ECCO 2005
Minsk, May 26 – 28 2005

A branch-and-price algorithm for the ordered open-end bin-packing problem

A. Ceselli, G. Righini
Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione

email:
ceselli@dti.unimi.it
righini@dti.unimi.it

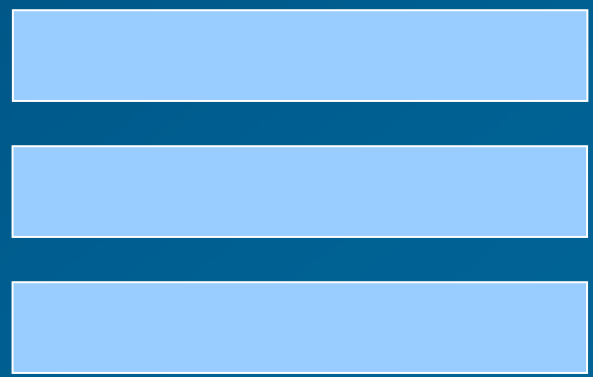
Outline of the talk

- The ordered open-end bin-packing problem (OOBP)
  - Literature review
- Formulations and LP dual bounds
  - Compact formulation
  - Set covering formulation
- A branch-and-price algorithm
  - Column generation approach
  - The open-end knapsack problem
  - Primal bounds
  - Branching scheme
- Combinatorial dual bounds
- Experimental analysis

The OOBP

Input:

(1) a sequence of items (J), each of given width w_j

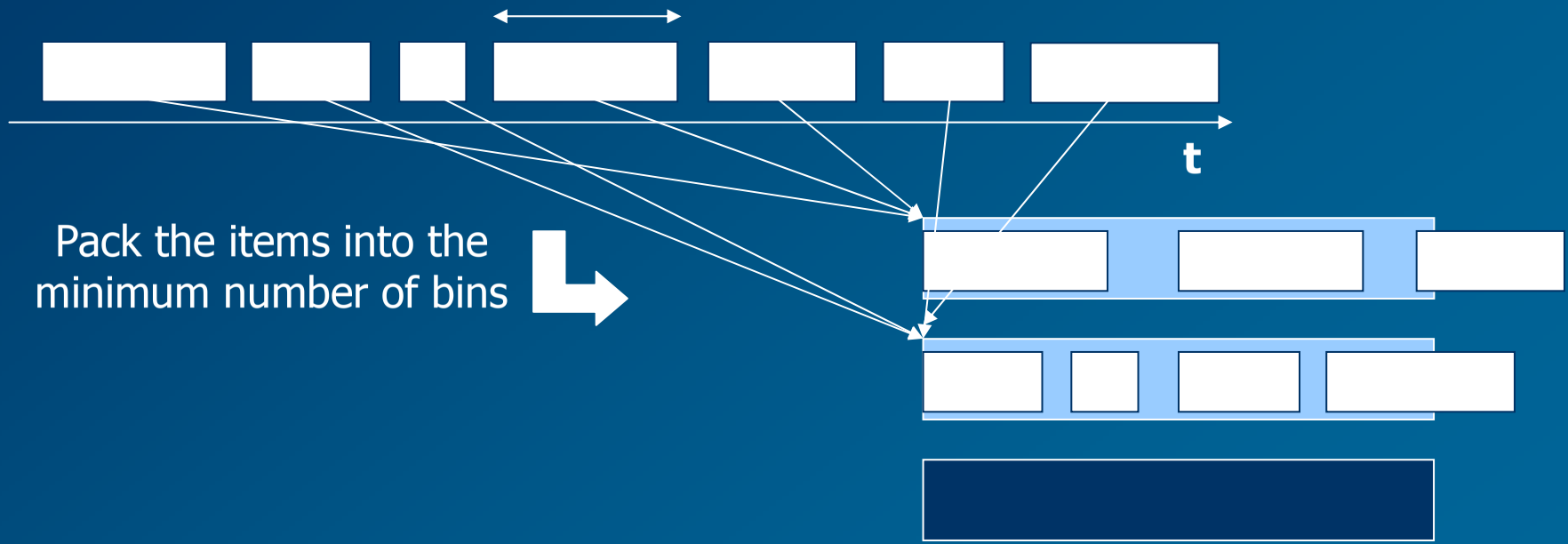


(2) a set of bins, each of the same width W

The OOBP

Objective:

