Mathematical Programming manuscript No.
(will be inserted by the editor)

A. Frangioni · C. Gentile

Perspective Cuts for a Class of Convex 0-1 Mixed Integer Programs

the date of receipt and acceptance should be inserted later

Abstract. We show that the convex envelope of the objective function of Mixed-Integer Programming problems with a specific structure is the perspective function of the continuous part of the objective function. Using a characterization of the subdifferential of the perspective function, we derive "perspective cuts", a family of valid inequalities for the problem. Perspective cuts can be shown to belong to the general family of disjunctive cuts, but they do not require the solution of a potentially costly nonlinear programming problem to be separated. Using perspective cuts substantially improves the performance of Branch & Cut approaches for at least two models that, either "naturally" or after a proper reformulation, have the required structure: the Unit Commitment problem in electrical power production and the Mean-Variance problem in portfolio optimization.

Key words. Mixed-Integer Programs, Valid Inequalities, Unit Commitment problem, Portfolio Optimization

1. Introduction, motivation

In many real-world problems, both discrete and continuous decisions have to be made about the same entity. A common case is when one (or more) continuous variable(s) p is constrained to lie in the disconnected set $\{0\} \cup [p_{min}, p_{max}]$ for some $0 < p_{min} \leq p_{max}$. This is the case when p represents the output of a production process that can either be "inactive", and therefore nothing is produced, or "active", and therefore the output of the process must lie between some minimum and maximum amount. This structure is so widespread that commercial solver suites such as `XPRESS-MP` and `CPLEX` provide built-in special support for these *semi-continuous* variables. Alternatively, p_{min} may be zero but a *fixed cost* is incurred for producing any positive output; both these cases are covered by our development. Examples of this structure can be found in several models, such as Distribution and Production Planning problems [3,11,18], Financial Trading and Planning problems [1,10], and many others [5]. As shown later on, this structure can also be "forced", at least partially, upon problems that have semi-continuous variables but a nonseparable objective function.

A. Frangioni: Department of Computer Science, University of Pisa, Largo B. Pontecorvo 3, 56124 Pisa (ITALY), e-mail: `frangio@di.unipi.it`

C. Gentile: Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", C.N.R., Viale Manzoni 30, 00185 Rome (ITALY), e-mail: `gentile@iasi.cnr.it`

As a mathematical program, this structure corresponds to a Mixed-Integer Program (MIP) of the form

$$\min \{ \ f(p) + cu \ : \ Ap \leq bu \ , \ u \in \{0,1\} \ \} \tag{1}$$

where $p \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ are such that $\{p : Ap \leq 0\} = \{0\}$; the example motivating our development is obtained with $A = [-I, I]^T$ and $b = [-p_{min}, p_{max}]^T$, but in the following we will only assume that $\mathcal{P} = \{p \in \mathbb{R}^n : Ap \leq b\}$ is a compact set, and that $f : \mathbb{R}^n \to \mathbb{R}$ is a closed convex function that is finite on $\mathcal{P}$. The binary variable u models the decision of "activating the process", i.e., decides whether $p = 0$ or $p \in \mathcal{P}$, at the fixed cost c; we can assume that $f(0) = 0$, because any constant term in f can be embedded in the constant c. Usually, (1) is only a small fragment of a larger problem where other constraints are imposed on u and/or p. Often, many fragments of the form (1) are simultaneously present in a given model, with different data A, b, f and c, to represent different products and/or different productive processes and/or different time instants and/or different geographical locations; this is the case of both applications described in Section 3 and Section 4.

When a (MIP) comprising blocks of the form (1) has to be solved with a Branch & Cut (B&C) algorithm [13], the continuous relaxation of (1)

$$\min \{ \ f(p) + cu \ : \ Ap \leq bu \ , \ u \in [0,1] \ \} \tag{2}$$

is usually a part of the problem that is solved in order to derive lower bounds on the objective function value of the original (MIP). The lower bounds can be improved by adding cuts that provide a better description of the convex hull of the integer solutions; these cuts, however, depend on the structure of the other constraints linking different blocks (1) together, as the feasible region in (2) has only vertices with an integer value for u. Here we focus on a different way for improving the lower bound that depends only on the structure of a single block. Problem (1) can be equivalently restated as the minimization over all (p, u) of the nonconvex function

$$f(p,u) = \begin{cases} 0 & \text{if } u = 0 \text{ and } p = 0 \\ f(p) + c & \text{if } u = 1 \text{ and } Ap \leq b \\ +\infty & \text{otherwise} \end{cases} . \tag{3}$$

The best possible convex relaxation of this problem is obtained by minimizing over all (p, u) the *convex envelope* of f, i.e., the (closed) convex function $\overline{co}f$ with the smallest (in set–inclusion sense) epigraph containing that of f. In general, computing the convex envelope is a nontrivial task [15,17]; in this case it can be readily done, showing that $\overline{co}f(p, u)$ is related to the well-known *perspective function* of $f(p)$ [9] §IV.2.2. Using a result of [4] we can exploit this relation to derive valid inequalities for the problem, related to those of [2,16], which can be used to improve the lower bound w.r.t. the one obtained by (2).

