Math. Program., Ser. A (2011) 128:205–230
DOI 10.1007/s10107-009-0300-y

FULL LENGTH PAPER

On the separation of disjunctive cuts

Matteo Fischetti · Andrea Lodi · Andrea Tramontani

Received: 17 March 2008 / Accepted: 14 July 2009 / Published online: 15 August 2009

Abstract Disjunctive cuts for Mixed-Integer Linear Programs (MIPs) were introduced by Egon Balas in the late 1970s and have been successfully exploited in practice since the late 1990s. In this paper we investigate the main ingredients of a disjunctive cut separation procedure, and analyze their impact on the quality of the root-node bound for a set of instances taken from MIPLIB library. We compare alternative normalization conditions, and try to better understand their role. In particular, we point out that constraints that become redundant (because of the disjunction used) can produce over-weak cuts, and analyze this property with respect to the normalization used. Finally, we introduce a new normalization condition and analyze its theoretical properties and computational behavior. Along the way, we make use of a number of small numerical examples to illustrate some basic (and often misinterpreted) disjunctive programming features.

Mathematics Subject Classification (2000) 90C10 · 90C11 · 90C57

Matteo Fischetti was supported in part by the EU project ARRIVAL (contract n. FP6-021235-2) and by MiUR, Italy (PRIN 2006 project "Models and algorithms for robust network optimization").
Andrea Lodi was supported in part by the EU project ARRIVAL (contract n. FP6-021235-2).

M. Fischetti (✉)
DEI, University of Padova, via Gradenigo 6A, 35131 Padova, Italy
e-mail: matteo.fischetti@unipd.it

A. Lodi · A. Tramontani
DEIS, University of Bologna, viale Risorgimento 2, 40136 Bologna, Italy
e-mail: andrea.lodi@unibo.it

A. Tramontani
e-mail: andrea.tramontani@unibo.it

1 Introduction

We consider the Mixed-Integer Linear Program (MIP)

$$\min\{cx : Ax \geq b, x_j \text{ integer for all } j \in J\} \tag{1}$$

with bounds on x (if any) included in $Ax \geq b$, where A is a given $m \times n$ matrix and $J \subseteq \{1, \ldots, n\}$. For technical reasons, we assume w.l.o.g. that the system $Ax \geq b$ implies (or contains explicitly) the trivial inequality $0x \geq -1$, in the sense that this latter inequality can be obtained as a nonnegative combination of the rows of $Ax \geq b$. (For problems with at least one bounded variable, the trivial inequality can always be obtained by adding the bound constraints on a single variable, say $x_j \geq LB_j$ and $-x_j \geq -UB_j$, and dividing the resulting inequality by $UB_j - LB_j > 0$.)

Let x^* denote an optimal solution of the continuous relaxation $\min\{cx : x \in P\}$ where $P := \{x \in \mathbb{R}^n : Ax \geq b\}$. We are given a disjunction of the form

$$\pi x \leq \pi_0 \quad \text{OR} \quad \pi x \geq \pi_0 + 1 \tag{2}$$

such that (π, π_0) is integer, $\pi_j = 0, \forall j \notin J$ and $\pi x^* - \pi_0 = \eta^*$, with $\eta^* \in\ ]0, 1[$.

In this paper we are interested in deriving the "strongest" (in some sense to be discussed later) disjunctive cut $\gamma x \geq \gamma_0$ violated by x^*, according to the classical approach of Balas [2]. (Disjunctive cuts which can be derived by imposing a single disjunction such as (2) on a polyhedron P are also known as *split* cuts; see Cook et al. [15].) To this end, let us denote by P_0 (respectively, P_1) the polyhedron obtained from P by imposing the additional restriction $\pi x \leq \pi_0$ (resp., $\pi x \geq \pi_0 + 1$). By Farkas Lemma, the validity of $\gamma x \geq \gamma_0$ for P_0 and for P_1, and hence for $conv(P_0 \cup P_1)$, can always be certified by means of nonnegative multipliers (u, u_0, v, v_0) associated with the inequalities defining P_0 and P_1 according to the following scheme:

$$\begin{array}{llll@{\qquad}llll}
 & & P_0 & & & & P_1 & \\
(u) & Ax & \geq & b & (v) & Ax & \geq & b \\
(u_0) & -\pi x & \geq & -\pi_0 & (v_0) & \pi x & \geq & \pi_0 + 1
\end{array}$$

A most-violated disjunctive cut can therefore be found by solving the following Cut Generating Linear Program (CGLP) that determines the Farkas multipliers so as to maximize the violation with respect to the given point x^*:

$$\text{(CGLP)} \quad \min \ \gamma x^* - \gamma_0 \tag{3}$$

$$\gamma = uA - u_0\pi \tag{4}$$

$$\gamma = vA + v_0\pi \tag{5}$$

$$\gamma_0 = ub - u_0\pi_0 \tag{6}$$

$$\gamma_0 = vb + v_0(\pi_0 + 1) \tag{7}$$

$$u, v, u_0, v_0 \geq 0. \tag{8}$$

Note that, according to Farkas Lemma, (6) and (7) defining γ_0 should be relaxed into $\leq$ inequalities. However, it is not difficult to see that, due to the (possibly implicit) presence of the trivial inequality $0x \geq -1$, one can always require that equality holds in both cases.

By construction, any feasible CGLP solution with negative objective function value corresponds to a violated disjunctive cut. However, as stated, the feasible CGLP set is a cone and needs to be truncated so as to produce a bounded LP in case a violated cut exists. This crucial step will be addressed in the next section.

Usually, the CGLP is projected onto the support of x^*. Given a variable x_k restricted to be nonnegative and such that $x_k^* = 0$, it is well known [5] that one can project x_k away. (Of course, variables with nonzero lower bound can be shifted, while variables at the upper bound can be complemented.) More precisely, one can avoid considering the CGLP constraints associated with γ_k and neglect the constraint $x_k \geq 0$ in both P_0 and P_1. The resulting (reduced) CGLP is then solved and the cut coefficient γ_k is derived afterwards by solving the trivial lifting problem

$$\min\{\gamma_k : \gamma_k = uA_k - u_0\pi_k = vA_k + v_0\pi_k, \quad u, v \geq 0\}, \tag{9}$$

where the Farkas multipliers u and v are fixed as in the optimal solution of the reduced CGLP apart from those related to the previously neglected bound constraint $x_k \geq 0$.

In practice, disjunction (2) is typically *elementary*, i.e., it involves only one integer variable—for 0-1 ILPs, it reads $x_j \leq 0$ OR $x_j \geq 1$, with x_j^* fractional. As such, the disjunctive cut only exploits the integrality requirement on a single variable and can be improved easily by an *a posteriori cut strengthening* procedure, such as the one proposed by Balas and Jeroslow [6]. Such a strengthening can be also seen as finding the best disjunction for the given set of multipliers.

Recently, Balas and Perregaard [8] developed an elegant and efficient way of solving the CGLP by making pivot operations in the "natural" tableau involving the original x variables only (plus surplus variables), which represents a crucial speed-up in the implementation of the method.

In this paper we investigate computationally the main ingredients of a disjunctive cut separation procedure, and analyze their impact on the overall performance at the root node of the branching tree. To be more specific, we consider a testbed of MIPs taken from MIPLIB library [11]. For each instance, we solve the root-node LP relaxation and generate 10 rounds of disjunctive cuts computed according to alternative strategies. In each round, a violated disjunctive cut is generated for each fractional LP components x_j^*, by exploiting the disjunction $x_j \leq \lfloor x_j^* \rfloor$ OR $x_j \geq \lfloor x_j^* \rfloor + 1$. In order to limit possible side effects, no *a posteriori* cut strengthening procedure is applied, unless otherwise stated.

The paper is organized as follows. In Sect. 2 we compare classical normalization conditions used to truncate the CGLP cone, and try to better understand their role. In Sect. 3 we characterize weak rays/vertices of the CGLP leading to dominated cuts and we propose a practical heuristic method to strengthen them. In Sect. 4 we show that using redundant constraints in the CGLP can lead to very weak cuts, and we analyze such an issue with respect to the normalization used. Finally, in Sect. 5 we introduce a new normalization which is particularly suited for set-covering type problems and we analyze its theoretical properties and computational behavior.

2 The role of normalization

In order to truncate the CGLP cone one can introduce a suitable cut normalization condition expressed as a linear (in)equality. A possible normalization, called *trivial* in the sequel, is as follows:

$$u_0 + v_0 = 1. \tag{10}$$

One of the most widely-used (and effective) truncation condition, called the *Standard Normalization Condition* (SNC) in the following, reads instead:

$$\sum_{i=1}^{m} u_i + \sum_{i=1}^{m} v_i + u_0 + v_0 = 1. \tag{11}$$

This latter condition was proposed in Balas [1] and investigated (among others) by Ceria and Soares [13], and by Balas and Perregaard [7,8].

The choice of the normalization condition turns out to be crucial for an effective selection of a "strong" disjunctive cut in that it affects heavily the choice of the optimal CGLP solution. Balas and Perregaard [8] showed that the well-known Gomory Mixed-Integer (GMI) cut [19] is a basic solution of the CGLP when either the SNC or the trivial normalization is applied. Our first result is to prove that this solution is indeed optimal when the trivial normalization (10) is used. We start with a useful lemma.

Lemma 1 *Let $x^* \in P$ and let $(\gamma, \gamma_0, u, v, u_0, v_0)$ be a feasible solution of the CGLP. Then valid upper bounds on the cut violation can be computed as follows*:

UB1: $\gamma_0 - \gamma x^* \le u_0 \eta^*$
UB2: $\gamma_0 - \gamma x^* \le v_0 (1 - \eta^*)$
UB3: $\gamma_0 - \gamma x^* \le (u_0 + v_0)\,(1 - \eta^*)\,\eta^*$.

Proof Because of (4) and (6), $\gamma x^* - \gamma_0 = u(Ax^* - b) - u_0(\pi x^* - \pi_0) \ge -u_0\eta^*$. Analogously, from (5) and (7) we obtain $\gamma x^* - \gamma_0 = v(Ax^* - b) + v_0(\pi x^* - \pi_0 - 1) \ge -v_0(1 - \eta^*)$. Adding up the two inequalities above weighed by $1 - \eta^*$ and η^*, respectively, one gets the claimed UB3 bound. □

Given a vertex x^* of P and the associated basis, the next theorem shows how to compute a solution of the CGLP whose violation is equal to bound UB3 above—for any given disjunction (2). Moreover, as shown in [7,8], for an appropriate choice of a non-elementary disjunction this CGLP solution yields precisely a GMI cut associated with the optimal LP tableau. As a consequence of Lemma 1, this easily-computable cut has a violation that is optimal among the cuts with constant $u_0 + v_0$, i.e., when the trivial normalization (10) is imposed. Note, however, that this is not necessarily the case when a different normalization (in particular, the SNC one) is applied.

