



UNIVERSITÀ DI PISA

DIPARTIMENTO DI MATEMATICA
CORSO DI LAUREA TRIENNALE IN MATEMATICA

TESI DI LAUREA TRIENNALE

Modelli matematici per l'ottimizzazione del consumo energetico in reti 5G

CANDIDATO:

Matteo Cacciola

RELATORE:

Prof. Antonio Frangioni

ANNO ACCADEMICO 2015/2016

Indice

Introduzione	ii
1 Il problema reale	1
1.1 Istanze	3
2 Il modello matematico	6
3 Approssimazione lineare del problema	11
3.1 Implementazione	14
3.2 Sperimentazione	16
4 Riformulazione quadratica del problema	19
4.1 Implementazione	20
5 Modello con discretizzazione binaria	22
5.1 Implementazione	28
6 Sperimentazione finale	30
6.1 Sperimentazione del modello esatto	39
6.2 Conclusioni	42

Introduzione

Al giorno d'oggi le reti di calcolatori sono diventate la maggiore causa d'impiego di energia elettrica nel mondo. Al fine di ridurre questo tipo di consumo si può operare su due aspetti principali: rendere le macchine più efficienti da un punto di vista hardware e lavorare a nuovi algoritmi per ottimizzare la prestazione della rete. Gli algoritmi utilizzati adesso, infatti, sono pensati per soddisfare costantemente una domanda molto elevata, che però si realizza solo in alcuni momenti critici della giornata, mentre nel resto del tempo si potrebbe diminuire il consumo energetico.

In questo contesto si inseriscono le reti cellulari di telecomunicazione, in particolare la rete LTE, che sono proprio quelle che tratteremo in questo lavoro. Proprio in queste reti risulta evidente che la potenza richiesta durante il giorno sia altamente variabile. La domanda più alta si verifica nelle aree commerciali durante le ore lavorative, mentre nel resto del giorno una parte dell'energia consumata va sprecata. Così l'idea è quella di attivare e disattivare alcuni nodi a secondo della richiesta corrente.

Un approccio di questo tipo risulta particolarmente efficace in reti *eterogenee* nelle quali ci sono dei nodi che soddisfano la maggior parte della richiesta in una grande area, chiamati *macro*, e altri nodi che, agendo in aree molto più piccole, vanno a coprire un eventuale richiesta che la macro non riesce ad assolvere, chiamati *micro*. Mentre si suppone che le macro restino sempre accese, le micro possono essere attivate e disattivate a seconda della richiesta attuale. Tuttavia l'operazione di spegnimento e accensione delle antenne non può essere fatta molte volte dato che ciò, a lungo andare, provocherebbe un'usura molto rapida delle componenti fisiche. Inoltre tale

operazione richiede diverso tempo per l'avvio del sistema operativo e di alcuni software. Tuttavia noi non ci occuperemo di tale aspetto, l'obbiettivo di questo lavoro è quello di trovare, data una certa configurazione e dei dati iniziali, quali sono le scelte più efficienti in termini di accensione e spegnimento delle antenne.

Nonostante vari problemi che si presentano nel seguire questo approccio (li esamineremo più avanti) i primi test evidenziano addirittura un risparmio tra il 10% e il 25% di energia durante l'intero giorno.

Capitolo 1

Il problema reale

La strategia che seguiremo per ottimizzare la resa della rete, senza accendere e spegnere troppe volte le micro, è quella di calcolare lo stato ottimale del sistema in base a quello che ci aspettiamo accada, in media, nel prossimo intervallo di tempo (circa 15 minuti), basandoci su dati e statistiche raccolti mediante delle misurazioni sul campo, e ripetere tale operazione quando il suddetto tempo sarà passato.

L'area che siamo interessati a coprire viene divisa in esagoni, in ogni esagono c'è una macro e varie micro (figura 1.1), ogni nodo può allocare un certo numero di *blocchi* ed associarli ad un utente per trasmettervi segnale, dunque l'energia consumata da ogni nodo è pari ad un certo costo base necessario per l'accensione dell' antenna, più il costo di un singolo blocco moltiplicato per il numero di blocchi attivati. Dato che le macro non verranno mai spente il costo base di queste ultime sarà sempre presente.

Figura 1.1:

Per ogni antenna e per ogni utente, data una configurazione, è possibile calcolare la potenza del segnale che, da ogni blocco del nodo, arriva all'utente, chiameremo questa potenza SINR, se quest'ultimo è abbastanza alto allora ci sarà una certa *velocità di trasmissione* che crescerà linearmente fino ad una velocità massima η_{\max} , chiameremo la funzione che descrive il crescere della velocità di trasmissione in relazione al SINR $F(\text{SINR})$ (figura 1.2).

Supponendo di avere un utente con richiesta D , allora il numero di blocchi da dedicare al dato utente sarà $\frac{D}{F(SINR)}$. Il fatto che questa quantità possa non essere intera non è un problema, infatti si tratta di una media su circa 15 minuti, se il numero di blocchi necessari è 1.5 vorrà dire che per un certo periodo di tempo ne servirà 1, poi 2...etc.

Figura 1.2:

Per calcolare il SINR abbiamo bisogno di sapere qual'è la probabilità $\Delta_{e,x}$ che due nodi, x ed e , trasmettano simultaneamente sugli stessi blocchi di un dato utente, data quest'ultima il SINR tra il nodo e e l'utente u è

$$SINR_u^e = \frac{P_{u,e}}{N_G + \sum_{x \neq e} P_{u,x} \Delta_{e,x} / n_e}$$

dove N_G è l'interferenza gaussiana e $P_{u,x}$ è la potenza di trasmissione dal nodo x all'utente u .

Adesso bisogna capire qual'è l'effetto di accendere e spegnere le micro in una data configurazione, sia dal punto di vista energetico che da quello dell'efficienza.

Quando una macro è in sovraccarico, l'accensione delle micro del suo esagono fa sì che quest'ultima generi meno interferenza andando a migliorare l'efficienza globale del sistema. Possiamo vedere che questa ipotesi è confermata da alcune misurazioni fatte in diverse configurazioni: supponiamo di interessarci ad un esagono h_0 e usiamo la seguente notazione, $(-|-)$ se tutte le micro sono spente, $(m|m)$ se tutte le micro sono accese, $(-|m)$ se sono accese tutte le micro tranne quelle in h_0 e $(m|-)$ se le uniche micro accese sono quelle in h_0 . I risultati delle misurazioni fatte sono illustrati in figura 1.3.

Figura 1.3:

Dal punto di vista energetico l'accensione di una micro comporta un ulteriore dispendio di energia, che si va a sommare alla potenza necessaria per l'allocazione di ogni blocco della micro. D'altro canto con l'accensione di

una micro la macro può diminuire il numero di blocchi che alloca (che in generale hanno bisogno di più energia rispetto a quelli della micro) dunque, energeticamente parlando, è vantaggioso accendere una micro quando questa allocherà un sufficiente numero di blocchi.

Purtroppo questo numero non è noto dato che per calcolarlo dovremmo sapere quale sarà il SINR della micro dopo essere stata accesa e questo valore dipende dalla disposizione degli utenti (che varia continuamente) e dall'interferenza totale del sistema che dipende essa stessa da quanti blocchi allocherà la micro.

1.1 Istanze

Per poter testare i modelli che presenteremo nel seguito abbiamo avuto bisogno di istanze del problema da far risolvere al solutore. Per ottenerle ci siamo rivolti al dipartimento d'Ingegneria dell'Informazione, essendo quest'ultimo un attore chiave a livello internazionale proprio in questo settore, ci siamo rivolti in particolare al professore Giovanni Stea.

Per capire qual è l'obiettivo del nostro progetto e perché è così importante avere istanze realistiche del problema è necessario spiegare, a grandi linee, il contesto in cui si inserisce il nostro lavoro.

L'ottimizzazione delle prestazioni di una rete come quella che stiamo trattando è un argomento molto complicato, infatti, anche volendosi disinteressare dell'aspetto energetico non ponendo limiti sulla quantità d'energia disponibile, rimangono aspetti difficili da bilanciare. Questo perché le antenne, durante il funzionamento, creano interferenza alle altre vicine e tanto più alto è il SINR di un'antenna tanto più alta è l'interferenza provocata da quest'ultima. Dunque se vogliamo aumentare la banda ricevuta dagli utenti non è detto che basti aumentare il numero di blocchi allocati dalle antenne, poiché questo aumenterebbe l'interferenza totale, rischiando di andare a compromettere qualsiasi miglioramento sperato o, addirittura, di peggiorare la situazione.

Perciò si rende necessario valutare tutte le scelte che possono essere fatte, comprendendo in profondità quali sono le variabili che influenzano la resa della rete.

Possiamo ridurre la trattazione di questo argomento all'analisi di tre sottoproblemi.

Il primo riguarda le singole antenne che, praticamente ogni millisecondo, devono decidere su quali blocchi trasmettere il segnale agli utenti connessi.

Il secondo problema consiste nello scegliere, circa ogni secondo, su quali blocchi può allocare ogni antenna, in modo da diminuire l'interferenza totale del sistema, infatti le antenne creano interferenza solo quando trasmettono sugli stessi blocchi.

L'ultimo problema è quello di scegliere, fissato un'intervallo di tempo di circa 15 minuti (che d'ora in poi chiameremo *snapshot*), quali antenne spegnere e quali accendere per evitare gli sprechi di energia. Chiaramente le decisioni che prenderemo riguardo all'ultimo problema avranno influenzeranno le soluzioni del secondo e quindi del primo.

Noi ci occuperemo solo, durante uno snapshot, di scegliere quali antenne saranno accese e quali spente durante il successivo. Lavorando su un intervallo di tempo molto lungo avremo bisogno di una visione più ampia del sistema.

I dati in input relativi allo stato della rete, come ad esempio il numero di utenti connessi, non saranno certi ma solo una media di quello che ci aspettiamo accada nel prossimo snapshot, così anche per i dati di output che rappresenteranno una media del comportamento che vorremo dalle antenne.

Per risolvere il problema sarà dunque necessario stimare l'interferenza in una data configurazione. Quest'ultima è fortemente influenzata da come si scelgono i blocchi sui quali possono trasmettere le antenne, dunque avremo bisogno di sapere come viene risolto il secondo problema. Essendo quest'ultimo ancora argomento di ricerca, supporremo che la scelta avvenga in modo randomico.

Il software che stiamo sviluppando lavora supponendo di avere un budget limitato di energia. In realtà questo budget non è conosciuto, anzi, uno degli obiettivi del nostro lavoro è proprio quello di capire come cambia l'ottimo

Figura 1.4: il "mondo" su cui lavora il software degli ingegneri, composto da 57 esagoni ognuno con 3 antenne

del problema al variare dell'energia massima resa disponibile e determinare così qual è il budget che andrà effettivamente utilizzato.

Per capire se effettivamente le scelte fatte dal nostro modello sono buone bisognerà simularlo. Il dipartimento d'Ingegneria dell'Informazione ha sviluppato un software che simula con precisione tutto il sistema. Quest'ultimo è frutto di un lungo lavoro che tiene conto di tutte le sfaccettature del problema. Attualmente, in questo software la parte del problema che stiamo trattando è risolta attraverso alcune euristiche [AV14]. Quando lo sviluppo del software sarà completo verrà testato in tale simulatore e si cercherà di valutare la qualità delle soluzioni proposte in relazione a quelle degli algoritmi correnti.

Dato che il nostro algoritmo si dovrà inserire proprio in questo simulatore, sarà opportuno che le istanze su cui lavoreremo siano create dal simulatore stesso

Capitolo 2

Il modello matematico

Vediamo ora come descrivere il problema di cui abbiamo parlato prima per mezzo di strumenti logico-matematici in modo da poterlo risolvere con gli algoritmi che conosciamo, ossia formuliamo un modello.