The structure of the paper is the following: in Section 2 $\overline{co}f$ is characterized, some of its useful properties are discussed and the valid inequalities are described. In sections 3 and 4 the use of the valid inequalities within a Branch & Cut approach is discussed for two Mixed-Integer Quadratic Problems (MIQP) with the required structure. Finally, in Section 5 some conclusions are drawn.

We use the following standard notation. Given a set X, $I_X(x)=0$ if $x \in X$ (and $+\infty$ otherwise) is its *indicator function*, *int* X is its *interior*, *ext* X the set of its *extreme points*, *co* X is its *convex hull* and $\overline{co}$ X is the *closure* of *co* X. Given a convex function f, *epi* $f = \{(v,x) : v \geq f(x)\}$ is its *epigraph*, *dom* $f = \{x : f(x) < \infty\}$ is its *domain*, $\partial f(x)$ is its *subdifferential* at x, and $f'(x;d)$ is its directional derivative at x along direction d. We will often use the shorthand p/u for $(1/u)p$, where p is vector and u is a scalar.

2. Characterization of $\overline{co}f$

To characterize $\overline{co}f$ we just need to compute the convex hull of points in the epigraphical space pertaining to the two disconnected "sides" of *domf*, that is, the set of points

$$(1-\theta)[0,0,0] + \theta[f(\bar{p}) + c, \bar{p}, 1] = [\theta(f(\bar{p}) + c), \theta\bar{p}, \theta] \ .$$

for all $\theta \in [0,1]$ and $\bar{p} \in \mathcal{P}$. Since for $\theta = 0$ we obtain $[0,0,0]$, we can assume $\theta > 0$ and make the identifications $\theta \equiv u$, $p \equiv u\bar{p}$ to obtain $[uf(p/u) + uc, p, u]$, and therefore

$$h(p,u) = \overline{co}f(p,u) = \begin{cases} 0 & \text{if } p = 0 \text{ and } u = 0, \\ uf(p/u) + cu & \text{if } Ap \leq bu, \ \ u \in (0,1], \\ +\infty & \text{otherwise.} \end{cases} \tag{4}$$

Thus, h is strongly related with a well-known object in convex analysis, the

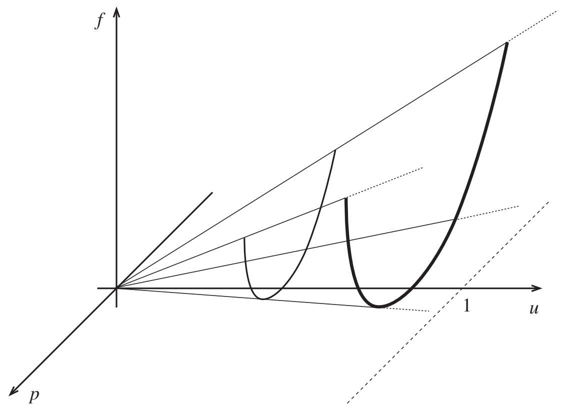

Fig. 1. The perspective function of $f(p)$

perspective function $g(p,u) = u\,f(p/u)$ of $f(p)$. The epigraph of $g(p,u)$ (for $u > 0$) defines a cone pointed in the origin and having as "lower shape" that of $f(p)$, as depicted in Figure 1; *epi* h is the section of the cone corresponding to $u \leq 1$. Hence, $\mathcal{F}$ = *dom* h = *co domf* is the pyramid having as base $\mathcal{P} \times \{1\}$

and vertex $[0, 0]$. Note that the explicit definition of $h(0, 0)$ in (4) is redundant, as the result is obtained by continuity: for every sequence $\{p_k, u_k\} \subset \mathcal{F}$ that converges to $[0, 0]$ we have

$$0 \leq u_k(f(p_k/u_k) + c) \leq u_k \left(c + \sup_{p \in \mathcal{P}} f(p) \right)$$

and therefore $\lim_{k \to \infty} u_k(f(p_k/u_k) + c) = 0$ since f is convex and finite on the compact set $\mathcal{P}$.

