Separation of Generic Cutting Planes in Branch-and-Price Using a Basis

Marco E. Lübbecke and Jonas T. Witt(✉)

Operations Research, RWTH Aachen University,
Kackertstr. 7, 52072 Aachen, Germany
{marco.luebbecke,jonas.witt}@rwth-aachen.de

Abstract. Dantzig-Wolfe reformulation of a mixed integer program partially convexifies a subset of the constraints, i.e., it *implicitly* adds all valid inequalities for the associated integer hull. Projecting an optimal basic solution of the reformulation's LP relaxation to the original space does in general not yield a basic solution of the original LP relaxation. Cutting planes in the original problem that are separated using a basis like Gomory mixed integer cuts are therefore not directly applicable. Range [22] (and others) proposed as a remedy to heuristically compute a basic solution and separate this auxiliary solution also with cutting planes that stem from a basis. This might not only cut off the auxiliary solution, but also the solution we originally wanted to separate.

We discuss and extend Range's ideas to enhance the separation procedure. In particular, we present alternative heuristics and consider additional valid inequalities strengthening the original LP relaxation before separation. Our full implementation, which is the first of its kind, is done within the GCG framework. We evaluate the effects on several problem classes. Our experiments show that the separated cuts strengthen the formulation on instances where the integrality gap is not too small. This leads to a reduced number of nodes and reduced solution times.

1 Introduction

Branch-and-price has become a widely used technique for solving mixed integer programs (MIPs) with an embedded structure. The *original problem* is first reformulated using *Dantzig-Wolfe reformulation* and the reformulated problem is then solved with *branch-and-price* [12], where the *linear programming (LP) relaxation* is solved using *column generation* and specialized branching rules are applied. When additionaly cutting planes are separated, the algorithm is called *branch-price-and-cut* [12].

Most often implementations are tailored for particular problems with known structure that can be exploited, but in the last decade also generic implementations were developed [15,20,21,25]. Bergner et al. [3] provide a computational proof-of-concept that the automatic detection of a suitable structure can be successful even when considering general problems.

Among others, cutting planes formulated with original variables were studied in the branch-price-and-cut literature. Adding these cuts to the problem does

E. Bampis (Ed.): SEA 2015, LNCS 9125, pp. 110–121, 2015.
DOI: 10.1007/978-3-319-20086-6_9

not change the structure of the pricing problem, whereas other types of cuts do [11]. In several applications problem specific cuts formulated with original variables are separated. Combinatorial cuts exploiting a particular substructure can also be separated in a generic way [15]. Moreover, Range [22] introduced a procedure to separate cuts in the original problem using a basis, but he only did a preliminary computational study on elementary shortest path problems with resource constraints, which was not successful (personal communication, 2013).

Our Contribution. A separation procedure that generates cuts in the original problem using a basis was mentioned by many authors [12,14], but only Range [22] presented such a procedure without providing a computational study. We discuss Range's ideas and present some extensions to enhance the separation procedure. Furthermore, we implemented all ideas in the branch-price-and-cut solver `GCG` [15] and tested the implementation on instances of several problem classes. In particular, we computationally investigate the strength of the separated cutting planes and determine their influence on the overall solution process.

2 Dantzig-Wolfe Reformulation and Branch-and-Price

Let $n, m_1, m_2 \in \mathbb{Z}_{\geq 1}, q \in \mathbb{Z}_{\geq 0}$ be some integers, let $A \in \mathbb{Q}^{m_1 \times n}, D \in \mathbb{Q}^{m_2 \times n}$ be some matrices, and let $b \in \mathbb{Q}^{m_1}, d \in \mathbb{Q}^{m_2}, c \in \mathbb{Q}^n$ be some vectors. Suppose we are given the following *original problem*

$$\min\{c^T x : Ax \geq b, Dx \geq d, x \in \mathbb{Z}^{n-q} \times \mathbb{Q}^q\}$$

with mixed integer hull $P_{MIP} := \text{conv}(\{x \in \mathbb{Z}^{n-q} \times \mathbb{Q}^q : Ax \geq b, Dx \geq d\})$, where $\text{conv}(S)$ denotes the convex hull of a set S. We will refer to its LP relaxation as *original LP relaxation* and denote the polyhedron of LP-feasible solutions by $P_{LP} := \{x \in \mathbb{Q}^n : Ax \geq b, Dx \geq d\}$.

