

ALGORITMI DI ASCESA DUALE PER PROBLEMI DI FACILITY LOCATION

Alessio Di Luzio

PROBLEMI DI FACILITY LOCATION

- Un insieme di Facility U offre un servizio, ogni facility ha un costo di apertura f_u .
- I clienti dell'insieme V devono scegliere a quale facility, tra quelle aperte, connettersi; la connessione di un cliente comporta un costo c_{uv} .
- Siamo interessati a minimizzare i costi totali di erogazione del servizio.
- $y_{uv} = \begin{cases} 1 & \text{se il cliente } u \text{ è connesso alla facility } v \\ 0 & \text{altrimenti} \end{cases}$
- $x_u = \begin{cases} 1 & \text{se la facility } u \text{ è aperta} \\ 0 & \text{altrimenti} \end{cases}$

UNCAPACITATED FACILITY LOCATION

Formulazione intera

- $\min (\sum_{u \in U} \sum_{v \in V} c_{uv} y_{uv} + \sum_{u \in U} f_u y_u)$
- s.t $\sum_{u \in U} x_{uv} = 1 \quad \forall v \in V$
- $y_{uv} \leq x_u \quad \forall u \in U, v \in V$
- $y_{uv} \in \{0,1\} \quad \forall u \in U, v \in V$
- $x_u \in \{0,1\} \quad \forall u \in U$

- E' un problema NP-Hard
- E' essenziale calcolare un Lower Bound di qualità per eseguire in modo efficiente algoritmi di Branch and Bound

Formulazione rilassata

- $\min (\sum_{u \in U} \sum_{v \in V} c_{uv} y_{uv} + \sum_{u \in U} f_u x_u)$
- s.t $\sum_{u \in U} y_{uv} \geq 1 \quad \forall v \in V$
- $y_{uv} \leq x_u \quad \forall u \in U, v \in V$
- $(x_u \leq 1 \quad \forall u \in U)$
- $y_{uv} \geq 0, x_u \geq 0 \quad \forall u \in U, v \in V$

- Ci consente sicuramente di calcolare un Lower Bound per il problema intero
- Ma nemmeno in questo caso è possibile farlo in modo efficiente

UNCAPACITATED FACILITY LOCATION

Duale del rilassamento lineare

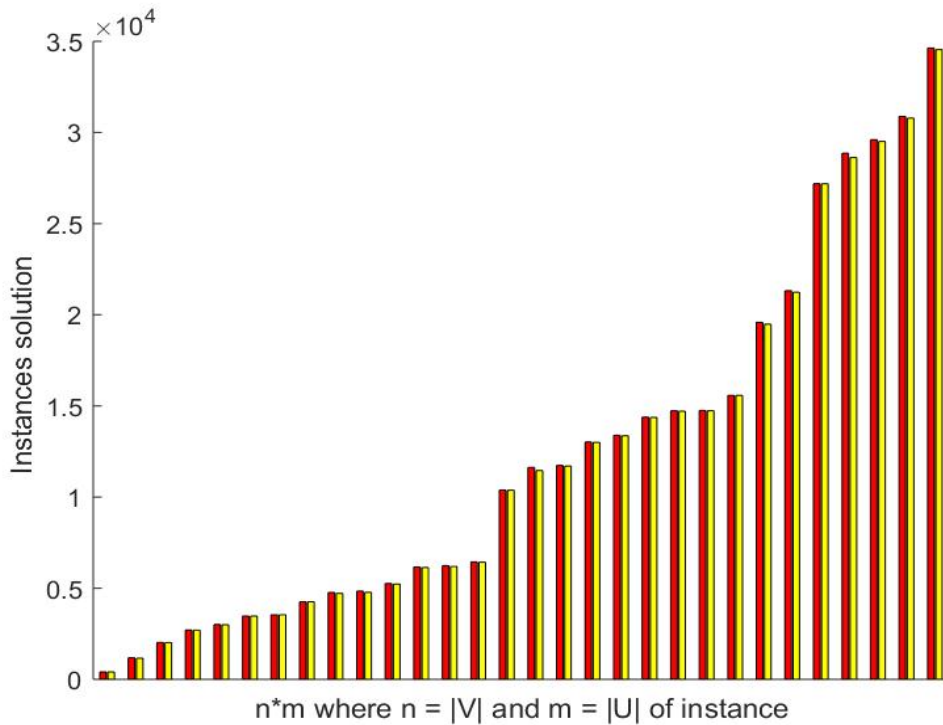
- $\max \quad (\sum_{v \in V} z_v - \sum_{u \in U} t_u)$
 - $\text{s.t.} \quad z_v - w_{uv} \leq c_u \quad \forall u \in U, v \in V$
 - $\sum_{v \in V} w_{uv} - t_u \leq f_u \quad \forall u \in U$
 - $z_v \geq 0, t_u \geq 0, w_{uv} \geq 0 \quad \forall u \in U, v \in V$
-
- Il Lower Bound sul problema Duale è altrettanto difficile da calcolare rispetto al corrispondente rilassamento lineare
 - Tuttavia la struttura del Duale ci consente di definire algoritmi euristici efficienti che individuano un 'buon' valore della funzione obiettivo