It is easy to verify that h is linear on the segments of the form $p = \bar{p}u$ with $u \in [0, 1]$ for any fixed $\bar{p} \in \mathcal{P}$: $h(\bar{p}u, u) = uf(\bar{p}) + cu$. This is confirmed by first-order analysis: it can be shown [4] that

$$\partial h(p, u) = \{ g(s) = [s, c + f(p/u) - s(p/u)] : s \in \partial f(p/u) \} , \tag{5}$$

i.e., ∂h depends only on p/u, and therefore it is constant on all points of the form $[\bar{p}u, u]$. This is immediate from ordinary first-order calculus when $f(p)$ is differentiable in $\mathcal{P}$, i.e., $\partial f(p) = \{\nabla f(p)\}$; then, h is differentiable in $int\, \mathcal{F}$. For the nondifferentiable case, linearity of h on the segments $p = \bar{p}u$ implies that h is directionally derivable in any point $[p, u]$ with $u > 0$ along the direction $[p, u]$:

$$h'([p, u]; [p, u]) = -h'([p, u]; -[p, u]) = [s_1, s_2][p, u] \qquad \forall [s_1, s_2] \in \partial h(p, u) .$$

Therefore, for any $[s_1, s_2] \in \partial h(p, u)$,

$$0 = h(0, 0) = h(p, u) + [s_1, s_2]([0, 0] - [p, u]) = h(p, u) - h'([p, u]; -[p, u])$$

yielding $[s_1, s_2][p, u] = h(p, u) = uf(p/u) + cu$ for all $[s_1, s_2] \in \partial h(p, u)$, whence $s_2 u = uf(p/u) + cu - s_1 p$; recalling that $u > 0$, all the subgradients of h in (p, u) must have the form $g(s_1)$, where s_1 has to be a subgradient of $h_u(p) = h(p, u)$. Using Theorem VI.4.2.1 in [9] it is not difficult to prove that $\partial h_u(p) = \partial f(p/u)$, which directly gives the $\subseteq$ of (5). Furthermore, every vector $g(s)$ in (5) satisfies the subgradient inequality

$$h(p, u) \geq h(\bar{p}, \bar{u}) + [s_1, s_2]([p, u] - [\bar{p}, \bar{u}]) \qquad \forall [s_1, s_2] \in \partial h(\bar{p}, \bar{u}) \tag{6}$$

which gives the reverse inclusion, proving (5). Thus, all points $[v, p, u] \in epi\ h$ must satisfy (6) for all $[\bar{p}, \bar{u}] \in \mathcal{F}$; since $\partial h(p, u)$ is constant on the lines $p = \bar{p}u$, it is sufficient to consider only points of the form $[\bar{p}, 1]$. All this proves:

Theorem 1. *The $n+2$-dimensional set epi h is bounded by the following linear inequalities:*

$$Ap \leq bu, \qquad u \leq 1 \ ; \tag{7}$$

$$v \geq f(\bar{p}) + c + [s, c + f(\bar{p}) - s\bar{p}]([p, u] - [\bar{p}, 1]) \qquad s \in \partial f(\bar{p}) \ , \ \bar{p} \in \mathcal{P} \ . \tag{8}$$

In particular, the inequalities (7) define maximal faces of epi h of dimension $n+1$, while the—possibly infinitely many—inequalities (8) define maximal faces of epi h of dimension at least one.

We refer to each inequality (8) as a *perspective cut* (P/C). Note that, since f is finite everywhere, $\partial f(p)$ is compact for all $p \in \mathcal{P}$; hence, from (8) we have that all the linear constraints in (6) for one $[\bar{p}, 1]$ corresponding to subgradients $s \notin ext\ \partial f(\bar{p})$ can be obtained as a convex combination of (at most $n+1$) constraints corresponding to subgradients in $ext\ \partial f(\bar{p})$. Hence, $\partial f(p)$ in (8) can be replaced with $ext\ \partial f(p)$; of course, the two coincide if f is differentiable in p, as $s = \nabla f(p)$ is the only possible choice in (8). For $n = 1$, $\partial f(p) = [f'_-(p), f'_+(p)]$, where f'_- and f'_+ are respectively the left and right derivative of f, and

$$\partial h(p, u) = co\{\ g(f'_-(p/u))\ ,\ g(f'_+(p/u))\ \}\ . \tag{9}$$

It is interesting to contrast h for two special cases of f, namely the linear case $f(p) = bp$ and the (convex) quadratic case $f(p) = ap^2 + bp$ (with $a > 0$, and $n = 1$ for simplicity). In the linear case we get

$$h(p, u) = uf(p/u) + cu + I_{\mathcal{F}} = bp + cu + I_{\mathcal{F}} = f(p) + cu + I_{\mathcal{F}}\ ;$$

that is, h coincides with the objective function of (2), which is therefore the best possible convex relaxation. In the quadratic case, instead, we get

$$h(p, u) = uf(p/u) + cu + I_{\mathcal{F}} = (1/u)ap^2 + bp + cu + I_{\mathcal{F}}\ ; \tag{10}$$

since $0 < u \leq 1$, $h(p, u) \geq ap^2 + bp + cu$, that is, h is a better objective function, for a continuous relaxation, than $f(p) + cu$. For $\mathcal{P} = [p_{min}, p_{max}]$, elementary calculus shows that the maximum of $h(p, u) - (ap^2 + bp + cu)$ over $\mathcal{F}$ is $ap^2_{max}/4$, attained at $[p_{max}/2, 1/2]$; that is, h "penalizes" precisely the "most nonintegral" points in $\mathcal{F}$. However, using $h(p, u)$ as the objective function has a serious drawback: it is a much "more nonlinear" function than $ap^2 + bp + cu$, and it is nondifferentiable at $[0, 0]$. The interior-point method of [4] could be used, but efficient implementations of that approach are not widely available, and have not yet been shown to be competitive with the sophisticated QP solvers available. Furthermore, interior-point methods are usually less well-suited than simplex-like methods in the context of enumerative approaches, since the latter reoptimize more efficiently.

