Exact Algorithms for the Vertex Separator Problem in Graphs

V. F. Cavalcante and C.C. de Souza

State University of Campinas, Brazil

victor.cavalcante@ic.unicamp.br

May 14, 2008

Outline

- ✓ Vertex Separation Problem (VSP)
- ✓ Related work
- ✓ Relax-and-cut algorithms
- ✓ Our Framework
- ✓ Computational Results
- ✓ Conclusions and future work

The Vertex Separator Problem (VSP)

- *A vertex separator in an undirected graph is a subset of the vertices, whose removal disconnects the graph in at least two nonempty connected components.*

The Vertex Separator Problem (VSP)

- *A vertex separator in an undirected graph is a subset of the vertices, whose removal disconnects the graph in at least two nonempty connected components.*

The vertex separator problem:

INSTANCE: *a connected undirected graph* $G=(V,E)$, *with* $|V|=n$, *a positive integer* $k \leq n$ *and a cost* c_i *associated with each vertex* $i \in V$.

PROBLEM: *find a partition of* V *into disjoint sets* A,B *and* C, *with* A *and* B *nonempty, such that:*

The Vertex Separator Problem (VSP)

- *A vertex separator in an undirected graph is a subset of the vertices, whose removal disconnects the graph in at least two nonempty connected components.*

The vertex separator problem:

INSTANCE: *a connected undirected graph* $G=(V,E)$, *with* $|V|=n$, *a positive integer* $k \leq n$ *and a cost* c_i *associated with each vertex* $i \in V$.

PROBLEM: *find a partition of* V *into disjoint sets* A,B *and* C, *with* A *and* B *nonempty, such that: (i) there is no edge* $(i,j) \in E$ *with* $i \in A$ *and* $j \in B$;

The Vertex Separator Problem (VSP)

- *A vertex separator in an undirected graph is a subset of the vertices, whose removal disconnects the graph in at least two nonempty connected components.*

The vertex separator problem:

INSTANCE: *a connected undirected graph* $G=(V,E)$, *with* $|V|=n$, *a positive integer* $k \leq n$ *and a cost* c_i *associated with each vertex* $i \in V$.

PROBLEM: *find a partition of* V *into disjoint sets* A, B *and* C, *with* A *and* B *nonempty, such that: (i) there is no edge* $(i,j) \in E$ *with* $i \in A$ *and* $j \in B$; *(ii)* $max\{|A|,|B|\} \leq k$

The Vertex Separator Problem (VSP)

- *A vertex separator in an undirected graph is a subset of the vertices, whose removal disconnects the graph in at least two nonempty connected components.*

The vertex separator problem:

INSTANCE: *a connected undirected graph* $G=(V,E)$, *with* $|V|=n$, *a positive integer* $k \leq n$ *and a cost* c_i *associated with each vertex* $i \in V$.

PROBLEM: *find a partition of* V *into disjoint sets* A,B *and* C, *with* A *and* B *nonempty, such that: (i) there is no edge* $(i,j) \in E$ *with* $i \in A$ *and* $j \in B$; *(ii)* $max\{|A|,|B|\} \leq k$ *and (iii) the cost of the separator* C *is given by* $\sum_{j \in C} c_j$ *is minimized.*

The Vertex Separator Problem (VSP)

- *A vertex separator in an undirected graph is a subset of the vertices, whose removal disconnects the graph in at least two nonempty connected components.*

The vertex separator problem:

INSTANCE: *a connected undirected graph* $G=(V,E)$, *with* $|V|=n$, *a positive integer* $k \leq n$ *and a cost* c_i *associated with each vertex* $i \in V$.

PROBLEM: *find a partition of* V *into disjoint sets* A,B *and* C, *with* A *and* B *nonempty, such that: (i) there is no edge* $(i,j) \in E$ *with* $i \in A$ *and* $j \in B$; *(ii)* $max\{|A|,|B|\} \leq k$ *and (iii) the cost of the separator* C *is given by* $\sum_{j \in C} c_j$ *is minimized.*

- *Complexity: $\mathcal{NP}$-hard.*
- *Applications: network connectivity and Linear Algebra ([Balas and de Souza, MP, 2005]).*

Related works and IP Formulation

- *[Balas and de Souza, MP, 2005]: integer programming formulation, polyhedral investigation.*
- *[de Souza and Balas, MP, 2005]: branch-and-cut algorithm.*

Related works and IP Formulation

- *[Balas and de Souza, MP, 2005]: integer programming formulation, polyhedral investigation.*
- *[de Souza and Balas, MP, 2005]: branch-and-cut algorithm.*
- *Formulation: for every* $i \in V$ *the binary variables* $u_{iA} = 1$ *if and only if* $i \in A$ *and* $u_{iB} = 1$ *if and only if* $i \in B$*; and* $u_L(S) = \sum(u_{iL} : i \in S)$.

$$\max \sum_{i \in V} c_i(u_{iA} + u_{iB})$$

$$u_{iA} + u_{iB} \leq 1, \qquad \forall\, i \in V \qquad (1)$$

$$u_{iA} + u_{jB} \leq 1,\ u_{jA} + u_{iB} \leq 1, \qquad \forall\, (i,j) \in E \qquad (2)$$

$$u_A(V) \geq 1, \qquad (3)$$

$$u_B(V) \leq k, \qquad (4)$$

$$u_A(V) - u_B(V) \leq 0, \qquad (5)$$

Valid Inequalities

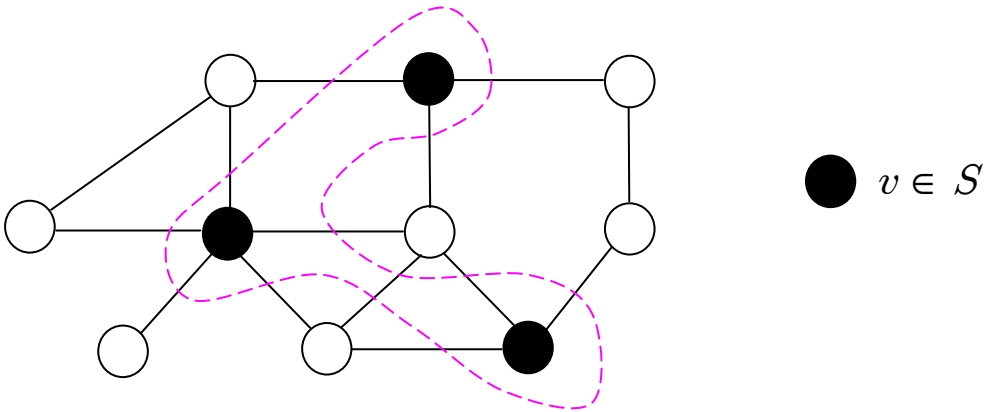
- *[Balas and de Souza, MP, 2005]:*
  - *CD inequalities: (minimal) Connected Dominators*
  - *LD inequalities: Lifting of (minimal) Dominators*

Valid Inequalities

- *[Balas and de Souza, MP, 2005]: CD and LD inequalities.*
- *A vertex dominator S is a subset of vertices of the graph such that all the remaining vertices are adjacent to at least one of them.*