For any vector v, let operator $[v]_+$ take the maximum between the argument and zero (componentwise); thus, $v \equiv [v]_+ - [-v]_+$ with $[v]_+ \geq 0$ and $[-v]_+ \geq 0$.

Theorem 2 *Assume w.l.o.g. that $rank(A) = n$. Given a vertex x^* of P, let the system $Ax \geq b$ be partitioned into $Bx \geq b_B$ and $Nx \geq b_N$, where $Bx^* = b_B$ and B is an $n \times n$ nonsingular matrix. Let (u_B, v_B) and (u_N, v_N) denote the Farkas multipliers associated with the rows of B and N, respectively. For a given disjunction (2) with $\eta^* = \pi x^* - \pi_0 \in [0, 1]$, let $u_0^* = 1 - \eta^*$, $v_0^* = \eta^*$, $u_N^* = v_N^* = 0$, $u_B^* = [\pi B^{-1}]_+$ and $v_B^* = [-\pi B^{-1}]_+$, while γ^* and γ_0^* are defined through (4) and (6), respectively. Then $(\gamma^*, \gamma_0^*, u^*, v^*, u_0^*, v_0^*)$ is an optimal CGLP solution w.r.t. the trivial normalization (10).*

Proof We first prove feasibility. Consistency between (4) and (5) requires $u^*A - u_0^*\pi = v^*A + v_0^*\pi$, i.e., $u_B^* - v_B^* = (u_0^* + v_0^*)\pi B^{-1} = \pi B^{-1}$, which follows directly from the definition of u_B^* and v_B^*. Analogously, consistency between (6) and (7) requires $(u_B^* - v_B^*)b_B = (u_0^* + v_0^*)\pi_0 + v_0^*$, i.e., $\pi B^{-1} b_B = \pi_0 + v_0^*$. This latter equation is indeed satisfied because $B^{-1}b_B = x^*$ and $v_0^* = \eta^* = \pi x^* - \pi_0$. As to optimality, we observe that $u_0^* + v_0^* = 1$ holds by definition. Because of (4) and (6), $\gamma x^* - \gamma_0 = u^*(Ax^* - b) - u_0^*(\pi x^* - \pi_0) = u_B^*(Bx^* - b_B) + u_N^*(Nx^* - b_N) - u_0^*\eta^* = 0 + 0 - (1 - \eta^*)\eta^*$, hence the cut violation attains bound UB3 of Lemma 1. □

The theorem above shows that, in case the trivial normalization is adopted, the CGLP can be solved in a closed form for any vertex x^*. Moreover, with this normalization, in all optimal CGLP solutions the slack constraints receive a null Farkas multiplier, i.e., only tight constraints play a role in the cut derivation. This is an unnecessary restriction that can actually lead to weak cuts, as computationally shown in the sequel.

The first set of experiments we designed was aimed at evaluating the actual practical impact of different normalization conditions. In particular, we compared the SNC normalization (11) with the alternative trivial normalization (10) by warm starting each CGLP with the basic feasible solution from Theorem 2. (The Balas and Perregaard [8] technique working on the original tableau and the solution of CGLP by using the GMI as a warm start, are equivalent procedures.) Moreover, unless explicitly stated, the CGLP is projected onto the support of x^*, possibly after complementing and shifting variables at their bound.

The outcome of our experiments is given in Table 1. As already mentioned, we applied 10 rounds of cuts. At each round, a cut was generated from each fractional variable. No a-posteriori cut strengthening was applied. As usual, the CGLP is solved projected on the x^* support. Instances denoted as "*" are neglected in the average computations. The table reports (i) the number of separated cuts, (ii) the quality of the lower bound (i.e., percentage gap closed at the root node) and (iii) the average cardinality of the support of vector $u + v$, denoted as $S(u, v) := \{i \in \{1, \ldots, m\} : u_i + v_i > 0\}$ ($|S|$ for short), i.e., how many constraints are actually used, on average, to generate a cut.

Table 1 shows clearly that normalizations (11) and (10) yield quite different results. As a matter of fact, the dual support of cuts separated with (11) is much sparser (i.e., less constraints are used in the cut derivation) and the quality of final bound is significantly improved. To get more insights on the different behaviors of (11) and

Table 1 Trivial versus SNC normalization

Instance	Trivial normalization (GMI)			SNC normalization		
	# cuts	%gap	$\|S\|$	# cuts	%gap	$\|S\|$
bell3a	137	70.74	59.49	71	70.74	43.72
bell5	202	28.18	31.20	178	94.29	11.75
blend2	156	28.73	11.70	192	30.51	8.10
flugpl	93	15.15	7.57	92	18.36	5.85
gt2	191	98.71	14.52	196	93.46	10.28
lseu	152	32.94	14.34	196	41.33	9.17
*markshare1	68	0.00	1.00	74	0.00	1.39
mod008	104	12.09	10.40	139	17.05	12.41
p0033	103	58.33	5.72	113	67.86	4.81
p0201	574	18.58	56.03	767	93.82	13.43
rout	445	8.52	135.39	434	24.26	68.07
*stein27	235	0.00	19.74	252	0.00	6.53
vpm1	255	36.95	9.03	263	55.84	5.39
vpm2	424	42.08	71.72	403	74.96	17.27
avg.	236.333	37.583	35.593	253.667	56.873	17.521

(10), for instance `p0201` we provide a full picture of the main differences between the separated inequalities.

Figures 1, 2 and 3 report, for each iteration, the dual bound reached after adding the cuts, the average density of the cuts (i.e., the number of nonzero coefficients), and the average cardinality of $S(u, v)$. Figure 4 reports, for each $k = 1, \ldots, 10$, the number of separated cuts having "rank" k. Here, we use a relaxed definition of rank, namely we compute the rank $rnk(\gamma, \gamma_0)$ of a cut $\gamma x \geq \gamma_0$ as

$$rnk(\gamma, \gamma_0) := 1 + \max_{i \in S(u,v)} rnk(a_i, b_i),$$

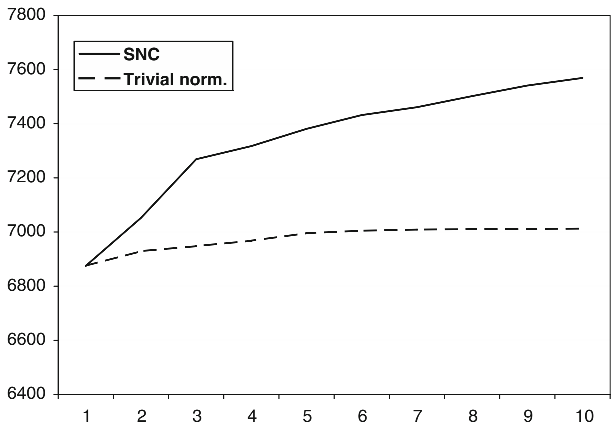

Fig. 1 SNC versus GMI: dual bound for instance *p0201*

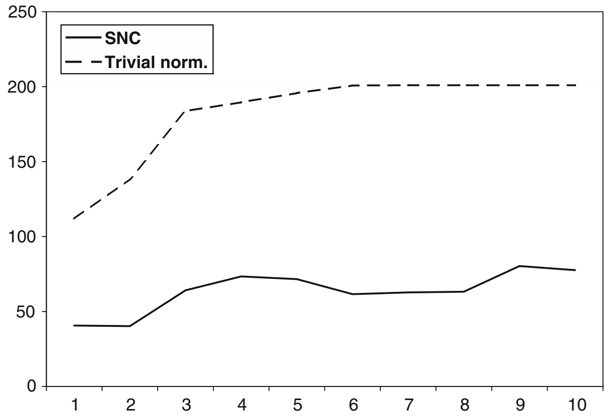

Fig. 2 SNC versus GMI: average cut density for instance $p0201$

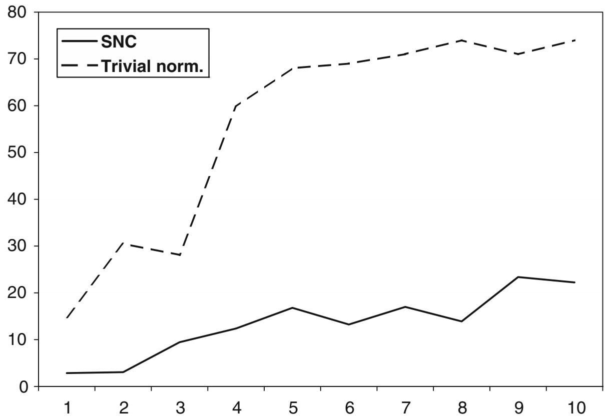

Fig. 3 SNC versus GMI: average cardinality of $S(u, v)$ for instance $p0201$

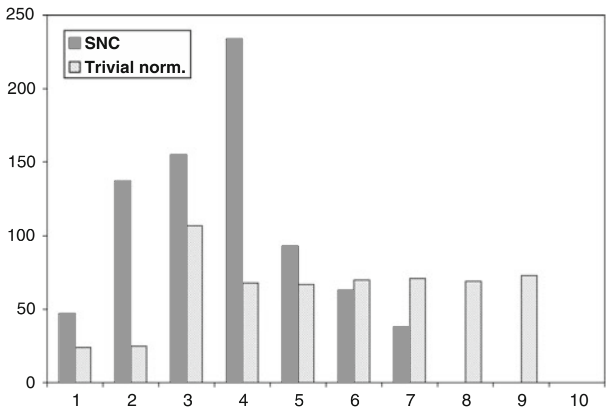

Fig. 4 SNC versus GMI: cut rank for instance $p0201$

where $rnk(a_i, b_i)$ is the rank of constraint $a_i x \geq b_i$ (constraints in the original formulation are defined to be of zero rank). (Note that this way of computing the rank provides just an upper bound on the definition of Chvátal rank [14].)

