Improving the Approximated Projected Perspective Reformulation by Dual Information

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa, frangio@di.unipi.it

Fabio Furini
LAMSADE, Université Paris-Dauphine, fabio.furini@dauphine.fr

Claudio Gentile
Istituto di Analisi dei Sistemi ed Informatica “A. Ruberti”, gentile@iasi.cnr.it

Abstract

We propose an improvement of the Approximated Projected Perspective Reformulation (AP²R) for dealing with constraints linking the binary variables. The new approach solves the Perspective Reformulation (PR) once, and then use the corresponding dual information to reformulate the problem prior to applying AP²R, thereby combining the root bound quality of the PR with the reduced relaxation computing time of AP²R. Computational results for the cardinality-constrained Mean-Variance portfolio optimization problem show that the new approach is competitive with state-of-the-art ones.

Keywords: Mixed-Integer Non-Linear Problems, Semi-continuous Variables, Perspective Reformulation, Projection, Lagrangian Relaxation, Portfolio Optimization

1. Introduction

We study solution techniques for convex separable Mixed-Integer Non-Linear Programs (MINLP) with \( n \) semi-continuous variables \( x_i \in \mathbb{R} \) for \( i \in N = \{1, \ldots, n\} \) which either assume the value 0 or lie in the interval \( \mathcal{X}_i = [l_i, \bar{x}_i] \) \((-\infty < l_i < \bar{x}_i < \infty)\). This can be expressed, introducing \( y_i \in \{0, 1\} \) for \( i \in N \), as

\[
\begin{align*}
\text{(P)} & \quad \min h(z) + \sum_{i \in N} f_i(x_i) + c_i y_i \\
& \quad Ax + By + Cz = b \\
& \quad (x, z) \in \mathcal{O} \\
& \quad \underline{x}_i y_i \leq x_i \leq \bar{x}_i y_i, \quad y_i \in [0, 1]^n, \quad x_i \in \mathbb{R}^n, \quad i \in N \\
& \quad y_i \in \mathbb{Z}, \quad i \in N.
\end{align*}
\]

We assume the functions \( f_i \) to be closed convex, one time continuously differentiable and finite in the interval \( \mathcal{X}_i = [l_i, \bar{x}_i] \); w.l.o.g. we also assume \( f_i(0) = 0 \). In (P) we single out the linking constraints (2) that contain all the relationships linking the \( y_i \) variables among them and with the other variables of the problem, except those (4) that “define” the semi-continuous nature of the \( x_i \). The reformulation technique developed in [7] require (2) to be empty; the extension developed in [1] allows to overcome this limitation, but potentially at the cost of a worse bound quality. The aim of this paper is to deal with constraint (2) in a cost-effective way. For our approach to work, (2) must have a compatible structure with that of (1); we initially assume equality constraints, with extensions discussed in [8]. Because our approach hinges on availability of dual information for the continuous relaxation, we assume that the function \( h(\cdot) \) in the “other variables \( z \)” and the “other constraints (3)” are convex, i.e., (P) is a convex MINLP. Actually, in many applications everything but (1) is linear. It will be sometimes expedient to refer to (3–4) as “\((x, y, z) \in \mathcal{P}\)”, and to \( \mathcal{P} \) as the set obtained by \( \mathcal{P} \) relaxing integrality constraints on \( z \) and \( x \), if any.
Often, the most pressing issue in solving (P) is to derive tight lower bounds on its optimal value $\nu(P)$, which is typically done by solving its (convex) continuous relaxation (P) (we denote by $\nu(X)$ and (X), respectively, the optimal value and the continuous relaxation of any problem (X)). However, often $\nu(P) \ll \nu(P)$, making the solution approaches inefficient. The presence of semi-continuous variables has been exploited to propose reformulations (P') of (P) such that $\nu(P) = \nu(P') \geq \nu(P)$ (see [1]). This starts from considering (P) as $h(z) + \sum_{i \in N} f_i(x_i, y_i)$, where $f_i(x_i, y_i) = f_i(x_i) + c_i$ if $y_i = 1$ and $x_i \leq x_i$, $f_i(0, 0) = 0$, and $f_i(x_i, y_i) = \infty$ otherwise. The convex envelope of $f_i(x_i, y_i)$ is known [4] to be $\bar{f}_i(x_i, y_i) = y_i f_i(x_i/y_i) + c_i y_i$—using the perspective function of $f_i$—which yields the Perspective Reformulation of (P)

\[
(P) \quad \min \{ h(z) + \sum_{i \in N} \bar{f}_i(x_i, y_i) : \nu, (x, y) \in \mathcal{P} \}
\]

