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

Approximated Perspective Relaxations: a Project&Lift Approach

Antonio Frangioni · Fabio Furini · Claudio Gentile

the date of receipt and acceptance should be inserted later

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 continuous relaxation requires appropriate techniques. Under some rather restrictive assumptions, the *Projected PR* (P^2R) can be defined where the integer variables are eliminated by projecting the solution set onto 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 (AP^2R) whereby the projected formulation is "lifted" back to the original variable space, with each integer variable 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 providing substantially improved bounds. In the process we also substantially extend the approach beyond the original P^2R development by relaxing the requirement that the objective function be quadratic and the left endpoint of the domain of the variables be non-negative. While the AP^2R 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 MINLP software; this is shown to be competitive with previously proposed approaches in some applications.

Keywords Mixed-Integer NonLinear Problems, Semi-continuous Variables, Perspective Reformulation, Projection

Mathematics Subject Classification (2000) 90C06 · 90C25

1 Introduction

Mixed-Integer NonLinear Programs (MINLP) involving only convex function in their description have the advantage that solution methods can be devised by extending approaches designed for the (mixed-integer) linear case. It is not surprising, then, that this class of problems is the subject

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

Fabio Furini
LAMSADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris – France, E-mail: fabio.furini@dauphine.fr

Claudio Gentile
Istituto di Analisi dei Sistemi ed Informatica "A. Ruberti", Consiglio Nazionale delle Ricerche (IASI-CNR), Viale Manzoni 30, Rome – Italy. E-mail: gentile@iasi.cnr.it

of a very intense research; see e.g. [1,7,22,26] for surveys on applications and solution algorithms. In this paper we study convex separable MINLP with n semi-continuous variables $p_i \in \mathbb{R}$ for $i \in N = \{1, \ldots, n\}$. That is, each p_i either assumes the value 0, or lies in some given compact nonempty interval $\mathcal{P}_i = [p^i_{min}, p^i_{max}]$ $(-\infty < p^i_{min} < p^i_{max} < \infty)$; this 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

$$\min \; h(x) + \textstyle\sum_{i \in N} f_i(p_i) + c_i u_i \tag{1}$$
$$p^i_{min} u_i \leq p_i \leq p^i_{max} u_i \quad i \in N \tag{2}$$
$$(p, u, x) \in \mathcal{O} \tag{3}$$
$$u \in \{0,1\}^n \quad , \quad p \in \mathbb{R}^n \;\; , \;\; x \in \mathbb{R}^q . \tag{4}$$

We assume that the functions f_i are closed convex; w.l.o.g. we assume $f_i(0) = 0$ and that they are finite in the interval (p^i_{min}, p^i_{max}). Indeed, if, say, $f_i(\bar{p}) = +\infty$ for $\bar{p} < p^i_{max}$ then one could set $p^i_{max} = \bar{p}$ as by convexity $f_i(p) = +\infty$ for all $p \geq \bar{p}$ (however, cf. §2.3 for an example where $f_i(p^i_{max}) = +\infty$). The function h in the "other variables x" and the "other constraints (3)" do not play any role in our development and we make no assumptions on them. However our technique is especially well-suited for the case where the objective function and all the constraints in (1)–(3) are convex, so that the corresponding continuous relaxation is a convex program. Indeed, in all applications presented in this paper everything but the functions f_i is actually linear.

Problem (1)–(4) can be used to model many real-world problems such as distribution and production planning problems [33,11,16], financial trading and planning problems [12,9], and many others [5,6,21,22,20,23]. 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 the only source of non-convexity in (1)–(4) lies in the fact that one is actually dealing with the nonconvex functions

$$f_i(p_i, u_i) = \begin{cases} 0 & u_i = 0 \; , \; p_i = 0 \\ f_i(p_i) + c_i & u_i = 1 \; , \; p^i_{min} \leq p_i \leq p^i_{max} \\ +\infty & \text{otherwise} \end{cases} .$$

One can therefore strive to devise tight convex under-estimators 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. [7,19,30] among the many others). In this particular case it is actually possible to characterize its *convex envelope*, i.e., the best possible such under-estimator. 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 [27]; this leads to the *Perspective Reformulation* of (1)–(4) [8,11]

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

where $\tilde{f}_i(p_i, u_i) = u_i f_i(p_i / u_i)$ is the *perspective function* of $f_i(p_i)$. This actually applies even if p_i is a vector of variables and $\mathcal{P}_i$ a general polytope, but since in our subsequent development we actually need p_i to be a single variable we directly present this case. 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 $\{ \, (p_i, u_i) \, : \, p^i_{min} u \leq p_i \leq p^i_{max} u \, , \, u_i \in (0, 1] \, \}$, and it can be extended by continuity in $(0, 0)$ assuming $0 f_i(0/0) = 0$. In other words, the continuous relaxation of (5), dubbed the *Perspective Relaxation* ($\underline{\text{PR}}$), is (often, significantly) stronger than the continuous relaxation of (1)—(4), and therefore is a more convenient starting point to develop exact and approximate solution algorithms [6,11,12,16,21]. This, however, hinges on the ability to solve $\underline{\text{PR}}$ with efficiency comparable to the ordinary continuous relaxation, despite the fact that optimizing $\tilde{f}_i$ can be significantly more difficult than optimizing the original f_i (e.g., it is nondifferentiable in $(0, 0)$). For instance, one can reformulate (5) either as a Mixed-Integer Second-Order Cone Program (MI-SOCP) [6,13,21,32] (provided that the original objective function is SOCP-representable) or as a Semi-Infinite MINLP (SI-MINLP) [11].

Recently, another approach has been proposed [14] for the case where the functions f_i are quadratic, that transforms $\underline{\text{PR}}$ into 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 $2n$ continuous variables replacing the n variables $p_1, p_2, \ldots, p_n$, but with no u variables. When $\mathcal{O}$ has some valuable structure, this leads to the development of specialized solution approaches for $\underline{\text{PR}}$ 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. However, this comes at the cost of significantly restrictive assumptions on the data of the original problem (1)–(4), possibly the most binding one being that *each u_i only appears in the corresponding constraint (2), but not in constraints (3)*. While there are applications where this holds (§4.1, §4.2), in other cases the u_i variables are also used to express structural constraints of the problem (§4.3, §4.4) and therefore the technique cannot be used. Further negative side-effects of this removal are that valid inequalities concerning the u_i variables cannot be added to the relaxation, 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 has been invested in them.

In this paper we show that a simple reformulation trick can be used to overcome the above difficulties, although (potentially) at a cost. In Section 2 we generalize the P^{2}R approach of [14], where only the quadratic case is considered; instead, we show that only a simple condition on the functions f_i, satisfied by several classes of functions in addition to quadratic ones, is required to apply the P^{2}R technique. We also show that one further assumption in [14], $p^i_{min} \geq 0$, can be relaxed, albeit at the cost of a somewhat more involved analysis that is deferred to the appendix. Then, in Section 3 we introduce the reformulation trick that allows us to construct an *Approximated* Projected PR (AP^{2}R). This is done in two steps: first the problem is reformulated over the variables p and x only, like in the P^{2}R approach, as if no constraint of type (3) contained variables u. Once this is done, an MINLP reformulation is constructed which re-introduces the integer variables u in a different way to entirely encode the obtained piecewise-convex function. The continuous relaxation of P^{2}R (hereafter denoted as $\underline{\text{P}^2\text{R}}$) and that of AP^{2}R ($\underline{\text{AP}^2\text{R}}$) are equivalent only when there are no constraints of type (3) linking the variables u; in general, $\underline{\text{AP}^2\text{R}}$ provides a weaker lower bound (§3.1). Nevertheless, the new approach allows us to extend the P^{2}R idea to many more applications. Perhaps more importantly, it allows to use off-the-shelf MINLP software to solve it, thereby benefiting from all the sophisticated machinery it includes. On the contrary, P^{2}R requires the development of ad-hoc B&C codes and PC requires advanced features such as callback functions. Then we also present an alternative way to derivate the AP^{2}R model by the Reformulation Linearization Technique (RLT, cf. §3.2). Even if this second derivation does not prove any strong relationship with the PR (on the contrary to first derivation), it opens interesting research lines on the RLT and on some simple ways to improve continuous relaxation bounds. Finally, we show the benefits of the AP^{2}R approach in some practical applications; in particular, the idea is tested on one-dimensional sensor placement problems (cf. §4.1), single-commodity fixed-charge network design problems (cf. §4.2), mean-variance portfolio optimization problems with min-buy-in and portfolio cardinality constraints (cf. §4.3), and unit commitment problems in electrical power production (cf. §4.4).

