

Plant Location Problem

Carlo Cabras

1 Sommario

In questo elaborato presento il lavoro svolto per il corso di Decision Science, dove studio il cosiddetto *Plant Location Problem*, un problema appartenente alla classe dei problemi NP-difficili.

Dopo aver scritto il modello matematico su CPLEX, l'ho prima testato su alcune istanze casuali create da me per poi testarlo su istanze di benchmark [3]. Le soluzioni ottimali ottenute su queste ultime istanze sono pressoché uguali a quelle riportate nei benchmark, tranne per un caso che differisce leggermente rispetto all'originale e per alcune soluzioni che nonostante avessero stesso valore ottimale, riportavano soluzioni diverse. Confronto poi i risultati con il corrispettivo rilassamento continuo, rimuovendo il vincolo di interezza delle variabili x e y , ottenendo (come ci si aspettava) tempi di risoluzione e f.o. inferiori.

Nella sez. 2 espongo il problema, nella sez. 3 descrivo come è strutturata la cartella del progetto, nella sez. 4 come ho eseguito il lavoro e nella sez. 5 espongo i risultati ottenuti.

L'intero progetto lo si può trovare nella mia repository GitHub [1].

2 Modello matematico

2.1 Plant Location Problem

Sia assegnato un insieme di n utenti di un certo servizio, e siano state individuate m possibili *locazioni* ove attivare il servizio stesso. Ciascuna locazione ha un costo fisso d_j di attivazione. Ogni utente $i \in \{1, \dots, n\}$ deve inoltre sostenere un costo c_{ij} per raggiungere la locazione attivata j a lui più vicina. Il *Problema della Localizzazione di Impianti* (*Plant Location Problem* [2]) consiste nell'individuare le locazioni in cui attivare il servizio, in modo da minimizzare la somma dei costi fissi e di collegamento. Questo problema può essere rappresentato da un grafo non orientato e bipartito $G = (V, E)$ in cui un insieme di vertici rappresenta gli utenti e l'altro insieme le possibili locazioni.

Introducendo una variabile per ogni lato $[i, j] \in E$

$$x_{ij} = \begin{cases} 1 & \text{se l'utente } i \text{ si collega alla locazione } j \\ 0 & \text{altrimenti} \end{cases}$$

ed una variabile per ogni locazione $j \in \{1, \dots, m\}$

$$y_j = \begin{cases} 1 & \text{se la locazione } j \text{ viene attivata} \\ 0 & \text{altrimenti} \end{cases}$$

si ottiene il modello di PLI:

$$\min \underbrace{\sum_{j=1}^m d_j y_j}_{\text{costo fisso}} + \underbrace{\sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}}_{\text{costo collegamento}} \quad (2.1)$$

$$\underbrace{\sum_{j=1}^m x_{ij}}_{\text{ogni utente si collega ad una località}} = 1, i \in \{1, \dots, n\} \quad (2.2)$$

$$\underbrace{x_{ij} \leq y_j}_{\text{collegamento } i-j \implies \text{loc. } j \text{ attivata}}, i \in \{1, \dots, n\}, j \in \{1, \dots, m\} \quad (2.3)$$

$$x_{ij} \text{ intero}, i \in \{1, \dots, n\}, j \in \{1, \dots, m\} \quad (2.4)$$

$$0 \leq y_j \leq 1 \text{ intero}, j \in \{1, \dots, m\} \quad (2.5)$$

I vincoli (2.3) esprimono la congruenza logica delle variabili x_{ij} e y_j .

Nel rilassamento continuo, i vincoli 2.4 e 2.5 sono uguali ma x_{ij} e y_j possono non essere interi. Questo è rappresentato su CPLEX semplicemente dichiarando le variabili come `float+` anziché `int+`.

2.2 Impostazione del lavoro

La prima parte del lavoro riguardava la rappresentazione del modello su CPLEX, in modo da avere un programma eseguibile. La sintassi del linguaggio permette una veloce ed agevole implementazione del modello. Il codice leggeva i dati da un file esterno ed inizialmente ho scritto uno script che creasse delle istanze casuali.

Sono poi passato all'esecuzione su istanze di benchmark [3] in modo da poter confrontare i miei risultati con quelli forniti, valutando correttezza e prestazioni del mio codice. Queste istanze erano in un formato completamente diverso da quello che il mio programma accetta, pertanto con uno script le ho trasformate in istanze leggibili.

Ho salvato la mia soluzione ottimale nello stesso formato di quelle fornite dal benchmark per poter essere direttamente confrontate. Successivamente ho implementato il rilassamento continuo del modello, rimuovendo i vincoli di interezza relativi alle variabili x e y , per poi confrontare questi risultati con quelli discreti ottenuti da me.

3 Struttura del progetto

3.1 Cartelle e file

La cartella del progetto è organizzata secondo la seguente struttura.

Innanzitutto, i file riguardanti la modellazione del modello su CPLEX:

- `plant_location_problem.mod`: il file principale del progetto, è quello da eseguire per far partire il programma;
- `plant_location_problem.dat`: contiene gli array relativi ai dati (utenti e locazioni) e le impostazioni per l'esecuzione del programma;
- `plant_location_problem_sub.mod`: modello matematico della versione standard discreta (cioè i vettori x e y sono interi);
- `plant_location_problem_sub_cont.mod`: modello matematico del rilassamento continuo.