When reformulating the original problem using Dantzig-Wolfe reformulation for mixed integer programs [12], a part of the constraints, here $Dx \geq d$, is *convexified*. Every solution $x \in X := \{x \in \mathbb{Z}^{n-q} \times \mathbb{Q}^q : Dx \geq d\}$ is reformulated as a convex combination of extreme points $\{\boldsymbol{x}^p\}_{p \in P}$ plus a non-negative linear combination of extreme rays $\{\boldsymbol{x}^r\}_{r \in R}$ of the associated convex hull $\text{conv}(X)$:

$$\sum_{p \in P} \boldsymbol{x}^p \lambda_p + \sum_{r \in R} \boldsymbol{x}^r \lambda_r = x, \qquad \sum_{p \in P} \lambda_p = 1, \qquad \lambda_p \in \mathbb{Q}_{\geq 0} \quad \forall p \in P \cup R \ .$$

Replacing x by this combination while introducing new λ-variables results in an *extended formulation* called the *master problem*. The corresponding LP relaxation is called *linear master problem* and is solved with *column generation*, where a *pricing problem* over X is iteratively solved in order to generate columns/variables having negative reduced cost. This procedure embedded in a branch-and-bound tree is called *branch-and-price* [12].

It is known [26] that the optimal solution value of the linear master problem is equal to $\min\{c^T x : Ax \geq b, x \in \text{conv}(X)\}$, which corresponds to implicitly adding all valid inequalities for $\text{conv}(X)$ to the original LP relaxation.

3 Separation of Cutting Planes in Branch-and-Price

In each node of the branch-and-bound tree cutting planes can be added in order to strengthen the LP relaxation, which is then called *branch-price-and-cut* [12].

We can deal with valid inequalities formulated with original variables of the form $\pi^T x \geq \pi_0$, where $\pi \in \mathbb{Q}^n$ and $\pi_0 \in \mathbb{Q}$, in the same way as with constraints $Ax \geq b$. Therefore, adding these inequalities as cutting planes to the problem does not change the structure of the pricing problem, i.e., the set X is not affected. On the contrary, other types of cutting planes, e.g., cuts formulated with λ-variables that were introduced in the master problem, may change the structure of the pricing problem, which can hamper its computational tractability.

Let $\bar{\lambda}$ be an optimal *basic feasible solution* [4] of the linear master problem and suppose that the projection $\bar{x} := \sum_{p \in P} x^p \bar{\lambda}_p + \sum_{r \in R} x^r \bar{\lambda}_r$ onto the x-variables is not integer feasible for the original problem. We can try to separate $\bar{x}$ using problem specific cuts. Some of these cuts, e.g., knapsack or clique cuts, are implemented in state-of-the-art MIP solvers [1] and can automatically be applied in branch-price-and-cut algorithms if the original problem contains a particular substructure [15]. Additionally, there exist cuts that stem from a basis like Gomory mixed integer (GMI) cuts. These cuts are in general not directly applicable, because $\bar{x}$ does not have to be basic in the original LP relaxation [16], see Fig. 1.

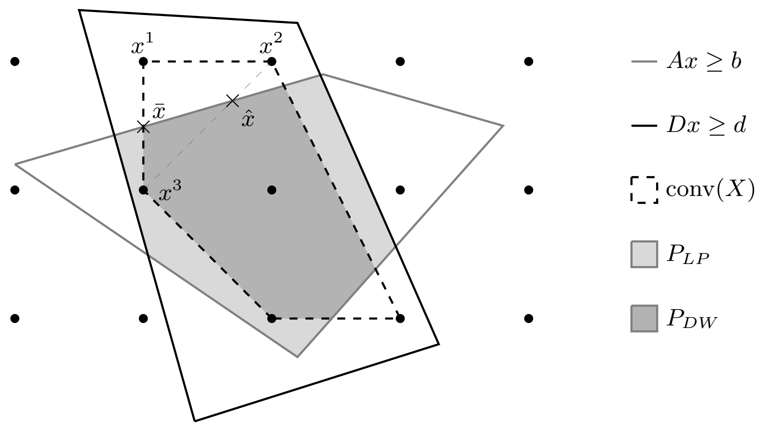

Fig. 1. Solution $\bar{x}$ is not a vertex of the polyhedron $P_{LP} = \{x : Ax \geq b, Dx \geq d\}$ and solution $\hat{x}$ is not even a vertex of $P_{DW} = \{x : Ax \geq b, x \in \text{conv}(X)\}$

In general we can check if the solution $\bar{x}$ is basic in the original LP relaxation by calculating the number of linear independent inequalities *active* at $\bar{x}$, i.e., satisfied with equality by $\bar{x}$. The solution $\bar{x}$ is basic if and only if this number is equal to the dimension n of the underlying vector space [4]. In case a description of conv(X) is known explicitly, Goncalves' criterion [16] can be applied. Rios and Ross [23] proved that if the pricing problem consists of affinely independent extreme points, Goncalves' criterion is satisfied.

Motivated by the fact that cuts obtained from a basis have in general a larger impact than combinatorial cuts [5], we would like to apply these cuts in the context of branch-price-and-cut, too.