2 P^{2}R for non-quadratic functions

We start by generalizing the analysis in [14] to a much larger class of functions. Since in this paragraph we only work with *one* block at a time, to simplify the notation we will drop the index "i", thus concentrating on the fragment

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

and on its $\underline{\text{PR}}$

$$\min \left\{ f(p,u) \;=\; \tilde{f}(p,u) + cu \;:\; p_{min}u \le p \le p_{max}u \;,\; u \in [0,1] \right\} \;. \tag{7}$$

The basic idea in [14] is to recast (7) as the minimization of the following (convex) function

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

of p alone; by convexity, the domain of z contains at least $conv(\; \{0\} \;\cup\; [p_{min}, p_{max}] \;)$. The 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

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

If $\tilde{u}(p)$ satisfying (9) exists and it is unique, it can be used to algebraically describe $u^*(p)$. In fact, if $\tilde{u}(p)p_{min} \le p \le \tilde{u}(p)p_{max}$ and $0 \le \tilde{u}(p) \le 1$ then clearly $u^*(p) = \tilde{u}(p)$; otherwise, $u^*(p)$ is the projection of $\tilde{u}(p)$ over the feasible region of (8). If instead (9) has no solution then the derivative always has the same sign and $u^*(p)$ can be similarly found by projection. Then, one has a case-by-case analysis of $u^*(p)$, which finally allows to obtain

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

In [14] this is done for the quadratic case $f(p) = ap^2 + bp$, where (9)

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

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

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

if and only if $c > 0$. We will now show that the $\mathrm{P}^2\mathrm{R}$ approach can be extended provided that the following property holds:

Property 1 Either (9) has no solution, or it has a unique solution of the form

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

for some values $g^+ \ge 0$ and $g^- \ge 0$ independent from p.

As we shall see, the fact that 0 lies in the interval (p_{min}, p_{max}) has a significant impact on the analysis; to simplify the presentation, we initially assume, as in [14], that $p_{min} \ge 0$; since $f(p,u)$ is only defined for $u \ge 0$, this implies that any solution $-pg^-$ to (9) is actually not relevant. Extending the analysis to the case where $p_{min} < 0$ (where $-pg^-$ becomes relevant) is actually possible, but somewhat more cumbersome, and therefore is avoided here for the sake of clarity of presentation; the details are available in the appendix.

To further simplify the presentation, we will assume $p_{min} > 0$, i.e., we will assume that p/p_{min} is always a well-defined quantity. If $p_{min} = 0$, the constraint $p_{min}u \le p$ 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 Property 1 holds, then $z(p)$ defined in (8) has the form*

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

where $p_{int} \in \{p_{min}, 1/g^+, p_{max}\}$ can be determined a-priori by a case-by-case analysis on the data of the problem.

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

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

Then we consider the following 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 $\frac{\partial f}{\partial u}(p,u)$ is negative for all $u \in [0,1]$, and therefore $u^*(p) = \min\{\ p/p_{min}\ ,\ 1\ \}$. This gives two sub-sub cases:
      a.1.1. $p/p_{min} \leq 1 \iff p \leq p_{min} \implies u^*(p) = p/p_{min} \implies$

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

      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

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

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

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

b. The only solution to (9) is given by (11). We consider three sub cases:
   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 sub cases 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

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

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

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

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

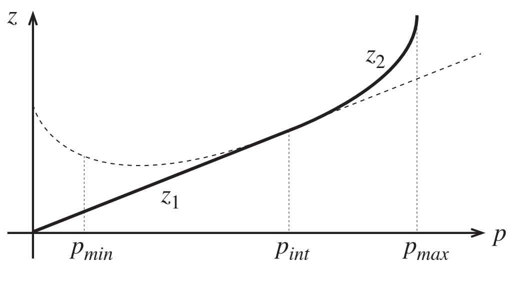

Fig. 1 The piecewise function $z(p)$

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 function 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.

Of course, the quadratic case is covered by the analysis (cf. (10)); an illustration of the process is provided by the following example.

Example 1 Consider the quadratic case such that $a = 2$, $b = 0$, $c = 8$, $p_{min} = 1$, $p_{max} = 10$. According to (10), $\tilde{u}(p) = p\sqrt{a/c} = p/2$, i.e., $g^+ = 1/2$. This means that we are in case b.2 in the proof of Proposition 1, as $10 = p_{max} \geq 1/g^+ = p_{int} = 2 \geq p_{min} = 1$. Hence $z(p)$ has the form of (19)

$$z(p) = \begin{cases} 8p & \text{if } 0 \leq p \leq 2 \\ 2p^2 + 8 & \text{if } 2 \leq p \leq 10 \end{cases} .$$

Obviously, the above formula can be statically computed once the problem is completely defined.

2.1 The rational exponent case

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

$$c - a\left(\frac{k}{h} - 1\right)\left(\frac{p}{u}\right)^{\frac{k}{h}} = 0 \tag{20}$$

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 $c \leq 0$ then the derivative is always negative (cf. point a.1 in the proof of Proposition 1) for $p \geq 0$, while, if $c \geq 0$ and k is odd, the derivative is always positive (cf. point a.2 in the proof of Proposition 1) for $p \leq 0$; in both cases (20) 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 2 If $k = 3$ (odd case), $h = 2$, $a = 1$, $c = 4$, and $0 \leq p_{min} \leq 4 \leq p_{max}$, one has

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

2.2 The exponential case

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

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

It is easy to verify that $g(x) = e^x(1 - x) \leq 1$ (the maximum being attained at $x = 0$); 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 $PL(x)$ (known as the "ProductLog" function) gives the principal solution for w in $x = we^w$, which is real for all $x \geq -1/e$; this can be efficiently computed numerically for a fixed argument such as c/e. Since in our case $x = c/e$, $\tilde{u}(p)$ 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 $w = 1$, $g^+ = 1/2$, $\tilde{u}(p) = p/2$, and hence

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

2.3 The Kleinrock delay function case

Another interesting non-quadratic 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. [25]). The function is convex as long as $0 \leq p_{min} \leq p < p_{max}$ and $a > 0$; then, by applying the Perspective Relaxation (7)

$$f(p,u) \;=\; uf(p/u) + cu = \frac{au^2}{up_{max} - p} + cu$$

with constraints $p \in [up_{min}, up_{max})$ and $u \in [0,1]$. 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 $p_{max}(cp_{max} + a)$. For $c > -a/p_{max}$, the form has the two roots

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

and therefore $\partial f / \partial u \leq 0$ for $\tilde{u}_{-}(p) \leq u \leq \tilde{u}_{+}(p)$ (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 $c \leq -a/p_{max}$ instead, then $\partial f / \partial u$ is always positive, i.e., $f(p,u)$ is always non increasing 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 $g^+ = 1/8$, $\tilde{u}(p) = p/8$, and hence

$$z(p) = \begin{cases} p/4 & \text{if } 0 \leq p \leq 8 \\ 4/(12 - p) + 1 & \text{if } 8 \leq p \leq 12 \end{cases} \quad .$$

3 Project and Lift

As already mentioned in the introduction, one of the main limitations of the $\mathrm{P}^2\mathrm{R}$ 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. In this section we show how to "lift back" the obtained piecewise characterization of the convex envelope in the original space. The result is somewhat surprising, since (at least if $p_{min} \geq 0$) what one ends up with is a (convex, if the original continuous relaxation was) program with *exactly the same size and structure* as the original one, but which provides a (much) better bound. This in turn allows us to apply the approach in the case where 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 PR. The idea is 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.

We start by introducing the required reformulation trick. Just like the previous section, we analyze the somewhat simpler case where $p_{min} \geq 0$ first and postpone the case $p_{min} < 0$ to the appendix.

The projected function $z(p)$ of Proposition 1 can always be *formulated* in terms of an appropriate nonlinear program by exploiting the following very well-known result (e.g., see [14]).