Poi, gli script per rendere le istanze leggibili dal mio programma e quelli di confronto dei risultati:

- `crea_istanze_benchmark.py`: converte i file di benchmark in file adatti per il mio progetto. Legge i file e crea i diversi vettori salvandoli nel file `plant_location_problem.dat` che sarà letto dal programma CPLEX;
- `controlla_ottimali_benchmark.py`: confronta i risultati ottimali ottenuti da me con quelli forniti dal benchmark. I file ottimali contengono una lista di numeri dove dal primo al penultimo indicano a quale locazione è collegato l'i-esimo utente; l'ultimo numero contiene il valore ottimale della funzione obiettivo. Nel caso l'i-esimo utente sia collegato ad una locazione diversa, cerco nel file originale di benchmark il costo di collegamento e stampo a video se i collegamenti hanno costo uguale o diverso;
- `controlla_continui.py`: confronta i risultati continui con i risultati discreti ottenuti dal mio codice;
- `settings.py`: contiene diversi flag che consentono di impostare quali classi di benchmark debbano essere eseguite e scegliere che tipo di confronti fare.

Ci sono infine due sottocartelle: `benchmark_data`, che contiene i file di benchmark, e `risultati`, che contiene il confronto dei risultati ed i tempi di risoluzione. Le soluzioni ottimali ottenute da me sono salvate nella stessa cartella dei file di benchmark, in quanto essa contiene sia i dati delle istanze che i risultati ottimali forniti.

3.2 File di benchmark

Per poter analizzare il mio codice ho usato dei file di benchmark analizzando la mia soluzione con quella fornita. Ci sono diverse classi di istanze.

3.2.1 Bilde-Krarup (bilde)

I benchmark Bilde-Krarup sono delle vecchie e piccole istanze. La struttura dei problemi è basata sul fatto che avere costi di apertura più piccoli o più grandi comporta che sia più conveniente aprire più o meno impianti. Per la maggior parte dei problemi i costi di apertura sono fissati ad un valore per tutti gli impianti, mentre i costi di connessione sono distribuiti uniformemente in un certo intervallo. Ci sono 22 classi di problemi, esposti nella tabella 1.

Problem class	Size	Opening costs	Connection costs
B	30/80	Disc. Uniform (1000, 10000)	Disc. Uniform (0, 1000)
C	50/100	Disc. Uniform (1000, 2000)	Disc. Uniform (0, 1000)
Dq*	30/80	Identical, q*1000	Disc. Uniform (0, 1000)
Eq*	50/100	Identical, q*1000	Disc. Uniform (0, 1000)
(*) q = 1,...,10			

Tabella 1: Tabella delle istanze di Bilde-Krarup.

3.2.2 Chessboard (chess)

Queste istanze sono basate su una scacchiera dalla lunghezza di $3k$. I confini sono collegati tra loro in modo da ottenere un toro. In ogni casella c'è un cliente ed un impianto, così abbiamo $n = m = k^2$. Per le istanze nel pacchetto $k = 4$ quindi $n = m = 144$. Un impianto può servire un cliente con un costo di $\{0, 1, 2, 3, 4\}$ se il Re può raggiungere la casella del cliente dalla casella dell'impianto. Tutti gli altri costi sono posti a infinito. Nella soluzione ottimale i k^2 Re coprono l'intera scacchiera. Tuttavia il numero di insiemi di k^2 Re che coprono l'intera scacchiera cresce esponenzialmente al crescere di k .

3.2.3 Euclidean (euclid)

In quadrato di lunghezza 7000 sono stati scelti n punti. Ogni punto è un impianto ed un cliente ($n = m$). I costi di apertura sono tutti pari a 3000 per ogni locazione, i costi di connessione sono le distanze nel piano euclideo. I valori sono arrotondati all'intero e soddisfano la disuguaglianza triangolare. Per queste istanze $n = 100$.

3.2.4 Finite projective planes, k=11 (fpp11) e k=17 (fpp17)

Questi benchmark possono essere risolti in tempo polinomiale, tuttavia rappresentano una grande sfida per gli algoritmi di ricerca locale. Basati sulle matrici di incidenza per il piano proiettivo, hanno $n = m$, dove $n = k^2 + k + 1$. In queste istanze abbiamo $k = 11$ e $k = 17$. I costi di apertura sono tutti fissati a 3000. La matrice di connessione ha esattamente $n + 1$ elementi finiti dall'insieme $\{0, 1, 2, 3, 4\}$ per ogni riga ed ogni colonna.

3.2.5 Large duality gap, Class A (gapa) e Class B (gapb)

Questi benchmark sono stati generati secondo i seguenti parametri. Il numero di impianti è uguale al numero delle città con $n = m = 100$. Tutti i costi di apertura sono pari a 3000. Nella classe A ogni città ha 10 connessioni economiche con valori nell'insieme $\{0, 1, 2, 3, 4\}$. Tutti gli altri costi sono posti a 3000 che rappresentano infinito. Nella classe B ogni impianto ha 10 connessioni economiche, mentre nella classe C ci sono 10 connessioni economiche per ogni impianto ed ogni città. Le classi sono in ordine crescente di difficoltà per la programmazione matematica e gli algoritmi di branch-and-bound dalla classe A alla classe C.

Non ho studiato la classe C in quanto stava risultando troppo pesante computazionalmente.

4 Svolgimento

4.1 Passi per eseguire il lavoro

Il lavoro si compone in 4 passi:

1. impostare il lavoro su `settings.py`, scegliendo quali classi di istanze eseguire e quali confronti si vogliono effettuare tra i risultati ottenuti. Impostare `True/False` per quanto riguarda le classi dei benchmark da analizzare e che messaggi stampare a video;
2. lanciare `crea_istanze_benchmark.py` che provvederà a convertire le istanze di benchmark in istanze leggibili dal programma, impostando inoltre i vettori dei dati nel file `plant_location_problem.dat`;
3. lanciare da CPLEX il file `plant_location_problem.mod` che salverà i risultati su file di testo;
4. studiare i risultati lanciando `controlla_ottimali_benchmark.py` se vogliamo studiare i risultati discreti ottenuti da me confrontandoli con quelli forniti, o `controlla_continui.py` se vogliamo studiare i miei risultati nel caso discreto con il corrispettivo rilassamento continuo.

