Noname manuscript No.
(will be inserted by the editor)

From the separation to the intersection sub-problem in standard column generation

Daniel Porumbel

the date of receipt and acceptance should be inserted later

Abstract The Column Generation (CG) method can be seen as a Cutting Planes algorithm working on the dual of the master Linear Program (LP). It proceeds by progressively removing infeasibility in the dual space using the separation sub-problem. We have proposed in [11] the generalization of the separation sub-problem to the intersection sub-problem: given any point $\mathbf{y}$ in the dual space, find the maximum t^* such that $t^*\mathbf{y}$ is dual-feasible. In [11], this sub-problem was only used inside the *Integer Ray Method*. The goal of this short note is to present the advantage of the intersection sub-problem in a pure CG framework: we consider a canonical CG algorithm in which we use the intersection sub-problem to derive dual-feasible solutions. This is the main advantage of the intersection sub-problem: it can be used to derive a dual (lower) bound at each iteration. Using this lower bounding approach, we propose a more general proof of the Farley (Lagrangian) lower bound. Compared to the Lagrangian proof, the new proof is more general in the sense that it also produces a dual-feasible solution and not only a bound value. Then, we present a procedure that transforms a primal heuristic solution into a dual-feasible solution; for the graph coloring problem, this procedure can be easily implemented because it does not even need to explicitly solve any LP. Finally, we extend the Farley bound to the case of columns with different objective coefficients in the primal formulation. Numerical results are presented on the graph coloring problem and on the capacitated arc routing problem.

1 Introduction

Column Generation (CG) is a well established technique for optimizing linear programs with prohibitively many columns and variables. Such programs

D. Porumbel
CEDRIC, Conservatoire National des Arts et Métiers, 292, rue Saint Martin, Paris
daniel.porumbel@cnam.fr

arise very often in operations research; the columns can represent cutting patterns (*e.g.*, in cutting stock and bin packing [7,3]), paths/routes (in routing problems [4,8]), subsets of clients (in p-median [9,14]), etc—see more examples and references in the introduction of [17,4] or in the book [5]. A classical CG process starts out with a subset of columns and it iteratively adds (only) necessary columns by solving a pricing (dual separation) sub-problem.

We recall the Column Generation model associated to Set Covering problems. Without loss of generality, we consider a minimization problem described (after linear relaxation) by the following primal-dual LPs.

$$\mathbf{y}:\ \begin{array}{ll} \min \sum c_a x_a & \\ \sum a_i x_a \geq b_i, & \forall i \in [1..n] \\ x_a \geq 0 & \forall [c_a, \mathbf{a}] \in \mathcal{A} \end{array} \quad (1.1a) \qquad \begin{array}{l} \max \mathbf{b}^\top \mathbf{y} \\ \mathbf{x}: \mathbf{a}^\top \mathbf{y} \leq c_a,\ \forall [c_a, \mathbf{a}] \in \mathcal{A} \\ \mathbf{y} \geq \mathbf{0}_n \end{array}, \quad (1.1b)$$

where all the sums are carried out over all columns $[c_a, \mathbf{a}] \in \mathcal{A}$. We do not formally impose any condition on the size of the column set $\mathcal{A}$, but we consider that the listing of all columns is computationally very exhausting, if not impossible. As such, a CG algorithm generates only relevant columns by iteratively solving a pricing (dual separation) sub-problem. Given current $\mathbf{y}$, this sub-problem asks to minimize $c_a - \mathbf{a}^\top \mathbf{y}$ over all $[c_a, \mathbf{a}] \in \mathcal{A}$; if the result is less than zero, a new primal column (dual constraint) has to be inserted.

The main disadvantage of a CG algorithm is that it can require too many iterations, *i.e.*, as [2] put it, a CG process can be "desperately slow". Lower bounds can then be used for tail-cutting, ensuring an earlier termination.