I dati che assumiamo ci vengano dati in input, che quindi possiamo usare per formulare il modello, sono i seguenti:

- Un insieme A di antenne o nodi
- Per ogni antenna il costo di accensione P_a e il costo di attivazione per ogni blocco p_a , non farà differenza sapere se un'antenna è una macro o una micro dato che basterà considerare 0 il costo di accensione delle macro (essendo quest'ultime sempre accese).
- B sarà il massimo numero di blocchi attivabili per ogni antenna
- P sarà il budget di energia disponibile.
- Un insieme di centroidi C , i quali rappresentano aggregazioni di utenti vicini che hanno una certa domanda.
- Per ogni centroide c la domanda D_c corrispettiva.
- Per ogni centroide avremo inoltre i valori $P_{a,c}$ della potenza che l'antenna a trasmette al centroide c da ogni blocco

- Due insiemi $S(c)$ ed $A(c)$ delle antenne che, rispettivamente, possono trasmettere segnale al centroide c e quelle che creano interferenza al segnale ricevuto da c . Questa distinzione è necessaria dato che può accadere che una data antenna abbia un segnale abbastanza forte da essere ricevuto dal dato centroide c , ma troppo basso da poter essere realmente usate per trasmettere segnale (tipicamente accade che il SINR è minore di σ_{min}). Risulta evidente che $S(c) \subset A(c)$, $A(c)$ può essere scritto come $A(c) = \{a \in A | P_{a,c} > 0\}$.
- Inoltre è utile definire $S = \{(a, c) | c \in C \quad a \in S(c)\}$
 $A(a) = \{c \in C | a \in A(c)\}$ $S(a) = \{c \in C | a \in S(c)\}$.

Per formulare il modello sarà necessario essere in grado di calcolare il SINR da un'antenna ad un centroide, per farlo occorrerà dunque saper trovare il valore di $\Delta_{e,x}$. Come detto in precedenza assumeremo che le antenne allochino i blocchi in maniera randomica, dunque avremo la seguente equazione:

$$\Delta_{a,a'}(n) = \sum_{k=\max\{1, n^a+n^{a'}-B\}}^{\min\{n^a, n^{a'}\}} k \frac{\binom{n^a}{k} \binom{B-n^a}{n^{a'}-k}}{\binom{B}{n^{a'}}} \quad (2.1)$$

che può essere riscritta come

$$\Delta_{a,a'}(n) = (n^a n^{a'}) / B \quad . \quad (2.2)$$

dove n è il vettore degli n_a .

Dunque dati un'antenna a , un centroide c e una certa configurazione del sistema, il SINR tra a e c , che chiameremo SINR_c^a , sarà

$$\text{SINR}_c^a(n) = \frac{P_c^a}{N_G + \sum_{a' \in A(c) \setminus \{a\}} P_c^{a'} n_{a'} / B} \quad . \quad (2.3)$$

Quindi, detto n il vettore degli n_a il SINR_c^a dipenderà unicamente da quest'ultimo. Chiamiamo $\text{SINR}_c^a(n)$ la funzione che associa ad ogni vettore n il SINR corrispondente.

Sapendo che la funzione che associa al SINR la rispettiva velocità di trasmissione è lineare a tratti, supponendo di conoscere il valore σ_{min} sotto il

quale non vi è trasmissione, il valore σ_{\max} sopra il quale c'è sempre trasmissione massima e il valore η_{\max} della massima velocità raggiungibile, possiamo definire la funzione $F(\text{SINR})$ nel seguente modo

$$F(s) = \begin{cases} 0 & \text{se } s \leq \sigma_{\min} \\ \eta_{\max} \frac{s - \sigma_{\min}}{\sigma_{\max} - \sigma_{\min}} & \text{se } \sigma_{\min} < s < \sigma_{\max} \\ \eta_{\max} & \text{se } s \geq \sigma_{\max} \end{cases}$$

Dunque, detto $m_{a,c}$ il numero di blocchi dell' antenna a che trasmettono al centroide c , la potenza fornita dall antenna a al centroide c sarà $F(\text{SINR}_c^a(n))m_{a,c}$.

Siamo dunque portati a definire le seguenti variabili per costruire il nostro modello:

- $x_a \quad \forall a \in A$, queste saranno variabili binarie indicanti se l'antenna a è accesa oppure no
- $n_a \quad \forall a \in A$, queste rappresenteranno il numero di blocchi attivi dell'antenna a
- $m_{a,c} \quad \forall a \in A \quad \forall c \in C$, indicanti quanti blocchi dell' antenna a sono dedicati al centroide c
- $v_c \quad \forall c \in C$, rappresentanti quanto è soddisfatta, in percentuale, la domanda del corrispettivo centroide c .

Riusciamo a formulare dunque il seguente modello:

$$\max \sum_{c \in C} v_c \quad (2.4)$$

$$v_c \leq 1 \quad c \in C \quad (2.5)$$

$$v_c \leq \left(\sum_{a \in S(c)} F(SINR_c^a(n)) m_c^a \right) / D_c \quad c \in C \quad (2.6)$$

$$\sum_{c \in S(a)} m_c^a \leq n^a \quad a \in A \quad (2.7)$$

$$0 \leq m_c^a \leq B \quad (a, c) \in S \quad (2.8)$$

$$0 \leq n^a \leq Bx^a \quad a \in A \quad (2.9)$$

$$\sum_{a \in A} P^a x^a + \rho^a n^a \leq P \quad (2.10)$$

$$x^a \in \{0, 1\} \quad a \in A \quad (2.11)$$

Il vincolo 2.5 fa sì che la potenza superflua fornita ai centroidi vada sprecata e quindi il modello tenda ad essere energeticamente-efficiente. Il lato destro di 2.6 è proprio la percentuale di soddisfazione della domanda del centroide, essendo il problema di massimo e dato che le v_c appaiono con segno positivo nella funzione obiettivo, il modello tenderà a far crescere queste ultime variabili che dunque assumeranno il massimo valore consentito da 2.5 e 2.6.

Le equazioni (2.7) (2.8)(2.9) sono i vincoli "naturali" che legano numero totale di centroidi attivati per antenna e numero di blocchi che un'antenna dedica ad un centroide, infatti 2.9 fa sì che le antenne possano allocare blocchi solo se accese e che al più ne vengano allocati B , 2.7 fa sì che la somma dei blocchi che un' antenna dedica ai vari centroidi non superi il numero di blocchi che ha effettivamente a disposizione, 2.8 è in realtà ridondante essendoci già 2.7 ma ci aiuta comunque a stringere la regione ammissibile del problema.

Infine il vincolo 2.10 garantisce che l'energia totale consumata non sia superiore al budget reso disponibile.

Un aspetto positivo di lavorare su un intervallo di tempo lungo è che non abbiamo avuto bisogno di imporre che le variabili n_a e $m_{a,c}$ siano in \mathbb{Z} , il che avrebbe aumentato notevolmente la complessità del modello. Infatti, nonostante queste ultime rappresentino effettivamente quantità intere, il loro valore deve essere una media di ciò che accade durante uno snapshot.

Un aspetto della funzione obiettivo così formulata è che questa non è

uniforme nella distribuzione del segnale. Ossia tra due soluzioni con lo stesso valore della funzione obiettivo, di cui una con alcuni v_c molto bassi e altri molto alti e una con i valori dei v_c molto vicini tra loro, il modello non avrà preferenze. Nel caso in cui si è interessati a garantire una copertura completa e senza buchi si può cambiare la parte da 2.5 a 2.6 con

$$\max v \quad (2.12)$$

$$v \leq 1 \quad \forall c \in C \quad (2.13)$$

$$v \leq \left(\sum_{a \in S(c)} F(SINR_c^a(n)) m_{a,c} \right) / D_c \quad \forall c \in C \quad (2.14)$$

Se invece vogliamo che il modello si interessi di più a soddisfare alcune domande piuttosto di altre, ossia si assume che alcuni centroidi siano più importanti di altri, basterà cambiare la funzione obiettivo con $\sum_{c \in C} v_c w_c$ dove i w_c rappresentano i pesi da dare ai centroidi.

Così formulato il problema presenta elementi quadratici e fortemente non convessi (si veda il vincolo 2.6) che rendono la risoluzione di quest'ultimo molto lunga e complicata anche utilizzando macchine potenti ed algoritmi efficienti.

Quello che faremo nel seguito sarà dunque modificare questo modello per renderlo *meno convesso* e quindi più semplice da trattare

Capitolo 3

Approssimazione lineare del problema

In questo capitolo vedremo come formulare un modello che approssima quello precedente in modo da eliminare i vincoli quadratici.

Per semplificare la discussione definiamo

$$r_c^a = N_G + \sum_{a' \in A(c)} \bar{P}_c^{a'} n_{a'} \quad \text{con} \quad \bar{P}_c^{a'} = P_c^{a'} / B$$

che rappresenta l'interferenza totale che il segnale dall'antenna a al centroide c percepisce. Per prima cosa bisognerà calcolare il valore massimo che può assumere ogni r_c^a , il che può essere fatto risolvendo

$$\rho_{c,max}^a = N_G + \max\{\sum_{a' \in A(c) \setminus a} \bar{P}_c^{a'} n_{a'} : (2.9)(2.10)(2.11)\}$$

Così l'intervallo in cui varia ogni r_c^a è $[N_G, \rho_{c,max}^a]$. L'idea è suddividere quest'intervallo in k parti caratterizzate da $k+1$ valori,

$$N_G = \rho_{c,0}^a \leq \rho_{c,1}^a < \dots < \rho_{c,i}^a < \rho_{c,i+1}^a < \dots < \rho_{c,k-1}^a \leq \rho_{c,k}^a = \rho_{c,max}^a$$

Gli indici 1 e $k-1$ sono "speciali" infatti vogliamo che i corrispondenti valori rappresentino rispettivamente la massima interferenza che si può avere per trasmettere comunque alla massima velocità e la minima interferenza che

non permette nessuna trasmissione di segnale. In quest'ottica poniamo

$$\rho_{c,k-1}^a = \min \{ P_c^a / \sigma_{\min}, \rho_{c,max}^a \}$$

Così avremo $\rho_{c,max}^a = \rho_{c,k-1}^a$ se l'interferenza più alta possibile permette passaggio di segnale, $\rho_{c,max}^a > \rho_{c,k-1}^a$ altrimenti. Similmente poniamo

$$\rho_{c,1}^a = \max \{ P_c^a / \sigma_{\max}, NG \}$$

Così avremo $\rho_{c,0}^a = \rho_{c,1}^a$ se l'interferenza più bassa possibile non permette di trasmettere alla velocità più alta, $NG < \rho_{c,1}^a$ altrimenti.

Ad ogni $\rho_{c,i}^a$ corrisponde una certa velocità di trasmissione (per blocco)

$$\beta_{c,i}^a = F(P_c^a / \rho_{c,i}^a)$$

dunque, dato che i $\rho_{c,i}^a$ sono crescenti in i , i $\beta_{c,i}^a$ sono decrescenti con $0 \leq \beta_{c,i}^a \leq \eta_{\max}$. Adesso abbiamo due modi possibili di scegliere i $\rho_{c,i}^a$: il primo fa sì che l'intervallo $[\rho_{c,1}^a, \rho_{c,k-1}^a]$ venga equipartizionato, il secondo è tale che ad essere equipartizionato sia l'intervallo $[\beta_{c,1}^a, \beta_{c,k-1}^a]$.

Dopo di ché definiamo delle variabili binarie $y_{c,i}^a$ per $i \in I = \{2, 3, \dots, k-1\}$, vogliamo che queste ultime rappresentino il massimo livello d'interferenza raggiunto, ossia $y_{c,i+1}^a = 1$ se $r_c^a > \rho_{c,i}^a$, in questo modo l'ultimo $\rho_{c,i}^a$ per il quale la corrispondente $y_{c,i}^a = 1$ sarà un upper bound per l'interferenza, questo si può ottenere con il seguente vincolo

$$r_c^a \leq \rho_{c,i}^a + y_{c,i+1}^a (\rho_{c,max}^a - \rho_{c,i}^a)$$

possiamo inoltre aggiungere il vincolo

$$y_{c,2}^a \geq y_{c,3}^a \geq \dots \geq y_{c,k-2}^a \geq y_{c,k-1}^a$$

Il valore da approssimare mediante la discretizzazione è $F(P_c^a / r_c^a) m_c^a$, per fare ciò supponiamo che la F sia costante negli intervalli $[\rho_{c,1}^a, \rho_{c,k-1}^a]$ così

avremo che la velocità di trasmissione sarà stimata da $\beta_{c,i}^a + \sum_{i \in I} (\beta_{c,i}^a - \beta_{c,i-1}^a) y_{c,i}^a$ dunque se h è il più grande indice tale per cui $y_{c,h}^a > 0$ allora la velocità di trasmissione (per blocco) sarà stimata da $\beta_{c,h}^a$.

Per riuscire a stimare la velocità totale di trasmissione introduciamo le variabili $m_{c,i}^a$, vogliamo che tali variabili rappresentino $y_{c,i}^a m_c^a$ così imponiamo

$$m_{c,i}^a \geq m_c^a - B(1 - y_{c,i}^a) \quad (a, c, i) \in T \quad (3.1)$$

$$m_{c,i}^a \geq 0 \quad (a, c, i) \in T \quad (3.2)$$

dove $T = S \times I$.