Lemma 1 *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

$$\gamma(p) = \begin{cases} \min \ \gamma_1(p_1 + \alpha_0) + \sum_{i=2}^{k} \big(\gamma_i(p_i + \alpha_{i-1}) - \gamma_i(\alpha_{i-1}) \big) \\ \qquad 0 \leq p_i \leq \alpha_i - \alpha_{i-1} \qquad i = 1, \ldots, k \\ \qquad \alpha_0 + \sum_{i=1}^{k} p_i = p \end{cases} . \tag{21}$$

Moreover, for any $p \in [\alpha_0, \alpha_k]$ let h be the smallest index such that $p \in [\alpha_{h-1}, \alpha_h]$: there always exists an optimal solution $p^* = [p_1^*, \ldots, p_k^*]$ to problem (21) such that $p_i^* = \alpha_i - \alpha_{i-1}$ for $i < h$, $p_i^* = 0$ for $i > h$, and $p_h^* = p - \alpha_{h-1}$.

Intuitively, Lemma 1 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 1 *For $z(p)$ defined in (12) and*

$$\bar{z}(p) = \begin{cases} \min_{q,u} \ h(u,q) = uz(p_{int}) + z_2(q + p_{int}) - z(p_{int}) \\ \qquad\qquad (p_{min} - p_{int})u \leq q \leq (p_{max} - p_{int})u \\ \qquad\qquad p = p_{int}u + q \quad , \quad u \in [0,1] \end{cases} , \tag{22}$$

we have $\bar{z}(p) = z(p)$ for any $p \in [0, p_{max}]$.

Proof We start by applying (21) to (12): we have $k = 2$, $\alpha_0 = 0$, $\alpha_1 = p_{int}$, $\alpha_2 = p_{max}$, and recalling that $z(p_{int}) = z_2(p_{int})$ we obtain that (12) can be alternatively computed as

$$z(p) = \begin{cases} \min_{p_1,p_2} \ z_1(p_1) + z_2(p_2 + p_{int}) - z(p_{int}) \\ \qquad 0 \leq p_1 \leq p_{int} \quad , \quad 0 \leq p_2 \leq p_{max} - p_{int} \quad , \quad p = p_1 + p_2 \end{cases} . \tag{23}$$

In order to prove the thesis we have therefore to show that (22) and (23) are equivalent, i.e., they have the same objective function value for all p.

The identification $p_1 = p_{int}u$ and $p_2 = q$ readily shows that the two problems are very similar. In fact, the constraints $u \in [0, 1]$ and $p = p_{int}u + q$ in (22) are then identical to the constraints $p_1 \in [0, p_{int}]$ and $p = p_1 + p_2$, respectively, in (23). Also, the two objective functions are easily seen to be identical (recall that z_1 is linear). The only non obvious argument is that the constraint

$$(p_{min} - p_{int})u \leq q \leq (p_{max} - p_{int})u \tag{24}$$

in (22) is *not* equivalent to $0 \leq p_2 \leq p_{max} - p_{int}$ in (23); indeed, its right-hand side is stronger ($u \leq 1$) while its left-hand side is weaker ($p_{min} - p_{int} \leq 0$). Nonetheless, the two problems are equivalent: for any fixed p, we can prove that there exists an optimal solution (p_1^*, p_2^*) of (23) that is feasible for (22), and an optimal solution (q^*, u^*) of (22) that is feasible for (23).

For the first part, we take an optimal solution (p_1^*, p_2^*) of (23) and we construct an equivalent (q^*, u^*) for (22). This is easily done: due to Lemma 1, (p_1^*, p_2^*) satisfies

1. either $p < p_{int}$, in which case $p_1^* < p_{int}$ $(= \alpha_1)$ and $p_2^* = 0$, so that we set $q^* = 0$ $(= p_2^*)$ and $u^* = p_1^*/p_{int} < 1$;
2. or $p \geq p_{int}$, in which case $p_1^* = p_{int}\,(= \alpha_1)$ and $0 \leq p_2^*$ $(\leq p_{max} - p_{int})$, so that we set $p_2^* = q^*$ (≥ 0) and $u^* = 1$.

It is immediate to verify that in either case the thusly constructed (q^*, u^*) is feasible for (22)—in particular, (24) is satisfied—and equivalent to the original (p_1^*, p_2^*) in terms of the objective function value. Hence, $\bar{z}(p) \leq z(p)$.

For the other direction, we consider (q^*, u^*) optimal for (22). It is easy to see that if $q^* \geq 0$ then $(p_1^*, p_2^*) = (p_{int}u^*, q^*)$ is feasible for (23), which proves that $\bar{z}(p) \geq z(p)$, and hence our thesis. We therefore want to prove that there always exists an optimal solution to (22) with $q^* \geq 0$. As before, we separately analyze the two cases $p \geq p_{int}$ and $p < p_{int}$. In the former, necessarily $q^* \geq 0$.

Indeed, the constraint $p = p_{int}u + q$ implies $u^* = (p - q^*)/p_{int}$, so $q^* < 0$ (together with $p \geq p_{int}$) would imply $u^* > 1$. Therefore, only the case $p < p_{int}$ still needs to be examined. It is easy to prove that, in this case, $(u^*, q^*) = (p/p_{int}, 0)$ is optimal (and hence, as desired, $q^* \geq 0$). In fact, due to the constraint $p = p_{int}u + q$, any alternative feasible solution of (22) can be written in the form $(u, q) = (u^* + \varepsilon/p_{int}, -\varepsilon)$, where ε is arbitrary in sign. The objective function value of such a solution is

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

Now, because z_1 is linear we have $z(p_{int})/p_{int} \in \partial z(\tilde{p})$ for each $\tilde{p} \in [0, p_{int}]$, and taking in particular $\tilde{p} = p_{int}$ we can write the subgradient inequality

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

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

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

and therefore $h(u^*, q^*) = p(z(p_{int})/p_{int}) \leq h(u, q)$. Since (u, q) is arbitrary, (u^*, q^*) is optimal. □

3.1 Approximated Projected Perspective Reformulation

We are now in the position to introduce the new approach, which we call the *Approximated Projected Perspective Reformulation* (AP^2R) of the MINLP (1)—(4). This is given by the following MINLP

$$\begin{array}{lll} \min & h(x) + \sum_{i \in N} h_i(u_i, q_i) & \\ & (p^i_{min} - p^i_{int})u_i \leq q_i \leq (p^i_{max} - p^i_{int})u_i & i \in N \\ & p_i = p^i_{int}u_i + q_i & i \in N \ , \\ & (p, u, x) \in \mathcal{O} & \\ & u \in \{0,1\}^n \quad , \quad p, q \in \mathbb{R}^n \quad , \quad x \in \mathbb{R}^q & \end{array} \tag{25}$$

where for each $i \in N$, $h_i(u_i, q_i)$ is defined as $h(u, q)$ in (22). Note that if, e.g., all other constraints of the problem involving p_i are linear, then one can use $p_i = p^i_{int}u_i + q_i$ in (25) as a *redefinition* for p_i and substitute the latter away in the problem, producing a formulation with exactly the same number of variables as the original one. We numerically illustrate the approach on the same data of Example 1.

Example 5 Consider a fixed index i, that for the sake of clarity we omit in the following. If $a = 2$, $b = 0$, $c = 8$, $p_{min} = 1$, $p_{max} = 10$, then the original problem is

$$\min \left\{ \, 2p^2 + 8u \ : \ u \leq p \leq 10u \ \ , \ \ u \in \{0,1\} \, \right\}. \tag{26}$$

As seen in Example 1, $z_2(p) = 2p^2 + 8$, $p_{int} = 2$ and $z(p_{int}) = 16$, thus the AP^2R is

$$\min \left\{ \, [\, 2(q + 2)^2 + 8 \,] + 16u - 16 \ : \ -u \leq q \leq 8u \ \ , \ \ p = 2u + q \ \ , \ \ u \in \{0,1\} \, \right\} \ . \tag{27}$$

It is easy to verify (27) provides a better bound for fractional values of u. For instance, the solution $(p, u) = (2, 1/5)$ in the relaxation of (26) has objective function value 9.6, whereas the equivalent $(q, u) = (8/5, 1/5)$ in the relaxation of (27) has value 21.12. The fair comparison, however, is between the minimal objective values attained for a fixed value of p (see also Example 6 and Figure 2); for $p = 2$, $u = 1/5$ is indeed optimal for (26), whereas $u = 1$, $q = 0$ is optimal for (27) (cf. Theorem 1), yielding the (larger) bound 16.