4 Basis Separation

We recall that cutting planes in the original problem stemming from a basis are not directly applicable, because the projected solution $\bar{x}$ is in general not basic in the original LP relaxation. An idea to overcome this issue is to calculate some basic feasible solution x^* and separate x^*. Since x^* is basic feasible, cuts stemming from a basis can be applied. The obtained cuts might not only cut off the basic feasible solution x^*, but also the solution $\bar{x}$ that we wanted to cut off initially. If the solution $\bar{x}$ is not cut off, we can strengthen the original LP relaxation by temporarily adding the obtained cuts to the problem formulation and repeat the procedure. Since the solution x^* is not feasible for the strengthened original LP relaxation, we will calculate a different basic feasible solution that can be used for separation in the following iteration. The resulting generic algorithm is described as Algorithm 1 and was initially proposed by Range [22].

Data: P_{MIP}, P_{LP}, $\bar{x}$, $p_{\min}$, and $i_{\max}$.
Result: Feasible solution $x^* \in P_{MIP}$ or set of coefficients $\bar{\Pi} \subseteq \mathbb{Q}^{n+1}$ corresponding to cuts $\pi^T x \geq \pi_0$ with $(\pi, \pi_0) \in \bar{\Pi}$ separating $\bar{x}$.

$i := 0$, $\bar{\Pi} := \emptyset$, $\Pi^* := \emptyset$, $P'_{LP} := P_{LP}$;
while $|\bar{\Pi}| < p_{\min}$ *and* $i < i_{\max}$ **do**
  Calculate a vertex x^* of P'_{LP} (guided by $\bar{x}$);
  if $x^* \in P_{MIP}$ **then**
    return x^*;
  Separate the solution x^* from P_{MIP} and let Π^* be the set of coefficients corresponding to the generated cuts;
  if $\Pi^* = \emptyset$ **then**
    break;
  for $(\pi, \pi_0) \in \Pi^*$ **do**
    if $\pi^T \bar{x} < \pi_0$ **then**
      $\bar{\Pi} := \bar{\Pi} \cup \{(\pi, \pi_0)\}$;
  end for
  $P'_{LP} := P'_{LP} \cap \{x : \pi^T x \geq \pi_0, (\pi, \pi_0) \in \Pi^*\}$;
  $i := i + 1$;
end while
return $\bar{\Pi}$;

Algorithm 1. The basis separation procedure

Note that we cannot guarantee to generate cuts that cut off the solution $\bar{x}$ when using Algorithm 1. Moreover, it highly depends on the types of cuts that are separated and how the basic feasible solutions are calculated.

4.1 Basis Heuristics

In this section we present approaches to cope with the crucial step in Algorithm 1 of calculating a basic feasible solution. Suppose we are given an optimal solution $\bar{\lambda}$ of the linear master problem. We want to find a basic feasible solution x^* of the original LP relaxation such that cuts separating x^* also tend to separate the solution $\bar{x} := \sum_{p \in P} x^p \bar{\lambda}_p + \sum_{r \in R} x^r \bar{\lambda}_r$. Approaches to obtain such a basic feasible solution are called *basis heuristics*. They were introduced by Range [22] in the context of branch-price-and-cut as well as by Dash and Goycoolea [10] in order to heuristically separate rank-1 GMI cuts. We will focus on basis heuristics based on solving linear programs in the following.

Original Objective. Probably the first idea that comes to mind is that we can obtain a basic feasible solution x^* of the original LP relaxation by solving the original LP relaxation. This basis heuristic will be called the *original objective*.

This approach is similar to *cut-first branch-and-price second* [6], where the original LP relaxation is solved first, cutting planes are added, and then the strengthened original problem is reformulated and solved using branch-and-price. A crucial difference is that cuts are added a priori to the problem in cut-first branch-and-price second without knowing if future solutions will ever violate these cutting planes. If we use the basis separation procedure instead, only cuts violated by the current solution of the master LP relaxation will be added, which is a clear advantage. A disadvantage of both approaches is that they are independent from the solution $\bar{x}$. They only depend on the original problem.

Range's Face Objective. In the following we present an alternative approach introduced by Range [22], where also the solution $\bar{x}$ is considered. Let

$$A' := \begin{pmatrix} A \\ D \end{pmatrix} \in \mathbb{Q}^{m \times n} \quad \text{and} \quad b' := \begin{pmatrix} b \\ d \end{pmatrix} \in \mathbb{Q}^{m}$$

with $m := m_1 + m_2$ be the constraint matrix and the left-hand side of the original problem. Furthermore, denote by A'_i the i-th row of the matrix A' for $i \in \{1, \ldots, m\}$ and let $I_0 := \{i \in \{1, \ldots, m\} : {A'}_i \bar{x} = b'_i\}$ be the set of indices corresponding to constraints of the original problem that are active at $\bar{x}$.

