Università di Pisa

Dipartimento di Informatica

Technical Report: TR-13-04

Approximated Perspective Relaxations: a Project&Lift Approach

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa
Largo B. Pontecorvo 3, 56127 Pisa – Italy
frangio@di.unipi.it

Fabio Furini
LIPN, Université Paris 13, 93430 Villetaneuse – France
fabio.furini@lipn.univ-paris13.fr

Claudio Gentile
Istituto di Analisi dei Sistemi ed Informatica, C.N.R.
Viale Manzoni 30, 00185 Rome – Italy
gentile@iasi.cnr.it

Approximated Perspective Relaxations: a Project&Lift Approach

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa
Largo B. Pontecorvo 3, 56127 Pisa – Italy
frangio@di.unipi.it

Fabio Furini
LIPN, Université Paris 13, 93430 Villetaneuse – France
fabio.furini@lipn.univ-paris13.fr

Claudio Gentile
Istituto di Analisi dei Sistemi ed Informatica, C.N.R.
Viale Manzoni 30, 00185 Rome – Italy
gentile@iasi.cnr.it

Abstract

The Perspective Reformulation (PR) of a Mixed-Integer NonLinear Program with semi-continuous variables is obtained by replacing each term in the (separable) objective function with its convex envelope. Solving the corresponding Perspective Relaxation requires appropriate techniques. Under some rather restrictive assumptions, the *Projected PR* can be defined where the integer variables are eliminated by projecting the solution set on the space of the continuous variables only. This approach produces a simple piecewise-convex problem with the same structure as the original one; however, this prevents the use of general-purpose solvers, in that some variables are then only implicitly represented in the formulation. We show how to construct an *Approximated* Projected PR whereby the projected formulation is “lifted” back to the original variable space, with the integer variables expressing one piece of the obtained piecewise-convex function; in some cases, this produces a reformulation of the original problem with exactly the same size and structure as the standard continuous relaxation but with a substantially improved bound. While the bound can be weaker than that of the PR, this approach can be applied in many more cases and allows direct use of off-the-shelf MIQP software; this is shown to be beneficial in different applications. In the process we also relax some of the other restrictive assumptions of the original development, such as the need for the objective function to be quadratic and the need for the left endpoint of the domain of the variables to be non-negative.

Keywords: *Mixed-Integer NonLinear Problems, Semicontinuous Variables, Perspective Reformulation, Projection*

1 Introduction

The interest in Mixed Integer Nonlinear Programming has steadily increased over the last decade, as researchers from both continuous optimization and discrete optimization find ways to collaborate. When only convex functions are involved, the extension of methods arising from the Mixed-Integer Linear case is easier and many works follow this line of research (e.g., see [1, 4, 17, 21] for surveys on applications and solution algorithms).

In this paper we consider convex separable Mixed-Integer NonLinear Programs (MINLP) with n semi-continuous variables $p_i \in \mathbb{R}^{m_i}$ for $i \in N = \{1, \dots, n\}$. That is, each p_i either assumes the value 0, or lies in some given convex polytope $\mathcal{P}_i = \{\, p_i \,:\, A_i p_i \leq b_i \,\}$; this implies that $\{\, p_i \,:\, A_i p_i \leq 0 \,\} = \{\, 0 \,\}$, and therefore allows the usual modeling trick where the semi-continuity of each p_i is expressed by using an associated binary variable u_i as in

$$\min \quad g(z) + \textstyle\sum_{i \in N} f_i(p_i) + c_i u_i \tag{1}$$

$$A_i p_i \leq b_i u_i \qquad\qquad i \in N \tag{2}$$

$$(p, u, z) \in \mathcal{O} \tag{3}$$

$$u \in \{0, 1\}^n \quad , \quad p \in \mathbb{R}^m \quad , \quad z \in \mathbb{R}^q \tag{4}$$

where all f_i and g are closed convex functions, z is the vector of all other variables, and $\mathcal{O}$ is any subset of $\mathbb{R}^{m+n+q}$ (with $m = \sum_{i \in N} m_i$), representing all the other constraints of the problem, beyond (2). Problem (1)—(4) can be used to model many real-world problems such as distribution and production planning problems [27, 7, 12], financial trading and planning problems [8, 6], and many others [2, 3, 16, 17, 15, 18]. As we shall see, in some applications (§4.3, §4.4) the binary variables u_i are not only useful to prescribe the semi-continuous status of the corresponding p_i, but also for representing some of the other constraints of the model; however, in some other cases (§4.1, §4.2) this does not happen, and hence the only source of non-convexity in (1)—(4) lies in the fact that one is actually dealing with the nonconvex functions

$$f_i(p_i, u_i) = \begin{cases} 0 & u_i = 0 \;,\; p_i = 0 \\ f_i(p_i) + c_i & u_i = 1 \;,\; p_i \in \mathcal{P}_i \\ +\infty & \text{otherwise} \end{cases} \quad .$$

One can therefore strive to devise tight convex underestimators of this function in order to guide exact or approximate solution approaches; this is the approach that has been most successfully followed in general-purpose approaches to MINLP (e.g. [4, 14, 24] among the many others). In this particular case it is actually possible to characterize its *convex envelope*, i.e., the best possible such underestimator. Indeed, the convex hull of the (possibly, disconnected) domain $\{0\} \cup \mathcal{P}_i$ of each p_i can be conveniently represented in a higher-dimensional space, which allows to derive *disjunctive cuts* for the problem [22]; this leads to the *Perspective Reformulation* of (1)—(4) [5, 7]

$$\text{(PR)} \qquad \min \left\{\, g(z) + \textstyle\sum_{i \in N} \tilde{f}_i(p_i, u_i) + c_i u_i \;:\; (2) \;,\; (3) \;,\; (4) \,\right\} \tag{5}$$

where $\tilde{f}_i(p_i, u_i) = u_i f_i(p_i / u_i)$ is the *perspective function* of $f_i(p_i)$. It is well-known that, since f_i is convex, $\tilde{f}_i$ is convex for $u_i \geq 0$; indeed, it coincides with the convex envelope of $f_i(p_i, u_i)$

on the set $\{ (p_i, u_i) \,:\, A_i p_i \leq b_i u_i \,,\, u_i \in (0,1] \}$, and it can be extended by continuity in $(0,0)$ assuming $0 f_i(0/0) = 0$. In other words, by dropping the integrality constraints in (5) one obtains the "best possible" convex relaxation, dubbed the *Perspective Relaxation* ($\overline{\text{PR}}$), which indeed turns out to be significantly stronger than the continuous relaxation of (1)—(4), and therefore is a more convenient starting point to develop exact and approximate solution algorithms [3, 7, 8, 12, 16]. Yet, all this hinges on the ability to solve ($\overline{\text{PR}}$) with comparable efficiency as the ordinary continuous relaxation despite the fact that $\tilde{f}_i$ can be "significantly more difficult" than the original f_i to deal with (for instance, it is nondifferentiable in $(0,0)$). For instance, one can reformulate (5) either as a Mixed-Integer Second-Order Cone Program [3, 9, 16, 26] (provided that the original objective function is SOCP-representable) or as a Semi-Infinite MINLP [7].

Recently, another approach has been proposed [10] for the case where the f_i are quadratic and $n = m$, leading to a reformulation of ($\overline{\text{PR}}$) in terms of a piecewise-convex optimization problem. By standard tricks, this is in turn equivalent to a QP of roughly the same size as the standard continuous relaxation, with at most $2m$ continuous variables replacing the m variables p, but with no u variables. When $\mathcal{O}$ has some valuable structure, this leads to the development of specialized solution approaches for ($\overline{\text{PR}}$) that can be significantly faster than those available for the continuous relaxations of the MI-SOCP or SI-MILP formulations, ultimately leading to better performances of the corresponding enumerative approaches. This Projected Perspective Reformulation (P^{2}R) approach is based on projecting out the u_i variables from the formulation by partial minimization, which can be carried out thanks to the particularly simple structure of the constraints in which the variables u_i are involved. However, it comes at the cost of three significantly restrictive assumptions on the data of the original problem (1)—(4):

A1) each p_i is a single variable ($m_i = 1$), thus $\mathcal{P}_i = [p^i_{min}, p^i_{max}]$, with $0 \leq p^i_{min} < p^i_{max}$;

A2) each u_i only appears in the corresponding constraint (2), but not in constraints (3);

A3) all functions are quadratic, i.e., $f_i(p_i) = a_i p_i^2 + b_i p_i$ (with $a_i > 0$).

While there are several relevant applications where A1, A2, and A3 hold, there are others where they do not. In particular, assumption A2 rules out all applications (§4.3, §4.4) in which the u_i variables are re-used to express other constraints, since in P^{2}R these have been entirely eliminated from the formulation. Further negative side-effects of this removal are that valid inequalities concerning the u_i variables cannot be added to the formulation, and that ad-hoc solution approaches must be developed, losing the possibility of exploiting off-the-shelf, general-purpose, state-of-the-art solvers that are both simpler to use and potentially more powerful given the huge amount of ingenuity and development/testing time that have been invested in them.

In this paper we improve on this state of affairs by developing an approach which works under less restrictive assumptions than A1, A2, and A3. In particular, in Section 2 we will extend the approach to the case where $p^i_{min} < 0$ and f_i respects one simple condition, satisfied by several classes of functions in addition to quadratic ones. However, the most relevant contribution is presented in Section 3, where we show how to eliminate A2 altogether, albeit at a cost. In fact, we construct an *Approximated* Projected PR (AP^{2}R), whereby the problem is reformulated over the variables p and z only, like in the P^{2}R approach, as if no constraints (3) were binding variables u; once this is done, a MINLP reformulation is constructed which re-introduces the integer variables u in a different way to entirely encode the obtained piecewise-convex function. While the corresponding continuous relaxation is in general weaker than that of the PR, the two being equivalent only when A2 holds, the new approach allows to extend the P^{2}R idea to the many non-projectable applications, and most importantly to use off-the-shelf MIQP software to solve it, thereby benefiting from all the sophisticated machinery it includes, rather than developing ad-hoc algorithms or advanced features such as callback functions. This is shown to be beneficial at least in some practical applications; in particular, the approach is tested on 1D sensor placement problems (§4.1), single-commodity fixed-charge Network Design problems (§4.2), Mean-Variance portfolio optimization problems with min-buy-in and portfolio cardinality constraints (§4.3), and Unit Commitment problems in electrical power production (§4.4).

It is also possible that the proposed technique, other than proving useful for MINLPs with semicontinuous variables, may lead to further advances for different structures. One striking consequence of our main result (cf. (28)) is that *by just translating the origin of a continuous variable in a MINLP one may improve the quality of the continuous relaxation bound*. This is quite an unusual concept that may perhaps find a broader application. Indeed, the computation of convex envelopes for specially-structured functions of "a few" variables is an important field for which several advances are being done; for instance, one of the most researched structures is that of functions $\phi(p,u) = f(p)g(u)$ where f is convex and g is concave [19, 23, 25]. The very recent [19] shows that the characterization of the convex envelope is possible in terms of piecewise definitions similar to our ones. However, the development in [19] requires p and u to live in a Cartesian product of intervals, while our development precisely rests on the assumption that "linking constraints" with a specific form exist between p and u. Yet, it is possible that techniques may be usefully exchanged between the two different settings.