Alternatively, Theorem 1 suggests using a *polyhedral approximation* of h as the objective function by iteratively collecting a finite subset of P/Cs; this is analogous to what is done in most NonDifferentiable Optimization algorithms [6,7]. Given a fractional solution $[v^*, p^*, u^*]$, corresponding to an approximation of h by a finite number of P/Cs, one can compute the value of $h(p^*, u^*)$ via (4) to check whether $v^* \approx h(p^*, u^*)$; if not, P/Cs corresponding to $[p^*/u^*, 1]$ are (strongly) violated by $[v^*, p^*, u^*]$, and therefore can be added to the current set of inequalities, improving the approximation of h and, possibly, the lower bound. This only requires the computation of one (preferably extreme) element of $\partial f(\bar{p})$, where $\bar{p} = p^*/u^*$.

This procedure is closely related to the *lift-and-project* (L&P) approach [2]. In L&P, violated valid inequalities are obtained by considering two disjoint subsets of the feasible region, such as those corresponding to fixing a fractional binary variable at 0 and at 1, respectively; a representation of the convex hull of these two sets can be obtained by "lifting" the problem in a larger space where the explicit convex multipliers are added, and a valid inequality can be obtained by solving a linear program on this space. Given a subgradient $s \in \partial f(\bar{p})$, one has the polyhedral approximation *epi* $f \subseteq \{\ (v,p) : v \geq f(\bar{p}) + s[p - \bar{p}]\ ,\ p \in \mathcal{P}\ \}$ for $u = 1$, while at the other side of the domain, i.e., for $u = 0$, one has $\{\ (v, 0) : v \geq 0\}$. Using results from [2], it is possible to show that the convex hull of the two regions is $\{\ (v,p) : v \geq sp + u(f(\bar{p}) + c - s\bar{p})\ ,\ p \in \mathcal{P}, u \in [0,1]\ \}$; this can be seen as an alternative definition of (8). Thus, the proposed procedure can be seen as a combination of an incremental linearization approach on $f(p)$ plus the application of L&P on the corresponding linearized problem. More generally, P/Cs are a special case of the inequalities produced by the generalized L&P approach of [16] for $0-1$ Nonlinear (MIP)s. However, due to the structure of (1) the P/C approach is much simpler than those of [2,16], in the following ways:

- the "convex combinator" u is a variable of the original formulation, rather than being added for algorithmic purposes; hence, separation of perspective cuts is very easy, with no need to set up and solve the separation problem—a large-scale problem of roughly twice the size of the original integer problem, with *nonlinear constraints* in the case of [16];
- we separate inequalities for each block (1) individually rather than only one inequality for the entire problem: therefore, our inequalities are global by nature and do not require lifting;
- our approach is simpler (even though not entirely straightforward) to implement using widely available and efficient optimization tools.

In the next sections we will show that P/Cs can significantly improve the performance of enumerative approaches for the solution of (MIP)s containing fragments of the form (1).

3. The Unit Commitment problem

The Unit Commitment (UC) problem in electrical power production is as follows. A set I of thermal generating units is given. Each unit $i \in I$ is characterized by a minimum and maximum power output, p^i_{min} and p^i_{max}, respectively, and by a convex quadratic power (fuel) cost function $f^i(p) = a^i p^2 + b^i p + c^i$. Over a set T of discretized time instants, covering some time horizon (e.g., hours or half-hours in a day or a week), an estimate d_t for $t \in T$ of the total power demand is available. The problem is that of generating, for each time period, enough power to meet the forecasted demand at minimal total cost. The operation of thermal units must satisfy a number of technical constraints, typically *minimum up- and down-time* ones: whenever unit i is turned on it must remain committed (actively generating power) for at least τ^i_u consecutive time instants, and, analogously, whenever unit i is turned off it must remain decommitted for at least τ^i_d consecutive time instants.

