Bounds and Bound Sets for Biobjective Combinatorial Optimization Problems :

set covering, set packing and set partitionning problems with two objectives

Xavier GANDIBLEUX[1] and Matthias EHRGOTT[2]

(1) LAMIH - Recherche Opérationnelle et Informatique
Université de Valenciennes et du Hainaut-Cambrésis
Le Mont Houy, F-59313 Valenciennes cedex 9 – FRANCE
Xavier.Gandibleux@univ-valenciennes.fr

(2) Department of Engineering Science
University of Auckland
Private Bag 92019, Auckland – NEW ZEALAND
m.ehrgott@auckland.ac.nz

MCDM 2002 Winter Conference
February 18-23, 2002 — Semmering, Austria

First study, 15th MCDM Int. Conf.

- **Bounds** :

  ideal point, nadir point.

- **Bound sets** :

  linear relaxation, greedy bounds, specific bounds (Martello-Toth bounds, Christofides'bound, . . .).

- **Biobjective combinatorial problems** :

  Assigment problem, Travelling salesman problem, Knapsack problem.

Today, 16th MCDM Int. Conf. :

- **Bound sets** :

  linear relaxation, greedy bounds.

- **Feedback following the numerical experiments**

  . limit of a general IP solver (LPSOLVE, CPLEX),

  . link between functions and difficulties to solve,

  . characteristics of solutions observed,

  . quality of approximations compared to exact solutions.

- **Biobjective combinatorial problems** :

  Set Covering Problem, Set Packing Problem, Set Partitioning Problem.

The Set Covering Problem (SCP)

$$\min \sum_{i=1}^{n} c_i^1 x_i$$

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

$$\begin{aligned} \text{subject to} \quad \sum_{i=1}^{n} a_{ji} x_i &\geq 1 \quad j = 1, \ldots, m \\ x_i &\in \{0, 1\} \end{aligned}$$

where a $a_{ji} \in \{0, 1\}$ and $a_{ji} = 1$ means variable x_i covers constraint j.

The Set Packing Problem (SPP)

$$
\begin{aligned}
\max \sum_{i=1}^{n} c_i^1 x_i \\
\max \sum_{i=1}^{n} c_i^2 x_i \\
\text{subject to} \quad \sum_{i=1}^{n} a_{ji} x_i &\leq 1 \quad j = 1, \ldots, m \\
x_i &\in \{0, 1\}
\end{aligned}
$$

where a $a_{ji} \in \{0, 1\}$ and $a_{ji} = a_{ji'} = 1$ means variable x_i and $x_{i'}$ are in conflict for ressource j.

The Set Partitioning Problem (SPA)

$$\min \sum_{i=1}^{n} c_i^1 x_i$$

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

$$\begin{aligned} \text{subject to} \quad \sum_{i=1}^{n} a_{ji} x_i &= 1 \quad j = 1, \ldots, m \\ x_i &\in \{0,1\} \end{aligned}$$

where a $a_{ji} \in \{0,1\}$ and $a_{ji} = 1$ means constraint j can be satisfied by variable x_i.

SCP/SPP/SPA : Important in practice

Airline crew scheduling

- Minimize cost
- Maximize robustness of solution

→ **Biobjective SPA**

Railway network infrastructure capacity

- Maximize number of trains
- Maximize robustness of solution

→ **Biobjective SPP**

Introduction

Notations and definitions

Numerical instances

Algorithms

Results

Conclusion

Notations and definitions

Multiobjective combinatorial optimization problem (MOCO)

$$\text{“}\min_{x \in X}\text{”} \ (z^1(x), \ldots, z^Q(x))$$

(MOCO) is a discrete optimization problem, with

- X **the decision space**,
- x **a binary vector** of variables $x \in \{0,1\}^n$,
- n **variables** x_i, $i = 1, \ldots, n$,
- Q **objectives** z^j, $j = 1, \ldots, Q$
- m **constraints** of specific structure defining X

- **Pareto optimality:**

  $x \in X$ Pareto optimal if there does not exist $x' \in X$ such that $z^q(x') \leq z^q(x)$ for all $q = 1, \ldots, Q$ and $z^p(x') < z^p(x)$ for some p

- **Efficiency:**

  x Pareto optimal then $z(x) = (z^1(x), \ldots, z^Q(x))$ is efficient/non-dominated

- set of Pareto optimal solutions: X_{Par}

- set of efficient values: E

Supported and Nonsupported Efficient Solutions

Linear programming

$$\min\{Cx : Ax = b, x \geq 0\}$$

E is set of solutions of

$$\min\left\{\sum_{\mathbf{j=1,\ldots,Q}} \lambda_{\mathbf{j}}\mathbf{c}^{\mathbf{j}}\mathbf{x} : \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\right\}$$

with $0 < \lambda < 1 \quad \sum_{J=1}^{Q} \lambda_j = 1$

(MOCO) $\rightarrow$ supported efficient solutions SE, **nonsupported efficient solutions** NE exist

Lower and upper bound sets (min)

A **lower bound set** for $\overline{Z}$ is a subset $L \subseteq \mathbb{R}^Q_+$ such that

1. for each $z \in \overline{Z} \exists l \in L$ such that $l_q \leq z^q(x),\ q = 1, \ldots, Q$
2. there is **no** pair $z \in \overline{Z}, l \in L$ such that z **dominates** l

An **upper bound set** for $\overline{Z}$ is a subset $U \subseteq \mathbb{R}^Q_+$ such that