2 Relaxing A1 and A3

We start by reproducing the analysis [10] under significantly weaker conditions than A1 and A3. Since in this paragraph we will only work with *one* block at a time, to simplify the notation we will drop the index "i". First, we relax A1 to

A$'$1) p is a single variable and $\mathcal{P}$ is a bounded real interval $[p_{min}, p_{max}]$,

where we do *not* require $p_{min} \geq 0$ (while, clearly, $p_{min} < p_{max}$), thereby allowing that $0 \in \operatorname{int} \mathcal{P}$. Note that, by changing sign to p ($p \to -p$) if necessary, we can always assume $p_{max} > 0$ without loss of generality. We will therefore concentrate on

$$\min \{\ f(p) + cu \ :\ p_{min}u \leq p \leq p_{max}u \ ,\ u \in \{0,1\}\ \} \tag{6}$$

and its Perspective Relaxation

$$\min \Big\{ f(p,u) \ =\ \tilde{f}(p,u) + cu \ :\ p_{min}u \leq p \leq p_{max}u \ ,\ u \in [0,1] \Big\} \quad . \tag{7}$$

The basic idea in [10] is to recast (7) as the minimization of the following function of p alone:

$$z(p) = \min_u f(p,u) = \min_u \Big\{ \tilde{f}(p,u) + cu \; : \; p_{min}u \leq p \leq p_{max}u \;,\; u \in [0,1] \Big\} \quad ; \tag{8}$$

by convexity, 0 has to belong to the domain of z even if $p_{min} > 0$ (that is, $p \in conv(\ \{0\} \cup [p_{min}, p_{max}]\ ))$; yet, allowing $p_{min} < 0$ complicates the analysis somewhat, as we will see.

When $f(p) = ap^2 + bp$, the (convex) function $z(p)$ can be algebraically characterized by studying the optimal solution $u^*(p)$ of the convex minimization problem in (8). In turn, $u^*(p)$ is easily obtained by the solution $\tilde{u}(p)$ (if any) of the first-order optimality conditions of the unconstrained version of the problem

$$\frac{\partial f}{\partial u}(p,u) = c + f(p/u) - f'(p/u)p/u = 0 \quad . \tag{9}$$

If $\tilde{u}(p)$ satisfying (9) exists and it is unique, it can be used to algebraically describe $u^*(p)$, and therefore

$$z(p) = \tilde{f}(p, u^*(p)) + cu^*(p) \quad .$$

In fact, if $\tilde{u}(p)p_{min} \leq p \leq \tilde{u}(p)p_{max}$ and $0 \leq \tilde{u}(p) \leq 1$ then clearly $u^*(p) = \tilde{u}(p)$; otherwise, $u^*(p)$ is the projection of $\tilde{u}(p)$ over the feasible region, i.e., the extreme of the interval nearer to $\tilde{u}(p)$. If instead (9) has no solution then the derivative always has the same sign and $u^*(p)$ can be similarly found by projection. This is indeed possible in the quadratic case, where

$$\frac{\partial f}{\partial u}(p,u) = c - \frac{ap^2}{u^2} = 0$$

has the solution (that, by convexity of $f(p,u)$, is a minimum)

$$\tilde{u}(p) = |p|\sqrt{a/c} = \begin{cases} p\sqrt{a/c} & \text{if } p \geq 0 \\ -p\sqrt{a/c} & \text{if } p \leq 0 \end{cases} \tag{10}$$

if and only if $c > 0$. We will now show that the treatment can be extended to any function f such as $\tilde{u}(p)$ has the following property:

A′3) f is such that either (9) has no solution, or for some $g^+ \geq 0$ and $g^- \geq 0$ dependent only on the data of the problem

$$\tilde{u}(p) = \begin{cases} pg^+ & \text{if } p \geq 0 \\ -pg^- & \text{if } p \leq 0 \end{cases} \quad . \tag{11}$$

is the unique stationary point of $f(p,u)$ with respect to u.

As $f(p,u)$ is only defined for $u \geq 0$, it follows that the solution $-pg^-$ is needed only if $p_{min} < 0$. Indeed, since the latter case requires a more complex treatment than the $p_{min} \geq 0$ case of [10] (for a quadratic f), we will deal with each separately.

2.1 The case $p_{min} \geq 0$

To simplify the presentation, in the following we will treat p_{min} as if it were a *positive* number, i.e., we will assume that p/p_{min} is always a well-defined quantity. If $p_{min} = 0$, the constraint $p_{min}u \leq p$ is redundant, and one can take $p/p_{min} = +\infty$; it can be easily verified that all the obtained formulae extend to this case.

Proposition 1 *If $A'3)$ is satisfied and $p_{min} \geq 0$ then $z(p)$ defined in (8) has the form*

$$z(p) = \begin{cases} z_1(p) = \big(\, f(p_{int})/p_{int} + c/p_{int}\,\big)p & 0 \leq p \leq p_{int} \\ z_2(p) = f(p) + c & p_{int} \leq p \leq p_{max}, \end{cases} \tag{12}$$

where $p_{int} \in \{p_{min}, 1/g^+, p_{max}\}$.

Proof. We start by rewriting the constraints in (8) as

$$(0 \leq)\ \frac{p}{p_{max}} \leq u \leq \min\left\{ \frac{p}{p_{min}}\ ,\ 1 \right\}. \tag{13}$$

Then we proceed by cases:

a. Equation (9) has no solution and the global minimum in (8) is attained at one of the two bounds for u in (13). So, there are two subcases:

a.1. The derivative is always negative, and therefore $u^*(p) = \min\{\ p/p_{min}\ ,\ 1\ \}$. This gives two sub-subcases:

a.1.1. $p/p_{min} \leq 1 \iff p \leq p_{min} \implies u^*(p) = p/p_{min} \implies$

$$z(p) = \big(\, f(p_{min})/p_{min} + c/p_{min}\,\big)p; \tag{14}$$

a.1.2. $p/p_{min} \geq 1 \iff p \geq p_{min} \implies u^*(p) = 1 \implies$

$$z(p) = f(p) + c\ \ . \tag{15}$$

In other words, $z(p)$ is the piecewise function

$$z(p) = \begin{cases} \big(\, f(p_{min})/p_{min} + c/p_{min}\,\big)p & \text{if } 0 \leq p \leq p_{min} \\ f(p) + c & \text{if } p_{min} \leq p \leq p_{max}. \end{cases} \tag{16}$$

a.2. The derivative is always positive, and therefore $u^*(p) = p/p_{max}$ (note that $0 \leq u^*(p) \leq 1$). This gives

$$z(p) = \big(\, f(p_{max})/p_{max} + c/p_{max}\,\big)p\ \ . \tag{17}$$

b. The only solution to (9) is (11), and three cases can arise:

b.1. $\tilde{u}(p) = p\ g^+ \leq p/p_{max} \iff p_{max} \leq 1/g^+ \implies u^*(p) = p/p_{max}$ and (17) holds.

b.2. $p/p_{max} \leq \tilde{u}(p) \leq p/p_{min} \iff p_{max} \geq 1/g^+ \geq p_{min}$; two further subcases arise:

b.2.1. $(p_{max} \geq)\ p \geq 1/g^+\ (\geq p_{min})$, which implies both $\tilde{u}(p) \geq 1$ and $p/p_{min} \geq 1$, so that $u^*(p) = 1$ and therefore (15) holds;

b.2.2. $p_{min} \leq p \leq 1/g^+\ (\leq p_{max})$, which gives $\tilde{u}(p) \leq 1$. Now, if $p_{min} \leq p$ then $p/p_{min} \geq 1$, and therefore $u^*(p) = \tilde{u}(p)$. However, because $p_{min} \leq 1/g^+$ we always have $p/p_{min} \geq pg^+ = \tilde{u}(p)$, thus even when $0 \leq p \leq p_{min}$ we have $u^*(p) = \tilde{u}(p)$, which finally implies

$$z(p) = \big(\ g^+ f(1/g^+) + cg^+\ \big)p\ \ . \tag{18}$$

Thus, $z(p)$ is the piecewise function

$$z(p) = \begin{cases} \big(\ g^+ f(1/g^+) + cg^+\ \big)p & \text{if } 0 \leq p \leq 1/g^+ \\ f(p) + c & \text{if } 1/g^+ \leq p \leq p_{max} \end{cases} \tag{19}$$

b.3. $\tilde{u}(p) \geq p/p_{min} \iff (p_{max} \geq)\ p_{min} \geq 1/g^+ \iff u^*(p) = \min\{p/p_{min}, 1\} \implies$ (16) holds.

■

Clearly,

$$z_1(p) = (\ z_2(p_{int})/p_{int}\ )p$$

which immediately shows that $z_1(p_{int}) = z_2(p_{int})$, and therefore allows us to write $z(p_{int})$ without further qualification. The analysis implies that $z_2(p) \geq z(p)$, since $z_2(p) = z(p)$ for $p \geq p_{int}$, and $z_2(p) \geq z_1(p) = z(p)$ for $p \leq p_{int}$. Furthermore, assuming $p_{min} \leq 1/g^+ \leq p_{max}$ one has that (9) computed at $p/\tilde{u}(p) = 1/g^+ = p_{int}$ gives (for a differentiable f)

$$(c + f(p_{int}))/p_{int} = f'(p_{int})$$

i.e., $z_1'(p_{int}) = z_2'(p_{int})$ as depicted in Figure 1. Thus, except in the two degenerate cases $p_{int} = p_{min} = 0$ and $p_{int} = p_{max}$, $z(p)$ is a two-piecewise function where the second piece coincides with the original objective function; moreover, if $p_{int} = 1/g^+$, the breakpoint is at the place where the first-order linearization of f targets the origin. Note that, in this case, the first piece of (19) is precisely this first-order linearization and $z(p)$ is also continuously differentiable.

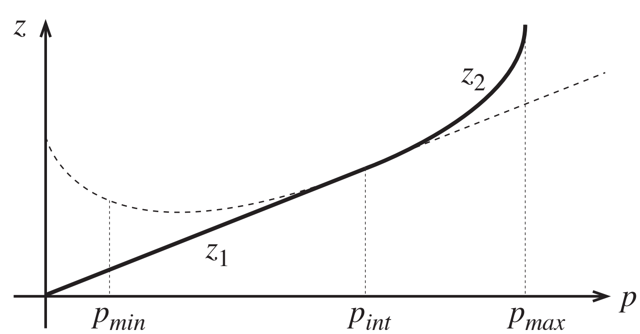

Figure 1: The piecewise function $z(p)$

2.2 The case $p_{min} < 0$

Proposition 2 *If $A'3)$ is satisfied and $p_{min} < 0$ then $z(p)$ defined in (19) has the form*

$$z(p) = \begin{cases} z_2(p) = f(p) + c & \textit{if } p_{min} \le p \le p_{int}^- \\ z_1^-(p) = \big(\ f(p_{int}^-)/p_{int}^- + c/p_{int}^-\ \big)p & \textit{if } p_{int}^- \le p \le 0 \\ z_1^+(p) = \big(\ f(p_{int}^+)/p_{int}^+ + c/p_{int}^+\ \big)p & \textit{if } 0 \le p \le p_{int}^+ \\ z_2(p) = f(p) + c & \textit{if } p_{int}^+ \le p \le p_{max} \end{cases} \tag{20}$$

where $p_{int}^- \in \{p_{min}, 1/g^-, 0\}$ and $p_{int}^+ \in \{0, 1/g^+, p_{max}\}$.