(1) a sequence of items (J), each of given width wj



(2) a set of bins, each of the same width W

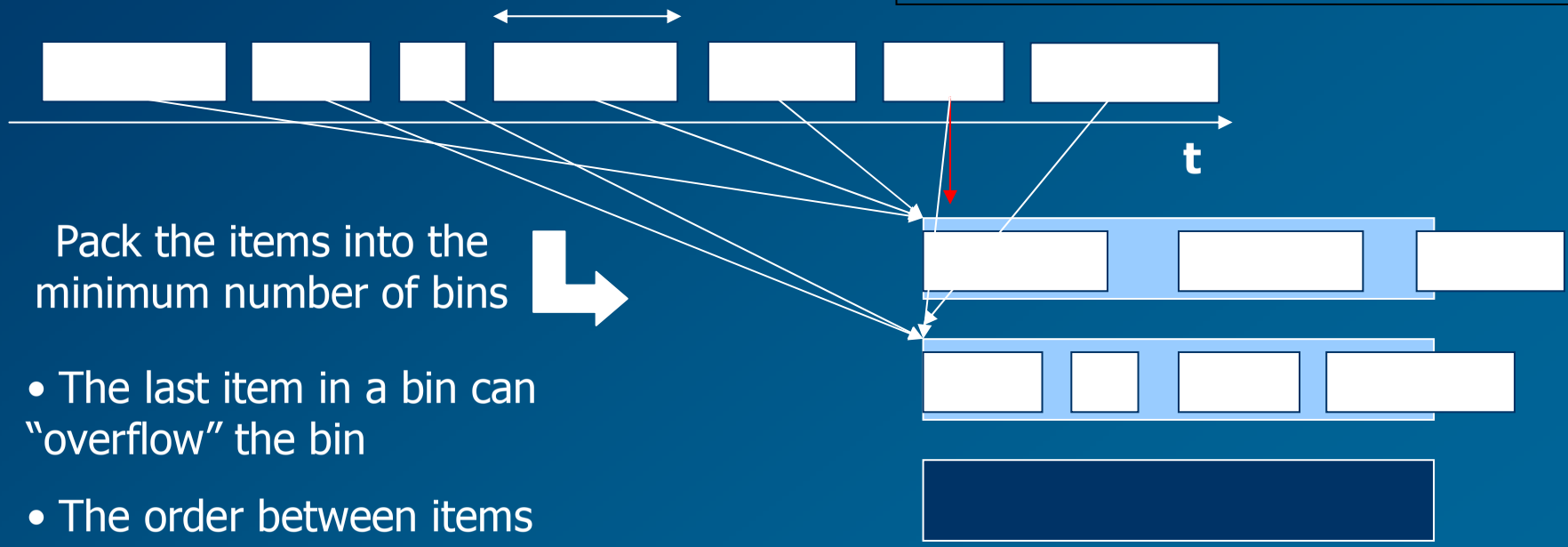
The OOBP

Constraints:

(1) a sequence of items (J), each of given width wj

Hong Kong subway:
items = travels, bins = tickets

Telephone cards:
items = calls, bins = cards



Pack the items into the minimum number of bins

- The last item in a bin can "overflow" the bin
- The order between items is preserved
- (Each item cannot be split)

(2) a set of bins, each of the same width W

Literature review

- Complexity: NP-Hard
- Main ideas: Yang and Leung, Operations Research (2003)
  - Online approximation algorithms:
    - A ≥ 1.630297 approx (with "1-pieces") 1.415715 approx (without "1-pieces")
    - MXF: [1.9231,1.9444] approx
  - Offline approximation algorithms:
    - GLANF: [1.35,1.5] approx (1.5 approx without "1 – pieces")
  - Average case analysis
    - A ≥ $2 - \sqrt{3}$
    - DP: ($2 - \sqrt{3}$) approx
- Our contribution:
  - ILP formulations
  - Lower bounds
  - Upper bounds
  - Exact algorithms

A compact ILP formulation

$$\min \sum_{j \in J} x_{jj}$$

$$s.t. \begin{cases} \sum_{j \geq i} x_{ij} = 1 & \forall i \in J \\ \sum_{i < j} w_i x_{ij} \leq (W-1) x_{jj} & \forall j \in J \\ x_{ij} \in \{0,1\} & \forall i, j \in J \end{cases}$$

LP bound

$$\min \sum_{j \in J} x_{jj}$$

$$s.t. \begin{cases} \sum_{j \geq i} x_{ij} = 1 & \forall i \in J \\ \sum_{i < j} w_i x_{ij} \leq (W-1) x_{jj} & \forall j \in J \\ 0 \leq x_{ij} \leq 1 & \forall i, j \in J \end{cases}$$

Set covering formulation

$$\min v = \sum_{k \in K} z^k$$

$$s.t.$$

$$\begin{cases} \sum_{k \in K} x_j^k z^k \geq 1 & \forall j \in J \\ z^k \in \{0,1\} & \forall k \in K \end{cases}$$

$$\forall k \in K:$$

$$\begin{cases} \sum_{j<j^*} w_j x_j^k \leq (W-1) x_{j^*}^k \\ x_j^k \in \{0,1\} \qquad \forall j \in J \end{cases}$$

- DW reformulation of the compact model

LP bound: column generation

$$\min v = \sum_{k \in K} z^k$$

$$s.t.$$

$$\begin{cases} \sum_{k \in K} x_j^k z^k \geq 1 & \forall j \in J \quad (\lambda_j) \\ 0 \leq z^k \leq 1 & \forall k \in K \end{cases}$$

$$\forall k \in K: \quad \begin{cases} \sum_{j<j^*} w_j x_j^k \leq (W-1) x_{j^*}^k \\ x_j^k \in \{0,1\} \qquad \forall j \in J \end{cases}$$