Algoritmi di Ascesa Duale considerati

- DUALOC 'semplice' e versione di Erlenkotter [1]
- Primale – Duale (Gupta) [2]

UNCAPACITATED FACILITY LOCATION

Le istanze di Test



I test sono stati eseguiti su 30 istanze per UFL costruite sfruttando alcune proprietà della libreria *random* di python .

Nel grafico a lato in Rosso la soluzione ottima per la formulazione intera e in Giallo la soluzione *rilassata*.

UNCAPACITATED FACILITY LOCATION

DUALOC 'semplice' [1]

```
1  Inizializzazione:  $\bar{z}_v = \min_{u \in U} \{c_{uv}\} \forall v \in V$   
2  Ripeti  $\forall \bar{v} \in V$   
3       $\tau_{\bar{v}u} = f_u - \sum_{v \in V - \{\bar{v}\}} \max\{0, \bar{z}_v - c_{uv}\};$   
4       $b_{\bar{v}} = \min_{u \in U} \{c_{u\bar{v}} + \tau_{\bar{v}u} - \bar{z}_{\bar{v}}\};$   
5       $\bar{z}_{\bar{v}} := \bar{z}_{\bar{v}} + b_{\bar{v}};$   
6  end;
```

- Calcola un buon Lower Bound
- Tuttavia è evidente la dipendenza della soluzione dall'ordinamento dell'insieme dei clienti

