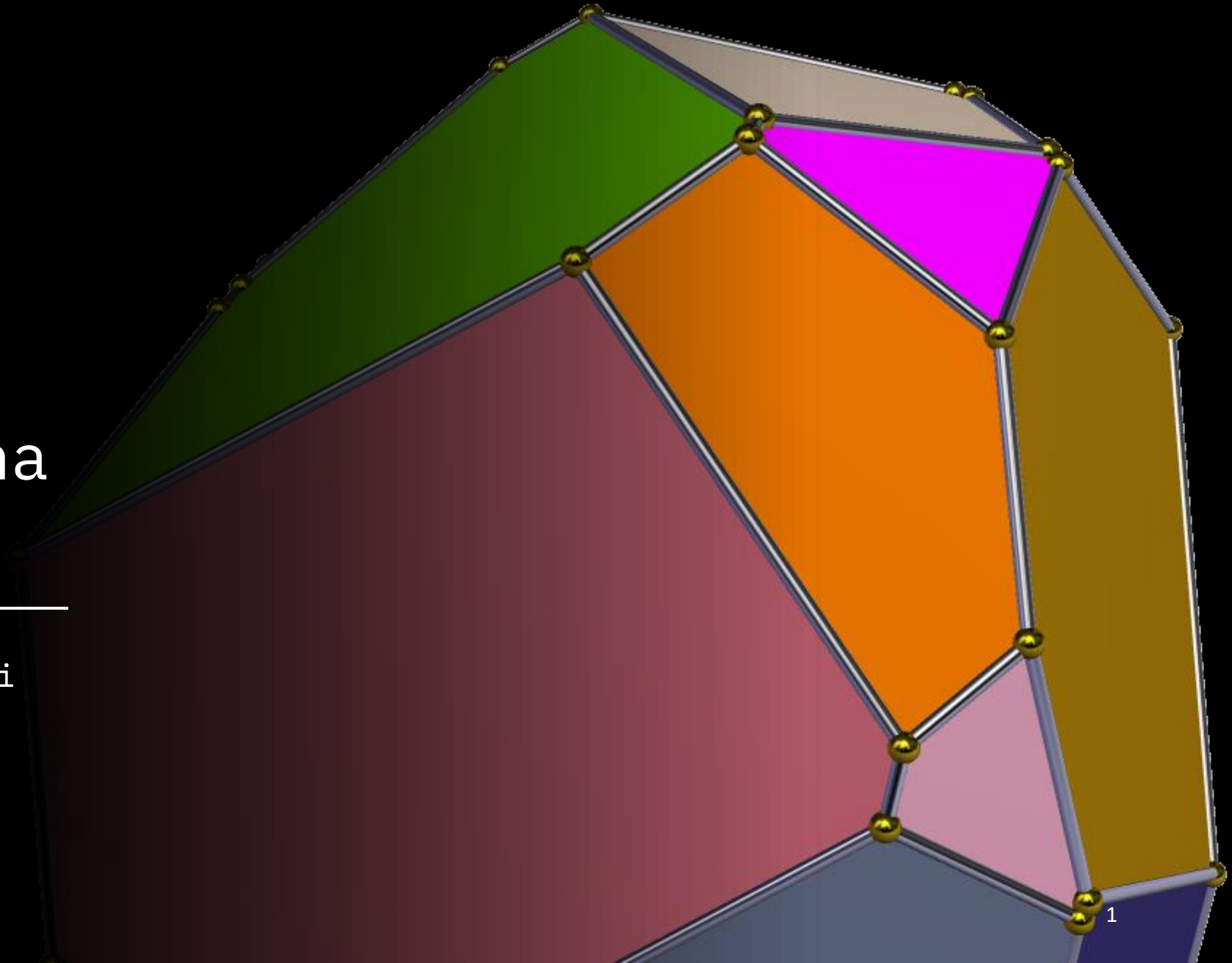
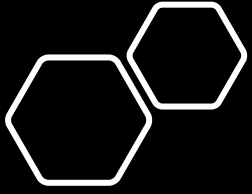


Applicazione dei tagli di Gomory ad un problema di Knapsack multidimensiona le binario

Presentazione del progetto di
Algoritmi e Modelli per
l'Ottimizzazione Discreta

Salvati Martina (0292307)
Ferretti Matteo (0300049)





Specifiche e scelte progettuali

Lo scopo è stato quello di implementare uno schema risolutivo del Multidimensional Knapsack Problem (MKP), con l'obiettivo di trovare un sottoinsieme di elementi che massimizzi un profitto (funzione obiettivo), rispettando una serie di vincoli.

Abbiamo scelto di utilizzare il solver CPLEX di IBM; in particolare l'API per Python.



Passi logici

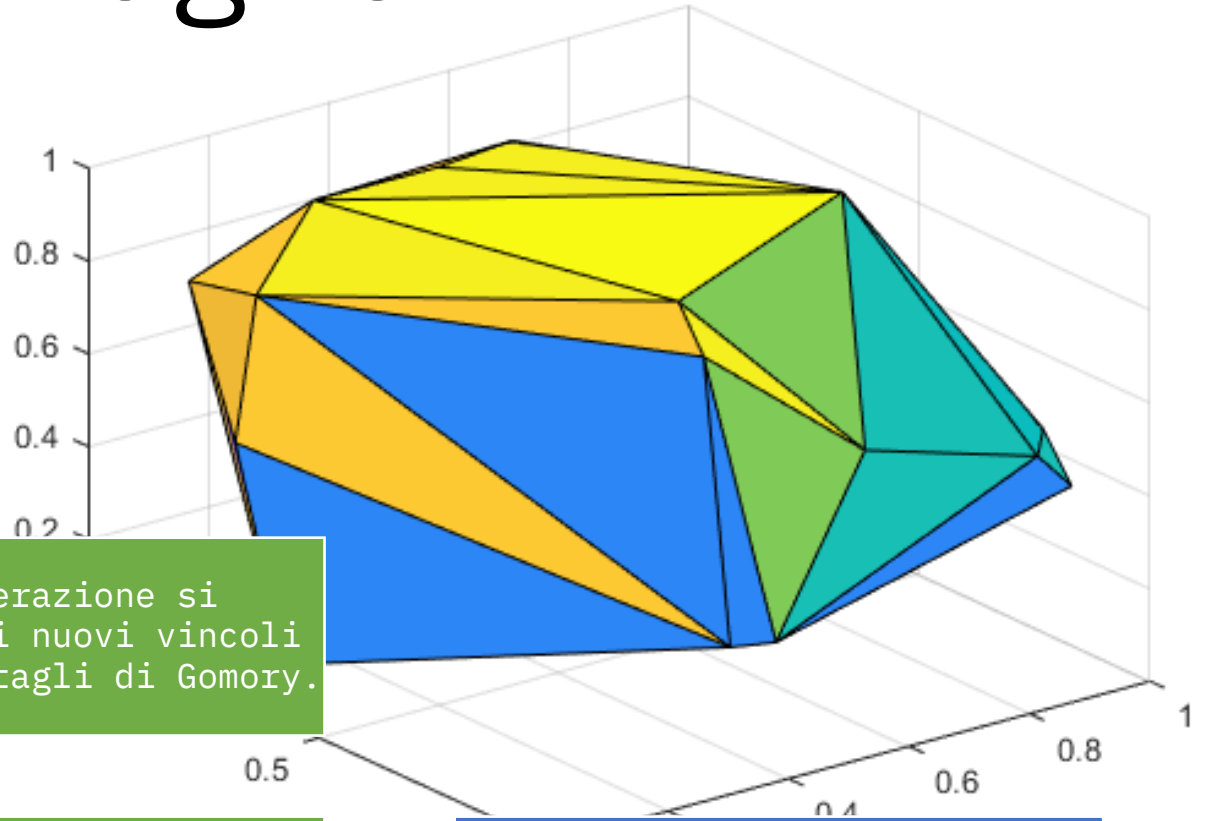
Risoluzione dell'istanza del problema come PLI ed ottenimento della soluzione ottima.