With the aim of obtaining a basic feasible solution of the original LP relaxation *near* $\bar{x}$, we solve the original LP relaxation using the *face objective* function

$$f(\bar{x}, x) := \sum_{i \in I_0} \frac{A'_i x - b'_i}{||A'_i||_2} \ ,$$

where $|| \cdot ||_2$ is the Euclidian norm.

The following proposition was initially proposed by Range [22].

Proposition 1 ([22]). *The solution $\bar{x}$ is an optimal feasible solution for the original LP relaxation with face objective and the optimal solution value is zero.*

Note that the *face* [4] $F := \{x \in \mathbb{Q}^n : A'x \geq b', A'_i x = b'_i \ \forall i \in I_0\}$ of the polyhedron P_{LP} is by definition of I_0 the face of smallest dimension containing $\bar{x}$. Since $x^* \in F$ holds, the solution x^* is at least contained in all faces the solution $\bar{x}$ is contained in. Hence, when solving the original LP relaxation with face objective using the simplex algorithm we obtain an optimal basic feasible solution x^* with $x^* \neq \bar{x}$ if and only if $\bar{x}$ is not basic feasible.

When the number of linearly independent rows A'_i with $i \in I_0$ is small in comparison to n, the information provided by the face objective is rather poor, because many linear independent inequalities active at $\bar{x}$ are missing to describe a basic solution.

Extended Face Objective. In the following we present an extension of the face objective taking also non-active constraints into account. For $k \in \mathbb{Z}_{\geq 0}$ we define the *k-activity* $g_k(\bar{x}, a, a_0)$ of an inequality $a^T x \geq a_0$ with $a \in \mathbb{Q}^n$ and $a_0 \in \mathbb{Q}$ at a given solution $\bar{x}$ as

$$g_k(\bar{x}, a, a_0) := \max\left(1 - \frac{a^T \bar{x} - a_0}{||a||_2}, 0\right)^k \quad .$$

The k-activity $g_k(\bar{x}, a, a_0)$ describes how close to being active the inequality $a^T x \geq a_0$ is at $\bar{x}$. Note that $0 \leq g_k(\bar{x}, a, a_0) \leq 1$ holds and $g_k(\bar{x}, a, a_0) = 1$ if and only if the constraint $a^T x \geq a_0$ is active at $\bar{x}$. Furthermore, the greater the value k is chosen the smaller is the k-activity of a fixed non-active inequality.

We define the k-extended face objective, which is an extension of the face objective using the k-activity as a measure of the influence of a constraint:

$$f_k(\bar{x}, x) := \sum_{i=1}^{m} g_k(\bar{x}, A'_i, b'_i) \cdot \frac{A'_i x - b'_i}{||A'_i||_2} \quad .$$

We additionally consider constraints that are almost active at $\bar{x}$, because if many of these constraints are active at a basic solution, this solution is intuitively a good approximation of the solution $\bar{x}$ we want to separate. Solving the original LP relaxation using the k-extended face objective yields such a basic solution.

Combination. We previously introduced three objectives that can be used as basis heuristics in combination with the original LP relaxation. The original objective is independent from the solution $\bar{x}$, whereas the face and the extended face objective are independent from the original objective function, they only depend on the solution $\bar{x}$ and the polyhedron P_{LP}. In the following we combine these objective functions in order to exploit as much information as possible.

We will combine the face and the original objective function by using a convex combination with coefficient $\alpha \in [0, 1]$

$$\min \alpha \cdot \frac{f(\bar{x}, x)}{|I_0|} + (1 - \alpha) \cdot \frac{c^T x}{||c||_2} \quad .$$

Note that $|I_0|$ is the norm of the face objective. Analogously, we can combine the extended face objective and the original objective by using the norm $\sum_{i=1}^{m} g_k(\bar{x}, A'_i, b'_i)$ of the k-extended face objective. In the following we will present an approach to automatically choose a good value for α.

Remark that n linear independent inequalities are active at $\bar{x}$ if and only if the solution $\bar{x}$ is basic. Let $n(\bar{x})$ be the maximum number of linear independent inequalities active at $\bar{x}$ and define $\alpha(\bar{x}) := \frac{n(\bar{x})}{n} \in [0,1]$, which can be used as a measure of how close $\bar{x}$ is to being basic. In the following we describe why $\alpha(\bar{x})$ is intuitively a suitable value for the convex combination coefficient α.

Obviously, $\alpha(\bar{x}) = 1$ if and only if $\bar{x}$ is basic. If $\alpha(\bar{x}) \approx 1$, then only few linear independent inequalities are missing to describe a basic solution. The influence of the face objective is increased, whereby the almost complete basis information of $\bar{x}$ will be exploited. On the contrary, if $\alpha(\bar{x}) \ll 1$, many linear independent inequalities are missing to describe a basic solution and the influence of the face objective, which contains only poor information, will be decreased.