As $f_i$ is convex, $\bar{f}_i$ is convex for $y_i \geq 0$; since $x_i = 0$ if $y_i = 0$, $\bar{f}_i$ can be extended by continuity assuming $0 f_i(0/0) = 0$. Hence, (PR) is a convex MINLP if (P) is. Its continuous relaxation (PR) —the Perspective Relaxation of (P)—usually has $\nu(PR) \gg \nu(P)$, making (PR) a more convenient formulation [3, 4]. If $f_i$ is SOCP-representable then so is $\bar{f}_i$, hence the PR of a Mixed-Integer Second-Order Cone Program (MI-SOCP) is still a MI-SOCP. Thus, (PR) is not necessarily more complex to solve—and, sometimes, even less so [2]—than (P). Alternatively, one can consider a Semi-Infinite MINLP reformulation of (PR) where Perspective Cuts [4]—linear outer approximations of the epigraph of $\bar{f}_i$—are dynamically added. This is often the best approach [6], in particular for “general” (P) where no other structure is available. It is appropriate to remark that the (PR) approach also applies if the $\nu$ vectors are removed from the formulation, branching has to be done “indirectly”, which rules out using the perspective function [1] to a more general class of functions, and allowing $\bar{z}_i < 0$. However, $\bar{z}_i < 0 < \bar{x}_i$ renders some of the arguments significantly more complex, hence for the sake of simplicity we will only present here the case where $\bar{z}_i \geq 0$; it will be plain to see that the arguments immediately extend to the more general one. The PR idea is to analyze $\bar{f}_i$ as a function of $x_i$ only, i.e., projecting away $y_i$: under appropriate assumptions, and if there are no linking constraints [2], this turns out to be a piecewise-convex functions with a “small” number of pieces, that can be characterized by just looking at the data of (P) (cf. [7]). Hence, (PR) can be reformulated in terms of piecewise-convex objective functions, which makes it easier to solve, especially when $\mathcal{O}$ has some valuable structure (e.g., flow or knapsack) [3]. However, in several applications [2] are indeed present [4, 8, 12, 10]. Furthermore, since the binary variables $y_i$ are removed from the formulation, branching has to be done “indirectly”, which rules out using off-the-shelf solvers. To overcome these two limitations, in [1] the Approximated P2R (AP2R) reformulation has been proposed whereby the $y_i$, after having been eliminated, are re-introduced in the formulation in order to encode the piecewise nature of $\bar{f}_i$. This is possible even if [2] are present, and it has the advantage that (AP2R) is still a MIQP if (P) is. However, $\nu(AP2R) < \nu(PR)$ may, and does, happen when linking constraints [2] are present, whence the “Approximate” moniker. This is still advantageous in some cases, but it may happen that the weaker bounds outweigh the faster solution time, making the approach not competitive with more straightforward implementations of the PR [1].
2. A quick overview of AP²R

We now quickly summarize the analysis in [1], albeit limited to the case \( x_2 \geq 0 \), in order to prepare the ground for the new extension. We focus on the basic problem corresponding to one pair \( (x_i, y_i) \)

\[
(P_i) \quad \min \left\{ f_i(x_i) + c_i y_i : x_i y_i \leq x_i \leq \bar{x}_i y_i, \ y_i \in \{0,1\} \right\}.
\]

The analysis hinges on considering the \((PR)\) of \((P_i)\) rewritten as

\[
(PR_i) \quad \min \left\{ p_i(x_i) = \min \left\{ \tilde{f}_i(x_i, y_i) : x_i y_i \leq x_i \leq \bar{x}_i y_i, \ y_i \in \{0,1\} \right\} : x_i \in [0, \bar{x}_i] \right\},
\]

i.e., first minimizing \( \tilde{f}_i(x_i, y_i) \) with respect to \( y_i \), and then minimizing the resulting function \( p_i(x_i) \) with respect to \( x_i \). The function \( p_i(x_i) \) is convex, and can be characterized by studying the optimal solution \( y_i^*(x_i) \) of the inner problem in \((PR_i)\). Differentiability of \( f_i \) implies that \( y_i^*(x_i) \) is strictly related to the solutions (if any) of the first-order optimality conditions

\[
c_i + f_i(x_i/y_i) - f_i^*(x_i/y_i) = 0.
\]

The approach of [1] requires \((6)\) to only have (at most) one solution, whose dependency on \( y_i \) is “easy”:

**Assumption 1.** \((6)\) has at most one solution for \( x_i \geq 0 \), which has the form \( \tilde{y}_i(x_i) = g_i x_i \), where \( g_i \geq 0 \) is a constant that can be determined by the data of the problem

