ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

A. Frangioni, C. Gentile, E. Grande, A. Pacifici

PROJECTED PERSPECTIVE REFORMULATIONS WITH APPLICATIONS IN DESIGN PROBLEMS

R. 09-09, 2009

Antonio Frangioni – Dipartimento di Informatica, Polo Universitario della Spezia, Via dei Colli 90, 19121 La Spezia – Italy. Email: frangio@di.unipi.it.

Claudio Gentile – Istituto di Analisi dei Sistemi ed Informatica, CNR, Viale Manzoni 30 - 00185 Roma, Italy. Email: gentile@iasi.cnr.it Ph. +39 06 77161, Fax +39 06 7716461..

Enrico Grande – Dipartimento di Ingegneria dell'Impresa, Università degli Studi di Roma "Tor Vergata", Via del Politecnico 1, 00133 Roma, Italy. Email: grande@disp.uniroma2.it.

Andrea Pacifici – Dipartimento di Ingegneria dell'Impresa, Università degli Studi di Roma "Tor Vergata", Via del Politecnico 1, 00133 Roma, Italy. Email: pacifici@disp.uniroma2.it.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR
viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

The *Perspective Relaxation* (PR) is a general approach for constructing tight approximations to Mixed-Integer NonLinear Problems with semicontinuous variables. The PR of a MINLP can be formulated either as a Mixed-Integer Second-Order Cone Program (provided that the original objective function is SOCP-representable), or as a Semi-Infinite MINLP. In this paper, we show that under some further assumptions (rather restrictive, but satisfied in several practical applications), the PR of Mixed-Integer Quadratic Program can also be reformulated as a piecewise linear-quadratic problem, ultimately yielding a QP relaxation of roughly the same size of the standard continuous relaxation. Furthermore, if the original problem has some exploitable structure, then this structure is typically preserved in the reformulation, thus allowing to construct specialized approaches for solving the PR. We report on implementing these ideas on two MIQPs with appropriate structure: a sensor placement problem and a Quadratic-cost (single-commodity) network design problem.

Key words: Mixed-Integer NonLinear Problems, Semicontinuous Variables, Perspective Relaxation, Sensor Placement Problem, Network Design Problem

1. Introduction

Semi-continuous variables are very often found in models of real-world problems such as distribution and production planning problems [20, 11, 14], financial trading and planning problems [12], and many others [1, 3, 15, 16]. These are variables which are constrained to either assume the value 0, or to lie in some given *convex* compact set $\mathcal{P}$; in our applications $\mathcal{P}$ will always be a polyhedron. Often $0 \notin \mathcal{P}$; this is e.g. the case when the variable represents the output of a production process that has a "nonzero minimum producible amount", but that can be switched off altogether. Alternatively, 0 may belong to $\mathcal{P}$, but one may incur in a fixed cost c to "activate" the process (produce a nonzero amount).

We will consider Mixed-Integer NonLinear Programs (MINLP) with n semi-continuous variables $x_i \in \mathbb{R}^{m_i}$ for each $i \in N = \{1, \dots, n\}$. Assuming that each $\mathcal{P}_i = \{x_i : A_i x_i \leq b_i\}$ has the property that $\{x_i : A_i x_i \leq 0\} = \{0\}$, each x_i can be modeled by using an associated binary variable y_i, leading to problems of the form

$$\min \qquad g(z) + \textstyle\sum_{i \in N} f_i(x_i) + c_i y_i \tag{1}$$

$$A_i x_i \leq b_i y_i \qquad\qquad i \in N \tag{2}$$

$$(x, y, z) \in \mathcal{O} \quad , \quad y \in \{0,1\}^n \quad , \quad x \in \mathbb{R}^m \quad , \quad z \in \mathbb{R}^q \tag{3}$$

where all f_i and g are closed convex functions, z is the vector of all the "other" variables, and $\mathcal{O}$ is any subset of $\mathbb{R}^{m+n+q}$ (with $m = \sum_{i \in N} m_i$), representing all the "other" constraints of the problem.

It is known that the convex hull of the (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 [18]; this leads to defining the *Perspective Reformulation* of (1)—(3) [5, 11]

$$\min \left\{ g(z) + \textstyle\sum_{i \in N} y_i f_i(x_i / y_i) + c_i y_i \ : \ (2) \ , \ (3) \right\} \tag{4}$$

whose continuous relaxation is significantly stronger than that of (1)—(3), and that therefore is a more convenient starting point to develop exact and approximate solution algorithms [11, 12, 3, 14, 16]. We remark that $y_i f_i(x_i / y_i)$ for $y_i > 0$ is called the *perspective function* of $f_i(x_i)$ (a well-known tool in convex analysis), whence the name; while the objective function in (4) is formally undefined when some $y_i = 0$, one can extend it by continuity to allow for null values (we assume that this is, in fact, done throughout the paper).

However, an issue with (4) is the high nonlinearity in the objective function due to the added fractional term. Two alternative reformulations of (4) have been proposed; one as a Mixed-Integer Second-Order Cone Program [19, 3, 16] (provided that the original objective function is SOCP-representable), and the other as a Semi-Infinite MILP [11]. In several cases, the latter outperforms the former in the context of exact or approximate enumerative solution approaches [13], basically due to the much higher reoptimization efficiency of active-set (simplex-like) methods for Linear and Quadratic Programs w.r.t. the available Interior Point methods for Conic Programs.