Notiamo che questi vincoli impongono solo che $m_{c,i}^a \geq y_{c,i}^a m_c^a$, senza dare nessuna limitazione superiore a queste nuove variabili, questo non è un problema. Infatti la velocità di trasmissione totale sarà stimata da

$$v_c \leq \left(\sum_{a \in S(c)} (\beta_{c,1}^a m_c^a + \sum_{i \in I} (\beta_{c,i}^a - \beta_{c,i-1}^a) s_{c,i}^a) \right) / D_c \quad c \in C \quad (3.3)$$

, dunque $m_{c,i}^a$ moltiplica $\beta_{c,i}^a - \beta_{c,i-1}^a < 0$, così aumentando $m_{c,i}^a$ i v_c decrescono e con loro la funzione obiettivo quindi il solutore sarà automaticamente portato a far prendere alle $m_{c,i}^a$ il valore più piccolo permesso. Per questo è necessario inserire anche il vincolo 3.2.

Alla fine il modello MILP che otteniamo è il seguente:

$$\max \sum_{c \in C} v_c \quad (2.4)$$

$$v_c \leq \left(\sum_{a \in S(c)} (\beta_{c,1}^a m_c^a + \sum_{i \in I} (\beta_{c,i}^a - \beta_{c,i-1}^a) s_{c,i}^a) \right) / D_c \quad c \in C \quad (3.3)$$

$$NG + \sum_{a' \in A(c) \setminus \{a\}} \bar{P}_c^{a'} n_{a'} \leq \rho_{c,i}^a + y_{c,i}^a (\rho_{c,max}^a - \rho_{c,i}^a) \quad (a, c, i) \in T \quad (3.4)$$

$$y_{c,2}^a \geq y_{c,3}^a \geq \dots \geq y_{c,k-2}^a \geq y_{c,k-1}^a \quad (3.5)$$

$$m_{c,i}^a \geq m_c^a - B(1 - y_{c,i}^a) \quad (a, c, i) \in T \quad (3.1)$$

$$m_{c,i}^a \geq 0 \quad (a, c, i) \in T \quad (3.2)$$

$$(2.5) , (2.7) , (2.8) , (2.9) , (2.10) , (2.11)$$

La soluzione di questo problema rappresenta un'approssimazione inferiore della soluzione vera, infatti il vincolo 3.3 garantisce che se l'interferenza su un certo segnale da antenna a a centroide c arriva ad essere più alta di

$\rho_{c,k}^a$ ma più bassa di $\rho_{c,k+1}^a$ allora nella nostra approssimazione calcoleremo $F(\text{SINR})$ come se l'interferenza fosse $\rho_{c,k+1}^a$, cioè stiamo approssimando $F()$ dal basso, questo fa sì che otterremo una valutazione inferiore della funzione obiettivo. Inoltre i dati in output (x,n,m) , dove x è il vettore del x_i , n quello delle n_i ed m quello delle $m_{i,j}$, definiscono una soluzione ammissibile del problema non approssimato con funzione obiettivo maggiore o uguale a quella calcolata. Se invece siamo interessati ad un approssimazione superiore basterà far scalare di uno l'indice della y nel vincolo 3.4, ossia scrivere

$$N_G + \sum_{a' \in A(c)} \bar{P}_c^{a'} n_{a'} \leq \rho_{c,i}^a + y_{c,i}^a (\rho_{c,\max} - \rho_{c,i}^a)$$

così il problema produrrà una valutazione superiore, infatti adesso se l'interferenza su un certo segnale da antenna a a centroide c arriva ad essere più alta di $\rho_{c,k}^a$ ma più bassa di $\rho_{c,k+1}^a$ allora nella nostra approssimazione calcoleremo $F(\text{SINR})$ come se l'interferenza fosse $\rho_{c,k}^a$, cioè stiamo approssimando $F()$ dall'alto, questo fa sì che otteniamo una valutazione superiore della funzione obiettivo. Inoltre otterremo anche una valutazione inferiore del problema non approssimato data dal valore della funzione obiettivo calcolata nella soluzione ammissibile data sempre dagli (x,n,m) ottenuti risolvendo l'approssimazione del problema.

3.1 Implementazione

L'implementazione di questo modello, come degli altri che presenteremo nel seguito, è stato un lavoro lungo e complicato di cui non riporteremo dettagliatamente le parti più tecniche, che risulterebbero noiose al lettore. Cercheremo invece di dare un'idea delle principali scelte che sono state fatte.

Per implementare questo modello abbiamo utilizzato IBM ILOG CPLEX Optimization Studio. Quest'ultimo è un toolkit per il supporto del processo decisionale analitico per il rapido sviluppo e implementazione di modelli di ottimizzazione utilizzando una programmazione matematica e con vincoli (per informazioni più dettagliate visitare la seguente pagina <http://www->

03.ibm.com/software/products/it/ibmilogcpleoptistud). Ci siamo interfacciati con CPLEX attraverso un software che abbiamo scritto in linguaggio C++ utilizzando l'IDE VisualStudio.

Per costruire il modello in modo tale che CPLEX lo possa risolvere abbiamo scritto la matrice dei coefficienti in una variabile di tipo opportuno.

Per quanto riguarda la gestione dell'istanza da ricevere in input abbiamo preferito dividere in due gruppi i dati in entrata scrivendoli su due file diversi, questo perché alcuni dati sono propri dell'istanza in questione, di tipo geometrico se vogliamo, mentre altri possono essere modificati senza cambiare i primi. Ad esempio il numero di antenne e di centroidi e le potenze dei segnali da un'antenna a un centroide sono dati caratteristici dell'istanza che dipendono da come sono disposti nello spazio antenne e utenti, mentre l'energia totale disponibile e le domande dei centroidi sono dati che possono essere cambiati senza modificare i primi.

Un'altra scelta è stata quella di utilizzare come base su cui costruire il modello quello scritto per trovare i $\rho_{c,\max}^a$ (anche per questi sotto-problemi abbiamo utilizzato CPLEX). Infatti i vincoli scritti per calcolare questi valori sono compresi in quelli del modello principale, è stato dunque sufficiente aggiungere i vincoli mancanti e modificare opportunamente la funzione obiettivo.

Una volta ottenuti i vari $\rho_{c,\max}^a$ abbiamo dovuto porre attenzione nel verificare che questo valore non fosse minore di P_c^a/σ_{\max} ossia che, anche con interferenza massima, il segnale passi con il massimo della banda (ossia η_{\max}). In questo caso abbiamo deciso di eliminare la discretizzazione della relativa variabile e impostare il fattore moltiplicativo della m_c^a corrispondente a η_{\max} nel vincolo 3.3.

Per finire, abbiamo utilizzato la soluzione restituita da CPLEX per calcolare il valore della funzione obiettivo non approssimata in modo da sapere quanto effettivamente è buona la soluzione che abbiamo ottenuto e da avere un lower bound per il problema esatto.

3.2 Sperimentazione

Grazie alle istanze prodotte dal software del dipartimento d'ingegneria dell'informazione abbiamo testato il modello su alcune istanze, facendo variare l'energia totale disponibile e la costante di discretizzazione K . Abbiamo inoltre scelto di fermare CPLEX dopo 60s (questo si ottiene facilmente settando un opportuno parametro) anche nel caso in cui non abbia ancora risolto completamente il problema, in modo da poter avere un'idea della qualità dei risultati ottenuti anche con tempi bassi.

La prima istanza è composta da un solo esagono, dunque 1 macro e 2 micro, e 25 centroidi. I risultati sono i seguenti:

K	Time	Max Energy	Result	True result	Upper bound
20	98,55s	100	24,1237	23,8336	24,1237
10	13,84s	100	25	21,4667	25
20	32,78s	60	23,8401	23,6612	23,8401
10	3,33s	60	24,8786	22,4429	24,8786
20	8,8s	50	23,4997	23,428	23,4997
10	1,34s	50	23,5931	22,4066	23,5931
20	1,33s	40	22,5571	22,5502	22,5571
10	0,44s	40	22,5674	22,5502	22,5674
20	0,33s	20	10,3529	10,3529	10,3529
10	0,11s	20	10,3529	10,3529	10,3529

La seconda istanza è composta da due esagoni, dunque 2 macro e 4 micro, e con 11 centroidi nel primo esagono e 8 nel secondo centroidi. I risultati sono i seguenti:

K	Time	Max Energy	Result	True result	Upper bound
20	0,36s	70	19	18,7909	19
10	0,17s	70	19	18,4576	19
20	5,66s	60	18,4197	17,9035	18,4197
10	1,33s	60	18,9563	17,0425	18,9563
20	2,97s	50	18,0683	17,8203	18,0683
10	0.6s	50	18,2578	17,5671	18,2578
20	1,05s	40	17,8485	17,6627	17,8485
10	0,64s	40	18	17,6115	18
20	0.2s	20	10.0633	10.0633	10.0633
10	0.09s	20	10.0633	10.0633	10.0633

La terza istanza è composta da due esagoni, dunque 2 macro e 4 micro, e con 23 centroidi nel primo esagono e 21 nel secondo centroidi. I risultati sono i seguenti:

K	Time	Max Energy	Result	True result	Upper bound
20	60s	100	39,3221	38,8254	43,3636
10	22,63s	100	41,2072	38,2224	41,2072
20	60s	90	38,7781	38,2791	44
10	35,11s	90	40,4571	39,0741	40,4571
20	60s	80	38,5836	38,1827	41,5335
10	38,28s	80	40,0449	39,084	40,0449
20	41,52s	70	38,8902	38,3291	38,8902
10	60s	70	39,0549	38,2507	39,1239
20	1.86s	50	30.1176	30.1176	30.1176
10	1.00s	50	30.1176	30.1176	30.1176

In fine abbiamo testato l'ultima istanza dando un budget di energia di 110 fermando il solutore a determinati tempi e poi facendolo ripartire da dove l'avevamo interrotto per vedere come varia la soluzione fornita lasciando lavorare CPLEX con tempi diversi

K=20	Time	Result	True Result	Upper bound
	60s	40,0913	39,4067	44
	180s	40,0913	39,4067	44
	480s	40,2527	39,2518	42,5768
	1080s	40,4729	39,4662	41,8126

Capitolo 4

Riformulazione quadratica del problema

In questa sezione vedremo come riformulare il problema, senza approssimarlo, in modo tale che gli elementi non lineari e non convessi del problema diminuiscano e siano più facilmente trattabili, in particolare renderemo 2.6 quadratico. Prima di tutto è necessario riformulare il suddetto vincolo, per farlo notiamo che la $F()$ è lineare a tratti dunque possiamo riuscire a portare la m del prodotto $m_c^a F()$ dentro F . Definiamo

$$\alpha = \frac{\eta_{\max}}{\sigma_{\max} - \sigma_{\min}}$$

$$\beta = \frac{-\sigma_{\min}\eta_{\max}}{\sigma_{\max} - \sigma_{\min}}$$

così riscriviamo 2.4:

$$v_c \leq \sum_{a \in S(c)} w_c^a / D_c \quad c \in C$$

$$w_c^a \leq m_c^a \eta_{\max} \quad (a, c) \in S \quad (4.1)$$

$$w_c^a \leq \alpha(SINR(n)_c^a m_c^a) + m_c^a \beta \quad (a, c) \in S \quad (4.2)$$

dove 4.1 garantisce che anche nel caso in cui il SINR superi σ_{\max} la trasmissione avvenga comunque a velocità η_{\max} .