This justifies the introduction of the somewhat troubling constraint $(p_{min} - p_{int})u \leq q$ in place of the natural (and stronger) $q \geq 0$, because then not only (22) is a *reformulation* of the PR, but if one adds the integrality constraint $u \in \{0, 1\}$ it also provides a *reformulation of the original integer program* (6) *whose ordinary continuous relaxation is (equivalent to)* $\underline{\text{PR}}$ *if there are no constraints in* $\mathcal{O}$ *containing the* u. 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. AP^{2}R is a simple algebraic reformulation to a problem with "the same degree of nonlinearity" as the original problem. The "Approximated" tag is related to the fact that the continuous relaxation of AP^{2}R (denoted as $\underline{\text{AP}^2\text{R}}$ in the following) is in general a relaxation of $\underline{\text{PR}}$. 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 [14]) 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);
- AP^{2}R has (at least in the linear case) 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 [32,6,13,21] has at least one more variable (per block), while the SI-MILP formulation [11] has one variable and infinitely many more constraints (cf. §4.3).

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

- Its continuous relaxation $\underline{\text{AP}^2\text{R}}$ may provide weaker bounds than $\underline{\text{PR}}$ when there are constraints in $\mathcal{O}$ binding the u (cf. §4.3 and §4.4);
- Unlike P^{2}R, AP^{2}R does not get any specific advantage of the combinatorial structure embedded in $\underline{\text{PR}}$ (e.g., single-commodity Min-Cost Flow problems when it is applied to single-commodity Network Design problems, cf. §4.2) unless the general-purpose solver used is able to detect and exploit it;
- 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 (22) have the slightly simpler form $(p_{min} - p_{max})u \leq q \leq 0$);
- compared with the MI-SOCP formulation [32,6,13,21], 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 [11], 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 An interesting relationship

It is worth remarking that an alternative, purely algebraic way exists for obtaining (22) from (6) in the quadratic case. In fact, applying the variable transformation $p = p_{int}u + q$ to (6) leads to the reformulation

$$\begin{array}{l} \min \;\; a(q + p_{int}u)^2 + b(q + p_{int}u) + cu \\ \quad (p_{min} - p_{int})u \leq q \leq (p_{max} - p_{int})u \quad . \\ \quad p = p_{int}u + q \qquad , \qquad u \in \{0,1\} \end{array}$$

This is *not* a simple separable MIQP like the original problem, due to the bilinear term in u and q produced by the expansion of the quadratic term (which also contains a "u^2" term). However, since $u \in \{0,1\}$ and $u = 0 \Longrightarrow q = 0$ due to (2) one can use the relationships

$$u^2 = u \qquad\qquad qu = q \tag{28}$$

and therefore (6) can be reformulated as

$$\begin{array}{l} \min aq^2 + (2ap_{int} + b)q + (ap_{int}^2 + bp_{int} + c)u \\ \quad (p_{min} - p_{int})u \leq q \leq (p_{max} - p_{int})u \\ \quad p = p_{int}u + q \qquad , \qquad u \in \{0,1\} \end{array} \tag{29}$$

which is easily seen to coincide with (22). The above process is closely related to the *Reformulation Linearization Technique* (RLT) [2,3,4], which strengthens the continuous relaxation of mixed-integer programs by exploiting the *strengthening relations* (28) (since for $u \in [0,1]$, $u \geq u^2$ and $q \geq qu$). The development of §3 is more general, for at least two reasons:

1. the RLT approach does not seem to easily extend beyond the polynomial case;
2. in order to obtain the best possible bound p_{int} has to be properly chosen, but (29) in itself provides no clue about how to do it.

The following example shows the importance of the second point above:

Example 6 Consider the quadratic case of examples 1 and 5. Without our analysis, the natural way to apply (28) would be to define $p = p_{min}u + t$ with $0 \leq t \leq (p_{max} - p_{min})u$. In our case this would yield

$$\min \left\{ 2t^2 + 4t + 10u \; : \; 0 \leq t \leq 9u \;\; , \;\; p = u + t \;\; , \;\; u \in \{0,1\} \right\}$$

This reformulation, that only differs from AP^{2}R in the choice of the translation (using p_{min} instead of a carefully chosen p_{int}) is weaker than the AP^{2}R of Example 5. To see that one can project it on the p variable only, minimizing over u and q similarly to the development of §2; some tedious algebra shows that this results in

$$\tilde{z}(p) = \begin{cases} p(230 + 81p)/50 & 0 \leq p \leq 5/3 \\ -9/2 + 10p & 5/3 \leq p \leq 5/2 \\ 8 + 2p^2 & 5/2 \leq p \leq 10 \end{cases} \quad .$$

This can be compared with $z(p)$ (cf. Example 1). We also compare it with the function $\underline{z}(p)$ obtained by projecting away u in the original formulation (cf. Example 5), which turns out to be $\underline{z}(p) = (4/5)p + 2p^2$, in order to show once again the improvement of the bound due to the PR technique. The three functions are plotted in Figure 2 in the interval $[0, 5/2]$, after which $z(p)$ coincides with $\tilde{z}(p)$, showing that while the application of the strengthening relationships (28) does a large part of the gap closing, an appropriate choice of the origin for the translation obtains an even better bound. Besides, AP^{2}R results in a more compact formulation than the three-piece (whose initial piece is not even linear) $\tilde{z}(p)$, hence there is no reason to choose any other initial point than the one prescribed by our analysis.

The above observation is potentially interesting because the computation of convex envelopes for specially-structured functions of "a few" variables is an important field in which several advances are being made; for instance, one of the most investigated structures is that of functions $\phi(p,u) = f(p)g(u)$ where f is convex and g is concave [24,29,31]. The recent paper [24] shows that the characterization of the convex envelope is possible in terms of piecewise functions similar to those of §2; however, the development in [24] 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, and that the idea of appropriately choosing a translation of the variable to improve the lower bound in MINLP problems may find even wider application.

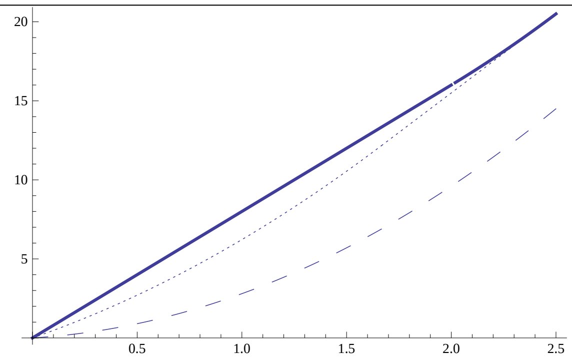

Fig. 2 Comparison of three reformulations: $z(p)$ (solid), $\underline{z}(p)$ (dashed), $\tilde{z}(\mathrm{p})$ (dotted)

4 Computational results

In this section we report results of computational tests performed on four classes of (MIQP)s with semi-continuous variables. For all the problems, it has already been clearly shown [11,12,13,14] that approaches based on the PR 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 PR. Among these, the SI-MILP formulation has been shown to be consistently more effective than the MI-SOCP one [13], 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 [14], denoted as "$\mathrm{P}^2\mathrm{R}$", when Assumption A2 holds and a specialized solver is available, and the newly proposed approach, denoted as "$\mathrm{AP}^2\mathrm{R}$". We will also denote as $\underline{\mathrm{PC}}$, $\underline{\mathrm{P}^2\mathrm{R}}$ and $\underline{\mathrm{AP}^2\mathrm{R}}$, respectively, the continuous relaxations for the problem at hand (to be deduced from the context) corresponding to the three reformulation approaches.

The experiments have been performed on a computer with a 3.40 Ghz 8-core Intel Core i7-3770 processor and 16Gb RAM, running a 64 bits Linux operating system. All the codes were compiled with `gcc 4.6.3` and `-O3` optimizations, using `Cplex 12.6.0` (ran single-threaded). PC and $\mathrm{AP}^2\mathrm{R}$ entirely rely on the (sophisticated) B&C machinery of `Cplex`. We have used as much as possible the standard parameters setting; in particular, the stopping condition of the B&C is an optimality gap below 0.01%. The only exceptions are that for using PC some reductions have to be deactivated, as this is necessary in order to be able to insert "lazy constraints", which is how `Cplex 12` now handles formulation with a very large number of constraints (the mechanism was somewhat different in previous versions). Furthermore, due to the nonlinear nature of the PR, sometimes the cuts produced by PC are rather badly scaled, which may create numerical problems. In order to solve them, it was occasionally needed (in particular, for the instances of §4.1) to turn on the "numerical emphasis" switch in `Cplex` and to sharpen the numerical tolerances, such as those for RHS violation; when this is done, it is done uniformly for all approaches. $\mathrm{P}^2\mathrm{R}$ instead 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%.