However, both reformulations of (4) require the solution of substantially more complex continuous relaxations than the original formulation of (1)—(3). In this paper, we show that under some further assumptions (rather restrictive, but satisfied in several practical applications), the PR of a Mixed-Integer *Quadratic* Program can also be reformulated as a piecewise linear-quadratic problem, ultimately yielding a QP relaxation of roughly the same size of the standard continuous relaxation; this is discussed in Section 2. Furthermore, if the original problem has some exploitable structure, then this structure is typically preserved in the reformulation, thus allowing to construct specialized approaches for solving the PR. We apply this approach on two MIQPs with appropriate structure: a sensor placement problem (Section 3) and a Quadratic-cost (single-commodity) network design problem (Section 4), reporting numerical experiments comparing state-of-the-art, off-the-shelf MIQP solvers with the new specialized solution approach (Section 5).

2. A piecewise description of the convex envelope

Here we refine the analysis of the properties of the Perspective Reformulation under three further assumptions on the data of the original problem (1)—(3):

A1) each x_i is a *single variable* (i.e., $m_i = 1$), therefore each $\mathcal{P}_i$ is a bounded real interval $[l_i, u_i]$ with $0 \leq l_i < u_i$;

A2) the variables y_i *only* appear each in the corresponding constraint (2), i.e., the "other" constraints $\mathcal{O}$ do not concern the y_i;

A3) all functions are *quadratic*, i.e., $f_i(x_i) = a_i x_i^2 + b_i x_i$ (and since they are convex, $a_i > 0$).

While these assumptions are indeed restricting, they are in fact satisfied by most of the applications of the PR reported so far [11, 12, 3, 15, 16]. Since in this paragraph we will only work with *one* block at a time, to simplify the notation in the following we will drop the index "i". We will therefore consider the (fragment of) Mixed-Integer Quadratic Program (MIQP)

$$\min \left\{ \; ax^2 + bx + cy \; : \; ly \leq x \leq uy \; , \; y \in \{0, 1\} \; \right\} \tag{5}$$

and its Perspective Relaxation

$$\min \left\{ \; f(x, y) = (1/y)ax^2 + bx + cy \; : \; ly \leq x \leq uy \; , \; y \in [0, 1] \; \right\} \quad . \tag{6}$$

The basic idea behind the approach is to recast (6) as the minimization over $x \in [0, u]$ of the following function:

$$z(x) = \min_y f(x, y) = bx + \min \left\{ \; (1/y)ax^2 + cy \; : \; ly \leq x \leq uy \; , \; y \in [0, 1] \; \right\} \quad . \tag{7}$$

It is well-known that $z(x)$ (partial minimization of a convex function) is convex; furthermore, due to the specific structure of the problem $z(x)$ can be algebraically characterized. In particular, due to convexity of $f(x, y)$, the optimal solution $y^*(x)$ of the inner optimization problem in (7) is easily obtained by the solution $\tilde{y}$ (if any) of the first-order optimality conditions of the unconstrained version of the problem

$$\frac{\partial f(x, y)}{\partial y} = c - \frac{1}{y^2} ax^2 = 0 \quad . \tag{8}$$

In fact, if $\tilde{y}$ is feasible for the problem, then it is optimal ($y^*(x) = \tilde{y}$); otherwise, $y^*(x)$ is the projection of $\tilde{y}$ over the feasible region, i.e., the extreme of the interval nearer to $\tilde{y}$ (this is where assumption A1 is used). Thus, by developing the different cases, one can construct an explicit algebraic description of $z(x) = f(x, y^*(x))$. To simplify the presentation, in the following we will treat l as if it were a *positive* number; e.g., we will assume that x/l is always a well-defined quantity. It can be easily verified that all the obtained formulae plainly extend to the case $l = 0$.

2.1. The piecewise description of $z(x)$

We start by rewriting the constraints in (7) as

$$(0 \leq) \; \frac{x}{u} \leq y \leq \min \left\{ \; \frac{x}{l} \; , \; 1 \; \right\} \tag{9}$$

(since $u \geq x \geq l \geq 0 \Rightarrow x/u \geq 0$); if $l = 0$ the constraint $ly \leq x$ is redundant, and one can imagine $x/l = +\infty$ so that the quantity "never gets in the way of y". We must now proceed by cases:

1) If $c \leq 0$, then (8) has no solution for $y > 0$: the derivative is always negative. Of course, $y = 0$ is not a solution, either. Thus, there is no global minima of the unconstrained problem, and therefore $y^*(x) = \min\{ \; x/l \; , \; 1 \; \}$. This gives two subcases:

   1.1) $x/l \leq 1 \Leftrightarrow x \leq l \Leftrightarrow y^*(x) = x/l \Rightarrow$

$$z(x) = \left(\; b + al + c/l \; \right) x \tag{10}$$

   1.2) $x/l \geq 1 \Leftrightarrow x \geq l \Leftrightarrow y^*(x) = 1 \Rightarrow$

$$z(x) = ax^2 + bx + c \quad . \tag{11}$$

In other words, $z(x)$ is the piecewise linear-quadratic function

$$z(x) = \begin{cases} (\, b + al + c/l \,)x & \text{if } 0 \leq x \leq l \\ ax^2 + bx + c & \text{if } l \leq x \leq u \end{cases} \tag{12}$$

Note that $z(x)$ is continuous in the (potential) breakpoint $x = l$; also, we have $z'_-(l) = b + al + c/l \leq 2al + b = z'_+(l)$ (as $a > 0$, $l > 0$ and $c \leq 0$; the case $l = 0$ is obvious), confirming that $z(x)$ is convex, as expected.