Risoluzione iterativa del problema di PL rilassato.

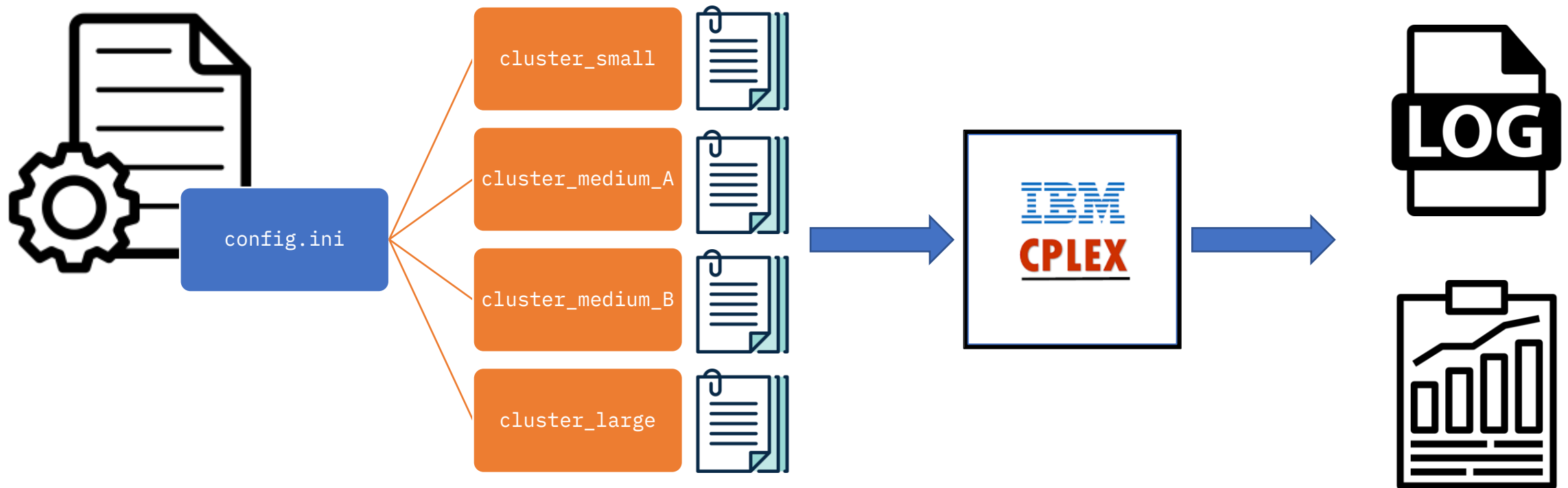
Ad ogni iterazione si introducono dei nuovi vincoli attraverso dei tagli di Gomory.

La procedura viene iterata per un tempo limite o fino a quando l'introduzione di nuovi vincoli non porta a miglioramenti.

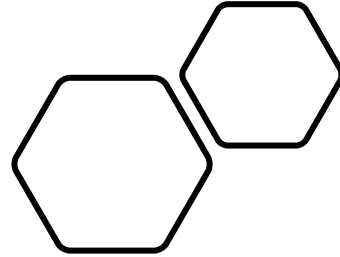
L'utilità dei tagli è valutata iterativamente confrontando la soluzione del rilassamento con la soluzione ottima.



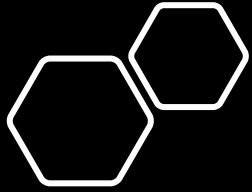
Progettazione



CPLEX Python API



- L'API CPLEX Python consiste in un strato sottile on top della C Callable Library
- Permette facilmente di:
 - Modificare e interrogare i dati del modello
 - Possibilità di utilizzare le Advanced API
 - Recuperare facilmente la soluzione dei modelli
 - Gestione dei messaggi di log
- [CPLEX for Python users - IBM Documentation](#)



Algoritmo dei tagli di Gomory

procedure GOMORY:

begin

rimuovi il vincolo di interezza da $ILP \Rightarrow LP$;

call TWO_PHASE per LP e sia Y il tableau finale;

if $infeasible = \text{false}$ and $unbounded = \text{false}$ then

begin

$feasible := \text{true}$;

$k := 0$ (contatore dei tagli);

while $\exists y_{i0}$ frazionario and $feasible = \text{true}$ do

begin

scegli un $i : y_{i0}$ è frazionario;

$k := k + 1$;

aggiungi al tableau l'equazione

$$- \sum_{A_j \notin B} f_{ij} x_j + s_k = -f_{i0};$$

call DUAL_SIMPLEX;

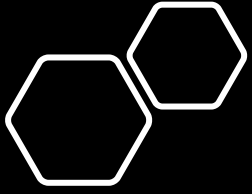
if $infeasible = \text{true}$ (duale LP illimitato) then

$feasible := \text{false}$ (ILP impossibile)

end

end

end.



