An Outer-Inner Approximation for separable MINLPs

Hassan Hijazi
LIX, École Polytechnique, F-91128 Palaiseau, France, hijazi@lix.polytechnique.fr

Pierre Bonami
LIF, Faculté des Sciences de Luminy, Université de Marseille, France, pierre.bonami@lif.univ-mrs.fr.

Adam Ouorou
Orange Labs R&D/CORE/TPN, 38-40 rue du Général Leclerc, 92794 Issy-Les-Moulineaux cedex 9, France,
adam.ouorou@orange-ftgroup.com

A common structure in convex mixed-integer nonlinear programs is additively separable nonlinear functions consisting of a sum of univariate functions. In the presence of such structures, we propose three improvements to the classical Outer Approximation algorithms that exploit separability. The first improvement is a simple extended formulation. The second a refined outer approximation. Finally, the third one is an Inner Approximation of the feasible region which approximates each univariate functions from the interior and can be used to find feasible solutions by solving mixed-integer linear programs. These methods have been implemented in the open source solver Bonmin and are available for download from the COIN-OR project website. We test the effectiveness of the approach on three applications.

Key words: Mixed-Integer Nonlinear Programming, Outer Approximation.

1 Introduction

A well known approach for solving convex mixed-integer nonlinear Programs (MINLPs where one minimizes a convex objective over the intersection of a convex region and integrity requirements) is the *Outer Approximation* of Duran and Grossmann (1986). Outer approximation consists in building a mixed-integer linear equivalent to the feasible region of the problem by taking tangents at well specified points. Outer approximation is a very successful approach to solve convex MINLPs. Different algorithms have been devised to build the mixed-integer linear equivalent and they are implemented in several state-of-the-art solvers such as DICOPT (Grossmann et al., 2001), Bonmin (Bonami et al., 2008) and filMINT (Abhishek et al., 2010).

In this paper, we propose several improvements to outer approximation for a subclass of MINLPs featuring particular structures. Specifically, we study *separable convex MINLPs* of the form

$$\begin{array}{lll} \min & \mathbf{c}^T\mathbf{x} & \\ \text{s.t.} & g_i(\mathbf{x}) \leq 0 & i = 1,\dots,m \\ & \mathbf{x} \in X & \\ & x_j \in \mathbb{Z} & j = 1,\dots,p \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} & \end{array} \tag{sMINLP}$$

where X is a polyhedral subset of $\mathbb{R}^n$ and for $i = 1,\dots,m$ the functions $g_i : X \to \mathbb{R}$ are *convex separable*. By convex separable, we mean that the function g_i can be rewritten as a sum of convex univariate function: $g_i(\mathbf{x}) = \sum_{j=1}^n g_{ij}(x_j)$ with $g_{ij} : \mathbb{R} \to \mathbb{R}$ is convex. Note that any problem with a convex separable objective function and constraints can be cast into that form by introducing an extra variable and moving the objective function into the constraints. Finally, we assume here that all the variables are bounded.

(sMINLP) is one of the simplest form of MINLP and one could expect outer approximation to perform very well on problems having such structure. We show that this is not the case and even a problem with one separable quadratic constraint can require an exponential number of iterations to be solved by outer approximation. We then propose three improvements allowing to overcome this issue. First, we exploit the separability to build a simple extended formulation which can be shown to have better convergence performances. Second, we build a good initial linear approximation of the problem. Third, we propose a new method for finding near-optimal feasible solutions.

Our approach to finding these solutions is also to build a mixed-integer linear approximation of the feasible region of (sMINLP) but instead of approximating it by the outside (i.e. by tangents), we approximate it by the inside. The separable structure of the problem we consider make this approximation simple to build. Indeed, it suffices to compute chords of each univariate convex function separately.

In the next section, we briefly recall the basic concepts behind outer approximation algorithms. In Section 3, we describe our approach. In Section 4, we present computational experiments comparing our approach to state-of-the art solvers. Finally, in Section 5, we present some conclusions.

In the remainder, we denote by $\mathbf{g}$, the function $X \to \mathbb{R}^m$ consisting of all the functions g_i. Given a vector $\mathbf{x} \in \mathbb{R}^n$, let $I = \{1, 2, .., p\}$ be the subset of the first p indices constrained to be integer, we denote by $\mathbf{x}_I \in \mathbb{R}^p$ the corresponding sub-vector.