2) Instead, if $c > 0$ the only solution to (8) is

$$\tilde{y} = x\sqrt{a/c} \tag{13}$$

(note that we have used $x \geq (l \geq)\ 0$, $c > 0, a > 0$). Actually, this gives $\tilde{y} = 0$ for $x = 0$, which leaves (8) ill-defined; however, this is the only solution in (9) if $l > 0$, while if $l = 0$ one could choose any $y \in [0, 1]$, but this again gives $\tilde{y} = y^*(x) = 0$ as $c > 0$. In general, three cases can arise:

2.1) $\tilde{y} \leq x/u \Leftrightarrow u \leq \sqrt{c/a} \Leftrightarrow y^*(x) = x/u \Rightarrow$

$$z(x) = (\, b + au + c/u \,)x \tag{14}$$

2.2) $x/l \geq \tilde{y} \geq x/u \Leftrightarrow u \geq \sqrt{c/a} \geq l$; two further subcases arise:

2.2.1) $(u \geq)\ x \geq \sqrt{c/a}\ (\geq l)$, which implies both $\tilde{y} \geq 1$ and $x/l \geq 1$, so that $y^*(x) = 1$ and therefore (11) holds;

2.2.2) $l \leq x \leq \sqrt{c/a}\ (\leq u)$, which gives $\tilde{y} \leq 1$. Now, if $l \leq x$ then $x/l \geq 1$, and therefore $y^*(x) = \tilde{y}$. However, because $l \leq \sqrt{c/a}$ we always have $x/l \leq x\sqrt{a/c} = \tilde{y}$, thus even when $0 \leq x \leq l$ we have $y^*(x) = \tilde{y}$, which finally implies

$$z(x) = (\, b + 2\sqrt{ac} \,)x \quad . \tag{15}$$

Thus, $z(x)$ is the piecewise linear-quadratic function

$$z(x) = \begin{cases} (\, b + 2\sqrt{ac} \,)x & \text{if } 0 \leq x \leq \sqrt{c/a} \\ ax^2 + bx + c & \text{if } \sqrt{c/a} \leq x \leq u \end{cases} \tag{16}$$

Note that (16) is continuous and differentiable even at the (potential) breakpoint $x = \sqrt{c/a}$, and therefore convex (as expected); the derivative is constant—hence nondecreasing—in the linear part and increasing in the quadratic one.

2.3) $\tilde{y} \geq x/l \Leftrightarrow (u \geq)\ l \geq \sqrt{c/a} \Leftrightarrow y^*(x) = \min\{x/l, 1\} \Rightarrow$ (12) (cf. 1)).

In all the cases, $z(x)$ is a convex piecewise-quadratic function with at most 2 pieces; except in case (12) it is also differentiable, and even (12) is differentiable if $l = 0$ (in which case the function actually has only one piece).

2.2. A convex reformulation

Hence, we are confronted with the problem of minimizing a convex function with the generic k-piecewise form

$$z(x) = z_h(x) \text{ if } \alpha_h \leq x \leq \alpha_{h+1} \quad h = 1, \ldots, k$$

where each $z_h(x)$ is, obviously, convex. For the convexity of the function z, one has

$$z(x) = \min \begin{cases} z_1(\chi_1 + \alpha_1) + \sum_{h=2}^{k} z_h(\chi_h + \alpha_h) - z_h(\alpha_h) \\ \chi_h \in [0, \alpha_{h+1} - \alpha_h] \qquad h = 1, \ldots, k \\ \alpha_1 + \sum_{h=1}^{k} \chi_h = x \end{cases} \tag{17}$$

for all $x \in [\alpha_1, \alpha_{k+1}]$. This equivalence is well-known (being related to the fact that the *infimal convolution* of convex functions is convex), but we briefly sketch a proof for illustratory purposes assuming for simplicity that all the z_h are continuously differentiable. It is easy to check that for each value of $x \in [\alpha_1, \alpha_{k+1}]$, the following "canonical representation" of x

$$\chi_h = \max\{\ \alpha_h\ , \min\{\ \alpha_{h+1}\ ,\ x\ \}\ \} - \alpha_h \qquad\qquad h = 1, \ldots, k$$

is equivalent to x in the sense that

$$z(x) = z_1(\chi_1 + \alpha_1) + \textstyle\sum_{h=2}^{k} z_h(\chi_h + \alpha_h) - z_h(\alpha_h)$$

(by convexity $\Rightarrow$ continuity of z, $z_h(\alpha_{h+1}) = z_{h+1}(\alpha_{h+1})$). Thus, (17) provides a *lower bound* on $z(x)$. However, it is easy to see that there exists an optimal solution to (17) representing the "canonical form" of x, that is, for which there exists an index $q \in \{1, \ldots, k\}$ such that $\chi_h = \alpha_{h+1} - \alpha_h$ for $h < q$, $\chi_q \in [0, \alpha_{q+1} - \alpha_q]$, and $\chi_h = 0$ for $h > q$. In fact, take an optimal solution $[\chi^*_h]$ of (17) and assume that there exist two indices $1 \leq q < j \leq k$ such that $\chi^*_q < \alpha_{q+1} - \alpha_q$ and $\chi^*_j > 0$. Because z is overall convex, its derivative must be nondecreasing, and therefore

$$z'_q(\chi^*_q + \alpha_q) \leq z'_j(\chi^*_j + \alpha_j)$$

(as $q < j$, $\alpha_j \geq \alpha_{q+1}$). Hence, for some small $\epsilon > 0$ the feasible solution to (17) obtained from $[\chi^*_h]$ by increasing χ^*_q of ϵ and decreasing χ^*_j of ϵ must be not worse than $[\chi^*_h]$; since the latter is optimal, the former must be optimal too. Hence, by increasing χ_q and decreasing χ_j we can construct an optimal solution where either the q-th interval is "full" ($\chi_q = \alpha_{q+1} - \alpha_q$) or the j-th interval is "empty" ($\chi_j = 0$); repeating this we show that an optimal solution to (17) is the canonical form of x.