5 Strengthening of the Original LP Relaxation Before Separation

In many applications the constraints $Dx \geq d$ are chosen in such a way that the LP relaxation of the master problem is much stronger than the one of the original problem. Thus, a basic solution of the original LP relaxation calculated during the basis separation procedure can only poorly approximate the solution $\bar{x}$ that was projected from the linear master problem. To counteract this and to enhance the basis separation procedure, we can try to imitate the convexification of the constraints $Dx \geq d$ by adding valid inequalities for $P_{DW} := \{x \in \text{conv}(X) : Ax \geq b\} \supseteq P_{MIP}$ before separation. In the following we present valid inequalities for P_{DW} that can be obtained while applying a branch-price-and-cut algorithm.

Range's Original Objective Cut. Range [22] suggests to add the *original objective cut* $c^T x \geq c^T \bar{x}$ to the problem in order to potentially strengthen the original LP relaxation. Note that this inequality holds for all $x \in P_{DW}$, because $\bar{x}$ is optimal for $\min\{c^T x : x \in P_{DW}\}$. If the objective function is known to be integral, e.g., $c \in \mathbb{Z}^n$ and $q = 0$, the inequality $c^T x \geq \lceil c^T \bar{x} \rceil$ can be added.

Reduced Cost Cuts. In each column generation iteration we solve a pricing problem over the set X in order to find negative reduced cost columns. Let $\pi^T x$ be the objective function of the pricing problem in some column generation iteration and let $\pi_0 := \min\{\pi^T x : x \in X\}$ be the optimal solution value of the corresponding pricing problem. Note that $\pi^T x \geq \pi_0$ is valid for $\text{conv}(X)$. Since $\text{conv}(X) \supseteq P_{DW}$, the inequality is also valid for P_{DW}. Inequalities of this type will be called *reduced cost cuts*, because they state that the reduced costs of all potential columns are greater than or equal to a specific value.

Pricing Cuts. Suppose the pricing problem is solved using branch-and-cut and in some column generation iteration a cutting plane $\pi^T x \geq \pi_0$ is separated in the pricing problem during separation at the root node. Since we optimize over X in the pricing problem, $\pi^T x \geq \pi_0$ is valid for $\text{conv}(X) \supseteq P_{DW}$. We will call such inequalities *pricing cuts*, because they are generated in the pricing problem.

6 Computational Setup and Results

We implemented the basis separation procedure including all presented features in GCG 2.0.1 [15] based on a development version of SCIP 3.1.0 [1] with CPLEX 12.5.0.0 as LP-solver. All computations were performed on Intel Core i7-2600 CPUs with 16GB of RAM on openSUSE 13.1 workstations running Linux kernel 3.11.10. We used a time limit of 3600 seconds in all our tests.

In GCG combinatorial cuts in the original problem are separated by default, but we will only report on the number of cuts that were separated by the basis separation procedure and were applied to the problem. Note that SCIP/GCG filters the separated cuts and only applies a subset of them. We used SCIP's separators with the aggressive setting to separate a basic feasible solution in Algorithm 1 and only separated cuts at the root node. In all our tests we used the values $p_{\min} = 1$ and $i_{\max} = 100$ for Algorithm 1. In order to compute $\alpha(\bar{x})$, we used the QR decomposition with column pivoting from Gnu Scientific Library [13].

We applied the branch-price-and-cut algorithm including basis separation to instances of the following problems: capacitated p-median problem (cpmp) [2], generalized assignment problem (gap) [8,9,18], resource allocation/temporal knapsack problem (rap) [7], and lot sizing problem (lotsizing) [24]. Furthermore, we applied the algorithm to instances of MIPLIB 2003 and MIPLIB 2010 (miplib) that were already successfully tested with a generic branch-price-and-cut code [3]. We only considered instances where separation at the root node could be applied.

6.1 Performance of the Basis Separation Procedure

In Table 1 we compare GCG using the default settings (def), basis separation with face objective (face), basis separation with the combination of face and original objective (face-conv), basis separation with the combination of 8-extended face and original objective (8-ext-conv), and basis separation with original objective (origobj). Additionally, we considered basis separation with k-extended face objective as well as the combination of k-extended face and original objective for $k \in \{4, 8, 12\}$, but preliminary tests have shown that the combination of 8-extended face and original objective outperforms these heuristics.