Introducing binary variables u_{it}, indicating the commitment of unit i at time instant t, and continuous variables p_{it} indicating the corresponding power output, a formulation of (UC) is

$$\min \left\{ \sum_{i \in I} \sum_{t \in T} a^i p_{it}^2 + b^i p_{it} + c^i u_{it} \;\middle|\; \begin{array}{ll} \sum_{i \in I} p_{it} = d_t & t \in T, \\ p^i_{min} u_{it} \le p_{it} \le p^i_{max} u_{it} & i \in I, t \in T \\ u \in U, \quad u_{it} \in \{0,1\} & i \in I, t \in T \end{array} \right\} \tag{11}$$

where U is the set of schedules respecting minimum up- and down-time constraints. This basic formulation can be extended to take into account other characteristics of the energy production environment, such as *spinning reserve* constraints, *network* constraints, *ramp rate* constraints, time-dependent *start-up costs*, other types of generating units (hydro units, nuclear units, ...); see e.g., [3] and the references therein. In the following, we will stick to formulation (11), which already contains $|I| \times |T|$ blocks of the form (1).

3.1. Implementation details

Implementing a B&C approach using P/Cs is not entirely straightforward, as some of the required operations are not supported by the API of available (MIQP) solvers such as `CPLEX 8.0`. In particular, changing the quadratic part of the objective function during the execution of the B&C is not allowed; furthermore, when an integer solution is found its objective function value for the "linearized" relaxation does not provide, in general, a valid upper bound, that can be obtained by evaluating the original quadratic objective function. All this prevented us from directly relying on the efficient and sophisticated B&C implementation in `CPLEX`, and forced us to implement a standard B&C algorithm from scratch, using `CPLEX` only to solve the relaxations at each node of the search tree.

The cut separation phase is quite standard: for any pair (i, t) such that u_{it} is fractional in the optimal solution (p^*, u^*) of the relaxation, the (unique) P/C associated with $(\bar{p}_{it} = p^*_{it}/u^*_{it}, 1)$ is

$$v_{it} \ge (2a^i \bar{p}_{it} + b^i) p_{it} + (c^i - a^i \bar{p}_{it}^2) u_{it}$$

this is added to the formulation if it is violated beyond a fixed threshold (10^{-4} in our experiments). However, some nonstandard operations are also required. The first time that a P/C is generated for a given pair (i, t), the corresponding fragment of the quadratic function must be removed from the formulation, and the extra variable v_{it} must be added to the model to represent the maximum between all the linear functions associated with the pair (i, t); in this case, we also add to the formulation the P/Cs associated with the two points $(p^i_{min}, 1)$ and $(p^i_{max}, 1)$.

We experimented with several different versions of the algorithm, varying the following choices:

- the separation procedure is applied only at the root node or throughout the B&C tree;
- the original fragment of the quadratic function is restored when the variable u_{it} of a "linearized" pair (i, t) is fixed to 1 in a branching, or the piecewise-linear approximation of h is kept;
- how many times the separation procedure is applied at each node of the B&C tree;
- the relaxations are solved with the dual simplex or with the barrier algorithm.

The experiments are not reported here for space and clarity reasons; the interested reader is referred to [8]. The experience showed that performing separation at each node but at most once for each node, avoiding "delinearization" (i.e., not returning to the original objective function) when fixing to 1 and using the dual simplex method is the best combination for the most difficult instances (although some exceptions exist). The main lesson to be learnt from the experiments is that reoptimization is crucial in a B&C framework: small losses in the quality of the bound, such as those incurred by only performing separation once and not delinearizing, may be worth paying as long as they allow the continuous solver to reoptimize more efficiently. The use of dual simplex is basically motivated by this choice, although the barrier solver is usually faster at the root node.

Since we were mainly interested on the effect of P/Cs on the lower bound computation, we did not include any heuristic in the B&C algorithm; instead, we provided the algorithm with an initial upper bound associated with a good feasible solution obtained by the Lagrangian heuristic in [3].

3.2. Computational results

We tested the B&C algorithm on a set of (UC) instances obtained as in [3]; the instances have 24 time periods and either 10 or 20 units (10 instances for each dimension). The stopping criterion for all variants was a relative gap lower than 0.1%. All variants were run on a PC with a 2.5 Ghz Pentium-4 processor and 1.5Gb RAM, running the Linux Debian 3.0 operating system (kernel 2.4.18). The codes were compiled with `gcc 3.0.4` using aggressive optimizations `-O3`. We also solved the same instances with the sophisticated general-purpose B&C algorithm of `CPLEX 8.0`, with a time limit of 10000 seconds, on the same machine.

The results are presented in Table 1: for both algorithms, "r.t", "r.g%", "time" and "nodes" stand respectively for the time and relative gap at the root node, the total time and the total number of nodes. For `CPLEX` we also report (column "gap%") the relative gap between the best integer solution and the best bound available when the algorithm is terminated.