UNCAPACITATED FACILITY LOCATION

DUALOC di Erlenkotter [1]

```

1  Inizializzazione:  $\bar{z}_v = \min_{u \in U} \{c_{uv}\} \forall v \in V, i := 1;$ 
2  Passo ( i )
3   $\bar{V} = V;$ 
4  Ripeti
5       $\bar{v} = \operatorname{argmin}_{v \in \bar{V}} \{h(v)\};$ 
6       $\bar{V} := \bar{V} - \{\bar{v}\};$ 
7       $\tau_{\bar{v}u} = f_u - \sum_{v \in V - \{\bar{v}\}} \max\{0, \bar{z}_v - c_{uv}\};$ 
8       $b_{\bar{v}} = \min_{u \in U} \{c_{u\bar{v}} + \tau_{\bar{v}u} - \bar{z}_{\bar{v}}\};$ 
9       $b'_{\bar{v}} = \min \left\{ b_{\bar{v}}, \min_{\substack{u \in U \\ c_{u\bar{v}} - \bar{z}_{\bar{v}} > 0}} \{c_{u\bar{v}} - \bar{z}_{\bar{v}}\} \right\};$ 
10 fino a che  $b'_{\bar{v}} > 0$  oppure  $\bar{V} = \emptyset$ 
11 Se  $b'_{\bar{v}} = 0 \Rightarrow STOP.$ 
12 Altrimenti
13      $\bar{z}_{\bar{v}} := \bar{z}_{\bar{v}} + b'_{\bar{v}};$ 
14      $i := i + 1;$ 
15 Vai al Passo ( i );

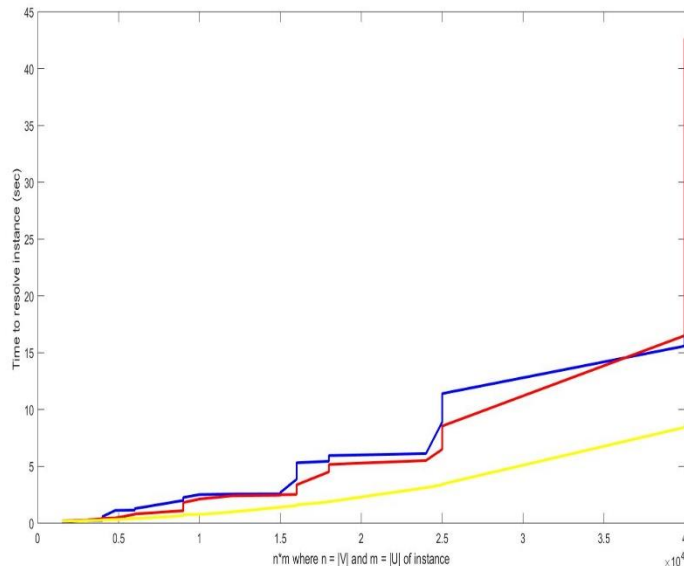
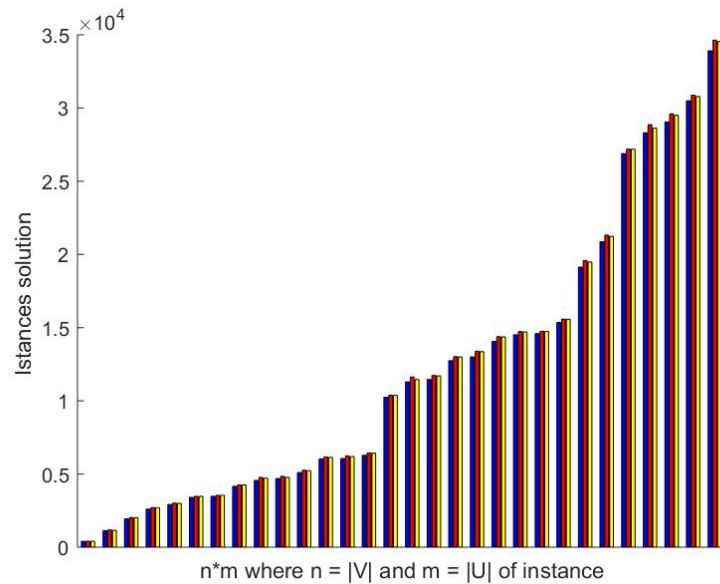
```

- Rispetto alla versione semplificata riduce la dipendenza dall'ordinamento dell'insieme dei clienti

$$h(v) = |\{u: z_{uv} \geq c_{uv}\}|$$
- 'Addolcisce' l'incremento calcolato in ogni iterazione aumentando la precisione della soluzione
- Uno dei metodi più efficaci per calcolare LB per UFL