As suggested by the Referees, we tested several options to see if they significantly impacted the results. Among them:

- We experimented with providing to the solver the optimal solution and disabling the heuristics, so as to gauge the effect of the different formulations to the bound computation only, removing any side effect on the heuristics. The results quite closely matched the ones where the heuristics are ran, proving that the heuristic do not behave significantly differently for the two formulations.
- We verified if the option for dynamic linearization of the quadratic objective function in `Cplex` ("`mip strategy miqcpstrat 2`") improved the performances of $\mathrm{AP}^2\mathrm{R}$. However, this did not happen: the performances were very similar, usually slightly worse. This confirmed the results of [13], obtained in the context of the MI-SOCP formulation.
- We tested different configurations for the presolver and the strong branching (disabling it, forcing it) in `Cplex`. For $\mathrm{AP}^2\mathrm{R}$, neither of these options had a significant impact on the efficiency

of the algorithm. For PC, some combination of these options did have a more visible impact but there was no clear winner, i.e., there were both instances where the performances improved as well as instances where they deteriorated.

For all these reasons, we found it appropriate to just report results using as much as possible the default parameters, and we don't further discuss the details of those experiments in the paper.

As far as $\mathrm{P^2R}$ is concerned, the B&B used is not particularly sophisticated (see [14] for details), and it could surely 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 `callback` functions for PC). Besides, they have several sophisticated options that can be activated or improved by appropriate parameter tuning, which, as discussed above, 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 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. The MIQP formulation

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

exhibits structure (3), and $\underline{\mathrm{P^2R}}$ boils down to a continuous convex quadratic knapsack problem with at most $2n$ variables that can be solved in $O(n \log n)$ [18].

We tested 210 random instances of SP, grouped in 10 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 "P" classes, with 36 instances each, derive from random instances of the PARTITION problem, according to the $\mathcal{NP}$-hardness proof for SP [5]. All these have already been used in [14], to which the interested reader is referred for further details. Because these instances were almost invariably solved at the root node by both $\mathrm{P^2R}$ and $\mathrm{AP^2R}$ (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 [5] (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 class of instances we report the gap between the continuous relaxation and the optimal solution (column "gap") only once since, as predicted by the theory, it is identical for all three formulations; note that we here intend the gap w.r.t. the "natural" relaxation, prior than `Cplex` starts adding cuts (for PC and $\mathrm{AP^2R}$). However, for this class of problems cuts have little to no effect, which is hardly surprising due to the clear lack of exploitable combinatorial structure. For each approach individually we then report, averaged among the instances of each group, the number of B&B nodes required to solve the problem to optimality (column "nodes"), the total running time (column "avg"), the relative standard deviation of the running times (i.e., the ratio between the standard deviation and the mean, column "dev") and the time it took to solve the root node relaxation (column "root"). The smallest average running time in each row is emphasized in bold.

Table 1 shows that, as already reported in [14], projected formulations are by far the most effective way to solve this (simple, yet $\mathcal{NP}$-hard) problem. While PC is more effective than the

	PC				P2R				AP2R				
	nodes	time			nodes	time			nodes	time			gap
		avg	dev	root		avg	dev	root		avg	dev	root	
2000-h	9	7.42	0.28	6.68	0	**0.07**	0.08	0.04	0	0.11	0.46	0.08	0.000
3000-h	6	18.08	0.22	16.85	0	0.16	0.02	0.09	0	**0.16**	0.40	0.12	0.000
2000-l	0	3.27	0.00	3.26	0	**0.02**	0.33	0.01	0	0.05	0.12	0.03	0.000
3000-l	0	7.65	0.00	7.66	0	**0.04**	0.13	0.02	0	0.07	0.09	0.04	0.000
P-2000	287	63.70	0.67	9.94	0	**0.07**	0.07	0.04	2	1.19	0.22	1.04	0.001
P-3000	514	181.46	0.67	23.97	0	**0.17**	0.03	0.09	1	2.55	0.20	2.35	0.001
p50	72	0.12	0.60	0.01	583	0.06	0.87	0.00	23	**0.02**	0.75	0.00	0.003
p100	104	0.48	0.60	0.07	10897	3.86	0.77	0.00	73	**0.06**	0.32	0.01	0.003
s50	231	0.22	0.52	0.02	747	0.07	1.09	0.00	143	**0.04**	0.85	0.00	0.029
s100	1411	1.45	0.50	0.07	23396	7.53	1.15	0.00	1109	**0.61**	1.12	0.00	0.029

Table 1 Results of the SP problem

MI-SOCP formulation, and much more so than using the standard continuous relaxation [14], it is considerably outperformed by P^2R among all instances that require a few or no branching nodes. This is partly due to the fact that $\underline{P^2R}$ is much faster to solve than $\underline{PC}$, as the "root" columns clearly show. Furthermore, very accurately solving $\underline{P^2R}$ pays off surprisingly well in this case: while the exact solution of $\underline{P^2R}$ produces a feasible solution which often immediately closes the gap, the approximate solution of $\underline{PC}$ (due to the fact that it has, in fact, an infinite number of constraints) can require a significant amount of branching to achieve the same effect (e.g., on the P instances).

Among projected methods, P^2R is faster than AP^2R when little or no branching is required; by a slim margin on the random instances, by a more significant one on the P ones. This is essentially because, as it can be expected, the specialized $O(n \log n)$ solution algorithm [18] used to solve $\underline{P^2R}$ is faster, as the "root" columns show (by up to more than an order of magnitude on the P instances). It is worth remarking that the advantage of a specialized approach is at times compounded by the fact $\underline{P^2R}$ is a strictly smaller continuous program than $\underline{AP^2R}$ 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). Clearly, SP is a worst case scenario as far as AP^2R vs. P^2R goes: the latter never requires any branching on those instances, so the faster relaxation pays off while the more efficient branching and cutting techniques uniquely available to AP^2R have no impact.

However, when branching is required ($p*$ and $s*$ instances) because the root node gap is not small enough, AP^2R is competitive with P^2R, outperforming it by as much as two orders of magnitude. This is due to all the sophisticated machinery (preprocessing, branching, heuristics, cutting planes, ...) in `Cplex`, which 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 $G = (V, A)$, where each node $i \in V$ 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 p^{ij}_{max}, paying a fixed cost c_{ij} if a nonzero amount of flow p_{ij} transits along the arc; flow cost is a convex quadratic function $b_{ij}p_{ij} + a_{ij}p^2_{ij}$. The MIQP formulation

$$
\begin{aligned}
\min\ & \textstyle\sum_{(i,j)\in A}(c_{ij}u_{ij} + b_{ij}p_{ij} + a_{ij}p^2_{ij}) \\
& \textstyle\sum_{(j,i)\in A} p_{ji} - \sum_{(i,j)\in A} p_{ij} = d_i \qquad && i \in V \\
& 0 \le p_{ij} \le p^{ij}_{max}u_{ij} \ \ , \ \ u_{ij} \in \{0,1\} && (i,j) \in A
\end{aligned} \tag{30}
$$

exhibits structure (3) together with a strong network structure, so that $\underline{\text{P}^2\text{R}}$ 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 [14]. 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 [14], 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.