For instance, for the quadratic case \( f_i(x_i) = a_i x_i^2 + b_i x_i \), the first-order optimality condition \((6)\) is \( c_i - a_i x_i^2/y_i^2 = 0 \), whence \( \tilde{y}_i(x_i) = |x_i| \sqrt{a_i/c_i} \), if \( c_i > 0 \), and there is no solution otherwise. Assumption 1 is satisfied by a surprisingly large set of functions, and a more general version can be stated for the case \( x_i < 0 \). If \((6)\) has a solution then \( y_i^*(x_i) \) can be found by projecting \( \tilde{y}_i(x_i) \) over the interval \([x_i/x_i, \min\{1, x_i/\bar{x}_i\}]\); if no solution exists then \( y_i^*(x_i) \) is in one of the two extremes. In all cases one can then write \( p_i(x_i) = \tilde{f}_i(x_i, y_i^*(x_i)) \). All this gives that there exists some \( x_i \leq \bar{x}_i \leq \bar{x}_i \) such that

\[
p_i(x_i) = \left\{ \begin{array}{ll}
\{ f_i(\bar{x}_i)/\bar{x}_i + c_i/\bar{x}_i \} x_i & \text{if } 0 \leq x_i \leq \bar{x}_i, \\
f_i(x_i) + c_i & \text{if } \bar{x}_i \leq x_i \leq \bar{x}_i \}.
\end{array} \right.
\]

Thus, \( p_i(x_i) \) is piecewise-convex with at most two pieces (although these become four if \( x_i < 0 \)), one of which is linear and the other is the original objective function. The crucial breakpoint \( \bar{x}_i \) can be determined a-priori: in particular, \( \bar{x}_i = 1/g_i \) if \((6)\) has a solution and \( 1/g_i \in \mathcal{X}_i \), and \( \bar{x}_i \in [x_i, \bar{x}_i] \) otherwise [1]. For a numerical illustration, the quadratic case

\[
(P_1) \quad \min \left\{ 2x_i^2 + 8y_1 : y_1 \leq x_1 \leq 10y_1, \ y_1 \in \{0,1\} \right\}
\]

has \( \tilde{y}_1(x_1) = x_1 \sqrt{a_1/c_1} = x_1/2 \), thus \( g_1 = 1/2 \), and therefore \( 1/g_1 = 2 \in \mathcal{X}_1 = [1, 10] \); hence,

\[
p_1(x_1) = \left\{ \begin{array}{ll}
8x_1 & \text{if } 0 \leq x_1 \leq 2, \\
2x_1^2 + 8 & \text{if } 2 \leq x_1 \leq 10 \}.
\end{array} \right.
\]

Writing \((7)\) as the objective function is typically done with the “variable splitting” approach [1], whereby two new variables \( 0 \leq x_i' \leq \bar{x}_i \) and \( 0 \leq x_i'' \leq \bar{x}_i - \bar{x}_i \) are introduced such that \( x_i = x_i' + x_i'' \) (although a different form is sometimes preferable [3]): \( x_i' \) gets the linear cost, while \( x_i'' \) has cost \( f_i(x_i) \). This yields the Projected Perspective Reformulation (P²R), having the same form as (P)—a MIQP if (P) was one—with at most (and, often, less than) twice as many variables. The corresponding (P²R) might be more efficient to solve, especially if (P) has some structure that allows application of specialized approaches [7]. However, removing the \( y_i \) variables from the formulation prevents from using off-the-shelf software to solve (P²R). This is why in [1] it was proposed to “lift back” \((7)\) in the original \((x_i, y_i)\) space by defining the problem

\[
(LP_1(x_i)) \min \left\{ y_i p_i(\bar{x}_i) + f_i(x_i') + c_i - p_i(\bar{x}_i) : (\bar{x}_i - x_i') y_i \leq x_i'' \leq (\bar{x}_i - \bar{x}_i) y_i, \ x_i = \bar{x}_i y_i + x_i'', \ y_i \in [0,1] \right\}.
\]

It can be proven [1] that \( \nu(LP_1(x_i)) = p_i(x_i) \) for each feasible \( x_i \); therefore,

\[
(AP²R_i) \min \left\{ y_i p_i(\bar{x}_i) + f_i(x_i'') + c_i - p_i(\bar{x}_i) : (x_i - \bar{x}_i') y_i \leq x_i'' \leq (\bar{x}_i - \bar{x}_i) y_i, \ x_i = \bar{x}_i y_i + x_i'', \ y_i \in [0,1] \right\}
\]