4.2 Specifiche macchina e S.O.

La macchina su cui è stato eseguito il lavoro è equipaggiata con un processore Intel Core i7 4510U @ 2,00 GHz e 8 GB di RAM DDR3 800 MHz. Il sistema operativo è Windows 10 Pro aggiornato all'ultima versione, mentre la versione di CPLEX è 12.7.

5 Risultati

Di tutte le 400 istanze analizzate, solo una ha ottenuto una f.o. ottimale diversa: si tratta di 30FPP_17S.txt della classe delle *Finite projective planes*, $k = 17$, dove l'originale è di 54541 mentre quella che ho calcolato io è di 54542.

Una delle prime cose che ho notato è che in alcuni casi ottenevo una soluzione diversa ma la f.o. ottimale aveva lo stesso valore: si avevano quindi diversi collegamenti utenti-locazioni. Controllando è risultato che a volte si hanno collegamenti *diversi* ma dal costo *uguale*, ed altre volte collegamenti *diversi* dal costo *diverso*. Ciò ci dice che la soluzione ottimale non è unica.

Riporto in tabelle riassuntive i risultati ottenuti, mostrando per ogni istanza il nome, il n° di utenti, il n° di locazioni, i tempi di calcolo nel caso discreto e nel rilassamento continuo, ed infine il gap percentuale tra f.o. discreta e f.o. continua, calcolato come $(f.o. \text{ discreta} - f.o. \text{ continua}) / f.o. \text{ discreta}$.

Dalle tabelle si evince come i tempi di calcolo del rilassamento continuo siano nettamente inferiori rispetto alla variante classica discreta; i singoli tempi di calcolo variano poi in base ad i tipi di test.

5.1 Bilde-Krarup

In questa classe di test il tempo medio di calcolo nel caso discreto è pari a 5.205 s¹ per un totale di 52.045 s, mentre nel rilassamento continuo si hanno 0.156 s di media per un totale di 1.563 s di esecuzione. Il gap medio è pari a 0.085012.

Tabella 2

bilde					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
B1.1	50	100	0.39 s	0.14 s	0.004244
B1.10	50	100	0.797 s	0.14 s	0.030727
B1.2	50	100	0.188 s	0.094 s	0.000000
B1.3	50	100	0.172 s	0.063 s	0.000000
B1.4	50	100	0.343 s	0.062 s	0.000101
B1.5	50	100	0.593 s	0.109 s	0.030583
B1.6	50	100	0.297 s	0.078 s	0.004283
B1.7	50	100	0.484 s	0.078 s	0.040311
B1.8	50	100	0.421 s	0.047 s	0.030354
B1.9	50	100	2.75 s	0.078 s	0.046142
C1.1	50	100	2.375 s	0.109 s	0.044348
C1.10	50	100	3.016 s	0.14 s	0.046930
C1.2	50	100	1.906 s	0.11 s	0.023714
C1.3	50	100	3.578 s	0.109 s	0.056687
C1.4	50	100	2.969 s	0.109 s	0.055241
C1.5	50	100	6.829 s	0.094 s	0.079608
C1.6	50	100	5.938 s	0.109 s	0.080700

¹Uso il punto come separatore dei decimali.

Tabella 2 – continuo della pagina precedente

bilde					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
C1.7	50	100	10.032 s	0.094 s	0.084020
C1.8	50	100	4.359 s	0.125 s	0.068185
C1.9	50	100	4.656 s	0.11 s	0.069028
D1.1	30	80	2.406 s	0.015 s	0.051354
D1.10	30	80	3.094 s	0.032 s	0.074286
D1.2	30	80	2.469 s	0.031 s	0.060127
D1.3	30	80	2.313 s	0.031 s	0.031458
D1.4	30	80	1.797 s	0.031 s	0.053927
D1.5	30	80	0.313 s	0.015 s	0.010859
D1.6	30	80	2.266 s	0.031 s	0.055643
D1.7	30	80	2.359 s	0.031 s	0.057118
D1.8	30	80	3.734 s	0.016 s	0.055311
D1.9	30	80	2.344 s	0.031 s	0.069578
D2.1	30	80	1.968 s	0.031 s	0.055255
D2.10	30	80	1.906 s	0.031 s	0.078120
D2.2	30	80	1.922 s	0.031 s	0.084759
D2.3	30	80	2.25 s	0.031 s	0.081356
D2.4	30	80	2.532 s	0.031 s	0.094422
D2.5	30	80	2.171 s	0.094 s	0.076754
D2.6	30	80	2.36 s	0.031 s	0.080400
D2.7	30	80	2.281 s	0.031 s	0.067598
D2.8	30	80	2.172 s	0.047 s	0.087026
D2.9	30	80	2.015 s	0.031 s	0.070420
D3.1	30	80	1.984 s	0.047 s	0.086052
D3.10	30	80	1.907 s	0.031 s	0.076246
D3.2	30	80	1.843 s	0.015 s	0.087789
D3.3	30	80	2.328 s	0.031 s	0.092683
D3.4	30	80	1.797 s	0.015 s	0.070289
D3.5	30	80	1.563 s	0.031 s	0.078163
D3.6	30	80	2.157 s	0.047 s	0.091798
D3.7	30	80	0.953 s	0.032 s	0.079850
D3.8	30	80	0.968 s	0.031 s	0.066701
D3.9	30	80	2.75 s	0.031 s	0.103254
D4.1	30	80	2.062 s	0.032 s	0.107330
D4.10	30	80	0.344 s	0.031 s	0.037377
D4.2	30	80	2.0 s	0.031 s	0.083613
D4.3	30	80	1.875 s	0.031 s	0.072516
D4.4	30	80	2.172 s	0.032 s	0.105700
D4.5	30	80	0.86 s	0.031 s	0.071065
D4.6	30	80	1.063 s	0.016 s	0.083699
D4.7	30	80	1.828 s	0.031 s	0.075861