UNCAPACITATED FACILITY LOCATION

DUALOC vs AMPL & Cplex

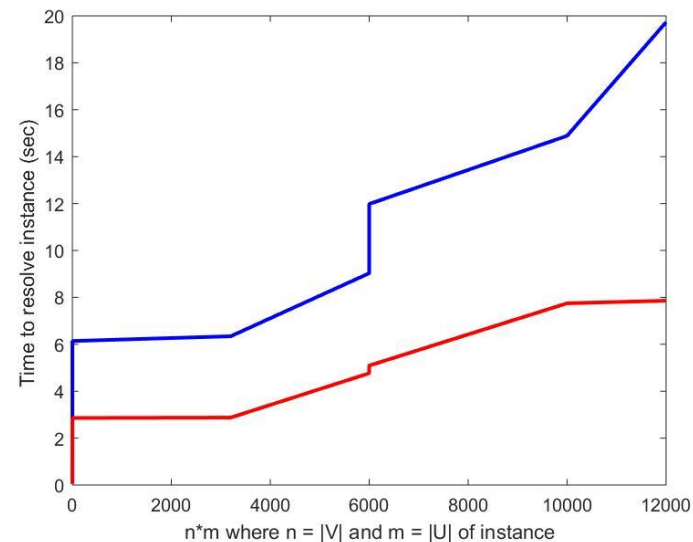
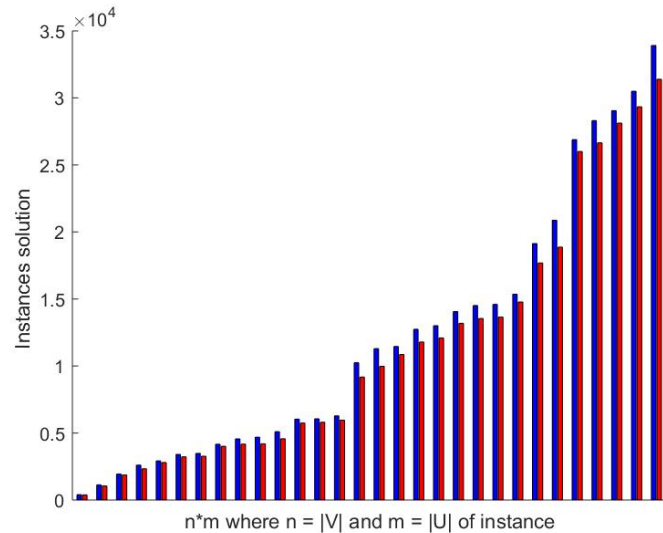


- Il Lower Bound calcolato sulle istanze di test conferma l'ipotesi di efficienza e bontà del DUALOC
- Nel grafico in alto in rosso la soluzione ottima calcolata con il solver Cplex e in blu quella ottenuta a seguito dell'esecuzione dell'algoritmo DUALOC
- Tuttavia per istanze di dimensioni grandi le prestazioni del DUALOC sembrerebbero nettamente peggiori di quelle di AMPL (grafico in basso)

UNCAPACITATED FACILITY LOCATION

DUALOC di Erlenkotter vs DUALOC 'semplice'

- La soluzione calcolata dalla versione 'semplificata' del DUALOC è effettivamente peggiore rimanendo pure sempre di ottima qualità come si vede nel primo grafico in cui la linea BLU è il valore della soluzione calcolato dalla versione di Erlenkotter mentre quella rossa è il valore della soluzione calcolato dalla versione 'semplice'.
- Tuttavia il tempo di esecuzione della versione semplificata è nettamente inferiore infatti l'algoritmo di Erlenkotter ha un'esecuzione volutamente 'rallentata' per favorire il calcolo di un Bound preciso



UNCAPACITATED FACILITY LOCATION

DUALOC vs Algoritmo Primale-Duale [2]