is a reformulation of \((P_i)\) and \( \nu(AP²R_i) = \nu(PR_i) \), which implies that, typically, \( \nu(AP²R_i) \gg \nu(P_i) \). For illustration, consider \((P_1)\): plugging \((8)\) into \((AP²R_i)\) yields
It can be verified that \( \nu(\text{AP}^2\text{R}_1) \geq \nu(\text{P}_1) \) for any fixed \( x_1 \). For instance, for \( x_1 = 2 \) the optimal solution to \((\text{P}_1)\) is \( y_1 = 1/5 \), yielding \( \nu(\text{P}_1) = 9 + 3/5 \), while the optimal solution to \((\text{AP}^2\text{R}_1)\) is \((y_1, x_1') = (1, 0)\), yielding \( \nu(\text{AP}^2\text{R}_1) = 16 \), i.e., the same estimate as \((\text{PR}_1)\) (for \( x_1 = 2 \)). In fact, 
\[
\nu(\text{PR}_1) = \min \left\{ 2x_1'^2 + 8x_1' + 16y_1 : -y_1 \leq x_1' \leq 8y_1 , \ y_1 = 2y_1 + x_1' , \ y_1 \in \{0, 1\} \right\}.
\]

Hence one can apply the PR to \((\text{P}_1)\) obtained by replacing (9) with \((\text{AP}^2\text{R}_1)\). This hinges on the availability of dual information, and hence mainly concerns the continuous relaxations. The Lagrangian relaxation of \((\text{P}_1)\) is \(\nu(\text{P}_1) = 9 + 3/5\), while the optimal solution to \((\text{AP}^2\text{R}_1)\) is \((y_1, x_1') = (1, 0)\), yielding \(\nu(\text{AP}^2\text{R}_1) = 16\), i.e., the same estimate as \((\text{PR}_1)\) (for \( x_1 = 2 \)). In fact, 
\[
\nu(\text{PR}_1) = \min \left\{ 2x_1'^2 + 8x_1' + 16y_1 : -y_1 \leq x_1' \leq 8y_1 , \ y_1 \in \{0, 1\} \right\} = 16
\]
since \(\min\{8y_1 + 8/y_1 : y_1 \in [1/5, 1]\}\) has optimal solution \(y_1 = 1\).

We will denote by \((\text{AP}^2\text{R})\) the reformulation of \((\text{P})\) where \((\text{AP}^2\text{R}_i)\) is separately applied to each \( x_i \) for \( i \in N \). If there are no linking constraints \((\text{P}_2)\), then obviously \(\nu(\text{AP}^2\text{R}) = \nu(\text{PR})\). Otherwise the reformulation is still possible, but \((\text{PR}_i)\) is a relaxation of the true projection problem, which, besides on \( x_i \), also depends on all the other variables that \( y_i \) is linked with. Hence, \(\nu(\text{AP}^2\text{R}) < \nu(\text{PR})\) can happen, and it does in practice. For illustration consider the problem \((\text{P}_{12})\) obtained by “duplicating” \((\text{P}_1)\) and adding the linking constraint \( y_1 + y_2 = 1 \). The optimal solution of \((\text{P}_{12})\) is \( x_1 = x_2 = 4 \), \( y_1 = y_2 = 1/2 \), yielding \(\nu(\text{P}_{12}) = 72 \ll \nu(\text{P}_{12}) = 136 \), the latter obtained by setting \( x_1 = 8 \), \( y_1 = 1 \), \( x_2 = y_2 = 0 \) (or the symmetric solution). The \((\text{PR})\), obtained by replacing (9) with 
\[
\min 2x_1'/y_1 + 2x_2'/y_2 + 8y_1 + 8y_2
\]
has the same optimal solution as \((\text{P}_{12})\): however, that same solution yields the much stronger (in fact, exact) bound of 136. Instead, for the \((\text{AP}^2\text{R})\)
\[
\text{(AP}^2\text{R}_{12}) \quad \min 2(x_1'')^2 + 2(x_2'')^2 + 8x_1'' + 8x_2'' + 16y_1 + 16y_2
\]
\[
\{y_1 \leq x_1' \leq 8y_1 , \ -y_2 \leq x_2' \leq 8y_2 , \ x_1 = 2y_1 + x_1' , \ x_2 = 2y_2 + x_2' \}
\]
has the same optimal solution as \((\text{P}_{12})\): however, that same solution yields the much stronger (in fact, exact) bound of 136. Instead, for the \((\text{AP}^2\text{R})\)
\[
\nu(\text{AP}^2\text{R}_{12}) = 72 \ll \nu(\text{AP}^2\text{R}_{12}) = 100 \ll \nu(\text{PR}_{12}) = 136.
\]
In the next section we modify the \((\text{AP}^2\text{R})\) to increase its lower bound, avoiding the bound disadvantage with the \((\text{PR})\)—at least at the root node—while retaining the simpler (hence, cheaper) model shape.