The interest of this procedure is that if we have a minimization problem where $z(x)$ is a part of the objective function and x is constrained to lie in $[\alpha_1, \alpha_{k+1}]$, we can obtain an equivalent problem by:

- replacing $z(x)$ in the objective function with $z_1(\chi_1 + \alpha_1) + \sum_{h=2}^{k} z_h(\chi_h + \alpha_h) - z_h(\alpha_h)$;
- replacing x everywhere in the constraints with $\alpha_1 + \sum_{h=1}^{k} \chi_h$, where each χ_h is constrained to lie in $[0, \alpha_{h+1} - \alpha_h]$.

Because these changes are quite simple, we can transform a problem with a "complex" convex objective function in a problem with more variables but "simpler" convex objective functions without interfering too much with the structure of the constraints. This may allow us to use specialized solution algorithms that exploit the structure of the constraints without the need to explicitly take into account the piecewise nature of the original objective functions. Two examples of application of this procedure are shown below.

3. A sensor placement problem

Consider the problem of optimally placing a set $N = \{1, \ldots, n\}$ of sensors to cover a given area, where deploying one sensor has a fixed cost plus a cost that is quadratic in the radius of the surface covered [1]. The problem, which is shown to be $\mathcal{NP}$-hard in [2], can be written as

$$\begin{array}{ll}
\min \sum_{i \in N} c_i y_i + \sum_{i \in N} a_i x_i^2 & \\
\quad 0 \leq x_i \leq y_i & i \in N \\
\quad \sum_{i \in N} x_i = 1 & \\
\quad y_i \in \{0, 1\} & i \in N
\end{array}$$

Since we can assume $c_i > 0$ (for otherwise y_i can surely be fixed to 1), in the continuous relaxation of this problem the "design" variables y_i can be "projected" onto the x_i; that is, since at optimality it surely is $y_i = x_i$, the y_i variables can be eliminated and their linear cost term is shifted onto the x_i. Such a problem can be solved in $O(n \log n)$; however, the bound provided by the continuous relaxation can be weak, leading to a large number of nodes in the enumeration tree and therefore to a large solution time.

We want to improve the bound by using the convex envelope of the single blocks of the objective function. As outlined in the previous sections, we can compute this bound by means of a *single* minimization involving the piecewise-linear-quadratic functions developed in Section 2; that is, according to the reformulation technique of Subsection 2.2, we rewrite the problem in the form

$$\begin{array}{ll} \min & \sum_{j=1}^{m} b_j\chi_j + \sum_{i=1}^{m} d_j\chi_j^2 \\ & \sum_{j=1}^{m} \chi_j = 1 \\ & \chi_j \in [0,\alpha_j] \quad j = 1,\dots,m \end{array} \tag{18}$$

where $m \leq 2n$ and the coefficients b_j and d_j are as follows:

- if $\sqrt{c_i/a_i} \geq 1$ then only one new variable χ_j is generated with coefficients $b_j = a_iu_i + c_i/u_i$, $d_j = 0$, and $\alpha_j = u_i$;
- if $\sqrt{c_i/a_i} < 1$ then two new variables χ_{j_1} and χ_{j_2} are generated such that $x_i = \chi_{j_1} + \chi_{j_2}$ with $b_{j_1} = 2\sqrt{a_ic_i}$, $d_{j_1} = 0$, $\alpha_{j_1} = \sqrt{c_i/a_i}$ for the first variable and $b_{j_2} = 2\sqrt{a_ic_i}$, $d_{j_2} = a_i$, $\alpha_{j_2} = 1 - \sqrt{c_i/a_i}$ for the second variable.

This problem can be easily solved in $O(m\log m) = O(n\log n)$ as follows. Consider its Lagrangian relaxation w.r.t. the "linking" constraint $\sum_{j=1}^{m}\chi_j = 1$ with Lagrangian multiplier μ:

$$\begin{array}{rl} \phi(\mu) = \mu + \min & \sum_{j=1}^{m}(b_j - \mu)\chi_j + \sum_{j=1}^{m} d_j\chi_j^2 \\ & \chi_j \in [0,\alpha_j] \quad j = 1,\dots,m \end{array}$$

Computation of $\phi(\mu)$ decomposes into the m independent quadratic problems

$$\min \{\ (b_j - \mu)\chi_i + d_j\chi_j^2\ :\ \chi_j \in [0,\alpha_j]\ \} \tag{19}$$

that can solved in $O(1)$. By convexity, the Lagrangian dual problem, $\max_{\mu\in\mathbb{R}}\ \phi(\mu)$, is equivalent to (18). To solve the dual efficiently, consider the solution to (19) parametrized in μ. Temporarily assuming, for simplicity, that $d_j > 0$ for all $j = 1,\dots,m$, the unconstrained minimum is

$$\tilde{\chi}_j(\mu) = \frac{\mu - b_j}{2d_j}$$

and therefore the optimal solution $\chi_j^*(\mu)$ of (19) is:

- 0 if $(\mu - b_j)/2d_j \leq 0 \Rightarrow \mu \leq b_j$;
- α_j if $(\mu - b_j)/2d_j \geq \alpha_j \Rightarrow \mu \geq 2\alpha_jd_j + b_j$;
- $(\mu - b_j)/2d_j$ if $0 \leq (\mu - b_j)/2d_j \leq \alpha_j \Rightarrow \mu \in [b_j, 2\alpha_jd_j + b_j]$ (note that $\alpha_jd_j > 0$).