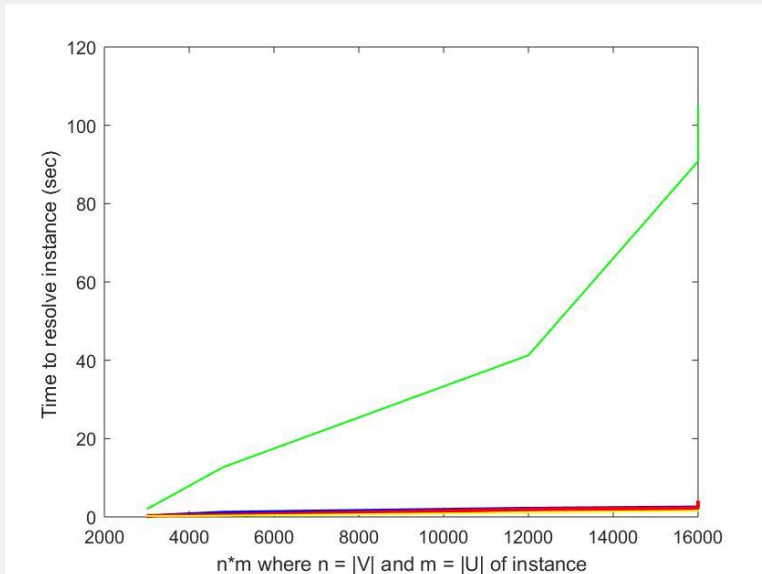
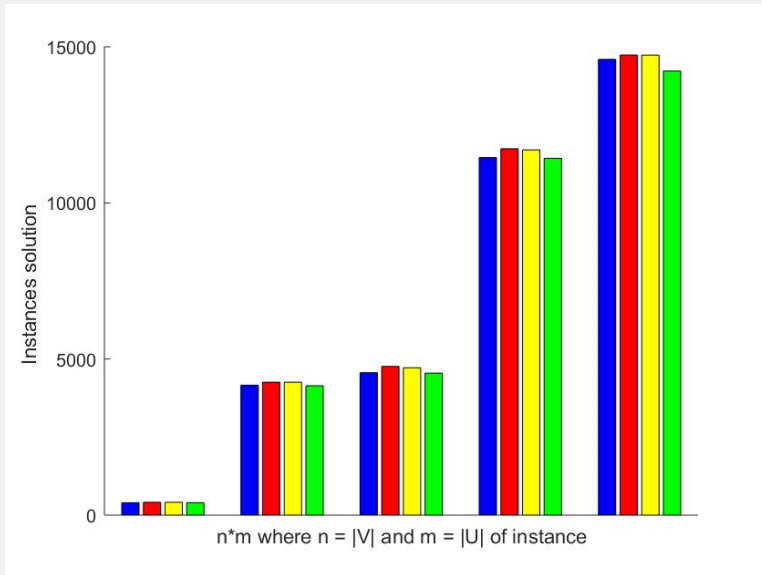
L'algoritmo primale duale in ogni iterazione incrementa tutte le variabili duali, traendo conclusioni rispetto al Primale.

Infatti durante l'esecuzione si blocca la crescita delle variabili legate a clienti assegnati a facility definite temporaneamente aperte

Ad ogni iterazione quindi il valore di incremento Δ viene calcolato tenendo conto di tutti i vincoli duali (esclusi quelli relativi a clienti già assegnati e/o facility già aperte).

Il risultato è un notevole rallentamento dell'esecuzione dell'algoritmo, come testimoniato dal grafico in alto in cui la linea Verde rappresenta il tempo di esecuzione dell'algoritmo Primale-Duale mentre quella blu il tempo di esecuzione del DUALOC (Le linee Rossa e Gialla sono per rispettivamente il solver AMPL sul problema intero e sul problema rilassato).

Tutta via nel grafico in basso si mostra la notevole precisione dell'algoritmo Primale-Duale che per istanze metriche garantisce una soluzione 3-approssimata.