1.1 Overview of the Farley Lagrangian bound

The classical approach to derive lower (dual) bounds in CG relies on the Lagrangian relaxation. We here briefly recall this approach. Using Lagrangian multipliers $\mathbf{y} \geq \mathbf{0}_n$ to relax the primal constraints $\sum \mathbf{a} x_a \geq \mathbf{b}$, the primal objective function becomes $\min\limits_{\mathbf{x} \geq 0} \sum c_a x_a - \mathbf{y}^\top (\sum \mathbf{a}^\top x_a - \mathbf{b})$. After several algebraic reformulations (see work such as [11, Appendix C], [1, § 2.2], [16, §. 3.2], [10, §. 2.1] or [2, § 1.2]), the Lagrangian bound can be written as:

$$\mathbf{b}^\top \mathbf{y} + u_b \cdot m_{rdc}, \tag{1.2}$$

where $m_{rdc} = \min_{[c_a, \mathbf{a}] \in \mathcal{A}} \left(c_a - \mathbf{a}^\top \mathbf{y} \right)$ is the minimum reduced cost with regards to multipliers $\mathbf{y}$, and $u_b \geq \sum x_a$ is an upper bound of the sum of all primal variables $\sum x_a$.

We say that the optimum of the CG model OPT_{CG} satisfies $\mathrm{OPT}_{CG} \geq b^\top \mathbf{y} + u_b \cdot m_{rdc}$ for any multipliers $\mathbf{y} \geq \mathbf{0}_n$. In case $c_a = 1, \forall [c_a, \mathbf{a}] \in \mathcal{A}$, the objective of (1.1a) is already a sum of variables ; we can thus insert $\sum x_a = \sum c_a x_a \leq \mathrm{OPT}_{CG}$ in all primal formulations. As such, we can use OPT_{CG} instead of u_b as an upper bound over the sum of variables $\sum x_a$. This yields

$\mathrm{OPT}_{CG} \geq \mathbf{b}^\top \mathbf{y} + \mathrm{OPT}_{CG} \cdot m_{rdc}$ which directly leads to the Farley bound [6]:

$$\mathrm{OPT}_{CG} \geq \frac{\mathbf{b}^\top \mathbf{y}}{1 - m_{rdc}}. \tag{1.3}$$

Notice that $1 - m_{rdc} = 1 - \min_{[c_a, \mathbf{a}] \in \mathcal{A}} \left(1 - \mathbf{a}^\top \mathbf{y}\right) = -\min_{[c_a, \mathbf{a}] \in \mathcal{A}} -\mathbf{a}^\top \mathbf{y} = \max_{[c_a, \mathbf{a}] \in \mathcal{A}} \mathbf{a}^\top \mathbf{y}$, and so, the Farley bound is equivalent to:

$$\mathrm{OPT}_{CG} \geq \frac{\mathbf{b}^\top \mathbf{y}}{\max_{[c_a, \mathbf{a}] \in \mathcal{A}} \mathbf{a}^\top \mathbf{y}}. \tag{1.4}$$

1.2 Overview of the new lower bounding approach

This paper presents a new approach for deriving lower bounds, without using the Lagrangian relaxation. It is based on projecting towards dual solutions in the dual space using the *intersection subproblem*. Generally speaking, the intersection sub-problem along $\mathbf{0}_n \to \mathbf{y}$ requires finding the maximum t^* such that $t^* \cdot \mathbf{y}$ is dual feasible (see also Figure 1). We will show in Section 2 that the objective value of $t^* \cdot \mathbf{y}$ is equal to the value of the Farley bound from (1.3)-(1.4) above. Besides reporting this bound value, the advantage of the intersection sub-problem is that it also generates the dual-feasible solution $t^* \cdot \mathbf{y}$.