Adesso andando a esplicitare il primo prodotto otteniamo:

$$SINR(n)_c^a m_c^a = \frac{m_c^a P_{c,a}}{NG + \sum_{a' \in A(c) \setminus a} \bar{P}_c^{a'} n_{a'}}$$

così ponendo

$$r_c^a = NG + \sum_{a' \in A(c) \setminus a} \bar{P}_c^{a'} n_{a'} \quad (a, c) \in S$$

$$\rho_{c,0}^a \leq r_c^a \leq \rho_{c,\max}^a \quad (a, c) \in S$$

possiamo riformulare il tutto con

$$w_c^a \leq \alpha z_c^a + \beta m_c^a \quad (a, c) \in S$$

$$z_c^a \leq (P_c^a / r_c^a) m_c^a \quad (a, c) \in S$$

Scriviamo dunque il seguente modello:

$$\max \sum_{c \in C} v_c \quad (4.3)$$

$$v_c \leq \sum_{a \in S(c)} w_c^a / D_c \quad c \in C \quad (4.4)$$

$$w_c^a \leq m_c^a \eta_{\max} \quad (a, c) \in S \quad (4.5)$$

$$w_c^a \leq \alpha z_c^a + m_c^a \beta \quad (a, c) \in S \quad (4.6)$$

$$z_c^a \leq (P_c^a / r_c^a) m_c^a \quad (a, c) \in S \quad (4.7)$$

$$r_c^a = NG + \sum_{a' \in A(c) \setminus a} \bar{P}_c^{a'} n_{a'} \quad (a, c) \in S \quad (4.8)$$

$$\rho_{c,0}^a \leq r_c^a \leq \rho_{c,\max}^a \quad (a, c) \in S \quad (4.9)$$

4.1 Implementazione

Il solutore che abbiamo scelto per risolvere questo modello è Couenne (Convex Over and Under ENvelopes for Nonlinear Estimation), che utilizza un'algoritmo branch&bound per risolvere problemi MINLP. Per interfacciarci con

questo solutore abbiamo scritto un software in JuMP, un linguaggio di programmazione specifico per la modellizzazione e l'ottimizzazione matematica.

Come nel primo modello dividiamo i dati in due tipi, risolviamo i mini problemi per il calcolo dei $\rho_{c,max}^a$ e controlliamo che questi ultimi non siano minori di P_c^a/σ_{max} .

Dai primi test è venuto fuori che Couenne ed in particolare Clp, il solutore lineare usato nel branch&bound, non riusciva a trattare in modo appropriato i numeri particolarmente piccoli, nell'ordine di 10^{-14} , tipici delle istanze del problema. Così è stato necessario aumentare il lower bound delle variabili r_c^a da NG a $\max\{NG, P_c^a/\sigma_{max}\}$, abbiamo dunque dovuto modificare il vincolo (4.8) mettendo un maggiore uguale al posto dell'uguale. Così le soluzioni date da Couenne sono passate dall'essere completamente nulle ad essere paragonabili a quelle fornite da CPLEX.

Purtroppo questo non è bastato, su istanze grosse Clp continuava a non lavorare correttamente e alla fine la soluzione è stata quella di riscaldare le r_c^a , abbiamo scritto $\rho_{c,1}^a 10^5 \leq r_c^a \leq 10^5 \rho_{c,max}^a$ e poi abbiamo riscaldato i vincoli (4.7) e (4.8) moltiplicando in entrambi il lato destro per 10^5 .

Capitolo 5

Modello con discretizzazione binaria

Vediamo adesso come, partendo del precedente modello, possiamo formulare un'approssimazione del problema che, a differenza della prima, non renda il problema lineare ma convesso. Quello che faremo è usare una discretizzazione binaria per le variabili r e z e usare alcune proprietà di questa discretizzazione per rendere 4.7 convesso. Scegliamo due interi positivi n e k , e definiamo i vettori p e q come segue

$$p = [1, 2, 4, \dots, 2^{n-1}]$$

$$q = [1, 2, 3, \dots, 2^{k-1}]$$

supponiamo inoltre di sapere che

$$\infty < \underline{z} \leq z \leq \bar{z} < \infty$$

$$\infty < \underline{r} \leq r \leq \bar{r} < \infty$$

allora possiamo discretizzare le variabili identificando ogni valore della discretizzazione con i vettori $\alpha \in \{0, 1\}^n$, $\beta \in \{0, 1\}^k$ in modo da avere

$$z = \underline{z} + (\bar{z} - \underline{z}) \alpha$$

$$r = \underline{r} + (\bar{r}'q)\beta$$

con

$$\bar{z}' = (\bar{z} - \underline{z})/(2^n - 1)$$

$$\bar{r}' = (\bar{r} - \underline{r})/(2^k - 1)$$

Abbiamo discretizzato le variabili z ed r dividendo gli intervalli $[\underline{z}, \bar{z}]$ e $[\underline{r}, \bar{r}]$ rispettivamente in $2^n - 1$ e $2^k - 1$ intervallini grandi \bar{z}' e \bar{r}' . Vediamo come questo può aiutarci a rendere convesso 4.7, moltiplichiamo questo vincolo per r (che è sempre strettamente positiva)

$$zr \leq mP$$

$$(\underline{z} + (\bar{z}'p)\alpha)(\underline{r} + (\bar{r}'q)\beta) \leq mP$$

matricialmente

$$[\alpha^t, \beta^t] \begin{bmatrix} 0 & p^t q/2 \\ q^t p/2 & 0 \end{bmatrix} [\alpha, \beta] \leq \frac{Pm - rz - \underline{r}(\bar{z}'p)\alpha - \underline{z}(\bar{r}'q)\beta}{\bar{r}'\bar{z}'}$$

Dove abbiamo usato che $\bar{r}'\bar{z}'$ è positivo. Chiamiamo J il lato destro dell'ultima equazione e Q la matrice del lato sinistro

$$[\alpha^t, \beta^t]Q[\alpha, \beta] \leq J$$

con Q ancora non convessa ovviamente. Adesso, dato che i vettori α e β hanno elementi in $\{0, 1\}$ vale la relazione

$$\lambda_i \alpha_i^2 - \lambda_i \alpha_i = 0 \quad \gamma_i \beta_i^2 - \gamma_i \beta_i = 0 \quad \forall \lambda_i, \gamma_i \in \mathbb{R}$$

Così aggiungo al lato destro

$$\sum_i \lambda_i \alpha_i^2 - \lambda_i \alpha_i \quad e \quad \sum_i \gamma_i \beta_i^2 - \gamma_i \beta_i$$

ottenendo

$$[\alpha^t, \beta^t]Q[\alpha, \beta] + \sum_i \lambda_i \alpha_i^2 + \sum_i \gamma_i \beta_i^2 \leq J + \sum_i \lambda_i \alpha_i + \sum_i \gamma_i \beta_i$$

$$[\alpha^t, \beta^t] \begin{bmatrix} \Lambda & p^t q/2 \\ q^t p/2 & \Gamma \end{bmatrix} [\alpha, \beta] \leq J + \sum_i \lambda_i \alpha_i + \sum_i \gamma_i \beta_i$$

dove $\Gamma = \text{diag}\{\gamma_i\}$ e $\Lambda = \text{diag}\{\lambda_i\}$

Così basta riuscire a trovare Γ e Λ in modo tale che la matrice del lato sinistro risulti definita positiva minimizzando la somma degli elementi per non aumentare troppo la regione di ammissibilità nel rilassamento continuo e per evitare errori numerici del solutore. Ciò corrisponde a risolvere il seguente problema SDP

$$\min \left\{ \sum_i \lambda_i + \sum_i \gamma_i : \begin{bmatrix} \Lambda & p^t q/2 \\ q^t p/2 & \Gamma \end{bmatrix} \succ 0 \right\}$$

Tutto ciò ci porta al seguente modello

$$\max \sum_{c \in C} v_c \quad (5.1)$$

$$v_c \leq \sum_{a \in S(c)} w_c^a / D_c \quad c \in C \quad (5.2)$$

$$w_c^a \leq m_c^a \eta_{\max} \quad (a, c) \in S \quad (5.3)$$

$$w_c^a \leq \alpha z_c^a + m_c^a \beta \quad (a, c) \in S \quad (5.4)$$

$$z_c^a \leq m_c^a \sigma_{\max} \quad (a, c) \in S \quad (5.5)$$

$$\begin{aligned} & \sum_{i=0}^{m-1} \sum_{j=0}^{k-1} \alpha_{i,c}^a \beta_{j,c}^a 2^{i+j} + \sum_i \lambda_i (\alpha_{i,c}^a)^2 + \sum_i \gamma_i (\beta_{i,c}^a)^2 \\ & \leq \frac{P_c^a m_c^a - r_c^a z_c^a - r_c^a (\overline{z_c^a} p) \alpha_c^a - \overline{z_c^a} (\overline{r_c^a} q) \beta_c^a}{\overline{r_c^a} \overline{z_c^a}} + \\ & + \sum_i \lambda_i \alpha_{i,c}^a + \sum_i \gamma_i \beta_{i,c}^a \quad (a, c) \in S \end{aligned} \quad (5.6)$$

$$r_c^a \geq NG + \sum_{a' \in A(c) \setminus a} \overline{P_c^{a'}} n_{a'} \quad (a, c) \in S \quad (5.7)$$

$$\underline{r_c^a} \leq r_c^a \leq \overline{r_c^a} \quad (a, c) \in S \quad (5.8)$$

$$\underline{z_c^a} \leq z_c^a \leq \overline{z_c^a} \quad (a, c) \in S \quad (5.9)$$

$$r_c^a = \underline{r_c^a} + \overline{r_c^a} \sum_{i=0}^{k-1} \beta_{i,c}^a 2^i \quad (a, c) \in S \quad (5.10)$$

$$z_c^a = \underline{z_c^a} + \overline{z_c^a} \sum_{i=0}^{m-1} \alpha_{i,c}^a 2^i \quad (a, c) \in S \quad (5.11)$$

$$\alpha_{i,c}^a \in \{0, 1\} \quad i \in \{0 \dots n-1\} \quad (5.12)$$

$$\beta_{i,c}^a \in \{0, 1\} \quad i \in \{0 \dots k-1\} \quad (5.13)$$

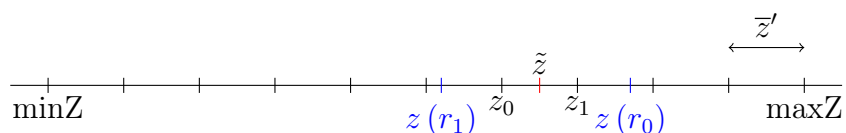
Si noti che la scelta di mettere un maggiore-uguale in in 5.7 è obbligata, altrimenti il problema risulterebbe vuoto la maggior parte delle volte. Infatti anche se le variabili n_a sono continue, sono *poche* rispetto alle r_c^a , quindi può essere impossibile trovare valori delle n_a che rendano contemporaneamente veri tutti i vincoli sulle r_c^a . Per rendere evidente questo problema guardiamo cosa succede in un esempio in bassa dimensione, prendiamo un istanza con 2 antenne e 2 centroidi, le equazioni relative alla prima antenna risultano essere : $r_0^1 = NG + P_1^1 * n_1$ $r_0^0 = NG + P_1^0 * n_1$ quindi $n_1 = (r_0^1 - NG)/P_1^1$
 $n_1 = (r_0^0 - NG)/P_1^0$ $(r_0^1 - NG)/P_1^1 = (r_0^0 - NG)/P_1^0$ dove r_0^1 e r_0^0 possono

assumere un numero finito di valori.

Vediamo come modificare il modello per ottenere una valutazione superiore del problema. Supponiamo di essere in una determinata configurazione e di interessarci a un singolo segnale antenna-centroide. Il valore dell'interferenza che realmente agisce su questo segnale ($NG + \sum Pn/B$), che chiameremo \tilde{r} , sarà compreso tra due valori $\min R$ e $\max R$. La variabile r del nostro modello può prendere solo alcuni valori dell'intervallo $[\min R, \max R]$, in particolare dato che nel modello è presente il vincolo $r \geq \tilde{r}$ e dato che il modello cercherà di diminuire r , quest'ultima prenderà il più piccolo valore della discretizzazione maggiore di \tilde{r} che chiameremo r_1 . Se cambiamo il vincolo di prima con $r \geq \tilde{r} - \bar{r}'$ (\bar{r}' era l'ampiezza degli intervallini in cui è diviso $[\min R, \max R]$), il valore di r sarà quello del più grande valore della discretizzazione minore di \tilde{r} , chiamiamo questo valore r_0 . La situazione è quella della figura 1.



Usiamo la stessa notazione per la variabile z , il vincolo del modello che limita quest'ultima è $z \leq z(r) = \tilde{z}$ dove $z(x)$ è $P * m/x$. Se assumiamo di aver cambiato il vincolo della r in modo che prenda il valore r_0 allora z prenderà il più grosso valore della discretizzazione minore di $z(r_0)$, questo però potrebbe comunque essere minore di \tilde{z} , infatti essendo $z()$ decrescente il valore di $z(r_0)$ è compreso tra \tilde{z} e $\max Z$ dunque se $z(r_0) \leq z_1$ avremo $z = z_0$. Se invece cambiamo il vincolo con $z \leq z(r) + \bar{z}'$ allora sicuramente $z \geq z_1 > \tilde{z}$. La figura 2 illustra la situazione. Così otteniamo un'approssimazione superiore di z e quindi della soluzione del problema non approssimato



Un altro modo per avere una valutazione superiore del problema è guardare le soluzioni del rilassamento continuo del problema, le quali vengono automaticamente calcolate da CPLEX per fornire upper bound per il problema approssimato.