UNCAPACITATED FACILITY LOCATION

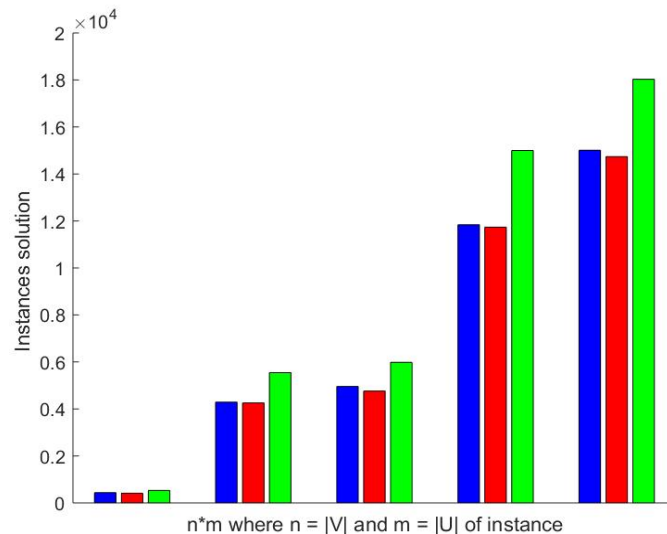
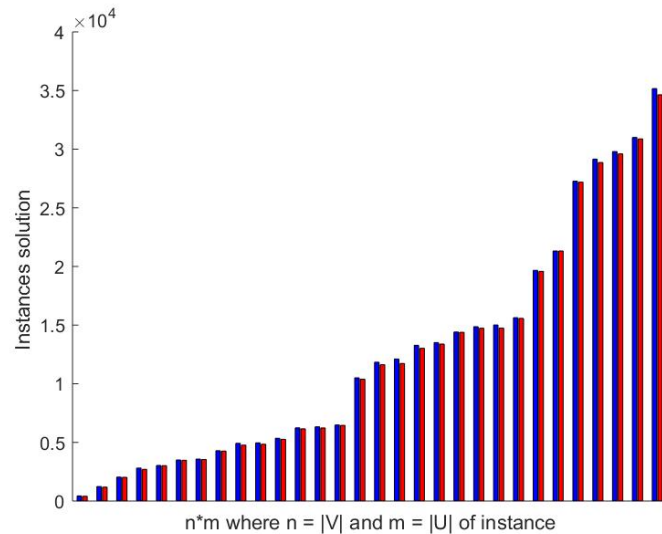
Soluzione primale

Sia il DUALOC che l'algoritmo Primale Duale forniscono dei semplici metodi per ricavare una soluzione primale ammissibile a partire da quella duale durante l'esecuzione degli algoritmi.

Anche per quanto riguarda la soluzione primale il DUALOC si comporta meglio dell'algoritmo Primale – Duale.

Nel grafico in basso sono rappresentate in Rosso la soluzione ottima, in blu quella (primale) calcolata da DUALOC e in verde quella derivata dall'esecuzione dell'algoritmo Primale – Duale su istanze metriche.

Nel grafico in alto vediamo invece un confronto tra la soluzione ottima del problema (Rosso) e quella fornita da DUALOC su istanze UFL non metriche.



SINGLE SOURCE CAPACITATED FACILITY LOCATION

Estensione dell'UFL in cui ogni cliente ha una domanda d_v da soddisfare e ogni facility una capacità K_u .

