

# Approccio Frank-Wolfe per la risoluzione di un problema con variabili semicontinue

Francesco Milizia  
Università di Pisa, Dipartimento di Matematica  
Laboratorio computazionale, a.a. 2017/2018

19 dicembre 2017

## 1 Presentazione del problema

Vogliamo studiare il seguente problema di ottimizzazione:

$$\min x^T Q x \tag{1}$$

$$\sum_{i=1}^n x_i = 1 \tag{2}$$

$$\sum_{i=1}^n \mu_i x_i \geq \rho \tag{3}$$

$$l_i y_i \leq x_i \leq u_i y_i \quad i \in \{1, \dots, n\} \tag{4}$$

$$y_i \in \{0, 1\} \quad i \in \{1, \dots, n\} \tag{5}$$

dove  $\mu \in \mathbb{R}^n$ ,  $\rho$ ,  $l_i$  ed  $u_i$  sono numeri reali che possiamo supporre positivi e infine  $Q$  è una matrice simmetrica  $n \times n$  semidefinita positiva. Le variabili  $y_i$  codificano la scelta di “attivare” le corrispondenti variabili  $x_i$ . Infatti se  $y_i = 0$  deve essere necessariamente  $x_i = 0$ , mentre se  $y_i = 1$  deve essere  $x_i \in [l_i, u_i]$ .

Per risolvere problemi di questo tipo si rende necessaria una qualche riformulazione in cui si rilassa la condizione di integralità delle variabili. In questo modo si allarga l’insieme delle soluzioni ammissibili e quindi il valore minimo della funzione obiettivo si abbassa. Quindi con questo approccio si ottengono delle valutazioni inferiori del valore ottimo del problema di partenza. Il nostro scopo è trovare una formulazione che consenta di ottenere valutazioni inferiori quanto più vicine possibili al valore ottimo originale.

Purtroppo è difficile che la semplice sostituzione della condizione (5) con il rilassamento continuo  $y_i \in [0, 1]$  porti a valutazioni inferiori di qualità, quindi si rende necessario studiare delle tecniche più raffinate.

## 2 Il caso diagonale

In questa sezione supponiamo che la matrice  $Q$  sia diagonale e che quindi si possa scrivere  $x^T Q x = \sum_{i=1}^n d_i x_i^2$ . Una tecnica che in questo caso consente

di ottenere degli ottimi risultati è la cosiddetta *riformulazione prospettica* (perspective reformulation, o PR in breve). Nel nostro caso consideriamo la funzione obiettivo  $f(x, y) = \sum_{i=1}^n d_i x_i^2$  definita nell'insieme

$$\chi = \{(x, y) \in \mathbb{R}^n \times \{0, 1\}^n : l_i y_i \leq x_i \leq u_i y_i\}$$

(che contiene l'insieme delle soluzioni ammissibili) e troviamo l'involuppo convesso<sup>1</sup>  $\bar{f}$ , che sarà definito sull'insieme strettamente più grande

$$\bar{\chi} = \{(x, y) \in \mathbb{R}^n \times [0, 1]^n : l_i y_i \leq x_i \leq u_i y_i\}$$

Nel caso più semplice in cui  $n = 1$  si ottiene facilmente  $\bar{f}(x, y) = d_1 \frac{x^2}{y}$ , che in linea di principio è definita solo per  $y \in (0, 1]$  ma si estende per continuità ponendo  $\frac{x^2}{y} = 0$  quando  $y = 0$  in virtù del vincolo (4). Visto che la funzione  $f$  si può scrivere come  $f(x, y) = \sum_{i=1}^n f_i(x_i, y_i)$ , nel caso generale vale

$$\bar{f}(x, y) = \sum_{i=1}^n d_i \frac{x_i^2}{y_i}$$

Notiamo che  $\bar{f} \geq f$  su tutto l'insieme  $\bar{\chi}$  e che nel dominio di partenza  $\chi$  le due funzioni coincidono. Quindi la soluzione del nostro problema non cambia se si sostituisce con  $\bar{f}$  la funzione obiettivo (1), però la valutazione inferiore che si ottiene per rilassamento continuo delle  $y_i$  non può che migliorare.

Ciò che abbiamo visto consente di fare dei progressi anche nel caso generale in cui  $Q$  non è diagonale. Infatti, se si trova una matrice diagonale  $D$  con elementi non negativi tale che  $Q - D$  continua ad essere semidefinita positiva, si può riscrivere la funzione obiettivo:

$$\sum_{i=1}^n d_i x_i^2 + x^T (Q - D) x$$

dove  $d_i \geq 0$  sono gli elementi della diagonale di  $D$ . Ora, se al posto di  $\sum_{i=1}^n d_i x_i^2$  sostituiamo  $\sum_{i=1}^n d_i \frac{x_i^2}{y_i}$ , otteniamo una funzione obiettivo che fornisce, col rilassamento continuo, valutazioni inferiori del valore ottimo migliori di quelle derivanti dalla semplice sostituzione del vincolo (5) con  $y_i \in [0, 1]$ . Chiaramente cambiando la diagonale estratta  $D$  cambia anche la valutazione ottenuta, e in un certo senso possiamo aspettarci di ottenere dei risultati migliori estraendo una diagonale "più grande possibile".

### 3 Il caso $n = 2$

Vogliamo ora trattare il caso in cui  $n = 2$ , per cui possiamo scrivere la funzione obiettivo:

$$q_1 x_1^2 + q_2 x_2^2 + 2q_{12} x_1 x_2$$

Ciò che impedisce di adottare una tecnica simile a quella vista nella sezione precedente è la presenza del termine misto  $2q_{12} x_1 x_2$  che coinvolge le variabili  $x_1$  e  $x_2$ , le quali sono governate da due diverse variabili binarie. Il modo di ovviare

<sup>1</sup>la funzione che ha come sopragrafico l'involuppo convesso del sopragrafico di  $f$

a questo inconveniente è definire delle nuove variabili legate alle precedenti dalle seguenti relazioni:

$$\begin{aligned}x_1 &= x_1^1 + x_1^{12} \\x_2 &= x_2^2 + x_2^{12} \\y_1 &= y^1 + y^{12} \\y_2 &= y^2 + y^{12}\end{aligned}$$

e soggette ai seguenti vincoli:

$$\begin{aligned}l_i y^i &\leq x_i^i \leq u_i y^i && i \in \{1, 2\} \\l_i y^{12} &\leq x_i^{12} \leq u_i y^{12} && i \in \{1, 2\} \\y^1 + y^2 + y^{12} &\leq 1 \\y^1, y^2, y^{12} &\in \{0, 1\}\end{aligned}$$