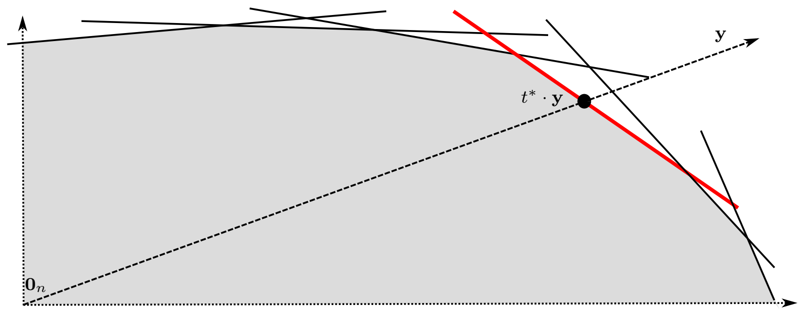

Fig. 1: An intuitive illustration of a ray projection. Besides finding a first-hit constraint (in red) that separates $\mathbf{y}$, the intersection sub-problem also produces dual feasible solution $t^* \cdot \mathbf{y}$.

We will present experimental results on the intersection sub-problem for graph coloring and for the capacitated arc routing problem. In both cases, we produce a vector $\mathbf{y}$ using a CG model and we solve the intersection sub-problem along $\mathbf{0}_n \to \mathbf{y}$. This leads to a lower bound value and to a dual-feasible solution. Unlike the Farley bound, the new lower bounding approach also works on CG models with different objective coefficients in the primal formulation.

The rest of the paper is organized as follows. Section 2 is devoted to the new proof of the Farley Lagrangian bound (Section 2.1) and to a generalization to the case of columns with different objective coefficients in the primal formulation (Section 2.2). Section 3 presents numerical results, followed by conclusions in the last section.

2 Generalizing a theoretical result on the Farley bound

To ease the exposition, we only deal with the case $c_a \geq 0$ and $\mathbf{a} \geq \mathbf{0}_n$ $\forall [c_a, \mathbf{a}] \in \mathcal{A}$. Under this condition, the solution $\mathbf{0}_n$ is dual feasible in (1.1b).

Definition 1 Given any $\mathbf{y} \in \mathbb{R}^n_+$, *the intersection subproblem* along ray $\mathbf{0}_n \to \mathbf{y}$ asks to find the maximum step length $t^* \geq 0$ such that $t^* \cdot \mathbf{y}$ is (dual) feasible in (1.1b). This maximum is found by minimizing the fraction $\frac{c_a}{\mathbf{a}^\top \mathbf{y}}$ over all columns $[c_a, \mathbf{a}] \in \mathcal{A}$ that verify $\mathbf{a}^\top \mathbf{y} > 0$.

A short proof and related discussions on the above min-max equivalence can be found in Proposition 3 of [11, §3.2]. The condition $\mathbf{a}^\top \mathbf{y} > 0$ is necessary to avoid zero denominators in the minimized fraction. For instance, if $\mathbf{y}$ has only one non-zero component $y_i \neq 0$ for some $i \in \{1, 2, \ldots, n\}$, than there are many columns $[c_a, \mathbf{a}] \in \mathcal{A}$ with $a_i = 0$ that yield $\mathbf{a}^\top \mathbf{y} = 0$.

2.1 The new proof of the Farley bound

We show that the Farley bound can be proved without using the Lagrangian relaxation, by projecting towards dual solution $\mathbf{y}$. Indeed, by projecting in the dual space along any $\mathbf{0}_n \to \mathbf{y}$, one obtains a dual feasible solution $t^* \cdot \mathbf{y}$ of objective value $t^* \cdot \mathbf{b}^\top \mathbf{y}$. When $c_a = 1$ $\forall [c_a, \mathbf{a}] \in \mathcal{A}$, the intersection subproblem asks to find t^* by minimizing $\frac{c_a}{\mathbf{a}^\top \mathbf{y}} = \frac{1}{\mathbf{a}^\top \mathbf{y}}$ over all $[c_a, \mathbf{a}] \in \mathcal{A}$. This requires maximizing $\max\limits_{[c_a, \mathbf{a}] \in \mathcal{A}} \mathbf{a}^\top \mathbf{y}$, and so, we obtain $t^* = \frac{1}{\max\limits_{[c_a, \mathbf{a}] \in \mathcal{A}} \mathbf{a}^\top \mathbf{y}}$. The lower bound value $t^* \cdot \mathbf{b}^\top \mathbf{y}$ becomes $\frac{\mathbf{b}^\top \mathbf{y}}{\max\limits_{[c_a, \mathbf{a}] \in \mathcal{A}} \mathbf{a}^\top \mathbf{y}}$, equivalent to (1.4).