Using P/Cs, even if within a "naive" B&C implementation, allows us to clearly outperform the sophisticated (MIQP) solver in `CPLEX 8.0`. This is easily justified by looking at the root node gaps and times: even one single pass of P/C separation routine reduces the gap by a factor of 5 at a relatively low computational cost; in most cases, this reduction is larger than the one obtained

	P/C				CPLEX				
instance	r.t	r.g%	time	nodes	r.t	r.g%	time	nodes	gap%
P10_24_a	0.63	0.46	4.39	7	0.20	1.52	15	809	-
P10_24_b	0.89	0.32	18.55	21	0.31	1.22	22	1024	-
P10_24_c	1.47	0.31	1.53	2	0.43	1.44	912	73349	-
P10_24_d	0.73	0.38	1.52	2	0.22	1.24	204	25075	-
P10_24_e	0.72	0.13	1.66	3	0.24	0.95	559	85934	-
P10_24_f	0.66	0.41	9.65	21	0.17	1.80	8	221	-
P10_24_g	0.84	0.77	2.30	2	0.22	5.17	5055	536048	-
P10_24_h	0.90	0.19	2.40	3	0.25	1.03	32	2189	-
P10_24_i	0.75	0.47	1.55	2	0.22	1.74	408	43617	-
P10_24_j	0.78	0.56	10.77	23	0.23	2.81	946	78269	-
average	0.84	0.40	5.42	9	0.25	1.89	816	84654	-
P20_24_a	4.17	0.28	15.61	3	1.41	2.36	10000	264179	1.27
P20_24_b	4.29	0.13	4.53	1	1.80	0.49	62	1205	-
P20_24_c	2.07	0.69	178.12	136	0.85	1.24	216	4083	-
P20_24_d	8.64	0.28	37.14	4	1.61	2.40	10000	331732	1.43
P20_24_e	8.42	0.20	23.75	2	1.71	1.63	10000	245582	0.87
P20_24_f	6.71	0.24	12.59	2	1.58	1.37	10000	268516	0.73
P20_24_g	4.83	0.28	12.71	3	0.87	2.23	10000	475400	1.45
P20_24_h	5.97	0.18	19.35	3	1.74	1.06	6137	189898	-
P20_24_i	6.73	0.23	44.35	44	1.55	2.60	10000	337915	1.69
P20_24_j	7.96	0.26	141.69	73	1.64	2.28	10000	286651	1.02
average	5.98	0.28	48.98	57	1.48	1.77	7642	240516	0.85

Table 1. Results for the Unit Commitment problem

by `CPLEX` in 10000 seconds. These results clearly show that using P/Cs can significantly improve the performance of a B&C approach for the solution of (UC) problems.

4. Markowitz Mean-Variance model

The Mean-Variance (MV) model in portfolio optimization [12] is as follows. A set of n risky assets are available; for each asset $i = 1, \ldots, n$, the expected unitary return μ_i for the considered time horizon is known. Also, the $n \times n$ variance-covariance matrix Q defined for the assets is available. Denoting by $p_i \in [0, 1]$ the fraction of the portfolio value invested in asset i, any vector p with $ep = 1$ (e being the vector of all ones) is a feasible allocation of the available resources over the assets, μp is the corresponding expected return and $p^T Qp$ is a measure of the associated risk (volatility). Thus, the problem faced by the "rational investor" is that of trading returns versus risk. There are several ways to do this, up to tracing all the *risk-return efficient frontier*. One simple approach is that of fixing a desired level of return ρ and minimizing the associated risk, i.e., solving

$$\min\{\, p^T Qp \;:\; ep = 1 \;,\; \mu p \geq \rho \;,\; p \geq 0 \,\} \;.$$

The above problem is a convex (QP), and therefore easy. However, in many real cases a number of further constraints over portfolio decisions exist. Typically, *minimum and maximum buy-in thresholds* p^i_{min} and p^i_{max} are set on each asset i, turning the problem into the much harder (MIQP)

$$\min \left\{ p^T Q p \;\middle|\; \begin{array}{l} ep = 1 \;\;,\;\; \mu p \geq \rho \;, \\ u_i p^i_{min} \leq p_i \leq u_i p^i_{max} \;,\; u_i \in \{0,1\} \;\; i = 1,\ldots,n \end{array} \right\} \;. \tag{12}$$

Further constraints can be easily imposed, such as maximum and minimum numbers of purchased assets, or fixed purchase costs can be considered. However, in the following we will stick with the basic formulation (12).

The (MV) problem has semi-continuous variables p_i, but its cost function is nonseparable. A diagonal objective function can be obtained if asset returns are estimated using *factor models* [10]. Even for a standard variance-covariance matrix Q, a reformulation of (12) with separable objective function can be obtained [10] by computing the Cholesky factorization $Q = LL^T$, introducing auxiliary variables $y = L^T p$ and replacing the objective function with $\sum_i y_i^2$. This is still not enough to enable direct application of our technique, since the p_i variables have zero costs. We therefore propose a—to the best of our knowledge, original—variant of this technique: select a positive diagonal $n \times n$ matrix D such that $Q - D$ is positive semidefinite, compute the Cholesky factorization $Q - D = \bar{L}\bar{L}^T$, introduce auxiliary variables $y = \bar{L}^T p$ and replace the objective function with $\sum_i y_i^2 + p^T D p$. The resulting model is amenable to application of P/Cs, although only "a fraction" of the overall objective function is "reflected" on the separable costs. Different ways exist for selecting a proper matrix D; we have found that computing the minimum eigenvalue λ_{min} of Q and setting $D = \lambda_{min} I$ already gives good results, although more sophisticated techniques may be used to "reflect more" of the original nonseparable cost matrix Q in its separable part D. Clearly, this reformulation technique can be used for any (MIQP) with semi-continuous variables and nonseparable cost matrix Q.

