'altronde questo metodo è pensato come una versione facilmente implementabile del metodo 3 e quindi non troviamo in letteratura una trattazione analoga.

### 2.1.4 Target Value stepsize

**Metodo 5 (Algoritmo Target Value)** Si seleziona una soluzione iniziale  $y_1 \in Y$ . Si inizializzano i parametri  $R > 0$ ,  $\delta_0 > 0$ , le quantità  $f_{rec}^0 = \infty$ ,  $\sigma_0 = 0$  e i contatori  $k = l = k(1) = 1$ .

**Passo 1** (Objective evaluation). Si calcolano  $f(y_k)$  e  $g_k \in \partial f(y_k)$ . Si aggiorna  $f_{rec}^k = \min\{f_{rec}^{k-1}, f(y_k)\}$ , e  $y_{rec}^k : f_{rec}^k = f(y_{rec}^k)$ .

**Passo 2** (Sufficient descent detection). Se

$$f(y_k) \leq f_{rec}^{k(l)} - \frac{1}{2}\delta_l \quad (2.52)$$

si inizia un nuovo gruppo di iterazioni:  $k(l+1) = k$ ,  $\delta_{l+1} = \delta_l$ ,  $l = l + 1$ ,  $\sigma_k = 0$ .

**Passo 3** (Target infeasibility detection). Se vale:

$$\sigma_k > R \quad (2.53)$$

allora il target è troppo basso e si inizia un nuovo gruppo di iterazioni:  $k(l+1) = k$ ,  $\delta_{l+1} = \frac{1}{2}\delta_l$ ,  $y_k = y_{rec}^k$ ,  $l = l + 1$ ,  $\sigma_k = 0$ . E si forza  $\alpha_k = 1$ .

**Passo 4** (Projections). Si calcola il livello  $f_{lev}^k = f_{rec}^{k(l)} - \delta_l$ . Si calcola il parametro di deflessione  $\alpha_k \in (0, 1]$  e la direzione aggregata tramite (2.17) o (2.20) e (2.18) Si sceglie

$$\beta_k \in (0, \alpha_k]. \quad (2.54)$$

Si calcola

$$\nu_k = \beta_k \frac{f(y_k) - f_{lev}^k}{\|d_k\|^2} \quad (2.55)$$

e si calcola la nuova soluzione tramite (2.13).

Si aggiorna

$$\sigma_{k+1} = \sigma_k + \|\hat{y}_{k+1} - y_k\| \quad (2.56)$$

Si incrementa  $k$  e si torna al passo 1.

Questo metodo dimostra la propria convergenza riconducendosi al metodo 2, quindi anche qui si può sostituire (2.17) con (2.20) ed ottenere un metodo convergente.

**Lemma 22** Utilizzando il metodo 5, ad ogni level update per improvement, cioè  $k = k(l)$  e:

$$f(y_{k(l)}) \leq f_{rec}^{k(l-1)} - \frac{1}{2}\delta_{l-1},$$

vale:

$$f_{lev}^k \leq f_{lev}^{k-1} + \frac{1}{2}\delta_{l-1} \leq f_{lev}^{k-1} \quad (2.57)$$

*Dimostrazione:*

$$\begin{aligned} f_{lev}^k &= f_{rec}^{k(l)} - \delta_l \leq \\ & f_{k(l)} - \delta_l \leq \\ f_{rec}^{k(l-1)} - \frac{1}{2}\delta_{l-1} - \delta_{l-1} &= \text{per ipotesi e perché } \delta_l = \delta_{l-1} \\ f_{lev}^{k-1} - \frac{1}{2}\delta_{l-1} &\leq f_{lev}^{k-1}. \end{aligned}$$

**Lemma 23** Per il metodo 5 vale:

$$\varepsilon_k \leq (1 - \alpha_k)(f(y_k) - f_{lev}^k). \quad (2.58)$$

*Dimostrazione:* Osserviamo che  $\forall k : f(y_k) \geq f_{lev}^k$ ; infatti

$$\forall k(l) \leq i \leq k : f(y_i) > f_{rec}^{k(l)} - \frac{1}{2}\delta_l \geq f_{rec}^{k(l)} - \delta_l = f_{lev}^k.$$

Per induzione su  $k$ :  $\varepsilon_0 = 0 = (1 - \alpha_0)(f(y_0) - f_{lev}^0)$ .

Consideriamo l'iterazione  $k$ : se c'è stato un reset, ovvero  $k = k(l)$  per qualche  $l$  e  $\sigma_k > R$ , allora  $\alpha_k = 1 \Rightarrow \varepsilon_k = 0 = (1 - \alpha_k)(f(y_k) - f_{lev}^k)$ .

Nei restanti casi, ovvero  $\nexists l : k = k(l)$  oppure  $k = k(l)$  per qualche  $l$  e  $f(y_k) \leq f_{rec}^{k(l-1)} - \frac{1}{2}\delta_{l-1}$ , vale:

$$f_{lev}^k \leq f_{lev}^{k-1}. \quad (2.59)$$

Infatti nel primo caso  $f_{lev}^k = f_{lev}^{k-1}$  e nel secondo vale (2.57).

$$\begin{aligned} \varepsilon_k &= (1 - \alpha_k)(f(y_k) - f(y_{k-1}) - d_{k-1}(y_k - y_{k-1}) + \varepsilon_{k-1}) \leq \\ & (1 - \alpha_k)(f(y_k) - f(y_{k-1}) + \nu_{k-1}\|d_{k-1}\|^2 + \varepsilon_{k-1}) = \text{per (2.14)} \\ (1 - \alpha_k)(f(y_k) - f(y_{k-1}) + \beta_{k-1}(f(y_{k-1}) - f_{lev}^{k-1}) + \varepsilon_{k-1}) &\leq \text{per (2.55)} \\ (1 - \alpha_k)(f(y_k) - f(y_{k-1}) + \alpha_{k-1}(f(y_{k-1}) - f_{lev}^{k-1}) + \varepsilon_{k-1}) &\leq \text{per (2.54)} \\ (1 - \alpha_k)(f(y_k) - f(y_{k-1}) + \alpha_{k-1}(f(y_{k-1}) - f_{lev}^{k-1}) + (1 - \alpha_{k-1})(f(y_{k-1}) - f_{lev}^{k-1})) &\leq \text{per ip. ind.} \\ & (1 - \alpha_k)(f(y_k) - f_{lev}^{k-1}) \leq \\ & (1 - \alpha_k)(f(y_k) - f_{lev}^k) \text{ per (2.59)}. \end{aligned}$$

**Osservazione** Sia  $H_k = \{y \in \mathbb{R}^n : f(y_k) + d_k(y - y_k) \leq f_{lev}^k\}$ , allora

$$d_{H_k}(y_k) = \frac{f(y_k) - f_{lev}^k}{\|d_k\|}.$$

**Lemma 24** Utilizzando il metodo 5 vale:

$$\forall k \forall y \in Y \cap \mathcal{L}_f(f_{lev}^k) \quad \|y_{k+1} - y\|^2 \leq \|y_k - y\|^2 - \beta_k(2\alpha_k - \beta_k)d_{H_k}^2(y_k).$$

*Dimostrazione:* Si noti che

$$\mathcal{L}_f(f_{lev}^k) \neq \emptyset \Rightarrow f^* \leq f(y) \leq f_{lev}^k. \quad (2.60)$$

Vale, quindi, (2.22) e, inoltre, (2.60) e (2.58)  $\Rightarrow$  (2.23).

Si ha:

$$\begin{aligned} d_k(y - y_k) &\leq f(y) - f(y_k) + \varepsilon_k \leq \\ &f(y) - f(y_k) + (1 - \alpha_k)(f(y_k) - f^*) \leq \quad \text{per (2.23)} \\ f^* - f(y_k) + (1 - \alpha_k)(f(y_k) - f^*) &= \alpha_k(f^* - f(y_k)). \end{aligned} \quad (2.61)$$

$$\begin{aligned} \|y_{k+1} - y\| &\leq \|y_k - y\|^2 - 2\nu_k(y_k - y)d_k + \nu_k^2\|d_k\|^2 = \quad \text{per (2.25)} \\ \|y_k - y\|^2 - 2\beta_k \frac{f(y_k) - f_{lev}^k}{\|d_k\|^2} (y_k - y)d_k + \left(\beta_k \frac{f(y_k) - f_{lev}^k}{\|d_k\|^2}\right)^2 \|d_k\|^2 &\leq \\ \|y_k - y\|^2 - 2\beta_k \alpha_k \frac{f(y_k) - f_{lev}^k}{\|d_k\|^2} (f(y_k) - f^*) + \left(\beta_k \frac{f(y_k) - f_{lev}^k}{\|d_k\|^2}\right)^2 \|d_k\|^2 &\leq \quad \text{per (2.61)} \\ \|y_k - y\|^2 - 2\beta_k \left(\frac{f(y_k) - f_{lev}^k}{\|d_k\|}\right)^2 + \beta_k^2 \left(\frac{f(y_k) - f_{lev}^k}{\|d_k\|}\right)^2 &\leq \quad \text{per (2.60)} \\ \|y_k - y\|^2 - \beta_k(2\alpha_k - \beta_k)d_{H_k}^2(y_k). \end{aligned}$$

**Lemma 25** Utilizzando il metodo 5, se  $c = \sup\{\|d_k\|\} < \infty$  e  $\xi = \inf_k\{\beta_k\} > 0$ , allora  $l \rightarrow \infty$ , e  $\lim_{k \rightarrow \infty} f(y_k) = -\infty \vee \delta_l \rightarrow 0$ .

*Dimostrazione:* Supponiamo, per assurdo,  $l \rightarrow \bar{l} < \infty$ . Si avrebbe:

$$\sigma_{k+1} = \sigma_k + \|\hat{y}_{k+1} - y_k\| \leq R \quad \forall k \geq K;$$

per qualche  $K$ ; e dunque  $\|\hat{y}_{k+1} - y_k\| \rightarrow 0$ .

Ma:

$$\begin{aligned} \|\hat{y}_{k+1} - y_k\| &= \nu_k \|d_k\| = \\ &\beta_k \frac{f(y_k) - f_{lev}^k}{\|d_k\|} \geq \\ &\frac{\xi}{c} (f(y_k) - f_{lev}^k). \end{aligned}$$

e dunque  $f(y_k) - f_{lev}^k \rightarrow 0$ .

Ma, poichè  $\bar{l}$  è finito,  $f(y_k) > f_{rec}^{k(l)} - \frac{1}{2}\delta_l$ ; e dunque  $\forall k > K$ , per qualche  $K$ , detto  $\delta = \delta_{\bar{l}}$ :

$$\begin{aligned} f(y_k) - f_{lev}^k &> \\ f_{rec}^{k(l)} - \frac{1}{2}\delta_l - f_{lev}^k &= \\ \frac{1}{2}\delta_l &> \frac{1}{2}\delta > 0; \end{aligned}$$

una contraddizione.

Supponiamo, ora  $\delta = \lim_{l \rightarrow \infty} \delta_l > 0$ :

$$\forall k \quad f_{rec}^{k(l+1)} \leq f_{rec}^{k(l)} - \delta_l \leq f_{rec}^{k(l)} - \delta \Rightarrow f(y_k) \rightarrow -\infty.$$

**Teorema 26** Utilizzando il metodo 5, se  $c = \sup\{\|d_k\|\} < \infty$ ,  $\inf_k\{\beta_k\} > 0$  e  $Y^* \neq \emptyset$ , allora  $\liminf_{k \rightarrow \infty} f(y_k) = f^*$ .

*Dimostrazione:* Se  $\lim_{l \rightarrow \infty} \delta_l > 0$  allora  $\lim_{k \rightarrow \infty} f(y_k) = f^* = -\infty$  e si avrebbe  $Y^* = \emptyset$ , dunque  $\lim_{l \rightarrow \infty} \delta_l = 0$ .

$$\sigma_k = \sum_{j=k(l)}^{k-1} \|\hat{y}_{j+1} - y_j\| = \sum_{j=k(l)}^{k-1} \nu_j \|d_j\|.$$

$$\begin{aligned} k(l+1) = k &\Rightarrow \\ R < \sum_{j=k(l)}^{k-1} \nu_j \|d_j\| &\leq c \sum_{j=k(l)}^{k-1} \nu_j \Rightarrow \\ \sum_{j=0}^k \nu_j &= \sum_{l \in L} \sum_{j=k(l-1)}^{k(l)-1} \nu_j > \sum_{l \in L} \frac{R}{c} = \infty. \end{aligned}$$

Supponiamo, per assurdo,  $\inf_k f(y_k) > f^*$ .

$$\exists \varepsilon > 0 : \quad \inf f(y_k) - \varepsilon > f^*.$$

e, inoltre,

$$\exists K : \forall k > K : \quad f_{lev} = f_{rec}^{k(l)} - \delta_l \geq \inf f(y_k) - \varepsilon$$

perché  $\delta_l \rightarrow 0$ .

Dunque  $y^* \in \mathcal{L}_f(f_{lev}^k)$  e:

$$\begin{aligned} \|y_{k+1} - y^*\|^2 &\leq \|y_k - y^*\|^2 - \beta_k(2\alpha_k - \beta_k)d_{H_k}^2(y_k) \leq \\ &\|y_k - y^*\|^2 - \xi^2 \frac{(f(y_k) - f_{lev}^k)^2}{\|d_k\|^2} \leq . \end{aligned}$$

E dunque:

$$\begin{aligned}
& \sum_{k=k(\hat{l})}^{\infty} (\|y_{k+1} - y^*\|^2 - \|y_k - y^*\|^2) = \\
& -\|y_{k(\hat{l})} - y^*\|^2 \leq -\xi^2 \sum_{k=k(\hat{l})}^{\infty} \nu_k^2 \|d_k\|^2 \Rightarrow \\
& \xi^2 \sum_{k=k(\hat{l})}^{\infty} \nu_k^2 \|d_k\|^2 \leq \|y_{k(\hat{l})} - y^*\|^2.
\end{aligned}$$

E  $\sum_{k=0}^{\infty} \nu_k^2 \|d_k\|^2 < \infty$ ; inoltre  $f_{lev}^k > f^*$  e (2.58) implicano (2.23). Dunque possiamo applicare il teorema 15 ottenendo  $f(y_k) \rightarrow f^*$ , contro le ipotesi, perciò  $\inf f(y_k) = f^*$ .

Ora, poiché  $l \rightarrow \infty$ ,  $\{f_{rec}^{k(l)}\}$  è una sottosuccessione di  $\{f(y_k)\}$  monotona non crescente inferiormente limitata e quindi convergente. Pertanto  $\liminf_{k \rightarrow \infty} f(y_k) = f^*$ .

**Nota:** La difficoltà di questa dimostrazione risiede sostanzialmente nel ricondursi al metodo 2; volendosi ricondurre al metodo 1, però, non possiamo utilizzare la condizione  $\inf\{\beta_k\} > 0$  poiché  $\nu_k \neq \beta_k \frac{f(y_k) - f^*}{\|d_k\|^2}$  e, quindi, dovremmo introdurre la seguente condizione:

$$\inf_k \left\{ \beta_k \frac{f(y_k) - f_{lev}^k}{f(y_k) - f^*} \right\} > 0.$$

Il metodo 2 permette di utilizzare lo stesso tipo di stepsize eliminando la condizione su  $\beta_k$ , introducendo la condizione (2.31).

**Remark** Un metodo del subgradiente deflesso con stepsize *target value* convergente è descritto in [6]. Questo metodo è sostanzialmente diverso da quello qui proposto. Durante la lavorazione di questo documento è stato pubblicato [24] che dimostra la convergenza di un metodo molto simile a quello qui proposto, sotto le stesse condizioni da noi considerate (in particolare  $\alpha_k \geq \beta_k$ ). L'unica differenza consiste nella proiezione delle direzioni che [24] non effettua. Inoltre, sebbene si tratti dello stesso metodo, le dimostrazioni sono molto diverse in quanto [24] estende la dimostrazione di Goffin-Kiwiel in [2] al caso deflesso, mentre noi estendiamo la dimostrazione di Bertsekas-Nedić per il caso incrementale in [11].

## Capitolo 3

# Un unico schema algoritmico

Abbiamo, fin qui, introdotto le principali caratteristiche dei metodi del subgradiente/volume che vogliamo implementare. A questo punto descriviamo uno schema algoritmico con l'intento di fornire un'astrazione che ci guidi nell'implementazione di ogni algoritmo specifico. Per ogni passo che compone l'algoritmo consideriamo varie implementazioni proposte in letteratura cercando di discuterne separatamente le proprietà di convergenza; in seguito presenteremo un resoconto di varie implementazioni presenti in bibliografia mostrando come ricondurle al nostro schema.

Quello che proponiamo è un particolare schema incrementale. Ogni  $h$  iterazioni il metodo calcola un subgradiente totale e valuta la funzione. A questo punto, per “sfruttare” al meglio l'informazione calcolata, esegue un'iterazione non incrementale. In questo modo utilizzando come parametro  $h = 1$  si ottiene il metodo non incrementale, per un metodo incrementale puro si può impostare  $h = \infty$ ; si tenga presente che con questa scelta l'algoritmo non calcola mai la funzione totale e, quindi, alcune regole potrebbero portare a fenomeni di non convergenza.

Di seguito vengono proposte delle versioni incrementali dei metodi deflessi e del volume. Nel cosiddetto caso *unsafe*, cioè quando non si calcola la funzione al termine di ogni ciclo incrementale, questi metodi utilizzano le approssimazioni introdotte nella sezione 1.9. Si tenga comunque presente che né l'utilizzo di dette approssimazioni né le tecniche di deflessione nel caso incrementale, sono supportati da dimostrazioni di convergenza.

### Notazione

- $\tilde{y}$  rappresenta il cosiddetto *stability center*, ovvero la soluzione aggiornata ad ogni *serious step* ed  $\tilde{f}$  rappresenta il valore della funzione obiettivo  $f(\tilde{y})$  (cfr. sez. 1.3).
- $\check{f}$  rappresenta il valore obiettivo all'ultimo aggiornamento del *target level*, per i metodi VTVM.
- $\bar{y}$  rappresenta il cosiddetto *record point*, ovvero la miglior soluzione in va-

lore obiettivo generata fino all'iterazione corrente; il miglior valore della funzione obiettivo è espresso come  $\bar{f} = f(\bar{y})$ .

- $\sigma$  è detto “indicatore del percorso” e viene utilizzato da alcuni algoritmi per definire delle condizioni di aggiornamento.
- $\Delta$  è un “indicatore del percorso” nello spazio dei valori obiettivi e viene anch'esso utilizzato da alcuni algoritmi per definire delle condizioni di aggiornamento.
- $\delta$  e  $\varphi$  sono dei parametri utilizzati per delle condizioni di aggiornamento: rappresentano il “miglioramento” atteso per l'aggiornamento rispettivamente di  $\tilde{y}$  e  $\tilde{f}$ .
- $\bar{x}$  rappresenta la soluzione primale aggregata.
- $p$  rappresenta la soluzione duale aggregata.
- $g$  indica il subgradiente, eventualmente condizionale (cfr. sez. 1.6), della funzione nella soluzione corrente.
- $d$  indica la direzione di spostamento all'iterazione corrente.
- $\nu$  indica la lunghezza del passo.
- $\varepsilon$  esprime l'approssimazione del subgradiente (cfr. sez. 1.4)
- $\tau$  rappresenta l'accuratezza dell'algoritmo (cfr. sez. 1.8)
- $\hat{d}$  rappresenta la direzione aggregata non ancora proiettata sul tangent cone dell'insieme ammissibile.
- $h$  indica ogni quante iterazioni incrementali si effettua un'iterazione non incrementale
- $\hat{f}$  e  $\hat{g}$  sono rispettivamente approssimazioni di  $f(y)$  e  $g \in \partial f(y)$ . (cfr. sez. 1.9).

**Inizializzazione** Si sceglie una soluzione iniziale  $y \in Y$ . Si inizializzano  $\tilde{y} := \bar{y} := y$ ,  $\bar{f} := f(y)$ ,  $\hat{f} := f(y)$ ,  $\sigma := 0$ ,  $\Delta := 0$ . Si inizializza l'errore  $\varepsilon > 0$  e la tolleranza  $\tau > 0$ . Si calcolano  $d := g \in \partial_\varepsilon f_Y(y)$ . Si sceglie il parametro  $h$ .

**Passo 1** (*Object evaluation*) Ogni  $h$  iterazioni si calcolano  $\hat{f} = f = f(y)$  e  $\hat{g} = g \in \partial_\varepsilon f_Y(y)$  e si aggiorna  $\bar{f} := \min\{\bar{f}, f(y)\}$ .

Se  $\hat{f} < \bar{f}$ , allora si aggiornano  $\Delta := \Delta + \bar{f} - \hat{f}$ .

**Passo 2** (*Stopping criterion*) Si valutano tutti i criteri di terminazione e si termina se uno di essi vale.

**Passo 3** (*Sufficient descent detection*) Ogni  $h$  iterazioni si valuta la *sufficient descent condition*:

$$(f(y) < f(\tilde{y}) + \delta); \quad (3.1)$$

se risulta vera si esegue un *serious step*: si aggiorna  $\delta$ , si aggiornano  $\tilde{y} := y$  e  $\tilde{f} := f(y)$ .