In questo modo le variabili  $y^1, y^2, y^{12}$  codificano quali tra le variabili  $x_1$  e  $x_2$  sono “attive” e quali no, in corrispondenza alle tre diverse possibili situazioni diverse da quella banale in cui tutte le variabili sono nulle. A questo punto, usando il fatto che  $x_i^i$  e  $x_i^{12}$  non possono essere contemporaneamente diverse da 0, la funzione obiettivo si riscrive nel seguente modo:

$$q_{11}(x_1^1)^2 + q_{22}(x_2^2)^2 + q_{11}(x_1^{12})^2 + 2q_{12}x_1^{12}x_2^{12} + q_{22}(x_2^{12})^2$$

ed ora le variabili binarie  $y^1, y^2, y^{12}$  governano addendi distinti. Infine non è difficile ottenere la riformulazione prospettica in cui la funzione obiettivo diventa:

$$\frac{q_{11}(x_1^1)^2}{y^1} + \frac{q_{22}(x_2^2)^2}{y^2} + \frac{q_{11}(x_1^{12})^2 + 2q_{12}x_1^{12}x_2^{12} + q_{22}(x_2^{12})^2}{y^{12}}$$

A questo punto vale lo stesso discorso fatto al termine della sezione precedente: nel caso generale ( $n > 2$ ) si riscrive il problema di partenza estraendo dalla matrice  $Q$  alcune matrici simmetriche semidefinite positive aventi elementi non nulli esclusivamente nelle posizioni  $(h, h), (k, k), (h, k), (k, h)$  con l'accortezza di assicurarsi che il residuo continui ad essere semidefinito positivo, e poi si modifica come abbiamo appena visto il termine della funzione obiettivo relativa a queste matrici estratte.

## 4 Ricerca della migliore decomposizione

Naturalmente è possibile adottare simultaneamente le due tecniche viste nelle sezioni precedenti, ovvero si cerca di decomporre

$$Q = R + D + \sum_{(h,k) \in P} X^{hk}$$

come somma di matrici simmetriche semidefinite positive, dove  $D$  è diagonale,  $P \subseteq \{1, \dots, n\}^2$  è un insieme di coppie di indici e  $X^{hk}$  è con entrate tutte nulle ad esclusione di quelle in posizioni  $(h, h), (k, k), (h, k), (k, h)$ . In linea di

principio considerare  $D$  è superfluo, in quanto può essere inglobato dentro le matrici  $X^{hk}$ . Tuttavia, visto che non vogliamo far crescere troppo il numero di variabili (ovvero non vogliamo considerare troppe matrici del tipo  $X^{hk}$ ), conviene tenere questo termine a parte.

Al variare della decomposizione possiamo ottenere una valutazione inferiore per la soluzione del problema di partenza: sia  $\phi(D, X)$  tale valutazione, ovvero

$$\begin{aligned} \phi(D, X) = \min \quad & x^T \left( Q - D - \sum_{(h,k) \in P} X^{hk} \right) x + \sum_{i=1}^n d_i \frac{(x_i)^2}{y_i} + \\ & + \sum_{(h,k) \in P} \left( \frac{X_{hh}^{hk} (x_{h'}^{hk})^2}{y_h^{hk}} + \frac{X_{kk}^{hk} (x_{k''}^{hk})^2}{y_k^{hk}} + \frac{X_{hh}^{hk} (x_{h'}^{hk})^2 + 2X_{hk}^{hk} x_{h'}^{hk} x_{k''}^{hk} + X_{kk}^{hk} (x_{k''}^{hk})^2}{y^{hk}} \right) \end{aligned}$$

con i vincoli seguenti per ogni  $(h, k) \in P$  oltre a quelli originali (2), (3) e (4):

$$x_i = x_{i'}^{hk} + x_{i''}^{hk} \quad i \in \{h, k\} \quad (6)$$

$$y_i = y_i^{hk} + y^{hk} \quad i \in \{h, k\} \quad (7)$$

$$l_i y_i^{hk} \leq x_{i'}^{hk} \leq u_i y_i^{hk} \quad i \in \{h, k\} \quad (8)$$

$$l_i y^{hk} \leq x_{i''}^{hk} \leq u_i y^{hk} \quad i \in \{h, k\} \quad (9)$$

$$y_h^{hk} + y_k^{hk} + y^{hk} \leq 1 \quad (10)$$

$$y_h^{hk}, y_k^{hk}, y^{hk} \in [0, 1] \quad (11)$$

Un problema di minimizzazione di questo tipo può essere passato a un solutore di problemi SOCP (dopo le opportune manipolazioni).

Visto che siamo interessati alla valutazione inferiore più vicina possibile al valore ottimo del problema di partenza, quello che vorremmo fare è risolvere

$$\max_{(D, X)} \phi(D, X) \quad (12)$$

Ci sono due importanti osservazioni da fare: la prima è che l'insieme delle decomposizioni  $(D, X)$  ammissibili è convesso, visto che le matrici semidefinite positive sono stabili rispetto alle combinazioni convesse. La seconda è che la funzione  $\phi$  appena definita è concava, e questo è vero perché  $\phi$  è puntualmente il minimo di un certo insieme (infinito) di funzioni lineari in  $D$  e  $X$ . Più concretamente, dalla definizione stessa di  $\phi$  è facile ricavare un *supergradiente*  $g$ , il cui calcolo è immediato una volta conosciuti i valori  $x_i, y_i, x_{h'}^{hk}, x_{h''}^{hk}, y_h^{hk}, y^{hk}$  che realizzano il minimo. Infatti basta prendere come componenti di  $g$  le derivate parziali rispetto a  $d_i, X_{hh}^{hk}, X_{kk}^{hk}, X_{hk}^{hk}$  della funzione minimizzata da  $\phi$ , che valgono rispettivamente:

$$\begin{aligned} \frac{(x_i)^2}{y_i} - (x_i)^2 & \quad \frac{(x_{h'}^{hk})^2}{y_h^{hk}} + \frac{(x_{h''}^{hk})^2}{y^{hk}} - (x_h)^2 \\ \frac{(x_{k'}^{hk})^2}{y_k^{hk}} + \frac{(x_{k''}^{hk})^2}{y^{hk}} - (x_k)^2 & \quad \frac{2x_{h'}^{hk} x_{k''}^{hk}}{y^{hk}} - 2x_h x_k \end{aligned}$$