3. Improving \(\text{AP}^2\text{R}\) using dual information

The idea is to reformulate \((\text{P})\) to include information about the linking constraints \((\text{P}_2)\) in the objective function \((\text{P}_1)\), so that it can be “processed” by the \(\text{AP}^2\text{R}\). This hinges on the availability of dual information, and hence mainly concerns the continuous relaxations. The Lagrangian relaxation of \((\text{P})\) w.r.t. \((\text{P}_2)\)
\[
(\text{P}_\lambda) \quad \min \{ h(z) + \sum_{i \in N} f_i(x_i) + c_i y_i + \lambda(Ax + By + Cz - b) : (x, y, z) \in \text{P} \}
\]
has an objective function that is still separable in the \( x_i \)
\[
h(z) + \lambda Cz + \sum_{i \in N} \left( f_i(x_i) + \lambda A^i x_i + (c_i + \lambda B^i) y_i \right) - \lambda b .
\]
Hence one can apply the PR to \((\text{P}_\lambda)\), which—since the PR does not change linear functions—yields
\[
(\text{PR}_\lambda) \quad \phi(\lambda) = \min \{ h(z) + \lambda Cz + \sum_{i \in N} \left( y_i f_i(x_i, y_i) + \lambda A^i x_i + (c_i + \lambda B^i) y_i \right) : (x, y, z) \in \text{P} \} - \lambda b .
\]
We will assume that the corresponding Lagrangian dual satisfies \(\max_\lambda \{ \phi(\lambda) \} = \nu(\text{PR})\), and in particular that an optimal dual solution \(\lambda^*\) is available that satisfies \(\phi(\lambda^*) = \nu(\text{PR})\). This requires some conditions that are typically satisfied in most applications (e.g., all the constraints are linear or some appropriate Slater-type condition holds), so that we can expect that \(\lambda^*\) is generated by any approach used to solve \((\text{PR})\).

Note that any inexact solution of the Lagrangian dual could be used as well, as long as it (significantly) improves the bound. Also, (12) clearly satisfies Assumption \((\text{P}_1)\) if \( f_i \) does: (9) for \( f_i(x_i) + \lambda A^i x_i \) only differs from (9) for \( f_i \) for the constant \( \lambda A^i \). Hence, one can form \((\text{AP}^2\text{R}_\lambda)\), the \(\text{AP}^2\text{R}\) of \((\text{PR}_\lambda)\), for any value of \( \lambda \). Taking \( \lambda = \lambda^* \) yields our main result:
Theorem 2. Assume that $f_i$ satisfies Assumption 7 and that $\lambda^*$ is available; for the reformulation of (P) 

\[(P+) \quad \min \left\{ h(z) + \lambda^* Cz + \sum_{i \in N} \left( f_i(x_i) + \lambda^* A^i x_i + (c_i + \lambda^* B^i) y_i \right) : (11), (x, y, z) \in P, (5) \right\} - \lambda^* b \]

denote by (PR+) its PR and by (AP²R+) its AP²R. Then, $\nu(AP²R+) = \nu(PR) = \nu(PR+)$.

Proof. First of all, (P+) is a valid reformulation of (P): since it contains (2), the Lagrangian term $\lambda^* (Ax + By + Cz - b)$ is always null at feasible points, hence both the feasible regions and the objective functions coincide. Also, as discussed for (AP²R+), the objective function of (P+) satisfies Assumption 1 since the $f_i$ do, and therefore (AP²R+) can be formed and $\nu(P) = \nu(P+) = \nu(AP²R+)$ holds. However, note that (11) and (12) are in general different: in fact, the equivalence between the two optimal values only holds when taking into account the constant $-\lambda^* b$. Now, clearly $\nu(PR) = \nu(PR^\lambda) = \nu(PR+)$ due to the choice of $\lambda^*$. Note that (PR+) is a relaxation of (PR+), which in itself would give only $\nu(PR^\lambda) \leq \nu(PR+)$; however the relaxed constraints are precisely (2), those of which $\lambda^*$ are the optimal multipliers, and therefore equality must hold. Now, since (PR+) has no linking constraints (2), $\nu(AP²R^\lambda) = \nu(PR^\lambda)$ for each $\lambda$, and therefore in particular for $\lambda = \lambda^*$. Hence, $\nu(PR) = \nu(PR^\lambda) = \nu(AP²R^\lambda) \leq \nu(AP²R+)$, because, again, (AP²R+) is the relaxation of (AP²R+) w.r.t. (2). But on the other hand $\nu(PR+) \geq \nu(AP²R+)$, as the bound of AP²R is always at most as good as that of the PR, and therefore the thesis is proven.