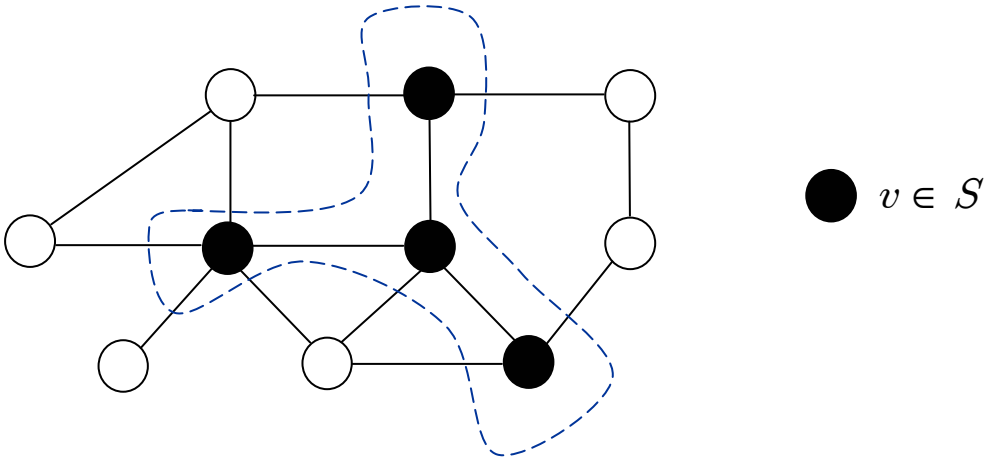
Valid Inequalities

- *[Balas and de Souza, MP, 2005]: CD and LD inequalities.*
- *A vertex dominator S is a subset of vertices of the graph such that all the remaining vertices are adjacent to at least one of them.*



Valid Inequalities

- *[Balas and de Souza, MP, 2005]: CD and LD inequalities.*
- *A vertex dominator S is a subset of vertices of the graph such that all the remaining vertices are adjacent to at least one of them.*
- *"Every separator and every connected dominator have at least one vertex in common".*
- *CD inequalities: if $S \subset V$ is a minimal connected dominator, the CD inequality for S is given by $u(S) = u_A(S) + u_B(S) \leq |S| - 1$.*



Branch-and-cut (B&C) for VSP

- *[de Souza and Balas, MP, 2005]: branch-and-cut algorithm developed based on CD and LD inequalities.*
- *Main experimental conclusions:*
  - *CD inequalities:*
    - *Very helpful to tighten the formulation;*
    - *More suitable to high-density ($\geq 35\%$) graphs;*
    - ***Drawbacks:***
      - *Separation is expensive ($\mathcal{NP}$-hard);*
      - *Frenquently CD cuts are dense $\Rightarrow$ hard to LP solvers.*
  - *LD inequalities:*
    - *Only suitable to low-density ($<35\%$) graphs;*
    - ***Drawback:** separation is expensive ($\mathcal{NP}$-hard);*

Relax-and-cut (R&C) algorithms

- *Main idea: to incorporate strong cutting planes in a Lagrangian framework.*
- *When to use it: in hard combinatorial optimization problems for which exponentially-sized families of strong valid inequalities are known.*
- *Aim: strengthen (Lagrangian relaxation) dual bounds.*

Relax-and-cut (R&C) algorithms

- *Main idea: to incorporate strong cutting planes in a Lagrangian framework.*
- *When to use it: in hard combinatorial optimization problems for which exponentially-sized families of strong valid inequalities are known.*
- *Aim: strengthen (Lagrangian relaxation) dual bounds.*
- *The Lagrangian relaxation scheme:*

$$IP \qquad z = max\left\{cx \,\middle|\, Ax \leq b,\ Dx \leq d\ , x \in \mathbb{B}_+^n\right\}$$

Relax-and-cut (R&C) algorithms

- *Main idea: to incorporate strong cutting planes in a Lagrangian framework.*
- *When to use it: in hard combinatorial optimization problems for which exponentially-sized families of strong valid inequalities are known.*
- *Aim: strengthen (Lagrangian relaxation) dual bounds.*
- *The Lagrangian relaxation scheme:*

$$IP \qquad z = max\left\{cx \,\middle|\, Ax \leq b,\ Dx \leq d\ , x \in \mathbb{B}_+^n\right\}$$

Dualized constraints

Relax-and-cut (R&C) algorithms

- *Main idea: to incorporate strong cutting planes in a Lagrangian framework.*
- *When to use it: in hard combinatorial optimization problems for which exponentially-sized families of strong valid inequalities are known.*
- *Aim: strengthen (Lagrangian relaxation) dual bounds.*
- *The Lagrangian relaxation scheme:*

$$IP \qquad z = max\left\{cx \,\middle|\, Ax \leq b,\ Dx \leq d\ , x \in \mathbb{B}_+^n\right\}$$

- *Lagrangian subproblem (with parameter* $\lambda \in \mathbb{R}_+^m$ *):*

$$IP(\lambda) \qquad z(\lambda) = max\left\{cx + \lambda(d - Dx) \,\middle|\, Ax \leq b,\ x \in \mathbb{B}_+^n\right\}$$

Relax-and-cut (R&C) algorithms

- *Main idea: to incorporate strong cutting planes in a Lagrangian framework.*
- *When to use it: in hard combinatorial optimization problems for which exponentially-sized families of strong valid inequalities are known.*
- *Aim: strengthen (Lagrangian relaxation) dual bounds.*
- *The Lagrangian relaxation scheme:*

$$IP \qquad z = max\{cx \,|\, Ax \leq b,\ Dx \leq d\ ,x \in \mathbb{B}^n_+\}$$

- *Lagrangian subproblem (with parameter* $\lambda \in \mathbb{R}^m_+$ *):*

$$IP(\lambda) \qquad z(\lambda) = max\{cx + \lambda(d - Dx) \,|\, Ax \leq b,\ x \in \mathbb{B}^n_+\}$$

$$+$$

$$\text{strong inequalities} \quad \alpha x \leq \beta \begin{cases} \alpha^1 x \leq \beta^1 & (\text{nondualized cuts}) \\ \alpha^2 x \leq \beta^2 & (\text{dualized cuts}) \end{cases}$$

Relax-and-cut (R&C) algorithms

- *Main idea: to incorporate strong cutting planes in a Lagrangian framework.*
- *When to use it: in hard combinatorial optimization problems for which exponentially-sized families of strong valid inequalities are known.*
- *Aim: strengthen (Lagrangian relaxation) dual bounds.*
- *The Lagrangian relaxation scheme:*

$$IP \qquad z = max\{cx \,|\, Ax \leq b,\ Dx \leq d\ , x \in \mathbb{B}_+^n\}$$

- *Lagrangian subproblem (with parameters* $\lambda, \mu \in \mathbb{R}_+^m$ *):*

$$IP(\lambda,\mu) \quad z(\lambda,\mu) = max\{\bar{c}x + k \,|\, Ax \leq b\ , x \in \mathbb{B}_+^n\}$$

$$where \quad \begin{cases} \bar{c} = \left(c - \lambda D - \mu\alpha^2\right) \\ k = \lambda d + \mu\beta^2 \end{cases}$$

- *Lagrangian Dual Problem:*

$$LDP(\lambda,\mu) = min\{z(\lambda,\mu) \,|\, \lambda,\mu \geq 0\}$$

Relax-and-cut (R&C) algorithms

- *R&C algorithms generally apply Subgradient Method (SM).*

Relax-and-cut (R&C) algorithms

- *R&C algorithms generally apply Subgradient Method (SM).*
- *Once the Lagrangian subproblem $z(\lambda, \mu)$ is computed, an integral solution x^* is available.*