Tutto ciò rende possibile l'utilizzo dell'algoritmo di Frank-Wolfe per la soluzione di (12). Tale algoritmo fornisce iterativamente soluzioni  $(D, X)$  sempre migliori, come schematizzato qui di seguito:

1. Si parte da una scomposizione  $(D_0, X_0)$  ammissibile. Si pone  $i = 0$ .
2. Si calcola  $\phi$  e il supergradiente  $g$  nel punto  $(D_i, X_i)$ .
3. Si trova  $(D^+, X^+)$  che massimizza la funzione lineare  $g$ .
4. Nel segmento tra  $(D_i, X_i)$  e  $(D^+, X^+)$  si cerca il punto  $(D_{i+1}, X_{i+1})$  che massimizza  $\phi$ .
5. Si incrementa  $i$  e si ritorna al secondo punto.

Per quanto riguarda la scomposizione iniziale, possiamo scegliere per semplicità  $X_0 = 0$  e una qualsiasi diagonale  $D_0$  tale che  $Q - D_0$  sia semidefinita positiva. Idealmente vorremmo avere  $D_0$  quanto più grande possibile, ed una scelta opportuna (quella che si è scelto di implementare) potrebbe essere quella di prendere  $D_0$  con somma degli elementi diagonali maggiore. In pratica si tratta di eseguire il punto 3 dell'algoritmo come appena schematizzato in cui al posto di  $g$  c'è un vettore con componenti tutte uguali a 1 in corrispondenza degli elementi di  $D_0$ .

## 5 La scelta delle coppie di indici

Ciò che finora risulta poco chiaro è come scegliere l'insieme di coppie  $P$  da usare per costruire la decomposizione. L'approccio che scegliamo di seguire fa variare l'insieme  $P$  ad ogni iterazione, a seconda della soluzione  $(x, y)$  trovata al passo precedente. In particolare, inizialmente viene posto  $P = \emptyset$ . Alle iterazioni successive, prima di eseguire il punto 2, si guarda il vettore  $y$  ottenuto al passo precedente: quindi si introducono in  $P$  le coppie  $(h, k)$  per cui le variabili  $y_h, y_k$  sono correntemente diverse da 0 e 1 (ovvero non sono intere). L'idea è che in questo modo le soluzioni non intere vengono "penalizzate" (in fin dei conti vogliamo trovare una soluzione più vicina possibile a quella intera ottima).

In pratica, invece di mantenere un insieme  $P \subseteq \{1, \dots, n\}^2$ , nel corso dell'algoritmo viene mantenuto un insieme di indici  $I \subseteq \{1, \dots, n\}$  sottintendendo che si considera implicitamente  $P = \{(h, k) \in I^2 : h < k\}$ . All'inizio di ogni iterazione prima di tutto si controlla se esistono degli indici  $k \in I$  tali che per ogni  $h \in I$  la matrice  $X^{hk}$  ottenuta al punto 4 del passo precedente è nulla (leggasi: di norma piccola). Tali indici si rimuovono da  $I$ . Poi vengono aggiunti gli indici  $h$  per cui  $y_h$  non è intero (leggasi: compresi tra  $\varepsilon$  e  $1 - \varepsilon$  per un certo  $\varepsilon$  fissato), come già spiegato.

## 6 Il supergradiente per le nuove coppie

La gestione dell'insieme di indici descritta nella sezione precedente introduce però un'ambiguità sulla scelta della funzione obiettivo da massimizzare al punto 4 dell'algoritmo. Infatti, se  $(h, k)$  è una coppia appena introdotta nell'insieme  $P$ , non si conoscono i valori  $x_{h'}^{hk}, x_{h''}^{hk}, x_{k'}^{hk}, x_{k''}^{hk}, y_h^{hk}, y_k^{hk}, y^{hk}$  che sono necessari per calcolare le componenti del supergradiente di  $\phi$  in corrispondenza di  $X_{hh}^{hk}, X_{kk}^{hk}, X_{hk}^{hk}$ . Più precisamente, si possono scegliere a piacere tali valori (essi infatti non influenzano la funzione obiettivo, perché possiamo far finta di aver considerato  $X^{hk} = 0$  all'iterazione precedente) a patto che essi soddisfino le condizioni dalla (6) alla (11), e ad ogni scelta di essi si ottiene un diverso supergradiente di  $\phi$ .

L'approccio che si è deciso di seguire è scegliere  $x_{h'}^{hk}, x_{h''}^{hk}, x_{k'}^{hk}, x_{k''}^{hk}, y_h^{hk}, y_k^{hk}, y^{hk}$  in modo che il supergradiente risultante sia di norma minima. È evidente che tale scelta è indipendente per ogni coppia  $(h, k)$  appena introdotta. Per ognuna di esse vogliamo quindi minimizzare la norma di

$$\left[ \begin{array}{c} \frac{(x_{h'}^{hk})^2}{y_h^{hk}} + \frac{(x_{h''}^{hk})^2}{y^{hk}} - (x_h)^2 \\ \frac{(x_{k'}^{hk})^2}{y_k^{hk}} + \frac{(x_{k''}^{hk})^2}{y^{hk}} - (x_k)^2 \\ \frac{2x_{h''}^{hk}x_{k''}^{hk}}{y^{hk}} - 2x_hx_k \end{array} \right] \quad (13)$$

sotto le condizioni dalla (6) alla (11). Per iniziare dimostriamo un semplice risultato:

**Lemma 1.** *Siano  $x_1, x_2, y_1, y_2 > 0$ . Allora*

$$\frac{x_1^2}{y_1} + \frac{x_2^2}{y_2} \geq \frac{(x_1 + x_2)^2}{y_1 + y_2}$$

e vale l'uguaglianza se e solo se

$$\frac{x_1}{y_1} = \frac{x_2}{y_2}$$

*Dimostrazione.* Riscriviamo la tesi moltiplicando per i denominatori:

$$x_1^2 y_2 (y_1 + y_2) + x_2^2 y_1 (y_1 + y_2) \geq (x_1^2 + 2x_1 x_2 + x_2^2) y_1 y_2$$

che dopo le opportune semplificazioni diventa:

$$\begin{aligned} x_1^2 y_2^2 + x_2^2 y_1^2 - 2x_1 x_2 y_1 y_2 &\geq 0 \\ (x_1 y_2 - x_2 y_1)^2 &\geq 0 \end{aligned}$$

Da questa riscrittura si ottiene subito il risultato che volevamo.  $\square$

Dal lemma appena mostrato otteniamo che le prime due componenti di (13), in virtù della condizione (6) sono rispettivamente maggiori o uguali a  $\frac{(x_h)^2}{y_h} - (x_h)^2$ ,  $\frac{(x_k)^2}{y_k} - (x_k)^2$ , e dunque in particolare sono positive (visto che  $y_h, y_k \leq 1$ ).