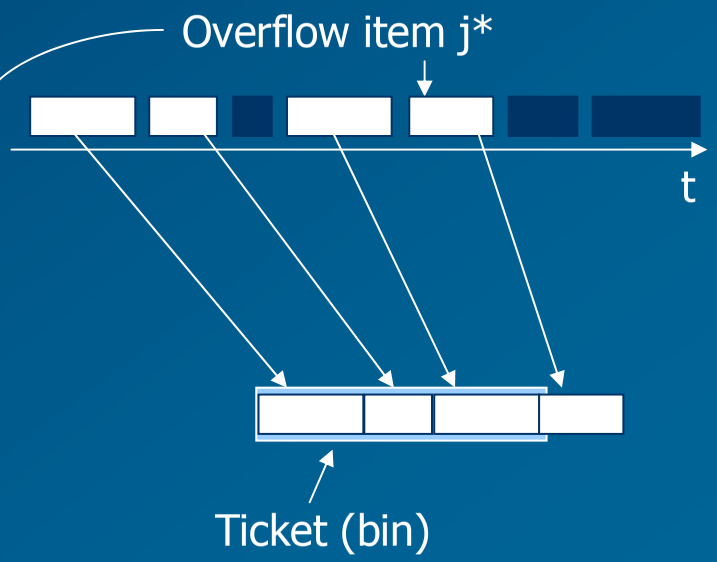
$$rc(k) = 1 - \sum_{j \in J} \lambda_j x_j^k$$

Pricing problem: ordered open-end KP

Reward λ_j for each item j

$$\max \sum_{j \in J} \lambda_j x_j^k$$

$$s.t. \begin{cases} \sum_{j<j^*} w_j x_j^k \leq (W-1) x_{j^*}^k \\ x_j^k \in \{0,1\} \qquad \forall j \in J \end{cases}$$



- O(|J| * (W-1)) with standard KP dynamic programming recursion

Pricing algorithm

Adaptation of an algorithm for PKP (Ceselli and Righini '05)

1. Preprocessing (removal of candidate overflow items)
2. LP bounds ($O(N^2)$ time)
3. Best bound search for an optimal overflow item:
   1. Solution of a KP
   2. Reduction criterion

Branching schemes

- Two level branching:
  - First level:
    fix the x_{jj} variables (overflow items)
  - Second level:
    solve the remaining GAP feasibility subproblem (fix the most fractional x_{ij} variable)

Primal bounds

- Algo BFDH:
  - Input: an OOBP instance <J,w,W>
  - Output: an upper bound on the number of needed bins
  - Init: T := 0, r_j := 0.0 for all j in J
  - For j := |J| to 1 do
    - T' := {t in T | $r_t \geq w_j$}
    - If (T' = 0)
      - T := T U {j}
      - r_j := W - 1
    - t* := argmin $_{t \text{ in } T'}$ {$r_t - w_j$}
    - $r_{t*} := r_{t*} - w_j$
  - Output |T|
- Randomized BFDH heuristic (select a random initial T set, run BFDH)
- Rounding heuristic for set covering, (MT-like heuristic, randomized rounding heuristic)

Combinatorial bounds: W bound

- Algo W-bound:
  - Input: an OOBP instance <J,w,W>
  - Output: a lower bound on the number of needed bins
  - Init: T := 0, availCap := 0.0
  - While (availCap < $\sum_j w_j$)
    - t* := argmax $_{j \text{ in } J \setminus T}$ $\{w_j\}$
    - T := T U {t*}
    - availCap := availCap + (W – 1) + w_{t*}
  - Output |T|
- O(|J| log |J|)

Combinatorial bounds: TB bound

- Algo TB-bound:
  - Input: an OOBP instance <J,w,W>
  - Output: a lower bound on the number of needed bins
  - T := 0, availCap := 0.0
  - For each j in J
    - If (availCap < w_j)
      - $t^* := \text{argmax}_{j' \text{ in } J \setminus T \mid j' \leq j} \{w_{j'}\}$
      - $T := T \cup \{t^*\}$, $b(t^*) := j$
      - availCap := availCap + (W – 1) + w_{t^*}
    - availCap := availCap – w_j
  - Output |T|
- $O(|J|^2)$