2 Outer Approximation

Consider any point $\mathbf{x}^* \in X$, by convexity all $\mathbf{x} \in X$ such that $g_i(\mathbf{x}) \leq 0$ verify the following *outer approximation* constraint:

$$\nabla g_i(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) + g_i(\mathbf{x}^*) \leq 0. \tag{1}$$

More generally, we call Outer Approximations constraints the set of constraints of the form (1) taken for $j = 1, \ldots, m$.

Outer approximation algorithms are based on the the mixed-integer linear relaxation of (sMINLP) obtained by replacing the nonlinear constraints by their linear outer approximations taken in a set of points $\mathcal{P} = \{\hat{\mathbf{x}}^{(1)}, \ldots, \hat{\mathbf{x}}^{(K)}\}$. We denote the polyhedral set given by these outer approximations constraints by

$$\mathcal{X}(\mathcal{P}) := \begin{cases} \nabla g_i(\hat{\mathbf{x}}^{(k)})^T(\mathbf{x} - \hat{\mathbf{x}}^{(k)}) + g_i(\hat{\mathbf{x}}^{(k)}) \leq 0, & \forall i \in \{1, \ldots, m\},\, \hat{\mathbf{x}}^{(k)} \in \mathcal{P}, \\ \mathbf{x} \in X, & \\ x_i \in \mathbb{Z}, & \forall i \in \{1, \ldots, p\}, \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, & \end{cases}$$

and the corresponding MILP:

$$\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{x} \in \mathcal{X}(\mathcal{P}). \end{array} \tag{OA($\mathcal{P}$)}$$

The main theoretical justification of outer approximation algorithms is that if (OA($\mathcal{P}$)) contains a suitable set of points then it has the same optimal value as (sMINLP). More precisely, let $\mathcal{Y} := \{\mathbf{y} \in \mathbb{Z}^p : l_j \leq y_j \leq u_j,\, j = 1, \ldots, p\}$ be the set of all possible assignments for the integer constrained variables. For each $\mathbf{y} \in \mathcal{Y}$ (note that by our assumptions $\mathcal{Y}$ is finite), we define:

$$\boldsymbol{\xi}(\mathbf{y}) := \begin{cases} \arg\min \ \{\mathbf{c}^T\mathbf{x} : \mathbf{g}(\mathbf{x}) \leq 0,\, \mathbf{x} \in X, \mathbf{x}_I = \mathbf{y}\} & \text{if it exists,} \\ \arg\min \ \{\sum_{i=1}^{m} \max(g_i(\mathbf{x}), 0) : \mathbf{x} \in X, \mathbf{x}_I = \mathbf{y}\} & \text{otherwise,} \end{cases}$$

and $\mathcal{P}(\mathcal{Y}) := \{\boldsymbol{\xi}(\mathbf{y}) : \text{for all } \mathbf{y} \in \mathcal{Y}\}$. OA($\mathcal{P}(\mathcal{Y})$) has the same optimal value as (sMINLP) (Duran and Grossmann, 1986; Fletcher and Leyffer, 1994; Bonami et al., 2008).

The aim of outer approximation algorithms is to build this equivalent mixed-integer linear program. To do so, two algorithmic ideas have been developed: the Outer Approximation Algorithm (Duran and Grossmann, 1986) and the Outer Approximation Branch-and-Cut (Quesada and Grossmann, 1992). In this paper we will mostly consider the first of these two algorithms which is specified in Algorithm 1. Note nevertheless that our improvements could also be applied in the context of the second one (for refinements of these two algorithms and implementation details the reader should report to Fletcher and Leyffer 1994; Bonami et al. 2008; Abhishek et al. 2010; Bonami et al. 2009).

The convergence of Algorithm 1 follows directly from the equivalence between OA($\mathcal{P}(\mathcal{Y})$) and (sMINLP). Note that, in the worst case, the algorithm may have to solve a MILP for all points $\mathbf{y} \in \mathcal{Y}$. Nevertheless, in practice the algorithm is one of the most competitive algorithms available (see for example Bonami et al. 2009 for a recent experimental comparison). In that light, one could expect it to perform well on simple MINLPs where all the nonlinear functions are separable. Below, we exhibit a simple example showing that this is not the case and even if there is one nonlinear constraints with quadratic separable terms the outer approximation algorithm may take an exponential number of iterations to converge.