Proof. In this case, the form (13) of the constraints in (8) is no longer valid; indeed, $up_{min} \le p$ rather gives $u \ge p/p_{min}$, and therefore one obtains

$$\max\left\{ \frac{p}{p_{max}}\ ,\ \frac{p}{p_{min}} \right\} \le u \le 1\ \ . \tag{21}$$

Yet, the result of the leftmost “max” only depends on the sign of p; in particular

$$\begin{aligned} p \ge 0 &\implies \max\{\ p/p_{max}\ ,\ p/p_{min}\ \} = p/p_{max} \\ p \le 0 &\implies \max\{\ p/p_{max}\ ,\ p/p_{min}\ \} = p/p_{min}\ \ . \end{aligned}$$

Therefore, we can proceed by cases, mirroring the previous development with the necessary changes:

a. If (9) has no solution, the global minimum in (8) is one of the bounds in (21), and there are two subcases:

- **a.1.** The derivative is always negative, and therefore $u^*(p) = 1 \implies$ (15) holds (i.e., $p_{min} = p_{max} = 0$).
- **a.2.** The derivative is always positive, and therefore
  for $p < 0$, $u^*(p) = p/p_{min} \implies$ (14) holds,
  for $p \ge 0$, $u^*(p) = p/p_{max} \implies$ (17) holds.

All in all, in this case

$$z(p) = \begin{cases} \big(\ f(p_{min})/p_{min} + c/p_{min}\ \big)p & \text{if } p < 0 \\ \big(\ f(p_{max})/p_{max} + c/p_{max}\ \big)p & \text{if } p \ge 0 \end{cases}\ . \tag{22}$$

b. If, instead, the only solution to (9) is (11), one has to separately consider $[p_{min}, 0]$ and $[0, p_{max}]$, since $u^*(p) = \tilde{u}(p)$ if

$$\begin{aligned} p \in [p_{min}, 0] &\implies p/p_{min} \le \tilde{u}(p) = -pg^- \le 1 \\ p \in [0, p_{max}] &\implies p/p_{max} \le \tilde{u}(p) = \ \ pg^+ \le 1 \end{aligned}$$

That is, *exactly two* of the following *four* cases hold:

b.1. $p \geq 0$ and $\tilde{u}(p) \leq p/p_{max} \iff p_{max} \leq 1/g^+ \implies u^*(p) = p/p_{max} \implies$ (17) holds.

b.2. $p \geq 0$ and $\tilde{u}(p) \geq p/p_{max} \iff p_{max} \geq 1/g^+$; two further subcases arise:

b.2.1. $(p_{max} \geq)\ p \geq 1/g^+\ (\geq 0) \implies \tilde{u}(p) \geq 1 \implies u^*(p) = 1 \implies$ (15) holds.

b.2.2. $(0 \leq)\ p \leq 1/g^+\ (\leq p_{max}) \implies \tilde{u}(p) \leq 1 \implies u^*(p) = \tilde{u}(p) \implies$ (18) holds.

This again gives (19).

b.3. $p \leq 0$ and $\tilde{u}(p) \leq p/p_{min} \iff (0 >)\ p_{min} \geq -1/g^- \implies u^*(p) = p/p_{min} \implies$ (14).

b.4. $p \leq 0$ and $\tilde{u}(p) \geq p/p_{min} \iff p_{min} \leq -1/g^-\ (< 0)$; two further subcases arise:

b.4.1. $-1/g^- \leq p \leq 0 \iff \tilde{u}(p) \leq 1 \implies u^*(p) = \tilde{u}(p) \implies$

$$z(p) = \big(\, -g^- f(-1/g^-) - c g^- \,\big) p \tag{23}$$

b.4.2. $p_{min} \leq p \leq -1/g^-\ (< 0) \iff \tilde{u}(p) \geq 1 \implies u^*(p) = 1 \implies$ (15).

All this gives

$$z(p) = \begin{cases} f(p) + c & \text{if } p_{min} \leq p \leq -1/g^- \\ \big(\, -g^- f(-1/g^-) - c g^- \,\big) p & \text{if } -1/g^- \leq p \leq 0 \end{cases} \tag{24}$$

To summarize, $z(p)$ is the convex function with *at most* 4 pieces

$$z(p) = \begin{cases} f(p) + c & \text{if } p_{min} \leq p \leq -1/g^- \\ \big(\, -g^- f(-1/g^-) - c g^- \,\big) p & \text{if } -1/g^- \leq p \leq 0 \\ \big(\, g^+ f(1/g^+) + c g^+ \,\big) p & \text{if } 0 \leq p \leq 1/g^+ \\ f(p) + c & \text{if } 1/g^+ \leq p \leq p_{max} \end{cases} \tag{25}$$

Under condition b.1, the two rightmost pieces are substituted with the linear piece $(f(p_{max})/p_{max} + c/p_{max})p$ and/or, under condition b.3, the two leftmost pieces are substituted with the linear piece $(f(p_{min})/p_{min} + c/p_{min})p$, yielding a 3- or 2-piecewise convex function (piecewise-linear in the latter case as in (22)).

■

The above results generalize those of [10] for the quadratic case to handling $p_{min} < 0$, since we have $1/g^+ = 1/g^- = \sqrt{c/a}$. There, (18) simplifies to $2p\sqrt{ac}$, and (23) reads $-2p\sqrt{ac}$.

2.3 Some examples

We now show a few examples of non-quadratic functions where assumption A′3 holds.

2.3.1 The rational exponent case

Let us consider the function $f(p) = ap^{k/h}$, where $a > 0$ and $k > h$ integers. We will also ask $p_{min} \geq 0$ if k is odd to ensure that we use it only in the region where it is convex. In this case, (9) reduces to

$$c - a\left(\frac{k}{h} - 1\right)\left(\frac{p}{u}\right)^{\frac{k}{h}} = 0 \tag{26}$$

which, provided $c \neq 0$, has only one real root $\tilde{u}(p) = pg^+$ if k is odd and two roots $\tilde{u}(p) = \pm pg^+$ if k is even where

$$g^+ = \left(\frac{k-h}{h}\frac{a}{c}\right)^{\frac{h}{k}} .$$

Note that, if $c \leq 0$, the derivative is always negative (cf. a.1) for $p \geq 0$, while, if $c \geq 0$ and k is even, the derivative is always positive (cf. a.2) for $p \leq 0$; in both cases (26) has no solution. In all other cases, $\tilde{u}(p)$ has the form (11), with $g^- = g^+$ when k is even, and the analysis in points b. of propositions 1 and 2 apply depending on k odd or even, respectively.

Example 1 *If $k = 3$ (odd case), $h = 2$, $a = 1$, $c = 4$, and $0 \leq p_{min} \leq 4 \leq p_{max}$, one has*

$$g^+ = \left(\frac{1}{2}\frac{1}{4}\right)^{\frac{2}{3}} = \frac{1}{4} \qquad \text{and then} \qquad z(p) = \begin{cases} 3p & \text{if } 0 \leq p \leq 4 \\ p^{3/2} + 4 & \text{if } 4 \leq p \leq p_{max}, \end{cases} .$$

Example 2 *If $k = 4$ (even case), $h = 3$, $a = 3$, $c = 1$, $p_{min} \leq -1$, and $p_{max} \geq 1$, one has*

$$g^\pm = \left(\frac{1}{3}\frac{3}{1}\right)^{\frac{3}{4}} = 1 \qquad \text{and then} \qquad z(p) = \begin{cases} 3p^{4/3} + 1 & \text{if } p_{min} \leq p \leq -1 \\ -4p & \text{if } -1 \leq p \leq 0 \\ 4p & \text{if } 0 \leq p \leq 1 \\ 3p^{4/3} + 1 & \text{if } 1 \leq p \leq p_{max}, \end{cases} .$$

2.3.2 The exponential case

In the non-polynomial case $f(p) = e^{ap}$, (9) reduces to

$$c + e^{ap/u}(1 - ap/u) = 0 \; .$$

It is easy to verify that $g(z) = e^z(1-z) \leq 1$ (the maximum being attained in $z = 0$); this implies that for $c < -1$ the system cannot have a solution, the derivative is always negative (cf. a.1). For $c = -1$, the unique solution requires $ap/u = 0$, that is undefined in the variable u. Otherwise, the above equation defines one or two stationary points (depending on $c \geq 0$ or $-1 < c < 0$, respectively). In both cases, there is only one local minimum that is defined by

$$\tilde{u}(p) = \frac{ap}{1 + PL(c/e)}$$

where the $PL(z)$ (known as the "ProductLog" function) gives the principal solution for w in $z = we^w$, which is real for all $z \geq -1/e$; this can be efficiently computed numerically for a fixed argument such as c/e. Since in our case $z = c/e$, $\tilde{u}(p)$ is well-defined, e.g., whenever $c \geq 0$. If $a < 0$, then $e^{ap/u}(1 - ap/u) \geq 0$ and the derivative is always positive (cf. a.2). For $a > 0$ instead, $\tilde{u}(p)$ has the form (11) with $g^+ = a/(1 + PL(c/e)) > 0$; therefore, it is possible to apply the above analysis to this case, too.

Example 3 *For $c = e^2$ and $0 \leq p_{min} \leq 2 \leq p_{max}$ one has $w = 1$, $\tilde{u}(p) = p/2$, and $g^+ = 1/2$, hence*

$$z(p) = \begin{cases} e^2 p & \text{if } 0 \leq p \leq 2 \\ e^p + e^2 & \text{if } 2 \leq p \leq p_{max} \end{cases} .$$

2.3.3 The Kleinrock delay function case

Another interesting nonquadratic objective function is the *Kleinrock delay function* $f(p) = a/(p_{max} - p)$, which is often used to model delay in a communication network when the flow p over a given arc nears its maximum capacity p_{max} (e.g. [20]). The function is convex as long as $0 \le p_{min} \le p < p_{max}$ and $a > 0$; then, by applying the Perspective Relaxation (7)

$$f(p,u) \;=\; \begin{cases} uf(p/u) + cu = \dfrac{au^2}{up_{max} - p} + cu & \text{if } 0 \le p \in [up_{min}, up_{max}) \text{ and } u \in [0,1] \\ +\infty & \text{otherwise} \end{cases} .$$

For this case, (9) reduces to

$$c + \frac{au}{up_{max} - p} - \frac{aup}{(up_{max} - p)^2} = 0 \;\; ;$$

this (using $up_{max} - p > 0$) reduces to a simple quadratic form with non-negative quadratic coefficient $(cp_{max} + a)^2$. For $c > -a/p_{max}$, the form has the two roots

$$\tilde{u}_{\pm}(p) = \frac{p}{p_{max}}\left(1 \pm \sqrt{\frac{a}{cp_{max} + a}}\right) \;\; .$$

and therefore $\partial f/\partial u$ is nonpositive in the interval $\tilde{u}_{-}(p) \le u \le \tilde{u}_{+}(p)$ (even assuming it is defined there, which is not necessarily the case). In other words, $\tilde{u}_{+}$ is the unconstrained minimum, and (11) gives

$$g^{+} = \frac{1}{p_{max}}\left(1 + \sqrt{\frac{a}{cp_{max} + a}}\right) > 0$$

so that the above analysis can be applied. If $c \le -a/p_{max}$ instead, then $\partial f/\partial u$ is always positive, i.e., *co* f is always nonincreasing with respect to u, which gives $u^{*}(p) = 1$ and again the above analysis applies.