  The separation problem for x^* has to be solved and the family of strong inequalities identified, $\alpha x \leq \beta$, is updated:

Relax-and-cut (R&C) algorithms

- *R&C algorithms generally apply Subgradient Method (SM).*
- *Once the Lagrangian subproblem $z(\lambda, \mu)$ is computed, an integral solution x^* is available.*

  The separation problem for x^* has to be solved and the family of strong inequalities identified, $\alpha x \leq \beta$, is updated:

  - *If a violated inequality $\pi x \leq \pi_0$ is found that is not yet among the dualized cuts, the following updates take place:*

$$\left(\alpha^2 x \leq \beta^2\right) \leftarrow \left(\alpha^2 x \leq \beta^2\right) \cup \{\pi x \leq \pi_0\}$$

$$\left(\alpha^1 x \leq \beta^1\right) \leftarrow \left(\alpha^1 x \leq \beta^1\right) \setminus \{\pi x \leq \pi_0\}$$

$$\Downarrow$$

A violated inequality is dualized as soon as it is encountered, at the end of the current SM iteration. This strategy is known as non-delayed R&C.

Relax-and-cut (R&C) algorithms

- *When SM is executed (LDP is solved) several times and the cuts found are dualized only at the end of each SM execution, the strategy is named buffered non-delayed R&C.*
- *If the separation is performed only when the LDP is solved (i.e., once at each SM execution) the strategy is called delayed R&C.*
- *[Lucena, AOR, 2005] discusses alternatives to embed cuttting planes in a Lagrangian framework.*

Relax-and-cut (R&C) for VSP

- *Dualized Constraints:*

$$\max \sum_{i \in V} c_i (u_{iA} + u_{iB})$$

$$u_{iA} + u_{iB} \leq 1, \qquad \forall\ i \in V$$

$$u_{iA} + u_{jB} \leq 1,\ u_{jA} + u_{iB} \leq 1, \qquad \forall\ (i,j) \in E$$

$$u_A(V) \geq 1,$$

$$u_B(V) \leq k,$$

$$u_A(V) - u_B(V) \leq 0,$$

$$u_A,\ u_B \in \{0,1\}^{|V|}$$

Relax-and-cut (R&C) for VSP

- *Dualized Constraints:*

$$\max \sum_{i \in V} c_i (u_{iA} + u_{iB})$$

$$u_{iA} + u_{iB} \leq 1, \qquad \forall\ i \in V$$

$$u_{iA} + u_{jB} \leq 1,\ u_{jA} + u_{iB} \leq 1, \qquad \forall\ (i,j) \in E$$

$$u_A(V) \geq 1,$$

$$u_B(V) \leq k,$$

$$u_A(V) - u_B(V) \leq 0,$$

$$u_A,\ u_B \in \{0,1\}^{|V|}$$

Relax-and-cut (R&C) for VSP

- *Lagrangian subproblem:*

$$\max \sum_{i \in V} (\bar{c}_{iA} u_{iA} + \bar{c}_{iB} u_{iB})$$

$$u_A(V) \geq 1,$$

$$u_B(V) \leq k,$$

$$u_A(V) - u_B(V) \leq 0,$$

$$u_A,\ u_B \in \{0,1\}^{|V|}$$

Solved in $O(|V| \log |V|)$ time by sorting the variables according to their Lagrangian costs and after performing a few simple calculations.

Relax-and-cut (R&C) for VSP

- *Our relax-and-cut algorithms use CD and/or LD inequalities as cutting planes.*

Relax-and-cut (R&C) for VSP

- *Our relax-and-cut algorithms use CD and/or LD inequalities as cutting planes.*

Separation of CD inequalities:

Let $\overline{u} = (\overline{u}_A, \overline{u}_B)$ *the optimal solution of the current Lagrangian subproblem.*

Relax-and-cut (R&C) for VSP

- *Our relax-and-cut algorithms use CD and/or LD inequalities as cutting planes.*

Separation of CD inequalities:

Let $\bar{u} = (\bar{u}_A, \bar{u}_B)$ *the optimal solution of the current Lagrangian subproblem.*

- *Define* $G_{\bar{u}} = (W, F)$ *as the subgraph induced by the subset* $i \in W \subseteq V$ *of vertices with* $\bar{u}_{iA} + \bar{u}_{iB} \geq 1$.

Relax-and-cut (R&C) for VSP

- *Our relax-and-cut algorithms use CD and/or LD inequalities as cutting planes.*

Separation of CD inequalities:

Let $\overline{u} = (\overline{u}_A, \overline{u}_B)$ *the optimal solution of the current Lagrangian subproblem.*

- *Define* $G_{\overline{u}} = (W, F)$ *as the subgraph induced by the subset* $i \in W \subseteq V$ *of vertices with* $\overline{u}_{iA} + \overline{u}_{iB} \geq 1$.
- *If W is a dominator and* $G_{\overline{u}}$ *is connected then there exists a CD inequality violated by* $\overline{u}$.

Relax-and-cut (R&C) for VSP

- *Our relax-and-cut algorithms use CD and/or LD inequalities as cutting planes.*

Separation of CD inequalities:

Let $\overline{u} = (\overline{u}_A, \overline{u}_B)$ *the optimal solution of the current Lagrangian subproblem.*

- *Define* $G_{\overline{u}} = (W, F)$ *as the subgraph induced by the subset* $i \in W \subseteq V$ *of vertices with* $\overline{u}_{iA} + \overline{u}_{iB} \geq 1$.
- *If W is a dominator and* $G_{\overline{u}}$ *is connected then there exists a CD inequality violated by* $\overline{u}$.
- *The converse is not true, but holds when* $u_{iA} + u_{iB} \leq 1, \forall i \in V$ *is imposed.*
- *Our separation routine is a heuristic.*

  Note: if $u_{iA} + u_{iB} \leq 1, \forall i \in V$ *are present, the Lagrangian subproblem can also be computed in polynomial time via Dynamic Programming.*

Relax-and-cut (R&C) for VSP

Separation of CD inequalities:

CD-Separation(G)

1. Construct $G_{\bar{u}} = (W, F)$;
2. Determine n_{CC}, the number of connected components of $G_{\bar{u}}$;
3. **if** $n_{CC} = 1$ **then** /* $G_{\bar{u}}$ is connected */
4. **if** $V \subseteq (W \cup \text{Adj}(W))$ **then** /* W is a dominator of V */
5. Turn W into a minimal CD;
6. **return** the CD inequality $u(W) \leq |W| - 1$;
7. **else return FAIL**; /* no new cut is returned for dualization */

- *Complexity:* $O(|V| + |E|)$.