1. for each $z \in \overline{Z} \exists u \in U$ such that $z^q \leq u_q, q = 1, \ldots, Q$
2. there is **no** pair $z \in \overline{Z}, u \in U$ such that u **dominates** z

Introduction

Notations and definitions

Numerical instances

Algorithms

Results

Conclusion

Characteristics of numerical instances

The sizes :

	n = #variables	m = #constraints
SCP/SPP	100 ... 1000	10... 200
SPA	100 ... 1000	10 ... 25

The constraints : reduced (SCP)

SCP/SPP	density = 2% ... 34%	max#1 = 10... 200
SPA	max#1 = 0.15*n ... 0.40*n	

Series :	11	41	42	43	61	62	81	82	101	102	201
n	100	200	400	200	600	600	800	800	1000	1000	1000
m	10	40	40	40	60	60	80	80	100	100	200
density	high	low	low	high	low	high	low	high	low	high	low

The objectives : four families

A: random

c_i^1, c_i^2 randomly generated $i = 1, \ldots, n$;

B: conflictual

c_i^1 randomly generated $i = 1, \ldots, n$; $c_{n-i+1}^2 = c_i^1 \; i = 1, \ldots, n$;

C: patterns

l_1=rnd(), l_2=rnd(),... ; v_1=rnd(), v_2=rnd(),... ;

$c_1^1 = c_2^1 = \ldots = c_{l_1}^1 = v_1$; $c_{l_1+1}^1 = c_{l_1+2}^1 = \ldots = c_{l_1+l_2}^1 = v_2$; ...

D: conflictual patterns

B and C combined;

11A, 11B, 11C, 11D, 41A, ..., 201D :

44 instances available on the MCDM society WWW site

Exact LP & 01 solutions: a two phases algorithm

firstPhase : **procedure** () **is**

--| Compute $x^{(1)}$ and $x^{(2)}$, the lexicographically optimal solutions for
--| permutations (z^1, z^2) and (z^2, z^1) of the objectives.

$x^{(1)} \leftarrow$ **SolveLexicography** $(z1 \downarrow\ ,\ z2 \downarrow)$
$x^{(2)} \leftarrow$ **SolveLexicography** $(z2 \downarrow\ ,\ z1 \downarrow)$
$S \leftarrow \{x^{(1)}, x^{(2)}\}$

--| Compute all solutions between $x^{(1)}$ and $x^{(2)}$.
--| Update S with all new solutions generated.

`solveRecursion`$(x^{(1)} \downarrow\ ,\ x^{(2)} \downarrow\ ,\ S \updownarrow)$

end firstPhase

`solveRecursion` **: procedure** ($x^{(A)} \downarrow$, $x^{(B)} \downarrow$, $S \updownarrow$) **is**

--| Compute the optimal solutions $x^{(C)}$ of (P_λ) : $\min\{\lambda_1 z^1(x) + \lambda_2 z^2(x) \mid x \in X\}$
--| where $\lambda_1 = z^2(x^{(A)}) - z^2(x^{(B)})$, and $\lambda_2 = z^1(x^{(B)}) - z^1(x^{(A)})$.

$x^{(C)} \leftarrow$ **Solve**P_λ $(\lambda \downarrow, z^1(x^{(B)}) \downarrow, z^2(x^{(A)}) \downarrow)$
if `exist`($x^{(C)}$) **then**
  $S \leftarrow S \cup \{x^{(C)}\}$
  `solveRecursion`($x^{(A)} \downarrow$, $x^{(C)} \downarrow$, $S \updownarrow$)
  `solveRecursion`($x^{(C)} \downarrow$, $x^{(B)} \downarrow$, $S \updownarrow$)
end if

end solveRecursion

SolveLexicography, SolveP_λ : CPLEX 6.6.1 library is called.

Approximated solutions: heuristics

- SCP : constructive greedy algorithm setting variables from 0 to 1
  - choose smallest $c_i(\lambda)$ such that x_i covers an additional constraint
  - stop when all constraints are covered (satisfied)
- SPP : constructive greedy algorithm setting variables from 0 to 1
  - choose biggest $c_i(\lambda)$ such that x_i satisfies an additional constraint
  - stop when it is impossible to saturate again one constraint
- SPA : simulated annealing coupled with a local search

SPA : simulated annealing coupled with a local search

1. compute a SCP feasible solution
2. change the solution to have a SPP compatible solution
3. **if** this is not a SPA feasible solution **then**

   --| start a simulated annealing using this solution

   a. the objective function : min the # of unsatisfied constraints
   b. the move : flip01 and flip10
   c. apply a local search ((1-1 Exchange) when SA accepts a neighbor
   d. restart (change the solution to have a SPP compatible solution) when SA does not produce a feasible solution after a given condition

   endIf
4. improve the feasible solution with a local search (1-1 Exchange) using the convex combination of the objective