Example 4 *For $a = 4$, $c = 1$, and $p_{max} = 12$ one has*

$$z(p) = \begin{cases} \frac{p}{4} & \textit{if } 0 \le p \le 8 \\ \frac{4}{12-p} + 1 & \textit{if } 8 \le p \le 12 \end{cases} \;\; .$$

3 Project and Lift

As already mentioned in the introduction, one of the main limitations of the P^2R approach lies in the fact that the u_i variables are removed from the formulation; this makes it impossible to use off-the-shelf software to solve the corresponding problem. The main result of this section is that it is actually possible to "lift back" the obtained piecewise characterization of the convex envelope in the original space. The result is somewhat surprising, since (at least if $p_{min} \ge 0$) what one ends up with is a convex program with *exactly the same size and structure* as the original one, but which provides a (much) better bound. As a side effect of our development, it turns out that one can also apply the approach in the case where assumption A2 does not hold, i.e., the constraints defining $\mathcal{O}$ bind different variables u_i together, albeit at the cost of accepting a weaker lower bound than that provided by the Perspective Relaxation ($\overline{\mathrm{PR}}$). The idea is relatively simple: even if constraints (3) involve the u variables, one disregards them and proceeds to compute the projected function $z(p)$ as in the previous section. Of course, this provides a *lower bound* on what the computation of the "true" projected function would achieve, since one is disregarding some constraints, i.e., solving a relaxation of the real projection problem. As in the previous section, we analyze the somewhat simpler case where $p_{min} \geq 0$ first.

3.1 The case $p_{min} \geq 0$

The projected function $z(p)$ of Proposition 1 can always be *formulated* in terms of an appropriate nonlinear program. This property is proved by exploiting the following very well-known result (e.g., see [10]).

Lemma 5 *Let $\gamma(p)$ be a generic convex function with a k-piecewise description*

$$\gamma(p) = \gamma_i(p) \qquad \text{if } \alpha_{i-1} \leq p \leq \alpha_i \qquad i = 1, \ldots, k$$

(with each $\gamma_i(p)$ convex, obviously). Then $\gamma(p)$ can be rewritten as

$$\gamma(p) = \left\{ \begin{array}{ll} \min & \gamma_1(p_1 + \alpha_0) + \sum_{i=2}^{k} \big(\gamma_i(p_i + \alpha_{i-1}) - \gamma_i(\alpha_{i-1}) \big) \\ & 0 \leq p_i \leq \alpha_i - \alpha_{i-1} \qquad i = 1, \ldots, k \\ & \alpha_0 + \sum_{i=1}^{k} p_i = p \end{array} \right. . \tag{27}$$

Moreover, for any $p \in [\alpha_0, \alpha_k]$, there always exists an optimal solution $p^* = [p_1^*, \ldots, p_k^*]$ to problem (27) such that $p_i^* = \alpha_i - \alpha_{i-1}$ for $i < h$, $p_i^* = 0$ for $i > h$, and $p_h^* = p - \alpha_{h-1}$ for the index h such that $p \in [\alpha_{h-1}, \alpha_h]$.

Intuitively, Lemma 5 comes from the fact that a convex function has increasing slope, so the leftmost intervals are "more convenient" than the rightmost ones; thus, to obtain a given value p the best way is to "fill up the intervals starting from the left".

Theorem 6 *Let $z(p)$ be the function defined by (12) in Proposition 1, then $z(p)$ can be reformulated by the following program:*

$$z(p) = \left\{ \begin{array}{ll} \min\ h(u,q) = & uz(p_{int}) + z_2(q + p_{int}) - z(p_{int}) \\ & (p_{min} - p_{int})u \leq q \leq (p_{max} - p_{int})u \\ & p = p_{int}u + q \qquad , \qquad u \in [0,1] \end{array} \right. . \tag{28}$$

Proof. Applying (27) to (12) ($\alpha_0 = 0$, $\alpha_1 = p_{int}$, $\alpha_2 = p_{max}$, $k = 2$) gives

$$z(p) = \left\{ \begin{array}{ll} \min & z_1(p_1) + z_2(p_2 + p_{int}) - z(p_{int}) \\ & 0 \leq p_1 \leq p_{int} \quad , \quad 0 \leq p_2 \leq p_{max} - p_{int} \quad , \quad p = p_1 + p_2 \end{array} \right. \tag{29}$$

(remember that $z(p_{int}) = z_2(p_{int})$). Now, if we identify $p_1 = p_{int}u$ and $p_2 = q$ in (29), we easily get the equivalence of the objective functions of (29) and of (28) (because z_1 is linear) and of the constraints $p = p_1 + p_2$ and $p = p_{int}u + q$ (because $u \in [0, 1]$).

To justify the substitution of constraint

$$(p_{min} - p_{int})u \leq q \leq (p_{max} - p_{int})u \tag{30}$$

in place of $0 \leq p_2 = q \leq p_{max} - p_{int}$, the first step is to note that, due to Lemma 5, for any fixed p the optimal solution (p_1^*, p_2^*) of (29) satisfies

1. $p < p_{int} \iff p_1^* < \alpha_1 \iff u^* < 1 \Longrightarrow p_2^* = q^* = 0$;
2. $p \geq p_{int} \iff p_2^* = q^* \geq 0 \Longrightarrow p_1^* = \alpha_1\ (= p_{int}) \iff u^* = 1$.

This implies that (30) is satisfied by (q^*, u^*) whatever the value of p: indeed, by construction $p_{min} - p_{int} \leq 0 \leq p_{max} - p_{int}$, which implies on one hand that $q = 0$ is always feasible in the constraint whatever the value of u, and on the other hand that $p_{min} - p_{int} \leq 0 \leq q^*$. Yet, (30) is *weaker* than the original constraint, at least on one side, since it does *not* imply $q \geq 0$ unless $u = 0$. This opens the possibility that $q^* < 0$. If $u^* = 1$, then a solution (u, q) with $q < 0 \leq q^*$ cannot be optimal for (28). Indeed, the constraints $p = p_{int}u + q$ will imply that $u = (p - q)/p_{int} > u^* = 1$, violating the contraint $u \leq 1$. When $u^* < 1$, the optimal solution (u, q) of (28) could present $q < 0$ if $z_2(p) < z_1(p)$ for $p < p_{int}$; in this case, it could be more convenient to use "the cheaper z_2 on the left of its interval rather than the more costly z_1". Fortunately, this does not happen here, as Figure 1 usefully reminds. This is easy to prove formally by showing that in case 1. ($p < p_{int}$) the optimal solution to (28) is $(u^*, q^*) = (p/p_{int}, 0)$. In fact, consider any alternative feasible solution of the form $(u, q) = (u^* + \varepsilon/p_{int}, -\varepsilon)$ such that $p = p_{int}u + q$, where ε is arbitrary in sign (z_1 being linear is both concave and convex, a quite peculiar object in the usually one-sided word of convex analysis); the objective function value of this solution is

$$h(u, q) = p\frac{z(p_{int})}{p_{int}} + \varepsilon\frac{z(p_{int})}{p_{int}} + z_2(p_{int} - \varepsilon) - z(p_{int}) \quad .$$

Now, because z_1 is linear we have

$$z(p_{int})/p_{int} \in \partial z(\tilde{p}) \qquad \text{for each } \tilde{p} \in [0, p_{int}]$$

and taking in particular $\tilde{p} = p_{int}$ we can write the subgradient inequality

$$z(p_{int} - \varepsilon) \geq z(p_{int}) + (p_{int} - \varepsilon - p_{int})\frac{z(p_{int})}{p_{int}} = z(p_{int}) - \varepsilon\frac{z(p_{int})}{p_{int}} \quad .$$

Since $z_2 \geq z_1$ for $p \in [0, p_{int}]$ one has

$$z_2(p_{int} - \varepsilon) - z(p_{int}) + \varepsilon z(p_{int})/p_{int} \geq 0$$

and therefore $h(u^*, q^*) \leq h(u, q)$ as desired. This result only uses properties of the objective function, and does not depend at all on the constraint $q \geq 0$, which is therefore redundant, thus terminating the proof. ■

It is important to justify our choice of (28), and in particular the introduction of the somewhat troubling constraint $(p_{min} - p_{int})u \leq q$ in place of the natural (and stronger) $q \geq 0$. The point is that not only is (28) a *reformulation* of $(\overline{\text{PR}})$ which uses the same number of variables, but it has no fractional terms in the objective function; even more interestingly, adding the integrality constraint $u \in \{0, 1\}$ to (28) one clearly obtains a *reformulation of the original integer program* (6) *whose ordinary continuous relaxation is (equivalent to) the Perspective Relaxation if A2 holds.* This clearly requires that q can span the whole interval $[p_{min} - p_{int}, p_{max} - p_{int}]$ when $u = 1$, so that $p = p_{int} + q$ can span the whole interval $[p_{min}, p_{max}]$, which in turn requires that the constraint $q \geq 0$ must not be present. In this way we have obtained a simple algebraic reformulation to a problem with "the same degree of nonlinearity" as the original problem, whose continuous relaxation provides the same bound as the PR, when A2 holds, and a weaker bound—but still potentially better than that of the continuous relaxation of the standard formulation—if it does not. We call that the *Approximated Projected Perspective Reformulation* (AP^{2}R), where the "Approximated" tag is only justified when A2 does not hold. The interesting properties of AP^{2}R are:

- the integer variables u are present and play exactly the same role as in the original formulation, therefore (unlike in [10]) AP^{2}R can be passed to any general-purpose MINLP solver that can handle the original problem, exploiting all of its sophisticated machinery: branching rules, preprocessing, heuristics, any valid inequality for (1)—(4) concerning the u variables (cf. §4.2);

- the AP^{2}R has as many variables and constraints as the original formulation, and thus is more compact than any other readily solvable PR: even the MI-SOCP formulation [26, 3, 9, 16] has at least one more variable (per block), while the SI-MILP formulation [7] has one variable and infinitely many more constraints (cf. §4.3).