Tabella 2 – continuo della pagina precedente

bilde					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
D4.8	30	80	2.531 s	0.063 s	0.093748
D4.9	30	80	1.031 s	0.031 s	0.061710
D5.1	30	80	1.282 s	0.031 s	0.069326
D5.10	30	80	0.766 s	0.032 s	0.059428
D5.2	30	80	1.578 s	0.047 s	0.072011
D5.3	30	80	2.047 s	0.031 s	0.097266
D5.4	30	80	0.765 s	0.031 s	0.075749
D5.5	30	80	2.313 s	0.047 s	0.090244
D5.6	30	80	1.328 s	0.109 s	0.072217
D5.7	30	80	1.797 s	0.031 s	0.068703
D5.8	30	80	1.672 s	0.031 s	0.081813
D5.9	30	80	2.172 s	0.078 s	0.082339
D6.1	30	80	1.265 s	0.031 s	0.080847
D6.10	30	80	1.797 s	0.063 s	0.087287
D6.2	30	80	0.938 s	0.031 s	0.088800
D6.3	30	80	0.844 s	0.031 s	0.076677
D6.4	30	80	2.437 s	0.032 s	0.086641
D6.5	30	80	1.203 s	0.047 s	0.085837
D6.6	30	80	1.719 s	0.046 s	0.083883
D6.7	30	80	1.172 s	0.032 s	0.070564
D6.8	30	80	2.672 s	0.031 s	0.105352
D6.9	30	80	1.781 s	0.031 s	0.089528
D7.1	30	80	1.031 s	0.032 s	0.083706
D7.10	30	80	0.891 s	0.031 s	0.081504
D7.2	30	80	1.406 s	0.047 s	0.085877
D7.3	30	80	0.984 s	0.031 s	0.061064
D7.4	30	80	1.157 s	0.031 s	0.086694
D7.5	30	80	0.578 s	0.047 s	0.039522
D7.6	30	80	2.031 s	0.062 s	0.092793
D7.7	30	80	0.86 s	0.063 s	0.062463
D7.8	30	80	0.844 s	0.032 s	0.029087
D7.9	30	80	1.906 s	0.032 s	0.094613
D8.1	30	80	1.156 s	0.047 s	0.061343
D8.10	30	80	0.938 s	0.031 s	0.067115
D8.2	30	80	1.297 s	0.031 s	0.070018
D8.3	30	80	0.813 s	0.031 s	0.070585
D8.4	30	80	0.922 s	0.031 s	0.050892
D8.5	30	80	0.969 s	0.032 s	0.060207
D8.6	30	80	2.156 s	0.063 s	0.086763
D8.7	30	80	0.797 s	0.032 s	0.072896
D8.8	30	80	0.75 s	0.031 s	0.057478

Tabella 2 – continuo della pagina precedente

bilde					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
D8.9	30	80	1.953 s	0.063 s	0.066038
D9.1	30	80	0.797 s	0.078 s	0.069917
D9.10	30	80	0.921 s	0.031 s	0.068966
D9.2	30	80	1.297 s	0.031 s	0.062896
D9.3	30	80	1.359 s	0.047 s	0.067949
D9.4	30	80	1.015 s	0.031 s	0.061404
D9.5	30	80	1.094 s	0.046 s	0.068238
D9.6	30	80	1.078 s	0.078 s	0.060987
D9.7	30	80	1.516 s	0.094 s	0.065197
D9.8	30	80	1.094 s	0.047 s	0.091155
D9.9	30	80	0.703 s	0.031 s	0.037118
D10.1	30	80	0.875 s	0.032 s	0.065635
D10.10	30	80	0.766 s	0.031 s	0.060077
D10.2	30	80	0.859 s	0.047 s	0.068690
D10.3	30	80	0.328 s	0.031 s	0.018209
D10.4	30	80	0.61 s	0.047 s	0.056484
D10.5	30	80	0.657 s	0.047 s	0.039675
D10.6	30	80	1.125 s	0.032 s	0.055013
D10.7	30	80	1.016 s	0.047 s	0.071633
D10.8	30	80	0.953 s	0.047 s	0.061497
D10.9	30	80	1.797 s	0.047 s	0.079668
E1.1	50	100	7.531 s	0.094 s	0.077482
E1.10	50	100	10.516 s	0.094 s	0.089233
E1.2	50	100	3.516 s	0.078 s	0.058265
E1.3	50	100	6.828 s	0.078 s	0.078874
E1.4	50	100	4.641 s	0.094 s	0.065378
E1.5	50	100	5.828 s	0.093 s	0.076527
E1.6	50	100	5.266 s	0.078 s	0.072179
E1.7	50	100	3.438 s	0.094 s	0.051150
E1.8	50	100	4.5 s	0.078 s	0.070265
E1.9	50	100	4.109 s	0.078 s	0.050797
E2.1	50	100	7.109 s	0.093 s	0.084286
E2.10	50	100	6.515 s	0.125 s	0.092248
E2.2	50	100	5.313 s	0.094 s	0.083745
E2.3	50	100	7.219 s	0.109 s	0.096184
E2.4	50	100	6.89 s	0.094 s	0.102588
E2.5	50	100	9.469 s	0.109 s	0.096835
E2.6	50	100	6.812 s	0.172 s	0.085746
E2.7	50	100	5.297 s	0.11 s	0.078281
E2.8	50	100	9.453 s	0.11 s	0.098330
E2.9	50	100	9.485 s	0.109 s	0.101384