Relax-and-cut (R&C) for VSP

Separation of CD inequalities:

CD-Separation(G)

1. Construct $G_{\bar{u}} = (W, F)$;
2. Determine n_{CC}, the number of connected components of $G_{\bar{u}}$;
3. **if** $n_{CC} = 1$ **then** /* $G_{\bar{u}}$ is connected */
4. **if** $V \subseteq (W \cup \text{Adj}(W))$ **then** /* W is a dominator of V */
5. Turn W into a minimal CD;
6. **return** the CD inequality $u(W) \leq |W| - 1$;
7. **else return FAIL**; /* no new cut is returned for dualization */

- *Complexity*: $O(|V| + |E|)$

for unitary costs and a given lower bound z_{LB}, if W induces a connected subgraph that cover $\geq |V| - max\{z_{LB} - k, 0\}$ vertices, then the conditional (CD) cut $u(W) \leq |W| - 1$ is valid for all solutions whose cost exceeds z_{LB}.

This fact is used in our routine to generate cuts with smaller supports.

Relax-and-cut (R&C) for VSP

Primal heuristic:

- *Construction phase*: *a greedy procedure that assigns vertices to shores A and B according to the Lagrangian costs of the associated variables and vertex degrees relative to A and B.*
- *Improvement phase*: *enlarges the separator C with the vertices of A and B that are in the adjacency of a vertex in C as long as A and B remain nonempty. Repeat the construction phase starting from this partial solution.*

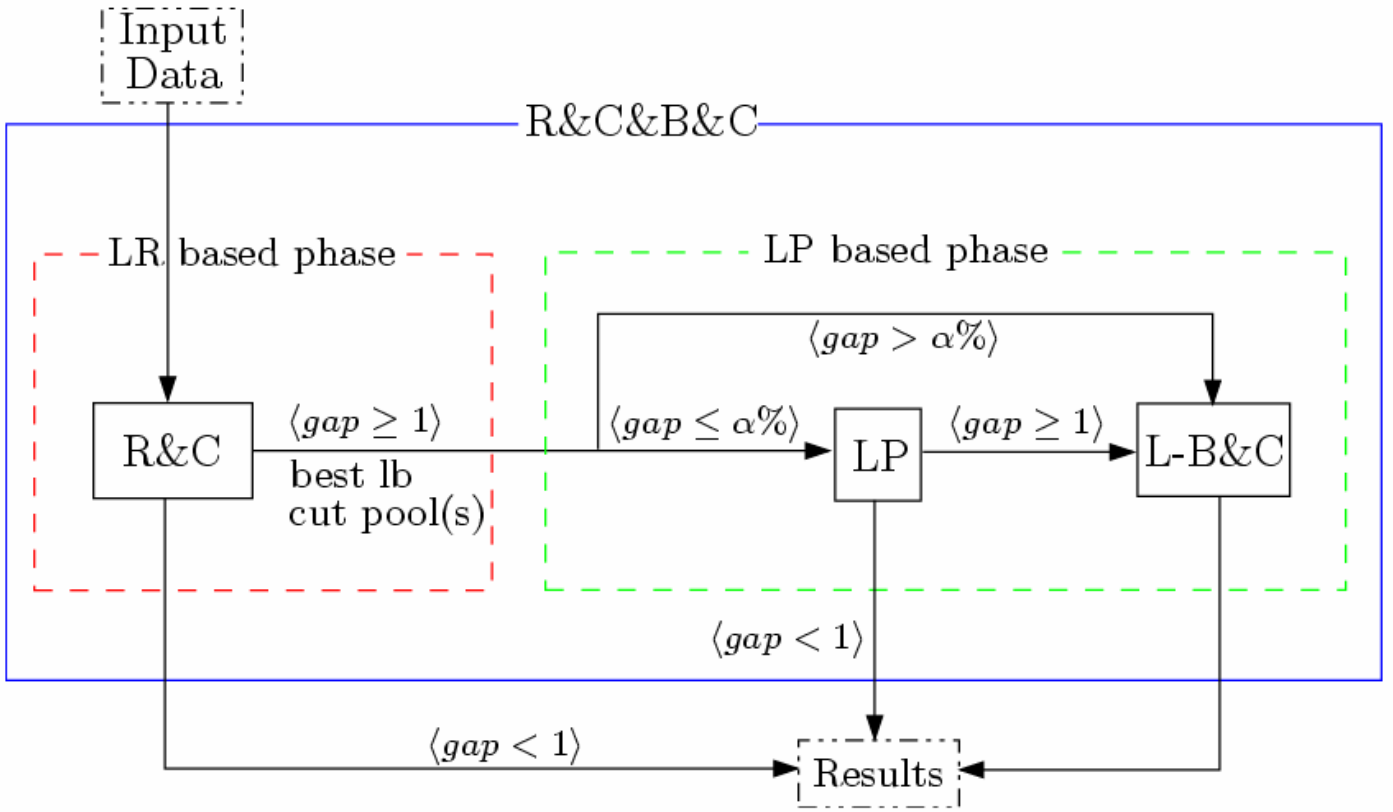
- *Complexity*: $O(|V| \log |V| + |E|)$, *including the improvement phase.*

Usage of the R&C algorithm

- *As a primal heuristic*
- *As a preprocessing phase for the B&C algorithm in [de Souza and Balas, MP, 2005]: LR/LP hybrid approach denoted R&C&B&C.*

Usage of the R&C algorithm

- *As a primal heuristic*
- *As a preprocessing phase for the B&C algorithm in [de Souza and Balas, MP, 2005]: LR/LP hybrid approach denoted R&C&B&C.*



R&C&B&C Framework: Flow Diagram

Computational Results

- *Computer environment:* Pentium IV, 1GB of RAM, Linux, C++, XPRESS Optimizer 17.01.02 (LP solver).
- *Data sets:*
  - subset of instances in www.ic.unicamp.br/cid/Problem-instances/VSP.html.
  - only instances not solved by XPRESS Optimizer with default settings ≤ 1 min.
  - Final tests with R&C&B&C framework:
    - 47 instances in total, all of which with unitary costs.
      - 38 graphs with density $>$ 20% (mid-high density instances);
      - 9 graphs with density $\leq$ 20% (low density instances);

Computational Results

- *Computer environment:* Pentium IV, 1GB of RAM, Linux, C++, XPRESS Optimizer 17.01.02 (LP solver).
- *Data sets:*
  - subset of instances in www.ic.unicamp.br/cid/Problem-instances/VSP.html.
  - only instances not solved by XPRESS Optimizer with default settings ≤ 1 min.
  - Final tests with R&C&B&C framework:
    - 47 instances in total, all of which with unitary costs.
      - 38 graphs with density $>$ 20% (mid-high density instances);
      - 9 graphs with density $\leq$ 20% (low density instances);
- *Assessment of the results (aspects considered):*
  - the quality of the primal bounds;
  - the effectiveness of using our approach as a preprocessing tool that provides B&C with a tighter formulation. Comparison of our best R&C&B&C algorithms:
    - with the best know algorithms to VSP;
    - with XPRESS solver;
    - among themselves.