2.1 Why does SNC normalization work so well?

A careful analysis of the computational results in Table 1 and Figs. 1, 2, 3 and 4 reveal a very (tricky but) important feature of the SNC scheme that improves significantly its performance. Indeed, it turns out that the use of the SNC normalization (11) enforces the following very nice properties:

1. The norm of the separated cuts tends to become smaller and smaller as a result of the small multipliers used for the newly generated cuts (that is, in turn, a consequence of having limited the multiplier sum to 1). This means that the separated cuts inserted in the LP are automatically scaled so as to have "small coefficients". Therefore, in the subsequent iterations these cuts would need big Farkas multipliers to become relevant, a situation that is, however, penalized by the normalization condition itself. As a consequence, the normalization penalizes implicitly the rank of the cuts to be generated, because high-rank cuts will be "expensive" in terms of multiplier sum, hence low-rank cuts tend to be separated at each step.
2. Since low-rank cuts are preferred and since the original (rank-0) inequalities are generally sparse, the separated cuts tend to remain sparse; this is also a consequence of the fact that the SNC normalization tends to reduce the sum of the components of the Farkas multiplier vector and hence it increases the sparsity of its support, so a small number of constraints are typically used in the disjunctive cut derivation.

The trivial normalization (10), instead, takes care only of the Farkas multipliers u_0 and v_0 associated with the disjunction. Indeed, as shown in Sect. 2, only constraints which are tight at x^* are used in the cut derivation, thus the rank of the cuts increases very quickly, basically at each iteration. Moreover, all other constraint multipliers are not penalized, hence (i) several constraints are used in the cut derivation, thus cuts increase their density, and (ii) Farkas multipliers can assume huge values, thus the subsequent cut lifting procedure may produce very weak coefficients for the variables outside the support of x^*.

In the SNC normalization case the coefficient lifting is not an issue. Indeed, since all the constraint multipliers in the SNC normalization are penalized and each multiplier tends to be small, the coefficient lifting of the variables outside the support of x^*—to be performed afterwards—is "safe", i.e., also the coefficients of these variables remain under control.

2.2 Nothing is perfect!

Although it produced good results in the experiments reported in Table 1, there are cases where normalization (11) may lead to very weak disjunctive cuts.

Table 2 "Classical" SNC approach versus "Bad scaled" SNC approach

Instance	"Classical" SNC			"Bad scaled" SNC		
	# cuts	%gap	$\lvert S\rvert$	# cuts	%gap	$\lvert S\rvert$
bell3a	71	70.74	43.72	69	70.74	44.32
bell5	178	94.29	11.75	214	88.83	17.47
blend2	192	30.51	8.10	166	28.91	11.71
flugpl	92	18.36	5.85	90	15.40	7.40
gt2	196	93.46	10.28	184	93.42	17.22
lseu	196	41.33	9.17	137	38.58	10.88
*markshare1	74	0.00	1.39	206	0.00	14.60
mod008	139	17.05	12.41	104	3.90	10.21
p0033	113	67.86	4.81	94	57.09	6.40
p0201	767	93.82	13.43	610	49.91	45.72
rout	434	24.26	68.07	435	13.03	152.66
*stein27	252	0.00	6.53	248	0.00	22.39
vpm1	263	55.84	5.39	244	47.59	8.50
vpm2	403	74.96	17.27	420	54.39	22.27
avg.	253.667	56.873	17.521	230.583	46.816	29.563

Bad scaling. A bad feature of the SNC normalization is its dependency on the relative scaling of the constraints, in the sense that the relative size of the Farkas multipliers (whose sum is fixed to 1) depends on the relative size of the coefficients of the corresponding constraints. Indeed, it is easy to see that the multiplication by a positive factor ϕ of the i-th constraint in the system $Ax \geq b$ implies that the corresponding u_i and v_i multipliers are divided by ϕ, which in turn is equivalent to use a coefficient $1/\phi$ (instead of 1) in the normalization condition (11). Thus, the scaled constraint is "cheaper" if one interprets the right hand side of (11) as a resource.

The following experiment clearly demonstrates this unstable behavior: we ran the CGLP code with the classical SNC normalization condition, as in Table 1, but we just multiplied by 1,000 each disjunctive cut before its addition to the current LP. At first glance, one could guess that this "innocent change" would not have any impact on the overall performance, but the actual results reported in Table 2 show that this is definitely not the case.

As explained, multiplying by 1,000 the generated cuts is equivalent to dividing by 1,000 the coefficient of the corresponding Farkas multipliers u_i and v_i in the normalization condition, so we actually weaken the penalty on the choice $u_i + v_i > 0$ that leads to low-rank sparse cuts. In other words, the scaling operation interferes with the nice SNC tendency of producing low-rank cuts, and the overall performance deteriorates significantly, as shown in detail for problem `p0201` in Figs. 5, 6, 7, and 8. (Incidentally, the above discussion shows the importance of "small implementation details" when evaluating the performance of a method—two apparently equivalent implementations of precisely the same idea lead to very different outcomes.)

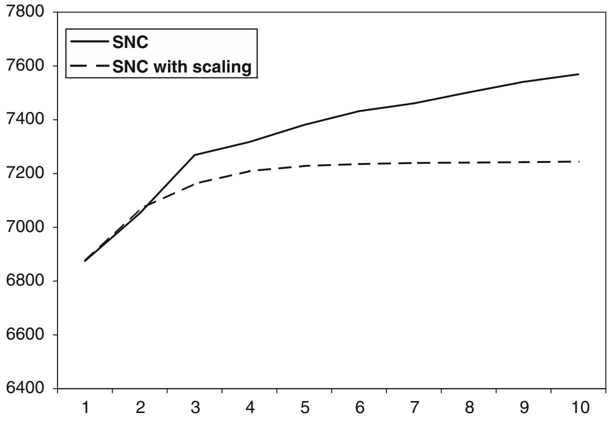

Fig. 5 “Classical” SNC versus “Bad scaled” SNC: dual bound for instance $p0201$

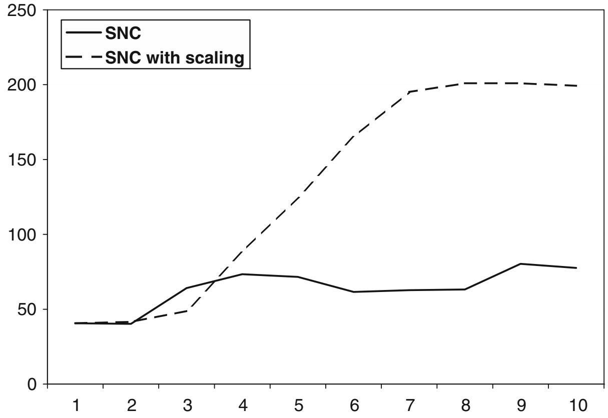

Fig. 6 “Classical” SNC versus “Bad scaled” SNC: average cut density for instance $p0201$

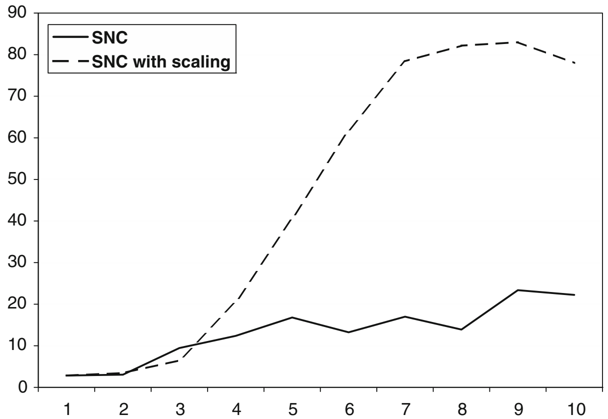

Fig. 7 “Classical” SNC versus “Bad scaled” SNC: average cardinality of $S(u, v)$ for instance $p0201$

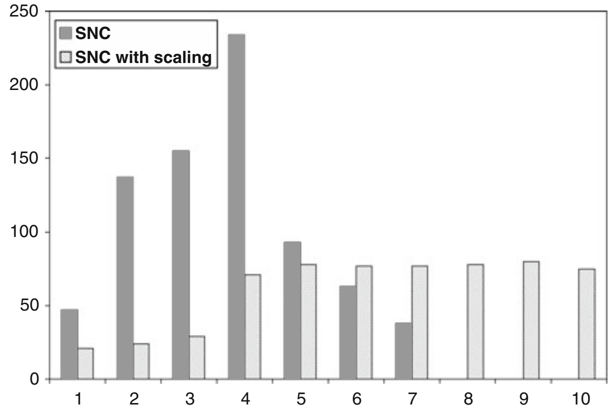

Fig. 8 "Classical" SNC versus "Bad scaled" SNC: cut rank for instance $p0201$

A bad example. Even for toy instances, the CGLP can have a hard time in finding a good disjunctive cut. This is illustrated by the following simple 2-dimensional case, where the optimal CGLP solution may correspond to a very weak cut.

Example 1 Consider the simple ILP

$$\begin{array}{lrrl} \min & -x_1 & -2x_2 & \\ & & & \\ (a1) & 4x_1 & -4x_2 & \geq -2 \\ (a2) & -2x_1 & -2x_2 & \geq -3 \\ (a3) & 8x_1 & -4x_2 & \geq -1 \\ (a4) & -x_1 & & \geq -1 \\ (a5) & & -kx_2 & \geq -k \\ (a6) & x_1 & & \geq \;\; 0 \\ (a7) & & x_2 & \geq \;\; 0 \end{array}$$

whose continuous relaxation, depicted in Fig. 9, has one of the constraints, namely (a5), scaled by a parameter $k > 0$: The optimal solution of the LP relaxation is $x^* = (\frac{1}{2}, 1)$ and three cuts can be derived from disjunction $x_1 \leq 0$ OR $x_1 \geq 1$, namely:

(c1) $2x_2 \leq 1$, corresponding to the basic solution of the CGLP (u_1, v_2, u_0, v_0), of value $z_1 = -\frac{2}{11}$, optimal for $k \leq 8$;
(c2) $-x_1+4x_2 \leq 1$, corresponding to the basic solution of the CGLP (u_3, v_2, u_0, v_0), of value $z_2 = -\frac{1}{6}$, never optimal.
(c3) $-x_1+2x_2 \leq 1$, corresponding to the basic solution of the CGLP (u_1, v_5, u_0, v_0), of value $z_3 = -\frac{k}{4+5k}$, optimal for $k \geq 8$.

So, depending on the value of k, the optimal CGLP solution gives weak cuts, either (c1) or (c3), whereas the facet-defining cut (c2) will never be selected. □

Note that the redundant (with respect to P) constraint (a5) is only used in the above example to show dependency on scaling. In fact, such a constraint can be removed without making cut (c2) optimal.