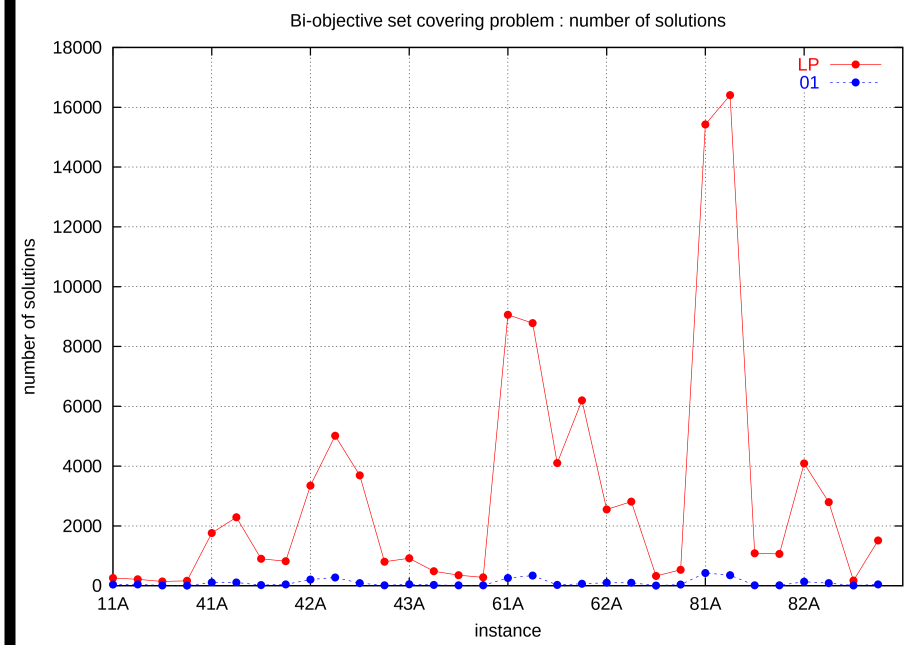
The context

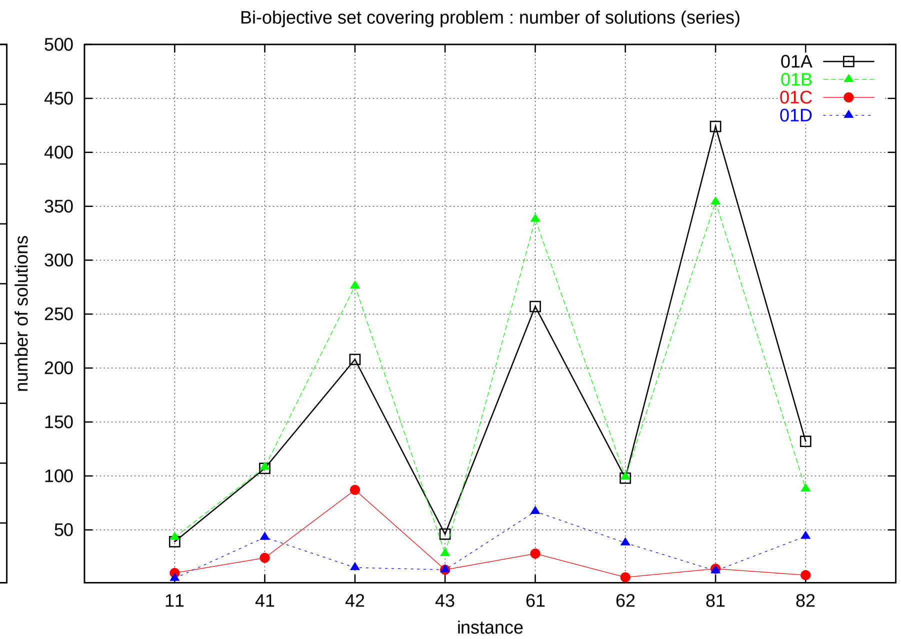
- Bounds sets (min)
  - **Lower bound** Computed as continuous relaxation
  - **Upper bound** Obtained after application of a heuristic

- Implementation
  - C language
  - Cplex : a mainframe / Unix
  - Heuristics : PowerPC G4 450Mhz / 128Mb / MacOS X

  Remarks
  - performances of both machines are close
  - CPUt / heuristics / SCP-SPP not significant