Se utilizziamo un metodo *target value* per l'aggiornamento di  $\nu$  (si veda A.2) valutiamo la *target update condition*: se

$$\widehat{f} < \check{f} - \varphi, \quad (3.2)$$

allora si aggiornano  $\check{y} := y$ ,  $\check{f} := \widehat{f}$  e si aggiorna  $\varphi$  secondo le regole dell'algoritmo specifico.

**Passo 4** (*Oscillation detection*) Si utilizzano delle condizioni di limitazione su alcune quantità (ad esempio, nel caso di algoritmi *path-based* si verifica una condizione su  $\sigma$ ), se queste valgono si effettua un reset:  $y := \bar{y}$  e si aggiornano le altre quantità. Si eseguono in questo passo anche le tecniche di *target unfeasibility detection* di VTVM.

**Passo 5** (*Incremental cycle*)

Ogni  $h$  iterazioni si effettua un'iterazione non incrementale: si calcolano i parametri  $\alpha$  e  $\alpha'$ . Si calcola la direzione aggregata  $\widehat{d}$  secondo (3.7i) o (3.7j) e la sua proiezione sul tangent cone:

$$d = -P_{T_Y(y)}(-\widehat{d}). \quad (3.3)$$

Si calcola la soluzione primale aggregata:

$$\bar{x} = \alpha' x + (1 - \alpha') \bar{x}. \quad (3.4)$$

Si calcola lo stepsize  $\nu$  e  $y := P_Y(y - \nu d)$ .

Negli altri casi si inizia un ciclo incrementale: si calcola lo stepsize  $\nu$ ,  $y_0 = y$  e: per ogni componente  $i = 1, \dots, m$ :

- Si calcola un subgradiente  $g_i \in \partial f_i(y_i)$  e il valore della funzione  $f_i(y_i)$ .
- Si calcola il coefficiente  $\alpha$  (utilizzato per la *direction deflection*) Si aggiornano le medie, in particolare si aggiorna la direzione aggregata  $\widehat{d}_i$  e si calcola la sua proiezione sul tangent cone

$$d_i = -P_{T_Y(y)}(-\widehat{d}_i) \quad (3.5)$$

- Si calcola la nuova soluzione incrementale

$$y_{i+1} = P_Y(y_i - \nu d_i)$$

Si incrementa  $i$ .

Si calcolano  $y_{m+1} = y$ ,  $\widehat{f} = \sum_{i=1}^m f_i(y_i)$  e  $\widehat{g} = \sum_{i=1}^m g_i$ ,  $\sigma := \sigma + \|\widehat{y} - y\|$ .

Per ogni passo, quindi, sono possibili diverse scelte implementative; cercheremo ora di recensirle e di discuterne la convergenza. Sostanzialmente le scelte che determinano l'algoritmo utilizzato sono la scelta della lunghezza del passo (al passo 5 e passi 3 e 4), il calcolo della direzione aggregata (passo 5) e il calcolo della nuova soluzione (passo 3) dove si distingue tra metodi del subgradiente ( $\delta = +\infty$ ) e metodi del volume ( $\delta < +\infty$ ).

**Passo 3** Sono possibili quattro differenti *sufficient descent condition* (scelte di  $\delta$  in (3.1)):

$$\delta = +\infty \quad \begin{array}{l} \text{metodo del} \\ \text{subgradiente} \end{array} \quad (3.6a)$$

$$\delta := 0 \quad \begin{array}{l} \text{metodo standard} \\ \text{del volume} \end{array} \quad (3.6b)$$

$$\begin{aligned} \hat{\varepsilon} &:= \alpha^-(1 - \alpha^-)(p^- - y^-)(g^- - d^-) + \\ &\quad (1 - \alpha^-)\hat{\varepsilon} \\ \delta &:= m_1(\nu\|d\|^2 + |d^T(p - \tilde{y})| + \hat{\varepsilon}) \quad \text{RVA.} \quad (3.6c) \\ &\quad m_1 \in (0, 1) \text{ costante.} \end{aligned}$$

$$\begin{aligned} \hat{\varepsilon} &:= \alpha(f(\tilde{y}) + g(y - \tilde{y}) - f(y)) + (1 - \alpha)\hat{\varepsilon} \\ \delta &:= m_1(\nu\|d\|^2 + \hat{\varepsilon}) \quad \text{BVA.} \quad (3.6d) \\ &\quad m_1 \in (0, 1) \text{ costante.} \end{aligned}$$

(3.6c) utilizza una soluzione duale aggregata, calcolata ad ogni iterazione come:

$$p := \alpha y + (1 - \alpha)p.$$

**Passo 5** Per il parametro  $\alpha$  sono disponibili sei scelte:

$$\alpha := 1 \quad \begin{array}{l} \text{subgradiente} \\ \text{ordinario} \end{array} \quad (3.7a)$$

$$\alpha \text{ costante} \quad (3.7b)$$

$$\begin{aligned} \alpha &:= \min\{\alpha_{min}, \alpha_{fact}\alpha\} \\ &\text{aggiornato ogni } \alpha_{int} \text{ iterazioni} \\ &\text{se } \frac{f(y) - \bar{f}}{|\bar{f}|} \leq \eta \quad (3.7c) \end{aligned}$$

$\alpha_{min}, \alpha_{fact}, \alpha_{int}, \eta$  costanti

$$\begin{aligned} E &:= (g)(y - \tilde{y}) \\ \hat{E} &:= (d)(p - \tilde{y}) + \hat{\varepsilon} \quad \text{RVA.} \quad (3.7d) \\ \alpha &:= \min_{\alpha \in [0,1]} \left\{ \frac{\nu}{2} \|\alpha g + (1 - \alpha)d\|^2 + \alpha E + (1 - \alpha)\hat{E} \right\} \end{aligned}$$

$$\begin{aligned} E &:= f(y) + g(y - \tilde{y}) - f(\tilde{y}) \\ \alpha &:= \min_{\alpha \in [0,1]} \left\{ \frac{\nu}{2} \|\alpha g + (1 - \alpha)d\|^2 + \alpha E + (1 - \alpha)\hat{\varepsilon} \right\} \quad \text{BVA.} \quad (3.7e) \end{aligned}$$

$$\alpha := \min_{\alpha \in (0,1]} \left\| g - d + \frac{1}{\alpha}d \right\|^2 \quad (3.7f)$$

$\nu_k := \lambda_k \mu_k, \lambda_k \rightarrow 0, \sum \lambda_k = \infty.$ $\mu_k := \max\{\ g_k\ , \ g_k\ ^2/G\}^{-1} = \min\{1, G/\ g_k\ \}\ g_k\ ^{-1}$ $\mu_k := \max\{1, \ g_k\ ^2/G^2\}^{-1} = \min\{1, G^2/\ g_k\ ^2\}$ $\mu_k := \max\{G^2, \ g_k\ ^2\}^{-1} = \min\{1, G^2/\ g_k\ ^2\}G^{-2}$ $\mu_k := \ g_k\ ^{-1}$ $\mu_k := (G + \ g_k\ )^{-1}$ $\mu_k := \max\{G, \ g_k\ \}^{-1}$ $\mu_k := (G^2 + \ g_k\ ^2)^{-1/2}$ $\mu_k := \max\{\lambda_k, \ g_k\ \}^{-1} = \min\{\lambda_k^{-1}, \ g_k\ ^{-1}\}$ $\mu_k := \ g_k\ ^{-2}$ $\mu_k := \max\{\lambda_k^2, \ g_k\ ^2\}^{-1}$
---

Tabella 3.1: Regole per i *diminishing stepsize*

Riscrivendo (3.7d) otteniamo:

$$\alpha = \min_{\alpha \in [0,1]} \left\{ \Phi(\alpha) = \frac{\nu}{2} \|g - d\|^2 \alpha^2 + (\nu(g - d)d + E - \hat{E})\alpha + \frac{\nu}{2} \|d\|^2 + \hat{E} \right\}.$$

$\Phi(\alpha)$  è un polinomio di secondo grado, convesso, dunque:

$$\alpha = P_{[0,1]}(\bar{\alpha}) \Leftrightarrow \frac{d}{d\alpha} \Phi(\bar{\alpha}) = 0 \Leftrightarrow \bar{\alpha} = \frac{\hat{E} - E - \nu(g - d)d}{\nu \|g - d\|^2}.$$

Analogamente (3.7e) si risolve con:

$$\alpha = P_{[0,1]} \left( \frac{\hat{\varepsilon} - E - \nu(g - d)d}{\nu \|g - d\|^2} \right)$$

(3.7f) è, invece, risolto da:

$$\alpha = \begin{cases} \frac{\|d\|^2}{\|d\|^2 - gd} & \text{se } gd < 0, \\ 1 & \text{altrimenti.} \end{cases}$$

Per il parametro  $\alpha'$  sono disponibili due scelte:

$$\alpha' := \alpha \tag{3.7g}$$

$$\begin{aligned} \nu_{sum} &:= \nu + \nu_{sum} \\ \alpha' &:= \frac{\nu}{\nu_{sum}}. \end{aligned} \tag{3.7h}$$

Per il calcolo della direzione aggregata abbiamo a disposizione due scelte:

$$\hat{d} := \alpha g + (1 - \alpha)d \tag{3.7i}$$

$$\hat{d} := \alpha g + (1 - \alpha)\hat{d} \tag{3.7j}$$

**Passo 5 - Stepsize** Per la scelta dello *stepsize* vi sono quattro possibili alternative:

$$\nu := \beta \frac{f(y) - f^*}{\|d\|^2}, \quad \beta \in (0, 1] \quad \text{Polyak stepsize} \quad (3.8a)$$

$$\nu_k = \frac{\eta}{\xi + \zeta\sqrt{k}}, \quad \eta, \xi, \zeta \text{ costanti}; \quad \begin{array}{l} \text{diminishing} \\ \text{stepsize} \end{array} \quad (3.8b)$$

$$\nu_k = \frac{\eta}{\xi + \zeta k}, \quad \eta, \xi, \zeta \text{ costanti}; \quad \begin{array}{l} \text{quadratically} \\ \text{convergent} \\ \text{stepsize} \end{array} \quad (3.8c)$$

$$\nu := \beta \frac{f(y) - f_{lev}}{\|d\|^2}, \quad \beta \in (0, 1] \quad \text{Target Value} \quad (3.8d)$$

(3.8b), ai fini dell'analisi di convergenza, può essere sostituito con qualunque stepsize che soddisfi:

$$\nu_k > 0, \quad \lim_{k \rightarrow \infty} \nu_k = 0, \quad \sum_{k=0}^{\infty} \nu_k = \infty. \quad (3.8e)$$

(3.8c), può essere sostituito con qualunque stepsize che soddisfi:

$$\nu_k > 0, \quad \lim_{k \rightarrow \infty} \nu_k = 0, \quad \sum_{k=0}^{\infty} \nu_k = \infty, \quad \sum_{k=0}^{\infty} \nu_k^2 < \infty \quad (3.8f)$$

La tabella 3.1 indica alcuni stepsize che soddisfano (3.8e). Ovviamente (3.8f)  $\Rightarrow$  (3.8e).

**passo 3 -  $f_{lev}$**  Il parametro  $\varphi$  può essere scelto tra:

$$\varphi \text{ costante ed aggiornato al passo 5} \quad \text{Kiwiel stepsize} \quad (3.9a)$$

$$\varphi \text{ costante.} \quad \text{Fumero stepsize} \quad (3.9b)$$

$$\varphi := \check{f} - f_{lev} - \bar{\varepsilon} \quad \text{Sherali stepsize} \quad (3.9c)$$

Se vale (3.2) si aggiorna  $f_{lev}$  in uno dei seguenti modi:

$$f_{lev} = \check{f} - \varphi \quad \text{Kiwiel stepsize} \quad (3.9d)$$

$$\begin{array}{l} f_{lev} = \xi LB + (1 - \xi)\check{f}, \quad LB \leq f^* \\ \text{se } r \geq r_2 \quad \beta = 2\beta. \end{array} \quad \text{Fumero stepsize} \quad (3.9e)$$

$r$  inizializzato a  $r_0 \in (0, 1)$ ,  $\bar{r}$  inizializzato a  $r_0 + 1$ ,

$\rho \in (0, \gamma]$ ,  $\eta \in (0, 1]$ ,  $\check{\varepsilon}$  costanti

$$f_{lev} := \bar{f} - \max\{\bar{\varepsilon} + \eta\Delta, r|\bar{f}|\}$$

$$\bar{\varepsilon} := \max\{(\bar{f} - f_{lev})\rho, \check{\varepsilon}\},$$

$$\text{se } f_{lev} = \bar{f} - r|\bar{f}| \text{ allora } r := \frac{r}{\bar{r}},$$

$$\Delta := 0.$$

Sherali stepsize (3.9f)

**passo 4 -  $f_{lev}$**  In questo passo si possono usare le seguenti *target unfeasibility condition*:

$$R \text{ costante, } \sigma > R \quad \text{Kiwiel stepsize} \quad (3.10a)$$

$$\begin{cases} \check{k} > K_1 & \text{se } r < r_2 \\ \check{k} > K_2 & \text{altrimenti} \end{cases} \quad \text{Fumero stepsize} \quad (3.10b)$$

$$K \text{ costante, } \check{k} > K, I \text{ costante } \bar{k} > I. \quad \text{Sherali stepsize} \quad (3.10c)$$

dove  $\check{k}$  indica il numero di iterazioni dall'ultimo aggiornamento di  $f_{lev}$  e  $\bar{k}$  il numero di iterazioni dall'ultimo aggiornamento di  $\bar{f}$ .

Se vale una di queste condizioni si aggiorna  $f_{lev}$  in uno dei seguenti modi:

$$\varphi := \frac{\varphi}{2}, f_{lev} := \check{f} - 2\varphi \quad \text{Kiwiel stepsize} \quad (3.10d)$$

$$\begin{aligned} & r = r + 1 \\ \beta &= \begin{cases} \frac{1}{\beta(\frac{1}{\beta}+2)} & \text{se } r > r_2 \\ \frac{1}{2}\beta & \text{altrimenti} \end{cases} \\ \xi &= \begin{cases} e^{-\gamma_1(\frac{r}{r_1})^{\gamma_2}} & \text{se } r < r_2 \\ \xi_{min} & \text{altrimenti} \end{cases} \quad \text{Fumero stepsize} \quad (3.10e) \\ f_{lev} &= \xi LB + (1 - \xi)\check{f}, \quad LB \leq f^* \\ & r < r_2 \vee \beta < \bar{\beta} \Rightarrow y_k = \bar{y}. \\ & r_2 = \left( -\frac{\log(\xi_{min})}{\gamma_1} \right)^{\frac{1}{\gamma_2}} r_1 \end{aligned}$$

$\bar{\beta}, \gamma_1, \gamma_2, \xi_{min}, r_1$  costanti  $r$  inizializzato a 0

$\rho \in (0, \gamma], \check{\epsilon}$  costanti

$$f_{lev} := \frac{(\bar{f} - \bar{\epsilon}) + f_{lev}}{2}$$

$$\bar{\epsilon} := \max\{(\bar{f} - f_{lev})\rho, \check{\epsilon}\}$$

$$\text{se } \bar{k} > I \text{ allora } I := \min\{I + \zeta, \eta\} \quad \text{Sherali stepsize} \quad (3.10f)$$

$$\text{se } (f_{lev} - f_{lev}^-) \leq \vartheta \text{ allora } \beta := \max\left\{\frac{\beta}{2}, \iota\right\}$$

$$\Delta := 0.$$

Per le regole (3.10f) e (3.9f) [6] suggerisce i seguenti valori:  $\gamma = \frac{1}{3}$ ,  $\zeta = 10$ ,  $\eta = 50$ ,  $\vartheta = 0.1$ ,  $\iota = 10^{-6}$ .

Per la regola (3.10e) [8] suggerisce i seguenti valori:  $\gamma_1 = 0.6933$ ,  $\gamma_2 = 3.26$ .

### 3.1 Analisi di convergenza

Nella precedente sezione abbiamo presentato uno schema algoritmico, parametrizzando alcune scelte, che generalizza i metodi discussi in precedenza e per ogni passo abbiamo evidenziato le possibili scelte. Ora elencheremo le scelte supportate da dimostrazioni di convergenza.

È importante far notare ancora una volta che tutti questi algoritmi non hanno terminazione finita, almeno da un punto di vista teorico. Si studia, quindi, la convergenza asintotica. Inoltre qui non si applica il classico modello computazionale per l'analisi degli algoritmi a terminazione finita: si interpreta il metodo mediante le successioni  $\{y_k\}$  e  $\{f_k\}$  che rappresentano, rispettivamente, la successione delle soluzioni generate ad ogni iterazione e i corrispondenti valori obiettivo.

Ovviamente ad ogni singola iterazione si può applicare l'analisi di efficienza del caso a terminazione finita: di ogni iterazione possiamo calcolare la complessità algoritmica. Questo tipo di analisi è molto importante: ad esempio se calcolare il valore di  $f(y)$  o un suo subgradiente è NP-Hard l'efficienza dell'algoritmo ne sarà condizionata; ad ogni modo il nostro intento è studiare e implementare algoritmi da un punto di vista *function independent*, pertanto non possiamo effettuare analisi di questo tipo.

In questo contesto diciamo che l'algoritmo converge se:

$$\lim_{k \rightarrow \infty} y_k = y^* \in Y^*, \quad \lim_{k \rightarrow \infty} f_k = f^*; \quad (3.11)$$

dove  $Y^* = \{y \in Y : f(y) = f^* = \min_{z \in Y} \{f(z)\}\}$  è l'insieme delle soluzioni ottime.

A volte si ha una forma di convergenza più debole:

$$\liminf_{k \rightarrow \infty} f_k = f^*, \quad \exists \{k_i\} : \lim_{i \rightarrow \infty} y_{k_i} = y^* \in Y^*. \quad (3.12)$$

L'utilità pratica dei metodi che hanno questo tipo di convergenza deriva dalla seguente proprietà, detta  $\{\bar{f}_k\}$  la successione dei *record point*:

$$\liminf_{k \rightarrow \infty} f_k = f^* \Leftrightarrow \lim_{k \rightarrow \infty} \bar{f}_k = f^*. \quad (3.13)$$

Di solito la convergenza nel senso di (3.12) si ha con gli stepsize (3.8b), cioè di tipo *diminishing*, o (3.8d), cioè di tipo *target value*.

Recensiamo ora le proprietà di convergenza del metodo non incrementale ( $h = 1$ ); in seguito tratteremo quelle del metodo incrementale ( $h > 1$ ).

**Conditional subgradient method** Le seguenti scelte, garantiscono la convergenza del metodo condizionale:

- (3.6a), (3.7a), (3.8b);
- (3.6a), (3.7a), (3.8a);
- (3.6a), (3.7a), (3.8c);
- (3.6a), (3.7a), (3.8d), (3.9a), (3.9d), (3.10a), (3.10d);
- (3.6a), (3.7a), (3.8d), (3.9c), (3.9f), (3.10c), (3.10f).

**Deflected subgradient method** I seguenti metodi sono coperti dalle dimostrazioni di convergenza del capitolo 2:

- (3.6a), (3.7b), (3.8a);

- (3.6a), (3.7c), (3.8a);
- (3.6a), (3.7f), (3.8a);
- (3.6a), (3.7b), (3.8d), (3.9a), (3.9d), (3.10a), (3.10d);
- (3.6a), (3.7c), (3.8d), (3.9a), (3.9d), (3.10a), (3.10d);
- (3.6a), (3.7b), (3.8c); con la condizione (2.34) o (2.47);
- (3.6a), (3.7c), (3.8c); con la condizione (2.34) o (2.47);

Se sostituiamo (3.3) con  $d = \hat{d}$  troviamo in letteratura dimostrazioni di convergenza per il seguente metodo non coperto dal caso precedente.

- (3.6a), (3.7b), (3.8d), (3.9c), (3.9f), (3.10c), (3.10f);

**Volume** Se, anche qui, sostituiamo (3.3) con  $d = \hat{d}$  troviamo dimostrazioni di convergenza per:

- (3.6c), (3.7d);
- (3.6d), (3.7e).

Ma nel primo caso deve valere la seguente condizione sullo stepsize:

$$\sum_{k \in L} \nu_k = +\infty, \quad \{\nu_k\} \text{ è limitata, } \exists \nu_{min} > 0 : \{\nu_k\}_{k \notin L} \rightarrow \nu_{min}; \quad (3.14)$$

e nel secondo caso:

$$\begin{aligned} \sum_{k \in L} \nu_k = +\infty, \quad \{\nu_k\}_{k \in L} \text{ è limitata,} \\ \{\nu_k\}_{k \notin L} \text{ è monotona non crescente, } \sum_{k \notin L} \frac{\nu_{k+1}^2}{\nu_k} = \infty. \end{aligned} \quad (3.15)$$

Dove, in entrambi i casi,  $L = \{k : k \text{ serious step}\}$ .