	PC				P2R				AP2R				gap
	nodes	time			nodes	time			nodes	time			
		avg	dev	root		avg	dev	root		avg	dev	root	
1000-l	3	13.58	0.27	4.38	4	**0.17**	4.20	0.02	3	0.26	0.33	0.21	0.002
1000-h	3	11.48	0.46	3.59	2	**0.10**	4.51	0.02	2	0.22	0.44	0.19	0.001
2000-l	144	48.58	1.69	21.98	759	96.46	6.83	0.03	109	**1.10**	1.75	0.58	0.004
2000-h	56	33.10	0.63	19.36	61	8.60	4.75	0.03	32	**0.81**	0.81	0.56	0.004
3000-l	230	98.59	1.60	48.27	670	111.09	6.33	0.05	143	**2.06**	1.46	1.08	0.005
3000-h	45	62.96	0.60	48.95	48	8.72	7.04	0.05	26	**1.36**	0.72	1.03	0.003

Table 2 Results of the ND problem

The results are reported in Table 2, with precisely the same organization as Table 1, and confirm those for the SP problem. Again, `Cplex` cuts have no discernible effect on the root node gap (this is not shown in the tables to save on space). $\underline{\text{PC}}$ is much more costly than $\underline{\text{P}^2\text{R}}$ and $\underline{\text{AP}^2\text{R}}$, which prevents PC from being competitive. For the small and "easy" instances with 1000 nodes, P^2R outperforms somewhat AP^2R; however, this is less so than the difference between $\underline{\text{P}^2\text{R}}$ (using a specialized network flow solver) and $\underline{\text{AP}^2\text{R}}$ times would suggest. This is due to a few outliers that require considerably more time than the average, skewing it (as the larger standard deviations show). We can therefore assume that P^2R would also be faster than AP^2R on the other half of the test bed of [14] that we left out. However, as the size of the instances grow all methods start to require branching: here, exploiting `Cplex` machinery again pays off, as both PC and AP^2R require significantly less nodes than P^2R. This in fact again makes PC occasionally competitive with P^2R (3000-l), but AP^2R is much faster than both. It also has comparable standard deviations as PC but much smaller ones than P^2R, meaning that it is also significantly more stable.

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 N 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 *non-separable* (MIQP) formulation is

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

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 has to be modified to extract the "largest" possible diagonal part; that is, one must find a "large" diagonal matrix D such that $Q - D$ is still positive semidefinite. The PR technique is then applied to the diagonal part of the objective function corresponding to D, while leaving the rest untouched; see [11] for details. In particular, efficient and effective D can be found with SemiDefinite Programming techniques, ad discussed in [12]. For our tests we used the 90 randomly-generated instances, 30 for each value of $n \in \{200, 300, 400\}$, already employed in [11,12] 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 by "+", "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 [12].

An important remark is that, once the objective function is made separable, the problem is actually suitable for P^{2}R. It was not considered in [14] because it lacks exploitable structure to develop specialized solution algorithms for $\underline{\text{P}^2\text{R}}$, 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

$$\sum_{i \in N} u_i \leq k \tag{31}$$

for some $k \leq n = |N|$. This provides a useful "gauge": decreasing k "increases the amount of non-separability" in the model, possibly impacting on the tightness of the $\underline{\text{AP}^2\text{R}}$ 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, (31) does impact on the tightness of the relaxation and on the performances of AP^{2}R.

	PC				AP2R				
	nodes	time			nodes	time			gap
		avg	dev	root		avg	dev	root	
200^+	4766	8.00	2.38	0.20	539	**0.72**	0.66	0.13	1.138
200^0	6394	10.39	1.43	0.19	9212	**7.02**	1.55	0.11	2.137
200^-	109421	136.19	1.36	0.18	118159	**77.17**	1.41	0.11	3.648
300^+	5656	22.84	1.45	0.56	4291	**6.02**	2.06	0.33	1.301
300^0	18342	60.25	1.24	0.54	23566	**29.56**	1.23	0.32	1.987
300^-	35990	114.40	1.76	0.55	33399	**43.65**	1.61	0.33	2.679
400^+	27241	208.12	1.69	1.44	25616	**52.17**	2.26	0.80	1.430
400^0	118869	722.60	1.69	1.17	132596	**260.64**	1.82	0.69	2.298
400^-	272348	1583.82	0.98	1.16	276225	**553.26**	0.92	0.68	3.063

Table 3 Results of the MV problem

The results are reported in Table 3 for the plain version of the problem, and in Table 4 for that with the constraint (31). Each row of the tables reports average results between 10 instances with the same characteristics, and the arrangement of the columns in Table 3 is the same as in the previous cases. For Table 4, the only difference is that the root node gap is reported separately for each approach, since, as the theory predicts, the one of AP^{2}R is different (larger) than the one of PC. Again, this is the gap of the continuous relaxation without cuts, which has the following interesting behavior (not reported in the tables for clarity). Without the cardinality constraints, the initial gaps are identical and cuts do help somewhat, fractionally reducing the gap (especially

	PC					AP2R				
	nodes	time			gap	nodes	time			gap
		avg	dev	root			avg	dev	root	
200^+	69	3.68	0.98	0.28	0.507	212	**0.43**	0.47	0.12	0.766
200^0	8781	153.75	1.58	0.27	2.748	24868	**22.01**	1.27	0.12	3.139
200^-	38028	674.13	1.35	0.29	4.173	173844	**157.54**	1.48	0.12	4.745
300^+	181	18.02	1.05	0.76	0.491	1303	**2.66**	0.69	0.33	1.080
300^0	19731	824.02	1.24	0.77	2.344	69706	**109.02**	1.25	0.34	2.906
300^-	88286	3409.24	0.82	0.75	3.573	440656	**704.32**	1.35	0.35	3.923
400^+	98	28.39	0.68	1.68	0.405	985	**3.68**	0.54	0.77	0.855
400^0	42531	3608.04	1.79	1.69	2.336	329242	**849.38**	1.81	0.71	2.999
400^-	121777	13608.03	2.93	1.71	3.798	1821932	**4769.89**	1.11	0.71	4.528

Table 4 Results of the MV problem with the cardinality constraint

on "$^+$" instances, less on "^{0}" ones and very little on "$^-$" ones); however, the reduction in the gap is identical for PC and AP^{2}R. With the cardinality constraints, the initial gaps are somewhat different, as Table 4 shows, but cuts do not have any effect, so the difference remains the same.

The Tables clearly show that AP^{2}R neatly outperforms PC. This is due to the fact that $\underline{\text{AP}^2\text{R}}$ is faster than $\underline{\text{PC}}$. This is already true at the root node, as the "root" column shows, but it is even more pronounced in reoptimization: $\underline{\text{AP}^2\text{R}}$ is re-solved after branching much more efficiently than $\underline{\text{PC}}$. This is clearly visible e.g. in the 400^- instances in Table 3: the two approaches require very nearly the same number of nodes, and $\underline{\text{AP}^2\text{R}}$ is less than two times faster than $\underline{\text{PC}}$ at the root node, yet overall it ends up almost three times faster. Even more dramatically, in Table 4 for the 400^+ instances $\underline{\text{AP}^2\text{R}}$ is slightly more than two times faster than $\underline{\text{PC}}$ at the root node: yet, despite requiring an order of magnitude more nodes, it ends up being almost an order of magnitude faster.

The introduction of the cardinality constraint (31) changes the behavior somewhat, but still AP^{2}R is clearly the best approach. This is despite the fact that the $\underline{\text{AP}^2\text{R}}$ bound is somewhat weaker, as testified by the visibly larger root node gap and by the fact that the number of nodes is always larger, often by about one order of magnitude. In fact, with the exception of the "$^+$" instances, cardinality constrained MV problems are harder to solve than those without (31). However, this is true for PC as well, and overall the total running time of AP^{2}R is always better than that of PC by a significant margin.

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 electrical generators 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 [15,16]. 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 [28]. In other words, *$\mathcal{O}$ contains many 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 [11,12,15,16,17], 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 [15,16,17]. Here we solved them with the default 0.01% accuracy as in the other cases; hence like in [12] 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 36000 seconds (10 hours). The results are displayed in Table 5. In the table, "h" is the number of hydro units and "t" is the number of thermal ones (hence, rows with $h = 0$ refer to "pure thermal" instances). Some instances, marked with "*" in the table, did not terminate before the time limit; for these, besides that root node gap (which, as in Table 4, is different between the two approaches), we then also have to report the gap at termination (column "exit"). Note that while the root node gap is computed using the (same) best known upper bound (for both approaches), the exit gap is that between the lower and upper bounds produced by each approach. Since none of the instances in the groups marked with "*" are solved within the time limit, it makes no sense to report the relative standard deviations of times (them being basically zero).