The above proof of the Farley bound is not only shorter than the Lagrangian-based proof, but also more general in the sense that it produces a dual feasible solution $\frac{\mathbf{y}}{\max_{[c_a, \mathbf{a}] \in \mathcal{A}} \mathbf{a}^\top \mathbf{y}}$. The availability of a dual feasible solution can be potentially useful for stabilization reasons in CG.

2.2 The case of columns with different objective coefficients c_a

Using the intersection sub-problem, we can further generalize the Farley bound to address problems in which the values c_a are not all equal to 1. To determine this generalized bound, it is enough to calculate t^* and report bound $t^* \cdot \mathbf{b}^\top \mathbf{y}$. Compared to the case $c_a = 1 \forall a \in \mathcal{A}$, the only difference is that t^* can no longer be determined by maximizing $\max_{[c_a,\mathbf{a}]\in\mathcal{A}} \mathbf{a}^\top\mathbf{y}$. This leads to a slightly modified lower bounding procedure. After calculating

$$t^* = \min_{\substack{[c_a,\mathbf{a}]\in\mathcal{A}\\ \mathbf{a}^\top\mathbf{y}>0}} \frac{c_a}{\mathbf{a}^\top\mathbf{y}} \tag{2.1}$$

we obtain the lower bound $t^*\cdot\mathbf{b}^\top\mathbf{y}$, that can be written:

$$\min_{\substack{[c_a,\mathbf{a}]\in\mathcal{A}\\ \mathbf{a}^\top\mathbf{y}>0}} \frac{c_a}{\mathbf{a}^\top\mathbf{y}}\cdot\mathbf{b}^\top\mathbf{y} \tag{2.2}$$

While the separation sub-problem requires minimizing a difference of the form $c_a-\mathbf{a}^\top\mathbf{y}$, the intersection sub-problem requires minimizing a ratio $c_a/\mathbf{a}^\top\mathbf{y}$ as in (2.1)-(2.2) above. When dynamic programming is used, the intersection sub-problem might have the same computational difficulty as the classical separation sub-problem. This is true especially if the states can be indexed by integer $\mathbf{y}$ values [11, §4.2.3].

3 Numerical Experiments

3.1 Graph Coloring

We here present a procedure that takes as input an upper bound of the chromatic number χ of a graph and returns a lower bound and a dual-feasible solution. The graph coloring problem fits well the model (1.1a)-(1.1b): each stable represents a column with cost $c_a = 1$. Notice there is a prohibitively large number of columns corresponding to all stable sets of the graph. For each vertex $i\in\{1,2,\ldots,n\}$, there is a dual variable y_i associated to a covering constraint with $b_i = 1$.

We use the following experimental protocol. Given a k-coloring found by a heuristic [15], we take each of the k constituent stables and distributes uniformly a value of 1 to all its vertices, *i.e.*, each vertex of stable V_j receives the dual value $\frac{1}{|V_j|}$ for all $j\in\{1,2,\ldots k\}$. Given the dual values $\mathbf{y}$ obtained this way, we observe that $\mathbf{b}^\top\mathbf{y}=\mathbf{1}_n^\top\mathbf{y}=k$. It is enough to solve the intersection sub-problem along $\mathbf{0}_n\to\mathbf{y}$ to find lower bound $t^*\cdot\mathbf{b}^\top\mathbf{y}=t^*k$.