Supponiamo di aver fissato in modo arbitrario i valori  $y_h^{hk}, y_k^{hk}, y^{hk}$ . Allora, scegliendo  $x_{h'}^{hk}, x_{h''}^{hk}, x_{k'}^{hk}, x_{k''}^{hk}$  in modo da soddisfare le relazioni

$$\frac{x_{h'}^{hk}}{y_h^{hk}} = \frac{x_{h''}^{hk}}{y^{hk}} = \frac{x_h}{y_h} \quad (14)$$

$$\frac{x_{k'}^{hk}}{y_k^{hk}} = \frac{x_{k''}^{hk}}{y^{hk}} = \frac{x_k}{y_k} \quad (15)$$

osserviamo che (sempre per il lemma, in particolare la condizione per l'uguaglianza) vale

$$\begin{aligned} \frac{(x_{h'}^{hk})^2}{y_h^{hk}} + \frac{(x_{h''}^{hk})^2}{y^{hk}} - (x_h)^2 &= \frac{(x_h)^2}{y_h} - (x_h)^2 \\ \frac{(x_{k'}^{hk})^2}{y_k^{hk}} + \frac{(x_{k''}^{hk})^2}{y^{hk}} - (x_k)^2 &= \frac{(x_k)^2}{y_k} - (x_k)^2 \end{aligned}$$

ovvero le prime due componenti del supergradiente (13) assumono il minimo valore possibile (che è il minimo anche in modulo, visto che è positivo).

Vediamo ora che, sotto le condizioni (14)(15), scegliendo  $y^{hk} = y_h y_k$  si annulla la terza componente di (13). Infatti

$$\frac{2x_{h''}^{hk} x_{k''}^{hk}}{y^{hk}} - 2x_h x_k = 2 \frac{x_{h''}^{hk} x_{k''}^{hk}}{y^{hk} y^{hk}} y^{hk} - 2x_h x_k = 2 \frac{x_h x_k}{y_h y_k} y^{hk} - 2x_h x_k = 0$$

Supponendo di porre  $y^{hk} = y_h y_k$ , in virtù del vincolo (7) siamo obbligati a porre:

$$y_h^{hk} = y_h - y_h y_k, \quad y_k^{hk} = y_k - y_h y_k$$

Verifichiamo ora che scegliendo tali valori per  $y^{hk}, y_h^{hk}, y_k^{hk}$  e scegliendo i valori di  $x_{h''}^{hk}, x_{h'''}^{hk}, x_{k''}^{hk}, x_{k'''}^{hk}$  che soddisfano (14) e (15) tutti i vincoli vengono rispettati. In effetti l'unico vincolo non del tutto banale è  $y^{hk} + y_h^{hk} + y_k^{hk} \leq 1$ , ovvero, sostituendo,  $y_h + y_k + y_h y_k \leq 1$ . Questo si riscrive:

$$y_h(1 - y_k) \leq 1 - y_k$$

e pertanto è automaticamente soddisfatto visto che  $y_h \in [0, 1]$ .

La conclusione è che possiamo scegliere le variabili  $x_{h''}^{hk}, x_{h'''}^{hk}, x_{k''}^{hk}, x_{k'''}^{hk}, y_h^{hk}, y_k^{hk}, y^{hk}$  in modo tale che le componenti del supergradiente valgano

$$\begin{bmatrix} \frac{(x_h)^2}{y_h} - (x_h)^2 \\ \frac{(x_k)^2}{y_k} - (x_k)^2 \\ 0 \end{bmatrix} \quad (16)$$

e questo è il supergradiente di norma minima tra tutti quelli possibili (perché in effetti minimizza ciascuna delle componenti).

## 7 Sperimentazioni

L'algoritmo descritto è stato implementato in Matlab; per la risoluzione dei problemi SDP e SOCP (nei punti 3 e 4 dell'algoritmo come è stato qui delineato) sono stati utilizzati i solutori SeDuMi e CPLEX. Al punto 4, in cui si deve massimizzare su un segmento una funzione concava, è stato utilizzato un algoritmo che non trova necessariamente il punto di massimo nel segmento dato, ma piuttosto si accontenta di trovare un punto di "sufficiente crescita" che rispetta quelle che sono note in letteratura come le condizioni di Armijo e di Wolfe.

Il numero massimo di iterazioni è stato fissato a 20; l'algoritmo può comunque terminare prima dell'esecuzione delle 20 iterazioni se la soluzione trovata smette di migliorare in modo sensibile. Più precisamente, detta  $f_i$  la valutazione inferiore trovata al passo  $i$ , l'algoritmo termina non appena trova  $f_{i+1} \leq (1 + \varepsilon)f_i$ . Negli esperimenti è stato posto  $\varepsilon = 10^{-6}$ .

Per testare l'algoritmo sono state utilizzate delle istanze contenenti matrici  $Q$  dense di dimensione  $n \times n$ , con  $n \in \{25, 50, 200, 300, 400\}$ , già utilizzate in altri test riguardanti il problema qui trattato. Queste istanze sono suddivise in varie categorie: le istanze "p", "z", "n" hanno matrici  $Q$  con tutte le entrate positive, e in particolare le "p" hanno diagonalmente fortemente dominanti, quelle "n" fortemente non dominanti, e quelle "z" sono una via di mezzo. Nei casi con  $n \in \{25, 50\}$

vi sono anche delle istanze “o”, “y”, “m” ottenute dalle precedenti cambiando il segno agli elementi fuori dalla diagonale ed eventualmente incrementando i valori sulla diagonale (per ottenere matrici semidefinite positive).