SCP : Solutions

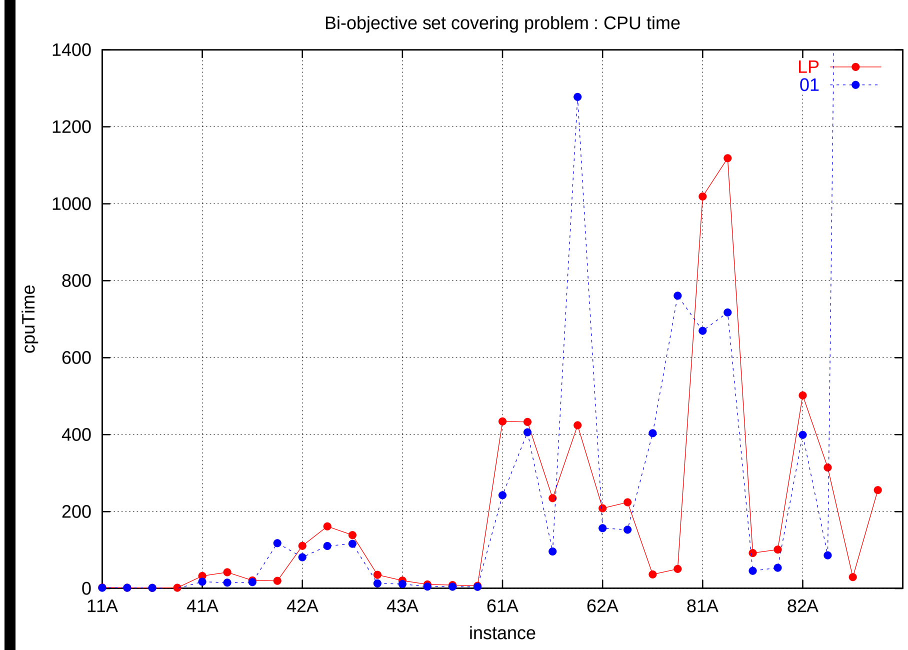


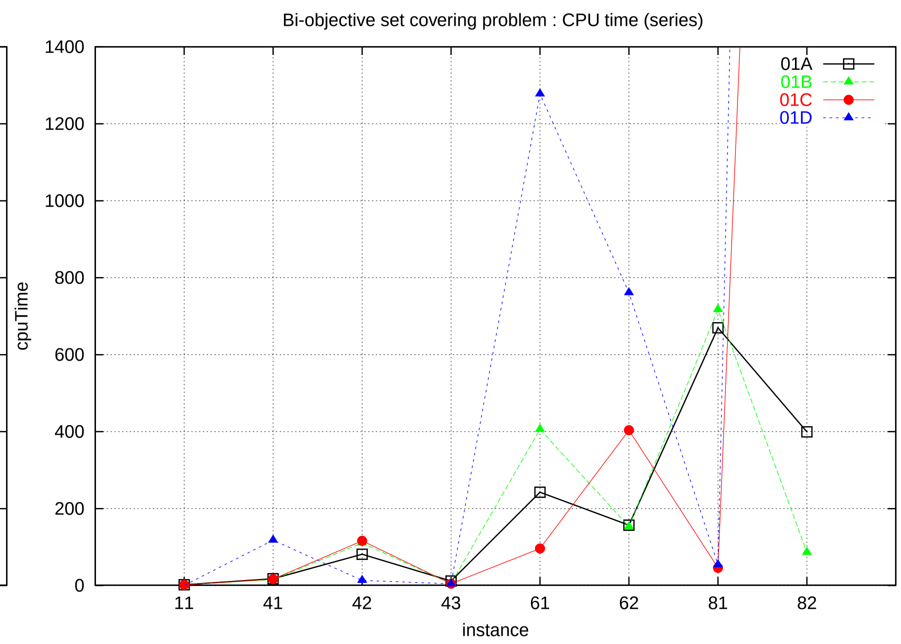


⇒ #solutions grows significantly for instances “low density”, “without pattern”;

⇒ in any case, #solutions for instances “with patterns” is low.

SCP : CPU time





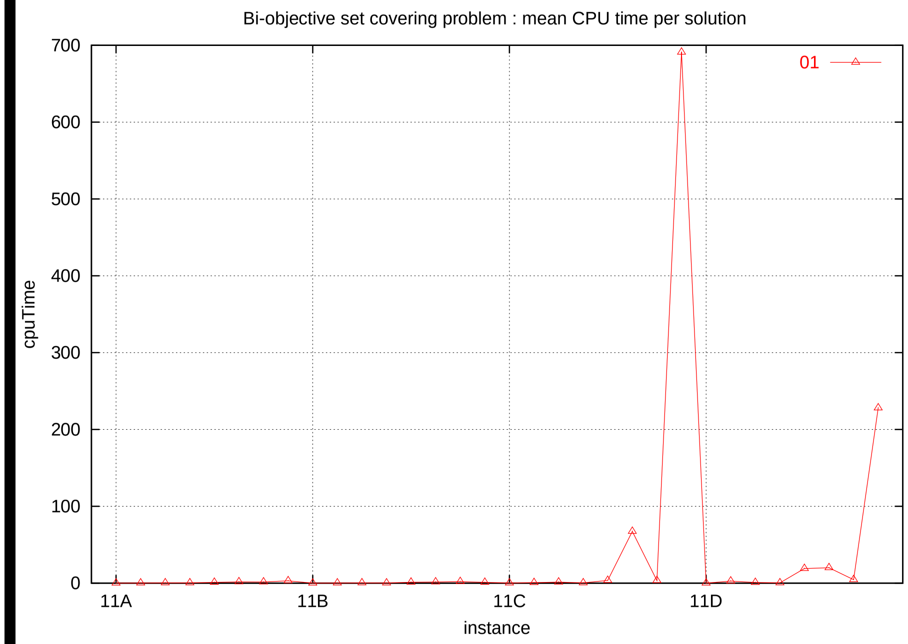
⇒ Until serie 82, CPLEX can solve;

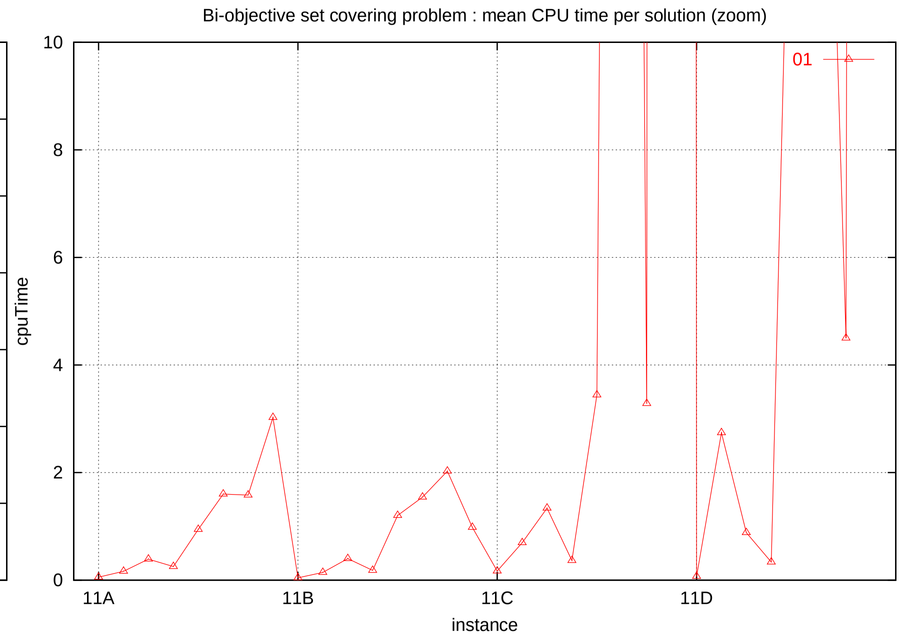
. 11/41/42/43 : **easy** instances

. 61/62/81/82 : **interesting** instances

. 101/102/201 : **difficult** instances (specially "with patterns")