**Convergenza Primale** Per i seguenti metodi è garantita la convergenza della variabile  $x$  al valore ottimo primale:

- (3.6a), (3.7a), (3.8a), (3.7h);
- (3.6a), (3.7a), (3.8c), (3.7h);
- (3.6a), (3.7b), (3.8a), (3.7h);
- (3.6c), (3.7d), (3.7g);
- (3.6d), (3.7e), (3.7g).

Ci sono dei metodi, non supportati da dimostrazioni di convergenza teorica, di cui è stata verificata sperimentalmente la convergenza; ad esempio i cosiddetti metodi del volume standard, quelli che utilizzano (3.6b) e qualsiasi tipo di stepsize o di parametro  $\alpha$ .

**Metodo incrementale** Per quanto riguarda il metodo incrementale la convergenza è stata studiata soltanto per il metodo ordinario. Dunque convergono teoricamente i seguenti metodi se  $h = \infty$ :

- (3.6a), (3.7a), (3.8b)

- (3.6a), (3.7a), (3.8c);

I seguenti metodi sono trattati in [11]

- (3.6a), (3.7a), (3.8a);

- (3.6a), (3.7a), (3.8d), (3.9a), (3.9d), (3.10a), (3.10d);

Tuttavia l'analisi di convergenza per essi prevede il calcolo della funzione e di un subgradiente totale al termine di ogni ciclo incrementale senza iterazione non incrementale. Pertanto il nostro algoritmo non si può ricondurre ad essi; comunque indicativamente ci aspettiamo che per  $h = 2$  questi metodi convergano.

Non sono, invece, presenti in letteratura metodi del volume incrementali o del subgradiente deflesso. La nostra implementazione è, quindi, innovativa; tuttavia non è supportata da un'analisi teorica di convergenza.

## Capitolo 4

# Implementazione e test computazionali

### 4.1 Implementazione

In questo capitolo descriviamo l'implementazione dello studio teorico ai capitoli precedenti.

La classe `SubGrad`, che implementa i metodi del subgradiente e del volume discussi in questo documento, è parte di un *framework* C++ per la definizione di problemi e metodi risolutivi dell'ottimizzazione non differenziabile. Il sistema consiste inanzitutto in due interfacce: la prima (`FiOracle`) permette di definire un oracolo che implementa una funzione non differenziabile, la seconda (`NDOSolver`) regola la definizione di algoritmi risolutivi (*solver*).

L'interfaccia pubblica di `SubGrad` è quella di `NDOSolver`, tuttavia è presente in `SubGrad` una gerarchia di classi, ignorabile per gli utenti non sviluppatori, che modella i vari metodi risolutivi implementati.

Come visto nei precedenti capitoli un metodo di risoluzione è individuato da una tripla Metodo di Proiezione  $\times$  Deflection/Volume  $\times$  Step size. In base a questa considerazione `SubGrad` identifica l'algoritmo risolutivo come l'aggregazione di tre tipi: `Projection`, `VolumeKND` e `Stepsize`. Ad ognuno di questi tipi corrisponde un'omonima interfaccia (i.e. una classe astratta) e diverse implementazioni differenti. Si ottiene un metodo diverso per ogni diversa combinazione di sottotipi.

Le classi conformi all'interfaccia `FiOracle`/`NDOSolver` utilizzano un'apposita libreria per la gestione dei dati numerici definita in `OPTtypes.h` e `OPTvect.h` che fornisce le definizioni di tipi e varie operazioni su vettori; questa libreria permette, in aggiunta, di utilizzare vettori in formato sparso che registrano solo i valori non nulli.

Per un'ulteriore approfondimento del codice è possibile consultare la documentazione generata automaticamente dal tool `doxygen`.

Possiamo individuare tre classi di "utenti" di questo solver:

- L'utente che ha disposizione un oracolo conforme all'interfaccia `FiOracle` e la classe `SubGrad` può utilizzare il nostro codice per risolvere un problema di ottimizzazione non differenziabile. In questo caso l'unico accorgimento riguarda la scelta del metodo e la configurazione dei relativi parametri.

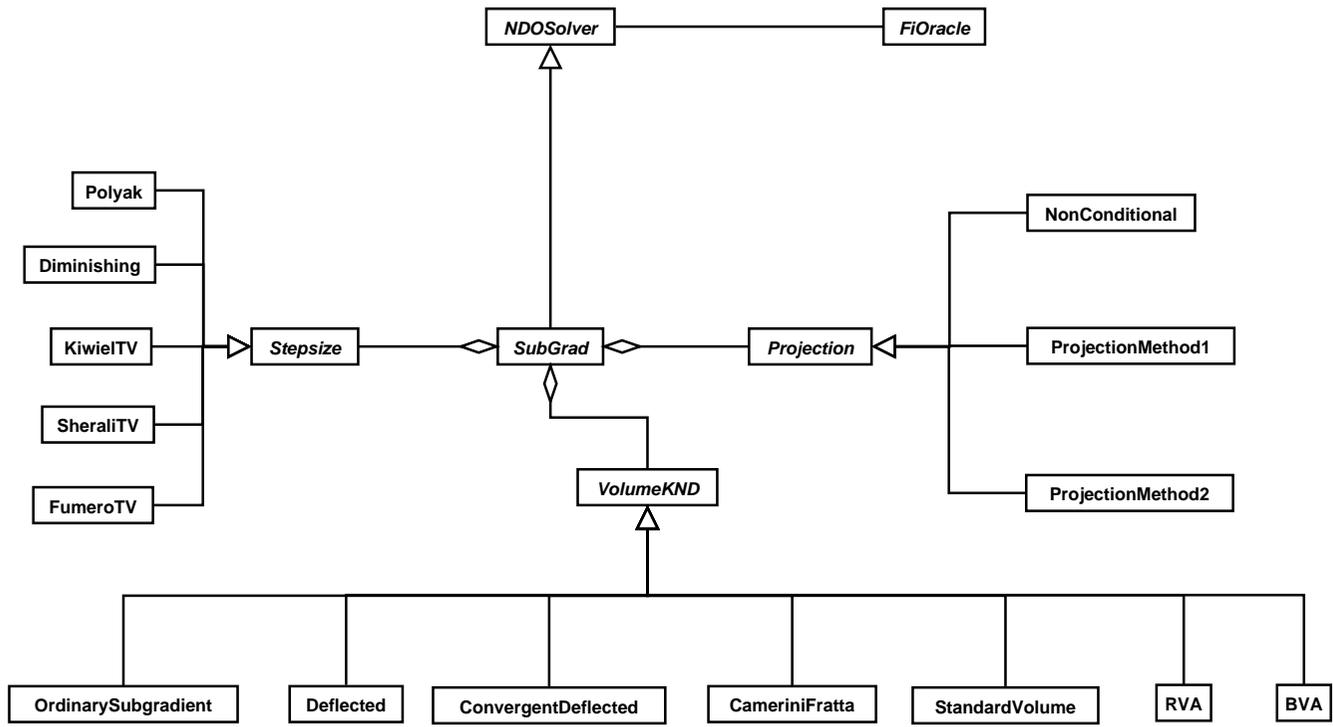


Figura 4.1: Diagramma delle classi del frammento di framework interessato.

- L'utente che vuole risolvere un problema di ottimizzazione non differenziabile per il quale non dispone di un oracolo già scritto può utilizzare gli algoritmi implementati in `SubGrad` e scrivere il proprio oracolo. Questo tipo di utente sarà interessato all'interfaccia di `NDOSolver` e alla sua implementazione in `SubGrad`.
- L'utente sviluppatore interessato ad implementare nuovi algoritmi del subgradiente o estendere quelli già implementati può sfruttare l'architettura modulare di `SubGrad` che facilita la comprensione e l'estensione del codice. Ad ogni modo queste operazioni richiedono la conoscenza e la comprensione del sistema di tipi che gli altri utenti possono ignorare.

La figura 4.1 mostra uno schema delle classi del frammento del framework interessato da questo progetto. Forniamo, quindi, diverse implementazioni per ogni interfaccia dell'architettura di `SubGrad`. Di seguito è presente una breve discussione di ognuna.

**Stepsize** L'interfaccia `Stepsize` modella le implementazioni delle diverse lunghezze del passo. Le classi da noi fornite implementano le regole (3.8a)-(3.8d) inoltre, come già visto nel capitolo 3, per (3.8d) sono possibili diverse scelte, ognuna delle quali caratterizza uno stepsize diverso. In particolare implementiamo tutte le scelte discusse nel capitolo 3.

**Polyak** Implementa (3.8a) dove  $\beta = \alpha$  ad ogni iterazione.

**Diminishing** Implementa (3.8c), non presenta particolari accorgimenti volti a migliorare l'efficienza. È bene notare come la scelta dei parametri  $\eta$ ,  $\xi$ ,  $\zeta$  sia particolarmente critica, dal punto di vista dell'efficienza, per questo tipo di stepsize.

**KiwielTV** Implementa (3.8d), (3.9a), (3.9d), (3.10a), (3.10d). Per questo tipo di stepsize si può notare come vicino all'ottimo l'algoritmo rallenti sensibilmente. Questo comportamento è principalmente dovuto al fatto che tanto più  $\|\hat{y}_{k+1} - y_k\| = \nu_k \|d_k\|$  è piccolo tanto meno frequentemente la condizione di aggiornamento (3.10a) è verificata. Per questo motivo la classe `KiwielTV` aggiorna  $R$  per un fattore  $R_{fact} \leq 1$  quando in  $R_{int}$  iterazioni non si ottiene un miglioramento giudicato sufficiente.

**SheraliTV** Implementa (3.8d), (3.9c), (3.9f), (3.10c), (3.10f). Non presenta particolari accorgimenti per migliorare l'efficienza; inoltre sembra essere poco sensibile, rispetto agli altri stepsize, alla scelta dei parametri.

**FumeroTV** Implementa (3.8d), (3.9b), (3.9e), (3.10b), (3.10e). Anche qui non sono presenti particolari accorgimenti; si nota, inoltre, come si ottenga un buon comportamento anche quando il parametro  $LB$  non è accurato.

**VolumeKND** L'interfaccia `VolumeKND` modella le strategie di *deflection* e le caratteristiche dei diversi algoritmi del volume. Le classi fornite implementano le scelte (3.6a)-(3.7f).

A volte è possibile che la direzione deflessa sia un subgradiente approssimato nell'iterata corrente e che il calcolo dell'approssimazione sia semplice (si veda ad esempio (2.19) o (A.8)). In questi casi i sottotipi di `VolumeKND` sono in grado di restituire il valore dell'approssimazione.

**OrdinarySubgradient** Implementa (3.6a), (3.7a). Si tratta del metodo discusso nella sezione 1.2.

**Deflected** Implementa (3.6a), (3.7c). Si noti come impostando  $\alpha_{fact}$  ad 1 si ottiene il metodo (3.6a), (3.7b). Si tratta, quindi, del metodo discusso nella sezione 1.5.

**ConvergentDeflected** Implementa il metodo 4 discusso nel capitolo 2. In particolare è equivalente a **Deflected** con la condizione (2.47). Quindi questa classe ha senso esclusivamente se utilizzata insieme a **Diminishing**.

**CameriniFratta** Implementa (3.6a), (3.7f). In particolare questo metodo è equivalente a (1.6) se si imposta  $\tau$  ad 1 (cfr. [18]).

**StandardVolume** Implementa (3.6b), (3.7c). Come già notato per **Deflected** si può ottenere (3.6b), (3.7b) impostando  $\alpha_{fact}$  ad 1.

**RVA** Implementa (3.6c), (3.7d).

**BVA** Implementa (3.6d), (3.7e).

**Projection** L'interfaccia **Projection** implementa gli aspetti relativi alla proiezione dei subgradienti e alla loro aggregazione.

**NonConditional** Non proietta le direzioni (ovvero sostituisce (3.3) con  $d = \hat{d}$ ) e dunque aggrega direzioni non proiettate.

**ProjectionMethod1** Implementa (3.3) e (3.7i)

**ProjectionMethod2** Implementa (3.3) e (3.7j)

Si noti come, nonostante il calcolo della direzione aggregata sia effettuato dai sottotipi di **VolumeKND** la scelta di quale direzione aggregare è demandata ai sottotipi di **Projection**.

La scelta del metodo di aggregazione primale (ovvero (3.7g)-(3.7h)) è implementata da **SubGrad**. In particolare **SubGrad** mette a disposizione la funzione

```
void SetPrimalMethod ( BOOL PrimalMethod );
```

dove  $\text{PrimalMethod} = \text{FALSE} \Rightarrow (3.7h)$  e  $\text{PrimalMethod} = \text{TRUE} \Rightarrow (3.7g)$

Infine **SubGrad** implementa lo schema algoritmico al capitolo 3; ovvero tutti gli aspetti non considerati dalle altre classi.

Inoltre **SubGrad** permette di utilizzare subgradienti in formato sparso. In questo caso si registrano in memoria solo le componenti non nulle del subgradiente e delle direzioni; la gestione di questo formato è in parte supportata dalla libreria **OPTvect** e in parte implementata in **SubGrad.h**. **SubGrad** fornisce la funzione

```
void SubGrad::SetSparse ( BOOL sprs );
```

dove  $\text{sprs} == \text{TRUE}$  imposta il solver per utilizzare il formato sparso. Sia la gestione dei subgradienti sparsi che il metodo di aggregazione primale possono essere impostati tramite file di configurazione.

## 4.2 Test computazionali

Gli algoritmi sono stati testati su un problema di ottimizzazione sul politopo semimetrico (cfr. [23]). Si tratta di un cosiddetto *large scale linear problem* e, pertanto, quello lagrangiano è un rilassamento “improprio” nel senso che fornisce il valore ottimo esatto. Nel seguito descriviamo brevemente il problema, le istanze utilizzate e i risultati.

### 4.2.1 Il politopo semimetrico

Dato un grafo *loopless*, non orientato, completo  $G = (V, E)$  chiamiamo politopo semimetrico (indicato con  $\mathcal{M}(G)$ ) il politopo definito dalle seguenti disequazioni:

$$\left. \begin{array}{l} x_{ij} + x_{ik} + x_{jk} \leq 2 \\ x_{ij} - x_{ik} - x_{jk} \leq 0 \\ -x_{ij} + x_{ik} - x_{jk} \leq 0 \\ -x_{ij} - x_{ik} + x_{jk} \leq 0 \end{array} \right\} \text{ per ogni } i, j, k \in V \text{ distinti.} \quad (4.1)$$

Queste sono anche dette *triangle inequalities*. Il politopo semimetrico può anche essere definito su grafi non completi e, in questo caso, è sufficiente completare il grafo con archi di costo nullo per ottenere un problema di ottimizzazione (lineare) equivalente.

Dunque siamo interessati a risolvere il problema

$$\max\{cx : x \in \mathcal{M}(G)\}; \quad (4.2)$$

questo problema è un comune rilassamento per i problemi di *max-cut*, ovvero il problema di ottimizzazione dei tagli su di un grafo.

Poiché (4.2) è un problema lineare potrebbe essere risolto tramite algoritmi per la programmazione lineare quali quello del *simplex* o il metodo *interior points*. Tuttavia (cfr. [23]) per questo problema gli algoritmi “standard” sembrano alquanto inefficienti.

Frangioni et al. in [23] introducono un rilassamento di (4.2). Selezionato un nodo radice  $r$  consideriamo le disequazioni relative a tutte le triple  $(r, i, j)$   $i, j \in V$  (indicate con  $A^r x \leq b^r$ ); queste definiscono il *rooted semimetric polytope*, indicato con  $\mathcal{M}^r(G)$ . Il problema

$$\max\{cx : A^r x \leq b^r\} \quad (4.3)$$

può essere risolto agevolmente tramite riformulazione come problema di flusso su reti.

Dunque, dato che (4.2) è un problema lineare, il rilassamento lagrangiano

$$\min_{y \geq 0} \left\{ \max_x \left\{ cx - \sum_{h \neq r} y^h (b^h - A^h x) : A^r x \leq b^r \right\} \right\} \quad (4.4)$$

è equivalente duale lineare di (4.2).

Un altro approccio possibile è costituito dalla decomposizione lagrangiana rispetto a tutti i blocchi  $A^r x \leq b^r : r = 1, \dots, n$ . Consideriamo, invece, un approccio *ibrido* che comprende il rilassamento e la decomposizione lagrangiani

come casi speciali. Definiamo un insieme di nodi radice  $R = \{r_1, r_2, \dots, r_k\} \subset V$  e consideriamo il problema (dove  $\bar{c} = (\frac{c}{k})$ )

$$\begin{aligned} & \max_{x^r} \bar{c} \sum_{r \in R} x^r \\ & A^r x^r \leq b^r \quad r \in R \\ & x^{r_i} = x^{r_{i+1}} \quad i = 1, \dots, k-1 \\ & A^q \left( \frac{1}{k} \sum_{r \in R} x^r \right) \leq b^q \quad q \notin R \end{aligned} \tag{4.5}$$

e il suo duale lagrangiano rispetto ai vincoli di uguaglianza e ai blocchi corrispondenti ai nodi non in  $R$

$$\begin{aligned} & \min_{\pi, y} \left\{ \max_{x^{r_1}} \left\{ (\bar{c} - \pi^1) x^{r_1} - \frac{1}{k} \sum_{q \notin R} y^q (b^q - A^q x^{r_1}) : A^{r_1} x^{r_1} \leq b^{r_1} \right\} + \right. \\ & \sum_{i=2}^{k-1} \max_{x^{r_i}} \left\{ (\bar{c} - \pi^{r_i} + \pi^{r_{i-1}}) x^{r_i} - \frac{1}{k} \sum_{q \notin R} y^q (b^q - A^q x^{r_i}) : A^{r_i} x^{r_i} \leq b^{r_i} \right\} + \\ & \left. \max_{x^{r_k}} \left\{ (\bar{c} + \pi^k) x^{r_k} - \frac{1}{k} \sum_{q \notin R} y^q (b^q - A^q x^{r_k}) : A^{r_k} x^{r_k} \leq b^{r_k} \right\} : y \geq 0 \right\} \end{aligned} \tag{4.6}$$

Questo problema è ancora equivalente al duale lineare di (4.2) e se  $|R| = 1$  è esattamente (4.4), mentre se  $|R| = n - 1$  è la decomposizione lagrangiana. Inoltre per  $|R| > 1$  la funzione è decomponibile e si può applicare un metodo del subgradiente incrementale. Sfortunatamente, come mostra [23], il tempo necessario per risolvere (4.6) cresce drammaticamente al crescere di  $|R|$  e il miglior valore di questo parametro risulta essere  $|R| = 1$ .

Il problema di ottimizzazione sul politopo semimetrico è implementato nella classe **FiSemiMetric**, sottotipo di **FiOracle**. In particolare, dal momento che anche un numero ragionevole di nodi genera un numero elevato di vincoli, e quindi di variabili duali, **FiSemiMetric** utilizza un approccio di generazione dinamica delle variabili. L'algoritmo viene eseguito, ad ogni iterazione, su di un sottoinsieme delle variabili corrispondenti ai vincoli di disequaglianza. Sia l'interfaccia **NDOSolver** che il solver **SubGrad** forniscono supporto per l'inserimento e la rimozione di variabili.

Quando **SubGrad** termina fornisce una soluzione primale, generalmente non ammissibile (cfr. sez. 1.7). **FiSemiMetric** implementa un'euristica basata su proiezioni sull'insieme ammissibile per produrre soluzioni primali; i metodi che convergono da un punto di vista primale generalmente producono una soluzione molto vicina all'insieme ammissibile. In questi casi l'euristica produce una "buona" soluzione primale. In particolare questo metodo fornisce un valore obiettivo primale che utilizziamo per valutare l'efficienza primale degli algoritmi.

I test computazionali sono stati effettuati su cinque classi di grafi:

- cliques graphs,

- grafi planari,
- simplex graphs,
- toroidal 2D graphs,
- toroidal 3D graphs.

In particolare con *toroidal graph* intendiamo delle griglie (rispettivamente di 2 e 3 dimensioni) dove l'ultimo nodo di ogni "linea" è adiacente al primo. I grafi sono stati generati casualmente e sono gli stessi utilizzati in [23].

Nell'attività di *testing* si pone il problema di determinare per ogni metodo e per ogni istanza dei valori adeguati dei parametri. I metodi implementati in **SubGrad** vengono confrontati con altri metodi già implementati (i.e. la classe **Bundle**, sottotipo di **NDOSolver**) che sono "stabili" sui parametri; ovvero la cui efficienza non è sensibilmente influenzata da variazioni nei parametri. Per questo motivo abbiamo cercato di tenere il più possibile stabile la configurazione dei parametri. In particolare i parametri delle strategie di *deflection* sono tenuti costanti su tutti i *run*, i parametri degli stepsize *diminishing* sono tenuti costanti su tutte le istanze della stessa classe e della stessa dimensione, quelli del metodo target value di *Kiwiel* sono costanti su tutte le classi e assumono valore diverso (costante) sui grafi toroidali, tutti gli altri stepsize hanno parametri di valore costante.