Therefore, AP^{2}R is a promising reformulation for (1)—(4). Of course, it also has some potential drawbacks:

- AP^{2}R may provide weaker bounds than $(\overline{\text{PR}})$ when Assumption A2 does not hold (cf. §4.3 and §4.4);

- AP^{2}R is unlikely to fully retain any useful combinatorial structure of the original problem like P^{2}R does, and therefore is likely to be less useful to develop specialized approaches for the bound computation (cf. §4.1);

- as already noted, in the extreme cases $p_{int} = p_{min} = 0$ and $p_{int} = p_{max}$ the $z(p)$ function is actually a single-piece one, and thus P^{2}R is much simpler than AP^{2}R: if $p_{int} = p_{max}$ for all variables (not an impossible event, cf. §4.1), for instance, P^{2}R has a linear objective function, whereas AP^{2}R keeps having the original nonlinear term (only, the constraints in (28) have the slightly simpler form $(p_{min} - p_{max})u \leq q \leq 0$);

- compared with the MI-SOCP formulation [26, 3, 9, 16], AP^{2}R has roughly the same size and degree of nonlinearity, so the relative performance of the two formulations should be expected to depend on fine details of the implementation, such as whether a solver is available which exploits the specific structure of f better than what interior-point methods can do for the MI-SOCP formulation (this is the case, e.g., when f is quadratic and the constraints linear, as one can use active-set quadratic solvers);

- compared to the SI-MILP formulation [7], AP^{2}R has roughly the same advantages as the MI-SOCP formulation (a compact and fixed formulation rather than the need for dynamically adding a potentially large number of constraints) as well as the same potential drawback: if the only nonlinearity in the model is that of f, the SI-MILP formulation solves sequences of Linear Programs, which can be faster than solving one nonlinear program especially when done iteratively during an enumerative approach thanks to the excellent reoptimization capabilities of LP codes (cf. §4.4).

Thus, the actual computational benefits of AP^{2}R over P^{2}R and the MI-SOCP and SI-MILP formulations can only be fully gauged experimentally.

3.2 The case $p_{min} < 0$

We can prove that (20), too, can be reformulated as a compact NLP.

Theorem 7 *Let $z(p)$ be the function defined by (20) in Proposition 2, then $z(p)$ can be reformulated by the following program:*

$$z(p) = \left\{ \begin{array}{ll} \min & h(u^+, u^-, q^+, q^-) \\ & -p^+_{int} u^+ \leq q^+ \leq (p_{max} - p^+_{int}) u^+ \\ & (p_{min} - p^-_{int}) u^- \leq q^- \leq -p^-_{int} u^- \\ & p = p^+_{int} u^+ + q^+ + p^-_{int} u^- + q^- \\ & u^+ + u^- \leq 1 \ \ , \ \ u^+ \in [0,1] \ \ , \ \ u^- \in [0,1] \end{array} \right. \tag{31}$$

where

$$h(u^+, u^-, q^+, q^-) = u^+ z_1^+(p^+_{int}) + z_2(q^+ + p^+_{int}) - z_1^+(p^+_{int}) + u^- z_1^-(p^-_{int}) + z_2(q^- + p^-_{int}) - z_1^-(p^-_{int}).$$

Proof. As in Theorem 6, the first step is to bring (20) in the form (27). Here $k = 4$, and using a slightly nonstandard numbering (to better highlight the fundamental symmetry of the function) we have $\alpha_{-2} = p_{min}$, $\alpha_{-1} = p^-_{int}$, $\alpha_0 = 0$, $\alpha_1 = p^+_{int}$, $\alpha_2 = p_{max}$, $z_{-2} = z_2$, $z_{-1} = z_1^-$, $z_1 = z_1^+$. Applying (27) to (20) gives

$$z(p) = \left\{ \begin{array}{lll} \min & z_2(p_{-2} + p_{min}) + z_1^-(p_{-1} + p^-_{int}) - z_1^-(p^-_{int}) + & \\ & \qquad z_1^+(p_1) + z_2(p_2 + p^+_{int}) - z(p^+_{int}) & \\ & 0 \leq p_{-2} \leq p^-_{int} - p_{min} & , \ \ 0 \leq p_{-1} \leq -p^-_{int} \\ & 0 \leq p_1 \leq p^+_{int} & , \ \ 0 \leq p_2 \leq p_{max} - p^+_{int} \\ & p = p_{min} + p_{-2} + p_{-1} + p_1 + p_2 & \end{array} \right. \tag{32}$$

(remember that $z_1^+(0) = 0$). We can now identify

$$p_{-2} + p_{min} = q^- + p^-_{int} \quad , \quad p_{-1} = p^-_{int}(u^- - 1) \quad , \quad p_1 = p^+_{int} u^+ \quad , \quad p_2 = q^+$$

to recover the objective function and most of the constraints in (31), as some simple but somewhat tedious algebra shows. Then, the general result about (27) can be applied to the optimal solution $(p^*_{-2}, p^*_{-1}, p^*_1, p^*_2)$ (or, equivalently, $(\hat{q}^-, \hat{u}^-, \hat{u}^+, \hat{q}^+)$) of (32) for any fixed p, yielding

p	p^*_{-2}	p^*_{-1}	p^*_1	p^*_2	$\hat{q}^-$	$\hat{u}^-$	$\hat{u}^+$	$\hat{q}^+$
$[p_{min}, p^-_{int}]$	≥ 0	0	0	0	≤ 0	1	0	0
$[p^-_{int}, 0]$	$p^-_{int} - p_{min}$	≥ 0	0	0	0	$\in [0,1]$	0	0
$[0, p^+_{int}]$	$p^-_{int} - p_{min}$	$-p^-_{int}$	≥ 0	0	0	0	$\in [0,1]$	0
$[p^+_{int}, p_{max}]$	$p^-_{int} - p_{min}$	$-p^-_{int}$	p^+_{int}	≥ 0	0	0	1	≥ 0

This shows that the constraints

$$-p^+_{int}u^+ \leq q^+ \leq (p_{max} - p^+_{int})u^+ \quad , \quad (p_{min} - p^-_{int})u^- \leq q^- \leq -p^-_{int}u^- \quad , \quad u^- + u^+ \leq 1$$

are satisfied by $(\hat{q}^-, \hat{u}^-, \hat{u}^+, \hat{q}^+)$ for each value of p. Again, the issue is that $-p^+_{int}u^+ \leq q^+$ is weaker than $0 \leq q^+$ and $q^- \leq -p^-_{int}u^-$ is weaker than $q^- \leq 0$ ($p^-_{int} \leq 0 \leq p^+_{int}$). However, reasoning as in Theorem 6 one easily shows that relaxing the constraints in this way does not change the optimal solution to (32). ■

Once again, the choice of (31) is motivated by the fact that, imposing integrality constraints $u^+ \in \{0,1\}$, $u^- \in \{0,1\}$ and with the identification $u = u^+ + u^-$, one obtains a *reformulation* of the original MINLP whose continuous relaxation is equivalent to ($\overline{\text{PR}}$) under Assumption A2, and weaker otherwise. This formulation has twice the number of continuous and binary variables than the ordinary formulation (counting the semicontinuous variables only), but possibly provides (much) stronger bounds.

4 Computational results

In this section we report results of computational tests performed on four classes of (MIQP)s with semicontinuous variables. For all the problems, it has already been clearly shown [7, 8, 9, 10] that Perspective Reformulations are largely preferable to the ordinary formulation; therefore, we will not report results for the latter, focussing only on the comparison between different forms of PRs. Among these, the SI-MILP formulation has been shown to be consistently more effective than the MI-SOCP one [9], and therefore we will refrain from testing the latter, too. Hence, we will compare three possible approaches: the SI-MILP formulation, denoted as "PC", the Projected Perspective Relaxation of [10], denoted as "P^{2}R", when Assumption A2 holds and a specialized solver is available, and the newly proposed approach, denoted as "AP^{2}R".

The experiments have been performed on a PC with a 2Ghz Opteron 246 processor and 2Gb RAM, running a 64 bits Linux operating system. All the codes were compiled with `gcc 4.4.3` and `-O3` optimizations, using `Cplex 12.3`. PC and AP^{2}R entirely rely on the (sophisticated) B&C machinery of `Cplex`, which is used here with all standard parameters setting; in particular, the stopping condition of the B&C is an optimality gap below 0.01%. By contrast, P^{2}R requires a "hand-made" B&B, in one case using `Cplex` to compute the lower bounds, and in another being entirely independent from it; of course, the stopping criterion has been set to the same 0.01%. The B&B used for P^2R is not a particularly sophisticated one (see [10] for details), and it surely could be improved. On the other hand, general-purpose solvers like `Cplex` keep improving all the time, usually at a much faster rate than the developers of any specialized solver can afford, and require almost no programming (except for setting appropriate `cutcallback` functions for PC). Besides, they have several sophisticated options that can be activated or improved by appropriate parameter tuning, which we purposely refrained from doing. Thus, while the results could possibly be improved somewhat for all the tested approaches, we believe this way of testing to be appropriate in that it shows the relative performance experienced by a non-expert user.

4.1 Sensor Placement problem

The (one-dimensional) Sensor Placement (SP) problem requires placing a set $N = \{1, \ldots, n\}$ of sensors to cover a given area while minimizing the fixed deployment cost plus an energy cost, that is quadratic in the radius of the surface covered. A simple MIQP formulation, which exhibits structure (3), is

$$\min \left\{ \sum_{i \in N} c_i y_i + \sum_{i \in N} a_i x_i^2 \ : \ \sum_{i \in N} x_i = 1 \ , \ 0 \leq x_i \leq y_i \ , \ y_i \in \{0, 1\} \quad i \in N \right\}$$

The P^2R relaxation boils down to a continuous convex quadratic knapsack problem with at most $2n$ variables that can be solved in $O(n \log n)$ [13].

We tested 168 random instances of the Sensor Placement problem, grouped in 8 classes. The first 4 classes, with 30 instances each, contain random instances with either 2000 or 3000 sensors and either "high" ("h") or "low" ("l") quadratic costs. The following two classes, with 24 instances each, derive from random instances of the Partition problem, according to the $\mathcal{NP}$-hardness proof for (SP) [2]. All these have already been used in [10], to which the interested reader is referred for further details. Because these instances were all solved at the root node by both P^2R and AP^2R (cf. Table 1), we also developed and tested some additional more difficult instances. These have been obtained by replicating the Partition and Subset Sum instances that can be found at

`http://people.sc.fsu.edu/~jburkardt/datasets/partition_problem/partition_problem.html`

and then applying the reduction procedure from Partition problem to the (SP) problem as in [2] (it is well-known that Subset Sum can be reduced to Partition, and therefore to (SP)). We constructed 9 instances with $n = 50$ and 9 instances of $n = 100$ sensors; of each group, 3 (denoted by $p*$) are derived from Partition and the rest (denoted by $s*$) from Subset Sum. All the instances can be freely downloaded from

`http://www.di.unipi.it/optimize/Data/RDR.html` .

The results are displayed in Table 1. For each approach we report the total running time (in seconds) and the B&B nodes required to solve the problem to optimality, averaged among the instances of each group.