SCP : mean time

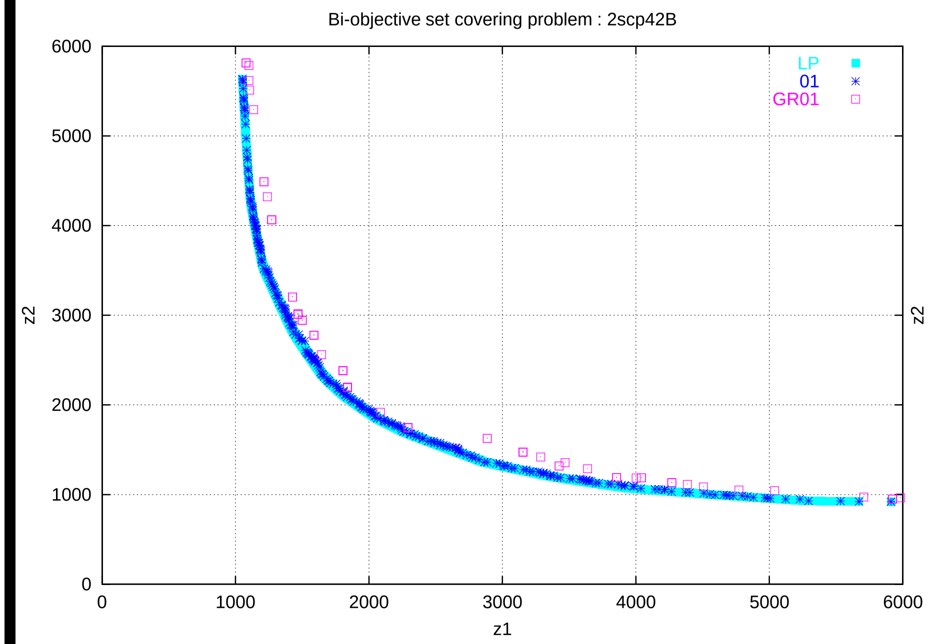


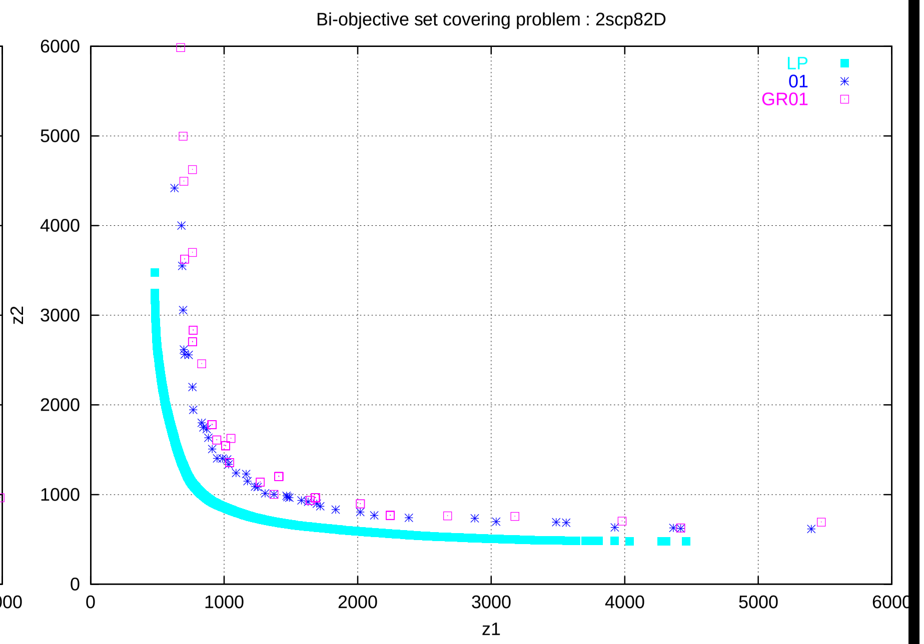


⇒ Mean time to generate one solution could be important specially for instances 62/82 (“medium size, high density”).

⇒ Families C&D (“high density, with patterns”) seems the most difficult to solve for any instances.