Algorithm 1 The Outer Approximation Algorithm

0. **Initialize.**
   $z_U \leftarrow +\infty$. $z_L \leftarrow -\infty$. $\mathbf{x}^* \leftarrow$ NONE. Let $\mathbf{x}^0$ be an optimal solution of the continuous relaxation of (sMINLP), $\mathcal{P} \leftarrow \{\mathbf{x}^0\}$.
1. **Termination?**
   If $z_U - z_L = 0$ or OA($\mathcal{P}$) is infeasible, then $\mathbf{x}^*$ is optimal.
2. **Lower Bound**
   Let $z_{\text{OA}(\mathcal{P})}$ be the optimal value of OA($\mathcal{P}$) and $\hat{\mathbf{x}}$ its optimal solution. $z_L \leftarrow z_{\text{OA}(\mathcal{P})}$
3. **Find new linearization point**
   $\mathcal{P} \leftarrow \mathcal{P} \cup \{\xi(\hat{\mathbf{x}}_I)\}$.
4. **Upper Bound?**
   If $\xi(\hat{\mathbf{x}}_I)$ is feasible for (sMINLP) and $\mathbf{c}^T\xi(\hat{\mathbf{x}}_I) < z_U$, then $\mathbf{x}^* \leftarrow \xi(\hat{\mathbf{x}}_I)$ and $z_U \leftarrow \mathbf{c}^T\xi(\hat{\mathbf{x}}_I)$.
5. **Iterate**
   $i \leftarrow i + 1$. Go to 1.

Example 1 Outer Approximation taking exponentially many iterations
We consider the subset of $\mathbb{R}^n$ obtained by intersecting the ball $B(\rho, r)$ of radius $r = \frac{\sqrt{n-1}}{2}$ centered in the point $\rho = \left(\frac{1}{2}, \dots, \frac{1}{2}\right)$ with the vertices of the unit hypercube $\{0,1\}^n$:

$$B^n = \left\{ \mathbf{x} \in \{0,1\}^n \mid \sum_{i=1}^{n} \left(x_i - \frac{1}{2}\right)^2 \leq \frac{n-1}{4} \right\}.$$

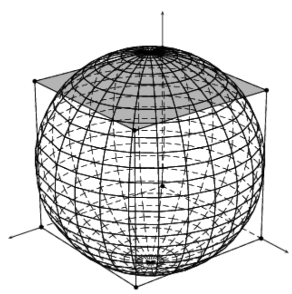

Figure 1: The set $\mathcal{B}^3$ in $\mathbb{R}^3$.

Clearly, since the vertices of the hypercube are at distance $\frac{\sqrt{n}}{2}$ from $\rho = \left(\frac{1}{2}, \dots, \frac{1}{2}\right)$, $B^n = \emptyset$. Note also that the ball is chosen so that it has a non-empty intersection with all the edges of the cube.

In our first lemma, we show that an outer approximation constraint obtained from the ball defining $\mathcal{B}^n$ can not cut simultaneously 2 vertices of the hypercube. Note that by definition outer approximation constraints are valid linear inequalities for $B(\rho, r)$ (i.e. are satisfied by all points in $B(\rho, r)$).

Lemma 2.1. A valid linear inequality $\alpha^T x \leq \beta$ for the ball $B(\left(\frac{1}{2}, \dots, \frac{1}{2}\right), \frac{\sqrt{n}}{2})$ can not cut two vertices of the unit hypercube simultaneously.

Proof. Suppose that an inequality cuts two vertices of the hypercube x and y (i.e. $\alpha^T x > \beta$ and $\alpha^T y > \beta$), then it cuts the whole segment $[x, y]$. By construction $B(\rho, r)$ has a non-empty intersection with $[x, y]$, therefore the inequality is not valid for $B(\rho, r)$. □

It follows directly from the lemma that the set $\mathcal{P}$ of linearization points would need to contain at least 2^n points for the outer approximation set $\mathcal{X}(\mathcal{P})$ to be empty.