Table 1 shows that, as already reported in [10], projected formulations are by far the most effective way to solve this (simple, yet $\mathcal{NP}$-hard) problem. While PC is more effective

	PC		P^2R		AP^2R	
	nodes	time	nodes	time	nodes	time
2000-h	3	24.88	0	0.28	0	0.40
2000-l	0	12.43	0	0.07	0	0.13
3000-h	1	58.00	0	0.65	0	0.64
3000-l	0	29.37	0	0.15	0	0.20
PTN-2000	6	79.16	0	0.31	0	1.86
PTN-3000	5	176.72	0	0.72	0	4.45
p50	26	0.16	583	0.22	23	0.06
s50	285	0.37	747	0.25	150	0.15
p100	33	0.46	10897	14.04	70	0.15
s100	1438	3.01	23396	27.88	1112	2.22

Table 1: Results of the (SP) problem

than the MI-SOCP formulation, and much more so than using the standard continuous relaxation [10], it is considerably outperformed by P^2R among all instances that require a few or no branching nodes. This is due both to the much faster solution of ($\overline{PR}$), and to the fact that very accurately solving ($\overline{PR}$) pays off surprisingly well in this case: while the exact solution of ($\overline{PR}$) produces a feasible solution which immediately closes the gap on the random generated instances, the approximate solution of PC can require a certain amount of branching to achieve the same effect.

Among projected methods, P^2R is faster than AP^2R when no branching is required; by a slim margin on the random instances, by a more significant one (up to a factor of six) on the PTN-* ones. This is not surprising, as (SP) is clearly a worst case as far as AP^2R versus P^2R goes: neither of the two methods require any branching on those instances, so the faster $O(n \log n)$ specialized solution algorithm of P^2R [13] clearly pays off in this case, while the more efficient branching and cutting techniques uniquely available to AP^2R have no impact. We remark that this is one case where P^2R solves a strictly smaller continuous program than AP^2R because the two-piece function is actually a single-piece one; indeed, around 2% of the variables in "h" instances and *all* the variables in "l" instances have $p_{int} = p_{max}$, and therefore only the linear piece is defined for P^2R (this also explains why "l" instances are solved much faster than "h" ones).

However, when branching is required ($p*$ and $s*$ instances) AP^2R is competitive with P^2R. In fact, all the sophisticated machinery (preprocessing, branching, heuristic, and cutting techniques) that is available in `Cplex` allows a significant reduction of the number of nodes w.r.t. the hand-coded enumerative approach required by P^2R; despite each node taking longer to solve, the final balance favors the new approach. Indeed, for larger instances even PC is competitive with P^2R for the same reason; yet, AP^2R is even better.

4.2 Nonlinear Network Design problem

The quadratic, separable, single-commodity Network Design (ND) problem requires routing a generic flow on a directed graph $G = (N, A)$, where each node $i \in N$ has a deficit $d_i \in \mathbb{R}$ indicating the amount of flow that the node demands. Each arc $(i, j) \in A$ can be used up to a given maximum capacity u_{ij}, paying a fixed cost c_{ij} if a nonzero amount of flow x_{ij} transits along the arc; flow cost is a convex quadratic function $b_{ij}x_{ij} + a_{ij}x_{ij}^2$. The MIQP formulation

$$\begin{array}{lll} \min & \sum_{(i,j)\in A}(c_{ij}y_{ij} + b_{ij}x_{ij} + a_{ij}x_{ij}^2) & \\ & \sum_{(j,i)\in A} x_{ji} - \sum_{(i,j)\in A} x_{ij} = d_i & i \in N \\ & l_{ij}y_{ij} \leq x_{ij} \leq u_{ij}y_{ij} \ \ , \ \ y_{ij} \in \{0,1\} & (i,j) \in A \end{array} \tag{33}$$

again exhibits structure (3) together with a strong network structure, so that the computation of the P^{2}R can be reduced to a convex quadratic min-cost flow problem on a graph with (at most) twice the number of arcs. For this problem, we tested 180 of the 360 instances used in [10]. These are randomly generated with the well-known `netgen` generator, different sizes (from 1000 to 3000 nodes) and fixed and quadratic costs generated as to be "high" ("h") or "low" ("l") w.r.t. the original linear costs of `netgen`; more details can be found in [10], and the instances can be freely downloaded from

`http://www.di.unipi.it/optimize/Data/MCF.html` .

For the current tests we discarded half of the original instances, those with "high" quadratic costs. The rationale for this choice is that all these instances are solved at the root node by all the methods, similarly to what happens with the "easy" (SP) instances. The results for all these instances are therefore easily inferred from those of the other half, many (but not all) of which are also solved at the root node, as discussed below.

The results are reported in Table 2. For each approach we report the number of B&B nodes and the required running time, averaged among groups of 30 instances with similar characteristics. However, since there are substantial differences among each group, with several instances solved at the root node while others require significant branching, for AP^{2}R we also report the average (among the 30 instances) of the ratio between the running time and that of the other two approaches.

	PC		P^{2}R		AP^{2}R			
	nodes	time	nodes	time	nodes	time	PC	P^{2}R
1000-l	4	34.11	4	0.30	5	0.43	0.06	11.30
1000-h	4	27.14	3	0.23	3	0.36	0.37	10.21
2000-l	259	103.23	415	71.18	219	5.09	0.03	11.49
2000-h	76	66.98	83	11.29	51	2.14	0.48	9.91
3000-l	309	145.95	280	76.13	273	9.85	0.50	9.06
3000-h	79	92.81	49	12.01	59	3.85	0.77	14.39

Table 2: Results of the (ND) problem

The results may at first appear contradictory: while the average running time of AP^{2}R is most often better than that of P^{2}R, the average ratio is always smaller than one, often significantly so. Similarly (although somewhat less markedly) for PC: the ratio between the average running times does not seem to correctly reflect the average ratio. The reason is that the relative performance of the algorithms is not uniform across different running times. To better gauge this phenomenon, the ratios (y axis) are plotted against the AP^{2}R running time (x axis) in Figure 2 for PC (left) and P^{2}R (right); note that both axes are in logarithmic scale.

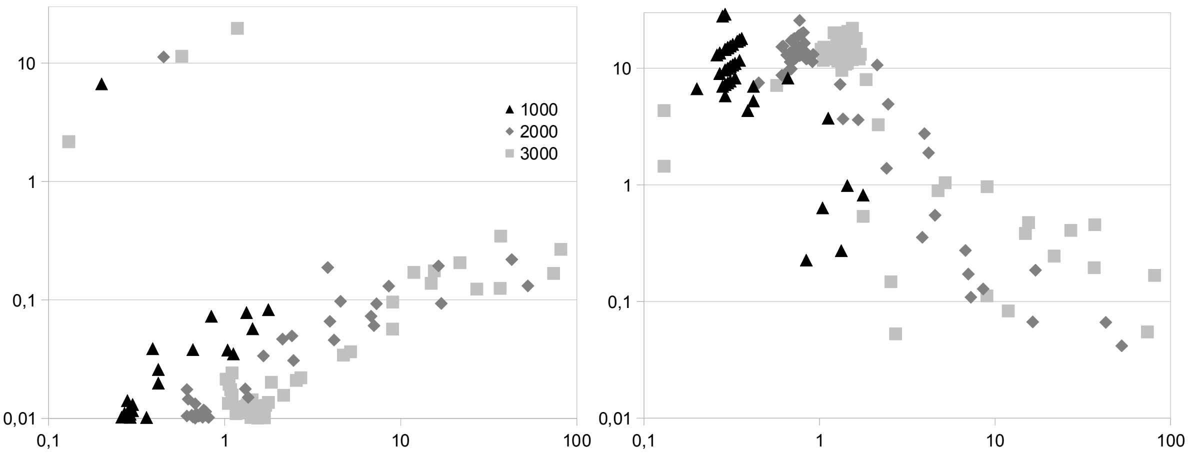

Figure 2: AP^{2}R time ratios for (ND): versus PC (left) and versus P^{2}R (right).

The figure shows that AP^{2}R is almost always faster than PC; the handful of instances where this is not true happen to be all and only the empty ones, that are proven so at the root node by both approaches. Here PC is faster because it can do so without generating any cut, and therefore solving a simpler linear program w.r.t. the quadratic program of AP^{2}R. In all the other instances that are solved at root node (the majority), AP^{2}R computes the PR bound faster, and therefore attains a very small ratio; however, these are all solved in a few seconds. For the instances that require significant branching, and therefore significantly longer running times, AP^{2}R is still markedly faster, but the ratio tends to be somewhat larger.

The opposite trend is apparent in the P^{2}R chart: due to the faster ($\overline{\mathrm{PR}}$) solution, the ratio is large for the (many) instances that are solved at root node. However, whenever significant branching occurs the benefits of the more sophisticated techniques available to AP^{2}R result in a significantly improved running time, with ratios that tend to decrease as the running time increases, being smaller than one for all instances requiring more than 10 seconds (as well as for some requiring less). Note that the other 180 instances of the test set of [10], being all solved at the root node, would have only contributed to make (much) thicker the “cloud” on the left of both charts without altering their fundamental shape.

4.3 Mean-Variance portfolio problem

The Mean-Variance (MV) portfolio problem with minimum and maximum buy-in thresholds requires optimally allocating wealth among a set A of assets in order to obtain a prescribed level of return ρ while minimizing the risk as measured by the variance of the portfolio. A *nonseparable* (MIQP) formulation is

$$\min \left\{ p^T Q p \ : \ \textstyle\sum_{i \in A} p_i = 1 \ , \ \sum_{i \in A} \mu_i p_i \geq \rho \ , \ p^i_{min} u_i \leq p_i \leq p^i_{max} u_i \ , \ u_i \in \{0,1\} \quad i \in A \right\}$$

where μ_i, p^i_{min} and p^i_{max} are respectively the expected unitary return and the minimum and maximum buy-in thresholds for asset i, while Q is the variance-covariance matrix. This apparently simple model is rather demanding for general-purpose (MIQP) solvers, since the root node gaps of the ordinary continuous relaxation are huge, and its very simple structure means that classical polyhedral approaches to improve the lower bounds are scarcely effective. To apply PR techniques to this problem, first the objective function need be modified to extract the "largest" possible diagonal part with efficient and effective (although the improvement largely depends on the characteristics of Q) SemiDefinite Programming techniques; see [8] for details. For our tests we used the 90 randomly-generated instances, 30 for each value of $n \in \{200, 300, 400\}$, already employed in [7, 8] and available at

`http://www.di.unipi.it/optimize/Data/MV.html` ;

the interested reader is referred to the cited sources for details. Here we only mention that each group of 30 instances is subdivided into three sub-groups denoted "+", "0" and "−" according to the fact that Q is strongly diagonally dominant, diagonally dominant, or not diagonally dominant, respectively; this turns out to have a substantial effect on the quality of the diagonal objective function that can be extracted and therefore on the effectiveness of the PR, making instances more and more difficult (for a fixed size) as they become less and less diagonally dominant [8].