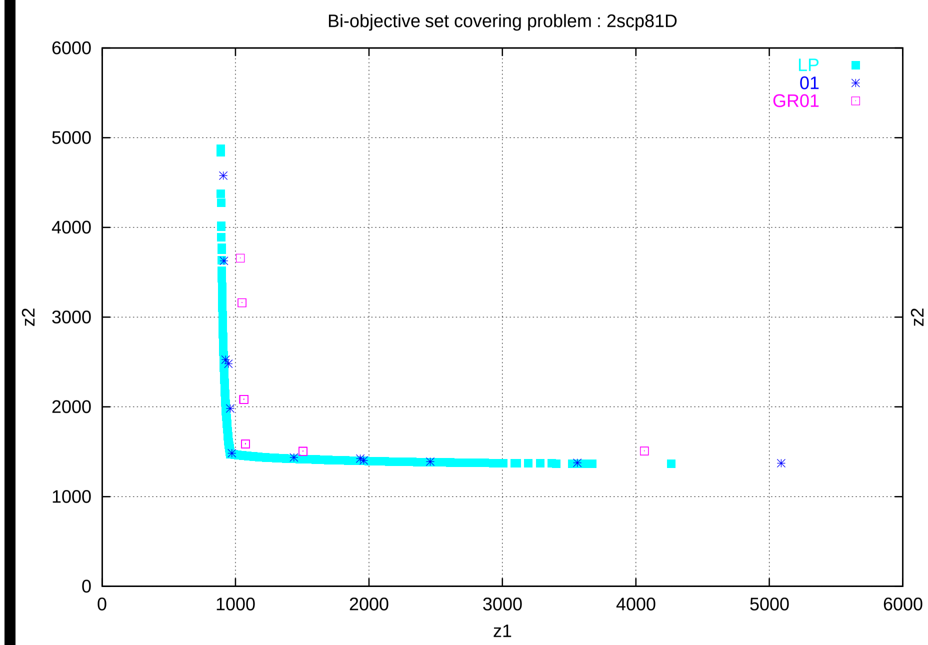
SCP : Lower bounds (LP)

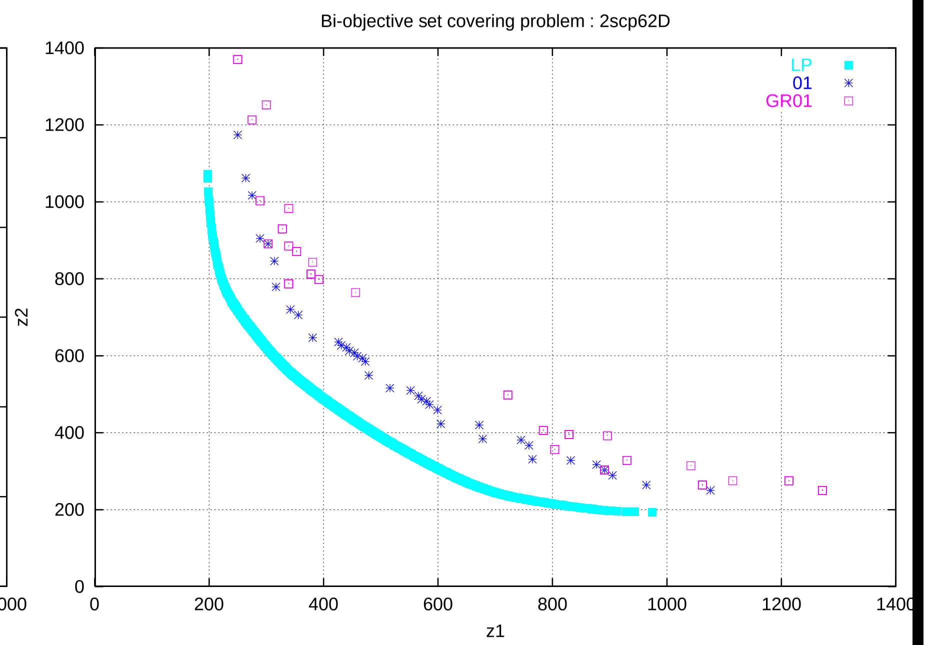




- good : - lower bound and E are very close ("low density")
  - LP overlaps E ("low density without pattern")
- bad : - "high density, with patterns"

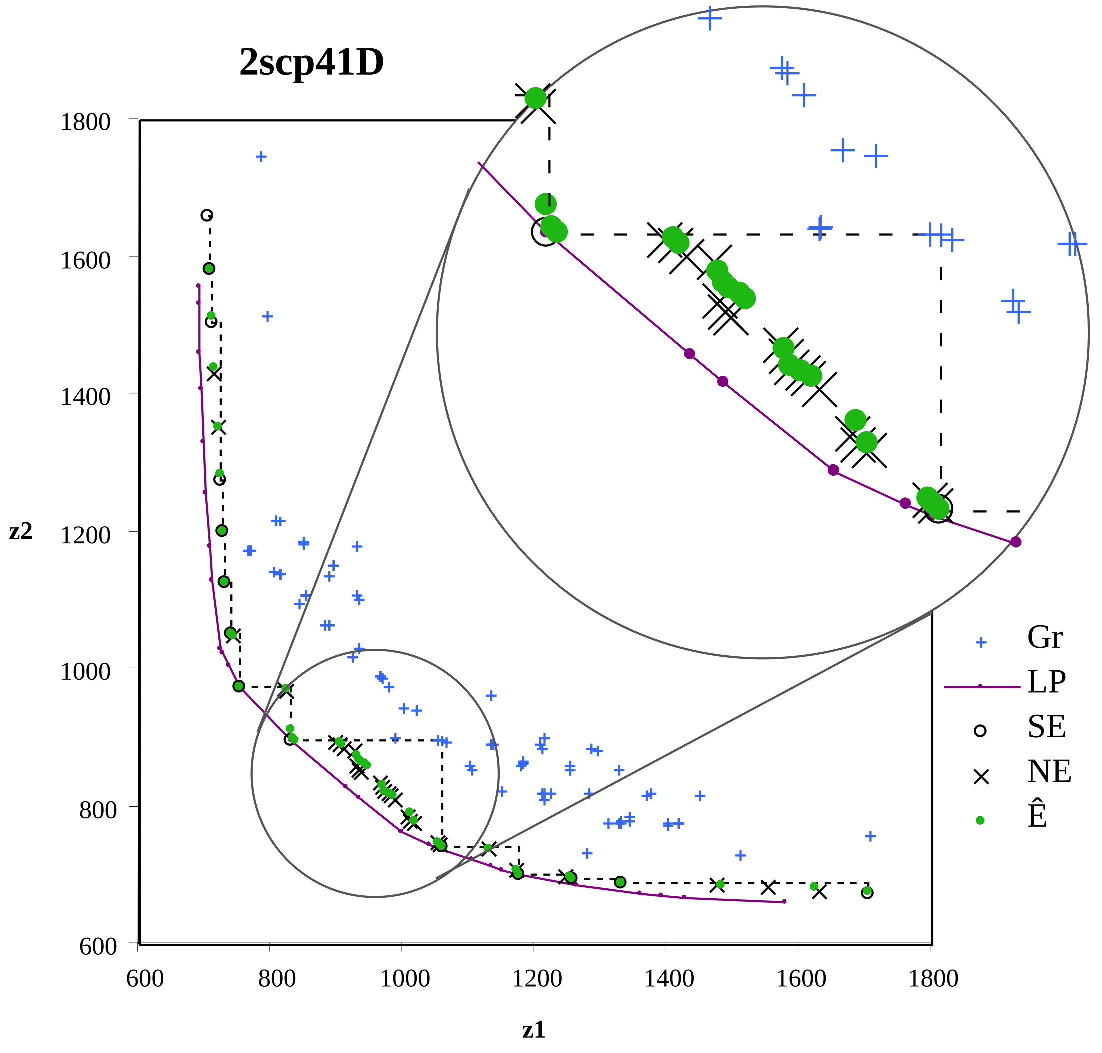
SCP : Upper bounds (GR01)