		PC						AP2R					
		nodes	time			gap		nodes	time			gap	
h	t		avg	dev	root	root	exit		avg	dev	root	root	exit
0	10	299	**8.40**	0.96	0.04	1.460	-	467	45.59	0.48	0.14	1.469	-
0	20	12932	**3123.76**	0.87	0.11	1.229	-	23932	4835.06	0.93	0.41	1.238	-
0	50	27977	*	-	0.50	1.139	0.08	36814	*	-	2.30	1.164	0.09
0	75	22168	*	-	0.79	1.208	0.11	20765	*	-	6.18	1.222	0.12
10	20	4008	**134.52**	0.82	0.12	0.561	-	9665	178.00	0.41	0.74	0.578	-
20	50	24232	**2596.84**	0.85	0.61	0.565	-	178708	10098.49	0.62	1.92	0.573	-
35	75	53520	**7874.43**	0.44	1.26	0.480	-	112675	*	-	5.46	0.489	0.02

Table 5 Results of the (UC) problem

Table 5 shows that AP^{2}R is not competitive with PC on UC. As the theory predicts, the $\underline{\text{AP}^2\text{R}}$ lower bound is (very slightly, but visibly) worse than that of $\underline{\text{PC}}$, albeit the difference is much less than in MV for $k = 10$ (whose gap however can be much larger, cf. Table 4). Unlike in the MV case, here `Cplex` cuts have a significant effect (not shown in the Table for clarity): the final root gap for PC is smaller than for AP^{2}R, the difference being in fact *larger* than that of the original bounds. In other words, in this case the *perspective cuts* added by the SI-MILP formulations act synergistically with the standard `Cplex` cuts, which results in a bound improvement larger than the sum of these each family of cuts would separately produce. However, this is not the main reason why PC is more efficient here: the crucial point is that solving one single QP in $\underline{\text{AP}^2\text{R}}$ takes significantly *longer* than repeatedly solving several LPs in $\underline{\text{PC}}$, as the root time shows. Somewhat surprisingly, in this application approximating the objective function by cutting planes is actually more convenient than having it explicitly represented as a 2-piece linear-quadratic curve. 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 cuts suffices for approximating the nonlinear objective function quite well [15,16]. As a result, in this case solving $\underline{\text{PC}}$ only amounts at a short sequence of LPs, and this turns out to be preferable to solving the single quadratic program $\underline{\text{AP}^2\text{R}}$. Compounded with the worse root node gap, this implies that AP^{2}R requires more nodes to solve one instance, and each node requires more time. All in all, PC is faster for the instances both approaches solve to optimality, solves more instances, and obtains better upper and lower bounds for those that it cannot solve. These results show the limits of the AP^{2}R technique: whenever solving $\underline{\text{AP}^2\text{R}}$ is not preferable to repeatedly solving the linearized version in $\underline{\text{PC}}$, and especially if $\mathcal{O}$ contains many linking constraints, the traditional PC approach is preferable.

5 Conclusions

The paper presents results that considerably extend the significance of the Projected Perspective Reformulation approach of [14]. The main contribution is the "project and lift" procedure giving rise to the Approximated Projected Perspective Reformulation approach. AP^{2}R allows to apply the projection technique to any MINLP with nonlinear (separable) semi-continuous variables, possibly (but not necessarily) at the cost of some bound degradation. Furthermore, AP^{2}R allows direct and easy use of off-the-shelf MINLP solvers rather than requiring the development of ad-hoc codes. Moreover, the significant extension of the class of possible objective functions and the chance to consider feasibility intervals having 0 in their interior (cf. the appendix) allows to apply the P^{2}R technique to a much wider class of problems than previously possible. The computational experiments show that AP^{2}R can be competitive with the best other available PR approaches; this happens e.g., with Network Design and Mean-Variance problems. When the problem is "easy" and with a very strong structure (cf. §4.1) the P^{2}R approach may still be preferable. On the contrary, when the problem contains many constraints linking the u_i variables and a few linear approximations suffice for constructing a good estimate of the nonlinear objective function (cf. §4.4), the PC technique prevails. 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 QP solvers w.r.t. LP 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 MINLP, especially since it is simpler to implement than the other alternatives. This is particularly relevant in view of the fact that the list of applications that have been shown to benefit from PR approaches is steadily growing [6, 9, 10, 22, 23].

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 [24, 29, 31]; cf. §3.2. 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 an 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.

Acknowledgements

The first and third authors gratefully acknowledge the contribution of the Italian Ministry for University and Research under the PRIN 2012 Project 2012JXB3YF "Mixed-Integer Nonlinear Optimization: Approaches and Applications", as well as of the European Union under the 7FP Marie Curie Initial Training Network n. 316647 "MINO: Mixed-Integer Nonlinear Optimization".

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. W.P. Adams and H.D. Sherali. A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986.
3. W.P. Adams and H.D. Sherali. Linearization strategies for a class of zero-one mixed integer programming problems. *Operations Research*, 38(2):217–226, 1990.
4. W.P. Adams and H.D. Sherali. Mixed-integer bilinear programming problems. *Mathematical Programming*, 59(3):279–306, 1993.
5. A. Agnetis, E. Grande, and A. Pacifici. Demand Allocation with Latency Cost Functions. *Mathematical Programming*, 132(1-2):277–294, 2012.
6. 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.
7. 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.

8. S. Ceria and J. Soares. Convex programming for Disjunctive Convex Optimization. *Mathematical Programming*, 86:595–614, 1999.
9. 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.
10. A. Frangioni, L. Galli, and M.G. Scutellà. Delay-Constrained Shortest Paths: Approximation Algorithms and Second-Order Cone Models. *Journal of Optimization Theory and Applications*, 164(3):1051–1077, 2015.
11. A. Frangioni and C. Gentile. Perspective Cuts for 0-1 Mixed Integer Programs. *Mathematical Programming*, 106(2):225–236, 2006.
12. A. Frangioni and C. Gentile. SDP Diagonalizations and Perspective Cuts for a Class of Nonseparable MIQP. *Operations Research Letters*, 35(2):181 – 185, 2007.
13. 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.
14. A. Frangioni, C. Gentile, E. Grande, and A. Pacifici. Projected Perspective Reformulations with Applications in Design Problems. *Operations Research*, 59(5):1225–1232, 2011.
15. 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.
16. 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.
17. A. Frangioni, C. Gentile, and F. Lacalandra. Sequential Lagrangian-MILP Approaches for Unit Commitment Problems. *International Journal of Electrical Power and Energy Systems*, 33:585–593, 2011.
18. A. Frangioni and E. Gorgone. A Library for Continuous Convex Separable Quadratic Knapsack Problems. *European Journal of Operational Research*, 229(1):37–40, 2013.
19. 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.
20. O. Günlük, J. Lee, and R. Weismantel. MINLP Strengthening for Separable Convex Quadratic Transportation-Cost UFL. IBM Research Report RC24213, IBM Research Division, 2007.
21. 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.
22. 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.
23. 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.
24. A. Khajavirad and N.V. Sahinidis. Convex envelopes generated from finitely many compact convex sets. *Mathematical Programming*, 137(1-2):371–408, 2013.
25. 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.
26. S. Leyffer. Experiments with MINLP branching techniques. In *European Workshop on Mixed Integer Nonlinear Programming*, 2010.
27. R.A. Stubbs and S. Mehrotra. A Branch-and-Cut Method for 0-1 Mixed Convex Programming. *Mathematical Programming*, 86:515–532, 1999.
28. M. Tahanan, W. van Ackooij, A. Frangioni, and F. Lacalandra. Large-scale Unit Commitment under uncertainty. *4OR*, 13(2):115–171, 2015.
29. M. Tawarmalani, J.-P.P. Richard, and C. Xiong. Explicit convex and concave envelopes through polyhedral subdivisions. *Mathematical Programming*, 138:531–577, 2013.
30. M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.
31. M. Tawarmalani and N.V. Sahinidis. Semidefinite relaxations of fractional programs via novel convexification techniques. *Journal of Global Optimization*, 20:133–154, 2001.
32. M. Tawarmalani and N.V. Sahinidis. Convex Extensions and Envelopes of Lower Semi-continuous Functions. *Mathematical Programming*, 93:515–532, 2002.
33. 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.

Appendix

In this appendix we show that P^{2}R and AP^{2}R can be extended to the case $p_{min} < 0$, albeit at the cost of slightly larger formulations. We first prove an analogous result to Proposition 1.