Così formulato non è per niente ovvio che il rilassamento continuo del modello fornisca un upper bound del problema non discretizzato. Data una soluzione del problema esatto dimostriamo che esiste una soluzione ammissibile del rilassamento continuo del modello discretizzato che ha funzione obbiettivo con lo stesso valore. Iniziamo col dimostrare che data una configurazione del sistema, scelti $a \in A$ e $c \in C$ e detti $\tilde{r} = N_G + \sum_{a' \in A(c) \setminus a} \bar{P}_c^{a'} n_{a'}$ e $\tilde{z} = m_c^a SINR_c^a(n)$ esistono dei valori $\alpha_i, \beta_i \in [0,1]$ tali che $r = \underline{r} + \bar{r}' \sum_{i=0}^{k-1} \beta_i 2^i = \tilde{r}$ e $z = \underline{z} + \bar{z}' \sum_{i=0}^{m-1} \beta_i 2^i = \tilde{z}$.

Facciamo il ragionamento solo per r , per z si procede in modo identico. Dividiamo il problema in due casi, se $\tilde{r} < \bar{r} + 2^{n-1} \bar{r}'$ allora ponendo $\alpha_i = 0$ $i < n-1$ e $\alpha_{n-1} = x$ esiste sicuramente x in $[0,1]$ t.c. $r = \underline{r} + x \bar{r}' 2^{n-1} = \tilde{r}$. Se $\tilde{r} > \underline{r} + (2^{n-1} - 1) \bar{r}'$ allora ponendo $\alpha_i = 1$ $i < n-1$ e $\alpha_{n-1} = x$ esiste x in $[0,1]$ t.c. $r = \underline{r} + \sum_{i < n-1} 2^i \bar{r}' + x \bar{r}' 2^{n-1} = \tilde{r}$. Dato che $\underline{r} + (2^{n-1} - 1) \bar{r}' < \underline{r} + 2^{n-1} \bar{r}'$ abbiamo coperti tutti i casi ammissibili.

Dunque data una soluzione del problema non discretizzato trovo α_i e β_i in modo che il valore della soluzione calcolata nel rilassamento continuo del modello discretizzato abbia lo stesso valore della funzione obbiettivo, questo infatti dipende solo dai valori di z_c^a che abbiamo mostrato essere uguali a quelli del problema reale.

Resta da vedere se questa soluzione, completata settando i valori delle restanti variabili uguali a quelli delle variabili nella soluzione che vogliamo rappresentare, è ammissibile. Dato che le uniche variabili diverse tra le due soluzioni sono le α_i e i β_i sicuramente tutti i vincoli sono rispettati, l'unico da verificare è (5.6), questo è identico a quello del problema non perturbato a cui abbiamo sommato al lato sinistro $\sum_i \lambda_i \alpha_i^2 - \lambda_i \alpha_i$ e $\sum_i \gamma_i \beta_i^2 - \gamma_i \beta_i$ che adesso non sono più 0. Essendo $\alpha_i, \beta_i \in [0,1]$ i valori che abbiamo sommato al lato sinistro sono negativi, dunque $zr - K < zr < Pm$ con $K > 0$ ossia il vincolo è rispettato anche nel rilassamento continuo.

Tutto il ragionamento che abbiamo fatto funziona solo se la variabile binaria che ha più peso è continua, dunque se vogliamo usare anche gli upper bound dei nodi successivi, e quindi il best bound finale, dovremo togliere il vincolo d'integrità all'ultima variabile binaria. Questo risultato ci autorizza dunque ad utilizzare gli upper bound che troviamo per il problema

discretizzato anche per il problema reale.

5.1 Implementazione

Come per il primo modello approssimato per implementare questo abbiamo usato Cplex e Visual Studio. Analogamente agli altri modelli dividiamo i dati in due tipi, calcoliamo i $\rho_{c,max}^a$ e controlliamo che siano maggiori di P_c^a/σ_{max} .

Uno dei principali problemi di questo modello è stato il tempo necessario a fornire una buona soluzione, infatti, spesso, questo è incredibilmente lungo. Per arginare questa complicazione abbiamo cercato di rimpicciolire il più possibile la regione ammissibile del problema. Così abbiamo inserito il vincolo

$$z_c^a \leq \sigma_{max} m_c^a$$

che forza le variabili z a non superare la soglia oltre la quale comunque si ha sempre trasmissione a velocità η_{max} . Un'altro passo fondamentale è decidere quali saranno i bound da dare alle variabili z_c^a ed r_c^a . Questa è una scelta molto importante dato che questi valori, oltre a contribuire alla definizione della regione ammissibile, determinano l'ampiezza degli intervalli che vogliamo discretizzare, è dunque essenziale scegliere i bound in modo che quest'intervallo risulti il più piccolo possibile, così da avere buoni risultati in termini di approssimazione anche con poche variabili binarie.

Per quanto riguarda r_c^a dato che il lato destro del vincolo 5.7 può effettivamente essere $\rho_{c,max}^a$ non possiamo far altro che mettere come upper bound di r_c^a quest'ultimo valore. Possiamo invece migliorare il lower bound settandolo a $\rho_{c,1}^a$ invece di $\rho_{c,0}^a$.

Per il lower bound di z_c^a non c'è scelta possibile diversa da zero, mentre la scelta più naturale per l'upper bound è $P*B/\rho_{c,1}^a$ dato che è questo il massimo valore che può assumere il lato destro di 4.7. In realtà, in alcuni casi possiamo migliorare questo bound, se il budget di energia P è abbastanza basso allora il massimo numero di blocchi che un'antenna può allocare non è B ma $(P - P_a)/p_a$, ossia non c'è abbastanza energia disponibile per allocare B blocchi per l'antenna a . Possiamo migliorare dunque l'upper bound di z_c^a settandolo a

$\min(B, (P - P_a)/p_a) * P_c^a / \rho_{c,1}^a$. Queste modifiche si sono rivelate fondamentali per migliorare le performance di questo modello anche con piccoli valori di n e k

Come detto in precedenza, per calcolare i vettori λ e γ , abbiamo risolto il seguente problema SDP.

$$\min \left\{ \sum_i \lambda_i + \sum_i \gamma_i : \begin{bmatrix} \Lambda & p^t q/2 \\ q^t p/2 & \Gamma \end{bmatrix} \succ 0 \right\}$$

Per risolverlo abbiamo usato il solutore SeDuMi interfacciandoci con esso attraverso YALMIP, un ToolBox di MATLAB. Abbiamo risolto tutte il problema per tutte le coppie (n, k) con $n, k \in \{2, \dots, 16\}$ e abbiamo salvato i risultati su file, ogni volta che lanciamo il software questo andrà a leggere da questo file i valori di λ e γ opportuni.

Capitolo 6

Sperimentazione finale

Dopo aver formulato i modelli ed aver fatto qualche test preliminare per controllare la qualità delle soluzioni calcolate ed in quanto tempo ne viene fornita una buona, abbiamo fatto una ulteriore sperimentazione su molte istanze per poter confrontare tra loro i vari modelli.

Abbiamo scelto 4 diversi scenari: uno con un esagono, un altro con 3, uno con 5 e uno con 7, mettendo all'interno di ognuno prima 5 centroidi poi 10 e infine 15. Per riuscire ad approssimare la curva pareto-efficiente abbiamo testato queste istanze dando 5 diversi budget di energia. Infine per avere un'idea di come la soluzione evolve nel tempo, e quindi di quali modelli sono più veloci a calcolarne una buona e di quali invece partono più lenti ma poi recuperano, abbiamo riportato la soluzione e i bound calcolati dai solutori dopo 1 minuto, dopo 2, dopo 4 e dopo 8.

Di seguito descriveremo i risultati ottenuti in alcune tabelle ed alcuni grafici nei quali potranno apparire i seguenti campi:

- Tipo di istanza: tre numeri rappresentanti numero di esagoni, di utenti per esagono e budget di energia fornito. Quest'ultimo numero, detto P_{\max} il minimo budget che consente il raggiungimento dell'ottimo più alto per l'istanza in questione, va letto secondo la seguente tabella:

1	2	3	4	5
$10\%P_{\max}$	$35\%P_{\max}$	$50\%P_{\max}$	$70\%P_{\max}$	$100\%P_{\max}$

- Gap a radice= $\frac{(\text{Upper bound a radice}-\text{Best lower bound})}{\text{Best lower bound}}$
- Gap primale= $\frac{(\text{Upper bound}-\text{Best lower bound})}{\text{Best lower bound}}$
- Gap duale= $\frac{(\text{Best upper bound}-\text{Lower bound})}{\text{Lower bound}}$
- Gap inerente= $\frac{(\text{Upper bound}-\text{Lower bound})}{\text{Lower bound}}$

Per calcolare i best bound che compaiono in queste formule abbiamo lanciato il primo modello con $K=16$ dando un limite di tempo di un ora e venti per ogni istanza. Di seguito riportiamo la tabella dei gap inerenti relativi a questa sperimentazione.

Tabella 6.1: Modello 1 K=16 con tempo limite alto

Istanza	Gap inerente %			
	10 min	20 min	40 min	80 min
1-5-1	0,00	0,00	0,00	0,00
1-5-2	0,00	0,00	0,00	0,00
1-5-3	0,00	0,00	0,00	0,00
1-5-4	0,00	0,00	0,00	0,00
1-5-5	0,00	0,00	0,00	0,00
1-10-1	0,00	0,00	0,00	0,00
1-10-2	0,00	0,00	0,00	0,00
1-10-3	0,00	0,00	0,00	0,00
1-10-4	0,00	0,00	0,00	0,00
1-10-5	0,78	0,78	0,78	0,78
1-15-1	0,00	0,00	0,00	0,00
1-15-2	0,00	0,00	0,00	0,00
1-15-3	0,00	0,00	0,00	0,00
1-15-4	0,00	0,00	0,00	0,00
1-15-5	0,00	0,00	0,00	0,00
3-5-1	0,00	0,00	0,00	0,00
3-5-2	0,39	0,39	0,39	0,39
3-5-3	0,63	0,63	0,63	0,63
3-5-4	5,49	5,49	5,49	5,49
3-5-5	7,73	7,73	7,73	7,73
3-10-1	0,00	0,00	0,00	0,00
3-10-2	0,83	0,83	0,83	0,83
3-10-3	1,48	1,48	1,48	1,48
3-10-4	5,80	5,80	5,80	5,80
3-10-5	5,58	6,35	6,35	6,35
3-15-1	0,00	0,00	0,00	0,00
3-15-2	0,85	0,85	0,85	0,85
3-15-3	14,13	2,83	2,83	2,83
3-15-4	18,05	9,28	4,07	4,07
3-15-5	23,41	24,46	24,88	21,66
5-5-1	0,00	0,00	0,00	0,00
5-5-2	1,73	1,73	1,73	1,73
5-5-3	0,93	0,93	0,93	0,93
5-5-4	3,52	3,52	3,52	3,52
5-5-5	15,43	13,40	12,31	12,31
5-10-1	0,00	0,00	0,00	0,00
5-10-2	1,02	1,02	1,02	1,02
5-10-3	9,52	7,64	7,15	5,85
5-10-4	15,00	12,35	10,55	7,76
5-10-5	13,77	13,76	13,76	13,83
5-15-1	0,00	0,00	0,00	0,00
5-15-2	10,01	2,61	0,71	0,71
5-15-3	26,05	14,89	12,45	9,20
5-15-4	16,46	12,07	10,59	9,30
5-15-5	23,43	13,24	12,46	12,46
7-5-1	0,53	0,53	0,53	0,53
7-5-2	5,05	0,07	0,07	0,07
7-5-3	19,30	5,51	3,79	3,79
7-5-4	22,88	16,28	16,28	16,28
7-5-5	14,35	14,35	14,35	14,47
7-10-1	0,38	0,38	0,38	0,38
7-10-2	19,72	10,16	7,75	7,22
7-10-3	30,83	22,61	18,88	17,83
7-10-4	32,78	26,87	26,87	21,69
7-10-5	31,57	19,93	19,85	19,32
7-15-1	0,48	0,48	0,48	0,48
7-15-2	206,73	21,79	19,60	16,48
7-15-3	23,28	23,28	23,28	23,28
7-15-4	33,44	27,65	27,65	27,65
7-15-5	254,42	28,38	28,38	28,38

Per il primo modello abbiamo scelto la versione che approssima il problema esatto dall'alto, testandolo prima con $K=8$ e poi con $K=16$

Anche per il modello con la discretizzazione binaria abbiamo scelto una variante che fornisce un'approssimazione superiore, come già visto esistono due modi per ottenerla a partire da questo modello. Per scegliere quale utilizzare li abbiamo testati entrambi con 4 bit per variabile sulle istanze su cui stiamo facendo la sperimentazione. I risultati, che sono riportati nella tabella sottostante, indicano che sugli 8 minuti la scelta di lasciare continua la variabile rappresentante il bit più importante risulta migliore, non tanto nel fornire un buon upper bound, compito nel quale i due modelli si equivalgono la maggior parte delle volte, ma nel fornire un lower bound migliore. Questo crediamo avvenga perchè il modello con il bit più significativo continuo è considerevolmente più veloce dell'altro, che in soli 8 minuti non riesce a dare delle buone soluzioni.