Per valutare la bontà di una certa valutazione inferiore  $f$  ottenuta si è calcolata la distanza, in percentuale, dalla migliore soluzione intera conosciuta ( $ub$ ), che in realtà nelle istanze con  $n \in \{25, 50\}$  è proprio la soluzione ottima. Più precisamente viene calcolato  $gap(f) = \frac{ub-f}{ub}$ . In particolare  $gap(f) = 1$  indica una valutazione inferiore pessima, mentre valori piccoli di  $gap(f)$  indicano buone valutazioni inferiori.

Nelle tabelle relative ai test con  $n \in \{25, 50\}$  sono riportati, in ogni riga:

- il nome dell’istanza (da cui si evince sia  $n$  sia il “tipo”)
- il numero di iterazioni effettuate
- il tempo richiesto complessivamente nel punto 3 (calcolo della direzione di crescita)
- il tempo richiesto complessivamente nel punto 4 (ricerca nel segmento)
- il numero medio di indici utilizzati per la scomposizione in matrici  $2 \times 2$
- il numero di indici utilizzati nella scomposizione finale
- il gap al termine delle iterazioni (come descritto sopra)
- il miglior gap ottenibile con scomposizioni del tipo qui trattato
- il gap tra la valutazione ottenuta alla prima iterazione rispetto alla migliore valutazione ottenibile
- il gap tra la valutazione ottenuta al termine e la migliore valutazione ottenibile
- il gap tra la valutazione ottenuta da un’euristica (qui non trattata) che utilizza solo la diagonale e la migliore valutazione ottenibile

Nelle tabelle relative alle istanze con  $n \geq 200$  mancano le informazioni riguardanti la migliore valutazione ottenibile (e i gap rispetto ad essa), in quanto troppo costosa da calcolare.

## 8 Problemi numerici

Nel corso delle sperimentazioni si sono verificate delle problematiche di tipo numerico. Infatti le matrici  $2 \times 2$  presenti nella decomposizione prodotta da SeDuMi al punto 3 dell’algoritmo a volte presentavano autovalori negativi (secondo l’aritmetica a precisione finita del calcolatore). Dopo numerose prove effettuate su alcune delle istanze che presentavano tali problemi, si sono definite le seguenti strategie per far fronte agli errori numerici:

- Quando al punto 3 viene restituita la decomposizione che massimizza il supergradiente (che poi sarà l’estremo del segmento su cui cercare il massimo), si controlla se le matrici  $2 \times 2$  della decomposizione individuata hanno autovalori positivi. Se non è così, se la matrice  $2 \times 2$  è di norma

molto piccola viene direttamente eliminata; altrimenti si scalano di un fattore  $1 - 10^{-8}$  gli elementi fuori dalla diagonale (della matrice  $2 \times 2$ ) e di un fattore  $1 - 10^{-9}$  gli elementi sulla diagonale (sempre della matrice  $2 \times 2$ ). Quest'ultima operazione sugli elementi della diagonale non è per assicurarsi la positività della matrice  $2 \times 2$ , ma per evitare che si perda la positività della matrice residua.

- Durante la line search (punto 4) se la matrice residua ha autovalori negativi si scala la sua diagonale di un fattore  $1 + 10^{-7}$  per cercare di renderla semidefinita positiva.

Anche se in tal modo si è molto ridotto il numero di inconvenienti numerici che impedivano di effettuare la line search, sono rimaste alcune istanze (specialmente per  $n \geq 200$ ) in cui essi non si sono riusciti ad evitare.

## 9 Considerazioni

Si può notare come in generale nei casi con entrate tutte positive si ottengono valutazioni inferiori nettamente migliori rispetto ai casi con entrate negative. Come si evince dai risultati riportati nelle tabelle, però, il fatto che in certe istanze non si ottengano valutazioni inferiori soddisfacenti nella maggior parte dei casi non è dovuto alla particolare implementazione o strategia usata per ottenere la decomposizione, ma piuttosto dal fatto che con il tipo di decomposizione che abbiamo trattato è *impossibile* ottenere buone valutazioni (si veda in particolare come in molte istanze di tipi “y” ed “m” la migliore valutazione ottenibile ha un gap del 100% rispetto alla soluzione intera).

Se facciamo un confronto tra i risultati ottenuti con Frank - Wolfe e quelli ottenuti con la decomposizione diagonale tramite un approccio euristico (ultima colonna delle tabelle) osserviamo che spesso l'euristica fornisce risultati migliori (e per di più meno costosi da calcolare) nelle istanze con entrate positive. Un aspetto molto negativo è che in queste istanze (quelle con entrate positive) sembra che Frank-Wolfe non produca sostanziali miglioramenti rispetto alla decomposizione diagonale trovata alla prima iterazione.

Si nota invece che nelle istanze di tipo “o” Frank - Wolfe ottiene dei risultati molto soddisfacenti (a differenza dell'approccio euristico), che addirittura coincidono quasi sempre con la migliore valutazione ottenibile. Purtroppo a tale bontà delle soluzioni si associa un tempo di esecuzione spesso molto elevato, dovuto al fatto che un gran numero di indici (spesso tutti) viene aggiunto all'insieme di quelli usati per le scomposizioni  $2 \times 2$ , facendo aumentare in particolar modo il tempo richiesto per la ricerca del massimo sul segmento.

## 10 Stabilizzazione

Si può apportare una modifica all'algoritmo di Frank - Wolfe in modo da considerare, quando si cerca una direzione di crescita, il fatto che il supergradiente in un punto dà delle informazioni molto utili per quanto riguarda la crescita della funzione nell'intorno del punto stesso, ma che più ci si allontana dal punto meno tale supergradiente è “affidabile” per cercare direzioni di crescita.