Computational Results

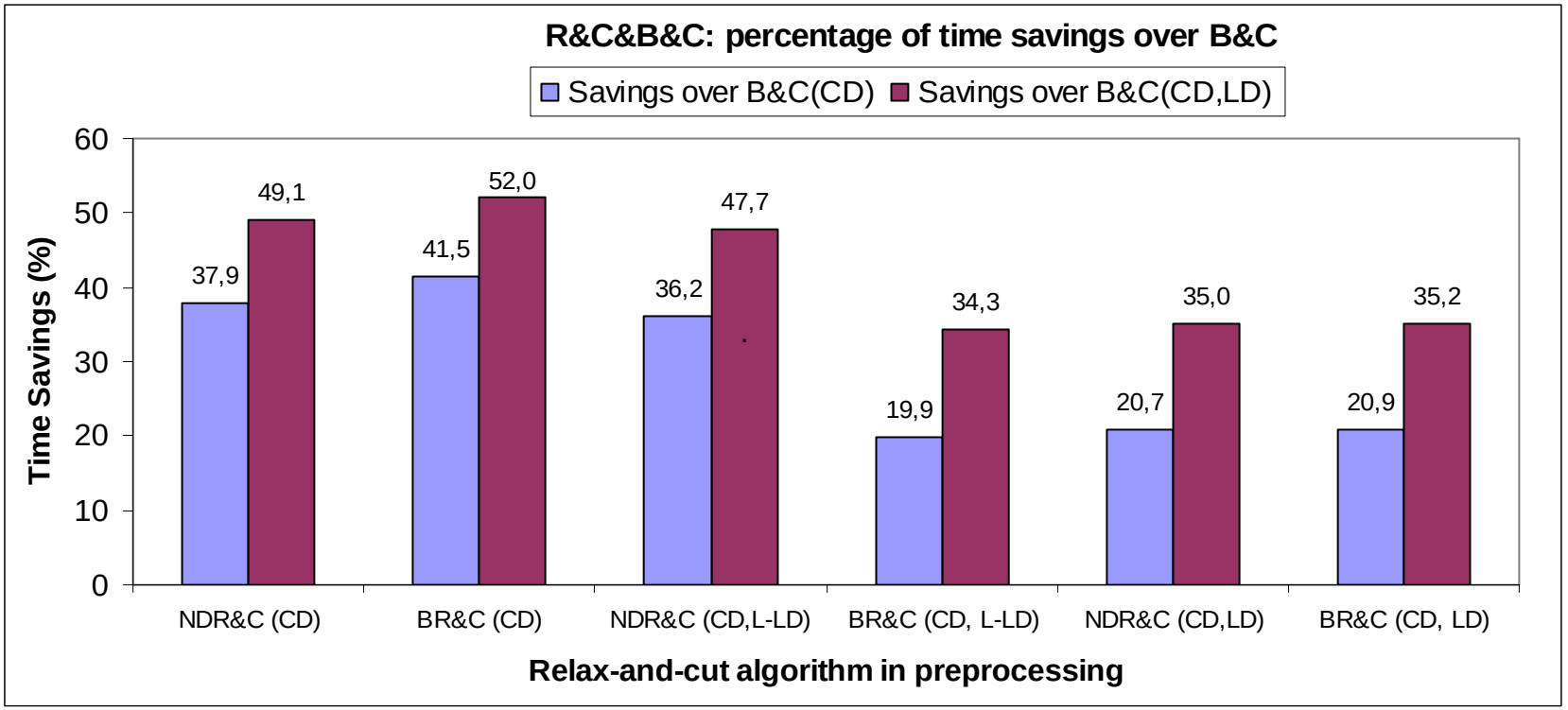
- *Computer environment:* Pentium IV, 1GB of RAM, Linux, C++, XPRESS Optimizer 17.01.02 (LP solver).
- *Data sets:*
  - subset of instances in www.ic.unicamp.br/cid/Problem-instances/VSP.html.
  - only instances not solved by XPRESS Optimizer with default settings ≤ 1 min.
  - final tests with R&C&B&C framework:
    - 47 instances in total, all of which with unitary costs.
      - 38 graphs with density $>$ 20% (mid-high density instances);
      - 9 graphs with density $\leq$ 20% (low density instances);
- *Assessment of the results (aspects considered):*
  - the quality of the primal bounds;
  - the effectiveness of using our approach as a preprocessing tool that provides B&C with a tighter formulation. Comparison of our best R&C&B&C algorithms:
    - with the best know algorithms to VSP;
    - with XPRESS solver;
    - among themselves.
- *Time limit: 30 min.*

Computational Results: mid-high density graphs

Mid-high density instance		Only CD Cuts				B&C [de Souza and Balas]				B&B (XP)	
		NDR&C&B&C		BR&C&B&C		B&C(CD)		B&C(CD,LD)			
label	$d(>20\%)$	nodes	t(s)	nodes	t(s)	nodes	t(s)	nodes	t(s)	nodes	t(s)
dim.queen8_8	0.36	1807	75.02	1707	90.22	4315	70.51	3143	58.27	23131	126.29
dim.miles1000	0.40	17	12.97	13	11.62	11	18.37	35	26.22	287	83.96
dim.queen7_7	0.40	391	14.11	313	16.73	431	10.77	265	9.93	27833	78.53
dim.DSJC125.9	0.90	4275	596.45	4143	537.01	33833	1107.29	28475	1291.54	51261	**1800.00**
mat.can73	0.25	5343	46.15	5505	50.57	5123	71.50	5615	46.44	33195	147.62
mat.lund_a	0.26	3145	401.39	2715	384.14	2231	462.03	2709	332.09	27506	**1800.00**
mat.L125.bcsstk05	0.35	709	85.65	-	4.53	1573	326.20	1625	196.01	1635	211.38
mat.L125.dwt_193	0.38	21	23.11	27	25.07	131	97.31	721	134.80	17767	1301.43
mat.L125.fs_183_1	0.44	29	28.11	21	26.16	25	35.45	29	31.74	1515	182.78
mat.bcsstk04	0.68	13	22.90	-	4.35	133	124.60	247	132.27	16572	**1800.00**
mat.arc130	0.93	83	160.19	83	163.80	101	370.67	101	329.59	957	926.12
mat.L100.steam2	0.36	45	21.02	41	20.67	149	40.83	241	37.98	11577	229.98
mat.L120.fidap025	0.39	-	2.50	-	2.59	13	12.00	49	17.67	889	107.73
mat.L120.cavity01	0.42	13	9.13	15	10.50	13	16.88	41	21.52	813	91.69
mat.L120.fidap021	0.43	5	5.63	7	5.88	35	24.74	67	36.63	1031	150.22
mat.L120.rbs480a	0.46	125	64.01	141	67.75	367	218.67	3007	249.27	15619	1308.79
mat.L120.wm2	0.47	33	28.04	35	36.68	33	47.82	33	45.71	351	88.71
mat.L100.rbs480a	0.52	59	11.90	67	15.60	63	21.33	91	21.34	2951	189.73
mat.L80.wm2	0.58	9	3.28	11	4.31	13	4.90	15	5.65	379	67.22
mat.L100.wm3	0.59	11	7.63	13	10.48	17	13.26	15	13.34	379	65.70
mat.L120.e05r0000	0.59	5	7.05	3	7.63	9	11.49	43	25.31	2703	543.05
mat.L100.wm1	0.60	19	10.74	13	9.18	25	15.67	35	24.36	877	94.07
mat.L120.fidap022	0.60	77	22.53	17	13.75	53	38.17	81	53.04	13319	1522.86
mat.L120.fidap001	0.63	-	4.94	-	5.07	31	32.57	189	84.70	33120	**1800.00**
mat.L100.e05r0000	0.64	15	8.39	17	8.17	19	11.19	39	12.49	3559	284.25
mat.L120.fidapm02	0.65	-	2.87	-	2.91	17	24.52	57	55.66	4457	552.37
mat.L100.fidap001	0.68	35	7.10	35	7.54	49	15.96	73	23.21	34321	950.38
mat.L100.fidap022	0.68	109	22.66	105	22.36	171	52.20	93	27.31	57415	1594.48
mat.L80.fidap001	0.72	-	1.55	-	1.55	1	1.76	33	5.20	3523	101.25
mat.L80.fidap022	0.76	197	14.25	159	11.90	173	15.28	45	5.51	19279	308.05
mat.L100.fidap002	0.82	5	3.22	3	3.52	7	4.75	29	10.19	2111	240.58
mat.L120.fidap002	0.82	5	6.88	1	5.73	73	41.59	93	48.59	10415	1284.46
miplib.khb05250.p	0.27	119	3.94	121	5.26	91	3.21	111	4.84	3641	66.37
miplib.l152lav.p	0.40	213	46.27	185	54.66	283	70.12	853	101.98	22885	567.68
miplib.lp4l.p	0.46	275	34.27	271	33.10	551	50.10	5965	151.72	27523	409.73
miplib.air03.p	0.61	115	111.94	117	119.98	135	167.35	135	180.01	14215	**1800.00**
miplib.misc03.p	0.63	2993	111.95	2717	121.67	4819	138.07	3947	155.22	53417	1794.42
miplib.misc07.p	0.80	173	1280.36	177	1519.38	125	**1800.00**	78	**1800.00**	2243	**1800.00**
$\sum$ L-B&Cs		20480	**3320.10**	18789	**3442.02**	-	-	-	-	-	-
$\sum$ B&Cs		20315	**2039.74**	18621	**1922.64**	47706	**3286.19**	58345	**4007.35**	-	-
$\sum$ B&Cs and XP		15912	**1303.51**	14361	**1256.23**	13574	**1854.38**	29299	**2318.83**	427260	**17471.88**