Generazione delle istanze

- A partire dal file di configurazione, è possibile impostare i parametri di generazione dei gruppi di istanze (clusters).
- La pseudo-randomicità è gestita tramite un seed basato sulle cifre del timestamp al momento dell'esecuzione.

```
[cluster_small]
MIN_N_VAR = 2
MAX_N_VAR = 10
MIN_COCONSTRAINTS = 2
MAX_COCONSTRAINTS = 10
MIN_COEFF_VAL = 1
MAX_COEFF_VAL = 20
NUM_INSTANCES = 20
MAX_TIME_PER_INSTANCE = 20000

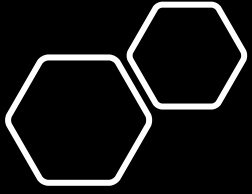
[cluster_medium_B]
MIN_N_VAR = 80
MAX_N_VAR = 100
MIN_COCONSTRAINTS = 2
MAX_COCONSTRAINTS = 10
MIN_COEFF_VAL = 1
MAX_COEFF_VAL = 20
NUM_INSTANCES = 20
MAX_TIME_PER_INSTANCE = 60000

[cluster_medium_A]
MIN_N_VAR = 2
MAX_N_VAR = 10
MIN_COCONSTRAINTS = 80
MAX_COCONSTRAINTS = 100
MIN_COEFF_VAL = 1
MAX_COEFF_VAL = 20
NUM_INSTANCES = 20
MAX_TIME_PER_INSTANCE = 60000

[cluster_large]
MIN_N_VAR = 80
MAX_N_VAR = 100
MIN_COCONSTRAINTS = 80
MAX_COCONSTRAINTS = 100
MIN_COEFF_VAL = 1
MAX_COEFF_VAL = 20
NUM_INSTANCES = 20
MAX_TIME_PER_INSTANCE = 120000
```

config.ini : impostazioni predefinite

```
def getSeed():
    time = str(datetime.now())
    return time[len(time)-6:]
```



Formato di un'istanza

Le istanze di esempio che abbiamo utilizzato sono:

- **cluster_small**: pochi vincoli, poche variabili
- **cluster_medium_A**: pochi vincoli, tante variabili
- **cluster_medium_B**: tanti vincoli, poche variabili
- **cluster_large**: tanti vincoli, tante variabili

```
3 10|
3 15 10
8 5 12
4 14 13
3 16 2
6 9 3
19 9 13
5 2 10
1 18 6
15 5 14
1 8 1
16 13 8
13 23 40 37 35 39 10 35 26 25
```

Esempio di istanza cluster small


```

05-Nov-22 11:41:44 - Solving problem instance 'inst_16.txt';
05-Nov-22 11:41:44 - This instance has 4 variables and 9 constraints
05-Nov-22 11:41:44 - This instance has 4 variables and 9 constraints
05-Nov-22 11:41:44 -
    *** OPTIMAL PLI SOLUTION ***
05-Nov-22 11:41:44 - -> Solution status = MIP_optimal
05-Nov-22 11:41:44 - -> Solution value = 12.000000

05-Nov-22 11:41:44 - SLACKS SITUATION:
05-Nov-22 11:41:44 - -> Row 0: Slack = 28.0
05-Nov-22 11:41:44 - -> Row 1: Slack = 72.0
05-Nov-22 11:41:44 - -> Row 2: Slack = 11.0
05-Nov-22 11:41:44 - -> Row 3: Slack = 67.0
05-Nov-22 11:41:44 - -> Row 4: Slack = 34.0
05-Nov-22 11:41:44 - -> Row 5: Slack = 41.0
05-Nov-22 11:41:44 - -> Row 6: Slack = 2.0
05-Nov-22 11:41:44 - -> Row 7: Slack = 61.0
05-Nov-22 11:41:44 - -> Row 8: Slack = 37.0
05-Nov-22 11:41:44 -
    PROBLEM VARIABLES:
05-Nov-22 11:41:44 - -> Column 0 (variable x0): Value = -0.0
05-Nov-22 11:41:44 - -> Column 1 (variable x1): Value = -0.0
05-Nov-22 11:41:44 - -> Column 2 (variable x2): Value = 1.0
05-Nov-22 11:41:44 - -> Column 3 (variable x3): Value = -0.0
05-Nov-22 11:41:44 -
    *** RELAXED PL SOLUTION (UPPER BOUND) ***
05-Nov-22 11:41:44 - -> Solution value = 13.789474

```

```

PROBLEM VARIABLES:
- -> Column 0 (variable x0): Value = 0.105
- -> Column 1 (variable x1): Value = 0.0
- -> Column 2 (variable x2): Value = 1.0
- -> Column 3 (variable x3): Value = 0.0
- -> Column 4 (variable s0): Value = 0.0
- -> Column 5 (variable s1): Value = 0.0
- -> Column 6 (variable s2): Value = 0.0
- -> Column 7 (variable s3): Value = 0.0
- -> Column 8 (variable s4): Value = 0.0
- -> Column 9 (variable s5): Value = 0.0
- -> Column 10 (variable s6): Value = 0.0
- -> Column 11 (variable s7): Value = 0.0
- -> Column 12 (variable s8): Value = 0.0
-
Resolution of the problem called 'inst_16': 4 Gomory cuts applied.
-
- -> Solution value = 13.214286

```

```

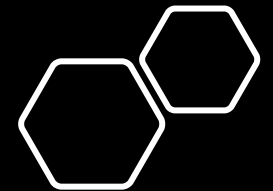
Cuts to generate: 9
Generating Gomory cuts...

Row 1 gives cut -> +14/19 x1 +5/19 x2 +18/19 x3
Row 2 gives cut -> +3/19 x1 +16/19 x2 +12/19 x3
Row 3 gives cut -> +2/19 x1 +17/19 x2 +8/19 x3 +
Row 4 gives cut -> +15/19 x1 +4/19 x2 +3/19 x3 +
Row 5 gives cut -> +5/19 x1 +14/19 x2 +1/19 x3 +
Row 6 gives cut -> +12/19 x1 +7/19 x2 +10/19 x3
Row 7 gives cut -> +16/19 x1 +3/19 x2 +7/19 x3 +
Row 8 gives cut -> +5/19 x1 +14/19 x2 +1/19 x3 +
Row 9 gives cut -> +12/19 x1 +7/19 x2 +10/19 x3

```

Logging (1)

Per tenere traccia del susseguirsi delle soluzioni e dei passi risolutivi, abbiamo implementato la possibilità di registrare gli eventi in un log.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CPLEXSolution version="1.2">
<header
  problemName="inst_0_cut_n7"
  objectiveValue="3.4999999999999978"
  solutionTypeValue="1"
  solutionTypeString="basic"
  solutionStatusValue="1"
  solutionStatusString="optimal"
  solutionMethodString="primal"
  primalFeasible="1"
  dualFeasible="1"
  simplexIterations="0"
  writeLevel="1"/>
<quality
  epRHS="9.999999999999995e-07"
  epOpt="9.999999999999995e-07"
  maxPrimalInfeas="0"
  maxDualInfeas="0"
  maxPrimalResidual="2.9976021664879227e-15"
  maxDualResidual="0"
  maxX="0.49999999999999967"
  maxPi="3.4999999999999947"
  maxSlack="53.5"
  maxRedCost="12.499999999999954"
  kappa="15.324324324324323"/>
<linearConstraints>
<constraint name="c0" index="0" status="BS" slack="3.0000000000000009" dual="0"
<constraint name="c1" index="1" status="BS" slack="9.0000000000000053" dual="0"
<constraint name="c2" index="2" status="BS" slack="36.5" dual="-0"/>
<constraint name="c3" index="3" status="BS" slack="24.5" dual="-0"/>
<constraint name="c4" index="4" status="BS" slack="42.5" dual="-0"/>
<constraint name="c5" index="5" status="BS" slack="2.0000000000000036" dual="0"
<constraint name="c6" index="6" status="BS" slack="53.5" dual="-0"/>
<constraint name="c7" index="7" status="BS" slack="4.0000000000000071" dual="0"
<constraint name="c8" index="8" status="BS" slack="14.500000000000005" dual="0"
<constraint name="c9" index="9" status="BS" slack="4.0000000000000009" dual="0"
<constraint name="c10" index="10" status="BS" slack="2.5000000000000009" dual="0"
<constraint name="c11" index="11" status="BS" slack="1.0000000000000009" dual="0"
<constraint name="c12" index="12" status="BS" slack="4.5000000000000053" dual="0"
<constraint name="c13" index="13" status="BS" slack="5.5000000000000071" dual="0"
<constraint name="c14" index="14" status="BS" slack="3.5000000000000062" dual="0"
<constraint name="c15" index="15" status="BS" slack="2.0000000000000027" dual="0"
<constraint name="c16" index="16" status="BS" slack="2.5000000000000071" dual="0"
<constraint name="c17" index="17" status="BS" slack="4.0000000000000071" dual="0"
<constraint name="c18" index="18" status="BS" slack="2.500000000000004" dual="0"
<constraint name="c19" index="19" status="BS" slack="4.0000000000000071" dual="0"
<constraint name="cut_1" index="20" status="BS" slack="2.0000000000000013" dual="0"
</linearConstraints>
</CPLEXSolution>
```

File xml