Formulazione intera

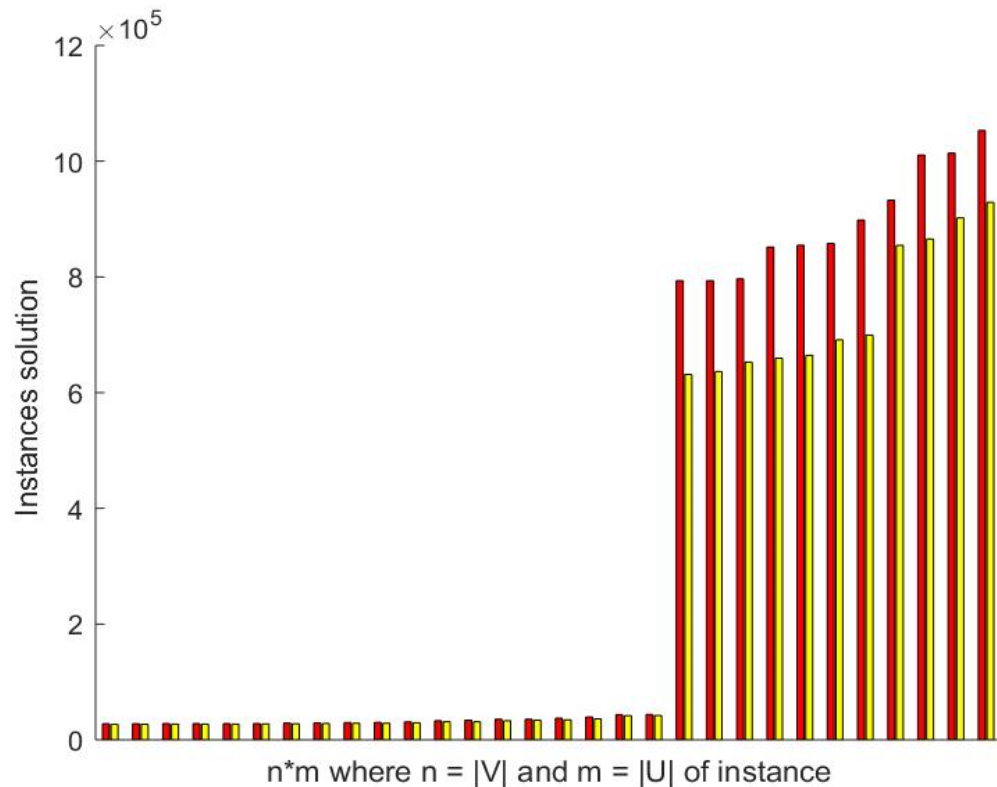
$$\begin{aligned} \min \quad & (\sum_{u \in U} \sum_{v \in V} c_{uv} y_{uv} + \sum_{u \in U} f_u x_u) \\ \text{s.t.} \quad & \sum_{u \in U} y_{uv} = 1 \quad \forall v \in V \\ & \sum_{v \in V} d_v y_{uv} \leq K_u x_u \quad \forall u \in U, v \in V \\ & y_{uv} \in \{0,1\} \quad \forall u \in U, v \in V \\ & x_u \in \{0,1\} \quad \forall u \in U \end{aligned}$$

Duale del rilassamento lineare

$$\begin{aligned} \max \quad & (\sum_{v \in V} z_v - \sum_{u \in U} t_u) \\ \text{s.t.} \quad & z_v - w_u d_v \leq c_{uv} \quad \forall u \in U, v \in V \\ & q_u w_u - t_u \leq f_u \quad \forall u \in U \\ & z_v \geq 0, t_u \geq 0, w_u \geq 0 \quad \forall u \in U, v \in V \end{aligned}$$

- Possiamo seguire il ragionamento che ha portato al DUALOC e cercare di adattare l'algoritmo per fornirci nuovamente un buon LB

SINGLE SOURCE CAPACITATED FACILITY LOCATION



Le istanze di Test

I test sono stati eseguiti su 30 istanze per SSCFL della libreria *Yang* [3]. Nel grafico al lato in Rosso la soluzione ottima del problema per la formulazione intera e in Giallo la soluzione *rilassata*.

SINGLE SOURCE CAPACITATED FACILITY LOCATION

```
1  Inizializzazione:  $\bar{z}_v = \min_{u \in U} \{c_{uv}\} \forall v \in V$ 
2   $\bar{V} = V$ ;
3  Ripeti
4       $\bar{v} = \operatorname{argmin}_{v \in \bar{V}} \{h(v)\}$ ;
5       $\bar{V} := \bar{V} - \{\bar{v}\}$ ;
6       $b_{\bar{v}} = \min_{u \in U} \left\{ \frac{q_u}{d_v} * f_u - z_v + c_{uv} \right\}$ ;
7       $\bar{z}_{\bar{v}} := \bar{z}_{\bar{v}} + b_{\bar{v}}$ ;
8  fino a che  $\bar{V} = \emptyset$ 
```