#### 4.2.2 Risultati numerici

Le tabelle da 4.1 a 4.17 mostrano i risultati numerici, dove 0 indica il metodo non condizionale 1, il metodo implementato in **ProjectionMethod1** ovvero la proiezione che aggrega le direzioni proiettate e 2 il metodo implementato in **ProjectionMethod2** ovvero la proiezione che aggrega direzioni non proiettate, *dgap* e *pgap* indicano il gap duale e primale, ovvero detti  $f_{ub}$  e  $f_{lb}$  i valori obiettivo duale e primale prodotti dall'algoritmo e  $f^*$  il valore ottimo:

$$dgap = \frac{f_{ub} - f^*}{|f^*|}, \quad pgap = \frac{f^* - f_{lb}}{|f_{lb}|}.$$

Inoltre per ogni dimensione e classe di grafo sono state testate più istanze (15 per i grafi planari e 5 per gli altri tipi di grafo) e vengono riportate le medie dei gap ottenuti.

I metodi del subgradiente ordinario e del volume standard risultano essere dominati, su quasi tutte le istanze, dai rispettivi metodi più sofisticati, vale a dire i metodi deflessi e i metodi del volume revisitati; pertanto non ne riportiamo i risultati. Questo d'altronde non stupisce: le strategie di *deflection* e i metodi del volume revisitati sono stati sviluppati proprio per ovviare ad un comportamento di lenta convergenza del subgradiente e del volume. Inoltre riportiamo qui soltanto i metodi implementabili; quindi non testiamo il metodo di Polyak che ha bisogno della conoscenza a priori del valore ottimo obiettivo.

Nel valutare le prestazioni di ogni metodo teniamo conto di quattro fattori:

- convergenza duale
- convergenza primale

- efficienza
- robustezza;

dove valutiamo i primi due come l'accuratezza delle soluzioni rispettivamente duale e primale, il terzo come la misura del tempo necessario per l'esecuzione e il quarto come la stabilità dei parametri di configurazione. Quest'ultimo fattore in particolare non è evidente nelle tabelle qui riportate; in generale possiamo affermare che le strategie di deflection sono stabili, infatti non c'è stato bisogno di adeguare i parametri alle istanze; gli stepsize più stabili invece sono quelli di tipo target value. Si noti come, a causa della condizione di terminazione (1.28) dual gap, primal gap e tempo non sono stime assolute. Ad esempio nel valutare i metodi di proiezione su metodi della stessa classe può capitare che (1.28) forzi un metodo a terminare prima di un'altro, ottenendo così gap poco accurati ma tempo migliore. In altre parole se confrontando vari metodi il tempo presenta una correlazione con uno tra dual gap e primal gap allora i metodi si possono dire confrontabili. Possiamo tuttavia affermare che alcuni metodi sono, secondo i nostri test, quasi completamente dominati da altri. In particolare, in tutte le strategie di deflection, il metodo target value di Kiwiel è dominato dal metodo di Sherali. Anche questo non stupisce: il metodo di Sherali, oltre ad essere più recente, è più sofisticato, in particolare calcola il livello secondo la "storia" nello spazio obiettivo anziché in quello delle soluzioni. Anche lo stepsize di Fumero ottiene dei buoni risultati, specialmente se utilizzato in congiunzione con la strategia di deflection di Camerini-Fratta (ovvero se si esegue l'algoritmo descritto in [8]). Gli stepsize square summable/diminishing ottengono quasi sempre dei buoni risultati, dimostrandosi una scelta affidabile, ma soffrono di instabilità sulla scelta dei parametri. In particolare nei nostri test è stato necessario adeguare i parametri per questo tipo di stepsize ad ogni coppia (classe di grafo, dimensione) e, probabilmente, modulando i parametri sulla singola istanza si sarebbero ottenuti risultati più soddisfacenti.

Complessivamente si può dire che i metodi che hanno registrato prestazioni migliori sono quello di Camerini-Fratta, principalmente con stepsize di Fumero o di Sherali, e BVA con stepsize diminishing. In particolare questi due sono i metodi che appaiono con maggiore frequenza nella frontiera efficiente di ogni istanza, ovvero nell'insieme dei metodi non dominati da altri sulla stessa istanza. Si noti come i metodi migliori siano due, uno del subgradiente, uno del volume. Questo non è affatto strano: tendenzialmente i metodi del volume sono più accurati da un punto di vista primale e leggermente meno da un punto di vista duale; d'altro canto metodi come quello di Camerini-Fratta non producono soluzioni primali soddisfacenti (ad eccezione del caso in cui viene individuata una soluzione 0-ottima). D'altra parte alcuni metodi, pur avendo prestazioni leggermente peggiori di BVA e Camerini-Fratta, mostrano dei comportamenti interessanti. In particolare il metodo deflesso, che ottiene migliori prestazioni con gli stepsize di Sherali e diminishing e prestazioni accettabili con gli stepsize di Fumero e Kiwiel, è meno accurato di Camerini-Fratta ma produce delle soluzioni primali accettabili, anche se non al livello dei metodi del volume. Questo è in qualche modo sorprendente: i metodi del volume, pur non godendo tutti di proprietà teoriche di convergenza, hanno alla base una teoria che giustifica informalmente il loro comportamento (cfr. [5]); per quanto riguarda il metodo deflesso durante lo studio teorico abbiamo congetturato che, in analogia con il volume, tener

traccia di una soluzione primale aggregata potesse portare a qualche miglioramento nella velocità di convergenza primale: i risultati numerici sembrano supportare questa ipotesi. In sostanza BVA e Camerini-Fratta sembrano essere i metodi più efficienti ma entrambi soffrono di qualche limitazione. Il primo richiede uno stepsize diminishing e, quindi, dipende fortemente dai parametri di configurazione; il secondo non produce soluzioni primali accettabili. Il metodo deflesso sembra essere un buon compromesso tra le due esigenze: produce una soluzione duale abbastanza accurata e una soluzione primale decisamente migliore di Camerini-Fratta anche se peggiore di BVA.

Nella tabella 4.16 riportiamo i risultati numerici per il metodo convergent deflected (metodo 4 nel capitolo 2) con stepsize diminishing. Si può notare come le prestazioni siano leggermente inferiori rispetto al deflesso non convergente teoricamente. Neanche questo stupisce: solitamente un metodo che converge teoricamente perde qualcosa da un punto di vista computazionale; in effetti il tentativo di ridurre questo gap è uno dei *leit-motiv* dell'attività di ricerca in questo campo. Possiamo riscontrare questo fenomeno anche in relazione agli altri metodi; di quelli che abbiamo implementato alcuni sono convergenti teoricamente, altri convergono secondo delle ipotesi non implementabili (è questo il caso, tra gli altri, di Camerini-Fratta e BVA), altri ancora non convergono teoricamente ma ne riscontriamo l'efficienza computazionalmente, mentre di altri riscontriamo empiricamente i fenomeni di non convergenza; in particolare rva con gli stepsize target value risulta essere un metodo non convergente. Questi ultimi metodi possono ancora essere utili. Utilizzando la condizione di terminazione (1.28) essi terminano velocemente producendo una soluzione decisamente poco accurata ma comunque non troppo lontano dal valore ottimo obiettivo (solitamente registriamo un gap duale dell'ordine di  $10^{-1}$ ). Quindi si potrebbe utilizzare un metodo non convergente per ottenere velocemente soluzioni poco accurate; tuttavia, a differenza degli altri metodi, questi non sempre producono una migliore soluzione duale al costo di un maggior numero di iterazioni.

Nel capitolo 2 abbiamo dimostrato la convergenza di alcuni metodi deflessi, in alcuni casi estendendo le analisi di convergenza già note ai metodi condizionali da noi sviluppati, in altri casi proponendo un'analisi di convergenza innovativa e in altri ancora proponendo dei metodi convergenti innovativi. Rimangono, comunque, dei metodi che convergono teoricamente nel caso deflesso non condizionale non trattati nel capitolo 2 (ad esempio lo stepsize di sherali); i risultati computazionali suggeriscono come ci aspettavamo l'ipotesi di convergenza, da un punto di vista empirico, di questi metodi.

Nella tabella 4.17 presentiamo i risultati ottenuti con un metodo di tipo Bundle. Possiamo notare come il metodo Bundle risulti molto più accurato dei metodi del subgradiente ma molto meno efficiente. In particolare il tempo necessario per la terminazione è quasi sempre notevolmente superiore a quello impiegato dai metodi del subgradiente. Fanno eccezione alcuni metodi non convergenti sui grafi di tipo cliques, che sembrano essere i più difficili per entrambe le classi di algoritmi. Si nota anche come le differenze tra i metodi Bundle e quelli del subgradiente si assottiglino sulle istanze di piccola dimensione; in questo caso entrambi gli algoritmi risultano accurati ed efficienti allo stesso tempo. D'altra parte anche l'approccio lineare in questi casi sembra essere efficiente.

c150		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	6.17e-04	2.33e-02	150.5	3.76e-04	1.49e-02	127.4	5.19e-04	2.42e-02	162.4
	fumerotv	5.13e-01	3.91e-01	145.8	7.32e-04	9.71e-02	239.2	8.20e-01	3.86e-01	154.5
	kiwieltv	4.07e-02	3.51e-01	674	3.12e-02	3.36e-01	706.4	3.78e-02	3.44e-01	619.4
	sheralitv	7.23e-03	2.22e-01	91.34	1.42e-02	2.31e+00	60.85	3.89e-03	3.25e-01	108.3
camerinifratta	diminishing	5.20e-03	1.72e+00	81.44	4.88e-03	1.59e+00	91.9	5.24e-03	1.44e+00	92.22
	fumerotv	5.06e-05	3.93e-01	56.55	1.96e-04	1.02e+00	58.48	8.62e-02	4.65e-01	36.53
	kiwieltv	5.24e-03	1.44e+00	92.22	5.20e-03	8.72e-01	166.2	9.01e-03	9.58e-01	201.2
	sheralitv	5.25e-05	7.80e-01	82.74	7.80e-05	1.15e+00	79.44	2.83e-02	7.79e-01	57.46
deflected	diminishing	1.95e-03	7.00e-02	112.3	1.93e-03	8.06e-02	102.4	3.80e-03	2.43e-01	94.52
	fumerotv	2.04e-03	2.39e-01	68.28	1.83e-03	2.42e-01	70.15	1.84e-03	4.10e-01	67.15
	kiwieltv	3.77e-02	2.89e-01	120.2	3.16e-02	3.22e-01	151.3	2.26e-02	1.88e-01	226.7
	sheralitv	7.68e-04	8.30e-02	115	3.96e-04	5.29e-02	131.6	8.71e-04	6.91e-02	114
rva	diminishing	2.74e-03	9.75e-02	316.4	4.63e-03	1.06e-01	140	1.06e-02	2.69e-01	189.5
	fumerotv	4.34e-01	5.81e-01	20	1.53e-01	4.62e-01	71.37	6.40e-01	5.83e-01	28.01
	kiwieltv	2.99e-01	6.54e-01	368.8	2.72e-01	7.18e-01	473.6	2.88e-01	7.09e-01	608.3
	sheralitv	6.44e-01	1.27e+00	17.18	5.90e-01	1.16e+00	18.66	5.90e-01	1.40e+00	18.71
c125		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	6.72e-04	2.11e-02	97.36	6.41e-04	2.01e-02	97.49	6.85e-04	2.09e-02	104.4
	fumerotv	4.99e-01	5.83e-01	131.9	7.10e-04	7.07e-01	201.3	6.48e-01	6.44e-01	126.9
	kiwieltv	1.37e-02	2.64e-01	325.5	9.48e-03	1.91e-01	428.6	8.12e-03	1.48e-01	444.6
	sheralitv	4.97e-03	7.20e-01	58.16	3.12e-04	9.85e-02	53.92	5.90e-03	1.03e+00	68.56
camerinifratta	diminishing	4.46e-03	2.51e+00	65.88	4.46e-03	1.86e+00	75.87	3.44e-03	1.51e+00	68.9
	fumerotv	6.34e-05	8.53e-01	34.79	4.55e-04	8.81e-01	35.65	5.67e-02	7.00e-01	28.19
	kiwieltv	3.44e-03	1.51e+00	68.9	5.62e-03	8.29e-01	112.8	5.56e-03	8.42e-01	193.4
	sheralitv	5.10e-05	2.72e+00	49.76	4.19e-05	1.23e+00	50.84	2.88e-02	1.12e+00	31.37
deflected	diminishing	3.36e-03	8.90e-02	89.8	2.53e-03	6.38e-02	86.77	2.73e-03	7.72e-02	84.39
	fumerotv	1.44e-03	1.67e-01	40.15	1.91e-03	2.50e-01	41.75	1.10e-03	4.62e-01	41.72
	kiwieltv	8.79e-03	1.95e-01	111.4	6.89e-03	1.42e-01	117.2	2.04e-02	1.89e-01	101.6
	sheralitv	1.56e-03	3.14e-01	58.66	8.98e-04	1.30e-01	73.57	8.92e-04	2.67e-01	65.37
rva	diminishing	1.68e-03	5.45e-02	226	1.09e-02	1.79e-01	173.7	1.33e-03	4.86e-02	207.3
	fumerotv	7.79e-01	6.95e-01	12.73	4.85e-01	5.16e-01	26.18	2.13e-01	5.04e-01	29.73
	kiwieltv	2.89e-01	9.59e-01	91.23	2.88e-01	7.28e-01	158.5	2.93e-01	7.04e-01	93.33
	sheralitv	6.68e-01	1.14e+00	11.53	6.11e-01	1.27e+00	12.8	6.14e-01	9.88e-01	12.87

Tabella 4.1: Risultati numerici c150 e c125

c100		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	8.72e-05	2.95e-03	70.91	1.43e-04	3.11e-03	64.44	9.76e-05	2.86e-03	89.23
	fumero	8.35e-01	6.29e-01	55.77	8.47e-04	8.26e-02	108	7.10e-01	6.72e-01	55.15
	kiwiel	2.07e-03	5.10e-02	286.3	7.43e-03	1.30e-01	179.6	4.33e-03	1.11e-01	162.6
	sherali	3.70e-03	1.66e-01	45.85	2.96e-04	7.91e-02	38.52	8.07e-03	8.78e-01	20.56
camerinfiratta	diminishing	7.77e-03	6.13e-01	21.94	7.84e-03	9.34e-01	25.99	7.49e-03	1.78e+00	24.42
	fumero	7.02e-05	1.05e+00	19.55	5.69e-04	5.83e-01	20.97	8.03e-02	6.43e-01	18.26
	kiwiel	7.49e-03	1.78e+00	24.42	3.76e-03	5.83e-01	111.1	6.21e-03	1.17e+00	57.31
	sherali	2.51e-05	7.15e-01	32.96	2.39e-05	4.41e-01	34.55	5.30e-05	1.81e+00	33.65
deflected	diminishing	3.51e-03	8.93e-02	33.2	3.51e-03	6.75e-02	36.32	3.32e-03	9.51e-02	34.02
	fumero	2.01e-03	1.93e-01	33.04	1.91e-03	2.88e-01	31.4	1.37e-03	1.84e-01	30.97
	kiwiel	6.21e-03	1.20e-01	56.92	7.09e-03	1.41e-01	58.18	1.28e-02	1.78e-01	68.44
	sherali	1.30e-03	5.55e-02	38.57	1.39e-03	7.63e-02	40.22	1.18e-03	1.95e-01	98.35
rva	diminishing	8.56e-03	1.39e-01	91.36	2.16e-04	9.16e-03	215.1	1.99e-04	8.83e-03	170.5
	fumero	3.74e-01	5.86e-01	7.202	4.70e-01	4.72e-01	10.61	2.44e-01	5.81e-01	7.646
	kiwiel	1.77e-01	5.46e-01	54.9	2.06e-01	1.03e+00	60.62	2.13e-01	1.04e+00	33
	sherali	6.33e-01	1.31e+00	6.394	5.78e-01	1.19e+00	7.038	5.79e-01	8.86e-01	6.98
c75		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	2.21e-04	9.22e-03	12.98	2.55e-04	9.85e-03	14.63	2.00e-04	9.54e-03	13.54
	fumerotv	9.17e-01	2.30e-01	21.33	1.12e-01	4.79e-01	49.32	4.28e-01	4.83e-01	30.05
	kiwieltv	8.91e-03	9.72e-02	70.29	7.62e-03	1.75e-01	38.12	9.97e-03	1.55e-01	34.52
	sherality	2.28e-03	1.62e-01	12.36	2.56e-04	8.09e-02	12.95	5.38e-03	4.74e-01	13.44
camerinfiratta	diminishing	1.32e-02	1.44e+00	11.23	1.35e-02	8.98e-01	11.04	1.21e-02	2.32e+00	11.7
	fumerotv	3.49e-04	8.67e-01	8.938	1.21e-01	8.67e-01	8.214	9.57e-02	5.39e+00	7.442
	kiwieltv	5.29e-03	1.02e+00	16.41	4.61e-03	9.76e-01	14.76	6.56e-03	9.49e-01	14.14
	sherality	5.46e-05	1.05e+00	12.51	9.68e-05	1.82e+00	13.37	5.64e-02	6.55e-01	9.216
deflected	diminishing	3.74e-03	7.31e-02	15.7	7.68e-03	2.14e-01	14.09	3.86e-03	7.48e-02	17.25
	fumerotv	1.48e-03	3.33e-01	10.57	2.30e-03	2.09e-01	11.59	1.15e-03	2.66e-01	11.49
	kiwieltv	7.68e-03	1.30e-01	21.02	7.14e-03	1.26e-01	21.37	4.73e-03	1.00e-01	21.93
	sherality	1.32e-03	2.66e-01	16.79	1.20e-03	1.11e-01	18.26	1.60e-03	1.72e-01	17.42
rva	diminishing	9.74e-04	3.38e-02	36.74	1.15e-03	2.88e-02	48.94	3.07e-01	1.96e-01	40.13
	fumerotv	1.59e-01	2.99e-01	10.31	5.44e-01	4.99e-01	5.402	4.16e-01	6.43e-01	4.552
	kiwieltv	1.08e-01	8.17e-01	15.13	1.31e-01	4.75e-01	19.72	7.24e-02	1.02e+00	10.01
	sherality	6.16e-01	1.13e+00	3.316	5.73e-01	1.16e+00	3.628	5.75e-01	1.48e+00	3.738

Tabella 4.2: Risultati numerici c100 e c75

c50		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	2.09e-04	9.33e-03	4.936	3.36e-04	8.57e-03	5.442	3.69e-04	8.86e-03	4.9
	fumerotv	8.93e-01	3.75e-01	3.184	2.90e-01	3.02e+00	20.33	1.61e-01	3.00e+00	5.922
	kiwieltv	4.08e-03	4.13e-02	4.826	8.97e-03	7.14e-02	4.34	7.41e-03	1.04e-01	5.386
	sheralitv	4.10e-03	2.83e-01	4.58	2.25e-04	5.50e-02	4.788	3.49e-03	1.88e-01	4.45
camerinifratta	diminishing	1.23e-02	6.04e-01	4.538	1.45e-02	6.26e-01	4.108	1.33e-02	1.62e+00	4.534
	fumerotv	2.95e-04	4.84e-01	3.36	4.32e-04	7.91e-01	3.506	3.47e-04	7.69e-01	3.422
	kiwieltv	4.96e-03	6.46e-01	6.634	6.98e-03	7.69e-01	6.216	5.90e-03	5.41e-01	5.364
	sheralitv	1.71e-04	8.43e-01	4.77	2.35e-04	9.55e-01	5.14	1.11e-04	1.21e+00	4.902
deflected	diminishing	7.97e-03	2.12e-01	5.578	4.00e-03	7.28e-02	6.708	4.31e-03	6.06e-02	6.454
	fumerotv	2.83e-03	2.24e-01	3.898	4.02e-03	2.81e-01	4.046	2.21e-03	1.94e-01	4.5
	kiwieltv	7.90e-03	1.04e-01	6.632	8.76e-03	1.94e-01	6.04	6.34e-03	8.84e-02	7.54
	sheralitv	2.51e-03	1.52e-01	5.14	2.90e-03	1.54e-01	4.932	2.42e-03	1.04e-01	5.486
rva	diminishing	8.62e-04	2.94e-02	12.57	1.99e-02	6.40e-02	18.87	8.81e-02	1.23e-01	8.27
	fumerotv	1.77e-01	6.13e-01	1.544	4.96e-01	2.97e+00	3.924	8.38e-02	2.75e+00	3.908
	kiwieltv	3.05e-02	9.88e-01	5.734	7.35e-02	4.72e-01	5.588	2.67e-02	8.19e-01	7.776
	sheralitv	6.29e-01	8.88e-01	1.34	5.72e-01	1.21e+00	1.554	5.69e-01	9.53e-01	1.48
c25		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	7.29e-05	2.64e-03	0.828	8.19e-05	2.67e-03	0.828	1.34e-04	3.02e-03	0.786
	fumerotv	5.19e-01	1.01e-01	0.454	5.35e-01	1.57e+00	2.316	4.00e-01	1.50e+00	0.308
	kiwieltv	1.01e-03	1.48e-02	0.68	2.75e-04	2.29e-03	0.682	9.32e-04	3.91e-03	0.674
	sheralitv	5.75e-03	4.24e-01	0.628	3.31e-03	2.15e+00	0.612	3.83e-03	1.61e-01	0.608
camerinifratta	diminishing	1.21e-02	3.31e-01	1.12	1.24e-02	5.16e-01	1.138	9.05e-03	4.14e-01	1.064
	fumerotv	4.52e-04	9.47e-01	0.778	2.14e-03	4.32e-01	0.936	2.63e-03	2.90e-01	0.828
	kiwieltv	4.29e-03	6.98e-01	3.962	6.71e-03	4.34e-01	2.34	3.03e-03	4.11e-01	3.192
	sheralitv	1.19e-03	4.24e-01	0.888	1.38e-03	5.40e-01	0.888	3.12e-02	3.72e-01	0.69
deflected	diminishing	3.30e-03	9.34e-02	0.946	1.04e-02	8.64e-01	1.022	5.80e-03	1.40e-01	0.89
	fumerotv	3.52e-03	1.15e-01	0.982	1.46e-03	1.31e-01	0.924	2.10e-03	9.52e-02	0.948
	kiwieltv	6.11e-03	5.72e-02	1.24	7.22e-03	5.44e-02	1.372	5.75e-03	6.80e-02	1.242
	sheralitv	3.81e-03	7.56e-02	1.062	2.23e-03	5.33e-02	1.322	2.54e-03	3.71e-02	1.144
rva	diminishing	2.88e-01	1.97e-01	3.442	3.23e-02	1.90e-01	2.938	4.39e-01	2.17e-01	1.856
	fumerotv	3.96e-01	6.47e-01	0.32	7.31e-02	1.84e+00	0.522	2.22e-01	2.12e+00	0.73
	kiwieltv	4.75e-01	6.59e-01	0.582	4.30e-01	4.95e-01	0.72	4.27e-01	6.11e-01	0.886
	sheralitv	4.86e-01	1.11e+00	0.332	4.57e-01	1.34e+00	0.384	4.51e-01	1.22e+00	0.384