	iter	tSDP	tSOCP	$\bar{k}$	k	gap	best	first	final	heur
25-p-a	20	16.8	67.2	3.0	5	2.23	2.21	0.08	0.02	0.01
25-p-b	20	9.0	72.0	0.9	0	0.04	0.03	0.02	0.02	0.00
25-p-c	2	1.5	1.6	4.5	4	2.48	2.30	0.19	0.19	0.06
25-p-d	2	1.3	1.0	3.0	3	0.78	0.61	0.17	0.17	0.04
25-p-e	3	3.1	6.5	7.7	8	1.82	1.29	0.55	0.54	0.18
25-p-f	7	7.0	9.4	6.6	8	0.67	0.44	0.25	0.23	0.05
25-p-g	2	1.4	1.5	4.0	4	2.15	1.71	0.46	0.46	0.05
25-p-h	2	1.1	4.0	1.0	0	1.49	1.43	0.06	0.06	0.02
25-p-i	2	2.3	3.5	8.0	8	0.70	0.55	0.15	0.15	0.00
25-p-j	3	2.5	2.4	4.7	5	1.21	0.88	0.34	0.33	0.07
25-o-a	16	30.8	98.6	11.4	15	2.80	2.80	1.19	0.00	0.03
25-o-b	8	5.8	26.7	4.6	5	0.09	0.08	0.27	0.00	0.01
25-o-c	19	139.5	419.0	23.7	25	2.48	2.47	5.90	0.01	2.68
25-o-d	4	27.6	659.2	24.2	25	7.35	7.35	10.40	0.00	8.36
25-o-e	12	73.1	1858.2	24.6	25	8.90	8.90	9.18	-0.00	7.03
25-o-f	20	140.5	666.3	23.6	25	1.73	1.52	10.13	0.21	8.46
25-o-g	9	62.0	183.3	24.1	25	3.05	3.05	7.51	0.00	5.22
25-o-h	3	19.4	19.4	23.0	25	0.85	0.85	7.26	0.00	5.36
25-o-i	6	39.5	94.6	24.7	25	6.58	6.58	8.94	0.00	7.57
25-o-j	9	59.1	509.7	22.9	25	2.30	2.30	6.22	0.00	2.52

Tabella 1:  $n = 25$ ; istanze di tipo p.o.

Quindi si modifica la funzione obiettivo (che nella versione originale è semplicemente il prodotto scalare con il supergradiente) sottraendo un termine proporzionale alla distanza dal punto di partenza. In questo modo, visto che la funzione obiettivo è da massimizzare, si privilegiano punti vicini a quello di partenza.

Questa stabilizzazione è governata da un parametro  $t > 0$ , il coefficiente per cui viene moltiplicata la distanza nella funzione obiettivo. Un valore di  $t$  elevato corrisponde a dare molto peso alla distanza, mentre ponendo  $t = 0$  si ottiene la versione originale non stabilizzata. Un modo ragionevole di gestire questo parametro è iniziarlo ad un determinato valore e procedere nelle iterazioni finché la soluzione continua a migliorare; quando la soluzione comincia a “stagnare” si decrementa il valore di  $t$  (ad esempio dividendolo per un fattore 10) e si prosegue.

Anche se in questo progetto non sono stati effettuati molti test significativi al riguardo, sembra che la stabilizzazione possa dare dei buoni risultati, soprattutto se si è interessati a trovare buone valutazioni in poche iterazioni. Nella figura 1 sono riportati i grafici che rappresentano l’andamento del gap in due delle istanze, sia con l’algoritmo stabilizzato che in quello non stabilizzato. Si nota come soprattutto nelle prime iterazioni la versione stabilizzata sembra fare progressi molto più rapidi. Tuttavia nell’istanza 25-o-b (quella a destra in figura) la versione non stabilizzata trova prima della decima iterazione una direzione di rapida crescita e lì si ferma; invece la versione stabilizzata continua a trovare piccoli miglioramenti della soluzione (comunque leggermente peggiori di quella trovata dall’algoritmo non stabilizzato) e giunge fino alla ventesima iterazione senza fare progressi significativi.

	iter	tSDP	tSOCP	$\bar{k}$	k	gap	best	first	final	heur
25-z-a	10	14.6	49.8	10.5	11	2.34	1.50	0.90	0.85	0.35
25-z-b	10	19.8	84.0	12.8	14	2.77	1.58	1.29	1.21	0.64
25-z-c	19	43.9	223.4	14.4	15	2.92	1.64	1.44	1.30	0.81
25-z-d	20	39.0	165.1	10.2	16	0.24	0.22	0.37	0.02	0.04
25-z-e	8	4.6	28.7	1.0	0	0.01	-0.00	0.02	0.02	0.00
25-z-f	20	17.3	31.5	5.7	6	1.29	0.48	0.94	0.82	0.19
25-z-g	20	37.7	67.8	9.7	17	1.30	1.21	0.34	0.09	0.12
25-z-h	18	34.2	166.3	9.3	18	2.44	2.41	0.40	0.03	0.07
25-z-i	17	19.9	118.5	6.9	10	0.79	0.78	0.20	0.01	0.03
25-z-j	20	23.9	130.1	6.5	10	2.44	2.37	0.26	0.07	0.07
25-y-a	3	31.8	297.2	25.0	25	100.00	100.00	0.00	0.00	0.00
25-y-b	1	7.1	286.1	25.0	0	100.00	100.00	0.00	0.00	0.00
25-y-c	1	8.6	290.9	25.0	0	100.00	100.00	0.00	0.00	0.00
25-y-d	15	42.4	131.8	14.0	14	1.02	0.33	7.83	0.70	7.83
25-y-e	3	4.4	30.7	9.0	4	0.49	0.21	1.94	0.28	1.89
25-y-g	6	26.8	164.4	18.5	17	14.44	8.71	17.37	6.27	17.37
25-y-h	20	45.2	73.2	12.3	13	7.02	6.05	10.19	1.02	10.19
25-y-i	5	14.5	90.2	14.0	10	9.65	1.61	14.58	8.16	14.59
25-y-j	3	9.8	78.5	15.0	15	18.68	8.20	16.67	11.41	16.68

Tabella 2:  $n = 25$ ; istanze di tipo z,y.

	iter	tSDP	tSOCP	$\bar{k}$	k	gap	best	first	final	heur
25-n-a	20	71.5	490.4	17.3	20	2.35	0.27	2.52	2.08	1.51
25-n-b	20	42.8	257.7	13.7	16	2.68	0.79	2.08	1.91	1.18
25-n-c	20	35.8	185.2	11.9	13	2.94	1.42	1.64	1.54	0.62
25-n-d	20	92.1	519.5	19.2	24	2.45	0.88	2.47	1.59	1.12
25-n-e	20	69.1	287.2	16.9	21	1.40	0.38	1.63	1.02	1.01
25-n-f	20	23.9	72.6	7.1	12	0.79	0.70	0.61	0.09	0.21
25-n-g	2	1.6	13.3	3.0	0	4.08	4.06	0.19	0.02	0.02
25-n-h	20	78.1	483.3	17.9	20	2.16	0.64	1.87	1.53	1.31
25-n-i	20	64.3	378.3	16.1	19	2.19	0.35	2.33	1.85	1.02
25-n-j	20	109.7	823.1	20.5	24	2.48	0.74	2.41	1.76	1.26
25-m-a	1	9.2	295.7	25.0	0	100.00	100.00	0.00	0.00	0.00
25-m-b	1	8.2	291.9	25.0	0	100.00	100.00	0.00	0.00	0.00
25-m-c	2	15.7	10.8	25.0	25	98.59	98.59	68.15	0.02	68.15
25-m-d	1	9.7	289.6	25.0	0	100.00	100.00	0.00	0.00	0.00
25-m-e	1	8.5	288.1	25.0	0	100.00	100.00	0.00	0.00	0.00
25-m-f	20	49.0	80.2	11.4	12	9.08	7.71	7.74	1.48	7.74
25-m-g	2	2.2	0.4	5.0	5	5.32	4.66	0.70	0.70	0.70
25-m-h	2	21.8	11.4	25.0	25	67.19	67.04	60.79	0.46	60.79
25-m-i	1	8.4	288.4	25.0	0	100.00	100.00	0.00	0.00	0.00
25-m-j	1	8.8	292.1	25.0	0	100.00	100.00	0.00	0.00	0.00