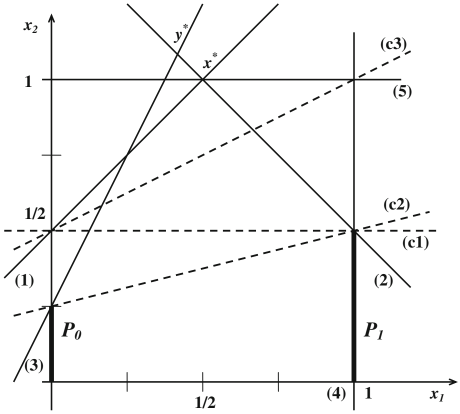

Fig. 9 Example 1 depicted

2.3 Comments

The examples above show clearly the following fact: even if the solution of the CGLP is a vertex, the corresponding disjunctive cut can be very weak. At first glance, this may be seen as a counter-intuitive result as one would expect that CGLP vertices correspond to facets of $conv(P_0 \cup P_1)$. This is, however, not the case, as discussed e.g. in Balas and Perregaard [7], since the CGLP is not defined in the "natural" reverse polar space (γ, γ_0) but in an enlarged space involving the Farkas variables explicitly. As a matter of fact, in the extended space $(\gamma, \gamma_0, u, v, u_0, v_0)$ there are several rays/vertices whose projection in the (γ, γ_0) space is nonextremal, therefore the corresponding cut can be obtained as the sum of other valid cuts and hence is dominated. By using software PORTA [16] we can get a clear picture of the situation in Example 1. In the natural polar space (γ, γ_0), the projected CGLP cone has only 4 extreme rays that correspond to the facets of $conv(P_0 \cup P_1)$. In space $(\gamma, \gamma_0, u, v, u_0, v_0)$, instead, the CGLP cone has 117 extreme rays that correspond to 117 vertices once normalization (11) is applied. Only 6 of these vertices correspond to violated constraints, and 3 of them correspond to the cuts depicted in Fig. 9. So, most CGLP vertices in the $(\gamma, \gamma_0, u, v, u_0, v_0)$ space correspond to very weak cuts, and the cut separation procedure can be in trouble in returning a facet-defining cut even in this toy example. As mentioned above, this is essentially due to the fact that the cut is separated in the extended space $(\gamma, \gamma_0, u, v, u_0, v_0)$, where a dominated cut could turn out not to be dominated in terms of the multipliers used for its generation. For instance, 3 extreme rays of the CGLP cone for Example 1 are reported below.

	γ_1	γ_2	γ_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	v_1	v_2	v_3	v_4	v_5	v_6	v_7	u_0	v_0
(r_1)	1	−4	−1	0	0	1	0	0	0	0	0	2	0	0	0	0	0	7	5
(r_2)	−1	0	−1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
(r_3)	0	−4	−2	1	0	0	0	0	0	0	0	2	0	0	0	0	0	4	4

In the (γ, γ_0) space, the third constraint is clearly dominated as it is just the sum of the previous ones, but there is no way to obtain ray r_3 as conic combination of rays r_1

and r_2 in the extended space, because of the Farkas components. The above drawback is even more evident in a slight modification of Example 1 where constraint (a3) is replaced by the constraint $4x_1 + 4x_2 \geq 3$, thus getting $P_0 = \emptyset$. Hence, $x_1 \geq 1$ itself is a valid cut, but not the best one for the corresponding CGLP.

3 Weak CGLP rays/vertices and dominated cuts

The previous examples show that some rays/vertices of the CGLP lead to weak cuts and should not be used. In the next section, we formally characterize those rays/vertices which correspond to cuts that are trivially dominated by other cuts associated with solutions of the same CGLP (Sect. 3.1). In Sect. 3.2, we give a heuristic procedure to strengthen cuts associated with dominated rays/vertices, whose practical effect is computationally investigated in Sect. 3.3.

3.1 Characterization

The first step to characterize weak rays/vertices is the following definition.

Definition 3 (Strictly dominated cuts) Let $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$ be a cut valid for $conv(P_0 \cup P_1)$ but not for P. If there exists another cut $\overline{\gamma} x \geq \overline{\gamma}_0$ valid for $conv(P_0 \cup P_1)$ such that $\{x \in P : \overline{\gamma} x \geq \overline{\gamma}_0\} \subsetneq \{x \in P : \widetilde{\gamma} x \geq \widetilde{\gamma}_0\}$, then the cut $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$ is said to be strictly dominated w.r.t. P.

Note that, in the above definition, the domination of cut $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$ only depends on a single other cut ($\overline{\gamma} x \geq \overline{\gamma}_0$).

Lemma 4 *Let $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$ be a valid cut for $conv(P_0 \cup P_1)$ such that $\widetilde{P} := \{x \in P : \widetilde{\gamma} x \geq \widetilde{\gamma}_0\} \subsetneq P$, and assume $\widetilde{P}$ full dimensional. If there exists another cut $\overline{\gamma} x \geq \overline{\gamma}_0$ valid for $conv(P_0 \cup P_1)$ and such that $\widetilde{\gamma} = \overline{\gamma} + \mu A, \widetilde{\gamma}_0 = \overline{\gamma}_0 + \mu b$ for a certain $\mu \in \mathbb{R}^m_+ \backslash \{0\}$, then $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$ is strictly dominated w.r.t. P.*

Proof Define $\overline{P} := \{x \in P : \overline{\gamma} x \geq \overline{\gamma}_0\}$. By definition, $x \in P$ and $\overline{\gamma} x \geq \overline{\gamma}_0$ imply $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$, hence $\overline{P} \subseteq \widetilde{P}$. We need to show that the above inclusion is always strict. Indeed, let $\widetilde{F} := \{x \in P : \widetilde{\gamma} x = \widetilde{\gamma}_0\}$ denote the face of $\widetilde{P}$ induced by $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$, and consider any given $h \in \{1, \ldots, m\}$ such that $\mu_h > 0$. Since $\widetilde{P}$ is full dimensional, there exists $\hat{x} \in \widetilde{F}$ such that $a_h \hat{x} > b_h$ (otherwise $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$ would be a positive multiple of $a_h x \geq b_h$, impossible since we are assuming $\widetilde{P} \subsetneq P$). Hence $\overline{\gamma} \hat{x} - \overline{\gamma}_0 = (\widetilde{\gamma} \hat{x} - \widetilde{\gamma}_0) - \mu(A\hat{x} - b) \leq -\mu_h(a_h \hat{x} - b_h) < 0$, i.e., $\hat{x} \in \widetilde{P} \backslash \overline{P}$. □

For any feasible solution $(\gamma, \gamma_0, u, v, u_0, v_0)$ of (4)–(8), we denote by $S(u) := \{i \in \{1, \ldots, m\} : u_i > 0\}$ and $S(v) := \{i \in \{1, \ldots, m\} : v_i > 0\}$ the support of vectors u and v, respectively. It is not difficult to show that in any extreme ray of (4)–(8) yielding a cut nonvalid for P, both u_0 and v_0 are strictly positive, while $S(u)$ and $S(v)$ are disjoint. This property is also inherited by the vertices of the CGLP with normalization (11) (see, Balas and Perregaard [8]). We next give a characterization of the extreme rays/vertices of the CGLP that lead to strictly dominated cuts according to Definition 3.

Theorem 5 *Assume $conv(P_0 \cup P_1)$ full dimensional. Let $(\widetilde{\gamma}, \widetilde{\gamma}_0, \widetilde{u}, \widetilde{v}, \widetilde{u}_0, \widetilde{v}_0)$ be an extreme ray of the CGLP cone (4)–(8) corresponding to a cut $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ nonvalid for P. Then $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ is strictly dominated w.r.t. P if and only if there exists a feasible solution $(\widetilde{\gamma}, \widetilde{\gamma}_0, \hat{u}, \hat{v}, \hat{u}_0, \hat{v}_0)$ of (4)–(8) such that $S(\hat{u}) \cap S(\hat{v}) \neq \emptyset$.*

Proof We first prove the if condition. Given a feasible solution $(\widetilde{\gamma}, \widetilde{\gamma}_0, \hat{u}, \hat{v}, \hat{u}_0, \hat{v}_0)$ of (4)–(8) such that $S(\hat{u}) \cap S(\hat{v}) \neq \emptyset$, define $\mu = \min\{\hat{u}, \hat{v}\}$ (componentwise) and note that $\mu_i > 0$ for any $i \in S(\hat{u}) \cap S(\hat{v})$. Then, define $\overline{u} = \hat{u} - \mu \geq 0, \overline{v} = \hat{v} - \mu \geq 0, \overline{\gamma} = \widetilde{\gamma} - \mu A, \overline{\gamma}_0 = \widetilde{\gamma}_0 - \mu b$. Since $(\overline{\gamma}, \overline{\gamma}_0, \overline{u}, \overline{v}, \hat{u}_0, \hat{v}_0)$ is a feasible solution of (4)–(8), the cut $\overline{\gamma}x \geq \overline{\gamma}_0$ is valid for $conv(P_0 \cup P_1)$ and dominates $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ w.r.t. P from Lemma 4. Concerning the only if condition, assume $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ to be strictly dominated w.r.t. P by $\overline{\gamma}x \geq \overline{\gamma}_0$, and let $(\overline{\gamma}, \overline{\gamma}_0, \overline{u}, \overline{v}, \overline{u}_0, \overline{v}_0)$ be a feasible solution of (4)–(8) yielding the dominating cut. Then, there exist $\mu \in \mathbb{R}^m_+ \setminus \{0\}$ and $\mu_0 > 0$ such that $\widetilde{\gamma} = \mu A + \mu_0 \overline{\gamma}, \widetilde{\gamma}_0 = \mu b + \mu_0 \overline{\gamma}_0$. Hence $(\widetilde{\gamma}, \widetilde{\gamma}_0, \hat{u}, \hat{v}, \hat{u}_0, \hat{v}_0)$ is a feasible solution of (4)–(8) yielding the dominated cut, where $\hat{u} = \mu + \mu_0 \overline{u}, \hat{v} = \mu + \mu_0 \overline{v}, \hat{u}_0 = \mu_0 \overline{u}_0, \hat{v}_0 = \mu_0 \overline{v}_0$ and $S(\hat{u}) \cap S(\hat{v}) \neq \emptyset$. □

Corollary 6 *Let $(\gamma, \gamma_0, u, v, u_0, v_0)$ be an optimal solution of the CGLP with normalization (11), yielding a cut violated by x^* (i.e., $\gamma x^* - \gamma_0 < 0$). Then $S(u) \cap S(v) = \emptyset$.*

Note that the above corollary holds even if the CGLP cone is truncated with a more general normalization than (11), e.g., the one to be discussed in Sect. 5.