It is easy to check that $\chi_j^*(\mu)$ is nondecreasing in μ; this is expected, since

$$\phi'(\mu) = 1 - \sum_{j=1}^{m}\chi_j^*(\mu)$$

must be nonincreasing since ϕ is concave. Each variable $\chi_j = \chi_j^*(\mu)$ gives a fixed contribution to the derivative outside the given interval $[b_j, 2\alpha_jd_j + b_j]$, while the contribution is linear inside the interval. It is then easy to find the unique value of μ such that $\phi'(\mu) = 0$ (the dual must have an optimal solution since the primal is surely nonempty and bounded). First, all the $2m$ extremes of the m intervals b_j and $2\alpha_jd_j + b_j$ are all inserted in a *unique* list that is then ordered in nondecreasing sense; let us denote by $\bar{\mu}_1$, $\bar{\mu}_2$, ..., $\bar{\mu}_{2m}$ the elements in the list after the ordering. For sufficiently small values of μ—smaller than $\bar{\mu}_1$—$\chi_j^*(\mu) = 0$ for all j and therefore $\phi'(\mu) = 1 \Rightarrow \phi$ is increasing. Then, μ is initialized to $\bar{\mu}_1$, that must be the left endpoint b_h for some variable h, the current value β of $\phi'(\mu)$ is initialized to 1, and the rate of change γ of $\phi'(\mu)$ is initialized to $1/d_h$. Then, the next element $\bar{\mu}$ in the list is looked at: it corresponds either to the left endpoint or to the right endpoint of the interval corresponding to some variable k. If $\beta - \gamma(\bar{\mu} - \mu) \leq 0$, then $\mu^* = \mu + \beta/\gamma$ is an optimal solution to the Lagrangian dual ($\phi'(\mu^*) = 0$) and an optimal solution to (18) can be derived from μ^* in $O(m)$. Otherwise, μ is updated to $\bar{\mu}$ (that surely has a better ϕ-value) and $\beta = \phi'(\bar{\mu})$ is updated by subtracting it $\gamma(\bar{\mu} - \mu)$. Then, if $\bar{\mu}$ is the left endpoint of variable k, this also becomes "active", and therefore $1/(2d_k)$ is added to γ; instead, if $\bar{\mu}$ is the right endpoint, then k becomes "inactive" and $1/(2d_k)$ is subtracted to γ. By iterating this procedure, μ^* is identified within $O(m)$ steps, each one costing $O(1)$; therefore, (18) is solved in $O(m \log m) = O(n \log n)$ overall, owing to the cost of ordering the list.

This sketch of solution procedure has to be slightly complicated to take into account all possibilities. First, note that γ can become zero if there are no "active" variables; in this case μ is immediately advanced to the next element in the list, since nothing happens to $\phi'(\mu)$ in the interval. Furthermore, if $d_i = 0$ the optimal solution to (19) is not unique; indeed, we have

$$\chi_j^*(\mu) \in \begin{cases} \{0\} & \mu < b_j \\ [0, \alpha_j] & \mu = b_j \\ \{\alpha_j\} & \mu > b_j \end{cases} .$$

Thus, in this case the interval where $\chi_j^*(\mu)$ varies is reduced to a single point, and any $\chi_j \in [0, \alpha_j]$ is an optimal solution there. It is not difficult to extend the above procedure to handle this case, too.

4. Quadratic-cost network design

A directed graph $G = (N, A)$ is given; for each node $i \in N$ a deficit $d_i \in \mathbb{R}$ is given indicating the amount of flow that the node demands (negative deficits indicate source nodes). Each arc $(i, j) \in A$ can be used up to a given maximum capacity u_{ij} paying a fixed cost c_{ij}. Otherwise, no cost is due if (i, j) is not installed but flow cannot pass through the arc. Additionally, if x_{ij} units of flow are sent through an installed arc (i, j), a quadratic flow cost $b_{ij}x_{ij} + a_{ij}x_{ij}^2$ is also incurred. The problem is to decide which arcs to install and how to route the flow in such a way that demands are satisfied and the total (installing + routing) cost is minimized. The problem can be written as

$$\begin{array}{lll} \min & \sum_{(i,j) \in A} c_{ij} y_{ij} + b_{ij} x_{ij} + a_{ij} x_{ij}^2 & \\ & \sum_{(j,i) \in A} x_{ji} - \sum_{(i,j) \in A} x_{ij} = d_i & i \in N \\ & l_{ij} y_{ij} \leq x_{ij} \leq u_{ij} y_{ij} \ \ , \ \ y_{ij} \in \{0, 1\} & (i, j) \in A \end{array} \tag{20}$$

This network design problem is $\mathcal{NP}$-hard, since it is a generalization of the sensor placement problem described in Section 3. A recent application of this general model in a Facility Location setting is given in [15, 16].

Again, since $c_{ij} > 0$ (for otherwise y_{ij} can surely be fixed to 1), in the continuous relaxation of (20) the design variables y_{ij} can be projected onto the x_{ij}; that is, at optimality $y_{ij} = x_{ij}/u_{ij}$. The resulting problem can be efficiently solved by means of (convex) Quadratic Min-Cost Flow (QMCF) algorithms; however, the bound provided by the continuous relaxation is usually weak.