As we can see, on the majority of the cpmp, lotsizing, and miplib instances cuts are separated no matter which basis heuristic is used. Although the number of applied cuts is in shifted geometrical mean at most 19 over a test set and mostly much smaller, a non negligible part of the integrality gap at the root node is closed in comparison to the default setting. When using basis separation with the combination of 8-extended face and original objective, 8 percent of the

Table 1. Comparison of the percentage of affected instances (aff), i.e., some cuts were separated, the shifted geom. mean with shift value 1 of the int. gap at the root node in percent (gap), the number of applied cuts at the root node of the affected instances (cuts), and the time spent in the basis separation procedure (tm) over the whole testset. The best gap is written bold.

	def	face				face-conv				8-ext-conv				origobj			
	gap	gap	aff	cuts	tm	gap	aff	cuts	tm	gap	aff	cuts	tm	gap	aff	cuts	tm
cpmp_easy	1.21	1.15	78.8	4.3	1.0	1.15	79.8	4.3	1.1	**1.11**	81.7	4.0	1.1	1.18	35.6	1.9	1.7
cpmp_hard	4.18	4.05	87.5	9.3	2.5	4.04	87.5	9.4	2.8	**4.01**	89.1	9.3	3.4	4.13	48.4	1.9	4.3
gap_easy	0.15	0.15	4.2	1.0	0.5	0.15	4.2	1.0	0.6	**0.14**	16.7	2.8	0.5	0.15	8.3	1.0	0.5
gap_hard	0.31	0.31	0.0	0.0	3.3	0.31	0.0	0.0	3.9	0.31	0.0	0.0	3.8	0.31	0.0	0.0	4.5
ls_easy	3.31	2.08	95.2	5.8	0.5	2.11	95.2	5.8	0.6	**2.07**	85.7	6.4	0.6	2.36	85.7	4.0	0.5
ls_hard	13.36	13.04	66.7	14.5	0.5	13.07	66.7	15.5	1.5	**12.83**	66.7	17.5	1.4	12.96	66.7	8.2	0.5
rap_easy	0.04	0.04	0.0	0.0	0.9	0.04	0.0	0.0	1.0	0.04	0.0	0.0	1.0	0.04	6.2	2.0	1.5
rap_hard	0.09	0.09	0.0	0.0	2.4	0.09	0.0	0.0	2.5	0.09	0.0	0.0	2.5	0.09	20.7	1.7	2.9
miplib_easy	1.30	1.13	100.0	2.7	0.5	1.13	100.0	2.7	0.5	**0.87**	100.0	4.1	0.5	1.04	33.3	1.4	0.5
miplib_hard	4.94	4.79	66.7	16.2	1.6	4.77	73.3	14.0	11.9	4.91	73.3	14.4	11.6	**4.56**	73.3	14.0	1.6

gap on easy cpmp, 33 percent of the gap on easy miplib, and 37 of the gap percent on easy lotsizing instances is closed. Using any other basis heuristic closes less of the gap. On the corresponding hard instances up to 8 percent of the gap was closed due to basis separation, where the usage of the original objective or the combination of 8-extended face and original objective perform best.

On the contrary, almost no cuts were separated on gap and rap instances, which is probably due to the already very small integrality gap. Consequently, the usage of basis separation closes hardly anything of the integrality gap.

Note that most often only a few seconds are spent in the basis separation procedure. Only when using basis heuristics that have to compute the number of linear independent inequalities active at the current solution $\bar{x}$ in order to determine the value $\alpha(\bar{x})$, separation can take a bit longer on some hard instances. But in shifted geometrical mean over a test set it does not exceed 12 seconds.

In Fig. 2 the number of nodes and the solution times required by the settings with basis separation are compared to the default settings. On most test sets the solution time and even more significantly the number of nodes is reduced due to

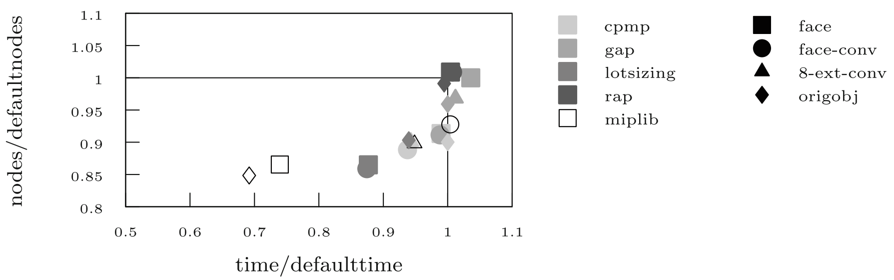

Fig. 2. Ratio between the shifted geom. mean with shift 100 (10) of the number of nodes (solution times) required by the settings with basis separation and the default settings.

separation. Only on gap and rap instances, where almost no cuts were separated, the solution time is increased. But since the basis separation procedure is quite fast, the increase in solution time is relatively small. Note that these results match the previous made observations concerning the integrality gap.