Tabella 6.2: Differenza tra i vari gap del modello 2 con il bit finale continuo e l'altra versione che fornisce un'approssimazione superiore

Istanza	Gap radice	Gap primale	Gap duale
1-5-1	0,00	0,00	0,00
1-5-2	0,00	0,00	0,00
1-5-3	0,00	0,00	0,00
1-5-4	0,00	0,00	0,00
1-5-5	0,00	-94,03	0,00
1-10-1	0,00	0,00	0,00
1-10-2	0,00	0,00	0,00
1-10-3	0,00	0,00	0,00
1-10-4	0,00	0,00	0,00
1-10-5	0,00	-210,73	0,00
1-15-1	0,00	0,00	0,00
1-15-2	0,00	0,00	0,00
1-15-3	0,00	0,00	0,00
1-15-4	0,00	0,00	0,00
1-15-5	0,00	-33,35	0,00
3-5-1	0,00	17,26	0,00
3-5-2	0,00	26,73	0,00
3-5-3	0,00	-4,10	0,00
3-5-4	0,00	41,81	0,00
3-5-5	0,00	75,26	0,00
3-10-1	0,00	1,61	0,00
3-10-2	-12,08	8,27	0,00
3-10-3	0,00	86,52	0,00
3-10-4	0,00	220,14	0,00
3-10-5	0,00	84,40	0,00
3-15-1	0,00	17,44	0,00
3-15-2	0,11	-115,51	0,00
3-15-3	0,00	-51,02	0,00
3-15-4	0,00	483,24	0,00
3-15-5	0,00	294,21	0,00
5-5-1	72,25	-1,50	0,00
5-5-2	0,00	-8,62	0,00
5-5-3	0,00	13,11	0,00
5-5-4	0,00	2,14	0,00
5-5-5	0,00	-24,41	0,00
5-10-1	0,00	53,97	0,00
5-10-2	0,50	-3,96	0,00
5-10-3	0,00	-17,51	0,00
5-10-4	0,00	98,46	0,00
5-10-5	0,00	383,27	0,00
5-15-1	0,00	-46,70	0,00
5-15-2	0,00	732,54	6,55
5-15-3	0,00	-1,08	0,00
5-15-4	0,00	141,23	0,00
5-15-5	0,00	281,35	0,00
7-5-1	0,00	7,35	0,00
7-5-2	0,00	-6,56	0,00
7-5-3	0,00	348,81	0,00
7-5-4	0,00	121,33	0,00
7-5-5	0,00	178,14	0,00
7-10-1	0,00	-63,71	0,00
7-10-2	-7,28	-33,63	3,01
7-10-3	0,00	445,37	0,00
7-10-4	0,00	328,25	0,00
7-10-5	0,00	304,04	0,00
7-15-1	0,00	2,40	11,93
7-15-2	0,00	217,32	0,00
7-15-3	0,00	-70,25	0,00
7-15-4	0,00	19,47	0,00
7-15-5	0,00	290,03	0,00

Una volta scelta la versione del secondo modello da utilizzare abbiamo fatto i test con 4 bit a variabile e con 8 bit variabile.

I risultati hanno evidenziato i seguenti trend:

- Su istanze piccole il primo modello da risultati migliori con $K=16$, non tanto nel dare dei buoni upper bound, cosa in cui è comunque leggermente migliore della versione con $K=8$, ma nella qualità dei lower bound. Questo perchè su istanze piccole il solutore riesce ad arrivare molto in profondità nell'espansione dell'albero delle soluzioni e quindi la versione che utilizza più variabili binarie e quindi una discretizzazione più fine ha il tempo di trovare soluzioni migliori. Per il secondo modello non ci sono differenze sostanziali tra le due versioni

Tabella 6.3: trend su bassa dimensione

Istanza	Modello	Gap radice	Gap primale				Gap duale			
			1 min	2 min	4 min	8 min	1 min	2 min	4 min	8 min
1-5-5	Mod 1, $K=8$	0,00	5,26	5,26	5,26	5,26	0,00	0,00	0,00	0,00
1-5-5	Mod 1, $K=16$	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-5-5	Mod 2, 4 bit	0,00	104,31	104,31	104,31	104,31	0,00	0,00	0,00	0,00
1-5-5	Mod 2, 8 bit	0,00	97,30	97,30	97,30	97,30	0,00	0,00	0,00	0,00
1-10-5	Mod 1, $K=8$	0,78	2,04	2,04	2,04	2,04	0,78	0,78	0,78	0,78
1-10-5	Mod 1, $K=16$	0,78	0,78	0,78	0,78	0,78	0,78	0,78	0,78	0,78
1-10-5	Mod 2, 4 bit	0,78	219,36	219,36	219,36	219,36	0,78	0,78	0,78	0,78
1-10-5	Mod 2, 8 bit	0,78	240,00	240,00	240,00	240,00	0,78	0,78	0,78	0,78
3-5-5	Mod 1, $K=8$	10,66	40,23	40,23	40,23	40,23	10,66	10,66	10,66	10,66
3-5-5	Mod 1, $K=16$	10,66	7,73	7,73	7,73	7,73	7,73	7,73	7,73	7,73
3-5-5	Mod 2, 4 bit	10,66	72,32	72,32	72,32	72,32	10,66	10,66	10,66	10,66
3-5-5	Mod 2, 8 bit	10,66	109,09	109,09	109,09	109,09	10,66	10,66	10,66	10,66

- Aumentare la dimensione delle istanze peggiora notevolmente i risultati, questo si può notare ad esempio confrontando i risultati ottenuti per istanze con circa 50 centroidi e istanze con circa 70. Se però aumentiamo solo il numero di antenne, i risultati potrebbero addirittura migliorare dato che sarà più facile soddisfare la domanda dei centroidi. Questo si può vedere nella tabella sottostante confrontando i risultati per le istanze 3-15-5 e 5-10-5 che hanno rispettivamente 45 e 50 centroidi e 9 e 15 antenne.

Tabella 6.4: trend sulle dimensioni aumentanti

Istanza	Modello	Gap radice	Gap primale				Gap duale			
			1 min	2 min	4 min	8 min	1 min	2 min	4 min	8 min
3-15-5	Mod 1 K=8	15,80	12,39	12,39	12,39	12,39	9,24	9,24	9,24	9,24
5-10-5	Mod 1 K=8	9,99	19,84	18,50	18,50	18,50	9,99	8,29	8,29	8,29
7-10-5	Mod 1 K=8	21,15	44,60	44,60	44,60	44,60	21,15	21,15	21,15	21,15
3-15-5	Mod 1 K=16	15,80	6,08	6,08	7,09	4,59	15,80	15,80	15,80	10,48
5-10-5	Mod 1 K=16	9,99	12,35	10,11	10,11	9,92	9,99	9,99	9,99	9,99
7-10-5	Mod 1 K=16	21,15	32,78	32,78	27,10	26,17	21,15	21,15	21,15	21,15
3-15-5	Mod 2 4 bit	15,80	183,75	183,75	183,75	183,75	15,80	15,80	15,80	15,80
5-10-5	Mod 2 4 bit	9,99	114,90	114,90	114,90	114,90	9,99	9,99	9,99	9,99
7-10-5	Mod 2 4 bit	21,15	159,06	159,06	159,06	159,06	21,15	21,15	21,15	21,15
3-15-5	Mod 2 8 bit	15,80	272,10	272,10	272,10	272,10	15,80	15,80	15,80	15,80
5-10-5	Mod 2 8 bit	9,99	194,52	194,52	194,52	194,52	9,99	9,99	9,99	9,99
7-10-5	Mod 2 8 bit	21,15	212,90	212,90	212,90	212,90	21,15	21,15	21,15	21,15

- Su istanze molto grosse il modello 1 con K=8 parte molto piu veloce della versione con K=16 che poi però tende a recuperare , questo è chiaramente dovuto alla differenza del numero di variabili binarie tra i due modelli. Inoltre può capitare che il lower bound del modello con K=8 vada peggiorando invece di migliorare, si veda cosa succede per l'istanza 7-15-5, questo è sicuramente dovuto all'approssimazione troppo blanda.

Il modello 2 riesce a fornire un migliore lower bound nella versione con 4 bit piuttosto che in quella a 8 bit.

Tabella 6.5: trend su grossa dimensione

Istanza	Modello	Gap radice	Gap primale				Gap duale			
			1 min	2 min	4 min	8 min	1 min	2 min	4 min	8 min
5-15-5	Mod 1 K=8	12,27	16,24	16,24	34,32	28,20	12,27	12,27	12,27	12,27
5-15-5	Mod 1 K=16	12,27	25,51	21,72	13,28	13,15	12,27	12,27	12,27	12,27
7-15-5	Mod 1 K=8	12,27	25,51	21,72	13,28	13,15	12,27	12,27	12,27	12,27
7-15-5	Mod 1 K=16	28,38	254,42	52,46	36,56	30,33	28,38	28,38	28,38	28,38
5-15-5	Mod 2 4 bit	12,27	270,86	270,86	270,86	270,86	12,27	12,27	12,27	12,27
5-15-5	Mod 2 8 bit	12,27	476,23	476,23	476,23	476,23	12,27	12,27	12,27	12,27
7-15-5	Mod 2 4 bit	28,38	169,30	169,30	169,30	169,30	28,38	28,38	28,38	28,38
7-15-5	Mod 2 8 bit	28,38	444,80	444,80	444,80	444,80	28,38	28,38	28,38	28,38

- Il modello 1 si è rivelato sostanzialmente migliore del modello 2, se

c'è già una grossa differenza sulla qualità dell'upper bound, quando si guardano i lower bound i risultati non sono neanche paragonabili