Theorem 2.2. The mixed-integer linear program (OA($\mathcal{P}$)) *built from the set* B^n *is feasible if* $|\mathcal{P}| < 2^n$.

Proof. Since, no linear inequality can cut two vertices of the hypercube simultaneously, if $|\mathcal{P}| < 2^n$, at least one vertex is feasible. □

It follows directly from this theorem that Algorithm 1 would need 2^n iterations to converge (remember that each iteration involves the solution of a MILP). We note that in the context of an Outer Approximation branch-and-cut also, at least 2^n nodes would need to be enumerated. This indicates that even on very simple examples of separable MINLPs, the outer approximation presents extremely bad behavior.

In the next section, we will show that, by a simple modification, this problem can be solved with an outer approximation procedure in much less iterations.

3 The new scheme

3.1 Univariate Extended Formulation

Our first ingredient is a simple reformulation of (sMINLP) in an extended space. An auxiliary variable is introduced for each univariate function in the problem and the separable functions are broken into univariate functions:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & \sum_{j=1}^{n} y_{ij} \le 0 & i = 1, \dots, m, \\ & g_{ij}(x_j) \le y_{ij} & \begin{array}{l} i = 1, \dots, m, \\ j = 1, \dots, n, \end{array} \\ & x \in X, & \\ & x_i \in \mathbb{Z} & i = 1, \dots, p, \\ & l \le x \le u. & \end{array} \tag{sMINLP*}$$

Clearly, if (x, y) is an optimal solution to (sMINLP*), then x is optimal for (sMINLP). Note, that it may happen that the same univariate function of the same variable appears in several constraints. In that case, it is of course advantageous to introduce only one auxiliary variable for all appearances of the function.

Extended formulations are widely used in integer programming and combinatorial optimization as a mean to obtain stronger relaxations. The difference between (sMINLP) and (sMINLP*) may seem innocuous; it is quite clear that the continuous relaxations of both problems are identical. Nevertheless , as noted previously by Tawarmalani and Sahinidis (Tawarmalani and Sahinidis, 2005), linear outer approximations of (sMINLP*) yield tighter approximations than those of (sMINLP). More precisely, Proposition 5 in (Tawarmalani and Sahinidis, 2005) shows that to obtain a linear relaxation of (sMINLP) of the same strength as a linear approximation of (sMINLP*) one needs exponentially many more linearization points. The univariate extended formulation obtained from example 1 gives a particularly striking illustration of this in the context of the outer approximation algorithm.

Example1 (continued)
We denote by B^{n*} the extended formulation of the mixed-integer set B^n:

$$B^{n*} = \left\{ (\mathbf{x}, \mathbf{z}) \in \{0,1\}^n \times \mathbb{R}^n \mid \left(x_i - \frac{1}{2}\right)^2 \le z_i, \ \forall i \in \{1, 2, ..n\}, \ \sum_{i=1}^{n} z_i \le \frac{n-1}{4} \right\}.$$

Given a set of point $\mathcal{P} := \left\{ \begin{pmatrix} \hat{\mathbf{x}}^{(1)} \\ \hat{\mathbf{z}}^{(1)} \end{pmatrix}, \dots, \begin{pmatrix} \hat{\mathbf{x}}^{(K)} \\ \hat{\mathbf{z}}^{(K)} \end{pmatrix} \right\}$, the outer approximation of B^{n*} is given by:

$$\mathcal{B}^{n*}(\mathcal{P}) = \left\{ \begin{array}{l} (\mathbf{x}, \mathbf{z}) \in \{0,1\}^n \times \mathbb{R}^n \mid \\ \left(2\hat{x}_i^{(k)} - 1\right) x_i \le (\hat{x}_i^{(k)})^2 - \frac{1}{4} + z_i, \ \forall i \in \{1, 2, ..n\}, \ \forall \hat{x}^{(k)} \in \mathcal{P}^K, \\ \sum_{i=1}^{n} z_i \le \frac{n-1}{4} \end{array} \right\}$$