On lotsizing and miplib instances the solution time is decreased by up to 12 and 30 percent, respectively. On cpmp instances the solution time is only marginally decreased. During our computational study, we additionally observed that the number of solved instances of these problems is slightly increased when applying the basis separation procedure.

6.2 Influence of Strengthening the Original LP Relaxation Before Separation

In Table 2 the influence of the valid inequalities presented in section 5 on basis separation with the combination of the 8-extended face and the original objective is investigated. Namely these valid inequalities are the original objective cut (origobjcut), the pricing cuts (ppcuts), and the reduced cost cuts (redcostcuts).

Table 2. Comparison similar to Table 1

	basis-conv-8-ext				+origobjcut				+ppcuts				+redcostcuts			
	gap	aff	cuts	tm	gap	aff	cuts	tm	gap	aff	cuts	tm	gap	aff	cuts	tm
cpmp_easy	**1.69**	86.2	5.4	1.5	1.72	84.1	5.3	1.5	1.70	81.2	12.0	41.6	1.74	79.0	5.9	33.8
cpmp_hard	3.25	91.3	8.6	3.2	3.27	89.1	8.4	3.3	**3.24**	78.3	24.6	159.6	3.30	82.6	8.6	261.3
gap_easy	0.18	14.8	2.8	0.6	0.18	14.8	1.9	0.6	**0.17**	18.5	1.9	0.6	**0.17**	29.6	2.3	2.7
gap_hard	0.40	0.0	0.0	1.0	0.40	0.0	0.0	0.9	0.40	0.0	0.0	0.5	0.40	0.0	0.0	110.8
ls_easy	2.02	88.2	6.5	0.5	**1.72**	88.2	5.7	0.5	2.32	94.1	6.5	0.6	1.97	94.1	4.8	0.5
ls_hard	31.18	42.9	30.6	2.1	**24.39**	42.9	6.3	0.8	29.48	57.1	38.1	3.6	25.34	42.9	6.3	1.2
rap_easy	0.04	0.0	0.0	0.9	0.04	0.0	0.0	0.9	0.04	0.0	0.0	0.9	0.04	0.0	0.0	6.2
rap_hard	0.09	0.0	0.0	1.9	0.09	0.0	0.0	1.9	0.09	0.0	0.0	2.0	0.09	0.0	0.0	86.8
miplib_easy	**0.91**	100.0	4.3	0.5	**0.91**	100.0	3.6	0.5	1.36	80.0	5.9	34.6	0.95	100.0	4.1	0.8
miplib_hard	4.91	73.3	14.4	11.6	4.90	73.3	14.0	12.1	5.26	80.0	13.0	15.6	**4.77**	73.3	13.6	25.3

Notice that the percentage of affected instances is of similar magnitude no matter if the additional valid inequalities were added or not, whereas the number of applied cuts and the integrality gap vary considerably. Surprisingly, every setting that is shown in Table 2 provides on some test set the smallest integrality gap. So the impact of the valid inequalities is not solely positive. The same observation can be made when considering the number of applied cuts. Furthermore, the number of applied cuts and the size of the integrality do not seem to correlate.

On some instances the time spent in the basis separation procedure is noticeably increased due to the valid inequalities that were added before separation.

7 Conclusions and Future Work

We discussed and extended Range's approach [22] to separate cuts in the original problem using a basis in the context of branch-price-and-cut algorithms.

Furthermore, we implemented all ideas in GCG and presented the first computational study on a separation procedure of this kind. The cuts close part of the integrality gap at the root node on instances of various problem types, reducing the number of nodes and the solution time. On instances, where no cuts were found, solution times just slightly increased, because the separation procedure is relatively fast.

Whereas the combination of the 8-extended face and the original objective seems to be the basis heuristic that improves performance the most, computational results concerning the strengthening of the original LP relaxation before separation are not that clear, because they do not solely improve the performance. A task of future research should be finding a selection of valid inequalities that exclusively have a positive influence on the separation procedure.

The presented basis heuristics only compute feasible basic solutions, but Dash and Goycoolea [10] also use basis heuristics that compute infeasible basic solutions in order to heuristically separate rank-1 GMI cuts. Future work should include the implementation of these basis heuristics in our framework.

Since only auxiliary basic solutions and not the solutions we want to separate are used to generate cutting planes, a subject of future research should be the generation of additional valid inequalities as discussed in section 5 such that the solution we want to separate becomes a basic solution in the original LP relaxation whenever this is possible. If we managed to achieve this, we could obtain a corresponding dual solution and apply reduced cost fixing [17,19].

Our experiments suggest that there is a strong relation between the strength of the Dantzig-Wolfe reformulation and the success of separating violated cuts in the original problem. Future research should further examine this relation both computationally and theoretically.