Tabella 2 – continuo della pagina precedente

bilde					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
E3.1	50	100	7.328 s	0.14 s	0.086171
E3.10	50	100	5.907 s	0.156 s	0.093423
E3.2	50	100	5.859 s	0.11 s	0.088050
E3.3	50	100	6.516 s	0.125 s	0.092557
E3.4	50	100	7.86 s	0.157 s	0.097611
E3.5	50	100	10.047 s	0.218 s	0.098804
E3.6	50	100	11.781 s	0.171 s	0.111529
E3.7	50	100	8.25 s	0.125 s	0.111547
E3.8	50	100	6.484 s	0.125 s	0.092491
E3.9	50	100	7.125 s	0.141 s	0.096839
E4.1	50	100	8.625 s	0.141 s	0.099630
E4.10	50	100	6.109 s	0.11 s	0.091918
E4.2	50	100	4.641 s	0.141 s	0.083900
E4.3	50	100	5.422 s	0.125 s	0.092334
E4.4	50	100	6.984 s	0.141 s	0.091118
E4.5	50	100	7.406 s	0.109 s	0.099832
E4.6	50	100	5.203 s	0.172 s	0.083631
E4.7	50	100	11.125 s	0.125 s	0.117832
E4.8	50	100	7.984 s	0.125 s	0.112444
E4.9	50	100	6.969 s	0.125 s	0.103809
E5.1	50	100	5.75 s	0.14 s	0.095377
E5.10	50	100	8.906 s	0.125 s	0.116275
E5.2	50	100	5.797 s	0.172 s	0.089326
E5.3	50	100	7.078 s	0.172 s	0.091480
E5.4	50	100	4.953 s	0.109 s	0.079054
E5.5	50	100	7.063 s	0.125 s	0.104646
E5.6	50	100	6.813 s	0.11 s	0.097951
E5.7	50	100	3.641 s	0.109 s	0.066119
E5.8	50	100	5.485 s	0.141 s	0.081465
E5.9	50	100	6.297 s	0.125 s	0.091059
E6.1	50	100	6.078 s	0.109 s	0.103635
E6.10	50	100	6.797 s	0.157 s	0.104724
E6.2	50	100	5.391 s	0.125 s	0.083509
E6.3	50	100	5.547 s	0.11 s	0.080587
E6.4	50	100	7.531 s	0.141 s	0.112148
E6.5	50	100	5.079 s	0.093 s	0.087883
E6.6	50	100	7.265 s	0.141 s	0.093468
E6.7	50	100	8.5 s	0.125 s	0.102911
E6.8	50	100	5.375 s	0.125 s	0.092098
E6.9	50	100	7.203 s	0.204 s	0.104995
E7.1	50	100	6.453 s	0.125 s	0.098364

Tabella 2 – continuo della pagina precedente

bilde					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
E7.10	50	100	7.047 s	0.11 s	0.104446
E7.2	50	100	7.125 s	0.14 s	0.103320
E7.3	50	100	7.094 s	0.171 s	0.105356
E7.4	50	100	5.406 s	0.141 s	0.101411
E7.5	50	100	6.984 s	0.141 s	0.099772
E7.6	50	100	6.968 s	0.14 s	0.115751
E7.7	50	100	5.078 s	0.156 s	0.079761
E7.8	50	100	5.219 s	0.14 s	0.086912
E7.9	50	100	6.015 s	0.141 s	0.091341
E8.1	50	100	6.782 s	0.141 s	0.102315
E8.10	50	100	6.0 s	0.125 s	0.086389
E8.2	50	100	6.563 s	0.14 s	0.104348
E8.3	50	100	5.156 s	0.14 s	0.089517
E8.4	50	100	5.281 s	0.125 s	0.088837
E8.5	50	100	6.063 s	0.156 s	0.109459
E8.6	50	100	6.297 s	0.125 s	0.102614
E8.7	50	100	6.984 s	0.171 s	0.092727
E8.8	50	100	4.953 s	0.218 s	0.089694
E8.9	50	100	7.687 s	0.141 s	0.107781
E9.1	50	100	7.281 s	0.141 s	0.109536
E9.10	50	100	4.969 s	0.14 s	0.086172
E9.2	50	100	5.485 s	0.218 s	0.094515
E9.3	50	100	5.5 s	0.125 s	0.101708
E9.4	50	100	5.859 s	0.14 s	0.082945
E9.5	50	100	5.547 s	0.172 s	0.084643
E9.6	50	100	5.015 s	0.125 s	0.082121
E9.7	50	100	6.157 s	0.125 s	0.103685
E9.8	50	100	7.156 s	0.156 s	0.104199
E9.9	50	100	5.453 s	0.157 s	0.097608
E10.1	50	100	6.125 s	0.172 s	0.092593
E10.10	50	100	5.797 s	0.156 s	0.089982
E10.2	50	100	5.765 s	0.141 s	0.092542
E10.3	50	100	4.312 s	0.156 s	0.084860
E10.4	50	100	4.125 s	0.157 s	0.077574
E10.5	50	100	5.125 s	0.188 s	0.080087
E10.6	50	100	3.89 s	0.141 s	0.071132
E10.7	50	100	4.688 s	0.156 s	0.080389
E10.8	50	100	6.312 s	0.14 s	0.093440
E10.9	50	100	5.906 s	0.156 s	0.087525

Tabella 2: Tabella dei risultati ottenuti dalle istanze bilde.

5.2 Chessboard

In questa classe di test il tempo medio di calcolo nel caso discreto è pari a 0.857 s per un totale di 25.713 s, mentre nel rilassamento continuo si hanno 0.375 s di media per un totale di 11.253 s di esecuzione. Il gap medio è pari a 0.000000, in quanto la f.o. ottimale è uguale nel caso discreto e nel rilassamento continuo.