Combined bound

$$\min v = \sum_{k \in K} z^k$$

$$s.t.$$

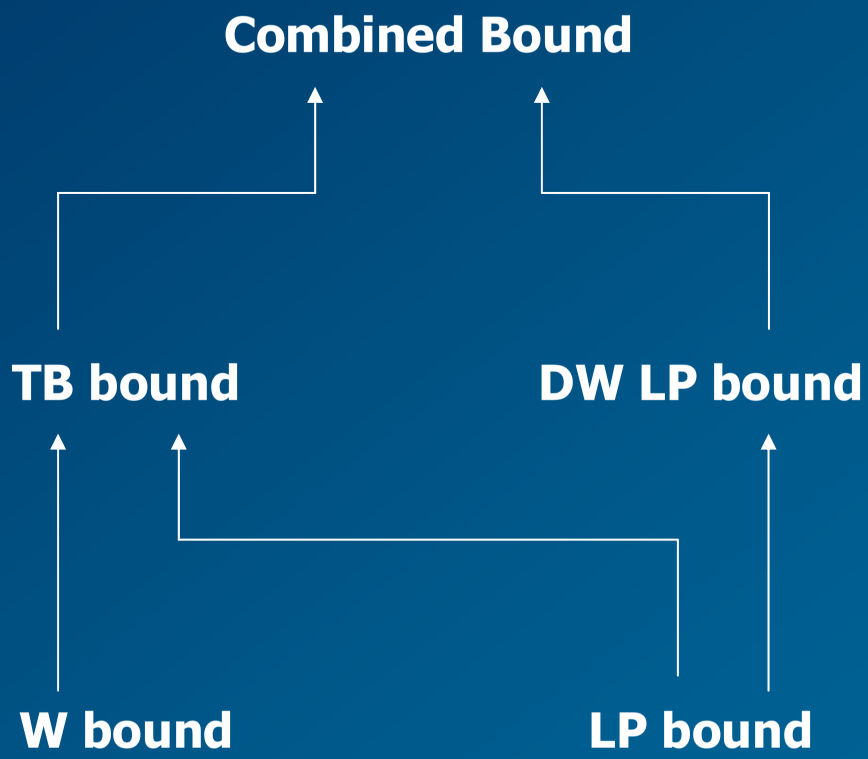
$$\begin{cases} \sum_{k \in K} x_j^k z^k \geq 1 & \forall j \in J \\ \sum_{k \in K_t} z^k \geq t & \forall t \in T \\ 0 \leq z^k \leq 1 & \forall k \in K \end{cases}$$

$$\forall k \in K:$$

$$\begin{cases} \sum_{j < j^*} w_j x_j^k \leq (W-1) x_{j^*}^k \\ x_j^k \in \{0,1\} \qquad \forall j \in J \end{cases}$$