3.2 Strengthening

Theorem 5 above suggests a way to strengthen disjunctive cuts arising from weak rays/vertices of the CGLP. Let us assume to be given a vertex $(\widetilde{\gamma}, \widetilde{\gamma}_0, \widetilde{u}, \widetilde{v}, \widetilde{u}_0, \widetilde{v}_0)$ of the CGLP associated with a valid disjunction (2) and truncated by any normalization, e.g., (10) or (11). Consider the following LP:

$$\max \quad 1^T \mu \tag{12}$$
$$\widetilde{\gamma} = (u + \mu)A - u_0 \pi \tag{13}$$
$$\widetilde{\gamma} = (v + \mu)A + v_0 \pi \tag{14}$$
$$\widetilde{\gamma}_0 = (u + \mu)b - u_0 \pi_0 \tag{15}$$
$$\widetilde{\gamma}_0 = (v + \mu)b + v_0(\pi_0 + 1) \tag{16}$$
$$u, v, \mu, u_0, v_0 \geq 0, \tag{17}$$

where $(\widetilde{\gamma}, \widetilde{\gamma}_0)$ is fixed. Assuming $conv(P_0 \cup P_1)$ to be full dimensional, it is not difficult to see that the above LP is always bounded and the optimal solution value is greater than 0 if and only if $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ is strictly dominated w.r.t. P. Moreover, in this case any optimal solution $(\overline{u}, \overline{v}, \overline{u}_0, \overline{v}_0, \overline{\mu})$ of (12)–(17) yields a valid cut $\overline{\gamma}x \geq \overline{\gamma}_0$ for $conv(P_0 \cup P_1)$, computed as $\overline{\gamma} = \overline{u}A - \overline{u}_0 \pi = \overline{v}A + \overline{v}_0 \pi, \overline{\gamma}_0 = \overline{u}b - \overline{u}_0 \pi_0 = \overline{v}b + \overline{v}_0(\pi_0 + 1)$, which strictly dominates $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ w.r.t P. However, the LP (12)–(17) involves three sets of Farkas multipliers and might be quite time consuming in practice.

A practical heuristic way to look for a dominating cut is based on the following *Cut Dominating LP* (CDLP), using only two sets of Farkas multipliers:

$$\text{(CDLP)} \quad \max \ z = \sum_{i \in S(\widetilde{v})} u_i + \sum_{i \in S(\widetilde{u})} v_i \tag{18}$$

$$\widetilde{\gamma} = uA - u_0\pi \tag{19}$$

$$\widetilde{\gamma} = vA + v_0\pi \tag{20}$$

$$\widetilde{\gamma}_0 = ub - u_0\pi_0 \tag{21}$$

$$\widetilde{\gamma}_0 = vb + v_0(\pi_0 + 1) \tag{22}$$

$$u, v, u_0, v_0 \geq 0. \tag{23}$$

Let us assume the above CDLP to be bounded and consider an optimal solution (u^*, v^*, u_0^*, v_0^*) of value z^*, yielding the same cut $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ as $(\widetilde{u}, \widetilde{v}, \widetilde{u}_0, \widetilde{v}_0)$. For any $\alpha \in [0, 1]$, the convex combination of (u^*, v^*, u_0^*, v_0^*) and $(\widetilde{u}, \widetilde{v}, \widetilde{u}_0, \widetilde{v}_0)$

$$\begin{aligned} \hat{u} &= \alpha\widetilde{u} + (1-\alpha)u^*, \quad \hat{v} = \alpha\widetilde{v} + (1-\alpha)v^*, \\ \hat{u}_0 &= \alpha\widetilde{u} + (1-\alpha)u_0^*, \quad \hat{v}_0 = \alpha\widetilde{v}_0 + (1-\alpha)v_0^*, \end{aligned} \tag{24}$$

still yields cut $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$. However, in case $z^* > 0$ we have $S(\hat{u}) \cap S(\hat{v}) \neq \emptyset$, i.e., we have obtained the same cut from two sets of nondisjoint multipliers. Hence, a valid disjunctive cut $\overline{\gamma}x \geq \overline{\gamma}_0$ which dominates $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ w.r.t. P can be computed through Theorem 5 as:

$$\begin{aligned} &\overline{\mu} = \min\{\hat{u}, \hat{v}\}, \quad \overline{u} = \hat{u} - \overline{\mu}, \quad \overline{v} = \hat{v} - \overline{\mu}, \\ &\overline{u}_0 = \hat{u}_0, \quad \overline{v}_0 = \hat{v}_0, \\ &\overline{\gamma} = \overline{u}A - \overline{u}_0\pi, \quad \overline{\gamma} = \overline{v}A + \overline{v}_0\pi, \\ &\overline{\gamma}_0 = \overline{u}b - \overline{u}_0\pi_0, \quad \overline{\gamma}_0, = \overline{v}b + \overline{v}_0(\pi_0 + 1). \end{aligned} \tag{25}$$

Clearly, the dominance might be not strict if $conv(P_0 \cup P_1)$ is not full dimensional.

3.3 Empirical analysis

In order to understand how much we can improve on the disjunctive cuts obtained by solving the CGLP with SNC, we performed the following experiment. For any violated cut $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ separated by solving the CGLP with SNC normalization (11), we try to strengthen it by solving the corresponding CDLP (18)–(23). If $z^* > 0$, then we compute the new dominating cut $\overline{\gamma}x \geq \overline{\gamma}_0$ by using (24), with $\alpha = 1/2$, and (25), and we replace the original cut $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$ by the dominating one. Otherwise, if either $z^* = 0$ or CDLP turns out to be unbounded, we keep the original cut $\widetilde{\gamma}x \geq \widetilde{\gamma}_0$.

The computational results reported in Tables 3 and 4 compare the cuts obtained by using the SNC normalization with those strengthened through the additional solution of the CDLP. In order to get a better understanding of the impact of the strengthening, we first solved both CGLP and CDLP *without* the projection onto the support

Table 3 SNC Normalization versus SNC Normalization + CDLP

Instance	SNC normalization			SNC normalization + CDLP				
	# cuts	%gap	Time	# cuts	# dom	# unb	%gap	Time
bell3a	71	70.74	0.1	71	19	14	70.74	0.4
bell5	188	94.12	0.5	186	11	125	94.31	1.2
blend2	197	30.49	4.7	210	54	2	32.72	12.0
flugpl	93	18.34	0.1	91	26	0	18.36	0.1
gt2	218	94.13	1.2	200	141	1	94.49	2.8
lseu	171	42.46	0.5	191	68	4	42.51	0.7
*markshare1	77	0.00	0.1	75	0	75	0.00	0.2
mod008	107	15.46	3.9	112	27	0	15.84	6.0
p0033	116	57.25	0.1	106	55	8	57.30	0.2
p0201	692	92.53	46.4	750	38	622	98.97	70.3
rout	349	29.46	80.5	351	159	142	30.93	118.1
*stein27	251	0.00	0.6	248	21	0	0.00	1.3
vpm1	267	50.62	2.3	275	8	115	59.91	5.9
vpm2	390	74.73	7.5	397	84	130	75.71	14.5
avg.	238.250	55.861	12.317	245.000			57.649	19.350

No projection (and no Balas–Jeroslow strengthening)

Table 4 SNC normalization versus SNC normalization + CDLP

Instance	SNC normalization			SNC normalization + CDLP				
	# cuts	%gap	Time	# cuts	# dom	# unb	%gap	Time
bell3a	71	70.74	0.1	69	11	23	70.74	0.2
bell5	172	96.16	0.3	179	38	63	96.16	0.7
blend2	215	33.45	0.7	225	77	30	33.66	5.1
flugpl	92	18.36	0.1	90	29	0	18.59	0.1
gt2	151	96.19	0.2	157	50	55	96.19	0.7
lseu	179	81.09	0.2	171	9	135	86.04	0.4
*markshare1	80	0.00	0.0	80	3	46	0.00	0.1
mod008	100	31.46	0.1	98	46	9	36.33	0.2
p0033	104	70.98	0.1	113	14	71	75.85	0.2
p0201	669	100.00	10.9	674	1	663	100.00	17.0
rout	603	47.91	38.0	613	6	600	49.50	55.7
*stein27	251	0.00	0.5	252	14	0	0.00	1.1
vpm1	298	57.88	1.2	255	25	23	58.97	2.1
vpm2	400	75.11	4.0	401	133	29	75.76	6.0
avg.	254.500	64.944	4.658	253.75			66.483	7.367

CGLP and CDLP solved projected onto the x^* support. Balas–Jeroslow strengthening applied before and after CDLP of x^* (Table 3). Indeed, solving the problem on the complete model guarantees that the strengthened cut dominates the original one – strict domination in the full dimensional case. Since the computing time in the complete variable space is not negligible, we also solved the problem in the projected space (Table 4). However, a domination on the support might not correspond to a dominated cut once the cut is lifted outside the support. In fact, a cut which is stronger in the support might be weaker overall. Hence, to limit such a phenomenon, in the experiments in Table 4 we strengthen the cut by the Balas-Jeroslow procedure [6] before defining and solving the CDLP. Of course, both the original and the dominating cuts are also strengthened afterwards.

Tables 3 and 4 report the number of separated cuts, the percentage gap closed (within ten rounds) and the computing time spent on separation (expressed in seconds on an Intel Pentium M 1.86 GHz processor). In addition, for the strengthened version of the cuts we also report how many times the CDLP returns $z^* > 0$ (column '# dom') and the number of times in which CDLP was instead unbounded (column '# unb'). Separation time of the strengthened version includes CGLP solution time.

Both Tables 3 and 4 show that the CDLP is indeed effective at changing the disjunctive cuts obtained using the SNC normalization. In general, the procedure is computationally rather cheap and allows an improvement in the %gap closed which is sometimes non-negligible.

Of course, the same CDLP can be constructed and solved to strengthen a disjunctive cut obtained by using any normalization, e.g., the trivial one (10). Indeed, we also tested it to strengthen classical GMIs. Due to the lack of space we do not report full tables of results but the improvement in the %gap closed goes from 46.463% to 49.428% with an additional time of 4.250 CPU seconds on average. This improvement is obtained with the same setting of Table 4 above, i.e., with both CGLP and CDLP solved projected onto the x^* support and Balas-Jeroslow strengthening applied before and after CDLP solution.

4 Redundancy hurts

Loosely speaking, a *redundant* constraint for a polyhedron is a constraint whose removal does not enlarge the polyhedron itself. By Farkas lemma, a constraint $a_i x \geq b_i$ is said to be *redundant* for $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ if there exist $\delta \geq 0$ and $\lambda_I \in \mathbb{R}^{m-1}_+$ such that $a_i = \lambda_I A_I$ and $\lambda_I b_I = b_i + \delta$, where A_I (resp., b_I) denotes the submatrix of A (resp., subvector of b) induced by the row index set $I = \{1, \dots, m\} \backslash \{i\}$. Redundancy is *strict* if $\delta > 0$.