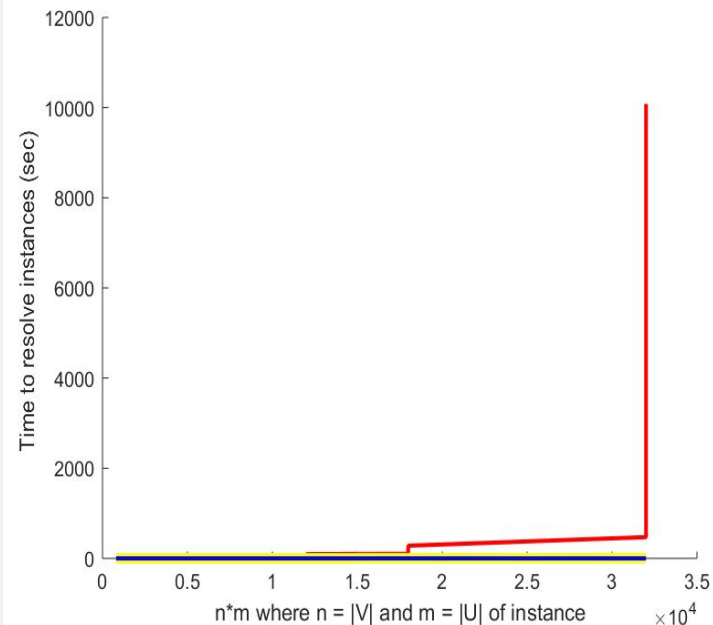
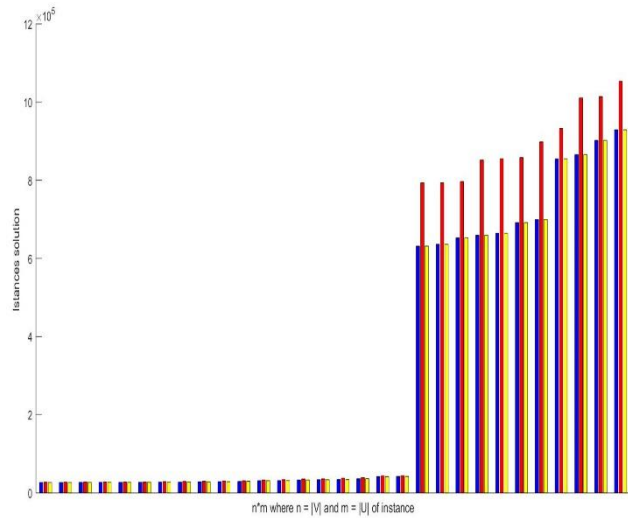
Sono stati apportati due cambiamenti principali

- Il valore di blocco è calcolato in modo differente (linea 6) per venire incontro alla nuova formulazione dei vincoli duali.
- A differenza del DUALOC di Erlenkotter si è scelto di incrementare ad ogni iterazione la variabile corrente del massimo Δ onde evitare un eccessivo rallentamento dell'algoritmo.

SINGLE SOURCE CAPACITATED FACILITY LOCATION

DUALOC adattato a SSCFL vs AMPL & Cplex

- Come è evidente dal grafico in alto , la soluzione calcolata dal DUALOC adattato (linea BLU) è sensibilmente peggiore di quella del solver AMPL sul problema intero (linea ROSSA) ma del tutto aderente a quella del problema rilassato (linea GIALLA)
- Le prestazioni del DUALOC adattato sono significativamente migliori di quelle del solver AMPL sul problema intero (grafico inferiore)



RIFERIMENTI

- [1] A. Sassano, Modelli e algoritmi della ricerca operativa, FrancoAngeli, 1999, Milano, pp.267 – 308.
- [2] A. Gupta, V. Gupta , “Lecture 5: Primal-Dual Algorithms and Facility Location”, 2008, pp. 1 - 7.
- [3] http://or-brescia.unibs.it/instances/instances_sscflp