```
> iteration0
> iteration1
v iteration2
  ≡ 0_cut.log
  ≡ 1_cut.log
  ≡ 2_cut.log
  ≡ 3_cut.log
  ≡ 4_cut.log
  ≡ 5_cut.log
  ≡ 6_cut.log
  ≡ 7_cut.log
  ≡ 8_cut.log
  ≡ 9_cut.log
  ≡ 10_cut.log
```

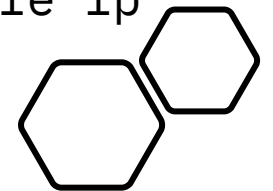
```
v cluster_small
  v inst_0
    > iteration0
      v iteration1
        ≡ 1_cut.lp
        ≡ 2_cut.lp
        ≡ 3_cut.lp
        ≡ 4_cut.lp
        ≡ 5_cut.lp
        ≡ 6_cut.lp
        ≡ 7_cut.lp
        ≡ 8_cut.lp
        ≡ 9_cut.lp
```

```
\Problem name: inst_0_cut_n3
Maximize
obj1: 7 x0 + 5 x1 + 16 x2
Subject To
c0: 2 x0 + 20 x1 + 17 x2 + s0 <= 4
c1: 16 x0 + 14 x1 + 9 x2 + s1 <= 17
c2: 3 x0 + 20 x1 + 10 x2 + s2 <= 38
c3: x0 + 4 x1 + 6 x2 + s3 <= 25
c4: 5 x0 + 12 x1 + x2 + s4 <= 45
c5: 12 x0 + 11 x1 + 3 x2 + s5 <= 8
c6: 9 x0 + 8 x1 + 5 x2 + s6 <= 58
c7: 20 x0 + 19 x1 + 17 x2 + s7 <= 14
c8: 15 x0 + 9 x1 + x2 + s8 <= 22
c9: 2 x0 + 16 x1 + 15 x2 + s9 <= 5
c10: 3 x0 + 19 x1 + 17 x2 + s10 <= 4
c11: 4 x0 + 9 x1 + 8 x2 + s11 <= 3
c12: 19 x0 + 27 x1 + 24 x2 + s12 <= 14
c13: 21 x0 + 31 x1 + 28 x2 + s13 <= 16
c14: 15 x0 + 18 x1 + 16 x2 + s14 <= 11
c15: 8 x0 + 16 x1 + 14 x2 + s15 <= 6
c16: 11 x0 + 13 x1 + 12 x2 + s16 <= 8
c17: 20 x0 + 20 x1 + 18 x2 + s17 <= 14
c18: 5 x0 + 18 x1 + 16 x2 + s18 <= 5
c19: 20 x0 + 21 x1 + 19 x2 + s19 <= 14
cut_1: 4 x0 + 19 x1 + 17 x2 <= 4
cut_2: 2 x0 + 4.999999999999999 x1 + 5 x2 <= 1