4.1. Implementation details

As discussed in Section 3.1, implementing a B&C approach with P/Cs using the callable libraries of `CPLEX 8.0` is not possible. Therefore, as for (UC) we have implemented a standard B&C algorithm for (12), using `CPLEX` to solve the continuous relaxations. However, as shown in the next section, (MV) instances are much more difficult to solve than (UC) instances of comparable size: instances with $n = 200$ cannot be solved within reasonable times if P/Cs are used (and even more so if they are not). We have therefore implemented a simple and interesting different approach. We first solve the continuous relaxation of the problem exactly as in the root node of the B&C approach, performing separation of the P/Cs and "linearizing" the corresponding variables. Then, we fix the resulting (MIQP) and solve it with `CPLEX`, with a time limit of 10000 seconds. Clearly, the obtained global lower bound (the optimal solution value if the B&C terminates) is valid for (12), while the optimal solution of the linearized formulation is not necessarily optimal for the original problem. Yet the solution is feasible, so by computing its true objective function value we obtain an estimate of the gap; as shown next, this gap is often pretty small, and always much smaller than that obtained by the B&C of `CPLEX` within the same time limit.

4.2. Computational results

To test the above approach, we have generated 10 (MV) instances with $n = 200$ and 10 instances with $n = 300$. The variance-covariance matrices Q have been generated using the well-known random generator of [14]. The desired level of return ρ has been randomly chosen in the interval $[0.002, 0.01]$, and the minimum and maximum buy-in thresholds p^i_{min} and p^i_{max} have been generated in the intervals $[0.075, 0.125]$ and $[0.375, 0.425]$, respectively. The experiments were performed in the environment described in Section 3.2.

	Heuristic with P/C					CPLEX		
instance	r.t	r.g%	time	nodes	gap%	r.g%	nodes	gap%
P200_a	1.52	16.43	964.54	27193	0.04	854.27	568093	126.96
P200_b	7.02	21.34	1111.32	9632	0.35	390.22	331854	18.70
P200_c	6.98	17.86	380.23	3776	0.01	392.68	142879	33.55
P200_d	5.76	11.60	1147.92	8025	0.01	425.71	279220	53.44
P200_e	1.13	11.43	1508.13	63789	0.28	893.62	1184880	94.08
P200_f	1.22	20.96	2788.23	58782	1.22	947.81	308271	163.60
P200_g	6.16	15.76	1483.17	19905	0.08	533.17	279081	101.76
P200_h	1.16	13.46	2023.67	43955	0.01	899.53	545995	118.64
P200_i	1.27	20.01	2222.45	27526	0.63	933.68	178916	87.60
P200_j	1.33	17.54	4165.96	208826	0.02	924.87	879640	133.42
average	3.35	16.64	1779.56	47141	0.27	719.56	469883	93.18
P300_a	4.14	9.90	1392.19	4175	0.20	1346.07	74658	276.16
P300_b	3.05	11.83	1913.31	3668	0.04	1357.27	62359	364.24
P300_c	8.74	11.88	3392.73	17173	0.02	1063.99	62130	246.63
P300_d	4.13	15.64	3510.94	21028	0.04	1362.35	59563	244.23
P300_e	4.16	13.69	10000.09	44980	0.60	1339.13	86591	286.03
P300_f	3.32	10.73	3784.57	17649	0.04	1316.18	175763	282.83
P300_g	17.49	18.07	9889.00	23590	0.17	796.61	41470	178.04
P300_h	3.28	17.23	10000.10	30469	5.48	1413.33	83525	326.15
P300_i	3.42	14.61	10000.18	44818	3.43	1270.11	95069	296.61
P300_j	3.22	17.31	10000.19	24571	5.40	1399.15	48369	324.42
average	5.50	14.09	6388.33	23212	1.54	1266.42	78940	282.53

Table 2. Results for the Mean-Variance problem

The results are shown in Table 2; the meaning of the data is the same as for the (UC) problem. For instances with $n = 200$, the heuristic always terminates in less than 10000 seconds (less than 2000 seconds on average), obtaining fairly small to extremely small gaps. For instances with $n = 300$, instead, the "linearized" problem is solved to optimality only in 6 cases out of 10; however, in each of these cases the solution obtained has a fairly small gap. In all the other cases the gap is somewhat larger, but still under 6%, and below 1% in one case. For both classes, `CPLEX` obtains, in 10000 seconds, a gap that is two orders of magnitude larger on average, and up to three orders of magnitude larger in several cases. This is clearly explained by comparing the root node gaps obtained by the two approaches: the standard formulation has huge gaps, indicating the extreme difficulty of these instances, but using P/Cs allow one in a few seconds to obtain a much better gap than that obtained by a standard B&C approach in 10000 seconds. Note that it might be possible to find “larger” matrices D, thereby probably improving the effectiveness of the corresponding P/Cs.