Applying the results of Section 2 to (20), a *Separable Convex-cost NonLinear MCF* problem is obtained, where the flow cost function on each arc is a piecewise linear-quadratic convex cost function. In turn, this can be rewritten as a QMCF problem

$$\begin{array}{lll} \min & \sum_{(i,j) \in A'} b'_{ij} \chi_{ij} + a'_{ij} \chi_{ij}^2 & \\ & \sum_{(j,i) \in A'} \chi_{ji} - \sum_{(i,j) \in A'} \chi_{ij} = d_i & i \in N \\ & 0 \leq \chi_{ij} \leq u'_{ij} & (i, j) \in A' \end{array} \tag{21}$$

on a graph $G' = (N, A')$ with the same node set and *at most* 2 times the number of arcs. For each of the original arcs (i, j), at most two "parallel" copies are constructed. If $u_{ij} \leq \sqrt{c_{ij}/a_{ij}}$ (case 2.1), then only one representative of (i, j) is constructed in G', with $b'_{ij} = b_{ij} + a_{ij}u_{ij} + c_{ij}/u_{ij}$, $a'_{ij} = 0$ and $u'_{ij} = u_{ij}$. Instead, if $u_{ij} \geq \sqrt{c_{ij}/a_{ij}} \geq l_{ij}$ (case 2.2) then two parallel copies of the arc (i, j) have to be constructed in G': the first has $b'_{ij} = b_{ij} + 2\sqrt{a_{ij}c_{ij}}$, $a'_{ij} = 0$, and $u'_{ij} = \sqrt{c_{ij}/a_{ij}}$, while the second has $b'_{ij} = b_{ij} + 2\sqrt{a_{ij}c_{ij}}$,

$a'_{ij} = a_{ij}$, and $u'_{ij} = u_{ij} - \sqrt{c_{ij}/a_{ij}}$. Finally, if $l_{ij} \geq \sqrt{c_{ij}/a_{ij}}$ (case 2.3) then two parallel copies of the arc (i,j) have to be constructed in G': the first has $b'_{ij} = b_{ij} + a_{ij}l_{ij} + c_{ij}/l_{ij}$, $a'_{ij} = 0$, and $u'_{ij} = l_{ij}$, while the second has $b'_{ij} = b_{ij} + 2a_{ij}l_{ij}$, $a'_{ij} = a_{ij}$, and $u'_{ij} = u_{ij} - l_{ij}$. For this kind of "partitioned" NonLinear MCF problems—where some of the arcs have strictly convex cost functions, while the other have linear cost functions—specialized algorithms have been proposed in [8]. In general, any algorithm for Convex (Quadratic) MCF problems (see e.g., [4]) can be used. While codes implementing these algorithms are either not available or not very efficient in practice, the off-the-shelf solver `Cplex` turns out to be quite efficient in solving these convex QMCFs.

5. Computational Results

In order to assess the behaviour of the Projected Perspective Reformulation technique we implemented it on the two problems discussed in sections 3 and 4 within a specialized B&B where the perspective relaxation is solved by computing the projection $z(p)$ as in (14)-(16). We considered the reformulations (18) and (21) and, for their solution, we applied the specialized $O(n \log n)$ algorithm for the Sensor Placement problem and the `Cplex` quadratic solver, respectively. We compared the new approach (denoted as P^2/R) against the following ones:

- a B&C on the PR (6) using the Semi-Infinite MILP formulation (denoted as P/C for Perspective Cut method);
- a B&C on the PR (6) using the MI-SOCP formulation (denoted as CPLEX-SOCP);
- a standard B&C on the continuous relaxation (5) (denoted as CPLEX).

These three alternative methods have all been implemented by means of `Cplex` B&C solver. In particular, the P/C method has been coded with a `cut-callback` function. We point out that the P^2/R method cannot be implemented within the `Cplex` B&C solver because it is not allowed by a `Cplex solve-callback` function. We thus used a simple implementation of a B&B method that can be certainly improved by adding new features (e.g. strong branching or more sophisticated primal heuristics). We also note that the solution of the relaxation is to be completed with the values for binary variables y_i; they can be derived by computing $y_i^*(x_i)$ as described in Subsection 2.1 substituing the values of x_i obtained by solving the convex quadratic reformulation. All the algorithms have been coded in `C++`, compiled with GNU `g++ 4.0.1` (with -O3 optimization option) and ran on an Opteron 246 (2 GHz) computer with 2 Gb of RAM, under Linux Fedora Core 3.

We generated 180 random instances of the Sensor Placement problem, grouped in 6 classes with 30 instances each. The first 4 classes contain instances with either 2000 or 3000 sensors and have either high or low quadratic costs. In the former ("h"), fixed costs are uniformly chosen in the interval $[1, n]$ while quadratic costs are uniformly chosen in the interval $[n, C_{max}]$, where $C_{max} \in \{10n, 20n, 30n\}$. In the latter ("l"), fixed costs are randomly generated in the interval $[n, B_{max}]$, where $B_{max} \in \{10n, 20n, 30n\}$, while quadratic costs are randomly generated in the interval $[1, n]$. The last two classes are generated starting from random instances of the Partition problem, according to the NP-hardness proof for the Sensor Placement problem in [2]. We considered 2000 and 3000 Partition items ranging in the intervals [100,1000], [500,1000], [1,100000]. Table 1 reports the obtained results.