Tabella 4.3: Risultati numerici c50 e c25

g2g144		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	1.34e-01	1.49e+00	58.52	1.13e-01	9.56e-01	81.21	9.10e-02	9.25e-01	99.92
	fumerotv	1.34e-02	4.90e-01	39.36	1.65e-01	1.89e+00	19.91	1.32e-01	1.91e+00	55.85
	kiwieltv	2.84e-02	9.37e-01	21.86	1.30e-02	7.53e-01	42.94	4.03e-02	1.39e+00	21.39
	sheralitv	1.67e-01	2.61e+00	11.48	1.66e-01	1.34e+01	16.27	1.67e-01	2.48e+00	12.7
camerinifratta	diminishing	7.61e-02	2.61e+00	43.37	3.35e-04	1.36e-01	82.32	5.51e-03	1.18e+00	62.37
	fumerotv	0.00e+00	0.00e+00	17.14	3.73e-02	9.77e-01	29.99	7.87e-02	4.88e+00	31.44
	kiwieltv	2.18e-02	9.16e-01	19.57	2.35e-02	9.28e-01	21.44	1.14e-02	8.51e-01	24.98
	sheralitv	8.72e-02	1.11e+00	41.16	8.47e-02	2.40e+00	41.39	1.56e-01	2.51e+00	15.66
deflected	diminishing	2.27e-02	4.84e-01	63.93	2.95e-03	2.65e-01	98.56	4.42e-03	3.61e-01	72.21
	fumerotv	1.59e-01	1.64e+00	38.19	1.62e-01	1.52e+00	27.56	1.59e-01	1.39e+00	30.34
	kiwieltv	1.93e-02	7.46e-01	23.13	2.60e-02	1.01e+00	21.15	2.64e-02	9.86e-01	26.5
	sheralitv	5.48e-02	1.04e+00	97.5	6.05e-02	8.80e-01	102.8	9.88e-03	4.12e-01	65.28
rva	diminishing	9.17e-02	1.09e+00	80.06	1.59e-01	1.35e+00	56.97	1.33e-01	1.10e+00	64.68
	fumerotv	2.85e-02	1.66e+00	60.98	6.12e-03	1.27e+00	24.85	6.30e-03	3.33e-01	24.31
	kiwieltv	1.63e-01	2.78e+00	18.7	1.48e-01	1.24e+00	32.22	1.51e-01	2.29e+00	28.88
	sheralitv	1.67e-01	2.86e+00	9.348	1.67e-01	2.73e+00	11.04	1.67e-01	2.87e+00	11.12
g2g100		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	3.63e-02	3.09e-01	12.78	5.44e-02	3.55e-01	18.5	0.00e+00	0.00e+00	14.02
	fumerotv	2.97e-02	2.09e-01	8.482	1.38e-01	1.28e+00	54.2	5.81e-02	1.43e+00	57.08
	kiwieltv	3.03e-02	9.09e-01	8.894	3.74e-02	9.13e-01	8.528	1.72e-01	2.04e+00	10.7
	sheralitv	1.72e-01	1.93e+00	5.038	1.43e-01	1.09e+00	7.098	1.72e-01	1.71e+00	6.424
camerinifratta	diminishing	1.44e-02	6.14e-01	8.122	3.21e-02	8.16e-01	9.742	9.74e-03	1.24e+00	8.744
	fumerotv	0.00e+00	0.00e+00	4.814	0.00e+00	0.00e+00	5.824	6.84e-02	1.90e+00	16.22
	kiwieltv	1.57e-02	5.44e-01	8.968	1.54e-02	7.96e-01	9.57	1.69e-02	4.90e-01	8.044
	sheralitv	3.90e-02	9.87e-01	15.22	3.69e-02	1.69e+00	23.28	1.11e-01	2.91e+00	5.56
deflected	diminishing	2.17e-02	3.97e-01	8.932	3.54e-02	5.97e-01	9.394	3.24e-02	5.66e-01	8.814
	fumerotv	9.19e-02	9.78e-01	18.02	1.41e-01	1.81e+00	13.13	1.59e-01	1.27e+00	9.684
	kiwieltv	1.18e-02	6.18e-01	7.642	2.92e-02	7.19e-01	8.626	3.19e-02	5.96e-01	9.406
	sheralitv	3.62e-03	1.45e-01	13.18	0.00e+00	0.00e+00	12.55	4.82e-02	4.95e-01	13.93
rva	diminishing	8.47e-02	7.07e-01	12.48	8.31e-02	5.67e-01	17.41	7.41e-03	5.87e-01	18.27
	fumerotv	1.83e-02	1.71e+00	14.17	8.26e-04	4.54e-01	9.884	0.00e+00	0.00e+00	4.538
	kiwieltv	1.38e-01	2.37e+00	11.78	1.41e-01	1.29e+00	10.18	1.71e-01	2.76e+00	7.95
	sheralitv	1.72e-01	2.28e+00	4.028	1.72e-01	2.45e+00	4.946	1.72e-01	2.50e+00	5.222

Tabella 4.4: Risultati numerici g2g144 e g2g100

g2g81		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	0.00e+00	0.00e+00	27.41	4.80e-02	3.83e-01	24.62	3.54e-02	4.42e-01	24.82
	fumerotv	0.00e+00	0.00e+00	17.07	7.86e-02	9.31e-01	73.79	1.14e-01	1.17e+00	6.96
	kiwieltv	4.66e-02	8.67e-01	5.594	7.12e-02	7.25e-01	4.308	1.23e-01	1.95e+00	6.516
	sheralitv	1.40e-01	1.68e+00	3.064	1.14e-01	1.15e+00	5.772	1.36e-01	1.93e+00	4.2
camerinifratta	diminishing	5.54e-02	6.00e+00	8.76	0.00e+00	0.00e+00	14.37	0.00e+00	0.00e+00	11.13
	fumerotv	0.00e+00	0.00e+00	2.674	0.00e+00	0.00e+00	3.742	2.35e-02	2.07e+00	49.43
	kiwieltv	1.81e-02	4.94e-01	4.774	2.40e-02	6.80e-01	5.338	1.33e-02	3.57e-01	4.234
	sheralitv	2.39e-04	7.10e-04	4.874	2.11e-02	6.23e-01	7.254	2.25e-02	3.65e-01	7.084
deflected	diminishing	3.90e-02	3.77e-01	11.51	2.33e-02	1.67e-01	14.53	1.38e-02	2.74e-01	11.08
	fumerotv	4.12e-02	3.24e-01	14.18	1.08e-01	1.22e+00	8.48	6.39e-02	7.50e-01	15.28
	kiwieltv	1.27e-02	4.97e-01	5.358	1.98e-02	5.55e-01	5.33	2.33e-02	5.22e-01	5.204
	sheralitv	1.56e-04	2.82e-02	10.31	0.00e+00	0.00e+00	6.474	0.00e+00	0.00e+00	5.594
rva	diminishing	6.16e-02	6.51e-01	20.4	1.10e-01	9.10e-01	17.23	5.10e-02	4.81e-01	23.86
	fumerotv	5.46e-04	1.33e-01	2.74	6.54e-04	3.87e-01	3.944	0.00e+00	0.00e+00	2.91
	kiwieltv	1.40e-01	2.00e+00	4.6	9.64e-02	8.64e-01	7.292	1.33e-01	1.78e+00	8.512
	sheralitv	1.40e-01	2.19e+00	3.122	1.40e-01	2.28e+00	3.924	1.40e-01	2.25e+00	4.384
g2g49		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	0.00e+00	0.00e+00	3.198	0.00e+00	0.00e+00	4.354	0.00e+00	0.00e+00	4.632
	fumerotv	4.05e-02	6.05e-01	1.916	8.65e-02	1.35e+00	1.87	4.80e-02	3.09e-01	0.91
	kiwieltv	7.31e-02	6.45e-01	1.58	1.06e-01	6.72e-01	1.384	1.11e-01	1.52e+00	1.948
	sheralitv	1.13e-01	1.81e+00	0.956	7.99e-02	1.19e+00	1.158	1.13e-01	1.67e+00	1.256
camerinifratta	diminishing	5.56e-02	1.01e+00	2.326	0.00e+00	0.00e+00	2.574	3.98e-02	5.65e-01	1.37
	fumerotv	0.00e+00	0.00e+00	0.652	0.00e+00	0.00e+00	0.712	0.00e+00	0.00e+00	0.694
	kiwieltv	1.60e-02	3.43e-01	1.47	1.61e-02	5.09e-01	1.648	1.30e-02	3.55e-01	1.466
	sheralitv	0.00e+00	0.00e+00	0.838	0.00e+00	0.00e+00	0.786	4.73e-02	8.65e-01	1.766
deflected	diminishing	6.37e-03	1.31e-01	2.856	0.00e+00	0.00e+00	3.168	1.05e-02	2.83e-01	2.978
	fumerotv	3.87e-02	5.95e-01	2.964	2.92e-02	3.90e-01	2.462	0.00e+00	0.00e+00	3.46
	kiwieltv	7.30e-03	1.29e-01	1.62	1.97e-02	3.79e-01	1.798	1.94e-02	3.39e-01	1.856
	sheralitv	0.00e+00	0.00e+00	1.172	0.00e+00	0.00e+00	1.374	0.00e+00	0.00e+00	1.192
rva	diminishing	3.03e-04	3.01e-04	4.772	6.34e-02	5.21e-01	5.51	2.41e-04	1.92e-03	5.612
	fumerotv	0.00e+00	0.00e+00	0.654	0.00e+00	0.00e+00	0.988	0.00e+00	0.00e+00	0.796
	kiwieltv	1.12e-01	1.88e+00	1.422	7.19e-02	9.41e-01	1.656	1.13e-01	1.49e+00	2.638
	sheralitv	1.13e-01	1.67e+00	1.034	1.13e-01	1.80e+00	1.312	1.13e-01	1.85e+00	1.346

Tabella 4.5: Risultati numerici g2g81 e g2g49

g2g25		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	0.00e+00	0.00e+00	1.112	0.00e+00	0.00e+00	0.78	0.00e+00	0.00e+00	0.77
	fumerotv	2.68e-02	5.96e-01	0.22	3.28e-02	2.39e-01	1.16	5.19e-02	6.30e-01	0.2
	kiwieltv	6.36e-02	1.10e+00	0.38	1.27e-01	1.44e+00	0.306	1.14e-01	2.12e+00	0.412
	sheralitv	1.10e-01	2.89e+00	0.304	1.96e-02	1.16e+00	0.184	1.27e-01	2.76e+00	0.3
camerinifratta	diminishing	3.69e-02	1.54e+00	0.526	0.00e+00	0.00e+00	0.666	4.02e-02	1.28e+00	0.278
	fumerotv	0.00e+00	0.00e+00	0.088	0.00e+00	0.00e+00	0.106	0.00e+00	0.00e+00	0.102
	kiwieltv	0.00e+00	0.00e+00	0.232	2.26e-03	1.27e-01	0.294	1.69e-03	1.33e-01	0.186
	sheralitv	0.00e+00	0.00e+00	0.13	0.00e+00	0.00e+00	0.078	1.63e-02	6.22e-01	0.344
deflected	diminishing	1.99e-02	2.00e-01	0.52	1.99e-02	3.98e-01	0.584	2.77e-02	2.97e-01	0.536
	fumerotv	0.00e+00	0.00e+00	0.162	0.00e+00	0.00e+00	0.144	0.00e+00	0.00e+00	0.138
	kiwieltv	0.00e+00	0.00e+00	0.326	1.19e-03	8.56e-02	0.296	2.92e-03	1.25e-01	0.366
	sheralitv	0.00e+00	0.00e+00	0.114	0.00e+00	0.00e+00	0.13	0.00e+00	0.00e+00	0.148
rva	diminishing	3.45e-02	3.90e-01	0.738	0.00e+00	0.00e+00	1.37	3.01e-02	2.75e-01	1.024
	fumerotv	2.59e-03	2.46e-01	0.136	1.75e-02	1.24e+00	0.194	2.18e-02	3.43e-01	0.182
	kiwieltv	1.27e-01	2.94e+00	0.302	1.27e-01	4.58e-01	0.302	1.27e-01	5.20e-01	0.302
	sheralitv	1.27e-01	3.28e+00	0.264	1.27e-01	3.34e+00	0.324	1.27e-01	3.30e+00	0.322
g2pm144		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	4.22e-02	2.23e-01	36.96	1.60e-01	3.88e-01	38.95	5.84e-02	2.36e-01	42.53
	fumerotv	4.29e-02	8.53e-01	51.66	2.42e-01	2.74e-01	263.7	1.76e-01	5.65e-02	36.75
	kiwieltv	3.42e-01	1.07e+00	25.13	4.73e-02	1.01e+00	18.56	3.54e-01	2.23e-01	26.3
	sheralitv	3.23e-01	3.24e-01	44.36	3.37e-01	8.96e-01	19.57	3.56e-01	5.70e-02	27.27
camerinifratta	diminishing	1.13e-01	3.56e-01	18.75	5.91e-02	1.17e+00	20.26	1.02e-02	3.59e-01	15.2
	fumerotv	3.27e-03	1.31e-01	38.79	8.33e-02	6.87e-01	28.07	8.48e-02	1.71e+00	44.45
	kiwieltv	4.85e-03	3.80e+00	39.72	2.92e-03	6.28e-01	36.74	2.74e-03	1.74e+00	29.5
	sheralitv	2.49e-01	8.98e-02	36.07	2.78e-01	1.83e-01	27.56	3.11e-01	1.58e-01	13.93
deflected	diminishing	4.46e-02	1.55e+00	26.64	6.57e-02	2.00e+00	23.54	3.91e-02	1.31e+00	26.91
	fumerotv	3.33e-01	9.37e-01	21.09	3.10e-01	8.74e-01	27.96	3.32e-01	8.30e+00	26.13
	kiwieltv	3.47e-03	6.06e-01	40.7	3.87e-03	3.08e-01	69.99	4.40e-03	5.94e-01	35.19
	sheralitv	2.65e-01	8.07e-01	39.91	2.13e-01	9.84e-01	50.7	1.72e-01	2.18e+00	59.96
rva	diminishing	2.27e-01	2.28e+00	35.17	1.76e-01	8.61e-01	40.12	1.61e-01	1.92e+00	41.93
	fumerotv	4.88e-02	4.54e-01	99.24	1.57e-01	2.58e-02	74.23	4.50e-02	1.43e-01	55.11
	kiwieltv	3.58e-01	3.56e-01	16.05	2.41e-01	1.85e+00	24.59	1.78e-01	6.99e-02	26.7
	sheralitv	3.58e-01	3.33e-01	9.85	3.58e-01	2.94e-01	14.2	3.58e-01	3.43e-01	13.42

Tabella 4.6: Risultati numerici g2g25 e g2pm144

g2pm100		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	2.96e-02	1.09e-01	37.33	2.23e-01	2.21e+00	33.57	6.26e-02	2.46e-01	41.53
	fumerotv	3.65e-03	2.67e-01	11.69	2.57e-01	5.76e-01	8.136	1.21e-01	9.10e-01	30.89
	kiwieltv	2.79e-01	4.71e-01	9.91	1.53e-01	1.06e+00	7.284	3.05e-01	1.94e-01	11.25
	sheralitv	3.03e-01	2.89e-02	15.49	2.31e-01	1.41e+00	8.608	3.00e-01	4.06e-02	26.99
camerinfiratta	diminishing	5.78e-02	4.05e-01	16.52	1.62e-03	8.01e+00	22.6	1.57e-03	1.13e+00	18.68
	fumerotv	2.48e-04	5.04e-02	11.65	4.90e-04	1.01e-01	12.03	2.54e-02	6.86e-01	56.4
	kiwieltv	9.66e-04	1.48e-01	13.32	3.17e-03	2.55e+00	14.41	3.16e-02	9.91e-01	11.38
	sheralitv	1.22e-01	6.42e-01	14.34	1.33e-01	6.70e-01	16.29	9.99e-02	5.40e-01	12.7
deflected	diminishing	5.23e-03	1.32e-01	22.61	8.81e-03	1.91e-01	32.19	5.04e-03	1.00e-01	26.67
	fumerotv	2.54e-01	1.75e+00	16.81	2.58e-01	6.71e-01	10.84	1.08e-01	1.29e+00	26.78
	kiwieltv	1.35e-03	2.48e-01	18.13	2.72e-03	1.57e-01	24.67	3.01e-03	2.44e-01	15.46
	sheralitv	1.27e-01	7.59e-01	17.62	1.19e-01	6.23e-01	29.43	6.31e-02	1.36e+00	21.37
rva	diminishing	1.06e-01	9.63e-01	59.4	2.84e-01	5.47e-01	27.43	2.28e-01	7.09e-01	28.83
	fumerotv	1.37e-01	2.34e-01	11.74	1.28e-01	5.81e-01	22.54	2.84e-02	2.00e-01	15.65
	kiwieltv	3.04e-01	2.34e-01	5.872	2.86e-01	1.33e+00	8.186	2.92e-01	3.27e-01	8.082
	sheralitv	3.05e-01	2.42e-01	4.572	3.05e-01	2.07e-01	6.15	3.05e-01	2.32e-01	5.958
g2pm81		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	1.72e-04	2.25e-02	5.254	2.25e-01	1.95e+00	5.744	1.30e-04	2.43e-02	6.378
	fumerotv	1.90e-02	1.98e-01	4.852	2.96e-01	4.31e-01	7.492	1.81e-01	1.60e+00	47.51
	kiwieltv	2.38e-01	1.48e-01	5.926	2.23e-01	4.22e+00	4.426	3.25e-01	1.53e+01	5.732
	sheralitv	3.21e-01	3.51e-01	15.64	2.62e-01	1.40e+01	4.35	3.27e-01	3.33e-01	3.74
camerinfiratta	diminishing	9.72e-02	8.96e-01	4.64	3.78e-02	1.35e+00	4.704	1.17e-02	4.01e-01	4.36
	fumerotv	1.80e-03	2.88e-02	10.04	4.04e-05	1.78e-02	5.23	3.93e-02	7.87e-01	16.39
	kiwieltv	3.59e-03	5.46e-01	6.396	3.13e-03	2.91e-01	6.342	3.14e-03	2.33e-01	5.922
	sheralitv	1.67e-02	2.01e+00	8.642	7.50e-02	2.23e+00	8.33	1.47e-01	3.52e+00	6.408
deflected	diminishing	1.34e-02	6.30e-01	6.834	4.07e-02	7.93e-01	6.154	4.23e-02	1.21e+00	7.316
	fumerotv	1.54e-01	5.48e-01	7.226	1.82e-01	3.24e+02	15.94	1.96e-03	2.04e-01	19.61
	kiwieltv	4.15e-03	4.97e-01	6.174	2.34e-03	1.99e-01	9.044	4.00e-03	9.13e-01	6.922
	sheralitv	9.57e-02	3.07e+00	8.176	7.94e-02	4.87e+01	8.56	3.55e-04	7.94e-02	6.108
rva	diminishing	2.33e-03	1.59e-01	7.172	2.08e-02	1.08e-01	7.736	2.21e-03	2.05e-01	7.198
	fumerotv	6.99e-05	1.38e-02	4.056	8.83e-02	6.96e-01	4.712	5.73e-04	3.40e-01	4.024
	kiwieltv	3.04e-01	2.37e-01	4.636	3.06e-01	4.38e-01	6.542	3.04e-01	3.67e-01	6.162
	sheralitv	3.27e-01	2.71e-01	3.03	3.27e-01	2.45e-01	4.164	3.27e-01	2.99e-01	4.36