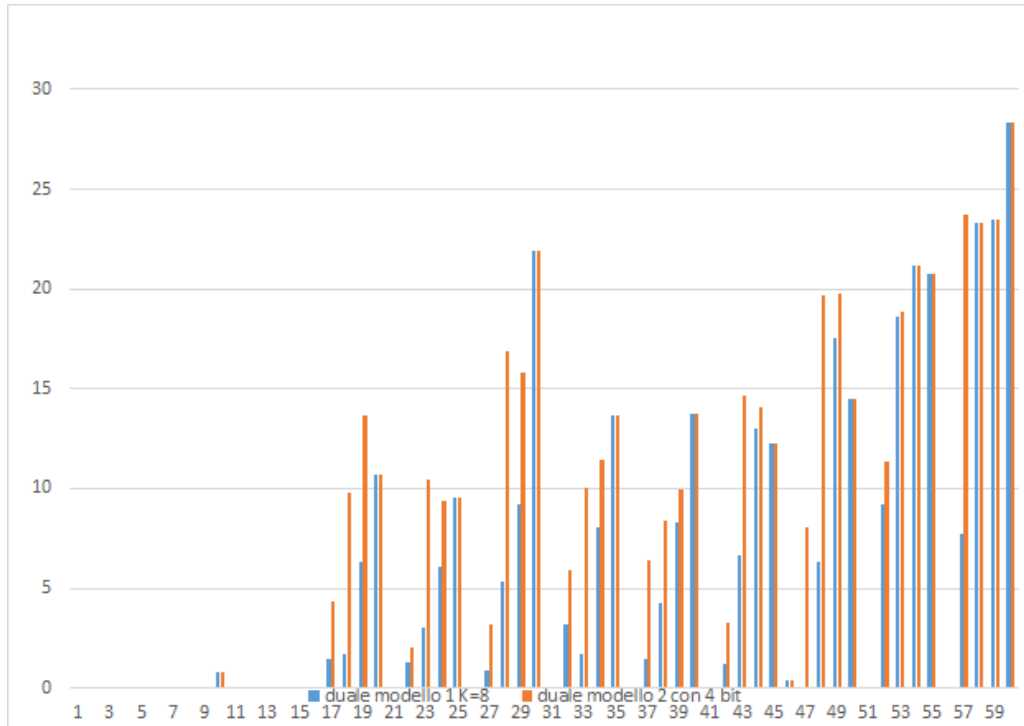


Figura 6.1: Confronto gap primali

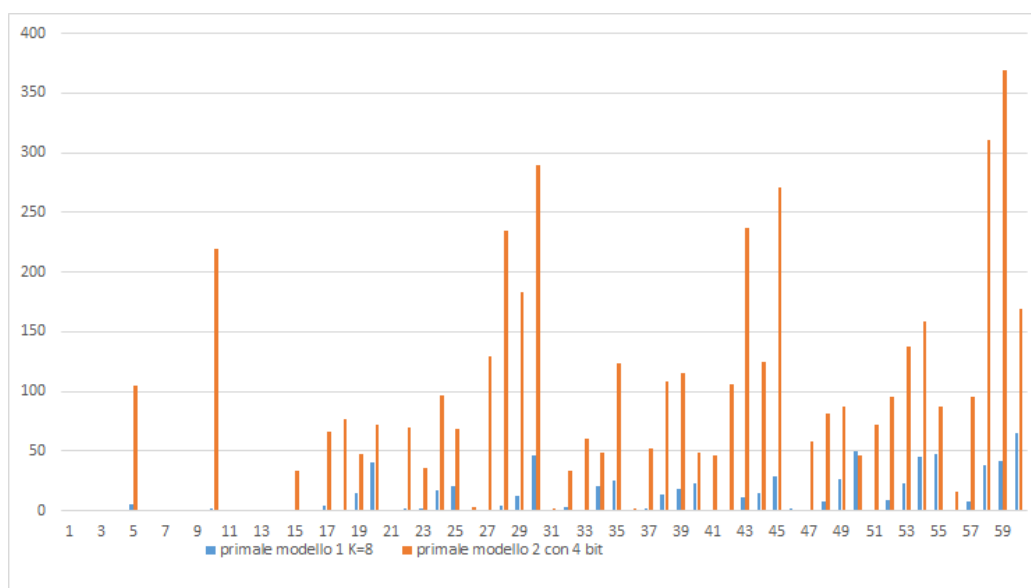


Figura 6.2: Confronto gap duali

- Il budget di potenza si è rivelato un fattore fondamentale nella convergenza dei modelli. Con budget di potenza bassi i modelli si sono mostrati tutti molto veloci. Mentre, quando l'energia fornita è sufficiente a soddisfare quasi tutta la domanda, i lower bound sono peggiorati moltissimo, mentre gli upper bound non così tanto. Questo perché gli upper bound banali (numero di centri) risultano vicini al valore esatto dell'ottimo, infatti, come si può notare dalla tabella sottostante, i gap duali dei vari modelli, relativi ad istanze con budget alto sono molto vicini tra di loro.

Tabella 6.6: trend sulla variazione del budget

Istanza	Modello	Gap radice	Gap primale				Gap duale			
			1 min	2 min	4 min	8 min	1 min	2 min	4 min	8 min
7-10-1	Mod 1 K=8	17,64	0,00	0,00	0,00	0,00	0,01	0,01	0,01	0,01
7-10-2	Mod 1 K=8	16,48	12,11	9,89	8,23	8,29	13,58	12,44	10,87	9,17
7-10-3	Mod 1 K=8	18,85	26,08	26,08	26,08	22,54	18,85	18,85	18,85	18,57
7-10-4	Mod 1 K=8	21,15	44,60	44,60	44,60	44,60	21,15	21,15	21,15	21,15
7-10-5	Mod 1 K=8	20,78	40,18	32,18	47,16	47,16	20,78	20,78	20,78	20,78
7-10-1	Mod 1 K=16	17,64	0,38	0,38	0,38	0,38	0,00	0,00	0,00	0,00
7-10-2	Mod 1 K=16	25,49	13,84	8,18	8,18	8,08	15,65	12,48	11,10	10,22
7-10-3	Mod 1 K=16	18,85	29,48	29,48	29,48	22,53	18,85	18,85	18,85	18,85
7-10-4	Mod 1 K=16	21,15	32,78	32,78	27,10	26,17	21,15	21,15	21,15	21,15
7-10-5	Mod 1 K=16	20,78	31,57	26,21	25,74	20,78	20,78	20,78	20,78	20,78
7-10-1	Mod 2 con 4 bit	17,64	72,61	72,61	72,61	72,61	0,00	0,00	0,00	0,00
7-10-2	Mod 2 con 4 bit	25,49	95,81	95,81	95,81	95,81	11,36	11,36	11,36	11,36
7-10-3	Mod 2 con 4 bit	18,85	138,03	138,03	138,03	138,03	18,85	18,85	18,85	18,85
7-10-4	Mod 2 con 4 bit	21,15	159,06	159,06	159,06	159,06	21,15	21,15	21,15	21,15
7-10-5	Mod 2 con 4 bit	20,78	87,11	87,11	87,11	87,11	20,78	20,78	20,78	20,78
7-10-1	Mod 2 con 8 bit	17,64	44,10	44,10	44,10	3,55	17,64	17,64	17,64	17,64
7-10-2	Mod 2 con 8 bit	18,31	83,82	83,82	83,82	59,63	17,34	17,02	17,02	11,36
7-10-3	Mod 2 con 8 bit	18,85	363,54	363,54	363,54	363,54	18,85	18,85	18,85	18,85
7-10-4	Mod 2 con 8 bit	21,15	212,90	212,90	212,90	212,90	21,15	21,15	21,15	21,15
7-10-5	Mod 2 con 8 bit	20,78	100,96	100,96	100,96	100,96	20,78	20,78	20,78	20,78

- Infine, anche solo guardando gli esempi e le tabelle riportati in precedenza, si può notare come i risultati ottenuti dopo il primo minuto non si differiscano molto, a volte per niente, da quelli ottenuti dopo 8 minuti. Ciò probabilmente è dovuto al fatto che: o l'istanza viene risolta molto velocemente o per riuscire a migliorare i risultati trovati all'inizio sia necessario molto tempo. Infatti se invece andiamo a guardare come evolve il gap inerente della tabella 6.1, nelle istanze più grosse, dopo 80 minuti i risultati iniziano a migliorare.

6.1 Sperimentazione del modello esatto

Per quanto riguarda il modello esatto con il vincolo quadratico abbiamo avuto dei problemi di compatibilità con alcuni componenti di JuMP, a causa dei quali non abbiamo potuto far girare il programma sulla stessa macchina sulla quale abbiamo fatto girare gli altri, ma abbiamo dovuto usarne una molto più modesta. Dunque non ci è possibile fare un reale confronto con gli altri

modelli. Inoltre non siamo riusciti a ricavare il miglior upper bound trovato quando Couenne si ferma, dunque riporteremo solo il gap primale.

In futuro vorremmo cercare di implementare questo modello in altri linguaggi per riuscire a testarlo su una macchina performante e poter avere anche gli upper bound calcolati.

Per ora i risultati sono questi

Tabella 6.7: Gap primale modello esatto con vincolo quadratico, affiancati gap primale e duale del modello 1 con K=16

Istanza \	Gap primale mod esatto %				Gap primale K=16 %				Gap duale K=16 %			
	1 min	2 min	4 min	8 min	1 min	2 min	4 min	8 min	1 min	2 min	4 min	8 min
1-5-1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-5-2	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-5-3	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-5-4	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-5-5	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-10-1	0,51	0,51	0,51	0,51	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-10-2	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-10-3	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-10-4	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-10-5	0,00	0,00	0,00	0,00	0,78	0,78	0,78	0,78	0,78	0,78	0,78	0,78
1-15-1	0,29	0,29	0,29	0,29	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-15-2	0,19	0,19	0,19	0,19	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-15-3	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-15-4	0,11	0,11	0,11	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1-15-5	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3-5-1	1,50	1,50	1,50	1,50	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3-5-2	19,01	19,01	19,01	18,87	0,39	0,39	0,39	0,39	0,39	0,39	0,39	0,39
3-5-3	27,63	26,56	23,58	23,58	0,63	0,63	0,63	0,63	0,63	0,63	0,63	0,63
3-5-4	25,51	24,83	23,50	20,40	5,49	5,49	5,49	5,49	5,49	5,49	5,49	5,49
3-5-5	21,39	19,30	19,30	19,30	7,73	7,73	7,73	7,73	7,73	7,73	7,73	7,73
3-10-1	0,03	0,03	0,03	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3-10-2	107,19	107,19	62,38	5,28	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83
3-10-3	118,68	105,94	79,29	37,67	1,48	1,48	1,48	1,48	1,48	1,48	1,48	1,48
3-10-4	82,32	82,32	82,32	37,10	5,80	5,80	5,80	5,80	5,80	5,80	5,80	5,80
3-10-5	190,33	190,33	47,38	20,92	2,18	4,57	6,35	6,35	9,49	6,62	6,01	6,01
3-15-1	0,06	0,06	0,06	0,06	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3-15-2	46,39	46,39	37,89	23,72	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85
3-15-3	163,85	163,85	71,03	26,36	3,02	3,02	2,83	2,83	13,42	6,94	2,83	2,83
3-15-4	169,59	169,59	48,29	9,71	6,08	6,08	7,09	4,59	15,80	15,80	15,80	10,48
3-15-5	200,00	200,00	19,11	17,22	23,41	23,41	23,23	25,08	21,94	21,94	21,94	21,94
5-5-1	0,11	0,11	0,11	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5-5-2	28,49	28,49	5,52	0,01	1,73	1,73	1,73	1,73	1,73	1,73	1,73	1,73
5-5-3	40,38	40,38	35,10	11,87	0,93	0,93	0,93	0,93	0,71	0,71	0,71	0,71
5-5-4	30,91	30,54	30,54	11,76	3,52	3,52	3,52	3,52	3,52	3,52	3,52	3,52
5-5-5	8,37	8,37	8,37	2,10	14,03	14,03	14,03	14,03	13,69	13,69	13,69	13,69
5-10-1	0,33	0,33	0,33	0,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5-10-2	∞	75,44	37,79	29,40	1,02	1,02	1,02	1,02	1,02	1,02	1,02	1,02
5-10-3	81,49	81,49	81,49	35,08	4,59	4,59	4,59	3,53	8,40	8,40	7,60	6,77
5-10-4	69,74	69,74	34,06	13,79	12,35	10,11	10,11	9,92	9,99	9,99	9,99	9,99
5-10-5	∞	99,84	30,48	13,07	13,77	13,77	13,77	13,77	13,77	13,77	13,77	13,77
5-15-1	∞	120,83	120,83	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5-15-2	∞	∞	82,87	8,18	2,27	1,24	1,14	0,78	8,42	6,83	4,57	3,11
5-15-3	∞	∞	74,40	53,64	17,30	11,75	11,75	10,71	14,63	14,63	14,63	13,41
5-15-4	∞	∞	67,92	29,74	11,91	11,91	11,91	11,91	14,11	14,11	14,11	14,11
5-15-5	∞	∞	38,22	26,75	25,51	21,72	13,28	13,15	12,27	12,27	12,27	12,27

Considerando appunto che la macchina su cui abbiamo testato il modello

non è molto all'avanguardia i risultati sembrerebbero incoraggianti, perlomeno quelli sugli 8 minuti. Si vedano ad esempio i valori relativi alle istanze piccole. Mentre quando la dimensione inizia a crescere il solutore impiega parecchio tempo a trovare dei buoni lower bound ed a volte non ci riesce, come accade ad esempio per le istanze più grandi.

6.2 Conclusioni

Quello che possiamo concludere è che, per istanze piccole, il modello 1 con $K=16$ è sicuramente il migliore, mentre su istanze grosse bisogna controllare che il tempo fornito sia sufficiente al modello per trovare buone soluzioni. Infatti se si è interessati ad avere subito dei buoni bounds per il problema il modello 1 con $K=8$ risulta il più veloce.