References

1. Achterberg, T.: Constraint Integer Programming. Ph.D. thesis, Technische Universität Berlin (2007)
2. Beasley, J.: OR-Library: Distributing test problems by electronic mail. J. Oper. Res. Soc. **41**(11), 1069–1072 (1990)
3. Bergner, M., Caprara, A., Ceselli, A., Furini, F., Lübbecke, M., Malaguti, E., Traversi, E.: Automatic Dantzig-Wolfe reformulation of mixed integer programs. Math. Prog. **149**(1–2), 391–424 (2015)
4. Bertsimas, D., Tsitsiklis, J.: Introduction to Linear Optimization. Athena Scientific, Belmont (1997)
5. Bixby, R., Rothberg, E.: Progress in computational mixed integer programming - A look back from the other side of the tipping point. Annals of Operations Research **149**(1), 37–41 (2007). http://dx.doi.org/10.1007/s10479-006-0091-y
6. Bode, C., Irnich, S.: Cut-first branch-and-price-second for the capacitated arc-routing problem. Oper. Res. **60**(5), 1167–1182 (2012)
7. Caprara, A., Furini, F., Malaguti, E.: Uncommon Dantzig-Wolfe reformulation for the temporal knapsack problem. INFORMS J. Comput. **25**(3), 560–571 (2013)
8. Cattrysse, D.G., Salomon, M., Wassenhove, L.N.V.: A set partitioning heuristic for the generalized assignment problem. European J. Oper. Res. **72**(1), 167–174 (1994)

9. Chu, P.C., Beasley, J.E.: A genetic algorithm for the generalised assignment problem. Comput. Oper. Res. **24**(1), 17–23 (1997)
10. Dash, S., Goycoolea, M.: A heuristic to generate rank-1 GMI cuts. Math. Program. Comput. **2**(3–4), 231–257 (2010)
11. Desaulniers, G., Desrosiers, J., Spoorendonk, S.: Cutting planes for branch-and-price algorithms. Networks **58**(4), 301–310 (2011)
12. Desrosiers, J., Lübbecke, M.E.: Branch-price-and-cut algorithms. In: Cochran, J.J., Cox, L.A., Keskinocak, P., Kharoufeh, J.P., Smith, J.C. (eds.) Wiley Encyclopedia of Operations Research and Management Science. John Wiley & Sons, Inc. (2010)
13. Galassi, M., et al.: GNU scientific library reference manual. ISBN 0954612078
14. Galati, M.: Decomposition methods for integer linear programming. Ph.D. thesis, Lehigh University (2010)
15. Gamrath, G., Lübbecke, M.E.: Experiments with a generic Dantzig-Wolfe decomposition for integer programs. In: Festa, P. (ed.) SEA 2010. LNCS, vol. 6049, pp. 239–252. Springer, Heidelberg (2010)
16. Goncalves, A.S.: Basic feasible solutions and the Dantzig-Wolfe decomposition algorithm. J. Oper. Res. Soc. **19**(4), 465–469 (1968)
17. Irnich, S., Desaulniers, G., Desrosiers, J., Hadjar, A.: Path-reduced costs for eliminating arcs in routing and scheduling. INFORMS J. Comput. **22**(2), 297–313 (2010)
18. Osman, I.H.: Heuristics for the generalised assignment problem: Simulated annealing and tabu search approaches. OR Spectrum **17**(4), 211–225 (1995)
19. Poggi de Aragão, M., Uchoa, E.: Integer program reformulation for robust branch-and-cut-and-price. In: Mathematical Programming in Rio: A Conference in Honour of Nelson Maculan, pp. 56–61 (2003)
20. Puchinger, J., Stuckey, P., Wallace, M., Brand, S.: Dantzig-Wolfe decomposition and branch-and-price solving in G12. Constraints **16**(1), 77–99 (2011)
21. Ralphs, T., Galati, M.: DIP - Decomposition for integer programming (2009). https://projects.coin-or.org/Dip
22. Range, T.: An integer cutting-plane procedure for the Dantzig-Wolfe decomposition: Theory. Discussion Papers on Business and Economics 10/2006, Dept. Business and Economics. University of Southern Denmark (2006)
23. Rios, J., Ross, K.: Converging upon basic feasible solutions through Dantzig-Wolfe decomposition. Optim. Lett. **8**(1), 171–180 (2014)
24. Tempelmeier, H., Derstroff, M.: A lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times. Management Science **42**(5), 738–757 (1996)
25. Vanderbeck, F.: BaPCod - A generic branch-and-price code (2005). https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/BaPCod
26. Vanderbeck, F., Savelsbergh, M.: A generic view of Dantzig-Wolfe decomposition in mixed integer programming. Oper. Res. Lett. **34**(3), 296–306 (2006)