Tabella 4.7: Risultati numerici g2pm100 e g2pm81

g2pm49		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	8.31e-06	5.91e-03	1.736	1.50e-05	1.20e-03	2.284	5.29e-06	1.02e-03	1.91
	fumerotv	9.21e-02	2.08e-03	5.48	9.23e-02	3.66e-01	21.21	1.68e-01	1.57e-01	1.856
	kiwieltv	3.16e-01	2.23e-01	1.668	3.52e-01	4.47e-01	1.308	3.52e-01	2.13e-01	1.796
	sheralitv	3.52e-01	3.21e-01	0.938	2.85e-01	4.15e-01	1.196	3.52e-01	3.59e-01	1.146
camerinfiratta	diminishing	1.33e-02	9.96e-01	2.002	1.26e-02	9.48e-01	1.774	8.47e-03	1.03e+00	2.268
	fumerotv	6.02e-06	1.66e-01	1.71	6.97e-06	1.53e-01	1.796	3.72e-02	3.15e-01	1.566
	kiwieltv	4.31e-03	9.28e-01	2.354	1.98e-03	1.05e+00	2.87	2.16e-02	9.81e-01	1.908
	sheralitv	2.39e-04	8.90e-02	1.958	6.63e-03	1.46e+00	2.35	6.84e-02	4.21e-01	1.582
deflected	diminishing	5.59e-03	2.60e-01	2.078	1.04e-02	4.54e-01	3.176	8.58e-03	3.35e-01	2.738
	fumerotv	2.13e-02	2.47e-01	10.3	1.55e-02	1.05e-01	3.454	2.54e-05	1.35e-02	6.204
	kiwieltv	4.02e-03	1.02e+00	2.742	1.56e-03	2.63e-01	3.318	3.78e-03	2.90e-01	2.932
	sheralitv	1.01e-02	1.44e+00	3.202	3.76e-04	4.89e-02	2.976	5.46e-03	2.85e-01	3.37
rva	diminishing	1.80e-03	2.80e-01	2.29	7.20e-02	1.60e-01	3.652	1.26e-03	1.75e-03	2.374
	fumerotv	3.48e-03	2.96e-01	1.416	1.34e-01	4.80e-01	2.864	1.88e-03	3.05e+01	1.566
	kiwieltv	3.52e-01	3.11e-01	1.224	3.52e-01	2.27e+00	1.224	3.52e-01	2.29e+00	1.224
	sheralitv	3.52e-01	3.19e-01	1.002	3.52e-01	2.58e-01	1.322	3.52e-01	2.81e-01	1.304
g2pm25		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	8.59e-04	1.76e-02	1.136	1.47e-03	1.71e-02	0.862	1.09e-03	1.82e-02	1.046
	fumerotv	2.15e-01	3.17e+00	0.228	1.47e-01	9.76e-01	0.228	1.67e-01	3.83e+00	0.362
	kiwieltv	2.73e-01	3.03e-01	0.268	2.73e-01	1.32e+00	0.382	2.73e-01	1.05e+01	0.264
	sheralitv	2.73e-01	1.26e-01	0.342	1.27e-01	1.28e-01	0.278	2.66e-01	2.52e-01	1.058
camerinfiratta	diminishing	2.89e-04	7.67e-02	0.474	3.05e-04	1.18e-01	0.608	2.21e-04	1.04e-01	0.572
	fumerotv	3.49e-05	5.37e-02	0.294	7.49e-05	1.89e+00	0.328	3.26e-04	5.90e-01	0.256
	kiwieltv	2.36e-03	1.53e-01	0.446	5.14e-03	1.61e-01	0.408	1.29e-02	2.41e-01	0.356
	sheralitv	8.95e-05	1.25e-01	0.37	7.09e-04	3.29e-01	0.376	6.81e-02	6.14e-01	0.432
deflected	diminishing	1.91e-03	1.31e-02	0.58	2.37e-03	3.09e-02	0.594	4.96e-04	8.20e-02	0.596
	fumerotv	8.53e-05	1.48e-02	0.63	1.14e-04	3.06e-02	0.368	7.18e-05	3.42e-02	0.398
	kiwieltv	1.40e-03	6.07e-02	0.51	5.16e-03	1.52e-01	0.47	2.52e-03	5.10e-01	0.506
	sheralitv	7.06e-05	2.86e-02	0.35	1.16e-04	6.54e-02	0.426	4.86e-04	2.26e-02	0.728
rva	diminishing	4.58e-02	9.23e-02	0.692	2.59e-03	8.18e-02	1.48	3.67e-02	1.48e-01	0.922
	fumerotv	8.66e-04	5.29e-01	0.318	1.35e-01	6.63e-02	0.254	7.57e-02	3.00e-02	0.274
	kiwieltv	2.23e-01	1.39e+00	0.374	2.20e-01	9.13e-01	0.422	2.73e-01	3.00e-01	0.504
	sheralitv	2.73e-01	1.90e-01	0.256	2.73e-01	2.58e-01	0.32	2.73e-01	3.94e-01	0.316

Tabella 4.8: Risultati numerici g2pm49 e g2pm25

g3g125		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	0.00e+00	0.00e+00	75.9	1.77e-01	7.37e-01	68.7	0.00e+00	0.00e+00	72.9
	fumerotv	4.43e-02	5.37e-01	39.6	3.04e-01	2.22e+00	30.5	1.50e-01	1.10e+00	70.97
	kiwieltv	3.24e-01	5.20e+00	17.57	3.24e-01	1.19e+00	14.29	3.24e-01	1.50e+00	15.44
	sheralitv	3.24e-01	4.42e+00	7.528	2.69e-01	1.38e+00	13.42	3.24e-01	2.18e+00	11.43
camerinifratta	diminishing	1.27e-01	2.50e+00	18.24	4.75e-02	1.41e+00	24.23	1.09e-01	1.57e+00	19.03
	fumerotv	0.00e+00	0.00e+00	18.46	0.00e+00	0.00e+00	23.07	3.01e-02	9.39e-01	237.3
	kiwieltv	1.24e-02	9.89e-01	24.47	1.02e-02	9.79e-01	21.96	3.99e-03	6.28e-01	24.86
	sheralitv	5.55e-02	8.23e-01	51.55	1.66e-01	1.42e+00	26.3	1.70e-01	1.66e+00	21.11
deflected	diminishing	9.43e-04	5.10e-01	35.81	1.27e-02	6.99e-01	26.76	1.26e-03	6.66e-01	37.6
	fumerotv	2.04e-01	1.17e+00	28.93	1.12e-01	8.58e-01	61.34	1.50e-01	8.78e-01	54.06
	kiwieltv	8.76e-03	9.98e-01	31.39	1.99e-02	9.11e-01	25.38	1.19e-02	7.85e-01	20.66
	sheralitv	1.65e-01	1.32e+00	38.22	8.74e-02	8.67e-01	48.41	5.40e-02	6.93e-01	50.4
rva	diminishing	5.83e-02	1.94e-01	49.35	1.77e-01	8.74e-01	49.54	3.45e-03	8.02e-02	57.16
	fumerotv	1.39e-01	4.32e+00	45.1	8.55e-03	7.24e-01	34.34	6.44e-02	3.14e+00	43.44
	kiwieltv	3.24e-01	2.74e+00	11.59	3.24e-01	3.06e+00	13.86	2.97e-01	3.33e+00	19.51
	sheralitv	3.24e-01	2.81e+00	7.364	3.24e-01	2.91e+00	9.628	3.24e-01	3.01e+00	9.76
g3g64		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	0.00e+00	0.00e+00	3.222	0.00e+00	0.00e+00	5.178	0.00e+00	0.00e+00	3.442
	fumerotv	5.22e-02	3.76e-01	4.09	1.05e-01	6.92e-01	26.61	1.67e-01	1.44e+00	7.754
	kiwieltv	2.67e-01	1.21e+00	3.39	2.67e-01	1.23e+00	2.232	2.67e-01	1.47e+00	3.934
	sheralitv	2.67e-01	2.16e+00	1.786	2.67e-01	1.11e+00	2.054	2.67e-01	2.19e+00	2.478
camerinifratta	diminishing	7.82e-02	3.21e+00	10.02	1.78e-02	6.83e-01	4.356	3.73e-02	8.95e-01	3.3
	fumerotv	0.00e+00	0.00e+00	2.382	0.00e+00	0.00e+00	2.372	3.36e-04	6.36e-01	5.632
	kiwieltv	1.03e-02	6.94e-01	4.188	6.12e-03	5.26e-01	4.944	2.83e-03	7.30e-01	4.724
	sheralitv	0.00e+00	0.00e+00	2.54	7.48e-03	6.89e-02	4.584	1.05e-01	1.03e+00	2.578
deflected	diminishing	7.54e-04	1.34e-01	5.556	8.30e-03	3.45e-01	6.31	5.67e-03	3.01e-01	4.562
	fumerotv	9.95e-02	7.92e-01	9.626	0.00e+00	0.00e+00	6.52	0.00e+00	0.00e+00	3.33
	kiwieltv	7.94e-03	5.64e-01	4.412	1.29e-02	5.14e-01	4.842	1.24e-02	6.04e-01	4.636
	sheralitv	0.00e+00	0.00e+00	3.208	4.37e-02	2.63e-01	3.064	2.24e-03	1.18e-01	5.112
rva	diminishing	0.00e+00	0.00e+00	3.642	1.92e-01	7.50e-01	3.53	5.83e-04	0.00e+00	4.328
	fumerotv	4.63e-04	3.29e-01	2.154	1.55e-03	9.54e-01	2.114	0.00e+00	0.00e+00	1.92
	kiwieltv	2.55e-01	1.44e+00	4.904	7.28e-03	2.54e-01	5.818	2.59e-01	1.99e+00	4.564
	sheralitv	2.67e-01	2.24e+00	1.902	2.67e-01	2.08e+00	2.522	2.67e-01	2.05e+00	2.516

Tabella 4.9: Risultati numerici g3g125 e g3g64

g3g27		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	0.00e+00	0.00e+00	0.106	0.00e+00	0.00e+00	0.128	0.00e+00	0.00e+00	0.128
	fumerotv	1.28e-01	1.13e+00	0.282	0.00e+00	0.00e+00	1.88	1.16e-01	8.17e-01	0.268
	kiwieltv	2.12e-01	1.73e+00	0.922	2.22e-01	2.40e+00	0.336	2.22e-01	1.42e+00	0.362
	sheralitv	2.22e-01	2.52e+00	0.206	1.36e-02	1.58e+00	0.242	2.23e-01	2.65e+00	0.314
camerinifratta	diminishing	0.00e+00	0.00e+00	0.104	0.00e+00	0.00e+00	0.126	0.00e+00	0.00e+00	0.076
	fumerotv	0.00e+00	0.00e+00	0.178	1.99e-04	4.43e-01	0.23	0.00e+00	0.00e+00	0.16
	kiwieltv	1.50e-04	4.56e-02	0.41	0.00e+00	0.00e+00	0.422	0.00e+00	0.00e+00	0.302
	sheralitv	0.00e+00	0.00e+00	0.156	0.00e+00	0.00e+00	0.132	1.90e-02	4.27e-01	0.37
deflected	diminishing	0.00e+00	0.00e+00	0.176	0.00e+00	0.00e+00	0.192	0.00e+00	0.00e+00	0.19
	fumerotv	0.00e+00	0.00e+00	0.178	0.00e+00	0.00e+00	0.196	0.00e+00	0.00e+00	0.186
	kiwieltv	0.00e+00	1.35e-01	0.458	2.40e-04	4.14e-02	0.486	1.40e-04	3.27e-02	0.48
	sheralitv	0.00e+00	0.00e+00	0.206	0.00e+00	0.00e+00	0.252	0.00e+00	0.00e+00	0.206
rva	diminishing	3.88e-02	3.63e-01	0.286	0.00e+00	0.00e+00	0.344	3.54e-02	1.69e-01	0.31
	fumerotv	4.46e-03	3.62e-01	0.21	1.05e-01	2.24e+00	0.276	0.00e+00	0.00e+00	0.176
	kiwieltv	2.23e-01	2.75e+00	0.364	1.82e-01	5.05e+00	0.52	2.22e-01	2.09e+00	0.526
	sheralitv	2.23e-01	2.04e+00	0.346	2.23e-01	2.36e+00	0.388	2.23e-01	3.03e+00	0.384
g3pm125		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	2.29e-04	1.48e-02	75.07	9.08e-02	1.55e-01	132.6	2.34e-04	1.53e-02	99.93
	fumerotv	5.81e-04	3.61e-01	23.33	4.79e-01	2.89e-05	21.51	1.30e-01	8.54e-02	62.37
	kiwieltv	5.11e-01	2.67e-01	11.09	5.11e-01	8.18e+00	16.09	5.11e-01	1.08e+00	8.628
	sheralitv	5.11e-01	3.12e-01	7.172	4.09e-01	5.17e-01	12.53	4.81e-01	1.86e-01	30.12
camerinifratta	diminishing	4.60e-01	8.72e-01	13.71	2.70e-02	8.33e-01	38.67	1.04e-01	2.19e+00	30.71
	fumerotv	3.20e-03	9.34e-02	50.36	1.28e-03	1.71e-01	37.04	1.56e-01	1.54e+00	167.9
	kiwieltv	3.62e-01	5.99e-02	26.28	5.11e-01	2.89e+00	18.47	5.11e-01	3.51e+00	17.15
	sheralitv	1.89e-01	9.61e-01	37.85	1.41e-01	9.65e-02	38.3	4.15e-01	2.98e-02	12.69
deflected	diminishing	1.03e-02	1.04e+00	35.14	1.18e-02	1.07e+00	52.73	4.95e-03	1.90e+00	48.42
	fumerotv	4.27e-01	2.57e-01	30.26	2.52e-01	5.86e-01	46.05	2.83e-01	5.99e-01	43.26
	kiwieltv	5.11e-01	4.04e-02	14.37	5.11e-01	8.29e-01	16.73	5.11e-01	8.38e-02	17.61
	sheralitv	2.20e-01	4.93e-01	39.26	1.37e-01	5.86e-01	47.55	8.09e-02	8.39e-01	52.04
rva	diminishing	7.41e-02	3.43e-01	65.12	2.66e-01	5.31e-01	72.74	8.38e-02	4.98e-01	67.25
	fumerotv	4.65e-02	3.32e-02	41.84	2.32e-02	4.98e-01	47.86	2.23e-01	3.53e-02	23.93
	kiwieltv	5.11e-01	2.47e-01	11.06	5.11e-01	2.54e+00	24.18	5.11e-01	5.21e+00	27.52
	sheralitv	5.11e-01	2.47e-01	7.694	5.11e-01	2.57e-01	10.38	5.11e-01	2.28e-01	10.24

Tabella 4.10: Risultati numerici g3g27 e g3pm125

g3pm64		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	7.79e-06	6.43e-03	2.488	6.95e-02	2.01e-01	7.016	4.70e-06	2.50e-03	2.462
	fumerotv	6.56e-02	1.71e-01	3.322	1.39e-01	4.73e-01	29.27	3.40e-01	9.88e-02	3.702
	kiwieltv	1.33e-01	8.21e-02	3.406	4.12e-01	1.21e+00	2.892	4.00e-01	2.06e+00	3.268
	sheralitv	4.18e-01	3.08e-01	1.772	2.57e-01	8.36e-02	3.854	4.14e-01	3.76e-01	6.268
camerinifratta	diminishing	1.36e-03	7.48e-02	6.51	2.33e-03	5.80e-02	6	2.01e-03	6.28e-02	4.224
	fumerotv	5.00e-05	4.10e-01	3.958	3.17e-02	7.59e-01	4.238	7.55e-04	5.07e-01	5.914
	kiwieltv	2.84e-04	3.74e-02	5.66	2.70e-04	3.00e-02	6.672	2.13e-04	1.61e-02	6.186
	sheralitv	3.58e-02	4.92e-01	5.206	1.28e-02	1.15e-01	6.772	6.84e-02	4.42e-01	3.688
deflected	diminishing	7.96e-04	4.08e-02	6.226	1.11e-03	2.44e-02	8.01	8.39e-04	5.17e-02	6.158
	fumerotv	1.61e-01	3.51e-01	7.836	3.06e-04	4.60e-01	5.968	2.67e-04	4.46e-02	5.28
	kiwieltv	4.33e-04	2.34e-01	5.16	3.87e-04	8.20e-02	7.482	3.79e-04	9.03e-01	6.714
	sheralitv	4.89e-02	1.25e+00	7.252	2.41e-02	1.40e+00	8.406	6.05e-02	7.88e-01	6.848
rva	diminishing	1.38e-03	9.12e-03	3.878	7.80e-02	1.85e-01	8.86	1.11e-03	6.02e-03	4.92
	fumerotv	4.83e-03	4.50e-01	2.698	4.81e-02	2.98e-01	3.226	1.86e-02	2.22e-01	2.858
	kiwieltv	4.18e-01	3.39e-01	2.428	4.18e-01	4.85e+00	2.428	4.18e-01	5.62e+00	4.706
	sheralitv	4.18e-01	2.99e-01	1.856	4.18e-01	3.72e-01	2.472	4.18e-01	3.16e-01	2.514
g3pm27		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	1.87e-03	3.57e-03	0.452	1.72e-04	1.18e-02	0.516	1.59e-04	6.31e-03	0.534
	fumerotv	1.61e-01	4.09e-01	0.348	1.05e-01	2.83e-01	1.718	1.55e-01	1.87e-01	0.352
	kiwieltv	4.40e-01	6.65e-01	0.324	2.70e-01	1.24e+00	0.464	4.13e-01	7.29e-01	0.516
	sheralitv	4.39e-01	1.28e-01	0.492	1.85e-01	2.50e-01	0.482	4.32e-01	2.03e+00	1.516
camerinifratta	diminishing	5.98e-03	1.72e-01	0.74	7.04e-03	1.70e-01	0.8	5.84e-03	1.49e-01	0.674
	fumerotv	8.81e-04	6.99e-02	0.536	1.37e-03	6.58e-02	0.582	1.69e-03	3.53e-02	0.568
	kiwieltv	2.46e-03	1.44e-01	0.798	2.97e-03	1.53e-01	0.774	2.06e-03	1.52e-01	0.838
	sheralitv	8.65e-04	2.90e-02	0.568	1.74e-03	4.40e-02	0.604	8.42e-02	2.82e-01	0.53
deflected	diminishing	3.27e-03	6.05e-02	0.79	2.13e-03	1.88e-02	0.946	2.74e-03	3.69e-02	0.838
	fumerotv	8.59e-04	1.25e-02	0.816	1.31e-03	5.33e-02	0.68	9.20e-04	5.87e-02	0.724
	kiwieltv	2.78e-03	7.06e-02	0.926	3.45e-03	1.21e-01	0.81	3.11e-03	9.64e-02	0.824
	sheralitv	7.57e-04	3.14e-02	0.768	5.79e-04	1.22e-02	0.88	6.43e-04	2.95e-02	0.818
rva	diminishing	5.99e-04	1.82e-02	0.924	1.00e-03	4.73e-03	0.902	1.56e-03	4.06e-03	0.72
	fumerotv	6.34e-02	8.94e-02	0.546	1.40e-01	5.42e-01	0.316	7.63e-02	3.52e-01	0.37
	kiwieltv	3.33e-01	1.35e-01	0.47	3.55e-01	2.21e-02	0.426	4.40e-01	2.89e-01	0.682
	sheralitv	4.40e-01	8.85e-02	0.3	4.40e-01	1.73e-02	0.374	4.40e-01	3.71e-02	0.374