- good : - greedy well distributed ("low density without pattern")
  - unusual frontier, greedy not well spread ("low density with patterns")
- bad : - not well spread, clusters ("high density")

GRASP metaheuristic to solve biSCP (14thMCDM)

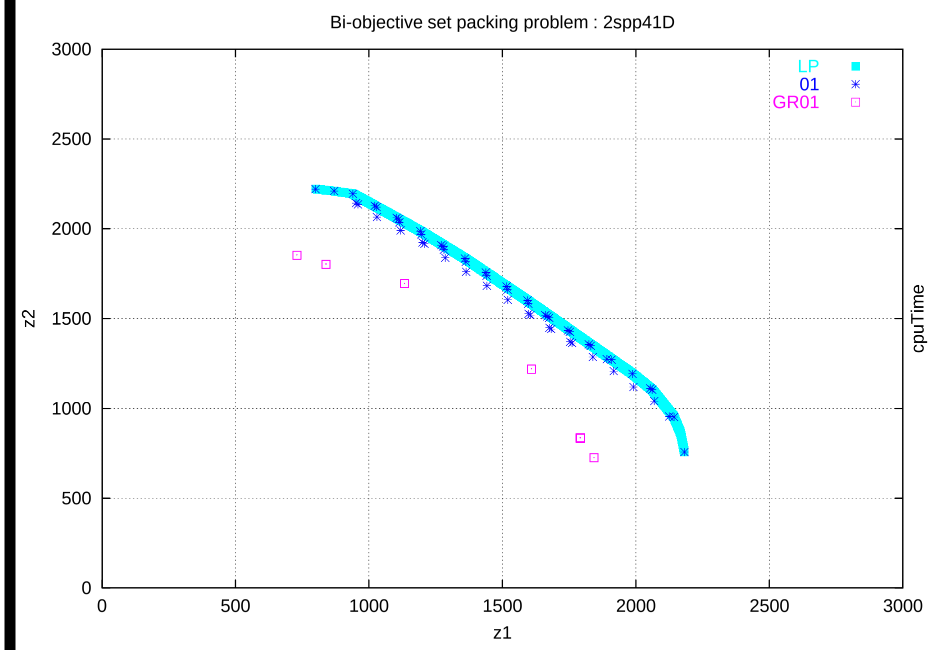


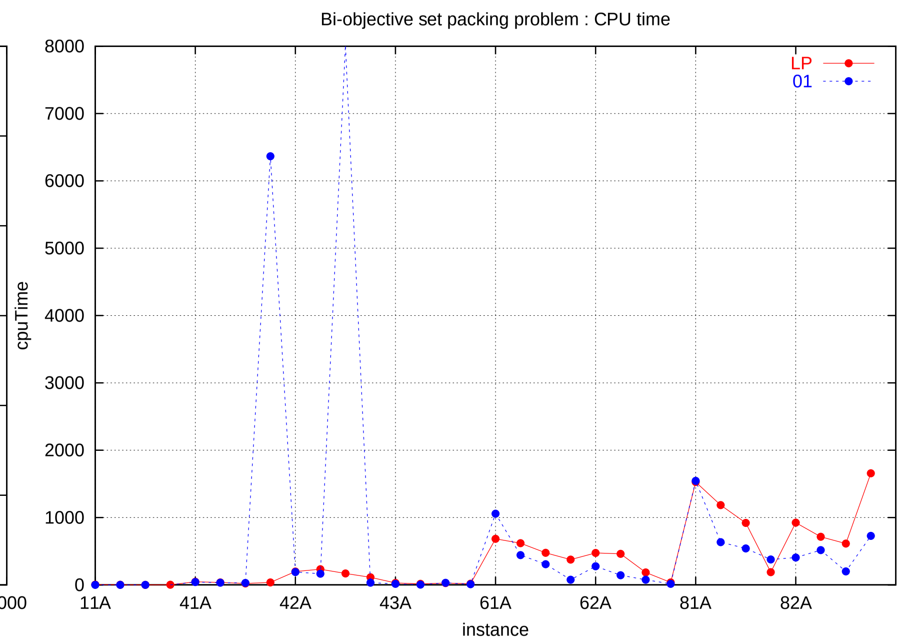
SPP : General remarks (11-82 / A-D)