An important remark is that, once the objective function is made separable, the problem actually satisfies Assumption A2. It was not considered in [10] because it lacks exploitable structure to develop specialized solution algorithms for the Projected PR, and therefore does not seem to be a promising candidate for the P^{2}R approach; clearly, this makes it an ideal candidate for AP^{2}R. Yet, in order to test the effect of non-separability on the quality of the bounds, and therefore the effectiveness of AP^{2}R, we experimented with adding to (MV) the simple cardinality constraint

$$\sum_{i=1}^{n} u_i \leq k \tag{34}$$

for some $k \leq n$. This provides a useful "gauge": while Assumption A2 is satisfied for $k = n$, decreasing k "increases the amount of non-separability" in the model, possibly impacting on the tightness of the AP^{2}R bound. We therefore tested all of the 90 instances twice: once with $k = n$, and once with $k = 10$. The latter is a quite strict requirement, considering that the min-buy-in constraints (which actually are the only source of difficulty in this problem) would allow about 20 assets to be picked. As we shall see, (34) significantly impacts on the performances of AP^{2}R.

The results are reported in Table 3. Each row of the table reports average results between 10 instances with the same characteristics; the upper part of the Table is relative to $k = n$, while the lower part is relative to $k = 10$ in (34). The leftmost part of the Table describes the efficiency and effectiveness of the PR at the root node. That is, for each approach, we report the time necessary to compute the root lower bound, before the introduction of any valid inequality (except those necessary for PC, of course) as well as the gap, in percentage, with the optimal solution (columns "time" and "dgap" under "lb"). We do not report times and gaps for the final root lower bound since, as already stated, polyhedral techniques have no discernible effect except (uselessly) consuming a little running time. Instead we report the time necessary to also obtain the first heuristic solution at the root node with the gap, in percentage, between the cost of that solution and that of the optimal one (columns "time" and "pgap" under "heur"). A specific twist is that AP^{2}R sometimes (albeit rarely) fails to find a primal solution at the root node; thus, in that case "pgap" excludes the instances for which no solution was found, whose number is reported next on the right (0 meaning that a solution was always found). The rightmost part of the table is devoted instead to the overall B&C performances. We do not report averaged number of nodes/running time since these vary wildly between different instances even for the same approach, even more so than in §4.2 (up to three orders of magnitude). Therefore, column "nodes" reports the ratio between the number of B&C nodes explored by AP^{2}R and those explored by PC, while columns "time" do the same for total running times, with maximum and minimum values (among the 10 instances of the same group) reported together with the average. To help better appraising the results, in Figure 3 we also plot, for each of the 90 instances, the value of the time ratio as a function of the running time of AP^{2}R; the chart on the left is for $k = n$, that on the right for $k = 10$. As in Figure 2, both axes are in logarithmic scale.

	AP^{2}R					PC				AP^{2}R / PC			
	lb		heur			lb		heur		nodes	time		
	time	dgap	time	pgap		time	dgap	time	pgap	avg	avg	min	max
200^+	0.56	1.14	0.62	7.51	1	0.80	1.14	2.45	7.52	0.81	0.35	0.13	0.71
200^0	0.48	2.14	0.56	10.69	0	0.64	2.14	1.64	10.69	1.02	0.61	0.36	0.84
200^-	0.49	3.65	0.59	18.81	1	0.66	3.65	1.78	15.58	0.97	0.63	0.38	0.98
300^+	1.51	1.30	1.64	2.85	1	2.18	1.30	5.31	11.82	0.51	0.29	0.17	0.45
300^0	1.46	1.99	1.68	18.59	0	2.12	1.99	5.17	17.48	0.71	0.45	0.16	0.74
300^-	1.50	2.68	1.73	18.83	1	1.92	2.68	5.22	14.52	0.92	0.51	0.36	0.74
400^+	3.91	1.43	4.17	2.68	0	5.71	1.43	10.69	16.18	0.52	0.33	0.11	1.14
400^0	3.32	2.30	3.66	14.18	1	5.18	2.30	10.98	56.54	0.73	0.41	0.27	0.61
400^-	3.29	3.06	3.62	7.87	0	5.06	3.06	10.98	32.82	0.94	0.46	0.31	0.70
200^+	0.49	0.77	0.55	8.23	0	1.38	0.51	2.95	1.65	2.67	0.41	0.21	0.69
200^0	0.46	3.14	0.55	39.73	0	1.34	2.75	3.01	17.88	3.26	0.70	0.33	1.41
200^-	0.49	4.75	0.60	76.76	0	1.45	4.18	3.51	30.53	4.56	0.87	0.23	2.17
300^+	1.43	1.08	1.57	16.43	0	3.96	0.49	7.51	2.64	6.02	0.78	0.37	1.37
300^0	1.40	2.91	1.59	30.77	0	4.14	2.35	8.69	15.61	4.72	0.84	0.32	1.66
300^-	1.47	3.92	1.77	83.38	0	4.03	3.58	7.91	41.35	3.50	0.63	0.40	1.10
400^+	3.43	0.85	3.61	5.51	0	9.99	0.41	14.09	1.35	7.94	0.87	0.30	1.54
400^0	3.09	3.00	3.55	50.41	1	9.29	2.34	15.09	26.70	6.55	1.50	0.32	4.49
400^-	3.15	4.53	3.77	81.93	1	9.43	3.80	15.19	47.78	6.87	0.94	0.37	1.99

Table 3: Results of the (MV) problem

The results clearly show that without (34) AP^{2}R outperforms PC. The reason is clearly explained by the root node results: gaps are (as the theory dictates) identical, running times are far shorter, especially when the heuristic is taken into account. Indeed, AP^{2}R is up to 5 times faster to get a feasible solution. The quality of the solutions varies, as it can be expected with heuristics, but there is no clear dominance of either approach; on average AP^{2}R seems to fare somewhat better, except for the occasional failures that PC does not suffer. The advantage of AP^{2}R over PC is mainly its faster ($\overline{\text{PR}}$) computation due to the compact model: this is confirmed when looking at the B&C results. In fact, while most

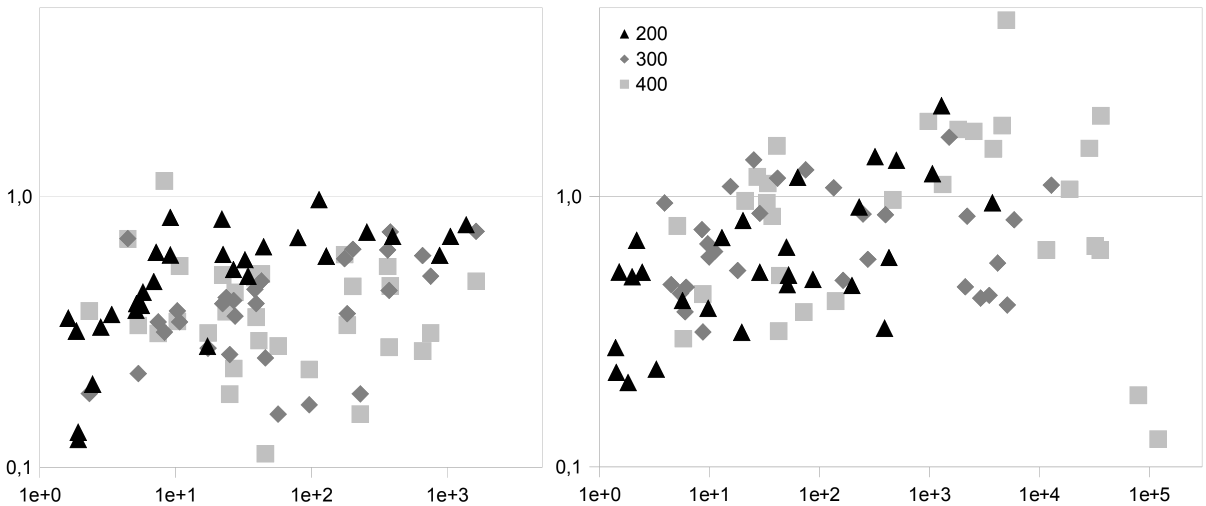

Figure 3: AP^{2}R time ratios for (MV): $k = n$ (left) and $k = 10$ (right).

often AP^{2}R is faster than PC by a factor of 2 or more in average, getting close to one order of magnitude faster in the best case and almost never being slower in the worst case, the number of nodes is much more comparable. Yet AP^{2}R requires on average less nodes, likely indicating that the more compact model provides better guidance for branching and/or heuristics. Figure 3 (left) shows that, with only one exception, AP^{2}R is always faster than PC; furthermore, the ratio seems, in general, to be somewhat better for larger and more difficult instances.

However, the results are markedly different for $k = 10$. While the root node running times of AP^{2}R are still significantly lower, the quality of the lower bound is worse, sometimes significantly so. This seems to directly reflect on the quality of the heuristic solutions: with only one exception, those computed by AP^{2}R are (on average) of worse quality, sometimes significantly so. All this should be expected to have, and indeed it has, adverse effects on the B&C. The number of nodes increase very significantly, being now on average between 3 and 7 times that of PC. Due to the faster relaxation solution, the effects on running times is much less dramatic: on all but one classes of instances AP^{2}R is still, on average, faster than PC. However, the average and minimum ratios are now much closer to 1, and the maximum ratio is almost always greater than 1, indicating that AP^{2}R is slower in at least one instance for each class, being up to 4.5 slower in one case. Figure 3 (right) shows that while AP^{2}R is competitive (the time ratio is smaller than one) in the majority of instances, there are several ones in which PC is faster, sometimes very consistently so. Also, the Figure seems to suggest that the ratio tends to get worse as the difficulty of the instance increases. This is consistent with the folklore that the more difficult an instance is, the more it pays off to invest on computing tight lower bounds, even if at the cost of increased relaxation times; this favors PC over AP^{2}R. Hence, these results show that, as it could be expected, there is a trade-off between faster computing times and quality of the bounds, when Assumption A2 does not hold, which may not always favor AP^{2}R over competing PR techniques.

4.4 Unit Commitment problem

The Unit Commitment (UC) problem in electrical power production requires optimally operating a set of t thermal and h hydro generating units to satisfy a given total power demand on the hours of a day. Each thermal unit is characterized by a minimum and maximum energy output $0 < p^{min} < p^{max}$, when the unit is operational, by a convex quadratic energy (fuel) cost function $f(p) = ap^2 + bp$ of the produced power p, and by a fixed cost c to be paid for each hour that the unit is operational; therefore, it exhibits structure (3) with $n = 24t$, where u is the binary variable indicating whether or not the unit is operational. The complete formulation is rather complex and we refrain from discussing it in detail; the interested reader is referred, e.g., to [11, 12]. For the purpose of the present discussion, however, it is important to mention that thermal units are subject to several complex constraints such as *minimum up- and down-time* and *ramp rate* ones, linking energy and commitment variables for the same unit at different hours, as well as (possibly) *spinning reserve* constraints linking energy and commitment variables for different units at any given hour. In other words, Assumption A2 is *strongly violated* in (UC), with several crucial constraints linking the u variables of different blocks together.

We have compared PC and AP^{2}R on a test bed of randomly generated realistic instances already employed in [7, 8, 11, 12], and freely available at

`http://www.di.unipi.it/optimize/Data/UC.html` .