Tabella 4.11: Risultati numerici g3pm64 e g3pm27

p150		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	1.90e-02	4.87e-01	74.16	2.00e-01	6.10e-01	38.94	1.57e-02	4.78e-01	92.36
	fumerotv	2.34e-02	6.48e-01	57.86	1.34e-01	1.23e+00	46.1	7.27e-02	1.17e+00	63.33
	kiwieltv	1.01e-01	8.79e-01	131.6	1.47e-01	6.22e-01	69.38	8.73e-02	1.14e+00	167.5
	sheralitv	2.48e-01	1.70e+00	11.59	1.81e-01	1.02e+00	22.37	2.49e-01	1.75e+00	13.12
camerinifratta	diminishing	8.15e-02	1.80e+00	33.03	1.84e-02	8.15e-01	29.93	1.83e-02	8.02e-01	26.48
	fumerotv	1.21e-02	6.53e-02	20.07	2.16e-04	1.98e-02	22.22	2.05e-02	2.97e+00	25.3
	kiwieltv	8.83e-02	3.29e-01	169.6	3.36e-02	3.03e-01	151.9	4.50e-05	1.14e-02	60.81
	sheralitv	6.81e-02	1.22e+00	31.13	8.26e-02	1.36e+00	31.74	1.14e-01	1.18e+00	24.03
deflected	diminishing	9.05e-03	2.58e-01	27.9	1.83e-02	2.87e-01	34.39	9.00e-03	2.13e-01	28.59
	fumerotv	7.15e-02	6.18e-01	46.68	4.46e-02	5.80e-01	39.09	7.15e-02	6.41e-01	43.92
	kiwieltv	1.73e-01	8.92e-01	44.78	1.90e-01	7.47e-01	45.79	1.97e-01	8.19e-01	42.3
	sheralitv	1.46e-01	8.62e-01	42.82	9.66e-02	8.58e-01	50.87	1.01e-01	8.33e-01	41.26
rva	diminishing	1.14e-01	8.56e-01	113.2	1.27e-01	8.26e-01	90.7	6.24e-02	6.29e-01	103.8
	fumerotv	2.21e-02	1.31e+00	72.35	2.92e-02	1.94e+00	101.6	3.58e-03	5.33e-01	37.2
	kiwieltv	1.56e-01	1.60e+00	21.65	2.12e-01	9.26e-01	57.81	1.61e-01	1.36e+00	27.86
	sheralitv	2.53e-01	1.60e+00	11.65	2.53e-01	1.65e+00	14.1	2.53e-01	1.65e+00	14.23
p100		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	2.41e-03	5.74e-02	34.2	8.27e-02	3.47e-01	23.77	2.02e-05	6.30e-03	32.77
	fumerotv	1.81e-02	1.02e-01	11.59	1.08e-01	9.64e-01	26.57	2.66e-02	5.86e-01	25.81
	kiwieltv	8.98e-02	8.84e-01	34.47	7.17e-02	4.85e-01	36.41	7.22e-02	6.31e-01	32.29
	sheralitv	2.48e-01	1.24e+00	4.568	1.57e-01	6.85e-01	7.221	2.43e-01	1.71e+00	8.8
camerinifratta	diminishing	5.47e-02	8.85e-01	10.49	1.26e-02	2.52e-01	11.3	1.10e-02	4.65e-01	10.9
	fumerotv	1.25e-06	5.02e-02	6.171	4.06e-05	1.33e-02	7.789	2.79e-02	8.66e-01	10.92
	kiwieltv	1.35e-02	3.31e-02	48.91	1.60e-02	3.87e-02	26.44	7.31e-03	1.89e-02	25.12
	sheralitv	4.10e-02	4.77e-01	8.153	8.27e-03	1.98e-01	11.82	1.16e-01	7.67e-01	6.724
deflected	diminishing	5.28e-03	1.34e-01	14.21	1.20e-02	1.61e-01	14.7	5.72e-03	1.38e-01	10.84
	fumerotv	4.25e-02	4.35e-01	17.47	6.43e-03	1.15e-01	10.05	3.01e-03	1.12e-01	9.527
	kiwieltv	1.62e-01	5.44e-01	15.6	1.72e-01	5.07e-01	15.95	1.66e-01	5.35e-01	13.71
	sheralitv	8.49e-02	6.06e-01	13.82	6.44e-02	5.08e-01	16.52	8.42e-02	4.92e-01	14.43
rva	diminishing	3.88e-03	2.04e-01	42.87	9.54e-02	5.19e-01	22.65	4.49e-03	8.21e-02	42.33
	fumerotv	1.81e-02	6.37e-01	10.44	2.50e-02	6.14e-01	12.51	8.01e-03	4.07e-01	8.048
	kiwieltv	1.85e-01	1.15e+00	7.631	1.81e-01	8.49e-01	26.96	1.52e-01	8.86e-01	12.28
	sheralitv	2.48e-01	1.45e+00	4.881	2.48e-01	1.35e+00	6.136	2.48e-01	1.36e+00	6.127

Tabella 4.12: Risultati numerici p150 e p100

p50		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	2.06e-05	2.52e-03	0.9207	9.18e-03	5.87e-02	1.623	7.30e-05	1.04e-02	1.343
	fumerotv	5.07e-02	9.81e-01	2.063	7.67e-02	1.51e+00	6.143	5.25e-02	9.26e-01	3.388
	kiwieltv	1.28e-01	6.14e-01	1.481	2.51e-03	6.59e-02	0.9153	1.03e-01	7.83e-01	1.418
	sheralitv	1.87e-01	1.39e+00	0.9947	3.93e-02	1.40e+01	1.286	1.86e-01	1.40e+00	1.223
camerinifratta	diminishing	4.60e-02	8.77e-01	1.303	4.05e-03	7.32e-02	1.269	3.00e-03	7.23e-02	1.387
	fumerotv	4.66e-07	1.02e-01	0.7753	2.13e-06	7.04e-02	0.92	2.36e-03	2.17e-01	0.8687
	kiwieltv	1.43e-04	1.30e-02	0.668	1.11e-04	8.82e-03	0.7553	1.13e-04	1.30e-02	0.6833
	sheralitv	3.36e-06	1.00e-02	0.9707	6.47e-04	2.21e-02	1.055	5.62e-02	5.53e-01	0.9307
deflected	diminishing	2.10e-03	2.00e-02	1.529	1.88e-03	5.46e-02	1.505	4.48e-03	1.23e-01	1.357
	fumerotv	3.01e-03	6.99e-02	1.591	4.23e-04	3.97e-02	1.479	5.64e-04	1.92e-02	1.199
	kiwieltv	4.52e-02	4.13e-01	3.071	4.35e-03	8.61e-02	2.569	5.27e-03	3.83e-02	1.293
	sheralitv	1.54e-02	2.56e-01	1.502	2.93e-03	7.19e-02	1.863	9.62e-03	9.93e-02	1.529
rva	diminishing	2.89e-03	3.44e-02	1.964	8.55e-03	9.90e-02	4.461	7.89e-03	1.11e-01	2.548
	fumerotv	5.04e-03	6.62e-02	0.886	1.94e-02	4.39e-01	1.909	1.80e-02	1.37e-01	0.9873
	kiwieltv	1.87e-01	1.50e+00	1.44	1.83e-01	1.56e+00	1.838	1.87e-01	1.39e+00	1.701
	sheralitv	1.87e-01	1.50e+00	1.159	1.87e-01	1.51e+00	1.477	1.87e-01	1.58e+00	1.456
s136		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	2.90e-01	5.43e-01	71.88	2.90e-01	4.95e-01	95.91	3.04e-01	7.78e-01	41.1
	fumerotv	5.24e-02	4.44e-01	25.79	2.01e-01	1.06e+00	55.35	4.62e-02	9.79e-01	98.1
	kiwieltv	2.23e-01	1.11e+00	69.73	2.85e-01	6.74e-01	21.12	1.94e-01	1.27e+00	83.64
	sheralitv	3.59e-01	1.35e+00	8.12	2.57e-01	7.21e+00	15.93	3.59e-01	1.32e+00	9.458
camerinifratta	diminishing	2.39e-01	7.36e-01	106.8	2.55e-01	9.04e-01	85.63	2.30e-01	7.09e-01	35.69
	fumerotv	5.36e-04	5.49e-02	19.5	1.57e-04	5.39e-03	19.76	3.07e-02	1.25e+00	28.36
	kiwieltv	1.74e-01	4.78e-01	145.7	4.27e-02	6.73e-01	179.9	5.68e-03	2.24e-01	235.2
	sheralitv	9.24e-02	8.27e-01	28.04	1.77e-01	7.49e-01	23.91	2.87e-01	1.04e+00	13.72
deflected	diminishing	2.57e-01	5.59e-01	59.99	2.85e-01	5.21e-01	27.45	2.84e-01	5.47e-01	21.26
	fumerotv	3.63e-02	5.34e-01	44.39	5.06e-02	5.50e-01	49.79	3.16e-04	1.62e-01	32.26
	kiwieltv	2.85e-01	6.53e-01	35.17	3.02e-01	6.48e-01	36.32	3.28e-01	7.40e-01	28.37
	sheralitv	1.18e-01	7.92e-01	34.27	1.87e-01	7.75e-01	37.07	1.59e-01	8.18e-01	38.07
rva	diminishing	3.57e-01	7.43e-01	24.21	3.11e-01	8.03e-01	87.98	3.52e-01	7.52e-01	69.08
	fumerotv	3.58e-02	9.68e-01	78	3.03e-02	3.84e+00	42.43	6.84e-02	8.42e-01	24.09
	kiwieltv	1.88e-01	1.24e+00	15.78	3.21e-01	1.22e+00	18.16	2.92e-01	1.60e+00	15.35
	sheralitv	3.59e-01	1.29e+00	8.82	3.59e-01	1.28e+00	11.21	3.59e-01	1.37e+00	11.31

Tabella 4.13: Risultati numerici p50 e s136

s91		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	1.49e-04	1.41e-02	27.59	1.14e-01	5.19e-01	33.09	3.65e-04	2.86e-02	48.1
	fumerotv	5.08e-02	7.55e-02	3.77	1.14e-01	1.02e+00	32.99	1.22e-01	1.10e+00	62.09
	kiwieltv	9.49e-02	2.17e-01	69.59	1.56e-01	4.80e-01	50.03	1.80e-01	3.06e-01	13.98
	sheralitv	3.55e-01	1.34e+00	3.624	1.37e-01	8.66e-01	7.142	3.54e-01	1.22e+00	3.608
camerimifratta	diminishing	8.03e-03	2.96e-01	6.782	5.52e-03	2.41e-01	14.51	9.53e-04	1.02e+01	9.506
	fumerotv	2.86e-04	5.33e-03	7.402	2.09e-02	4.38e-01	7.268	2.79e-02	4.98e-01	22.6
	kiwieltv	4.86e-02	4.41e-02	12.68	2.88e-03	5.80e-02	62.67	4.16e-05	9.75e-02	23.67
	sheralitv	9.20e-03	9.53e-01	9.222	4.43e-02	3.80e-01	12.2	1.16e-01	4.41e-01	7.044
deflected	diminishing	4.32e-03	1.94e-01	10.37	1.56e-02	1.10e-01	11.7	5.77e-03	3.24e-01	11.14
	fumerotv	2.86e-02	2.99e-01	17.93	5.29e-05	1.56e-02	10.7	1.31e-01	3.69e-01	8.576
	kiwieltv	2.43e-01	8.52e-01	9.19	2.55e-01	5.75e-01	11.52	2.41e-01	5.51e-01	11.69
	sheralitv	1.08e-01	8.15e-01	10.73	1.10e-01	7.64e-01	16.71	1.03e-01	6.57e-01	13.44
rva	diminishing	1.54e-02	3.96e-01	29.83	1.18e-01	7.13e-01	29.59	2.08e-03	9.47e-02	32.9
	fumerotv	4.60e-02	7.23e-01	5.932	3.22e-02	1.19e+00	11.74	3.17e-02	1.12e-01	4.436
	kiwieltv	3.53e-01	1.28e+00	5.456	2.75e-01	8.48e-01	12.22	3.55e-01	1.32e+00	6.224
	sheralitv	3.55e-01	1.31e+00	3.952	3.55e-01	1.26e+00	5.19	3.55e-01	1.37e+00	5.188
s56		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	6.54e-04	3.71e-03	4.654	0.00e+00	0.00e+00	5.536	4.36e-04	9.24e-03	5.934
	fumerotv	3.10e-01	8.61e-01	1.266	2.04e-01	8.17e-01	17.12	1.91e-01	7.76e-01	12.26
	kiwieltv	1.39e-01	3.61e-01	4.716	3.59e-05	1.28e-02	8.086	1.99e-02	4.42e-01	3.924
	sheralitv	4.77e-01	1.78e+00	1.428	2.26e-01	2.58e+00	1.486	4.63e-01	1.63e+00	1.41
camerimifratta	diminishing	2.54e-02	5.19e-01	4.602	3.05e-04	1.19e-02	5.658	5.03e-05	1.86e-01	4.046
	fumerotv	0.00e+00	0.00e+00	1.468	6.61e-02	1.40e+00	1.468	7.21e-04	1.16e-02	3.124
	kiwieltv	4.96e-01	1.37e+00	1.618	5.16e-01	1.29e+00	2.518	5.06e-01	4.48e+00	2.504
	sheralitv	8.83e-02	5.79e-01	2.17	5.00e-02	5.97e-01	2.828	2.15e-01	1.47e+00	2.304
deflected	diminishing	0.00e+00	0.00e+00	3.328	1.78e-03	2.04e-01	4.24	1.24e-03	2.39e-02	3.934
	fumerotv	0.00e+00	0.00e+00	2.176	1.17e-04	8.20e-03	2.902	0.00e+00	0.00e+00	2.056
	kiwieltv	4.74e-01	1.24e+00	2.002	5.16e-01	5.07e-01	2.736	5.16e-01	1.16e+00	2.846
	sheralitv	2.86e-02	1.88e-01	4.928	3.73e-02	6.86e-01	5.712	8.85e-04	2.80e-02	3.906
rva	diminishing	3.00e-03	3.01e-02	8.624	4.82e-03	4.41e-02	16.53	1.98e-02	7.17e-01	5.42
	fumerotv	1.68e-02	2.16e+00	1.96	2.02e-02	8.43e-01	3.112	4.69e-02	5.71e-01	1.654
	kiwieltv	5.16e-01	1.69e+00	1.7	5.16e-01	3.91e+00	3.254	5.16e-01	3.18e+00	3.906
	sheralitv	5.16e-01	1.99e+00	1.34	5.16e-01	1.93e+00	1.674	5.16e-01	1.82e+00	1.678

Tabella 4.14: Risultati numerici s91 e s56

s21		0			1			2		
		dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
bva	diminishing	0.00e+00	0.00e+00	0.062	0.00e+00	0.00e+00	0.084	0.00e+00	0.00e+00	0.072
	fumerotv	1.10e-01	5.00e-01	0.13	8.42e-02	3.13e+05	0.258	5.62e-02	3.25e+00	0.214
	kiwieltv	3.33e-02	1.02e-01	0.084	1.62e-02	3.60e-01	0.112	4.17e-02	5.12e-01	0.146
	sheralitv	1.88e-01	2.13e+00	0.248	7.82e-05	4.34e-02	0.254	1.87e-01	5.02e+00	0.208
camerinifratta	diminishing	0.00e+00	0.00e+00	0.064	0.00e+00	0.00e+00	0.07	0.00e+00	0.00e+00	0.046
	fumerotv	0.00e+00	0.00e+00	0.074	3.33e-02	4.12e-01	0.058	0.00e+00	0.00e+00	0.076
	kiwieltv	0.00e+00	0.00e+00	0.04	0.00e+00	0.00e+00	0.04	0.00e+00	0.00e+00	0.05
	sheralitv	0.00e+00	0.00e+00	0.068	2.93e-03	5.58e-02	0.134	5.01e-03	6.69e-02	0.132
deflected	diminishing	0.00e+00	0.00e+00	0.096	0.00e+00	0.00e+00	0.14	0.00e+00	0.00e+00	0.098
	fumerotv	0.00e+00	0.00e+00	0.092	0.00e+00	0.00e+00	0.096	0.00e+00	0.00e+00	0.102
	kiwieltv	4.00e-12	0.00e+00	0.144	0.00e+00	0.00e+00	0.076	0.00e+00	0.00e+00	0.046
	sheralitv	0.00e+00	0.00e+00	0.096	0.00e+00	0.00e+00	0.114	0.00e+00	0.00e+00	0.124
rva	diminishing	3.40e-02	4.30e-01	0.254	0.00e+00	0.00e+00	0.278	0.00e+00	0.00e+00	0.276
	fumerotv	2.44e-02	2.93e-01	0.118	1.50e-02	2.63e-01	0.152	5.23e-02	1.48e+00	0.162
	kiwieltv	1.89e-01	1.64e+00	0.248	1.52e-01	1.96e+00	0.21	1.92e-01	2.88e+00	0.276
	sheralitv	1.92e-01	3.61e+00	0.182	1.92e-01	1.38e+00	0.228	1.92e-01	1.33e+00	0.226

Tabella 4.15: Risultati numerici s21

convergent deflected	0			1			2		
	dgap	pgap	time	dgap	pgap	time	dgap	pgap	time
c150	3.72e-03	9.93e-01	93.47	5.16e-03	1.20e+00	78.51	3.42e-03	5.00e-01	107.9
c125	3.21e-03	2.42e-01	67.29	2.98e-03	4.97e-01	90.76	2.91e-03	2.89e-01	83.37
c100	6.46e-03	6.46e-01	26.28	6.73e-03	4.53e-01	30.49	6.46e-03	5.97e-01	28.13
c75	1.28e-02	4.52e-01	12.48	8.43e-03	6.64e-01	12.62	1.42e-02	4.55e-01	12.92
c50	5.26e-03	1.61e-01	5.632	4.42e-03	7.34e-02	6.134	4.37e-03	7.04e-02	6.296
c25	3.89e-03	6.46e-02	0.884	9.11e-03	4.12e-01	0.932	1.11e-02	6.22e-01	0.89
g2g144	4.35e-02	1.97e+00	36.79	4.44e-02	1.88e+00	48.51	1.27e-02	7.14e-01	54.63
g2g100	4.10e-02	2.52e+01	5.304	3.96e-02	8.19e+00	5.698	3.29e-02	5.21e+00	5.596
g2g81	7.43e-04	6.24e-01	10.73	5.57e-04	1.45e-01	12.22	0.00e+00	0.00e+00	12.58
g2g49	0.00e+00	0.00e+00	2.06	0.00e+00	0.00e+00	3.016	0.00e+00	0.00e+00	2.792
g2g25	0.00e+00	0.00e+00	0.614	0.00e+00	0.00e+00	0.716	0.00e+00	0.00e+00	0.556
g2pm144	4.07e-02	6.83e-01	24.89	6.04e-02	2.86e+00	21.68	2.97e-02	1.77e+01	28.13
g2pm100	1.49e-02	2.54e+00	19.67	5.91e-03	3.98e-01	30.2	1.12e-02	5.66e-01	19.14
g2pm81	3.44e-02	1.21e+00	5.072	3.16e-02	7.45e-01	7.208	2.24e-02	4.58e-01	6.508
g2pm49	1.90e-02	1.88e+00	1.952	1.14e-02	3.58e-01	3.196	2.36e-03	1.19e-01	3.068
g2pm25	8.10e-04	4.21e-02	0.566	2.02e-03	1.49e-02	0.668	4.86e-04	1.72e-02	0.592
g3g125	1.13e-01	1.64e+00	11.02	2.04e-01	2.72e-01	11.18	2.09e-01	5.92e-01	10.9
g3g64	1.25e-01	4.96e+00	2.092	7.23e-02	4.17e+00	2.642	8.52e-02	4.78e+00	2.31
g3g27	2.31e-02	1.27e+00	0.31	2.59e-02	4.64e+00	0.396	8.41e-02	1.43e+00	0.306
g3pm125	2.37e-02	1.86e-01	33.62	1.30e-02	6.02e-01	35.86	1.12e-02	3.81e+00	31.24
g3pm64	4.99e-04	3.89e-02	6.668	4.31e-03	4.93e-01	7.76	7.31e-04	4.86e-02	6.536
g3pm27	2.04e-03	3.31e-02	0.886	4.29e-03	6.46e-02	0.968	2.04e-03	2.97e-02	0.908
p150	2.76e-02	8.48e-01	22.32	1.72e-02	4.80e-01	25.92	1.27e-02	5.44e-01	27.39
p100	3.22e-03	1.63e-01	10.77	1.31e-02	2.87e-01	11.18	7.36e-03	1.64e-01	10.17
p50	4.81e-03	3.38e-02	1.541	3.20e-03	3.90e-02	1.229	1.46e-03	6.49e-02	1.391
s136	2.51e-01	5.68e-01	76.7	2.82e-01	5.15e-01	40.54	2.83e-01	5.53e-01	31.98
s91	8.00e-02	4.07e-01	8.534	6.82e-03	2.13e-01	10.17	1.10e-02	1.58e-01	7.468
s56	0.00e+00	0.00e+00	3.216	6.77e-04	1.33e-02	4.694	6.20e-04	1.61e-02	4.792
s21	4.00e-12	0.00e+00	0.098	4.00e-12	0.00e+00	0.104	4.00e-12	0.00e+00	0.118

Tabella 4.16: Risultati numerici convergent deflected, diminishing

istanza	dgap	pgap	time	istanza	dgap	pgap	time
c150	2.54e-09	1.70e-06	396.1	g2g144	2.89e-09	2.59e-06	64.77
c125	3.57e-09	2.02e-06	220.1	g2g100	6.19e-10	2.39e-07	12.7
c100	6.44e-09	1.06e-06	81.62	g2g81	4.81e-10	2.49e-08	4.838
c50	7.93e-09	6.99e-07	6.212	g2g49	0.00e+00	0.00e+00	0.72
c75	6.50e-09	8.55e-07	26.56	g2g25	0.00e+00	0.00e+00	0.076
c25	3.32e-09	1.66e-07	0.394				
g2pm144	1.62e-04	4.01e-03	4424	g3g125	4.32e-09	3.18e-06	119.6
g2pm100	1.73e-09	4.95e-08	210.7	g3g64	0.00e+00	0.00e+00	2.046
g2pm49	0.00e+00	0.00e+00	9.756	g3g27	0.00e+00	0.00e+00	0.12
g2pm25	0.00e+00	0.00e+00	0.48				
g2pm81	1.34e-09	1.96e-08	249.9				
g3pm125	0.00e+00	0.00e+00	976.8	p150	5.93e-10	2.16e-02	36.15
g3pm27	2.96e-09	4.13e-08	1.036	p100	5.31e-10	2.01e-01	12.68
g3pm64	1.36e-09	2.95e-09	5.328	p50	4.63e-10	3.42e-03	1.424
s136	0.00e+00	4.37e-02	62.57				
s21	0.00e+00	0.00e+00	0.048				
s56	0.00e+00	0.00e+00	2.258				
s91	1.34e-09	1.17e-01	8.778				

Tabella 4.17: Risultati numerici bundle

# Conclusioni

In questo documento abbiamo studiato approfonditamente i metodi del subgradiente per l'ottimizzazione non differenziabile. Il lavoro è stato caratterizzato da un importante studio bibliografico volto a recensire le più comuni varianti di questa classe di algoritmi, sintetizzarli in un unico schema algoritmico, sviluppare un'applicazione per la risoluzione di problemi NDO ed effettuare uno studio numerico comparativo.