cut_3: 4 x0 + 16 x1 + 15 x2 <= 4
Bounds
0 <= x0 <= 1
0 <= x1 <= 1
```

File lp

Logging (2)

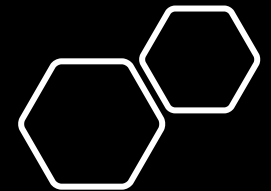
- Sono stati salvati ulteriori file :
 - ❑ File lp per ogni taglio, iterazione e istanza
 - ❑ File log per ogni taglio, iterazione e istanza
 - ❑ File xml di Cplex per ogni taglio, iterazione e istanza



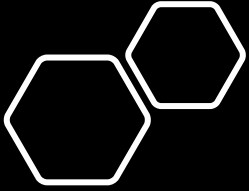
name ▼	cluster_type ▼	nvar ▼	nconstraints ▼	optimal_sol ▼	sol ▼	sol_is_integer ▼	ncuts ▼	elapsed_time ▼	gap ▼	relative_gap ▼	iterations ▼
inst_16	cluster_small	4	9	12	13,78947368	FALSO	0	1	1,78947368	0,149122807	0
inst_16	cluster_small	4	18	12	13,78947368	FALSO	1	0	1,78947368	0,149122807	1
inst_16	cluster_small	4	18	12	13,78947368	FALSO	2	2	1,78947368	0,149122807	1
inst_16	cluster_small	4	18	12	13,78947368	FALSO	3	4	1,78947368	0,149122807	1
inst_16	cluster_small	4	18	12	13,21428571	FALSO	4	6	1,21428571	0,101190476	1
inst_16	cluster_small	4	18	12	13,21428571	FALSO	5	7	1,21428571	0,101190476	1
inst_16	cluster_small	4	18	12	13,13333333	FALSO	6	8	1,13333333	0,094444444	1
inst_16	cluster_small	4	18	12	12,91666667	FALSO	7	10	0,91666667	0,076388889	1
inst_16	cluster_small	4	18	12	12,91666667	FALSO	8	12	0,91666667	0,076388889	1
inst_16	cluster_small	4	18	12	12,91666667	FALSO	9	13	0,91666667	0,076388889	1
inst_16	cluster_small	4	27	12	12,91666667	FALSO	17	2	0,91666667	0,076388889	2