$$K_t = \{k \in K \mid j^* \geq b(t)\}$$

Quality of bounds



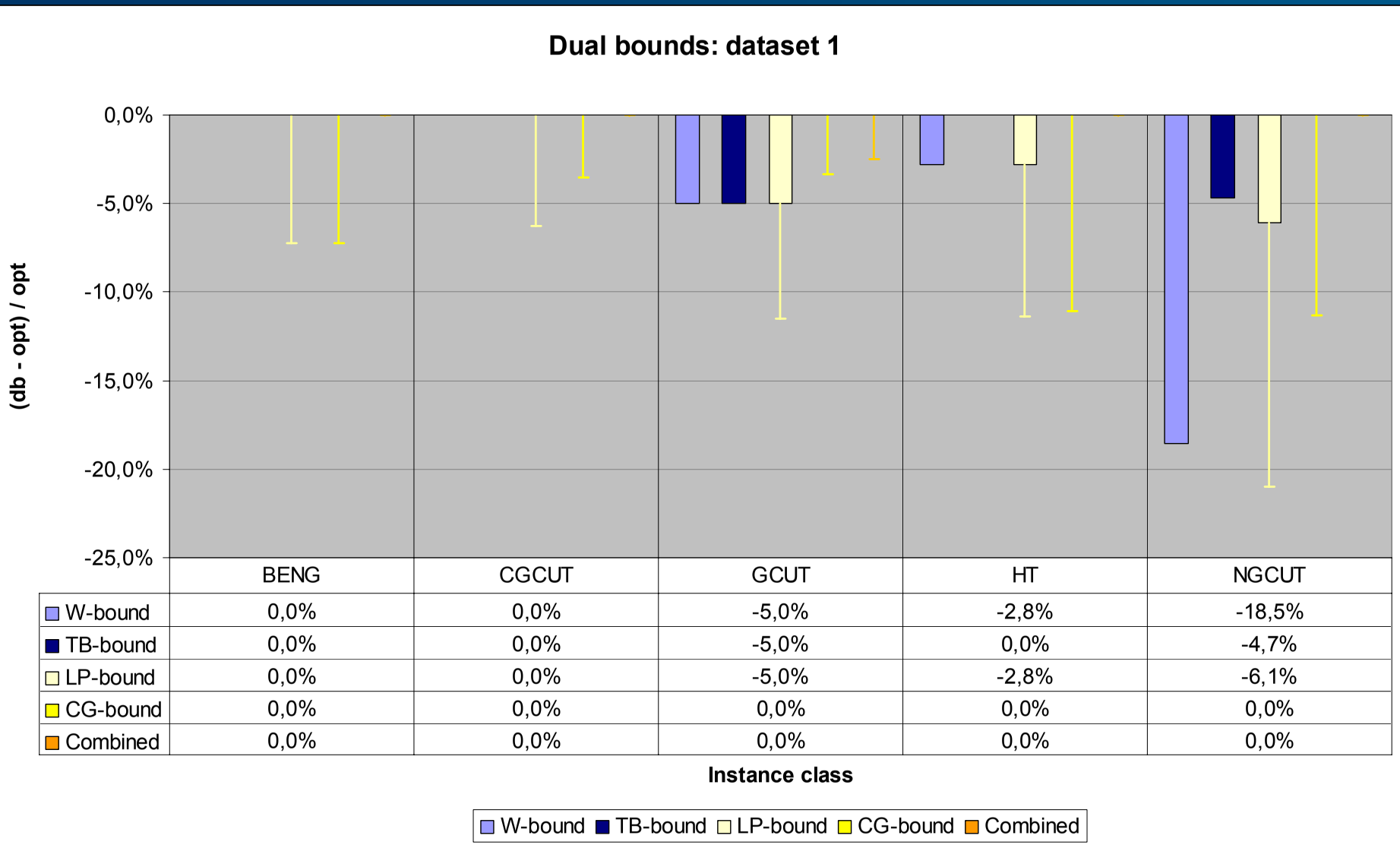
Experimental analysis

- Dataset 1: 38 instances for 2SP taken from the literature (ORLib)
- Dataset 2: 500 instances for 2BP taken from the literature (Lodi et al. '04)
- (Heights interpreted as 'time stamps')

- C++ implementation
- P IV 1.6 GHz CPU, 512 MB RAM
- Linux O.S.
- Resorce limitations: time limit of 1 hour, memory overflow

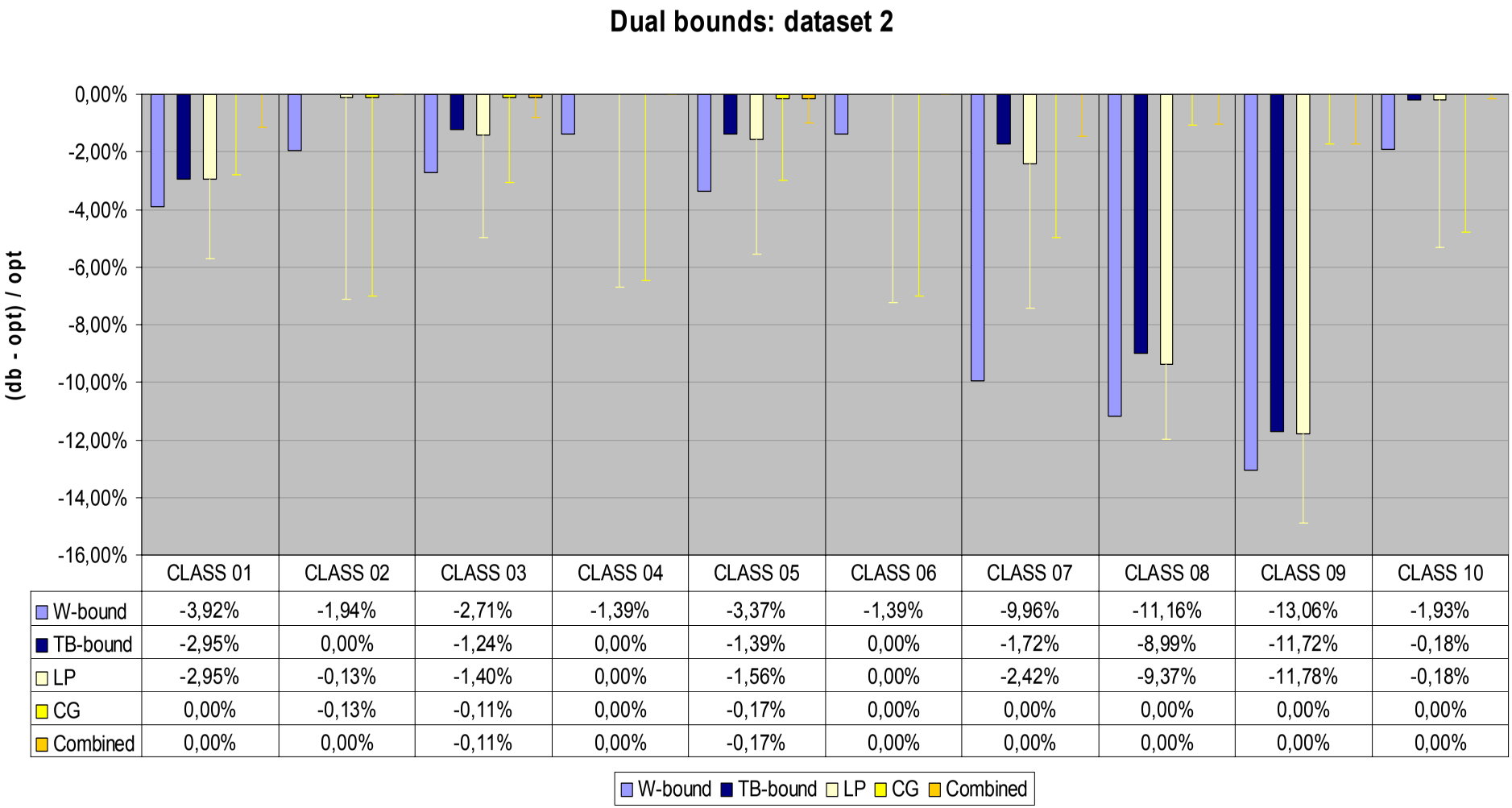
- Benchmark: CPLEX 8.1 general purpose solver, compact formulation