Tabella 3:  $n = 25$ ; istanze di tipo n,m.

	iter	tSDP	tSOCP	$\bar{k}$	k	gap	best	first	final	heur
50-p-a	2	6.6	4.2	6.0	6	1.47	1.20	0.27	0.27	0.03
50-p-b	8	8.3	81.5	3.2	4	2.10	2.07	0.11	0.04	0.01
50-p-c	2	2.4	3.6	4.5	4	2.44	2.11	0.34	0.34	0.05
50-p-d	3	4.6	10.1	8.0	8	2.60	2.28	0.33	0.33	0.05
50-p-e	3	6.2	20.8	10.7	11	0.89	0.26	0.63	0.63	0.07
50-p-f	2	4.2	8.7	10.0	10	1.80	1.66	0.14	0.14	0.03
50-p-g	2	1.9	2.3	4.0	4	0.82	0.68	0.15	0.15	0.06
50-p-h	2	2.8	4.4	6.0	6	2.95	2.46	0.51	0.50	0.03
50-p-i	20	28.4	89.6	8.2	13	1.27	1.24	0.09	0.03	0.00
50-p-j	2	2.9	6.2	8.0	8	0.97	0.74	0.24	0.24	0.04
50-o-a	7	1243.9	22770.9	48.1	50	3.31	3.31	15.37	0.01	11.33
50-o-b	6	13.7	193.1	10.5	12	2.53	2.52	1.29	0.01	0.02
50-o-c	20	6079.2	25120.9	46.0	50	3.35	2.44	13.60	0.93	7.05
50-o-d	3	468.0	382.9	44.3	50	3.80	3.80	16.22	-0.00	11.58
50-o-e	9	2015.8	54334.3	48.4	50	7.95	7.95	15.09	0.00	10.03
50-o-f	6	965.4	4927.4	46.3	50	3.25	3.25	14.14	0.00	8.60
50-o-h	20	5196.6	27121.0	46.5	50	4.82	4.07	17.22	0.78	11.99
50-o-j	20	6236.0	30303.4	46.5	50	2.31	0.98	15.40	1.34	10.01

Tabella 4:  $n = 50$ ; istanze di tipo p,o.

	iter	tSDP	tSOCP	$\bar{k}$	k	gap	best	first	final	heur
50-z-a	20	27.8	83.2	3.6	4	0.31	0.27	0.19	0.04	0.03
50-z-b	11	54.5	348.9	16.5	17	2.90	2.00	0.96	0.92	0.31
50-z-c	12	104.8	1649.5	23.4	25	2.66	0.61	2.18	2.06	0.93
50-z-d	16	145.3	2444.6	23.6	25	2.37	0.50	2.01	1.88	0.60
50-z-f	18	116.0	1852.6	20.9	22	2.88	1.04	2.01	1.86	0.44
50-z-h	9	34.9	194.6	14.3	15	2.21	0.63	1.64	1.59	0.43
50-z-i	8	20.3	83.6	10.9	11	2.85	1.36	1.57	1.51	0.43
50-z-j	8	23.0	98.9	12.0	12	1.50	0.23	1.34	1.27	0.29
50-y-a	4	20.8	118.4	13.0	9	4.17	0.14	8.68	4.03	8.68
50-y-b	1	391.0	8130.8	50.0	0	100.00	100.00	0.00	0.00	0.00
50-y-c	2	526.4	328.9	50.0	50	98.09	98.09	87.24	0.00	87.24
50-y-d	1	320.2	8227.8	50.0	0	100.00	100.00	0.00	0.00	0.00
50-y-e	1	350.3	8436.4	50.0	0	100.00	100.00	0.00	0.00	0.00
50-y-f	2	551.9	273.8	49.0	49	52.07	50.08	78.17	3.99	78.17
50-y-g	1	325.9	8360.0	50.0	0	100.00	100.00	0.00	0.00	0.00
50-y-h	2	639.0	316.2	50.0	50	79.58	79.28	84.82	1.46	84.82
50-y-i	2	587.3	305.2	50.0	50	73.27	73.26	79.87	0.01	79.87
50-y-j	2	538.0	302.6	50.0	50	90.06	90.06	85.72	0.00	85.72

Tabella 5:  $n = 50$ ; istanze di tipo z,y.

	iter	tSDP	tSOCP	$\bar{k}$	k	gap	best	first	final	heur
50-n-a	20	91.0	455.1	13.6	15	3.81	2.43	1.60	1.42	0.41
50-n-b	19	93.2	957.4	18.3	19	2.52	0.47	2.20	2.06	0.69
50-n-c	14	125.3	2648.1	23.6	24	4.57	1.05	3.65	3.55	1.82
50-n-d	20	456.7	6258.8	29.0	34	2.92	0.75	2.48	2.18	0.88
50-n-e	1	9.7	239.1	21.0	0	3.12	0.36	2.77	2.77	1.19
50-n-f	20	158.6	2012.6	21.1	24	4.91	2.31	2.96	2.66	1.56
50-n-g	18	46.7	246.1	11.9	12	2.70	1.45	1.61	1.27	0.22
50-n-h	20	226.6	3949.9	23.8	26	3.94	1.17	3.01	2.80	1.30
50-n-i	20	463.8	6744.9	29.1	34	3.44	0.57	3.37	2.88	1.42
50-n-j	20	581.2	8353.2	29.9	38	3.69	0.64	3.82	3.07	1.34
50-m-a	20	203.3	1678.6	22.8	24	42.85	15.19	52.80	32.62	52.80
50-m-b	1	335.8	8444.4	50.0	0	100.00	100.00	0.00	0.00	0.00
50-m-c	1	329.6	8574.8	50.0	0	100.00	100.00	0.00	0.00	0.00
50-m-d	1	336.9	8600.6	50.0	0	100.00	100.00	0.00	0.00	0.00
50-m-e	1	317.5	9031.9	50.0	0	100.00	100.00	0.00	0.00	0.00
50-m-f	2	608.7	302.1	50.0	50	73.63	73.31	83.93	1.20	83.93
50-m-g	2	11.5	7.2	19.0	19	19.36	6.24	32.11	14.00	32.12
50-m-h	2	476.2	235.5	47.0	47	55.29	51.01	73.96	8.73	73.96
50-m-i	2	554.4	301.2	50.0	50	90.22	90.20	84.84	0.26	84.84
50-m-j	2	528.5	310.6	50.0	50	90.15	90.15	85.18	-0.00	85.20