In practical applications these problems need to be solved quickly, and therefore are solved with low required accuracy [11, 12]. Here we solved them with the default 0.01% accuracy as in the other cases; hence like in [8] we only report results for the instances of small size (up to $t = 75$, $h = 35$) and with a(n already unrealistic) time limit of 10000 seconds. The results are displayed in Table 4: rows with $h = 0$ refer to "pure thermal" instances, and each row reports averaged results of 5 instances of the same size. The uppermost part of the table reports results for AP^{2}R, the lowermost for PC. The leftmost part of the table concerns root node results. In particular, we report the gap between the obtained lower bound and the optimal solution (column "dgap") for both the "pure" PR (columns "NCNH", i.e., "No Cuts, No Heuristic") and after that valid inequalities have been added (columns "CNH", i.e., "Cuts enabled, but No Heuristic"), together with the required time. Also, we report the gap between the root node heuristic solution value and the optimal solution (column "pgap"), together with the required time. The rightmost part of the table is devoted to the results of the B&C: we report the number of nodes and the running time at termination, the gap between the best solution found and the optimal solution ("pgap" again), and the gap between the upper and lower bound at termination (column "gap").

Table 4 shows that AP^{2}R is not competitive with PC on (UC). As the theory predicts, the AP^{2}R lower bound is (very slightly, but visibly) worse than that of PC, albeit the difference is much less than in (MV) for $k = 10$ (whose gap however can be much larger, cf. Table 3). Yet, AP^{2}R is somewhat *less* efficient in solving $(\overline{\text{PR}})$. Differently from (MV), valid inequalities here help in considerably reducing the bound, more often than not increasing the (small but visible) gap between the AP^{2}R lower bound and the PC one; furthermore, PC obtains this (better) bound two/three times faster. Results for the heuristic vary, with AP^{2}R often being capable of obtaining more and better solutions (except in hydro-thermal instances,

		AP^{2}R									
		NCNH		CNH		CH		B&C			
t	h	time	dgap	time	dgap	time	pgap	nodes	time	pgap	gap
10	0	0.31	1.49	1.50	0.29	2.59	0.04	1229	284	0.00	0.01
20	0	0.75	1.25	6.97	0.29	13.99	0.15	4635	9999	0.01	0.17
50	0	3.19	1.19	50.53	0.22	65.12	0.23	1078	9999	0.02	0.23
20	10	0.88	0.58	3.04	0.15	7.91	0.16	16477	1078	0.00	0.01
50	20	2.91	0.58	18.97	0.09	28.33		3780	9999	0.00	0.07
75	35	5.46	0.49	39.18	0.06	45.73		1727	9999	0.03	0.08
		PC									
10	0	0.17	1.48	0.99	0.23	1.25	0.40	365	17	0.00	0.01
20	0	0.49	1.24	3.93	0.25	5.38		15607	4851	0.00	0.02
50	0	2.85	1.16	16.59	0.19	20.63		14286	9986	0.00	0.13
20	10	0.52	0.56	1.92	0.13	3.14	0.51	8107	240	0.00	0.01
50	20	2.05	0.57	6.17	0.07	13.11		66945	6649	0.00	0.02
75	35	4.19	0.48	11.23	0.05	20.22	0.08	57456	9999	0.00	0.02

Table 4: Results of the (UC) problem

that have a larger continuous part); however, the penalty in running time worsens. All this clearly impacts on the behavior of the B&C: AP^{2}R requires more nodes to solve one instance and each node requires more time, so PC can ultimately solve more instances and obtain better upper and lower bounds for those that cannot be solved within the time limit. This is likely due to the fact that (UC) instances are known to have a quite flat objective function (small quadratic coefficients), so that a small number of linear approximations suffices for approximating the nonlinear objective function quite well [11, 12]. The result is that in this case solving a (short) sequence of LPs to find the (non-approximated) ($\overline{\text{PR}}$) bound is preferable to solving one quadratic program as AP^{2}R does; furthermore, the latter seem to reoptimize much less efficiently when other valid inequalities are added and branching is performed, leaving PC as the uncontested best approach for this class of problems.

5 Conclusions

The paper presents results that very considerably extend the significance of the Projected Perspective Reformulation approach of [10]. The main contribution is the "project and lift" procedure giving rise to the Approximated Projected Perspective Reformulation approach, which allows the idea to be applied to any MINLP with nonlinear (separable) semicontinuous variables, possibly (but not necessarily) at the cost of some bound degradation, as opposed to the much narrower class of these satisfying assumption A2; furthermore, it allows direct and easy use of off-the-shelf MINLP solvers rather than requiring the development of ad-hoc codes. Coupled with the significant extension of the class of possible objective functions and to feasible regions having 0 in their interior, this allows the successful application of the Projected Perspective Reformulation to a much wider class of problems than previously possible. The computational experiments show that AP^{2}R is competitive with the best other available Perspective Reformulation approaches except in the extreme cases where either the problem is "easy" and with a very strong structure (cf. §4.1), or the problem strongly violates assumption A2 and a few linear approximations suffice for constructing a good estimate of the nonlinear objective function (cf. §4.4). Clearly, the trade-off here is mostly a technological issue, and it may change in the future according to the evolution of the relative efficiency of Quadratic Programming solvers w.r.t. Linear Programming ones, in particular during reoptimization. Hence, we believe that AP^{2}R can be a useful tool to have available in the "bag of tricks" of Mixed-Integer NonLinear Programming, especially since it is simpler to implement than the other alternatives. This is especially relevant in view of the fact that the list of applications that have been shown to benefit from Perspective Reformulation approaches is steadily growing [3, 6, 17, 18].

We also believe that the "project and lift" technique employed here could be useful in other contexts as well, possibly (but not necessarily exclusively) in the growing field of the study of convex envelopes for specially structured functions [19, 23, 25]. We find it particularly remarkable that a very substantial improvement of the continuous relaxation bound can be obtained with a technique that ultimately boils down to appropriately translating a continuous variable in a MINLP, leaving a problem with exactly the same size and structure of the original one. If such an approach could be replicated in other settings this could actually prove quite interesting for general MINLP; research in this direction is currently underway.

References

[1] K. Abhishek, S. Leyffer, and J. Linderoth. FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs. *INFORMS Journal on Computing*, 22:555–567, 2010.

[2] A. Agnetis, E. Grande, and A. Pacifici. Demand Allocation with Latency Cost Functions. *Mathematical Programming*, 132(1-2):277–294, 2012.

[3] S. Aktürk, A. Atamtürk, and S. Gürel. A Strong Conic Quadratic Reformulation for Machine-job Assignment with Controllable Processing Times. *Operations Research Letters*, 37(3):187–191, 2009.

[4] P. Bonami, M. Kilinç, and J. Linderoth. Algorithms and Software for Convex Mixed Integer Nonlinear Programs. In S. Leyffer J. Lee, editor, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 61–89. 2012.

[5] S. Ceria and J. Soares. Convex programming for Disjunctive Convex Optimization. *Mathematical Programming*, 86:595–614, 1999.

[6] X. Cui, X. Zheng, S. Zhu, and X. Sun. Convex Relaxations and MIQCQP Reformulations for a Class of Cardinality-constrained Portfolio Selection Problems. *Journal of Global Optimization*, online first, 2012.

[7] A. Frangioni and C. Gentile. Perspective Cuts for 0-1 Mixed Integer Programs. *Mathematical Programming*, 106(2):225–236, 2006.

[8] A. Frangioni and C. Gentile. SDP Diagonalizations and Perspective Cuts for a Class of Nonseparable MIQP. *Operations Research Letters*, 35(2):181 – 185, 2007.

[9] A. Frangioni and C. Gentile. A Computational Comparison of Reformulations of the Perspective Relaxation: SOCP vs. Cutting Planes. *Operations Research Letters*, 37(3):206 – 210, 2009.

[10] A. Frangioni, C. Gentile, E. Grande, and A. Pacifici. Projected Perspective Reformulations with Applications in Design Problems. *Operations Research*, 59(5):1225–1232, 2011.

[11] A. Frangioni, C. Gentile, and F. Lacalandra. Solving Unit Commitment Problems with General Ramp Contraints. *International Journal of Electrical Power and Energy Systems*, 30:316 – 326, 2008.

[12] A. Frangioni, C. Gentile, and F. Lacalandra. Tighter Approximated MILP Formulations for Unit Commitment Problems. *IEEE Transactions on Power Systems*, 24(1):105–113, 2009.

[13] A. Frangioni and E. Gorgone. A Library for Continuous Convex Separable Quadratic Knapsack Problems. *European Journal of Operational Research*, to appear, 2013.

[14] C.E. Gounaris and C.A. Floudas. Tight convex underestimators for C^2-continuous problems: I. univariate functions. *Journal on Global Optimization*, 42:51–67, 2008.

[15] O. Günlük, J. Lee, and R. Weismantel. MINLP Strengthening for Separable Convex Quadratic Transportation-Cost UFL. IBM Research Report RC24213, IBM Research Division, 2007.

[16] O. Günlük and J. Linderoth. Perspective Relaxation of MINLPs with Indicator Variables. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *Proceedings 13th IPCO*, volume 5035 of *Lecture Notes in Computer Science*, pages 1–16, 2008.

[17] O. Günlük and J. Linderoth. Perspective Reformulation and Applications. In S. Leyffer J. Lee, editor, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 61–89. 2012.

[18] H. Hijazi, P. Bonami, G. Cornuejols, and A. Ouorou. Mixed Integer NonLinear Programs featuring “On/Off” Constraints: Convex Analysis and Applications. *Electronic Notes in Discrete Mathematics*, 36(1):1153–1160, 2010.

[19] A. Khajavirad and N.V. Sahinidis. Convex envelopes generated from finitely many compact convex sets. *Mathematical Programming*, 137(1-2):371–408, 2013.

[20] C. Lemaréchal, A. Ouorou, and G. Petrou. A Bundle-type Algorithm for Routing in Telecommunication Data Networks. *Computational Optimization and Applications*, 44(3):385–409, 2009.

[21] S. Leyffer. Experiments with minlp branching techniques. In *European Workshop on Mixed Integer Nonlinear Programming*, 2010.

[22] R.A. Stubbs and S. Mehrotra. A Branch-and-Cut Method for 0-1 Mixed Convex Programming. *Mathematical Programming*, 86:515–532, 1999.

[23] M. Tawarmalani, J.-P.P. Richard, and C. Xiong. Explicit convex and concave envelopes through polyhedral subdivisions. *Mathematical Programming*, pages 1–47, 2012.

[24] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

[25] M. Tawarmalani and N.V. Sahinidis. Semidefinite relaxations of fractional programs via novel convexification techniques. *Journal of Global Optimization*, 20:133–154, 2001.

[26] M. Tawarmalani and N.V. Sahinidis. Convex Extensions and Envelopes of Lower Semi-continuous Functions. *Mathematical Programming*, 93:515–532, 2002.

[27] J.M. Zamora and I.E. Grossmann. A Global MINLP Optimization Algorithm for the Synthesis of Heat Exchanger Networks with no Stream Splits. *Compututers & Chemical Engineering*, 22:367–384, 1998.