Dual bounds



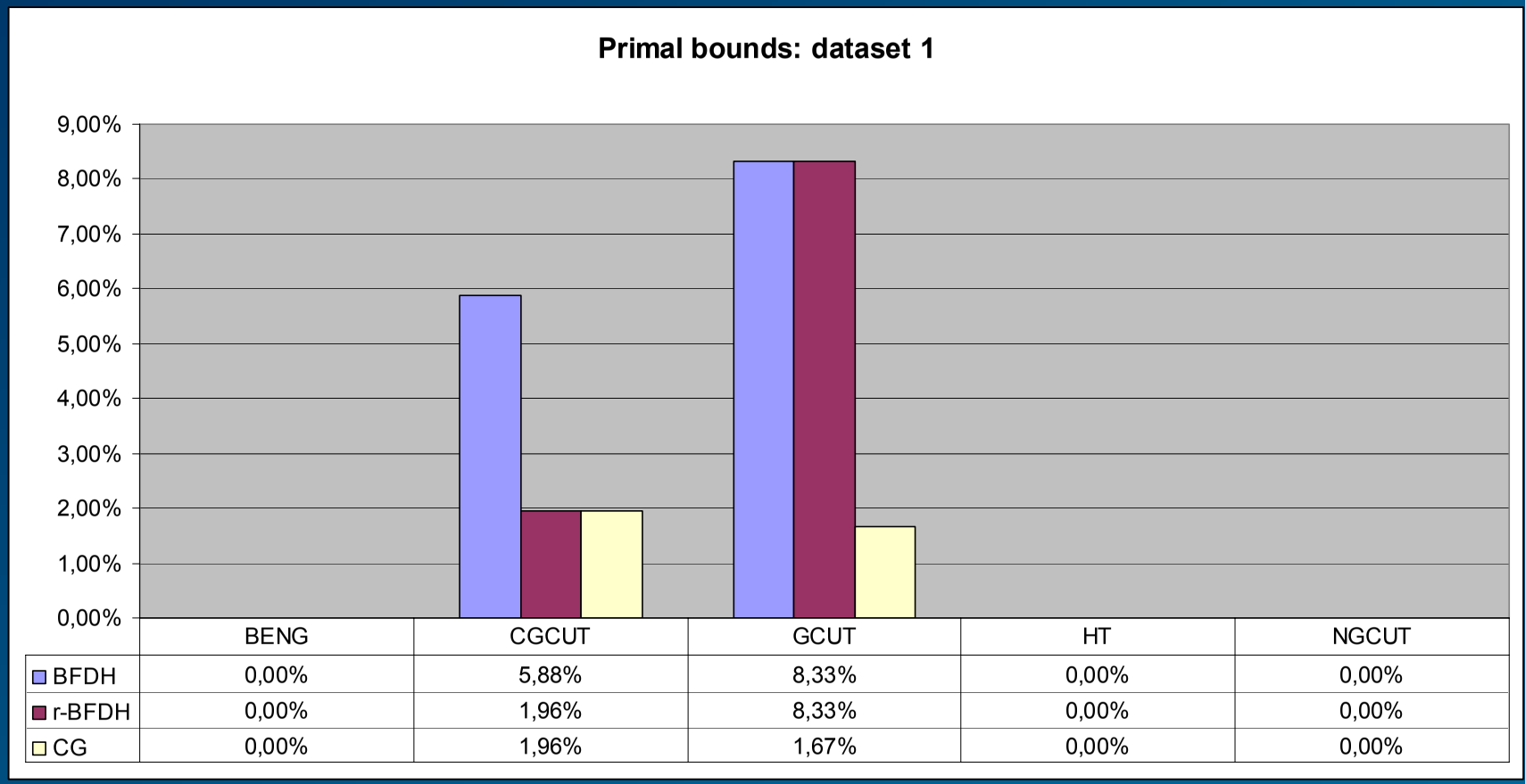
	BENG	CGCUT	GCUT	HT	NGCUT
W-bound	0,0%	0,0%	-5,0%	-2,8%	-18,5%
TB-bound	0,0%	0,0%	-5,0%	0,0%	-4,7%
LP-bound	0,0%	0,0%	-5,0%	-2,8%	-6,1%
CG-bound	0,0%	0,0%	0,0%	0,0%	0,0%
Combined	0,0%	0,0%	0,0%	0,0%	0,0%