In the attempt to find a way to get rid of the "weak vertices" in the CGLP, we looked for more combinatorial properties. A more careful analysis of Example 1 reveals a more general property that allows one to classify as "bad" certain constraints. Indeed, consider the role of constraint (a1) with respect to the left-branch polytope P_0. This constraint is clearly redundant for P_0 (note that this is not the case if the original P is considered). However, if constraint (a1) participates with a positive multiplier to the definition of the disjunctive cut whereas constraint (a3) does not (i.e., if $u_1 > 0$ and $u_3 = 0$), then the cut itself has to be valid for the point $x_1 = 0$, $x_2 = 1/2$ and cannot be "pushed" any further inside P_0. This is precisely what happens for the weak cuts (c3) and (c1), that cannot be supporting for P_0 precisely because of the bad choice $u_1 > 0$.

The role of redundancy is formally stated as follows.

Proposition 7 *If a constraint that is strictly redundant for P_0 (resp. P_1) is used in the cut derivation with a nonzero multiplier, then the resulting disjunctive cut is nonsupporting in P_0 (resp. P_1).*

Proof Let $(\widetilde{\gamma}, \widetilde{\gamma}_0, \widetilde{u}, \widetilde{u}_0, \widetilde{v}, \widetilde{v}_0)$ be a feasible solution of the CGLP, with $\widetilde{u}_i > 0$, and assume that constraint i is strictly redundant for P_0. Then there exists $(\lambda, \lambda_0, \delta) \in \mathbb{R}_+^{m+1}$, with $\delta > 0$ such that $a_i = \lambda_I A_I - \lambda_0 \pi$ and $b_i = \lambda_I b_I - \lambda_0 \pi_0 - \delta$. By using (4) and (6), we get

$$\widetilde{\gamma} = \widetilde{u}_I A_I + \widetilde{u}_i a_i - \widetilde{u}_0 \pi = (\widetilde{u}_I + \widetilde{u}_i \lambda_I) A_I - (\widetilde{u}_0 + \widetilde{u}_i \lambda_0)\pi$$
$$\widetilde{\gamma}_0 = \widetilde{u}_I b_I + \widetilde{u}_i b_i - \widetilde{u}_0 \pi_0 = (\widetilde{u}_I + \widetilde{u}_i \lambda_I) b_I - (\widetilde{u}_0 + \widetilde{u}_i \lambda_0)\pi_0 - \widetilde{u}_i \delta.$$

Thus, for each $x \in P_0$ we have $\widetilde{\gamma} x - \widetilde{\gamma}_0 = (\widetilde{u}_I + \widetilde{u}_i \lambda_I)(A_I x - b_I) - (\widetilde{u}_0 + \widetilde{u}_i \lambda_0)(\pi x - \pi_0) + \widetilde{u}_i \delta \geq \widetilde{u}_i \delta > 0$, and this shows that cut $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$ is nonsupporting in P_0. In the same way it can be shown that if $v_h > 0$ for a constraint h strictly redundant for P_1, then the cut $\widetilde{\gamma} x \geq \widetilde{\gamma}_0$ does not support P_1. □

By definition, a redundant constraint for P_0 or P_1 can be obtained as a conic combination of other constraints. If the sum of the multipliers in the conic combination is greater than 1, then using a redundant constraint is cheaper [with respect to normalization (11)] than using the constraints that generate it, hence a redundant constraint can in fact be preferred by the CGLP. This is formally proved by the following theorem dealing with redundancy for P_0 (the case dealing with P_1 being perfectly analogous).

Theorem 8 *Assume that constraint $a_i x \geq b_i$ is redundant for P_0 as conic combination of $A_I x \geq b_I$, $-\pi x \geq -\pi_0$ with multipliers $(\lambda_I, \lambda_0) \in \mathbb{R}_+^m$, and let $(\overline{\gamma}, \overline{\gamma}_0, \overline{u}, \overline{u}_0, \overline{v}, \overline{v}_0)$ be a feasible solution of the CGLP with normalization (11), such that $\overline{\gamma} x^* < \overline{\gamma}_0$ and $\overline{u}_i > 0$. Then there exist $\theta > 0$ and a feasible solution $(\widetilde{\gamma}, \widetilde{\gamma}_0, \widetilde{u}, \widetilde{u}_0, \widetilde{v}, \widetilde{v}_0)$ of the CGLP with normalization (11) such that $\widetilde{u}_i = 0$, $\widetilde{\gamma} := \overline{\gamma}/\theta$, $\widetilde{\gamma}_0 = \overline{\gamma}_0/\theta$, $\overline{\gamma} x^* - \overline{\gamma}_0 = \theta(\widetilde{\gamma} x^* - \widetilde{\gamma}_0)$, and $\theta > 1$ if and only if $1\lambda_I + \lambda_0 > 1$.*

Proof Since $(\overline{\gamma}, \overline{\gamma}_0, \overline{u}, \overline{u}_0, \overline{v}, \overline{v}_0)$ is feasible for the CGLP with normalization (11), writing $a_i x \geq b_i$ in terms of the multipliers (λ_I, λ_0) one gets

$$\overline{\gamma} = (\overline{u}_I + \overline{u}_i \lambda_I) A_I - (\overline{u}_0 + \overline{u}_i \lambda_0)\pi = \overline{v} A + \overline{v}_0 \pi$$
$$\overline{\gamma}_0 = (\overline{u}_I + \overline{u}_i \lambda_I) b_I - (\overline{u}_0 + \overline{u}_i \lambda_0)\pi_0 = \overline{v} b + \overline{v}_0 (\pi_0 + 1)$$

while from the normalization condition $1\overline{u} + 1\overline{v} + \overline{u}_0 + \overline{v}_0 = 1$ one obtains

$$\theta := 1(\overline{u}_I + \overline{u}_i \lambda_I) + (\overline{u}_0 + \overline{u}_i \lambda_0) + 1\overline{v} + \overline{v}_0,$$

which is then simplified as $\theta = 1 + \overline{u}_i (1\lambda_I + \lambda_0 - 1)$.

Since cut $\overline{\gamma}x \geq \overline{\gamma}_0$ is violated, one must have $\overline{u}_0 + \overline{v}_0 > 0$, hence $\theta > 0$ holds. Therefore, one can define the nonnegative quantities $\widetilde{u}_I := (\overline{u}_I + \overline{u}_i\lambda_I)/\theta, \widetilde{u}_i = 0, \widetilde{u}_0 := (\overline{u}_0 + \overline{u}_i\lambda_0)/\theta, \widetilde{v} = \overline{v}/\theta, \widetilde{v}_0 = \overline{v}_0/\theta, \widetilde{\gamma} := \overline{\gamma}/\theta, \widetilde{\gamma}_0 = \overline{\gamma}_0/\theta$, thus getting a feasible solution of the CGLP satisfying (11) and such that $\overline{\gamma}x^* - \overline{\gamma}_0 = \theta(\widetilde{\gamma}x^* - \widetilde{\gamma}_0)$. Moreover, being $\overline{u}_i > 0$, one has $\theta > 1$ if and only if $1\lambda + \lambda_0 > 1$. □

The above theorem shows that redundant constraints do not introduce new cuts, but just scaled copies of already-existing cuts that may have a better objective function (violation). Loosely speaking, redundant constraints can "trick" normalization (11), in the sense that they can create vertices of the CGLP corresponding to scaled copies of cuts that are strictly dominated but more attractive (i.e., with a better objective function value) than the dominating ones.

A natural way to cope with redundancy is to just eliminate the redundant constraints from the CGLP, or equivalently to fix their Farkas multipliers to zero. In Example 1, the CGLP without redundant constraints has only 9 extreme rays and 9 vertices (instead of 117), and only one of them corresponds to a violated constraint – namely, the facet-defining cut (c2). At a first glance, this example seems to suggest that only *strictly redundant* (i.e., nonsupporting) constraints should be avoided in the cut generation. However, redundant constraints should be avoided even in case they are supporting, as shown by the example below.

Example 2 For the simple ILP

$$\begin{array}{lrrrcr}
\min & -x_1 & -2x_2 & +10x_3 & & \\
(a1) & 4x_1 & -4x_2 & & \geq & -2 \\
(a2) & -2x_1 & -2x_2 & -x_3 & \geq & -3 \\
(a3) & 3x_1 & -2x_2 & -x_3 & \geq & -1 \\
(a4) & x_1 & & & \geq & 0 \\
(a5) & & x_2 & & \geq & 0 \\
(a6) & & & x_3 & \geq & 0
\end{array}$$

only two cuts violated by the optimal solution of the LP relaxation $x^* = (\frac{1}{2}, 1, 0)$ can be derived from disjunction $x_1 \leq 0$ OR $x_1 \geq 1$, namely:

(c1) $2x_2 \leq 1$, corresponding to the basic solution of the CGLP $(u_1, v_2, v_6, u_0, v_0)$, of value $z_1 = -\frac{2}{13}$ (optimal);

(c2) $2x_2 + x_3 \leq 1$, corresponding to the basic solution of the CGLP $(u_3, v_2, v_6, u_0, v_0)$, of value $z_2 = -\frac{1}{7}$ (nonoptimal).

$Conv(P_0 \cup P_1)$ has 6 vertices, namely $V_1 = (0, 0, 0)$, $V_2 = (0, \frac{1}{2}, 0)$, $V_3 = (0, 0, 1)$, $V_4 = (\frac{3}{2}, 0, 0)$, $V_5 = (1, \frac{1}{2}, 0)$, and $V_6 = (1, 0, 1)$. In the reverse polar space (γ, γ_0), the projected CGLP cone has only 5 extreme rays that correspond to the facets of $conv(P_0 \cup P_1)$. In space $(\gamma, \gamma_0, u, v, u_0, v_0)$, instead, the CGLP cone has 33 extreme rays that correspond to 33 vertices once normalization (11) is applied. In the optimal basis, constraint (a1) (which is redundant but supporting for P_0) is used with $u_1 > 0$, and the corresponding cut (c1) supports both P_0 and P_1 in V_2 and V_5, respectively, but it is not facet-defining. The CGLP without redundant constraints (in particular without (a1)) has only 10 extreme rays and 10 vertices (instead of 33), and only one of them corresponds to a violated constraint—namely, the facet-defining cut (c2). Note that cut (c2) dominates cut (c1) and is facet-defining since it supports $conv(P_0 \cup P_1)$ in the 3 affinely independent vertices V_2, V_3, and V_5. For illustration purposes, 3 extreme rays r_1–r_3 and a nonextremal direction α of the CGLP cone are reported below.