Proposition 2 *If $p_{min} < 0$ then $z(p)$ defined in (19) has the form*

$$z(p) = \begin{cases} z_2(p) = f(p) + c & \textit{if } p_{min} \le p \le p^-_{int} \\ z^-_1(p) = \big(\ f(p^-_{int})/p^-_{int} + c/p^-_{int}\ \big)p & \textit{if } p^-_{int} \le p \le 0 \\ z^+_1(p) = \big(\ f(p^+_{int})/p^+_{int} + c/p^+_{int}\ \big)p & \textit{if } 0 \le p \le p^+_{int} \\ z_2(p) = f(p) + c & \textit{if } p^+_{int} \le p \le p_{max} \end{cases} \tag{32}$$

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

$$\max\left\{ \frac{p}{p_{max}} , \frac{p}{p_{min}} \right\} \le u \le 1 \ . \tag{33}$$

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

$$p \ge 0 \Longrightarrow \max\{\ p/p_{max}\ ,\ p/p_{min}\ \} = p/p_{max}$$
$$p \le 0 \Longrightarrow \max\{\ p/p_{max}\ ,\ p/p_{min}\ \} = p/p_{min} \ .$$

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 (33), and there are two sub cases:
   a.1. The derivative is always negative, and therefore $u^*(p) = 1 \Longrightarrow$ (15) holds (i.e., $p_{min} = p_{max} = 0$).
   a.2. The derivative is always positive, and therefore
      for $p < 0$, $u^*(p) = p/p_{min} \Longrightarrow$ (14) holds,
      for $p \ge 0$, $u^*(p) = p/p_{max} \Longrightarrow$ (17) holds.
   All in all, in this case

$$z(p) = \begin{cases} \big(\ f(p_{min})/p_{min} + c/p_{min}\ \big)p & \text{if } p < 0 \\ \big(\ f(p_{max})/p_{max} + c/p_{max}\ \big)p & \text{if } p \ge 0 \end{cases} \ . \tag{34}$$

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

$$p \in [p_{min}, 0] \quad \Longrightarrow \quad p/p_{min} \le \tilde{u}(p) = -pg^- \le 1$$
$$p \in [0, p_{max}] \quad \Longrightarrow \quad p/p_{max} \le \tilde{u}(p) = \ pg^+ \le 1$$

   That is, *exactly two* of the following *four* cases hold:
   b.1. $p \ge 0$ and $\tilde{u}(p) \le p/p_{max} \Longleftrightarrow p_{max} \le 1/g^+ \Longrightarrow u^*(p) = p/p_{max} \Longrightarrow$ (17) holds.
   b.2. $p \ge 0$ and $\tilde{u}(p) \ge p/p_{max} \Longleftrightarrow p_{max} \ge 1/g^+$; two further sub cases arise:
      b.2.1. $(p_{max} \ge)\ p \ge 1/g^+\ (\ge 0) \Longrightarrow \tilde{u}(p) \ge 1 \Longrightarrow u^*(p) = 1 \Longrightarrow$ (15) holds.
      b.2.2. $(0 \le)\ p \le 1/g^+\ (\le p_{max}) \Longrightarrow \tilde{u}(p) \le 1 \Longrightarrow u^*(p) = \tilde{u}(p) \Longrightarrow$ (18) holds.
      This again gives (19).
   b.3. $p \le 0$ and $\tilde{u}(p) \le p/p_{min} \Longleftrightarrow (0 >)\ p_{min} \ge -1/g^- \Longrightarrow u^*(p) = p/p_{min} \Longrightarrow$ (14).
   b.4. $p \le 0$ and $\tilde{u}(p) \ge p/p_{min} \Longleftrightarrow p_{min} \le -1/g^-\ (< 0)$; two further subcases arise:
      b.4.1. $-1/g^- \le p \le 0 \Longleftrightarrow \tilde{u}(p) \le 1 \Longrightarrow u^*(p) = \tilde{u}(p) \Longrightarrow$

$$z(p) = \big(\ -g^- f(-1/g^-) - cg^-\ \big)p \tag{35}$$

      b.4.2. $p_{min} \le p \le -1/g^-\ (< 0) \Longleftrightarrow \tilde{u}(p) \ge 1 \Longrightarrow u^*(p) = 1 \Longrightarrow$ (15).
      All this gives

$$z(p) = \begin{cases} f(p) + c & \text{if } p_{min} \le p \le -1/g^- \\ \big(\ -g^- f(-1/g^-) - cg^-\ \big)p & \text{if } -1/g^- \le p \le 0 \end{cases} \tag{36}$$

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

$$z(p) = \begin{cases} f(p) + c & \text{if } p_{min} \leq p \leq -1/g^- \\ (\, -g^- f(-1/g^-) - cg^- \,)p & \text{if } -1/g^- \leq p \leq 0 \\ (\, g^+ f(1/g^+) + cg^+ \,)p & \text{if } 0 \leq p \leq 1/g^+ \\ f(p) + c & \text{if } 1/g^+ \leq p \leq p_{max} \end{cases} \tag{37}$$

Under condition b.1, the two rightmost pieces are substituted with the linear piece (17) $(f(p_{max})/p_{max} + c/p_{max})p$ for $0 \leq p \leq p_{max}$ and/or, under condition b.3, the two leftmost pieces are substituted with the linear piece (14) $(f(p_{min})/p_{min} + c/p_{min})p$ for $p_{min} \leq p \leq 0$, yielding a 3- or 2-piecewise convex function (piecewise-linear in the latter case as in (34)). □

Example 7 We can extend the rational exponent case of § 2.1. For instance, if $k = 4$ (even case), $h = 3$, $a = 3$, $c = 1$, $p_{min} = -2$, and $p_{max} = 2$, one has

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

We now prove that also (32) can be reformulated as a compact NLP, thus extending the result of Theorem 1 and the AP^{2}R technique to the case $p_{min} < 0$.

Theorem 2 *For $z(p)$ defined in (32) and*

$$\bar{z}(p) = \begin{cases} \min_{u^+, u^-, q^+, q^-} & h(u^+, u^-, q^+, q^-) \\ & -p^+_{int} u^+ \leq q^+ \leq (p_{max} - p^+_{int})u^+ \\ & (p_{min} - p^-_{int})u^- \leq q^- \leq -p^-_{int} u^- \\ & p = p^+_{int} u^+ + q^+ + p^-_{int} u^- + q^- \\ & u^+ + u^- \leq 1 \ , \ u^+ \in [0, 1] \ , \ u^- \in [0, 1] \end{cases} , \tag{38}$$

where

$$h(u^+, u^-, q^+, q^-) = u^+ z^+_1(p^+_{int}) + z_2(q^+ + p^+_{int}) - z^+_1(p^+_{int}) + u^- z^-_1(p^-_{int}) + z_2(q^- + p^-_{int}) - z^-_1(p^-_{int}) \ ,$$

we have $z(p) = \bar{z}(p)$ for all $p \in [p_{min}, p_{max}]$.

Proof As in Theorem 1, the first step is to bring (32) in the form (21). 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 (21) to (32) gives

$$z(p) = \begin{cases} \min_{p_{-2}, p_{-1}, p_1, p_2} & z_2(p_{-2} + p_{min}) + z^-_1(p_{-1} + p^-_{int}) - z^-_1(p^-_{int}) + \\ & \qquad z^+_1(p_1) + z_2(p_2 + p^+_{int}) - z(p^+_{int}) \\ & 0 \leq p_{-2} \leq p^-_{int} - p_{min} \quad , \quad 0 \leq p_{-1} \leq -p^-_{int} \\ & 0 \leq p_1 \leq p^+_{int} \qquad\qquad , \quad 0 \leq p_2 \leq p_{max} - p^+_{int} \\ & p = p_{min} + p_{-2} + p_{-1} + p_1 + p_2 \end{cases} \tag{39}$$

(remember that $z^+_1(0) = 0$), and we want to prove that $z(p)$ given in (39) is equivalent to $\bar{z}(p)$ given in (38) for all $p \in [p_{min}, p_{max}]$. To do that, we start by identifying

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

to recover the objective function and most of the constraints in (38), as some simple but somewhat tedious algebra shows. Then, the general result about (21) 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 (39) 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. 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 1 one easily shows that relaxing the constraints in this way does not change the optimal solution to (39). □

Once again, the choice of (38) 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 PR if $\mathcal{O}$ does not contain constraints linking the u variables, and weaker otherwise. This formulation has twice the number of continuous and binary variables than the ordinary formulation (counting the semi-continuous variables only), but possibly provides (much) stronger bounds.