Il modello 2 ha dato risultati deludenti, non si è riusciti a trovare una versione di questo modello che fornisca anche un'approssimazione superiore abbastanza velocemente e senza compromettere troppo la qualità dei lower bound. Infatti se teniamo binaria la variabile relativa al bit più significativo, anche se non otteniamo upper bound validi, troviamo dei lower bound considerevolmente migliori

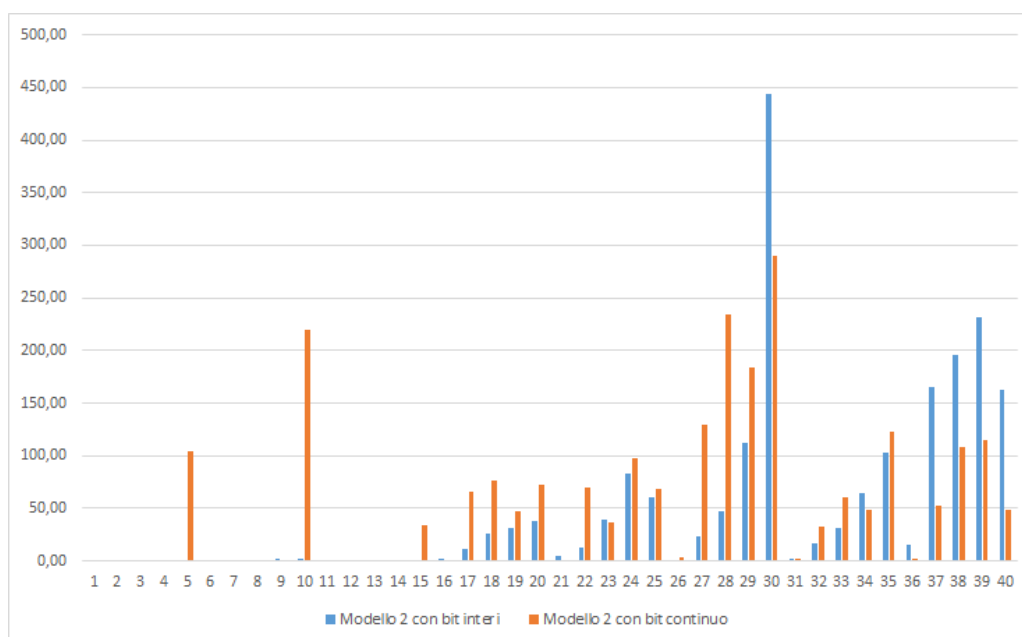


Figura 6.3: Confronto tra i gap primali

Infine, facendo una media tra i vari risultati forniti dalla sperimentazione che abbiamo fatto all'inizio per trovare i vari best bound utilizzando il modello 1 con $K=16$, siamo riusciti a tracciare un'approssimazione della curva pareto-ottimale, ossia di come cambia l'ottimo del problema al variare del budget di potenza fornito

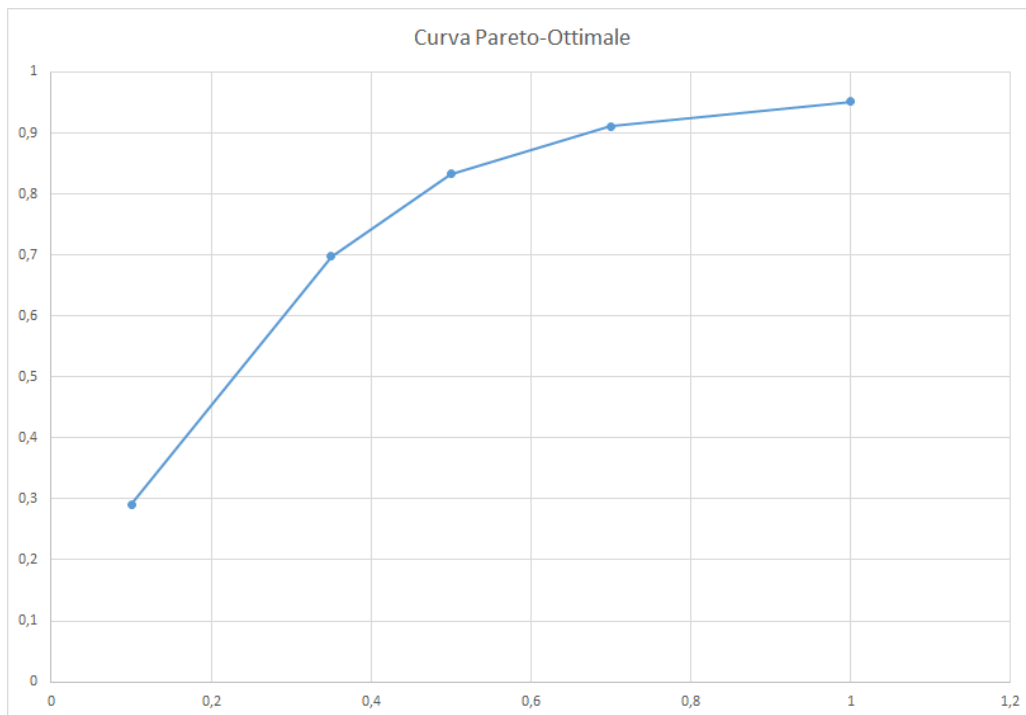


Figura 6.4: Curva pareto ottimale costruita con la media dei risultati

in generale le curve relative alle istanze più piccole tendono a crescere più linearmente, mentre le curve relative ad stanze grosse crescono più velocemente all'inizio e poi tendono a stabilizzarsi.

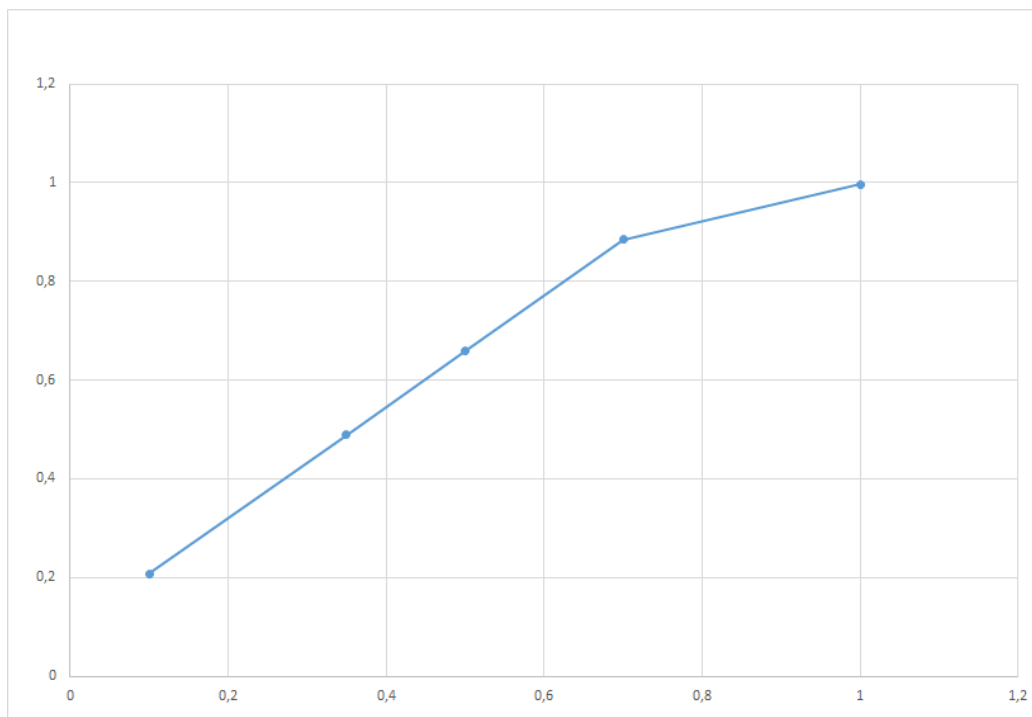


Figura 6.5: Curva relativa ad un istanza piccola (10 centroidi 3 antenne)

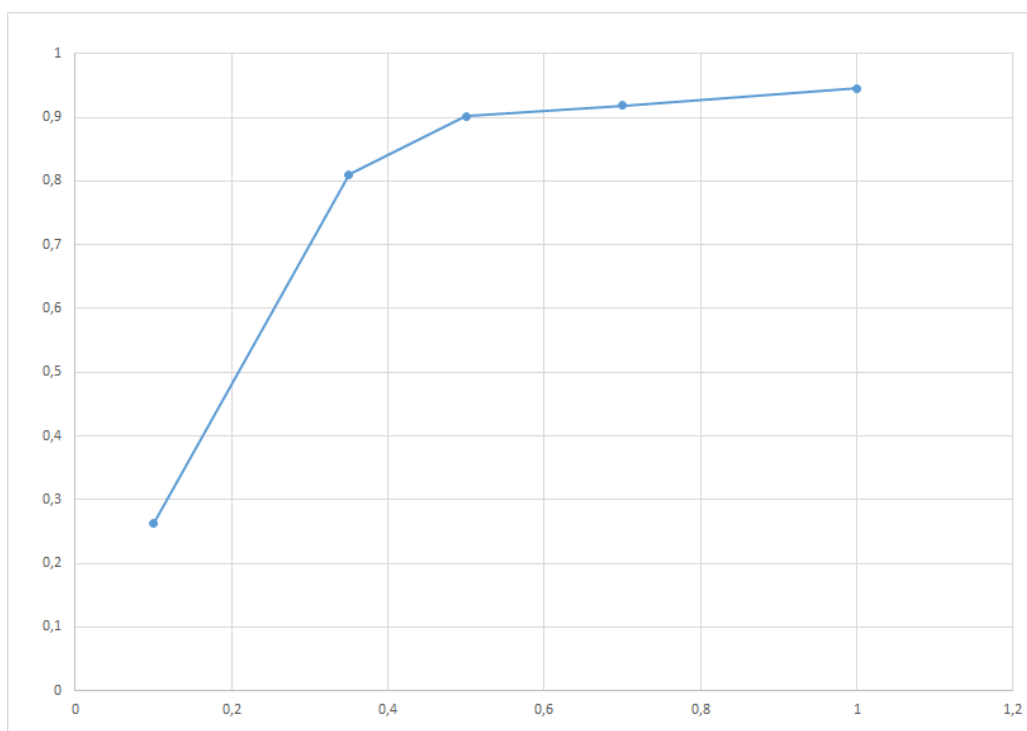


Figura 6.6: Curva relativa ad un istanza grossa (75 centroidi e 15 antenne)

Bibliografia

- [Ach09] Tobias Achterberg. SCIP: solving constraint integer programs. *Math. Program. Comput.*, 1(1):1–41, 2009.
- [AV14] D. Sabella M. Caretti A. Viridis, G. Stea. A practical framework for energy-efficient node activation in heterogeneous lte networks. *Italian workshop without proceedings*, October 2014.
- [BEL12] A. Billionnet, S. Elloumi, and A. Lambert. Extending the qcr method to general mixed-integer programs. *Mathematical Programming*, 131(1):381–401, 2012.
- [BEP09] A. Billionnet, S. Elloumi, and M.-C. Plateau. Improving the performance of standard solvers for quadratic 0–1 programs by a tight convex reformulation: the qcr method. *Discrete Applied Mathematics*, 157(6):1185–1197, 2009.
- [RRW10] Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele. Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations. *Math. Programming*, 121(2):307, 2010.

Ringraziamenti

In questa sezione spero di non aver dimenticato nessuno, in caso contrario perdonatemi.

Ringrazio prima di tutto la mia famiglia che, se pur distante, mi è sempre stata vicino. Ringrazio la mensa per i suoi manicaretti. Ringrazio Sandra e Claudio e tutti gli amici brasiliani, è bello sapere che anche dall'altra parte del mondo c'è qualcuno che ti vuole bene. Ringrazio Pisa e il suo clima stupendo. Ringrazio Ale e Dario, gli amici di una vita. Ringrazio The Beards e Mattia Cerrito per avermi tenuto compagnia nelle interminabili ore passate a programmare. Un grazie speciale va al gruppo dei "94" di casa Paolo, compresi i membri che adesso si vedono raramente e quelli che sono entrati da poco, in questi tre anni siete stati la mia famiglia a Pisa. Ringrazio Paolo, chiunque egli sia. Ringrazio l'aula studenti per essere stata un posto di divertimento e apprendimento allo stesso tempo. Ringrazio Super hexagon, CTR, Minecraft, Rocket League e tutti i vari giochi che durante questi anni mi hanno permesso di rimanere concentrato sull'università senza passare ore e ore a perdere tempo. Ringrazio il mio relatore per avermi proposto questo lavoro e per l'enorme aiuto che mi ha dato nel realizzarlo. E ovviamente ringrazio Kuzco.