	γ_1	γ_2	γ_3	γ_0	u_1	u_2	u_3	u_4	u_5	u_6	v_1	v_2	v_3	v_4	v_5	v_6	u_0	v_0
(r_1)	0	−2	−1	−1	0	0	1	0	0	0	0	1	0	0	0	0	3	2
(r_2)	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
(α)	0	−2	0	−1	0	0	1	0	0	1	0	1	0	0	0	1	3	2
(r_3)	0	−2	0	−1	1/2	0	0	0	0	0	0	1	0	0	0	1	2	2

The weak cut (c1) is strictly dominated w.r.t. P by (c2), as the vector α is just the sum of the extreme rays r_1 and r_2, the latter corresponding to the original constraint $x_3 \geq 0$. However, the redundant constraint (a1) creates an extremal copy of the weak cut – the extreme ray r_3 – which turns out to be the optimal vertex once normalization (11) is applied. □

The previous discussion shows that extreme rays of the CGLP cone in the extended space $(\gamma, \gamma_0, u, v, u_0, v_0)$ may be nonextremal when projected onto the (γ, γ_0) space, and redundant constraints can add to the cone several extreme-rays corresponding to very weak cuts. The SNC normalization (11) simply maps extreme rays to vertices, and creates a possibly "wrong" ranking among the vertices. Unfortunately, as far as we know no normalization equation is able to truncate the CGLP cone so as to guarantee that an optimal CGLP vertex in the extended space remains a vertex when projected in the (γ, γ_0) space. For instance, consider normalizations of the form

$$\gamma(q - x^*) = 1, \tag{26}$$

which have been deeply investigated in Bonami [12]. Balas and Perregaard [7] proved that, if $q \in conv(P_0 \cup P_1)$, then the CGLP truncated with (26) has a finite optimum and that there *exists* an optimal vertex of the resulting polyhedron in the extended space whose projection in the natural reverse polar space (γ, γ_0) remains extremal. However, this does not imply that any optimal vertex in the extended space is a vertex in the projected space – hence even normalization (26) could not help in finding a facet-defining cut.

Example 3 (Example 2 continued) Consider again the simple ILP discussed in Example 2. If the corresponding CGLP cone is truncated with normalization (26), with $q = (0, 0, 0)$, the resulting polyhedron has 60 extreme rays and 20 vertices. As before, only two vertices correspond to violated cuts, namely:

i) the basic solution $(u_1, v_2, v_6, u_0, v_0)$, of value $z_1 = -\frac{1}{2}$ (optimal), corresponding to the weak cut (c1);
ii) the basic solution $(u_3, v_2, v_6, u_0, v_0)$, of value $z_2 = -\frac{1}{2}$ (optimal), corresponding to the facet-defining cut (c2).

So, the separation procedure could select the weak cut (c1), since the choice of q makes (c1) and (c2) completely equivalent in terms of objective function. □

4.1 Empirical analysis

In a preliminary set of experiments we eliminated redundant constraints in a trivial way (i.e., by solving LPs) before solving the CGLP. To get a clearer picture, we did not project the separation problem onto the support of x^* since such a projection makes the definition of what is redundant and what is not less clear. In summary, the results show that removing redundant constraints is indeed very useful. More precisely, we report an average improvement in the percentage gap closed of around 2.5%, and only for two problems, namely `bell5` and `gt2`, the "Classical" SNC is slightly better than the "No redundancy" SNC version, while for some single problems the improvement is substantial, up to 13% for instance `p0033`.

However, a more meaningful experiment implies projecting the separation problem onto the support of x^* which has of course the advantage of dealing with a problem of smaller size. On the other hand, according to our experience the projection can enlarge the set of redundant constraints in a way that decreases the positive effects associated with their removal. A possible explanation of this behavior is that projection may hide the redundancy of some bound constraints, hence weakening the final disjunctive cut. Indeed, consider a variable x_k restricted to being nonnegative and such that $x_k^* = 0$. If x_k is projected away with the aim of computing coefficient γ_k afterwards through (9), then we lose any control on the Farkas variables associated with the constraint $x_k \geq 0$, say $u_{i(k)}$ and $v_{i(k)}$. In fact, if it happens that constraint $x_k \geq 0$ is redundant, it is very useful to keep explicitly constraints $\gamma_k = uA_k - u_0\pi_k = vA_k + v_0\pi_k$ in the CGLP and to impose the additional requirement $u_{i(k)} = 0$ and/or $v_{i(k)} = 0$.

As the above property seems to be crucial for the variable bounds, we defined an *extended support* of x^* by avoiding projecting away any variable whose bound condition is (tight in x^* and) redundant. The results when using the support and the extended support are given in Table 5.

The table reports the same figures as Table 1, plus a column which indicates the average percentage of the x variables which are kept in the (extended) support (column '% supp'). The first part of the table shows that the gain due to the redundancy removal is lost if the CGLP is projected onto the x^* support (the average gap closed of 56.873% deteriorates to 54.269%), thus confirming our intuition about the smaller precision of the redundancy test in such a case. However, the situation is totally recovered using the extended support, as shown in the second part of the table. Indeed, the percentage value 56.873 improves to 58.793, and the average size of the support does not increase much (from 50.268 to 53.529%). The only large increase in the size of the CGLP arises for problem `rout`, which is in fact a very instructive case: the "Classical" SNC closed 24.26% of the gap, the "No redundancy" SNC version on the support closes only 6.56%, while in the extended support the situation is totally recovered (and improved) with 30.88% gap closed. For such a particular instance the extended support size is substantially different from the support size, namely 69.46% with respect to 42.19%. Our interpretation is the following: in order to forbid the use of some variable bounds in the derivation of the cut we have to enlarge the support considerably (half of the projected variables are re-inserted) with the overall effect of generating much stronger cuts.

Table 5 "Classical" SNC versus "No redundancy" SNC with cuts separated projected on the support

Instance	"Classical" SNC				"No redundancy" support			
	# cuts	%gap	%supp	$\|S\|$	# cuts	%gap	%supp	$\|S\|$
bell3a	71	70.74	69.25	43.72	88	70.74	69.32	44.82
bell5	178	94.29	72.69	11.75	207	94.62	72.88	13.32
blend2	192	30.51	53.06	8.10	200	30.99	53.54	10.84
flugpl	92	18.36	86.11	5.85	93	18.94	86.11	5.89
gt2	196	93.46	18.30	10.28	191	94.13	18.14	10.58
lseu	196	41.33	29.44	9.17	191	40.16	27.08	12.28
*markshare1	74	0.00	11.94	1.39	130	0.00	13.39	2.56
mod008	139	17.05	4.51	12.41	136	17.70	4.42	12.17
p0033	113	67.86	55.76	4.81	106	70.32	55.76	5.74
p0201	767	93.82	45.02	13.43	873	81.59	43.43	25.83
rout	434	24.26	42.19	68.07	355	6.56	38.11	58.23
*stein27	252	0.00	93.70	6.53	252	0.00	93.70	6.68
vpm1	263	55.84	62.14	5.39	275	50.18	62.25	6.30
vpm2	403	74.96	64.74	17.27	377	75.30	65.08	18.10
avg.	253.667	56.873	50.268	17.521	257.667	54.269	49.677	18.675
Instance	"Classical" SNC				"No redundancy" ext. support			
	# cuts	%gap	%supp	$\|S\|$	# cuts	%gap	%supp	$\|S\|$
bell3a	71	70.74	69.25	43.72	54	70.74	65.61	44.60
bell5	178	94.29	72.69	11.75	180	94.29	71.64	11.99
blend2	192	30.51	53.06	8.10	193	30.53	53.99	8.34
flugpl	92	18.36	86.11	5.85	93	18.86	86.29	5.95
gt2	196	93.46	18.30	10.28	187	93.88	20.00	13.10
lseu	196	41.33	29.44	9.17	178	43.45	29.41	9.08
*markshare1	74	0.00	11.94	1.39	77	0.00	12.59	1.69
mod008	139	17.05	4.51	12.41	157	19.13	5.85	14.43
p0033	113	67.86	55.76	4.81	146	70.29	58.84	5.89
p0201	767	93.82	45.02	13.43	769	100.00	48.93	13.39
rout	434	24.26	42.19	68.07	353	30.88	69.46	140.29
*stein27	252	0.00	93.70	6.53	251	0.00	93.61	7.13
vpm1	263	55.84	62.14	5.39	259	57.63	65.18	6.60
vpm2	403	74.96	64.74	17.27	373	75.84	67.15	17.71
avg.	253.667	56.873	50.268	17.521	245.167	58.793	53.529	24.281

5 An effective (and fast) normalization for set covering

As shown in the previous sections, the standard normalization has the main advantage of generating low-rank inequalities, which is in general a desirable property. As a matter of fact, it has been recently showed that rank-1 inequalities on general disjunctions are able to close a large portion of the integrality gap (see, e.g., Fischetti and Lodi [18], Balas and Saxena [9], Dash, Günlük and Lodi [17]). When normalization (11) is applied, the norm of the separated cuts tends to be smaller with respect to the constraints used for their generation, and small-norm constraints are implicitly penalized by the normalization itself. Thus, high-rank constraints are selected in the cut derivation only if needed, hence generating weak cuts does not hurt the overall separation procedure in that these cuts are less likely to be used in the next iterations. However, as stated in Sect. 4, the standard normalization creates a ranking among the CGLP vertices which depends on the scaling of the constraints, i.e., the overall separation procedure is heavily affected by the scaling of the constraints in the original formulation. To overcome the latter drawback, one can replace the standard normalization with the following *Euclidean Normalization* (EN):

$$\sum_{i=1}^{m} \|a_i\| u_i + \sum_{i=1}^{m} \|a_i\| v_i + \|\pi\| u_0 + \|\pi\| v_0 = 1, \tag{27}$$

where $\|t\|$ denotes the Euclidean norm of vector t.