Tabella 3

chess					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
1034ChessS.txt	144	144	0.891 s	0.438 s	0.000000
1134ChessS.txt	144	144	0.875 s	0.25 s	0.000000
1234ChessS.txt	144	144	0.922 s	0.312 s	0.000000
1334ChessS.txt	144	144	0.921 s	0.438 s	0.000000
1434ChessS.txt	144	144	0.859 s	0.328 s	0.000000
1534ChessS.txt	144	144	0.937 s	0.359 s	0.000000
1634ChessS.txt	144	144	0.86 s	0.407 s	0.000000
1734ChessS.txt	144	144	0.875 s	0.375 s	0.000000
1834ChessS.txt	144	144	0.765 s	0.282 s	0.000000
1934ChessS.txt	144	144	0.922 s	0.391 s	0.000000
2034ChessS.txt	144	144	1.14 s	0.407 s	0.000000
2134ChessS.txt	144	144	0.734 s	0.359 s	0.000000
2234ChessS.txt	144	144	0.719 s	0.297 s	0.000000
2334ChessS.txt	144	144	0.875 s	0.312 s	0.000000
2434ChessS.txt	144	144	0.891 s	0.531 s	0.000000
2534ChessS.txt	144	144	0.859 s	0.359 s	0.000000
2634ChessS.txt	144	144	0.781 s	0.563 s	0.000000
2734ChessS.txt	144	144	0.671 s	0.297 s	0.000000
2834ChessS.txt	144	144	0.89 s	0.281 s	0.000000
2934ChessS.txt	144	144	0.844 s	0.422 s	0.000000
3034ChessS.txt	144	144	0.812 s	0.375 s	0.000000
3134ChessS.txt	144	144	0.687 s	0.359 s	0.000000
3234ChessS.txt	144	144	0.844 s	0.531 s	0.000000
334ChessS.txt	144	144	0.921 s	0.438 s	0.000000
434ChessS.txt	144	144	0.859 s	0.328 s	0.000000
534ChessS.txt	144	144	0.937 s	0.359 s	0.000000
634ChessS.txt	144	144	0.86 s	0.407 s	0.000000
734ChessS.txt	144	144	0.875 s	0.375 s	0.000000
834ChessS.txt	144	144	0.765 s	0.282 s	0.000000
934ChessS.txt	144	144	0.922 s	0.391 s	0.000000

Tabella 3: Tabella dei risultati ottenuti dalle istanze chess.

5.3 Euclidean

In questa classe di test il tempo medio di calcolo nel caso discreto è pari a 0.286 s per un totale di 8.594 s, mentre nel rilassamento continuo si hanno 0.045 s di media per un totale di 1.360 s di esecuzione. Il gap medio è pari a 0.000016, solo due istanze hanno riportato una f.o. ottimale leggermente inferiore nel rilassamento continuo.

Tabella 4

euclid					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
1011EuclS.txt	100	100	0.266 s	0.046 s	0.000000
1111EuclS.txt	100	100	0.266 s	0.047 s	0.000000
111EuclS.txt	100	100	0.266 s	0.047 s	0.000000
1211EuclS.txt	100	100	0.25 s	0.047 s	0.000000
1311EuclS.txt	100	100	0.25 s	0.047 s	0.000000
1411EuclS.txt	100	100	0.266 s	0.047 s	0.000000
1511EuclS.txt	100	100	0.484 s	0.047 s	0.000262
1611EuclS.txt	100	100	0.25 s	0.046 s	0.000000
1711EuclS.txt	100	100	0.25 s	0.047 s	0.000000
1811EuclS.txt	100	100	0.281 s	0.031 s	0.000000
1911EuclS.txt	100	100	0.25 s	0.047 s	0.000000
2011EuclS.txt	100	100	0.328 s	0.047 s	0.000000
2111EuclS.txt	100	100	0.234 s	0.047 s	0.000000
211EuclS.txt	100	100	0.25 s	0.047 s	0.000000
2211EuclS.txt	100	100	0.375 s	0.031 s	0.000000
2311EuclS.txt	100	100	0.406 s	0.047 s	0.000000
2411EuclS.txt	100	100	0.219 s	0.047 s	0.000000
2511EuclS.txt	100	100	0.235 s	0.047 s	0.000000
2611EuclS.txt	100	100	0.219 s	0.032 s	0.000000
2711EuclS.txt	100	100	0.359 s	0.047 s	0.000000
2811EuclS.txt	100	100	0.375 s	0.063 s	0.000000
2911EuclS.txt	100	100	0.25 s	0.047 s	0.000000
3011EuclS.txt	100	100	0.234 s	0.047 s	0.000000
311EuclS.txt	100	100	0.25 s	0.047 s	0.000000
411EuclS.txt	100	100	0.266 s	0.047 s	0.000219
511EuclS.txt	100	100	0.484 s	0.047 s	0.000000
611EuclS.txt	100	100	0.25 s	0.046 s	0.000000
711EuclS.txt	100	100	0.25 s	0.047 s	0.000000
811EuclS.txt	100	100	0.281 s	0.031 s	0.000000
911EuclS.txt	100	100	0.25 s	0.047 s	0.000000

Tabella 4: Tabella dei risultati ottenuti dalle istanze euclid.

5.4 Finite projective planes, $k = 11$

In questa classe di test il tempo medio di calcolo nel caso discreto è pari a 8.507 s per un totale di 255.223 s (6 minuti e 15.223 secondi), mentre nel rilassamento continuo si hanno 0.620 s di media per un totale di 18.614 s di esecuzione. Il gap medio è pari a 0.074776.