inst_16	cluster_small	4	27	12	12,91666667	FALSO	18	3	0,91666667	0,076388889	2
inst_16	cluster_small	4	27	12	12,91666667	FALSO	19	5	0,91666667	0,076388889	2
inst_16	cluster_small	4	27	12	12,91666667	FALSO	20	7	0,91666667	0,076388889	2
inst_16	cluster_small	4	27	12	12	VERO	21	9	0	0	2

Statistiche

- I dati delle iterazioni sono stati acquisiti in un foglio di calcolo.
- Il foglio di calcolo è stato utilizzato per estrapolarne dei grafici, eseguiti tramite la libreria in Python matplotlib.



Risultati



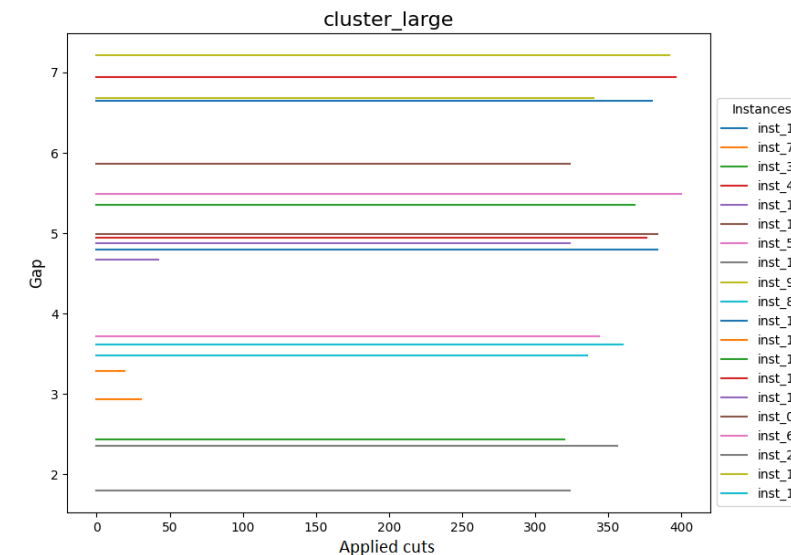
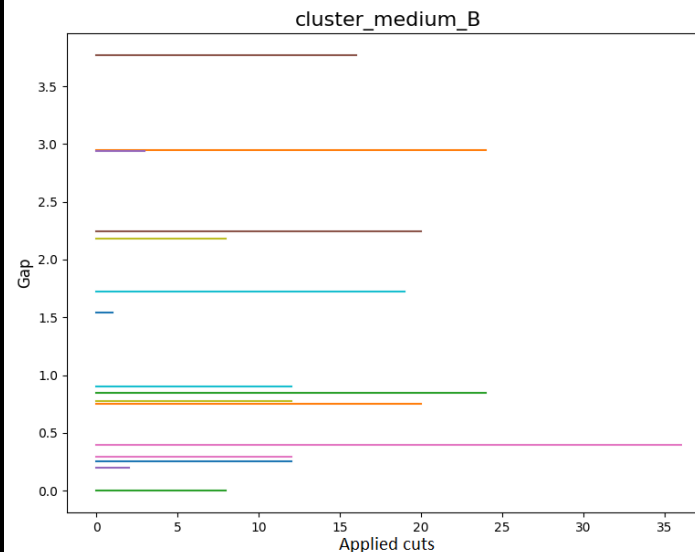
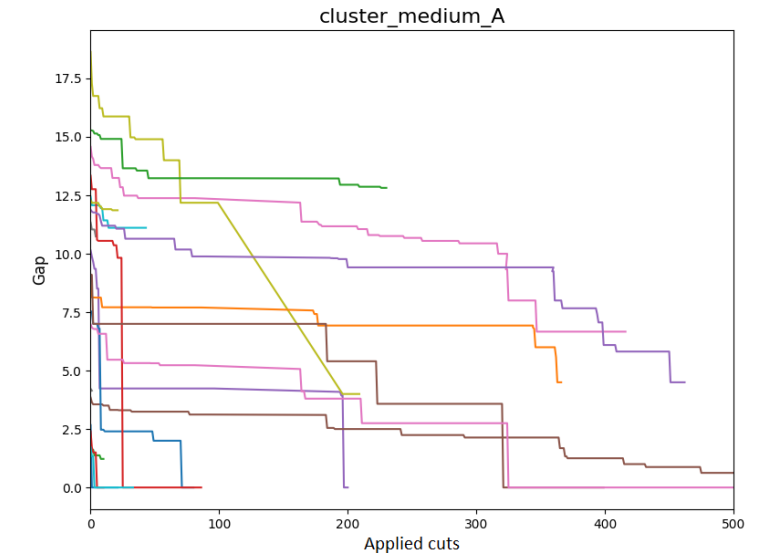
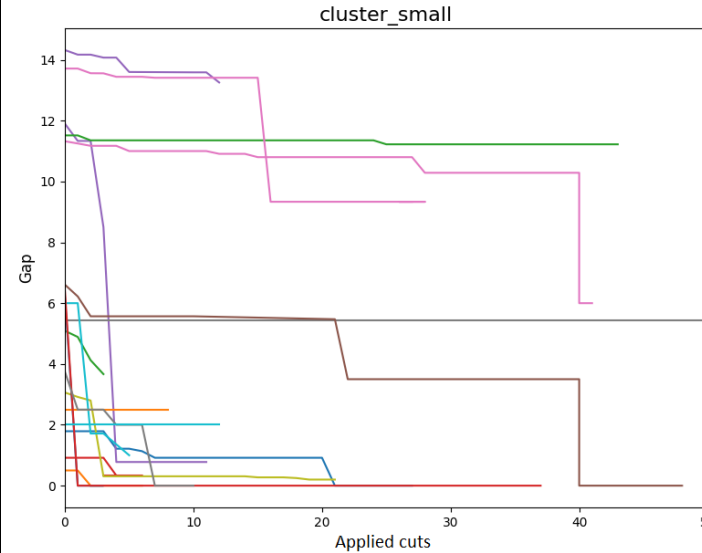
Metriche di valutazione dell'algoritmo

- Variabili :
 - ❖ **bestbound** : valore della soluzione del problema rilassato
 - ❖ **bestinteger** : valore della soluzione ottima
- RELATIVE GAP formula
 - ❖ $\frac{|\text{bestbound} - \text{bestinteger}|}{1e-10 + |\text{bestinteger}|}$
- GAP formula :
 - ❖ $|\text{bestbound} - \text{bestinteger}|$
- Queste metriche vengono utilizzate per valutare il divario relativo tra la soluzione ottima intera e la soluzione del problema rilassato.
- Un relative gap ≤ 0.05 risulta un buon compromesso per la nostra computazione.

Variazione del gap rispetto ai tagli