Lemma 9 *Let $\widetilde{A}x \geq \widetilde{b}$ be a scaled copy of system $Ax \geq b$ where, for all $i \in \{1, \ldots, m\}$, $\widetilde{a}_i := a_i/K_i$ and $\widetilde{b}_i := b_i/K_i$, with $K_i > 0$. For any solution of the CGLP with normalization (27) corresponding to a cut $\gamma x \geq \gamma_0$, there exists a solution of the CGLP associated with the system $\widetilde{A}x \geq \widetilde{b}$, still with normalization (27), corresponding to the same cut.*

Proof Let $(\gamma, \gamma_0, u, v, u_0, v_0)$ be a solution of the CGLP with (27), and for all $i \in \{1, \ldots, m\}$ define $\widetilde{u}_i = K_i u_i$ and $\widetilde{v}_i = K_i v_i$. From (4), we obtain

$$\gamma = uA - u_0\pi = \sum_{i=1}^{m} u_i a_i - u_0\pi = \sum_{i=1}^{m} (K_i u_i)(a_i/K_i) - u_0\pi = \widetilde{u}\widetilde{A} - u_0\pi.$$

Analogously, from (5) we get $\gamma = \widetilde{v}\widetilde{A} + v_0\pi$ and from (6)–(7) we have $\gamma_0 = \widetilde{u}\widetilde{b} - u_0\pi_0 = \widetilde{v}\widetilde{b} + v_0(\pi_0 + 1)$. Hence $(\gamma, \gamma_0, \widetilde{u}, \widetilde{v}, u_0, v_0)$ is a feasible solution of the CGLP associated with system $\widetilde{A}x \geq \widetilde{b}$ yielding the same cut as $(\gamma, \gamma_0, u, v, u_0, v_0)$. Since $\|\widetilde{a}_i\|\widetilde{u}_i + \|\widetilde{a}_i\|\widetilde{v}_i = \|a_i\|u_i + \|a_i\|v_i \; \forall\, i \in \{1, \ldots, m\}$, then $(\gamma, \gamma_0, \widetilde{u}, \widetilde{v}, u_0, v_0)$ fulfills normalization (27) as well. □

The above lemma shows that the CGLP with Euclidean normalization is not affected by scaling issues. Moreover, the CGLP with (27) is the same as the CGLP with the standard normalization for a system $\widetilde{A}x \geq \widetilde{b}$ where all the constraints have been scaled in order to have Euclidean norm equal to 1 (i.e., $\|\widetilde{a}_i\| = 1 \; \forall\, i \in \{1, \ldots, m\}$). By replacing normalization (11) with (27) we are losing the implicit penalization of high-rank inequalities hidden in the standard normalization. However, the Euclidean normalization associates penalties with the Farkas multipliers which are in some way related to the structure of the corresponding constraints instead of being all equal to 1 without any distinction.

Unfortunately, normalization (27) is not likely to work well in all cases. For example, consider a constraint $a_i x \geq b_i$ having both positive and negative coefficients, say $a_{ij} > 0$ and $a_{ik} < 0$, and assume the nonnegative variables x_j and x_k are nonbasic (cut coefficients γ_j and γ_k do not affect violation). The effect of the Farkas multipliers u_i and v_i on the strength of the cut coefficients γ_j and γ_k is not univocal: a large value of the Farkas multipliers u_i or v_i would lead to a weak (resp. strong) cut coefficient γ_j (resp. γ_k). Hence, associating a penalty $||a_i||$ with constraint a_i in the normalization might not be the right choice.

On the other hand, for constraints with nonnegative coefficients only, the Euclidean norm of a constraint is likely to give a reliable measure on how bad it is to use its associated Farkas multipliers u_i and v_i. In particular, the Euclidean Normalization seems to be particularly suited for Set Covering problems. Indeed, set covering constraints have nonnegative coefficients only, and this property is known to be inherited by nontrivial valid inequalities, including the disjunctive cuts we can separate through our procedure. This implies that weighing a constraint by means of its norm has a more direct impact on the cut density, and gives a sensible indication for nonbasic variables.

Finally, note that it is not necessary to solve the CGLP in the lifted space to use normalization EN. This has been recently shown by Balas and Bonami [4], who implemented our normalization on the original tableau by adapting the approach of Balas and Perregaard [8], thus proving that EN might be used within a fast implementation.

We performed a set of computational experiments on a test-bed of Set Covering instances taken from the OR–Library [10], and the results are reported in Table 6. The improvement in the percentage gap closed by EN w.r.t. SNC is quite substantial as it ranged from 2.38% to 4.19%, with an average of 3.12%.

Figures 10 and 11 describe the behavior of the two normalizations on the particular instance `scpnre5`.

Table 6 SNC normalization versus Euclidean normalization on SCP instances

Instance	SNC normalization			Euclidean normalization		
	# cuts	%gap	$\|S\|$	# cuts	%gap	$\|S\|$
scpnre1	904	13.67	89.22	951	17.35	93.62
scpnre2	963	9.38	95.42	997	12.51	98.14
scpnre3	923	15.14	91.41	944	18.13	92.82
scpnre4	878	13.25	85.99	897	15.70	87.82
scpnre5	889	16.84	87.77	935	21.03	91.16
scpnrf1	678	10.23	67.75	682	12.62	67.77
scpnrf2	655	9.62	65.42	689	12.90	68.50
scpnrf3	586	12.08	58.34	617	15.58	60.93
scpnrf4	664	10.21	66.35	692	12.59	68.91
scpnrf5	661	8.63	66.05	700	11.85	69.70
avg.	780.100	11.905	77.372	810.400	15.026	79.937

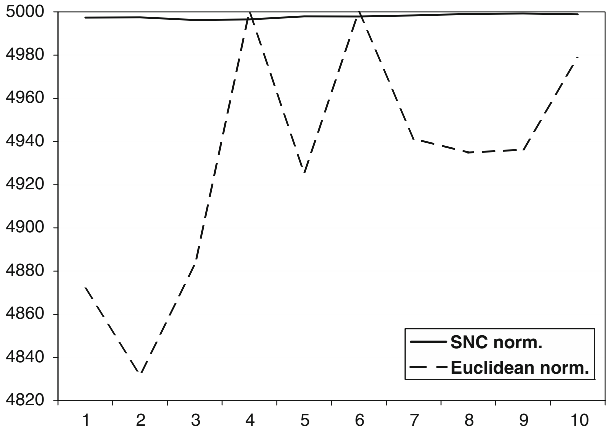

Fig. 10 SNC versus Euclidean normalization: average cut density for instance *scpnre*5

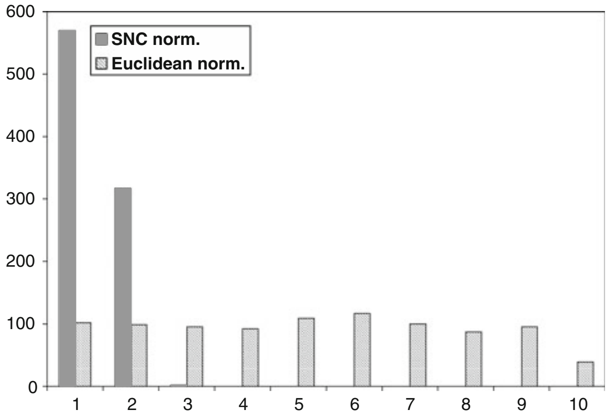

Fig. 11 SNC versus Euclidean normalization: cut rank for instance *scpnre*5

One can observe that the considerably higher rank of the cuts generated using the Euclidean normalization with respect to those obtained through SNC (see Fig. 11) does not correspond at all to denser cuts. Indeed, the former cuts are consistently sparser than the latter (Fig. 10). (Cuts generated using SNC are fully dense: the number of variables of the instance is 5,000 and the number of nonzero coefficients is almost always close to 5,000 too.)

Natural questions concern the effectiveness of the Euclidean Normalization (27) on the MIPLIB instances [11] used in the previous sections and, in addition, what happens when redundant constraints are removed. Due to the lack of space, we do not report the associated tables but the results show an overall (minor) improvement of the % gap closed with respect to SNC from 56.873 to 57.063%. Such an improvement becomes slightly larger once the redundant constraints are removed, namely from 58.793 to 60.014%. These results refer to the separation performed by projecting the CGLP onto the support (or the extended support in the case the redundant constraints are removed) of x^*.

Acknowledgments We thank the referees for their constructive and useful comments and remarks.

References

1. Balas, E.: A modified lift-and-project procedure. Math. Program. **79**, 19–31 (1997)
2. Balas, E.: Disjunctive programming. Ann. Discrete. Math. **5**, 3–51 (1979)
3. Balas, E.: Disjunctive programming: properties of the convex hull of feasible points. Discrete Appl. Math. **89**, 3–44 (1998)
4. Balas, E., Bonami, P.: Generating lift-and-project cuts from the LP simplex tableau: open source implementation and testing of new variants. Technical Report, Tepper School of Business, CMU (2008)
5. Balas, E., Ceria, S., Cornuéjols, G.: Mixed 0–1 programming by lift-and-project in a branch-and-cut framework. Manage. Sci. **42**, 1229–1246 (1996)
6. Balas, E., Jeroslow, R.: Strengthening cuts for mixed integer programs. Eur. J. Oper. Res. **4**, 224–234 (1980)
7. Balas, E., Perregaard, M.: Lift-and-project for mixed 0–1 programming: recent progress. Discrete Appl. Math. **123**, 129–154 (2002)
8. Balas, E., Perregaard, M.: A precise correspondence between lift-and-project cuts, simple disjunctive cuts, and mixed integer gomory cuts for 0–1 programming. Math. Program. Ser. B **94**, 221–245 (2003)
9. Balas, E., Saxena, A.: Optimizing over the split closure. Math. Program. Ser. A **113**, 219–240 (2008)
10. Beasley, J.E.: OR-Library: a collection of test data sets for a variety of Operations Research (OR) problems. http://people.brunel.ac.uk/~mastjjb/jeb/info.html
11. Bixby, R.E., Ceria, S., McZeal, C.M., Savelsbergh, M.W.P.: An updated mixed integer programming library: MIPLIB 3.0. http://www.caam.rice.edu/~bixby/miplib/miplib.html
12. Bonami, P.: Étude et mise en oeuvre d'approches polyédriques pour la résolution de programmes en nombres entiers ou mixtes généraux, PhD Thesis, Université de Paris 6 (2003)
13. Ceria, S., Soares, J.: Disjunctive cuts for mixed 0–1 programming: duality and lifting. GSB, Columbia University (1997)
14. Chvátal, V.: Edmonds polytopes and a hierarchy of combinatorial problems. Discrete Math. **4**, 305–337 (1973)
15. Cook, W.J., Kannan, R., Schrijver, A.: Chvátal closures for mixed integer programming problems. Math. Program. Ser. A **47**, 155–174 (1990)
16. Christof, T., Löbel, A.: PORTA—POlyhedron Representation Transformation Algorithm. http://www.zib.de/Optimization/Software/Porta/
17. Dash, S., Günlük, O., Lodi, A.: On the MIR closure of polyhedra. In: Fischetti, M., Williamson, D.P. (eds.) Integer Programming and Combinatorial Optimization—IPCO 2007. Lecture Notes in Computer Science, vol. 4513, pp. 337–351. Springer, Berlin (2007)
18. Fischetti, M., Lodi, A.: Optimizing over the first Chvátal closure. Math. Program. Ser. B **110**, 3–20 (2007)
19. Gomory, R.E.: An algorithm for the mixed integer problem, RM-2597, The RAND Corporation (1960)