For the Network Design Problem we generated 360 problems, grouped into 12 classes with 30 instances each, as follows:

- the underlying flow networks with 1000, 2000, or 3000 nodes have been generated by `netgen` [17], where: (i) the minimum arc cost is 1 and the maximum is randomly generated between 10 and 100, (ii) the total supply d_s is randomly generated between 100 and 1000, and (iii) the minimum arc capacity is $0.05d_s$ and the maximum arc capacity is randomly generated in the interval $[0.2d_s, 0.4d_s]$;
- the fixed costs which are either low or high with respect to the linear costs generated by `netgen`, i.e., c_{ij} is uniformly generated either in $[0.5b_{ij}, b_{ij}]$ ("l") or in $[3b_{ij}, 10b_{ij}]$ ("h");

name	P^2/R			CPLEX			
	time	nodes	av. t/n	time	nodes	av. t/n	gap
2000-h	0.39	1	0.39	1020.51	223293	0.01	4.03
2000-l	0.09	1	0.09	101.58	3713	0.03	0.00
3000-h	0.92	1	0.92	1057.09	144406	0.01	7.18
3000-l	0.21	1	0.21	270.49	5724	0.05	0.00
PTN-2000	0.43	1	0.43	1018.13	4149	0.25	2.98
PTN-3000	1.02	1	1.02	1008.42	568	1.79	3.14
name	P/C			CPLEX - SOCP			
	time	nodes	av. t/n	time	nodes	av. t/n	gap
2000-h	47.74	924	30.43	1066.02	507	2.11	207.04
2000-l	17.02	1	17.02	49.32	38	7.60	0.00
3000-h	91.24	88	74.09	1069.73	332	3.24	412.54
3000-l	40.27	1	40.27	135.95	72	12.08	0.00
PTN-2000	94.30	6	56.93	23.79	1	23.80	0.00
PTN-3000	202.63	6	114.72	53.74	1	53.74	0.00

Table 1: Results for the Sensor Placement problem

- the quadratic costs which are either low or high with respect to the linear costs generated by `netgen`, i.e., a_{ij} is uniformly generated either in $[3b_{ij}, 10b_{ij}]$ ("l") or in $[100b_{ij}, 1000b_{ij}]$ ("h").

Table 2 reports the obtained results.

name	P^2/R			CPLEX			
	time	nodes	av. t/n	time	nodes	av. t/n	gap
1000-h-h	0.05	1	0.05	108.80	35630	0.28	0.00
1000-h-l	0.31	5	0.05	1037.63	324447	0.01	0.02
1000-l-h	0.05	1	0.05	163.67	46685	0.18	0.00
1000-l-l	0.32	5	0.05	1046.89	304305	0.01	0.01
2000-h-h	0.10	1	0.10	690.09	101868	0.11	0.00
2000-h-l	45.42	278	1.10	1031.75	141485	0.01	0.06
2000-l-h	0.09	1	0.09	858.22	131954	0.03	0.00
2000-l-l	8.78	63	0.10	1036.79	140877	0.01	0.04
3000-h-h	0.15	1	0.15	1041.96	88541	0.01	0.00
3000-h-l	71.02	269	0.17	1051.93	73591	0.01	0.12
3000-l-h	0.15	1	0.15	988.74	89209	0.12	0.00
3000-l-l	19.05	79	0.16	1062.45	85878	0.01	0.04
name	P/C			CPLEX - SOCP			
	time	nodes	av. t/n	time	nodes	av. t/n	gap
1000-h-h	17.03	3	10.14	967.30	26	62.86	0.01
1000-h-l	5.89	25	0.38	79.17	46	16.98	0.00
1000-l-h	8.89	4	4.60	620.77	21	38.62	0.00
1000-l-l	4.68	22	0.33	30.46	63	17.37	0.00
2000-h-h	57.09	7	13.84	895.70	8	207.60	0.01
2000-h-l	51.60	348	0.72	252.98	36	27.65	0.00
2000-l-h	42.3	6	16.57	525.35	9	63.35	0.00
2000-l-l	20.60	131	0.51	252.82	193	40.02	0.00
3000-h-h	117.30	11	18.90	564.41	2	407.97	0.01
3000-h-l	140.47	584	1.39	366.95	27	36.76	0.00
3000-l-h	101.18	12	12.01	372.16	4	89.53	0.01
3000-l-l	45.43	153	0.89	292.41	83	62.39	0.00

Table 2: Results for Network Design problems

For our experiments we fixed a time limit of 1000 seconds. All problems where solved at optimality within this time limit with the P^2/R and the P/C methods, therefore we do not report the gap at termination for them. For all methods, we report the running time in seconds, the number of B&B nodes and the average time for node. As expected from previous results [11, 13], the P/C method overcomes CPLEX B&C algorithm both with standard and SOCP formulations. However, the newly proposed P^2/R approach significantly overcomes the P/C method. This is mainly because of the much faster specialized solution methods used for the relaxations, which significantly reduces the effort required at each node. Furthermore, P/C approximates the true perspective relaxations by means of a finite number of cutting planes, thereby introducing some (small) approximation errors; these seem to cause the generation of more B&C nodes w.r.t. the "exact" solutions provided by P^2/R.

6. Conclusions

In this paper we describe a new method, called Projected Perspective Relaxation (P^2/R), to solve the Perspective Relaxation of Mixed-Integer Nonlinear Programming problems with convex objective function and semicontinuous variables. The new method is based on a reformulation which projects the problem onto the subspace of the continuous variables only. The P^2/R method requires three simplifying hypotheses: each semicontinuous variable x_i is univariate, the corresponding binary variable y_i is not involved in other constraints, and the objective functions $f_i(x_i)$ are convex quadratic. The Perspective Relaxation is reformulated as a piecewise convex quadratic programming problem with at most two pieces for each semicontinuous variable in the original model, and then the resulting model is further reformulated by defining a new variable associated with each convex quadratic piece, thus obtaining a new convex quadratic programming problem with at most twice the number of continuous variables. This in turn means that the resulting relaxation has at most the same number of variables of the original Perspective Relaxation; moreover, P^2/R contains only convex quadratic functions, as opposed to rational convex functions (the perspective functions), and the structure of the constraints is now simplified by the elimination of the relaxed binary variables.