To determine t^*, the intersection sub-problem asks to minimize $\frac{1}{\mathbf{a}^\top\mathbf{y}}$ over all columns $\mathbf{a}$ associated to stables, or, equivalently, to maximize $\mathbf{a}^\top\mathbf{y}$. This reduces to finding the maximum weighted stable $S_{\max}$ using weights $\mathbf{y}$. We obtain $t^*=\frac{1}{S_{\max}}$ and the resulting lower bound is $\frac{k}{S_{\max}}$; we also obtain the dual-feasible solution $\frac{\mathbf{y}}{S_{\max}}$. Notice one can use a relaxation (an upper bound) of the maximum weighted stable, *e.g.*, the fractional maximum weighted stable that can be more easily calculated. Indeed, we can write $\frac{k}{S_{\max}}\geq\frac{k}{S_{\max}^{rlxd}}$, where $S_{\max}^{rlxd}$ is the value of the fractional maximum weighted stable.

Table 1 presents graph coloring results using 4 columns: the instance name, the number of vertices, the upper bound k found by the heuristic, and the

Instance	n	Upper bound k	Lower bound
dsjc125.1	125	5	3.30
dsjc125.5	125	18	12.25
dsjc125.9	125	44	24
dsjc250.1	250	9	4.5*
dsjc250.5	250	28	19.58
dsjc250.9	250	73	49.21
dsjc500.1	500	12	4.4*
r125.1	125	5	2.31
r125.1c	125	46	25.09
r125.5	125	36	20.57
r250.1	250	8	3.31
r250.1c	250	64	36.57
r250.5	250	65	24.11*

Table 1: Graph coloring results. The values marked * correspond to lower bounds obtained using a relaxation (upper bound) of the maximum stable.

lower bound calculated with our intersection sub-problem. The results marked* represent lower bounds obtained using a relaxation of the maximum stable, *i.e.*, we stopped `Cplex` after 10 minutes and we took the best upper bound of the weighted stable.

In general, the upper bound $k \geq \chi$ leads to a lower bound between $\frac{k}{2}$ and $\frac{k}{3}$. While this bound is not very tight, one should be aware that it is obtained only by simply solving a maximum weighted stable problem. To solve this problem, one does not even need to explicitly use any LP, *i.e.*, one could use a purely combinatorial algorithm[1] to determine $S_{\max}$, or a meta-heuristic to find $S_{\max}^{rlxd}$.

3.2 Capacitated Arc Routing

We here illustrate the proposed lower bounding approach on the Capacitated Arc-Routing Problem (CARP), *i.e.*, edge-focused counterpart of the celebrated capacitated vehicle routing problem. The goal is to find a set of routes of total minimum cost that service a set of required edges. The problem fits well the model (1.1a)-(1.1b): a column $\mathbf{a}$ corresponds to the incidence vector associated to the required edges of a feasible route and the cost c_a is the total distance traversed by the route. As such, the objective function coefficients in the primal formulation differ from route to route. Notice there is a prohibitively-large number of columns corresponding to all feasible routes. For each required edge, there is a dual variable y_i associated to a covering constraint with $b_i = 1\ \forall i \in \{1, 2, \ldots, n\}$.

The experimental protocol consists of first generating n initial columns as follows: for each required edge e, we consider a route that only goes to e, services e and goes back to depot. Then, we generate $4{\cdot}n$ columns by calling $4{\cdot}n$

[1] Off the shelf software is available on the internet, *e.g.*, for example, the Cliquer (`users.aalto.fi/~pat/cliquer.html`) due to S. Niskanen and P. Östergård.

times the pricing algorithm presented in [12, §4] and we obtain the current dual solution $\mathbf{y}$. This is followed by a call to the intersection sub-problem towards a direction obtained by transforming $\mathbf{y}$ as described next.