- Si può notare come il cluster_small e il cluster medium A si comportino meglio sotto l'esecuzione dell'algoritmo, diminuendo iterativamente il gap.
- Cluster small e cluster medium A hanno in comune il fatto di avere **un numero di vincoli limitato**.
- Cluster medium B e cluster large, con un numero elevato di vincoli, non rispondono bene all'algoritmo, non variando il gap.

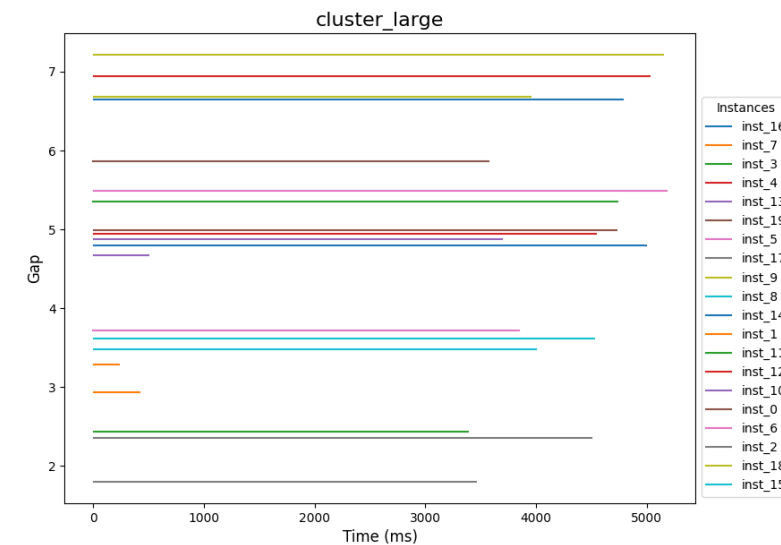
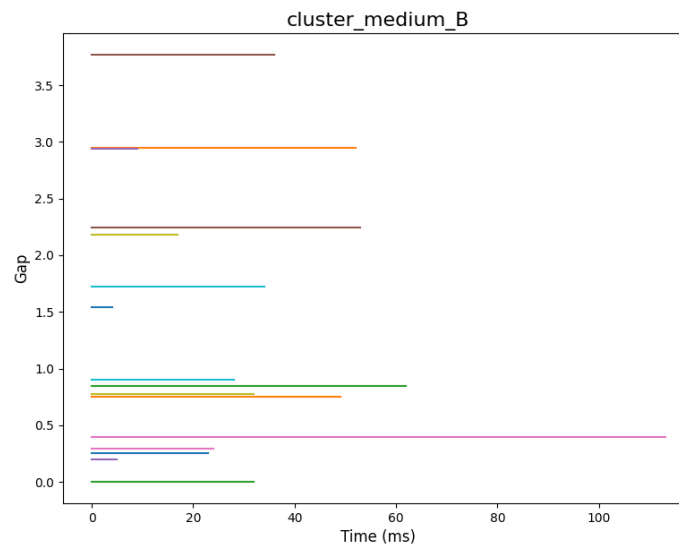
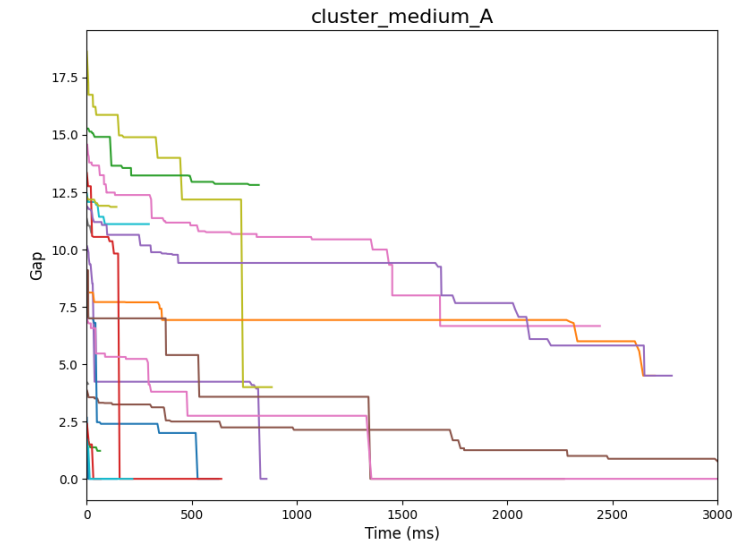
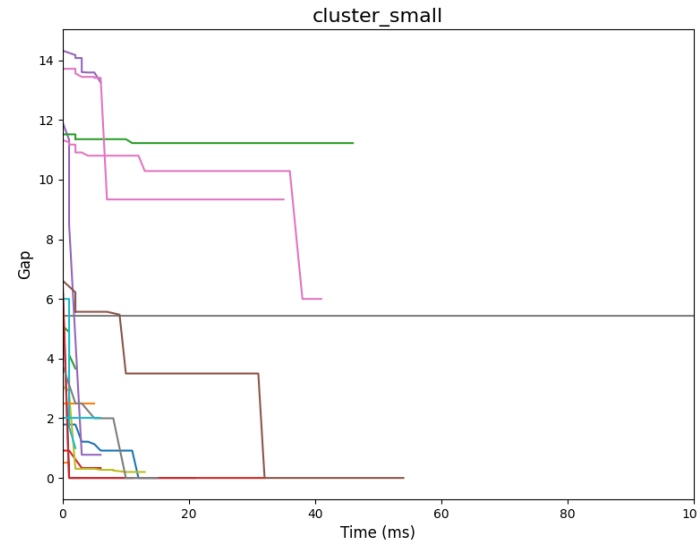
Gap variations over cuts

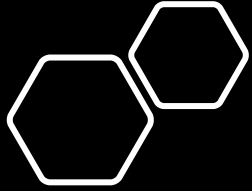


Variazione del relative gap rispetto al tempo

- Questo grafico mette in luce come al passare del tempo la variazione del gap è evidente per cluster small e cluster medium A

Gap variations over time

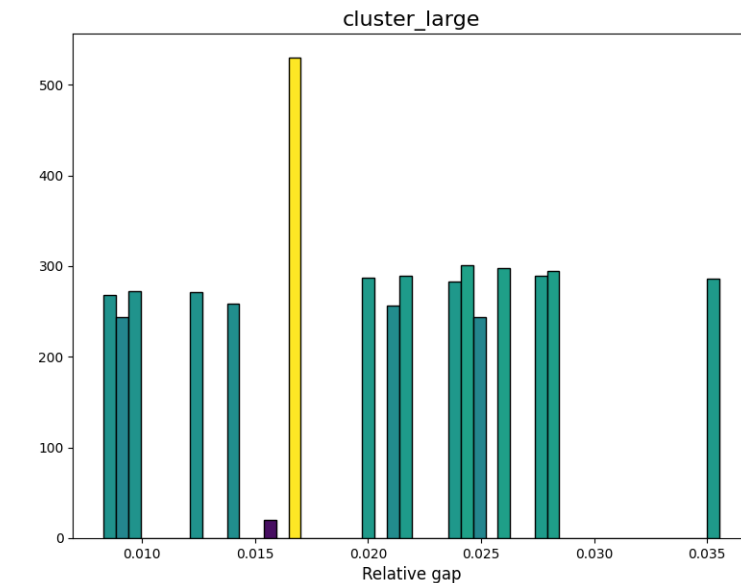
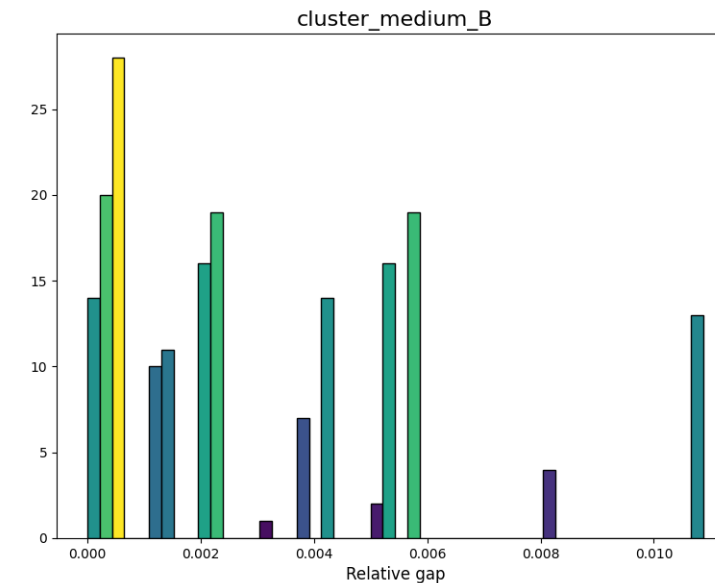
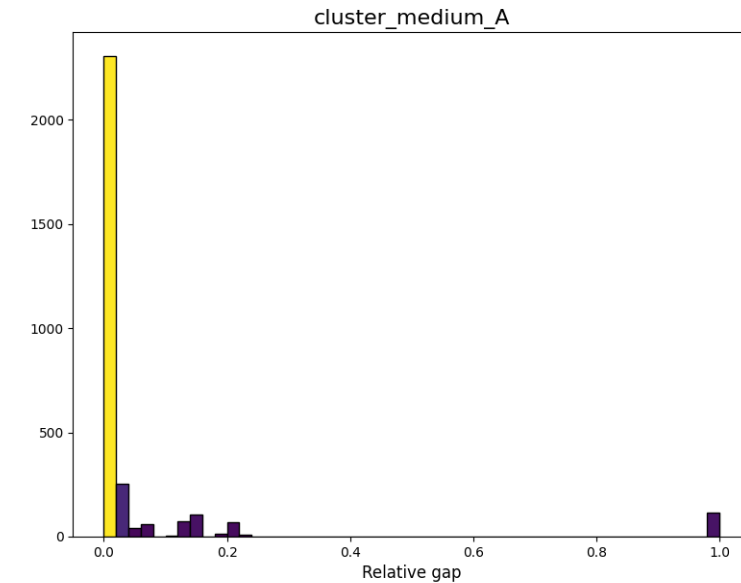
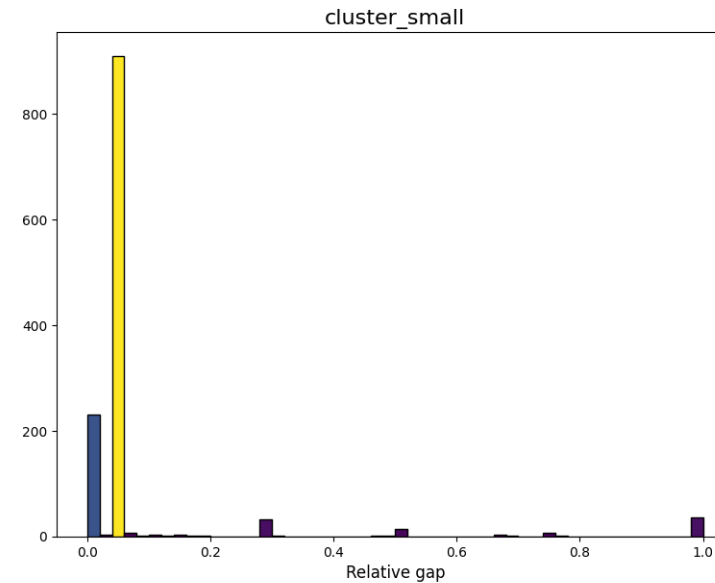


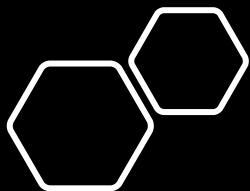


Distribuzioni dei relative gap

- Questi istogrammi rappresentano le distribuzioni dei valori del relative gap per ogni cluster
- Si evince come il trend di ogni cluster rispetti la threshold di 0.05

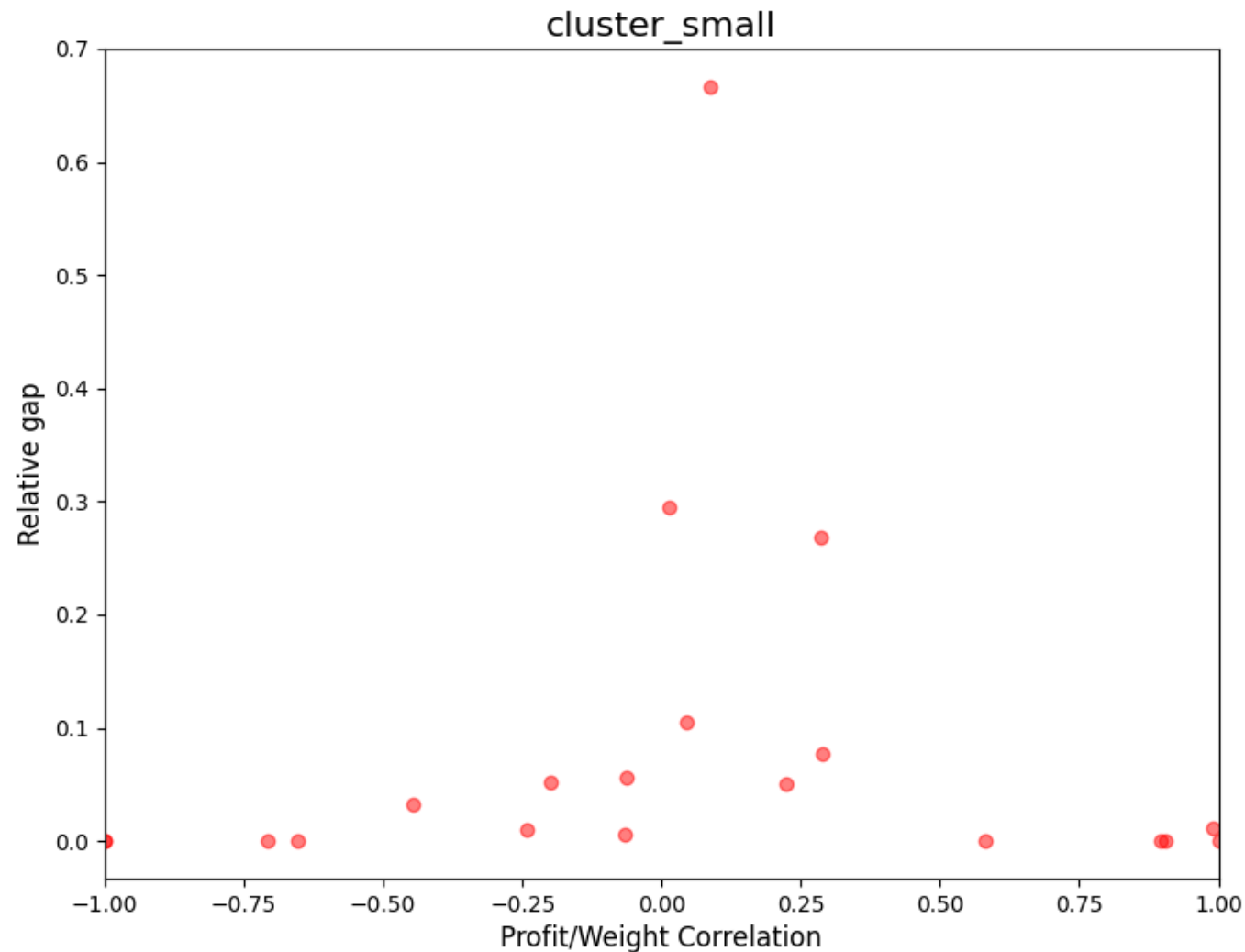
Relative gap distributions

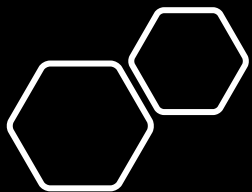




Prestazioni rispetto alla correlazione peso/profitto (1) : cluster small