Dual bounds



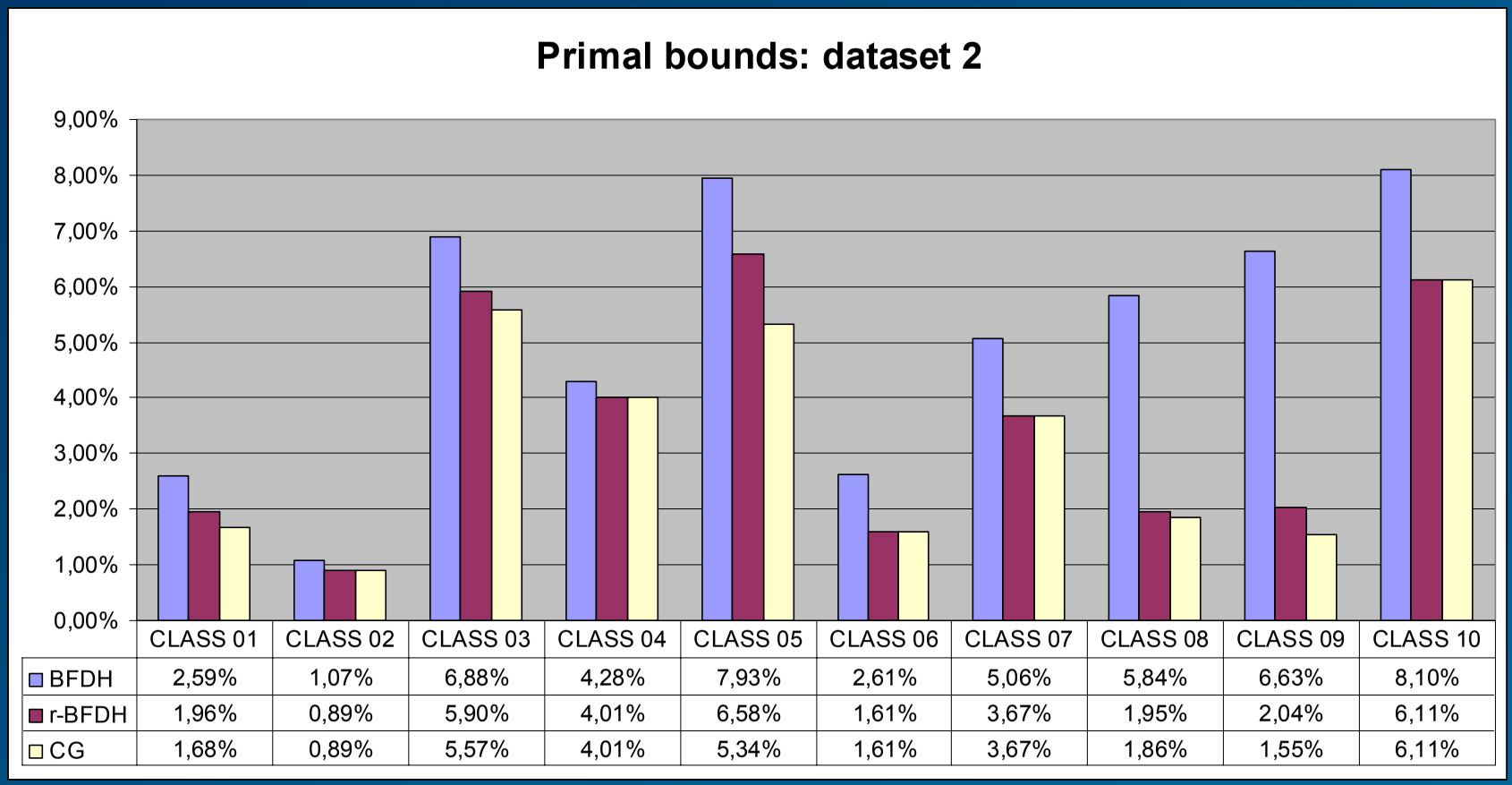
	CLASS 01	CLASS 02	CLASS 03	CLASS 04	CLASS 05	CLASS 06	CLASS 07	CLASS 08	CLASS 09	CLASS 10
W-bound	-3,92%	-1,94%	-2,71%	-1,39%	-3,37%	-1,39%	-9,96%	-11,16%	-13,06%	-1,93%
TB-bound	-2,95%	0,00%	-1,24%	0,00%	-1,39%	0,00%	-1,72%	-8,99%	-11,72%	-0,18%
LP	-2,95%	-0,13%	-1,40%	0,00%	-1,56%	0,00%	-2,42%	-9,37%	-11,78%	-0,18%
CG	0,00%	-0,13%	-0,11%	0,00%	-0,17%	0,00%	0,00%	0,00%	0,00%	0,00%
Combined	0,00%	0,00%	-0,11%	0,00%	-0,17%	0,00%	0,00%	0,00%	0,00%	0,00%

Experimental analysis: primal bounds



	BENG	CGCUT	GCUT	HT	NGCUT
BFDH	0,00%	5,88%	8,33%	0,00%	0,00%
r-BFDH	0,00%	1,96%	8,33%	0,00%	0,00%
CG	0,00%	1,96%	1,67%	0,00%	0,00%