To illustrate Theorem 2, consider again (P₁₂). The optimal dual multiplier of the linking constraint $y_1 + y_2 = 1$ in (PR₁₂) can be seen to be $\lambda^* = 120$. Hence,

\[(P₁₂+) \quad \min \left\{ 2x_1^2 + 2x_2^2 + 128y_1 + 128y_2 : (10), (11) \right\} - 120 . \]

In its AP²R, $c_i = 128$ gives $g_i = \sqrt{a_{i}/c_{i}} = \sqrt{2/128} = 1/8$, i.e., $\bar{x}_i = 8$ for $i = 1, 2$. Hence,

$p_i(x_i) = y_i p(\bar{x}_i) + f_i(x''_i + \bar{x}_i) + c_i - p_i(\bar{x}_i) = 256y_i + 2(x''_i)^2 + 32x''_i \implies

(AP²R₁₂+) \quad \min \left\{ 2(x''_1)^2 + 2(x''_2)^2 + 32x''_1 + 32x''_2 + 256y_1 + 256y_2 : (11) \right\} - 7y_1 \leq x''_1 \leq 2y_1 , -7y_2 \leq x''_2 \leq 2y_2 , x_1 = 8y_1 + x''_1 , x_2 = 8y_2 + x''_2 . \]

The optimal solution of (AP²R₁₂+) is $x_1 = x_2 = 4, y_1 = y_2 = 1/2, x''_1 = x''_2 = 0$, yielding an objective function value of 256: counting the constant $-\lambda^* b = -120$, this finally gives $\nu(AP²R₁₂+) = 136$: (much) better than $\nu(PR₁₂) = 100$, and in fact precisely equal to $\nu(PR₁₂)$ as predicted.

We end this section by remarking that the assumption that (2) is an equality constraint is not crucial. Inequality linking constraints $Ax + By + Cz \leq b$ can be transformed into equalities by the addition of slack variables: $Ax + By + Cz + s = b, s \geq 0$. Note that this has to be done in (P), so that after the reformulation they will have cost $\lambda^* s$ in the objective function, i.e., they no longer will be slack variables. In principle the constraints could also be nonlinear in $x$ and $z$ (linearity in $y$ can always be assumed without loss of generality), provided that $A(x) = \sum_{i \in N} A^i(x_i)$ with each $A^i(\cdot)$ convex. Assumption 11 holds for $f_i(x_i) + \lambda^* A^i(x_i)$, and $C(\cdot)$ is convex. However, in order for (2) to be convex they necessarily had to be inequalities. This means that the optimal dual multipliers $\lambda^*$ would be non-negative, retaining convexity of $\lambda^* A^i(x_i)$, but the constraints would have to be turned into nonlinear equalities, that can never be convex.

4. Computational results

In this section we report results of computational tests of the proposed approach for the Mean-Variance cardinality-constrained portfolio optimization problem on $n$ risky assets

\[(MV) \quad \min \left\{ x^T Q x : \sum_{i \in N} x_i = 1 , \sum_{i \in N} \mu_i x_i \geq \rho , \sum_{i \in N} y_i \leq k : (1), (5) \right\} , \]

where $\mu$ is the vector of expected unitary returns, $\rho$ is the prescribed total return, $Q$ is the variance-covariance matrix, and $k \leq n$ is the maximum number of purchasable assets. Without the cardinality constraint ($k = n$), (MV) is well suited for AP²R: the bound is the same as that of the PR, and the computation time per node is greatly reduced with respect to the Perspective Cut (P/C) technique. While AP²R is competitive also for $k < n$, it becomes less so as the quality of the bound significantly deteriorates.
Hence, (MV) is a promising application for AP$^2$R+. Since (MV) is a non-separable MIQP, a diagonal matrix $D$ has to be determined such that $Q - D$ is positive semidefinite: the PR technique is applied to $\sum_{i \in N} D_{ii} x_i^2$, leaving the remaining part $x^T (Q - D) x$ untouched. Choosing $D$ is nontrivial: one can use e.g., a “small” SDP as advocated in [3], or a “large” SDP as proposed in [10]. We denote these two by $D_s$ and $D_l$. Although $D_l$ provides a better root node bound, it is not necessarily the best choice throughout the enumeration tree: sometimes a convex combination between the two, denoted by $D_c$, works better [10].