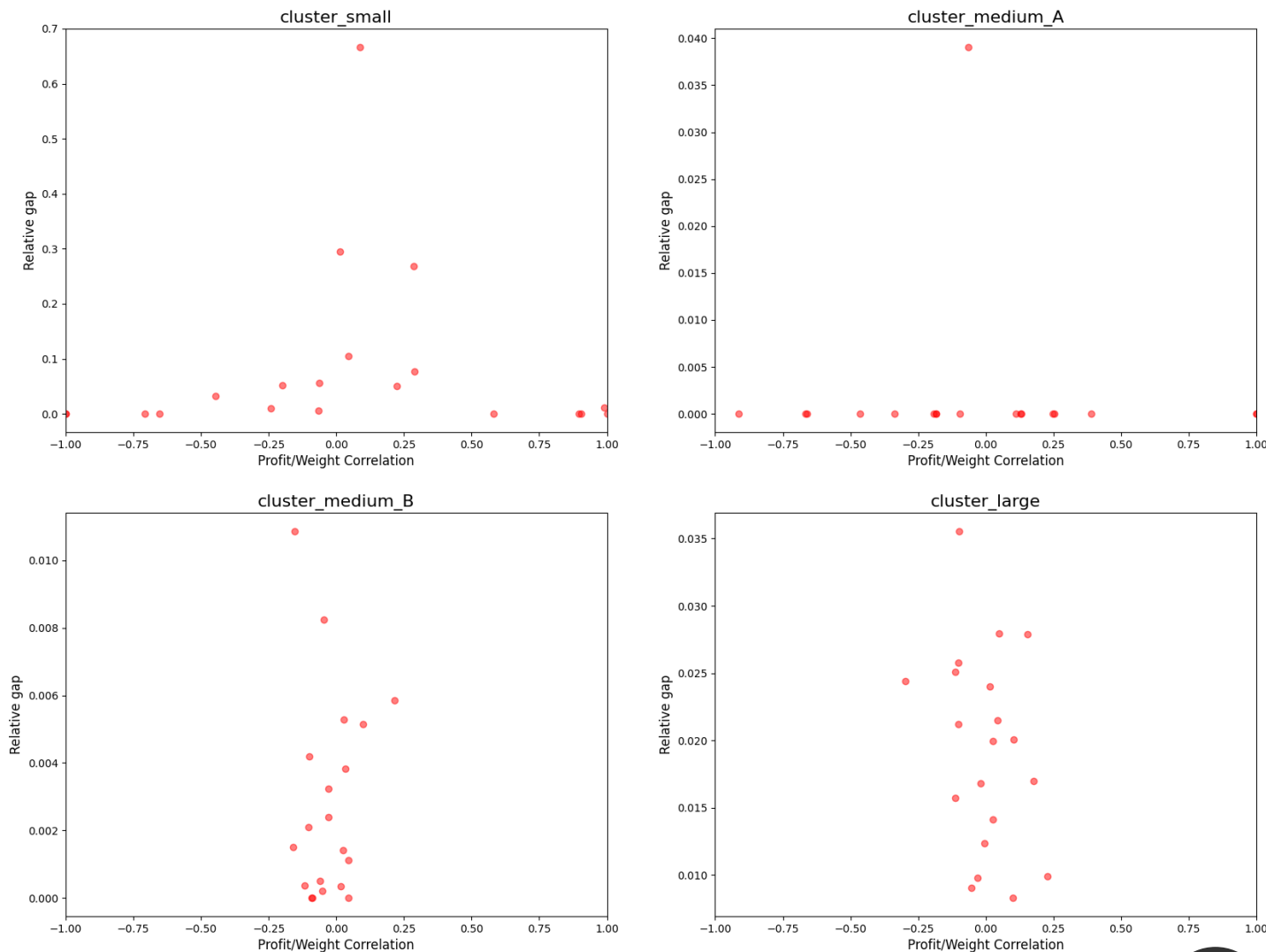
- Per ogni istanza è stata calcolata :
 - ❖ la correlazione tra i pesi e i profitti di ogni singola variabile
- Il dato correlation è stato salvato in un csv per ogni singolo cluster.
- Si può notare come all'aumentare della correlazione tra profitto e peso delle variabili nelle istanze, diminuisce il relative gap ($\text{corr}=[-1,1]$).

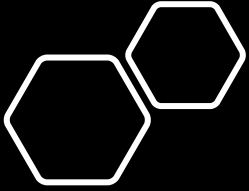




Prestazioni
rispetto alla
correlazione
peso/profitto
(2) :
tutti i clusters

Gomory effectiveness among profit/weight correlations





Conclusioni

- Dal nostro studio si evince come l'algoritmo dei tagli di Gomory tenda ad avere una efficacia migliore nel caso in cui il problema di multiknapsack presenta un numero limitato di vincoli (≤ 100).
- Il numero di variabili non sembra influire in modo significativo sull'efficacia dell'algoritmo dei tagli di Gomory.
- La quantità di istanze su cui è stato effettuato lo studio è limitata a 20 per motivi temporali.
 - Per avere una evidenza teorica più attendibile sarebbe necessario ripetere l'esperimento con una quantità maggiore di istanze e più tempo risolutivo a disposizione del solver.
- Tendenzialmente la correlazione tra profitto e peso delle variabili del problema di multiknapsack è direttamente proporzionale all'efficacia dell'algoritmo di Gomory.

GRAZIE PER
L'ATTENZIONE