Tabella 6:  $n = 50$ ; istanze di tipo n,m.

	iter	tSDP	tSOCP	$\bar{k}$	k	first	gap	gap	heur
200-p-a	2	14.3	38.2	8.0	8	1.26	1.26	0.28	
200-p-b	2	21.8	65.5	7.0	7	0.97	0.97	0.08	
200-p-d	2	11.3	43.7	6.0	6	0.75	0.75	0.23	
200-p-e	2	18.3	98.1	13.0	13	0.69	0.69	0.46	
200-p-f	3	17.8	96.2	6.3	7	0.89	0.88	0.12	
200-p-g	2	11.7	73.3	9.0	9	0.85	0.84	0.24	
200-p-h	2	16.3	79.5	11.0	11	0.47	0.47	0.20	
200-p-j	2	12.3	54.0	8.0	8	1.89	1.89	0.23	
200-z-a	3	39.8	910.4	19.7	21	1.52	1.50	0.95	
200-z-c	4	54.7	1171.3	20.2	21	2.89	2.86	1.33	
200-z-d	4	60.1	1776.5	21.0	22	1.80	1.77	1.11	
200-z-e	6	85.9	2384.9	20.7	21	1.61	1.57	1.34	
200-z-f	5	104.2	3563.1	24.2	25	3.10	3.06	1.33	
200-z-h	4	72.5	2666.3	24.0	25	1.66	1.63	1.27	
200-z-i	2	25.3	334.6	18.5	18	2.05	2.02	0.88	
200-n-d	7	171.0	7291.5	26.1	27	2.81	2.71	2.07	
200-n-e	4	49.4	494.3	15.0	16	3.25	3.21	1.27	
200-n-f	12	483.6	17536.9	29.9	31	4.95	4.86	2.05	

Tabella 7: Istanze con  $n = 200$ . Nell'ultima colonna è riportato il gap tra la valutazione ottenuta con Frank - Wolfe e quella della decomposizione diagonale ottenuta con il procedimento euristico.

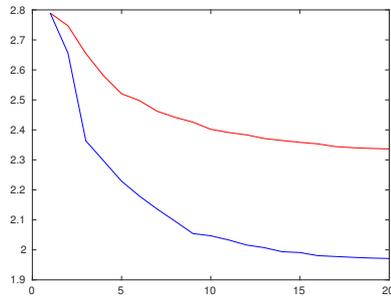
	iter	tSDP	tSOCP	$\bar{k}$	k	first gap	gap	heur
300-p-a	2	48.2	177.6	7.0	7	1.61	1.61	0.12
300-p-b	2	41.3	187.9	12.0	12	0.51	0.51	0.18
300-p-c	2	42.3	173.0	9.0	9	1.30	1.30	0.10
300-p-d	2	65.7	258.6	8.5	8	1.16	1.15	0.19
300-z-a	2	52.5	262.7	12.0	12	2.69	2.67	0.77
300-z-c	4	116.1	3089.6	22.0	23	2.12	2.09	1.29
300-z-d	2	46.3	994.6	19.5	18	1.63	1.60	1.09
300-n-b	4	113.0	3877.1	22.8	24	2.15	2.12	1.14
300-n-c	6	582.4	23873.4	35.2	36	3.56	3.48	1.83
300-n-g	3	63.3	620.5	14.7	15	2.42	2.38	1.25
300-n-h	4	321.3	11605.4	32.8	35	2.44	2.39	1.64

Tabella 8: Alcune istanze con  $n = 300$

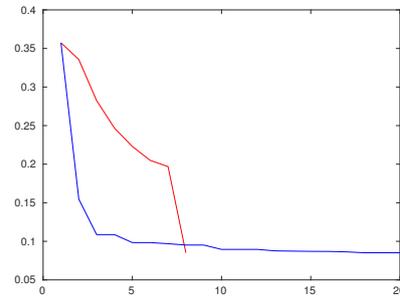
	iter	tSDP	tSOCP	$\bar{k}$	k	first gap	gap	heur
400-p-a	2	53.0	358.8	12.5	12	1.47	1.47	0.17
400-p-b	5	90.3	3540.4	3.0	3	1.88	1.87	0.05
400-p-c	2	48.1	384.3	4.5	4	0.48	0.48	0.09
400-p-d	2	48.0	467.6	6.0	6	1.23	1.23	0.05
400-z-b	2	94.0	1466.7	20.0	19	1.79	1.77	1.07
400-z-c	2	97.9	1924.8	20.5	18	3.00	2.98	1.22
400-z-d	2	125.6	4121.8	28.5	26	2.30	2.28	1.36
400-z-f	2	97.1	1216.2	19.5	18	2.17	2.16	0.66
400-n-a	4	231.4	7537.8	26.5	27	3.54	3.51	1.36
400-n-d	3	140.0	4436.4	24.0	25	1.98	1.95	1.39
400-n-e	3	148.7	3960.2	24.3	25	1.89	1.86	1.16
400-n-g	3	176.0	6284.6	26.7	28	2.65	2.62	1.45

Tabella 9: Alcune istanze con  $n = 400$

Figura 1: Andamento del gap (rispetto alla migliore valutazione ottenibile) in alcune istanze. In rosso per l'algoritmo non stabilizzato; in blu quello stabilizzato.



(a) Istanza 25-n-a



(b) Istanza 25-o-b