For our tests we used the 90 randomly-generated instances, 30 for each value of $n \in \{200, 300, 400\}$, already employed in [1, 4, 5, 6, 10] to which the interested reader is referred for details. Here we only remark that “+” instances are strongly diagonally dominant, “0” ones are weakly diagonally dominant, and “−” ones are not diagonally dominant; the less diagonally dominant, the harder an instance is. We have set constraints and convex quadratic nonseparable objective function. We have obtained the (PR) diagonal. All the formulations have been given in the natural form, i.e., as a MIQP with linear constraints and convex quadratic nonseparable objective function. We have obtained the (PR) root node bound with the P/C technique, implemented through callbacks, which is typically the best choice when AP$^2$R is not available [3]; hence, for completeness we also report results for the full B&C using P/C. Since we are not interested in comparing the cost/effectiveness of the different diagonal choices, this having been done in [10], we don’t report detailed SDP times beyond saying that the “small” SDP requires on average about 0.2, 0.7, and 1.6 seconds while the “large” SDP requires 9, 21 and 47 seconds, respectively for $n = 200, 300$ and 400 (with little variance for the same $n$). We also don’t report comparisons with either not using PR techniques at all, or using SOCP formulations, because they are well-known to be from dramatically [1, 3] to significantly [6] less effective than the ones employed here.

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 g++ (version 4.8.4) using -O3 as optimization option. We have tested AP$^2$R+ vs. AP$^2$R using Cplex 12.7, single-threaded, with all default parameters (save for one explicitly described below, and only for the tests with the $D_l$ diagonal). All the formulations have been given in the natural form, i.e., as a MIQP with linear constraints and convex quadratic nonseparable objective function. We have obtained the (PR) root node bound with the P/C technique, implemented through callbacks, which is typically the best choice when AP$^2$R is not available [3]; hence, for completeness we also report results for the full B&C using P/C. Since we are not interested in comparing the cost/effectiveness of the different diagonal choices, this having been done in [10], we don’t report detailed SDP times beyond saying that the “small” SDP requires on average about 0.2, 0.7, and 1.6 seconds while the “large” SDP requires 9, 21 and 47 seconds, respectively for $n = 200, 300$ and 400 (with little variance for the same $n$). We also don’t report comparisons with either not using PR techniques at all, or using SOCP formulations, because they are well-known to be from dramatically [1, 3] to significantly [6] less effective than the ones employed here.

The results are reported in Table 1 for the three diagonals $D_s$, $D_c$, and $D_l$, respectively. In the tables we report the (average) total B&C time and root time when using P/C. For AP$^2$R and AP$^2$R+ we report the (average) total number of B&C nodes, total B&C time, root node time and root gap (in percentage). As predicted by Theorem 2 the root node gap of AP$^2$R+ already includes the P/C root time, so it is needed to compute $\lambda^*$ prior to performing the reformulation, and therefore starting the AP$^2$R+ B&C.

<table>
<thead>
<tr>
<th>$n$</th>
<th>P/C</th>
<th>time</th>
<th>nodes</th>
<th>AP$^2$R</th>
<th>time</th>
<th>root</th>
<th>gap</th>
<th>$n$</th>
<th>AP$^2$R+</th>
<th>time</th>
<th>root</th>
<th>gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>200+</td>
<td>1.17</td>
<td>0.23</td>
<td>205</td>
<td>0.23</td>
<td>0.12</td>
<td>0.77</td>
<td>118</td>
<td>0.36</td>
<td>0.13</td>
<td>0.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>1256.99</td>
<td>0.18</td>
<td>23354</td>
<td>0.12</td>
<td>3.14</td>
<td>9325</td>
<td>0.13</td>
<td>2.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200+</td>
<td>21941.53</td>
<td>0.20</td>
<td>170682</td>
<td>0.13</td>
<td>4.75</td>
<td>40193</td>
<td>0.12</td>
<td>4.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300+</td>
<td>68043.87</td>
<td>0.49</td>
<td>1288</td>
<td>0.35</td>
<td>1.08</td>
<td>320</td>
<td>0.18</td>
<td>0.49</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>1256.99</td>
<td>0.18</td>
<td>23354</td>
<td>0.12</td>
<td>3.14</td>
<td>9325</td>
<td>0.13</td>
<td>2.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300+</td>
<td>21941.53</td>
<td>0.20</td>
<td>170682</td>
<td>0.13</td>
<td>4.75</td>
<td>40193</td>
<td>0.12</td>
<td>4.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400+</td>
<td>37836.53</td>
<td>1.10</td>
<td>334730</td>
<td>0.12</td>
<td>3.80</td>
<td>49841</td>
<td>0.17</td>
<td>2.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4000</td>
<td>1256.99</td>
<td>0.18</td>
<td>23354</td>
<td>0.12</td>
<td>3.14</td>
<td>9325</td>
<td>0.13</td>
<td>2.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400+</td>
<td>21941.53</td>
<td>0.20</td>
<td>170682</td>
<td>0.13</td>
<td>4.75</td>
<td>40193</td>
<td>0.12</td>
<td>4.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Results with diagonal $D_s$.