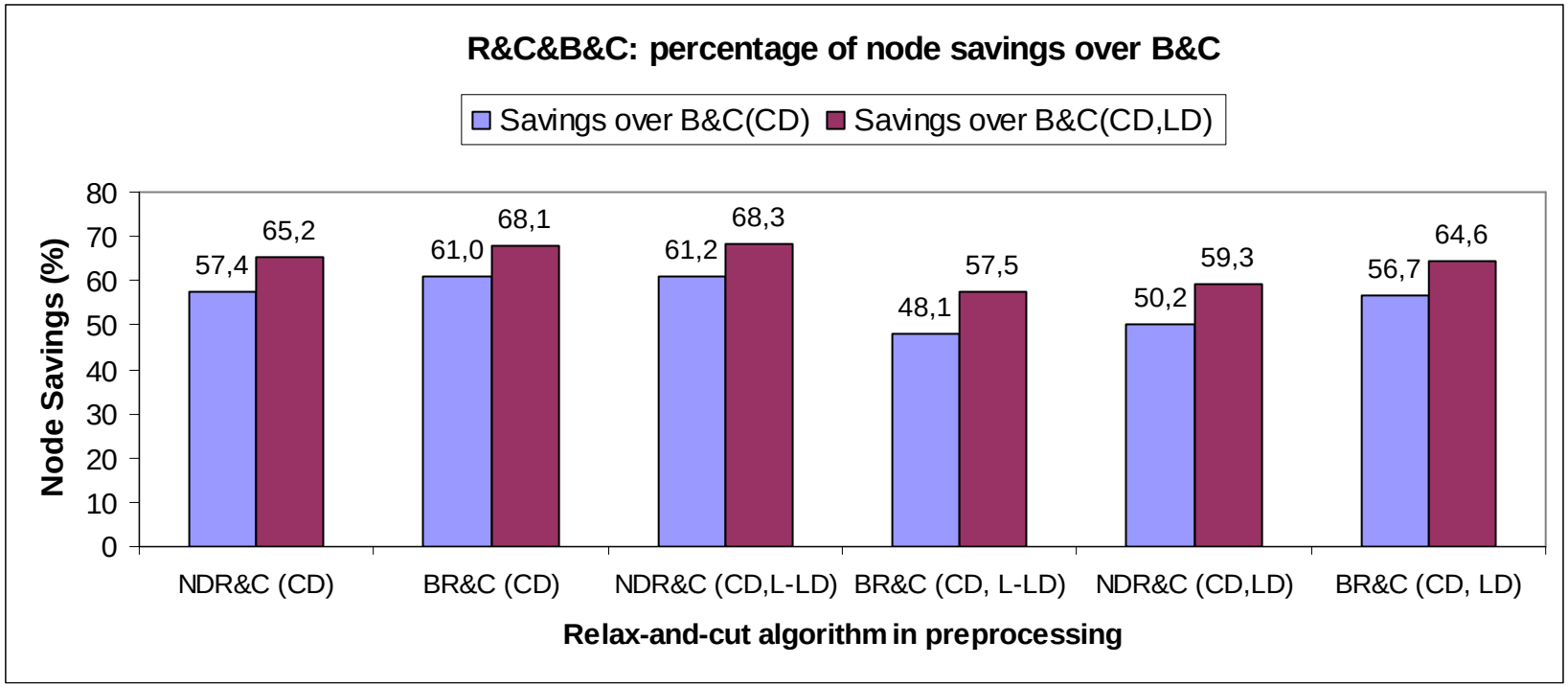
Computational Results: mid-high density graphs

- *Total time savings when compared to [de Souza and Balas, MP, 2005] to solve the group of instances solved by all the B&C based algorithms.*