Experimental analysis: primal bounds



	CLASS 01	CLASS 02	CLASS 03	CLASS 04	CLASS 05	CLASS 06	CLASS 07	CLASS 08	CLASS 09	CLASS 10
BFDH	2,59%	1,07%	6,88%	4,28%	7,93%	2,61%	5,06%	5,84%	6,63%	8,10%
r-BFDH	1,96%	0,89%	5,90%	4,01%	6,58%	1,61%	3,67%	1,95%	2,04%	6,11%
CG	1,68%	0,89%	5,57%	4,01%	5,34%	1,61%	3,67%	1,86%	1,55%	6,11%

Experimental analysis: exact solution

- Dataset 1 (38 inst.):
  - All instances solved within resource limitations
  - BENG instances: CPLEX 1.73s, B&P 0.08s, B&P with comb. bound 0.44s
  - GCUT04: CPLEX 403.09s, B&P 6.44s, B&P with comb. bound 0.74s

- Dataset 2 (500 inst.):
  - 420 'easy instances':
    - Solved in less than 10 minutes, about 12s on the average, for all methods
  - 80 'hard instances':
    - CPLEX
      - Solved 50, 16 out of time, 14 out of memory
      - Loose both primal and dual bounds
    - B&P
      - Solved 76, 4 out of memory
      - Dual bound always tight, loose primal bound
    - B&P with comb. bound
      - Very tight dual bounds
      - Too fractional solution: GAP feasibility hard to prove
  - Either CPLEX or B&P solved every instance

References

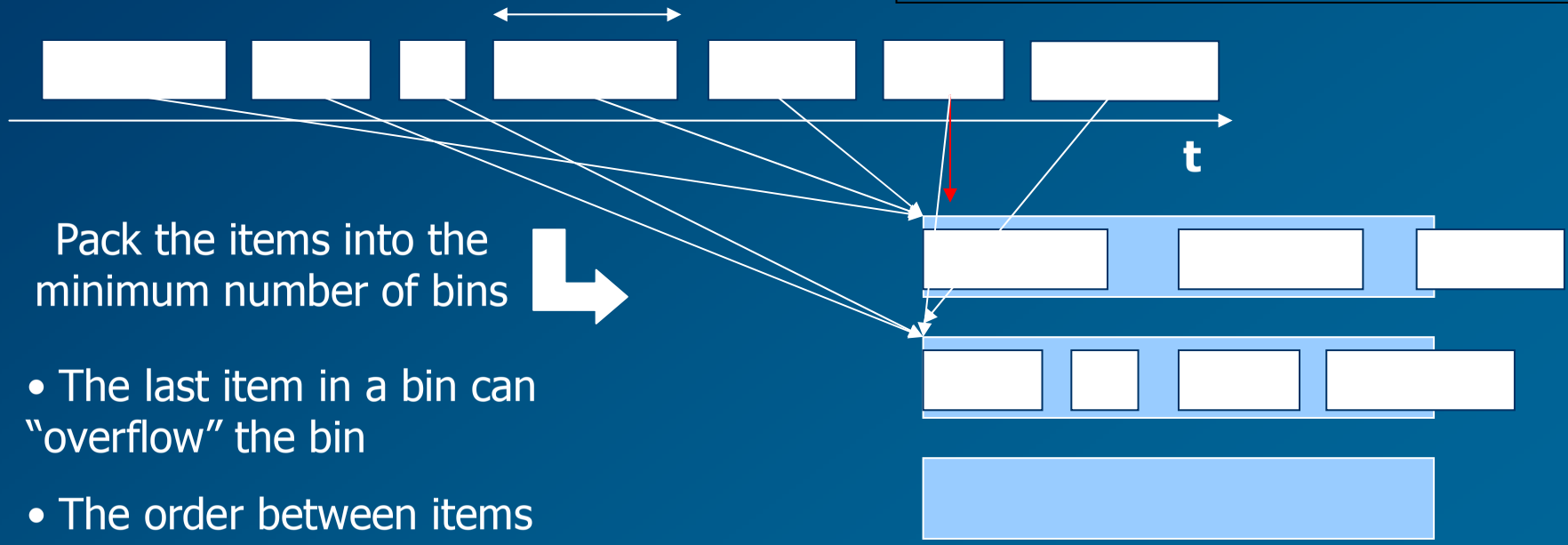
- Ceselli A., Righini G., "An optimization algorithm for a penalized knapsack problem", submitted for publication (2005).
- Lodi A., Martello S., Vigo D., "Models and bounds for two-dimensional level packing problems", Journal of Combinatorial Optimization 8:363-379 (2004).
- Yang J., Leung J. Y. T., "The ordered open-end bin-packing problem", Operations Research 51:759-770 (2003)

1 – The OOBP

Hong Kong subway:
items = travels, bins = tickets

Telephone cards:
items = calls, bins = cards

(1) a sequence of items (J), each of given width wj



Pack the items into the minimum number of bins

- The last item in a bin can "overflow" the bin
- The order between items is preserved
- (Each item cannot be split)

(2) a set of bins, each of the same width W