Tables 1 and 2 show that AP$^2$R+ is highly competitive with AP$^2$R, and a fortiori with P/C, reducing total time by up to an order of magnitude. While the exact ratio depends on $n$, the type of instance and...
Table 2: Results with diagonal $D_c$

<table>
<thead>
<tr>
<th></th>
<th>P/C</th>
<th>AP$^2$R</th>
<th>AP$^2$R+</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>nodes</td>
<td>time</td>
</tr>
<tr>
<td>200$^+$</td>
<td>0.93</td>
<td>0.20</td>
<td>129</td>
</tr>
<tr>
<td>200$^0$</td>
<td>47.54</td>
<td>0.20</td>
<td>6318</td>
</tr>
<tr>
<td>200$^-$</td>
<td>622.25</td>
<td>0.22</td>
<td>45267</td>
</tr>
<tr>
<td>300$^+$</td>
<td>2.36</td>
<td>0.51</td>
<td>558</td>
</tr>
<tr>
<td>300$^0$</td>
<td>71.40</td>
<td>0.52</td>
<td>20698</td>
</tr>
<tr>
<td>300$^-$</td>
<td>1649.26</td>
<td>0.54</td>
<td>122138</td>
</tr>
<tr>
<td>400$^+$</td>
<td>3.17</td>
<td>1.12</td>
<td>524</td>
</tr>
<tr>
<td>400$^0$</td>
<td>613.01</td>
<td>1.13</td>
<td>94154</td>
</tr>
<tr>
<td>400$^-$</td>
<td>8095.26</td>
<td>1.15</td>
<td>345940</td>
</tr>
</tbody>
</table>

Table 3: Results with diagonal $D_l$, default and with option CPX_PARAM_SUBALG = 5

<table>
<thead>
<tr>
<th></th>
<th>P/C</th>
<th>AP$^2$R</th>
<th>AP$^2$R+</th>
<th>AP$^2$R++</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>time</td>
<td>nodes</td>
<td>time</td>
<td>root</td>
</tr>
<tr>
<td>200$^+$</td>
<td>1.41</td>
<td>0.25</td>
<td>131</td>
<td>0.78</td>
</tr>
<tr>
<td>200$^0$</td>
<td>4.46</td>
<td>0.36</td>
<td>887</td>
<td>2.56</td>
</tr>
<tr>
<td>200$^-$</td>
<td>14.99</td>
<td>0.40</td>
<td>5893</td>
<td>9.81</td>
</tr>
<tr>
<td>300$^+$</td>
<td>3.18</td>
<td>0.78</td>
<td>784</td>
<td>4.82</td>
</tr>
<tr>
<td>300$^0$</td>
<td>8.81</td>
<td>0.96</td>
<td>2211</td>
<td>10.12</td>
</tr>
<tr>
<td>300$^-$</td>
<td>26.86</td>
<td>0.99</td>
<td>13032</td>
<td>39.86</td>
</tr>
<tr>
<td>400$^+$</td>
<td>5.33</td>
<td>1.59</td>
<td>906</td>
<td>10.35</td>
</tr>
<tr>
<td>400$^0$</td>
<td>50.60</td>
<td>1.91</td>
<td>9195</td>
<td>63.93</td>
</tr>
<tr>
<td>400$^-$</td>
<td>52.91</td>
<td>2.15</td>
<td>18096</td>
<td>132.09</td>
</tr>
</tbody>
</table>

5. Conclusions

The main advantage of the proposed AP$^2$R+ technique is its simplicity: just solving $(PR)$—possibly even approximately with a dual approach—produces the dual solution $\lambda^*$ which can be used to first
construct \((P^+)\) and then its \((AP^2R)\). Yet, this improves many-fold the performances over plain \(AP^2R\), and even more so over \(P/C\). Notably, \(AP^2R^+\) is quite general and applies to a much larger class than MIQP. It may be worth contrasting 175843 seconds (\(P/C\) in Table 1) with 56 seconds (\(AP^2R^{++}\) in Table 3) for 400\(^-\) instances: this is well over over three orders of magnitude difference for solving the same instances with the same underlying solver, and the gap with the standard MIQP formulation would be even more humungous. This nicely illustrates the power of reformulation techniques like \(AP^2R^+\).

Acknowledgements

The first and third authors acknowledge the contribution of the Italian Ministry for University and Research under the PRIN 2012 Project 2012JXB3YF “Mixed-Integer Nonlinear Optimization: Approaches and Applications”. All the authors acknowledge networking support by the COST Action TD1207. We are grateful to the anonymous Referee and the Associate Editor of the Journal for constructive comments that helped us in significantly improve the manuscript.

References