Una parte significativa dello studio teorico è stata dedicata allo sviluppo di un'analisi di convergenza per diversi metodi del subgradiente condizionale deflesso. La necessità di una teoria di convergenza per questi metodi nasce sia dallo sforzo di sintesi dei diversi algoritmi che dalla necessità di progettare metodi convergenti il più possibile efficienti. Uno degli obiettivi dello studio di convergenza è stato la progettazione di metodi implementabili. In altre parole i nostri teoremi di convergenza presuppongono condizioni di immediata implementazione e non richiedono un'ulteriore analisi teorica. In aggiunta abbiamo sviluppato un'analisi di convergenza del tutto innovativa per il metodo condizionale deflesso con stepsize di tipo square summable.

La nostra analisi di convergenza può dirsi completamente innovativa per quel che riguarda l'integrazione dei metodi condizionale e deflesso. Tuttavia questo passaggio, seppur necessario, può sembrare soltanto formale dal momento che sappiamo che il metodo condizionale non peggiora le proprietà di convergenza del metodo deflesso. D'altra parte anche questa considerazione è parte della nostra analisi e in sostanza si tratta del lemma 7. Ad ogni modo molte delle analisi di convergenza note per il metodo deflesso presuppongono delle condizioni difficili che le rendono di scarsa utilità. Ad esempio quando si utilizzano gli stepsize di Polyak le analisi in letteratura (ad esempio [18]) necessitano della condizione (2.30); la nostra condizione, invece, è più restrittiva perché imponiamo  $\beta_k \leq \alpha_k \leq 1$  ma non richiede sforzi aggiuntivi per essere utilizzata. Allo stesso modo per ottenere metodi convergenti utilizzando stepsize square summable bisogna considerare la condizione  $\varepsilon_k = o(\nu_k)$  che deriva dall'analisi di convergenza del metodo approssimato. Anche qui la nostra condizione è più restrittiva, perché imponiamo una limitazione sul parametro  $\alpha_k$ , ma più semplice da implementare; d'altra parte abbiamo già discusso nel capitolo 2 la necessità di una condizione sul parametro di deflessione. Inoltre il metodo 4 è completamente innovativo e in un certo senso originale; infatti utilizziamo metodi di approssimazione propri degli stepsize target value per garantire la convergenza di una strategia di deflection. I metodi deflessi che utilizzano stepsize target value, invece, sono maggiormente coperti dalle dimostrazioni in letteratura. In particolare il metodo target value di Sherali che utilizza una direzione deflessa è descritto in [6] mentre il metodo target value di Kiwiel (noto anche come *level*

*algorithm*) nel caso deflesso è trattato in [24]. Quest'ultimo articolo è di pubblicazione molto recente e in effetti noi abbiamo sviluppato l'analisi di convergenza per lo stesso metodo al capitolo 2 prima di venire a conoscenza di esso. Comune la nostra analisi rimane interessante in prima battuta perché consideriamo il metodo condizionale che [24] non tratta e secondariamente perché presentiamo una dimostrazione di convergenza più semplice. In particolare [24] estende al caso deflesso una dimostrazione di Goffin-Kiwiel (cfr. [2]) mentre noi estendiamo al caso condizionale deflesso una dimostrazione di Bertsekas-Nedic (cfr. [11]).

Un altro aspetto che ci preme sottolineare è lo studio computazionale comparativo degli algoritmi discussi in questo documento e implementati. L'analisi dei risultati numerici indica vari metodi che sembrano essere più efficienti rispetto agli altri. In particolare riscontriamo l'utilità delle tecniche di deflection per accelerare la convergenza dal punto di vista duale e delle rivistazioni dei metodi del volume per la produzione di soluzioni primali. Possiamo anche dire che i metodi del subgradiente e quelli del volume si equivalgono, nel senso che presentano comportamenti diversi e rispondono ad esigenze diverse; in particolare i primi ottengono una buona convergenza duale ma soluzioni primali poco accurate mentre i secondi producono buone soluzioni primali a discapito di quelle duali. I metodi più promettenti per queste due classi sembrano essere rispettivamente quello di Camerini-Fratta e BVA. D'altra parte il metodo deflesso sembra essere un buon compromesso tra le due esigenze e, sebbene sia spesso dominato dai due metodi appena menzionati, converge in maniera soddisfacente dal punto di vista duale ed in maniera accettabile dal punto di vista primale. Alla luce di questi risultati anche i metodi del subgradiente deflesso convergenti assumono interesse. In particolare, sebbene la condizione (1.28) eviti il presentarsi di fenomeni di non convergenza, può accadere in alcuni casi che il metodo deflesso non converga; in queste situazioni quello che abbiamo chiamato metodo 4 o uno dei metodi deflesso con *stepsize target value* garantiscono la convergenza e "buone" prestazioni.

Il lavoro presentato in questo documento è aperto a diversi sviluppi. Nelle applicazioni spesso si usano metodi del subgradiente che non hanno proprietà teoriche di convergenza, mentre quelli che le hanno risultano poco efficienti, lo sforzo di completamento delle analisi di convergenza è una delle possibili direzioni dell'attività di ricerca in questo campo. Un altro aspetto molto studiato anche di recente riguarda la generazione di soluzioni primali, si tratta infatti di uno degli aspetti in cui gli algoritmi del subgradiente sono più carenti, soprattutto per quanto riguarda le proprietà teoriche di convergenza. Infine un possibile sviluppo riguarda i metodi del subgradiente incrementale. In particolare abbiamo implementato dei metodi incrementali che utilizzano approssimazioni sui subgradienti e sui valori obiettivo evitando il calcolo ad ogni iterazione i valori esatti, operazione computazionalmente costosa, e un metodo del subgradiente incrementale deflesso; entrambi questi metodi sono sprovvisti di proprietà teoriche di convergenza.

# Appendice A

## Algoritmi proposti in letteratura

In questa sezione cercheremo di recensire in dettaglio i metodi proposti in letteratura e presenti in bibliografia, evidenziando come essi si possano ricondurre allo schema generico presentato. Questi metodi sono generalmente supportati da dimostrazioni di convergenza e, comunque, sempre sottoposti a test computazionali che ne analizzano le prestazioni in applicazioni comuni descritti in bibliografia.

In particolare ci concentreremo sui diversi metodi disponibili per calcolare lo stepsize spiegando di ognuno le motivazioni e le proprietà. Tratteremo, inoltre, i metodi del volume modificati. Tra i metodi del volume questi sono gli unici supportati da proprietà teoriche di convergenza.

### A.1 Polyak stepsizes

I cosiddetti *Polyak stepsizes* calcolano la lunghezza del passo come:

$$\nu_k = \beta_k \frac{f(y_k) - f^*}{\|d_k\|^2}, \quad \beta \in [\beta_{min}, \beta_{max}] \subset (0, 2).$$

Furono introdotti da Polyak in [15], che rappresenta uno dei lavori pionieristici nel campo dell'ottimizzazione non differenziabile.

Se  $\beta_k = 1 \forall k$  e la funzione è differenziabile il metodo si comporta esattamente come il metodo di Newton per il calcolo delle radici della funzione  $f(y) - f^*$ . In altre parole, dato il modello lineare della funzione:

$$\phi_k(y) = f(y_k) + d_k(y - y_k),$$

l'algoritmo calcola la soluzione successiva come

$$y_{k+1} : \phi_k(y_{k+1}) = f^*.$$

Questo tipo di stepsize rappresenta senza dubbio la migliore scelta per i metodi del subgradiente, posta ovviamente la conoscenza del valore ottimo  $f^*$ .

Infatti Polyak dimostra, per il metodo del subgradiente ordinario, la proprietà di convergenza geometrica, vale a dire:

$$\|y_k - y^*\| \leq Cq^k, \quad \text{con } C, q \text{ costanti.}$$

Questa proprietà, nonostante non considerata nel nostro studio teorico, sembra conservarsi in molti degli altri metodi considerati.

## A.2 Target Value stepsizes

I metodi Target Value sono trattati in [1], [2], [3] e consistono nella scelta del passo come:

$$\nu = \beta \frac{f(y) - f_{lev}}{\|d\|^2}, \quad \beta \in (0, 2); \quad (\text{A.1})$$

Abbiamo già visto come spesso la miglior scelta dello stepsize sia costituita dagli stepsize di Polyak; tuttavia questi metodi sono raramente implementabili poiché richiedono la conoscenza a priori del valore ottimo della funzione obiettivo. I metodi target value sono un'approssimazione di questi: sostituiscono  $f^*$  con un target variabile  $f_{lev}$  che, al limite, converge al valore ottimo.

L'idea su cui si basano questi algoritmi è semplice: dato un valore detto target level  $f_{lev}$  ci si sposta verso l'insieme:

$$H_k = \{z : f(y_k) + d_k(z - y_k) \leq f_{lev}^k\} \quad (\text{A.2})$$

$H(y)$  approssima, quindi, l'insieme detto *level set* della funzione  $f$ :

$$\mathcal{L}(f_{lev}^k) = \{y : f(y) \leq f_{lev}^k\}. \quad (\text{A.3})$$

Più formalmente, se  $d_k$  è un subgradiente, la relazione che intercorre tra i due insiemi è:  $\mathcal{L}(f_{lev}^k) \subseteq H_k$ . Ad ogni modo  $f_{lev}^k$  non è sempre un'approssimazione superiore del valore ottimo, dunque può accadere  $\mathcal{L}(f_{lev}^k) = \emptyset$ .

Lo spostamento ad ogni iterazione è dato dalla seguente:

$$\hat{y} := y + \beta(P_{H(y)}(y) - y) = y + \beta \left( \frac{f(y) - f_{lev}}{\|d\|^2} \right) d \quad (\text{A.4})$$

Il valore  $f_{lev}^k$ , quindi, è un'approssimazione del valore ottimo della funzione. Nel processo di approssimazione, però, non si conserva la proprietà di convergenza geometrica.

Esistono in letteratura diversi metodi per calcolare  $f_{lev}^k$ ; ad esempio [2] e [3]. Questo sembra essere l'aspetto più critico di questi stepsize: si solito  $f_{lev}^k$  non è aggiornato ad ogni iterazione e può in breve tempo diventare inadeguato. In altre parole l'algoritmo si può trovare presto a muoversi "poco": un comportamento necessario nelle vicinanze dell'ottimo ma deleterio in altri punti. Dunque una scelta non attenta delle condizioni di aggiornamento di  $f_{lev}^k$  può portare ad un rallentamento artificiale dell'algoritmo.

Ad esempio nel metodo proposto in [2] e discusso per il metodo condizionale deflesso nel capitolo 2 il parametro critico è la limitazione  $R$  sul percorso  $\sigma$ : un valore del parametro troppo elevato per il problema in esame può rallentare di molti ordini di grandezza le iterazioni necessarie alla convergenza subottimale.

### A.3 Proprietà teoriche dell'algoritmo del volume

L'algoritmo del volume, esposto in [5], è un metodo efficiente, ampiamente utilizzato nelle applicazioni; tuttavia esso manca di proprietà teoriche di convergenza. In [7] si trova un'analisi delle proprietà di questi metodi che porta, poi, a definire due metodi convergenti: BVA ed RVA.

Il metodo del volume costruisce una direzione  $d_k$  aggregando subgradienti nelle soluzioni generate nei null step, che sono calcolate a partire dal centro di stabilità  $\tilde{y}_k$ . Ovviamente, quindi, il metodo del volume non utilizza subgradienti esatti come direzione. Possiamo interpretare  $d_k$  in più modi: se calcoliamo l'approssimazione di  $d_k$  come subgradiente in  $\tilde{y}_k$  otteniamo l'algoritmo BVA, se interpretiamo  $d_k$  come extragradiente, ovvero subgradiente in un punto diverso da  $\tilde{y}_k$  o  $y_k$ , otteniamo l'algoritmo RVA.

L'algoritmo RVA calcola ad ogni iterazione la seguente quantità:

$$p_k = \alpha_k y_k + (1 - \alpha_k) y_{k-1}. \quad (\text{A.5})$$

Per ogni algoritmo del volume vale:

$$\begin{aligned} d_k &\in \partial_{\varepsilon_k} f(p_k) \\ \varepsilon_{k+1} &= \alpha_k (1 - \alpha_k) (g_k - d_k)(p_k - y_k) + (1 - \alpha_k) \varepsilon_k; \\ \varepsilon_0 &= 0. \end{aligned}$$

In alternativa, seguendo il metodo utilizzato in BVA, si può calcolare l'errore su  $d_k$  come subgradiente in  $\tilde{y}_k$ . Sia  $\varepsilon_{k-1}$  tale che  $d_{k-1} \in \partial_{\varepsilon_{k-1}} f(\tilde{y}_k)$  e  $e_k$  tale che  $g_k \in \partial_{e_k} f(\tilde{y}_k)$ , allora

$$\varepsilon_k = \alpha_k e_k + (1 - \alpha_k) \varepsilon_{k-1}.$$

Dove  $e_k$  può essere calcolato come:

$$e_k = f(y_k) + g_k(y_k - \tilde{y}_k) - f(\tilde{y}_k).$$

Una volta calcolato l'errore in  $\tilde{y}_k$ ,  $d_k$  può essere a sua volta "spostato" in  $y_k$ ; questo può essere utile, ad esempio per utilizzare  $d_k$  nelle condizioni di terminazione.

Dunque [7] ci permette di interpretare ogni iterazione del metodo del volume: come nei metodi bundle abbiamo a disposizione un modello economico della funzione obiettivo costituito da:

$$\hat{f}_k(y) = \max\{f(y_k) + g_k(y - y_k), f(\tilde{y}_k) + d_k(y - p_k) + \varepsilon_k\}$$

In [7] questo modello viene utilizzato per calcolare, come nei bundle methods, la nuova soluzione introducendo una misura sul valore obiettivo (detta *expected gain*) utilizzata per discriminare tra *serious step* e *null step* e calcolando opportunamente il parametro di deflessione.

Ad ogni iterazione, per determinare il coefficiente  $\alpha_k$ , viene risolto il seguente problema:

$$\begin{aligned} \min_{\alpha \in [0,1]} \frac{\nu_k}{2} \|\alpha g_k + (1 - \alpha) d_{k-1}\|^2 + \alpha e_k + (1 - \alpha) \hat{e}_k, \\ \hat{e}_k = f(p_{k-1}) + d_{k-1}(p_{k-1} - \tilde{y}_k) - f(\tilde{y}_k) + \varepsilon_k. \end{aligned}$$

L'*expected gain*  $\delta_k$ , invece, viene calcolato come

$$\delta := f(\tilde{y}_k) - \hat{f}(y_{k+1}).$$

Questa appena descritta sarebbe una versione *bundle-like* dell'algoritmo del volume; tuttavia richiede la conoscenza del valore  $f(p_k)$ , non calcolato dai metodi del volume, che può generare una forte penalizzazione in termini di efficienza. Per questo motivo [7] propone delle approssimazioni che evitano il calcolo di  $f(p_k)$ : gli algoritmi RVA e BVA già presentati nel capitolo 3.

**RVA - Revised Volume Algorithm** Si utilizza l'approssimazione  $f(p_k) \approx f(\tilde{y}_k)$  e si sostituiscono le quantità  $e$  ed  $\hat{e}$  con le seguenti:

$$e_k \approx E = g_k(\tilde{y}_k - y_k), \quad \hat{e}_k \approx \hat{E}_k = d_k(\tilde{y}_k - p_k) + \varepsilon_k \quad (\text{A.6})$$

**BVA - Bundle Volume Algorithm** In questo algoritmo si interpretano le direzioni come subgradienti approssimati in  $\tilde{y}_k$ .

Si calcola, quindi, l'*expected gain* come:

$$\delta_k = \nu_k \|d_k\|^2 + \varepsilon_k \quad (\text{A.7})$$

e si sostituisce  $\hat{e}_k$  con  $\varepsilon_k$ .

Poiché in questo metodo  $d_k \in \partial_{\varepsilon_k} f(\tilde{y}_k)$  possiamo agevolmente calcolare  $\xi_k : d_k \in \partial_{\xi_k} f(y_k)$ :

$$\xi_k = f(y_k) - d_k(y_k - \tilde{y}_k) - f(\tilde{y}_k) + \varepsilon_k. \quad (\text{A.8})$$

# Bibliografia

- [1] Krzysztof C. Kiwiel: *Convergence of approximate and incremental subgradient methods for convex optimization*. SIAM J. Optim. Vol.14, 2004.
- [2] Jean-Louis Goffin, Krzysztof C. Kiwiel: *Convergence of a simple subgradient level method*. Mathematical Programming 85, 1999.
- [3] Hanif D. Sherali, Gyunghyun Choi, Cihan H. Tuncbilek: *A variable target value method for nondifferentiable optimization*. Operations Research Letters 26, 2000.
- [4] Francisco Barahona, Ranga Anbil: *The volume algorithm: producing primal solutions with a subgradient method*. Mathematical Programming 87, 2000.
- [5] Francisco Barahona: *On the Volume Algorithm and some combinatorial linear programs*. 1999.
- [6] Hanif D. Sherali, Churlzu Lim: *On embedding the volume algorithm in a variable target value method*. Operations Research Letters 32, 2004.
- [7] L. Bahiense, N. Maculan, C. Sagastizabal, et al.: *The volume algorithm revisited: relation with bundle methods*. Math. Programming 94, 2002.
- [8] Francesca Fumero: *A modified subgradient algorithm for Lagrangean relaxation*. Computers & Operations Research 28, 2001.
- [9] Torbjorn Larsson, Michael Patriksson, Ann-Brith Stromberg: *On the convergence of conditional  $\varepsilon$ -subgradient methods for convex programs and convex-concave saddle-point problems*. European Journal of Operational Research 151, 2003.
- [10] Krzysztof C. Kiwiel, Torbjorn Larsson, P.O. Lindberg: *Dual properties of ballstep subgradient methods, with application to lagrangian relaxation*. 1999.
- [11] Angelia Nedić, Dimitri P. Bertsekas: *Incremental subgradient methods for nondifferentiable optimization*. SIAM J. Optim., Vol. 12, 2001.
- [12] Torbjorn Larsson, Michael Patriksson, Ann-Brith Stromberg: *Conditional subgradient optimization - Theory and applications*. European Journal of Operational Research 88, 2003.

- [13] Rockafellar R.T.: *The theory of subgradients and its applications to problems of optimization: convex and nonconvex functions*. Heldermann Verlag, Berlin, 1981.
- [14] Behanu Guta: *Subgradient Optimization Methods in Integer Programming with an Application to a Radiation Therapy Problem*: Tesi di dottorato, Università tecnica di Kaiserslautern, 2003.
- [15] B.T. Polyak: *Minimization of unsmooth functionals*: USSR Computational Mathematics and Mathematical Physics 9, 1969.
- [16] Dimitri P. Bertsekas: *Convex Analysis and Optimization*. Athena Scientific, 2003.
- [17] Antonio Frangioni: *Dual Ascent Methods and Multicommodity Flow Problems*, Tesi di dottorato, Università di Pisa, 1997.
- [18] Ulf Brannlund: *A generalized subgradient method with relaxation step*, Mathematical Programming 71, 1995.
- [19] Krzysztof C. Kiwiel: *The efficiency of subgradient projection methods for convex optimization, Part I: general level methods*, SIAM J. Control Optim. 34, 1996.
- [20] Krzysztof C. Kiwiel: *The efficiency of subgradient projection methods for convex optimization, Part II: Implementations and extensions*, SIAM J. Control Optim. 34, 1996.
- [21] P.M. Camerini, L. Fratta, F. Maffioli: *On improving relaxation methods by modified gradient techniques*, Mathematical Programming Study 3, 1975.
- [22] T. Larsson, M. Patriksson, A.B. Stromberg: *Ergodic convergence in subgradient optimization*, 1995.
- [23] A. Frangioni, A. Lodi, G. Rinaldi: *New approaches for optimizing over the semimetric polytope*, Mathematical Programming 104, 2005.
- [24] Churlzu Lim, Hanif D. Sherali: *Convergence and Computational Analyses for Some Variable Target Value and Subgradient Deflection Methods*, Computational Optimization and Applications 34, 2006.
- [25] T. Larsson, M. Patriksson, A.B. Stromberg: *Ergodic, primal convergence in dual subgradient schemes for convex programming*, Mathematical Programming 86, 1999.