These results clearly show that the information provided by perspective cuts can be exploited to improve the performance of exact and approximate approaches to (MIQP)s that, either “naturally” or after a proper reformulation, have the structure (1).

5. Conclusion

We have shown that the convex envelope of a family of nonconvex functions that appear as a fragment of the objective function of many (MIP)s is closely related to the perspective function. This leads to the definition of perspective cuts, a family of valid inequalities for the original (MIP)s. These cuts turn out to be a special case of the disjunctive cuts of [2, 16], but their separation does not require the solution of a large-scale (non)linear program (with nonlinear constraints). With some effort, P/Cs can be integrated in a B&C approach with a partial linearization of the objective function, using widely available optimization tools such as `CPLEX`. We have also shown how to reformulate some (MIQP) with semi-continuous variables and nonseparable cost matrix in such a way that perspective cuts can still be used.

Despite the low dimensionality of the faces that they represent, P/Cs can be used, at least in two relevant cases, to significanlty improve the efficiency of—exact or approximate—enumerative approaches to the corresponding (MIQP). The results suggest that this technique may be valuable for a large class of optimization problems; should this be confirmed by further experiments, some better support for this type of approach could be expected in general tools in the future. Even at the current state of technology, however, the improvements in the lower bounds provided by P/Cs may largely outweigh the extra computational burden required for handling them.

One interesting issue that still has to be explored is whether other widespread structures in Mixed Integer Programs are amenable to analogous treatments, i.e., whether closed-form convexification formulae can be used for other interesting fragments as well, avoiding the general but potentially costly approach of [2, 16]; we plan to investigate this issue in the future. A different, interesting direction of research concerns whether optimization algorithms using higher-order information about the convex envelope, such as that of [4] or others, may be more efficient than the cutting-plane algorithm employed for our experiments.

Acknowledgements. The authors thank Giovanni Rinaldi and Giovanni Felici (IASI-CNR) for their helpful suggestions, and Andrea Lodi (University of Bologna) for an implemetation of the random generator described in [14]. We are also grateful to the referees for their helpful comments.

References

1. S. Ahn, L.F. Escudero, and M. Guignard-Spielberg. On modeling robust policies for financial trading. In T.A. Ciriani and R.L. Leachman, editors, *Optimization in Industry 2*, pages 163–184. Wiley, Chichester, 1994.
2. E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–324, 1993.
3. A. Borghetti, A. Frangioni, F. Lacalandra, and C.A. Nucci. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems*, 18(1):1–10, 2003.
4. S. Ceria and J. Soares. Convex programming for disjunctive convex optimization. *Mathematical Programming*, 86:595–614, 1999.
5. M.A. Duran and I.E Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.
6. A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13(1):117–156, 2002.
7. A. Frangioni. About Lagrangian Methods in Integer Optimization. *Annals of Operations Research*, to appear, 2005.
8. A. Frangioni and C. Gentile. Perspective Cuts for 0-1 Mixed Integer Programs. Technical Report 577, Istituto di Analisi dei Sistemi ed Informatica “Antonio Ruberti” (IASI-CNR), 2002.
9. J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms* I—*Fundamentals*. Grundlehren Math. Wiss. 305. Springer-Verlag, New York, 1993.
10. N.J. Jobst, M.D. Horniman, C.A. Lucas, and G. Mitra. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. In *Quantitative Finance*, volume 1, pages 1–13. Wiley, Chichester, 2001.
11. J. Kallrath and J.M. Wilson. *Business Optimization*. Macmillan Press Ltd., Houndmills, 1997.
12. H.M. Markowitz. Portfolio Selection. *Journal of Finance*, 7:77–91, 1952.
13. M.W. Padberg and G. Rinaldi. A Branch and Cut Algorithm for Resolution of Large Scale Symmetric Salesman Problems. *SIAM Review*, 33:60–100, 1991.
14. P.M. Pardalos and G.P. Rodgers. Computing aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 45:131–144, 1990.
15. A.D. Rikun. A convex envelope formula for multilinear functions. *Journal of Global Optimimization*, 10:425–437, 1997.
16. R.A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.
17. M. Tawarmalani and N.V. Sahinidis. Convex extensions and envelopes of lower semi-continuous functions. *Mathematical Programming*, 93:515–532, 2002.
18. J.M. Zamora and I.E. Grossmann. A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits. *Comput & Chem. Engin.*, 22:367–384, 1998.