Tabella 5

fpp11					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
10FPP_11S.txt	133	133	8.234 s	0.641 s	0.074606
11FPP_11S.txt	133	133	8.609 s	0.61 s	0.074896
12FPP_11S.txt	133	133	8.312 s	0.61 s	0.074855
13FPP_11S.txt	133	133	8.656 s	0.609 s	0.074935
14FPP_11S.txt	133	133	8.0 s	0.625 s	0.074815
15FPP_11S.txt	133	133	8.531 s	0.594 s	0.074994
16FPP_11S.txt	133	133	8.485 s	0.625 s	0.074487
17FPP_11S.txt	133	133	8.36 s	0.625 s	0.074930
18FPP_11S.txt	133	133	8.735 s	0.61 s	0.074973
19FPP_11S.txt	133	133	8.484 s	0.625 s	0.074849
1FPP_11S.txt	133	133	8.609 s	0.61 s	0.074933
20FPP_11S.txt	133	133	9.641 s	0.61 s	0.074813
21FPP_11S.txt	133	133	8.422 s	0.625 s	0.074662
22FPP_11S.txt	133	133	8.813 s	0.609 s	0.074771
23FPP_11S.txt	133	133	8.297 s	0.625 s	0.074508
24FPP_11S.txt	133	133	8.282 s	0.625 s	0.074753
25FPP_11S.txt	133	133	7.532 s	0.625 s	0.074698
26FPP_11S.txt	133	133	8.375 s	0.672 s	0.074685
27FPP_11S.txt	133	133	9.672 s	0.641 s	0.074734
28FPP_11S.txt	133	133	8.516 s	0.609 s	0.074767
29FPP_11S.txt	133	133	8.485 s	0.625 s	0.074644
2FPP_11S.txt	133	133	8.312 s	0.61 s	0.074680
30FPP_11S.txt	133	133	8.61 s	0.641 s	0.074773
3FPP_11S.txt	133	133	8.656 s	0.609 s	0.074837
4FPP_11S.txt	133	133	8.0 s	0.625 s	0.074825
5FPP_11S.txt	133	133	8.531 s	0.594 s	0.074852
6FPP_11S.txt	133	133	8.485 s	0.625 s	0.074935
7FPP_11S.txt	133	133	8.36 s	0.625 s	0.074505
8FPP_11S.txt	133	133	8.735 s	0.61 s	0.074673
9FPP_11S.txt	133	133	8.484 s	0.625 s	0.074895

Tabella 5: Tabella dei risultati ottenuti dalle istanze fpp11.

5.5 Finite projective planes, $k = 17$

In questa classe di test il tempo medio di calcolo nel caso discreto è pari a 587.481 s (9 minuti e 47.481 secondi) per un totale di 17624.438 s (4 ore, 53 minuti e 44.438 secondi), mentre nel rilassamento continuo si hanno 5.000 s di media per un totale di 149.998 s (2 minuti e 29.998 secondi) di esecuzione. Il gap medio è pari a 0.050640. Queste sono state le istanze più pesanti.

Tabella 6

fpp17					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
10FPP_17S.txt	307	307	617.531 s	4.891 s	0.050639
11FPP_17S.txt	307	307	655.016 s	4.938 s	0.050430
12FPP_17S.txt	307	307	673.375 s	4.969 s	0.050791
13FPP_17S.txt	307	307	596.984 s	4.907 s	0.050726
14FPP_17S.txt	307	307	539.36 s	5.125 s	0.050661
15FPP_17S.txt	307	307	541.485 s	4.796 s	0.050765
16FPP_17S.txt	307	307	595.969 s	5.234 s	0.050648
17FPP_17S.txt	307	307	670.547 s	5.062 s	0.050660
18FPP_17S.txt	307	307	589.578 s	4.89 s	0.050468
19FPP_17S.txt	307	307	540.421 s	4.938 s	0.050490
1FPP_17S.txt	307	307	655.016 s	4.938 s	0.050745
20FPP_17S.txt	307	307	541.546 s	4.781 s	0.050725
21FPP_17S.txt	307	307	530.578 s	4.937 s	0.050695
22FPP_17S.txt	307	307	536.688 s	4.844 s	0.050635
23FPP_17S.txt	307	307	537.797 s	5.0 s	0.050676
24FPP_17S.txt	307	307	568.703 s	4.953 s	0.050512
25FPP_17S.txt	307	307	614.969 s	5.0 s	0.050593
26FPP_17S.txt	307	307	659.219 s	5.062 s	0.050727
27FPP_17S.txt	307	307	587.328 s	5.859 s	0.050607
28FPP_17S.txt	307	307	548.562 s	4.875 s	0.050844
29FPP_17S.txt	307	307	537.39 s	5.156 s	0.050662
2FPP_17S.txt	307	307	673.375 s	4.969 s	0.050525
30FPP_17S.txt	307	307	538.657 s	4.922 s	0.050727
3FPP_17S.txt	307	307	596.984 s	4.907 s	0.050567
4FPP_17S.txt	307	307	539.36 s	5.125 s	0.050643
5FPP_17S.txt	307	307	541.485 s	4.796 s	0.050634
6FPP_17S.txt	307	307	595.969 s	5.234 s	0.050739
7FPP_17S.txt	307	307	670.547 s	5.062 s	0.050393
8FPP_17S.txt	307	307	589.578 s	4.89 s	0.050536
9FPP_17S.txt	307	307	540.421 s	4.938 s	0.050736

Tabella 6: Tabella dei risultati ottenuti dalle istanze fpp17.

5.6 Large duality gap, Class A

In questa classe di test il tempo medio di calcolo nel caso discreto è pari a 52.653 s per un totale di 1579.596 s (26 minuti e 19.596 secondi), mentre nel rilassamento continuo si hanno 0.220 s di media per un totale di 6.613 s di esecuzione. Il gap medio è pari a 0.256299. Queste, insieme a **gapb** sono quelle che hanno ottenuto il gap maggiore, pari rispettivamente al 25% ed al 21%.