- Cplex is able to solve biggest instances (efficient preprocessing before to solve the instance)

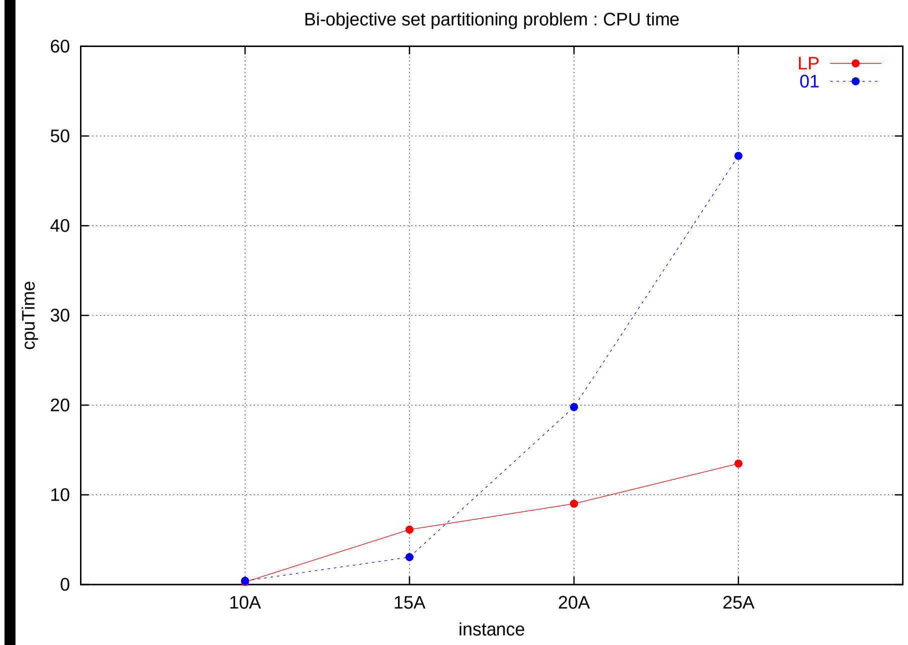
  ⇒ instances SCP not pertinent to draw the the limits and difficulty of a solver;
- Mean time to generate a solution is often greater than for the SCP;
- Upper bound (LP) : same comments than for the SCP;
- Lower bound (GR01) : in general, not famous, well distributed, some clusters (instances "with patterns");
- ⇒ all instances are **easy** except...

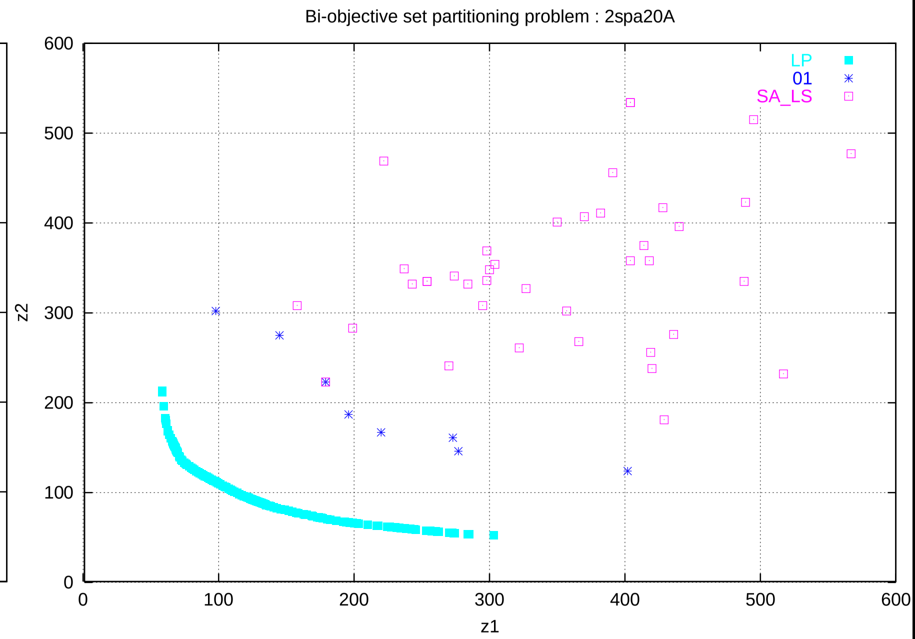
SPP : two very hard instances : 41D and 42C !





SPA : (10-25 / A)





No enough results to draw any robust conclusion, just some comments. . .

SPA : some comments

- Lower bound (LP) is not good.
- Feasible solutions are interesting even if they are constructed without a strong optimization toward the efficient frontier.
- Testing :
  - penalty function
  - reject rule based on performances
  - Strength Pareto Evolutionary Algorithm (SPEA by Zitzler and Thiele, 1998)

Introduction

Notations and definitions

Numerical instances

Algorithms

Results

Conclusion

Conclusion

- Not all results have been presented (supported and non-supported solutions, distance measures, etc.)
- SCP : highlighted a class of not friendly problems :
  SCP with high density and patterns :
  - difficult to solve
  - Bad lower bound (LP) : great distance
  - Bad upper bound (GR01) : clusters and holes
- SPP : to build other instances
- SPA : to continue. . .