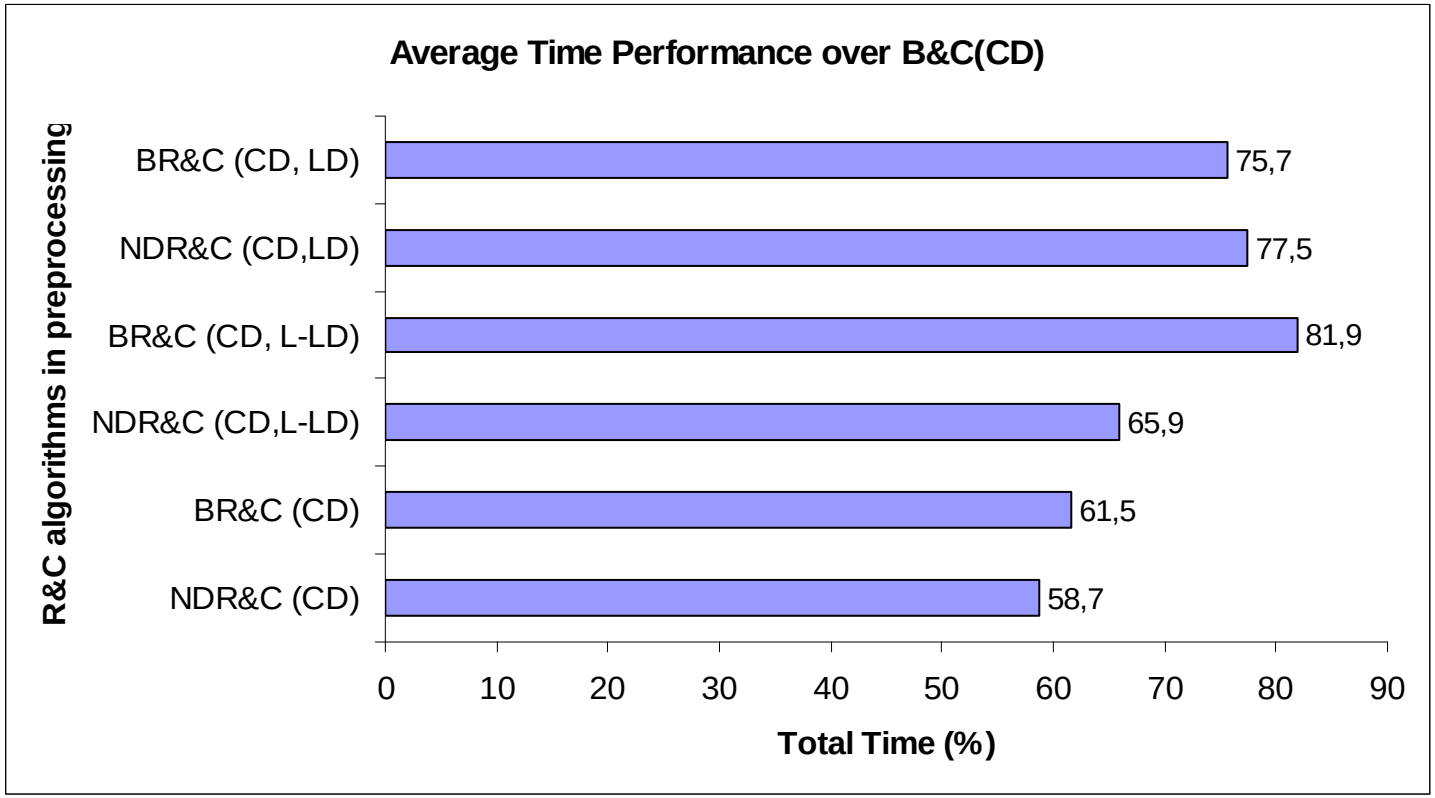
Computational Results: mid-high density graphs

- *Savings relative to the total number of nodes generated by the B&B search trees constructed by each algorithm.*



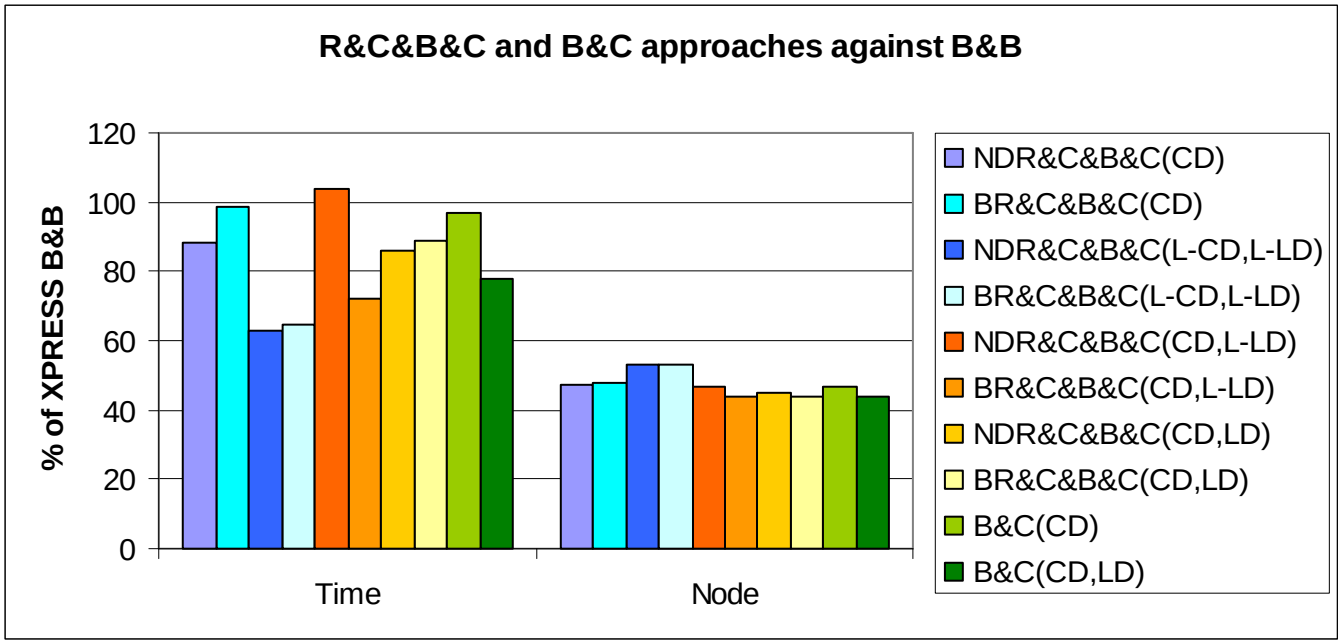
Computational Results: mid-high density graphs

- *Average percentage of B&C(CD) time needed by each R&C&B&C version to solve a VSP instance.*



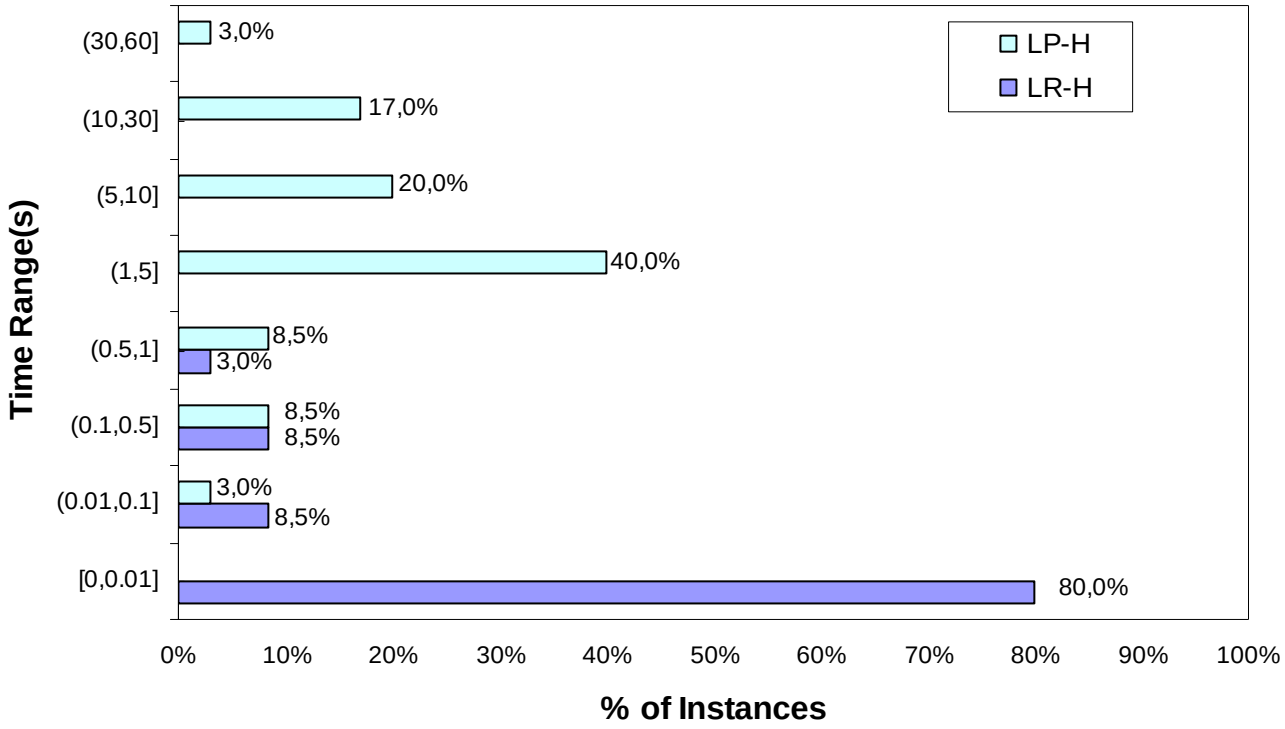
Computational Results: low density graphs