First, to ease the intersection sub-problem, we need to use the model (4.8) from [11, §4.2.2]. The main particularity of this model is the use of a change of variable: we replace y_i with $y'_i + c_i$, where c_i is the length of edge i. As described in [11, §4.2.2], the cost of a route in the resulting model in variables $\mathbf{y}'$ is no longer the total travelled distance, but the total distance travelled without service (the sum of the lengths of the deadheaded edges). This is necessary to simplify the algorithm for the intersection sub-problem. The new dual variables y'_i have negative lower bounds, *i.e.*, we no longer have $\mathbf{y} \geq \mathbf{0}_n$, but $y'_i \geq -c_i$.[2]

Secondly, recall [11, §4.2.3] that the intersection algorithm uses a dynamical programming scheme in which the states are indexed by *integer* ray coefficients. To run the intersection algorithm in a reasonable time, we have to transform y'_i into a (small) integer value. If $n > 50$, we divide each y'_i by 20 and round it to the nearest integer. If $n < 50$, we multiply each y'_i by a large integer factor[3] and round it to the nearest integer.

After solving the intersection sub-problem, we obtain a lower bound via (2.2), more easily than by applying the Lagrangean relaxation. We recall that the Lagrangian bound (1.2) would require the availability of an upper bound u_b over the sum of variables $\sum x_a$.

Table 2 presents the lower bounds obtained by our new method on different CARP instances. Column 1 is the instance name, Column 2 indicates the number of required edges, Column 3 shows the number of vertices, Column 4 indicates the optimum or the best known integer feasible solution, and Column 5 is the lower bound obtained with our new approach.

This table shows that the gap with regards to the best known integer solution (Column 4) varies from less than 15% (*e.g.*, instances `gdb2`, `gdb4`, `gdb5`, all `kshs` instances, `val3C`) to 50% (especially on larger `egl` instances). However, when comparing it to other bounds, one should take into account that the time required by our bound is minimal, *i.e.*, it only needs to run several CG iterations and to solve an intersection sub-problem. In many cases, this can require less than one second on a mainstream computer.

Finally, all results from this paper have been obtained using `C++` programs compiled by `gnu g++` with code optimization option `-O3`. The maximum weighted stable from Section 3.1 and the LPs of the CG algorithm from Section 3.2 were solved by `Cplex` 12.6. We used a mainstream Linux laptop (kernel version 3.16) with a CPU i7-5500U with two cores (4 threads) clocked at 2.40GHz.

---

[2] When solving the intersection sub-problem in this model, one has to pay attention that the optimal solution $t^* \cdot \mathbf{y}'$ (the hit point) can belong to a facet $y'_i \geq -c_i$.

[3] We tried both 1000 and $9 \cdot 5 \cdot 8 \cdot 11$.

Instance	n	$\lvert V\rvert$	OPT_{IP}(best*)	Lower bound
gdb1	22	12	316	283.33
gdb2	26	12	339	312.8
gdb3	22	12	275	237.53
gdb4	19	11	287	270
gdb5	26	13	377	358.5
gdb6	22	12	298	273.32
gdb7	22	12	325	274.89
gdb8	46	27	348	243.86
gdb9	51	27	303	257.6
gdb10	25	12	275	253.6
gdb11	45	22	395	356
gdb12	23	13	458	444.12
gdb13	28	10	536	509
gdb14	21	7	100	98.47
gdb15	21	7	58	56.5
gdb16	28	8	127	121.66
gdb17	28	8	91	84
gdb18	36	9	164	158
gdb19	11	8	55	51.8
gdb20	22	11	121	105
gdb21	33	11	156	151.41
gdb22	44	11	200	196.19
gdb23	55	11	233	223
kshs1	15	8	14661	13553
kshs2	15	10	9863	8387.54
kshs3	15	6	9320	8498
kshs4	15	8	11498	11296.33
kshs5	15	8	10957	10357.8
kshs6	15	9	10197	9210.7
val1C	39	24	245	203.35
val2C	39	24	457	263.99
val3C	35	24	138	123.41
val4C	69	41	530	343
val5C	65	34	575	367
val6C	50	31	317	257.14
egl-e1-C	51	77	5595	2836.864
egl-e2-C	72	77	8335	3797.222
egl-e3-C	87	77	10292*	5696.75
egl-e4-C	98	77	11562*	6423.769
egl-s1-C	75	140	8518	4312.842
egl-s2-C	147	140	16425	5336.901
egl-s3-C	159	140	17188	6025.717
egl-s4-C	190	140	20481*	4186