We applied the P^2/R method to two cases where we can exploit the structure of the resulting relaxations to speed up the overall solution method: a Sensor Placement problem and a (single-commodity) Network Design problem. In the Sensor Placement problem we obtained a simple continuous knapsack problem with a number of variables that is at most twice the number of possible sensors; in the Network Design problem we obtained a Min Cost Network Flow Problem with at most two copies of the arcs of the original graph. For both problems we carried on an extensive computational experience showing that the new method overcomes the `Cplex` B&C method on both the original continuous quadratic relaxation and the Second Order Cone Programming implementation of the Perspective Relaxation, as well as the Perspective Cuts method implemented by a `cut-callback` function within the `Cplex` B&C solver. We point out that the P^2/R method is the first nonlinear technique that improves on the linearization technique of Perspective Cuts in the two applications here presented.

Finally we outline two directions for future research:

- On the one hand, the P^2/R approach is likely to be applicable to several other problems. In a nutshell, the two applications of the present paper show respectively knapsack and flow structures, which are found in many other problems. A relevant one is Multicommodity Network Design [6] which, especially when approached through decomposition techniques [9], actually displays *both* [7, 10]. Another example are portfolio optimization problems [11], which typically display very few (e.g. two) knapsack-like constraints; while they also typically sport a non-separable function, the PR idea can still be applied, e.g. by means of appropriate diagonalization tricks [12]. However, several combinatorial structures such as paths, cuts, assignments and many others for which specialized algorithms exist are found in applications; each of them is a potential candidate for successfull application of the P^2/R idea.

- On the other hand, it would be interesting to relax some or all of the three basic hypotheses of Section 2. While the hypotheses A1) and A3) do not look to be particularly restrictive, in that they are satisfied by most of the applications of the semicontinuous variables described in the literature so far, hypothesis A2) forbids to improve the bound by means of valid inequalities concerning the y_i variables, and therefore relaxing it may lead to performance improvements for current applications. This does not look to be straightforward, though, as y_i variables have been "projected away" from the formulation.

References

[1] A. Agnetis, E. Grande, P. Mirchandani, and A. Pacifici, "Covering a line segment with variable radius discs," *Computers & Operations Research*, vol. 36, no. 5, pp. 1423–1436, 2009.

[2] A. Agnetis, E. Grande, and A. Pacifici, "Demand allocation with latency cost functions," *CoRR*, vol. abs/0810.1650, 2008.

[3] S. Aktürk, A. Atamtürk, and S. Gürel, "A strong conic quadratic reformulation for machine-job assignment with controllable processing times," *Operations Research Letters*, vol. 37, no. 3, pp. 187–191, 2009.

[4] J. Castro and N. Nabona, "An Implementation of Linear and Nonlinear Multicommodity Network Flows," *European J. of Operational Research*, vol. 92, pp. 37–53, 1996.

[5] S. Ceria and J. Soares, "Convex programming for disjunctive convex optimization," *Mathematical Programming*, vol. 86, pp. 595–614, 1999.

[6] T. Crainic, A. Frangioni, and B. Gendron, "Multicommodity Capacitated Network Design," in *Telecommunications Network Planning* (Soriano, P. and Sanso, B., eds.), pp. 1–19, Kluwer Academics Publisher, 1999.

[7] T. Crainic, A. Frangioni, and B. Gendron, "Bundle-based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems," *Discrete Applied Mathematics*, vol. 112, pp. 73–99, 2001.

[8] R. De Leone, R. Meyer, and A. Zakarian, "A Partitioned ε-Relaxation Algorithm for Separable Convex Network Flow Problems," *Computational Optimization and Applications*, vol. 12, pp. 107–126, 1999.

[9] A. Frangioni, "About Lagrangian Methods in Integer Optimization," *Annals of Operations Research*, vol. 139, pp. 163–193, 2005.

[10] A. Frangioni and B. Gendron, "0-1 Reformulations of the Multicommodity Capacitated Network Design Problem," *Discrete Applied Mathematics*, vol. 157, no. 6, pp. 1229–1241, 2009.

[11] A. Frangioni and C. Gentile, "Perspective Cuts for 0-1 Mixed Integer Programs," *Mathematical Programming*, vol. 106, no. 2, pp. 225–236, 2006.

[12] A. Frangioni and C. Gentile, "SDP Diagonalizations and Perspective Cuts for a Class of Nonseparable MIQP," *Operations Research Letters*, vol. 35, no. 2, pp. 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*, vol. 37, no. 3, pp. 206–210, 2009.

[14] A. Frangioni, C. Gentile, and F. Lacalandra, "Tighter Appro-ximated MILP Formulations for Unit Commitment Problems," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 105–113, 2009.

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

[16] O. Günlük and J. Linderoth, "Perspective relaxation of MINLPs with indicator variables," in *Proceedings 13^{th} IPCO* (A. Lodi, A. Panconesi, and G. Rinaldi, eds.), vol. 5035 of *Lecture Notes in Computer Science*, pp. 1–16, 2008.

[17] D. Klingman, A. Napier, and J. Stutz, "NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems," *Management Science*, vol. 20, no. 5, pp. 814–821, 1974.

[18] R. Stubbs and S. Mehrotra, "A branch-and-cut method for 0-1 mixed convex programming," *Mathematical Programming*, vol. 86, pp. 515–532, 1999.

[19] M. Tawarmalani and N. Sahinidis, "Convex extensions and envelopes of lower semi-continuous functions," *Mathematical Programming*, vol. 93, pp. 515–532, 2002.

[20] J. Zamora and I. Grossmann, "A global MINLP optimization algorithm for the synthesis of heat exchanger networks with no stream splits," *Comput & Chem. Engin.*, vol. 22, pp. 367–384, 1998.