- *Percentage of B&B time and number of nodes needed by R&C&B&C B&C versions to solve all theVSP low density instances.*



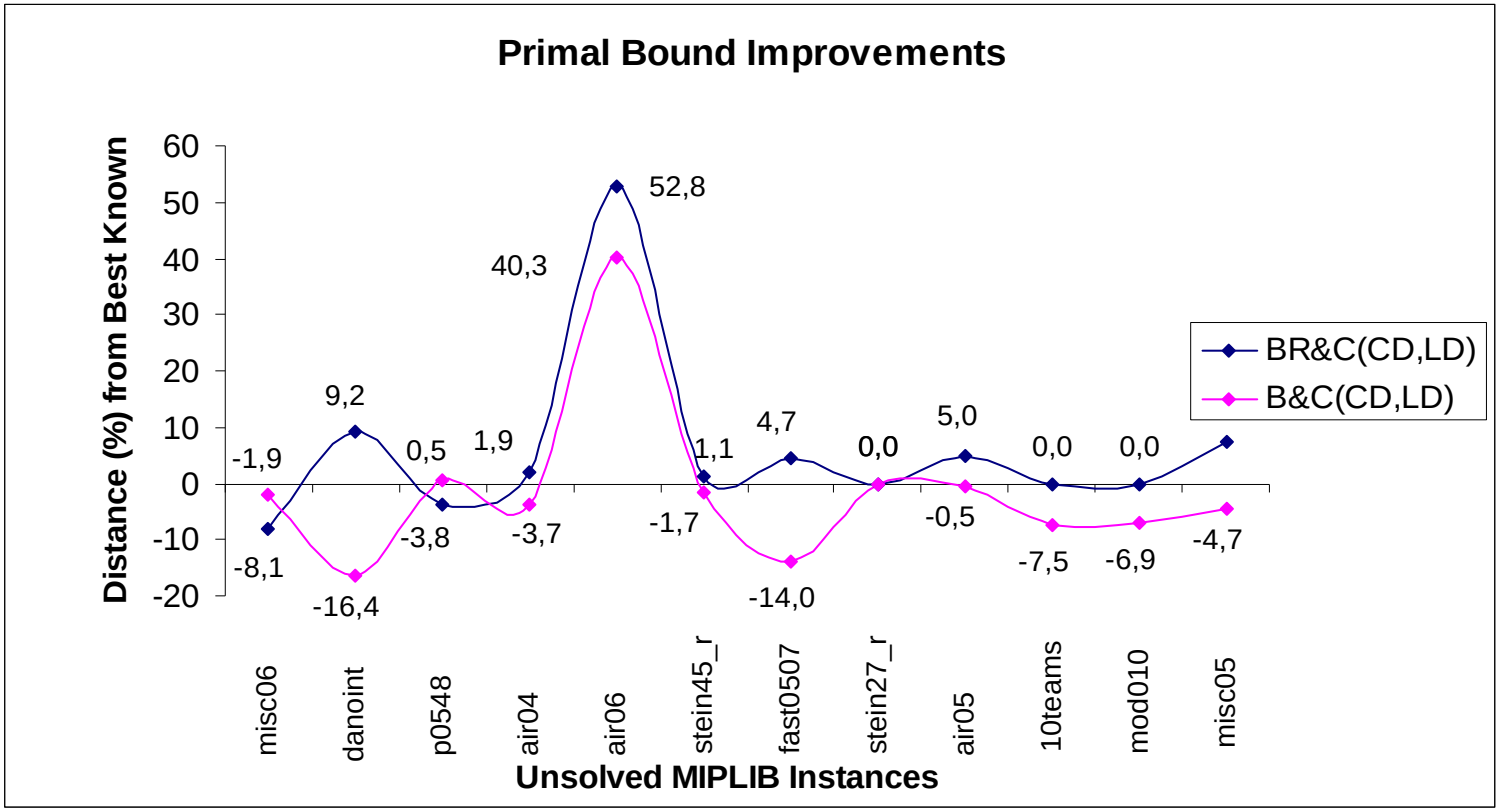
Computational Results: primal bounds

- *In about 70% of the cases an optimal solution was produced during the LR phase (LP-H).*



Computational Results: primal bounds

- *To MIPLIB hard to solve instances [Borndorfer et al., SIAM Journal on Opt., 1998].*



Conclusions and future work

- *Preprocessing with R&C accelerates the B&C of [de Souza and Balas, MP, 2005]: our framework improves the best know approach.*
- *In practice, the embedding of CD cuts in SM often decreases the dual bounds to values that are smaller than the trivial bound arising from linear relaxation.*

  Note: this is expected from the theory however convergence problems may occur!
- *The Lagrangian heuristic produces very good primal bounds*

Conclusions and future work

- *Preprocessing with R&C accelerates the B&C of [de Souza and Balas, MP, 2005]: our framework improves the best know approach.*
- *In practice, the embedding of CD cuts in SM often decreases the dual bounds to values that are smaller than the trivial bound arising from linear relaxation.*

  Note: this is expected from the theory however convergence problems may occur!
- *The Lagrangian heuristic produces very good primal bounds.*
- *Other cuts? Better tuning? Improve primal heuristic?*

References

- *E. Balas and C.C.de Souza.*
  The vertex separator problem: a polyhedral investigation.
  Mathematical Programming, 103:583-608, 2005.
- *C.C.de Souza and E. Balas.*
  The vertex separator problem: algorithms and computations.
  Mathematical Programming, 103:609-631, 2005.
- *A. Lucena.*
  Non delayed relax-and-cut algorithms.
  Annals of Operations Research, 140(1):375-410, 2005.
- *V.F. Cavalcante, C.C.de Souza and A. Lucena*
  A Relax-and-Cut algorithm for the set partitioning problem.
  Computers & Operations Research, 35(6):1963-1981, 2008.

References

- *E. Balas and C.C.de Souza.*
  The vertex separator problem: a polyhedral investigation.
  Mathematical Programming, 103:583-608, 2005.
- *C.C.de Souza and E. Balas.*
  The vertex separator problem: algorithms and computations.
  Mathematical Programming, 103:609-631, 2005.
- *A. Lucena.*
  Non delayed relax-and-cut algorithms.
  Annals of Operations Research, 140(1):375-410, 2005.
- *V.F. Cavalcante, C.C.de Souza and A. Lucena*
  A Relax-and-Cut algorithm for the set partitioning problem.
  Computers & Operations Research, 35(6):1963-1981, 2008.

Thanks for your attention!