Table 2: Capacitated Arc Routing results

4 Conclusion and Perspectives

We presented an approach for calculating lower (dual) bounds in CG models using the intersection sub-problem. Compared to existing methods, the new approach has the following advantages:

- it provides a very short (the first paragraph of Section 2.1) and more general proof of the Farley's bound. Compared to the Lagrangean relaxation proof,

the new proof is more general in the sense that it can produce a dual feasible solution and not only a bound value;
- it allows one to transform a primal heuristic solution (an upper bound) into a dual-feasible solution (a lower bound), using a single call to the intersection sub-problem. For graph coloring, the intersection sub-problem reduces to calculating a maximum weighted stable (Section 3.1).
- by solving the intersection sub-problem, one can determine lower bounds (and dual-feasible solutions) for problems with columns having different objective coefficients in the primal formulation (Section 3.2).

The paper is a continuation of [11] and it is a part of a sequence of planned work concerning the intersection sub-problem in different mathematical programming fields (e.g., Benders decomposition [13], robust programs with prohibitively-many constraints).

References

1. H. Ben Amor and J. M. V. de Carvalho. Cutting stock problems. In Desaulniers et al. [5], pages 131–161. (ISBN 978-0-387-25485-2).
2. O. Briant, C. Lemaréchal, P. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical Programming*, 113(2):299–344, 2008.
3. F. Clautiaux, C. Alves, J. M. V. de Carvalho, and J. Rietz. New stabilization procedures for the cutting stock problem. *INFORMS Journal on Computing*, 23(4):530–545, 2011.
4. G. Desaulniers, J. Desrosiers, and M. M. Solomon. Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems. In *Essays and surveys in metaheuristics*, pages 309–324. Springer, 2002.
5. G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column generation*, volume 5. Springer Science & Business Media, 2005. (ISBN 978-0-387-25485-2).
6. A. A. Farley. A note on bounding a class of linear programming problems, including cutting stock problems. *Operations Research*, 38(5):922–923, 1990.
7. P. Gilmore and R. Gomory. A linear programming approach to the cutting stock problem. *Operations Research*, 9:849–859, 1961.
8. A. Letchford and A. Oukil. Exploiting sparsity in pricing routines for the capacitated arc routing problem. *Computers & Operations Research*, 36:2320–2327, 2009.
9. L. A. N. Lorena and E. L. F. Senne. A column generation approach to capacitated p-median problems. *Computers & Operations Research*, 31(6):863–876, 2004.
10. M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
11. D. Porumbel. Ray projection for optimizing polytopes with prohibitively many constraints in set-covering column generation. *Mathematical Programming*, 1(155):147–197, 2016.
12. D. Porumbel, G. Goncalves, H. Allaoui, and T. Hsu. Iterated local search and column generation to solve arc-routing as a permutation set-covering problem. *European Journal of Operational Research*, 256(2):349–367, 2017.
13. D. C. Porumbel. From separation to intersection subproblems for optimizing polytopes with prohibitively many constraints in a Benders decomposition context. *Submitted Paper*, 2017. Available online at `http://cedric.cnam.fr/~porumbed/papers/benders.pdf`.
14. D. C. Porumbel and G. Goncalves. Using dual feasible functions to construct fast lower bounds for routing and location problems. *Discrete Applied Mathematics*, 196:83–99, 2015.
15. D. C. Porumbel, J.-K. Hao, and P. Kuntz. A search space "cartography" for guiding graph coloring heuristics. *Computers & OR*, 37(4):769–778, 2010.

16. F. Vanderbeck. Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming*, 86(3):565–594, 1999.
17. F. Vanderbeck and L. A. Wolsey. An exact algorithm for IP column generation. *Operations Research Letters*, 19(4):151 – 159, 1996.