Tabella 7

gapa					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
1032GapAS.txt	100	100	34.922 s	0.266 s	0.261579
1132GapAS.txt	100	100	50.157 s	0.188 s	0.256630
1232GapAS.txt	100	100	192.609 s	0.218 s	0.296850
1332GapAS.txt	100	100	30.218 s	0.235 s	0.244936
1432GapAS.txt	100	100	37.219 s	0.172 s	0.271860
1532GapAS.txt	100	100	47.922 s	0.203 s	0.235458
1632GapAS.txt	100	100	60.281 s	0.157 s	0.263300
1732GapAS.txt	100	100	30.672 s	0.281 s	0.242673
1832GapAS.txt	100	100	48.235 s	0.265 s	0.257870
1932GapAS.txt	100	100	28.453 s	0.188 s	0.243378
2032GapAS.txt	100	100	34.625 s	0.188 s	0.256213
2132GapAS.txt	100	100	28.313 s	0.266 s	0.237007
2232GapAS.txt	100	100	29.39 s	0.25 s	0.235134
2332GapAS.txt	100	100	31.969 s	0.156 s	0.243561
2432GapAS.txt	100	100	39.391 s	0.219 s	0.273659
2532GapAS.txt	100	100	34.234 s	0.281 s	0.250917
2632GapAS.txt	100	100	45.812 s	0.188 s	0.234803
2732GapAS.txt	100	100	189.032 s	0.203 s	0.285783
2832GapAS.txt	100	100	29.453 s	0.172 s	0.252097
2932GapAS.txt	100	100	29.641 s	0.297 s	0.245022
3032GapAS.txt	100	100	51.61 s	0.266 s	0.270110
3132GapAS.txt	100	100	166.172 s	0.25 s	0.304836
3232GapAS.txt	100	100	26.266 s	0.203 s	0.258179
332GapAS.txt	100	100	30.218 s	0.235 s	0.254316
432GapAS.txt	100	100	37.219 s	0.172 s	0.247043
532GapAS.txt	100	100	47.922 s	0.203 s	0.276815
632GapAS.txt	100	100	60.281 s	0.157 s	0.243985
732GapAS.txt	100	100	30.672 s	0.281 s	0.217253
832GapAS.txt	100	100	48.235 s	0.265 s	0.262190
932GapAS.txt	100	100	28.453 s	0.188 s	0.265509

Tabella 7: Tabella dei risultati ottenuti dalle istanze gapa.

5.7 Large duality gap, Class B

In questa classe di test il tempo medio di calcolo nel caso discreto è pari a 112.241 s (1 minuto e 52 secondi) per un totale di 3367.236 s (56 minuti e 7.236 secondi), mentre nel rilassamento continuo si hanno 0.229 s di media per un totale di 6.879 s di esecuzione. Il gap medio è pari a 0.210955.

Tabella 8

gapb					
Istanza	n locazioni	n utenti	t. discreto	t. continuo	gap
1031GapBS.txt	100	100	33.953 s	0.219 s	0.168058
1131GapBS.txt	100	100	12.984 s	0.203 s	0.176471
1231GapBS.txt	100	100	169.797 s	0.188 s	0.196062
1331GapBS.txt	100	100	12.797 s	0.281 s	0.179691
1431GapBS.txt	100	100	29.359 s	0.219 s	0.185639
1531GapBS.txt	100	100	41.063 s	0.235 s	0.175926
1631GapBS.txt	100	100	118.141 s	0.219 s	0.217827
1731GapBS.txt	100	100	52.734 s	0.219 s	0.191309
1831GapBS.txt	100	100	191.844 s	0.219 s	0.226091
1931GapBS.txt	100	100	85.438 s	0.203 s	0.226522
2031GapBS.txt	100	100	66.297 s	0.25 s	0.224464
2131GapBS.txt	100	100	236.688 s	0.265 s	0.239822
2231GapBS.txt	100	100	47.328 s	0.25 s	0.196381
2331GapBS.txt	100	100	117.734 s	0.204 s	0.217207
2431GapBS.txt	100	100	198.484 s	0.25 s	0.241807
2531GapBS.txt	100	100	332.562 s	0.219 s	0.261280
2631GapBS.txt	100	100	195.469 s	0.235 s	0.237458
2731GapBS.txt	100	100	47.922 s	0.218 s	0.168773
2831GapBS.txt	100	100	42.016 s	0.204 s	0.174536
2931GapBS.txt	100	100	391.922 s	0.265 s	0.244786
3031GapBS.txt	100	100	106.985 s	0.219 s	0.208256
3131GapBS.txt	100	100	45.484 s	0.25 s	0.215414
3231GapBS.txt	100	100	258.859 s	0.25 s	0.234601
331GapBS.txt	100	100	12.797 s	0.281 s	0.241493
431GapBS.txt	100	100	29.359 s	0.219 s	0.223791
531GapBS.txt	100	100	41.063 s	0.235 s	0.191579
631GapBS.txt	100	100	118.141 s	0.219 s	0.191325
731GapBS.txt	100	100	52.734 s	0.219 s	0.238625
831GapBS.txt	100	100	191.844 s	0.219 s	0.227572
931GapBS.txt	100	100	85.438 s	0.203 s	0.205873

Tabella 8: Tabella dei risultati ottenuti dalle istanze gapb.

Riferimenti bibliografici

- [1] Carlo Cabras. Plant location problem repository: https://github.com/carlocabras21/plant_location_problem.
- [2] M. Fischetti. *Lezioni Di Ricerca Operativa*. Independently Published, 2018.
- [3] Max-Planck-Institut Informatik. <http://resources.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/packages.html>. [Online; accessed 6-July-2020].