Note that $\hat{\mathbf{z}}^{(k)}$ does not appear in the outer approximation and can be neglected. It can easily be seen that $\mathcal{P}$ only needs 2 points to make $\mathcal{B}^{n*}$ empty. Take $\mathbf{x}^1 \in \{0,1\}^n$ and $\mathbf{x}^2$ the complement of $\mathbf{x}^1$ ($x_i^2 = 1 - x_i^1$, $i = 1,\dots,n$). $\mathcal{B}^{n*}(\{\mathbf{x}^1, \mathbf{x}^2\})$ is

$$\left\{ \begin{array}{l} (\mathbf{x},\mathbf{z}) \in \{0,1\}^n \times \mathbb{R}^n \mid \\ \frac{1}{4} - x_i \le z_i, \ \forall i \in \{1,2,..n\}, \\ x_i - \frac{3}{4} \le z_i, \ \forall i \in \{1,2,..n\}, \\ \sum_{i=1}^{n} z_i \le \frac{n-1}{4}. \end{array} \right\}$$

This set is empty since $x \in \{0,1\}^n$ implies $z_i \ge \frac{1}{4}$ for $i = 1,\dots,n$ and $\sum_{i=1}^{n} z_i \ge \frac{n}{4} > \frac{n-1}{4}$. This shows that on this simple problem while the classical version of the outer approximation takes 2^n iterations, the extended formulation can be solved in only 2 iterations (note that if we count the number of constraints instead of linearization points, $2n$ constraints are generated versus 2^n). Thus in general, we can expect the extended formulation to perform much better than the initial one in an outer approximation algorithm.

3.2 Refined Initial Outer Approximation

The second ingredient of our approach consists in building a better initial outer approximation of the feasible region of (sMINLP*). Note that in Algorithm 1, the first outer approximation is built by taking only one linearization point, at an optimal solution of the continuous relaxation of (sMINLP). In general, it is not obvious how to choose other linearization points that will improve the bound obtained by solving the associated MILP. Here we just exploit the simplicity of univariate functions to choose a better initial set of points.

Consider one of the nonlinear constraints of (sMINLP*) $g_{ij}(x_j) \le y_{ij}$ and the set $S = \{(x_j, y_{ij}) \in \mathbb{R}^2 \ s.t. \ g_{ij}(x_j) \le y_{ij}, \ l_j \le x_j \le u_j\}$. We enrich the first outer approximation with a set of points of the form $(\hat{x}_j, g_{ij}(\hat{x}_j))$, for $\hat{x}_j \in [l_j, u_j]$:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & \sum_{j=1}^{n} y_{ij} \le 0 & i = 1,\dots,m, \\ & \begin{array}{c} \nabla g_{ij}(\hat{x}_j^k)(x_j - \hat{x}_j^k) + \\ g_{ij}(\hat{x}_j^k) \le y_{ij} \end{array} & \begin{array}{l} i = 1,\dots,m, \\ j = 1,\dots,n, \\ k = 1,\dots K \end{array} \\ & x \in X & \\ & x_i \in \mathbb{Z} & i = 1,\dots,p, \\ & l \le x \le u & \end{array} \tag{OA*}$$

This set may be obtained in different ways. For example, one could sample points uniformly in the interval $[l_j, u_j]$ or perform a random sampling, or try to make a finner approximation by taking into account function curvature. After different tries, in our experiments, we sticked to the simplest option: uniform sampling. It typically provided good computational results. Figures 2 and 3 give example of S and S_{out} for the constraint $\frac{1}{6-x} \le y$.

This initial approximation can be used as a starting point by any classical Outer Approximation based algorithm.

3.3 Inner Approximation

The third and last ingredient, is a procedure for finding a first feasible solution to the problem. To do so we build a second mixed-integer linear program. The difference is that instead of building an outer approximation, we now approximate the feasible region by the inside.

Consider again one of the nonlinear constraints of (sMINLP*) $g_{ij}(x_j) \le y_{ij}$ and the set $S = \{(x_j, y_{ij}) \in \mathbb{R}^2 \ s.t. \ g_{ij}(x_j) \le y_{ij}, \ l_j \le x_j \le u_j\}$. Instead of taking tangents, we now take chords at the boundary of S. Consider a sequence of K points $(\hat{x}_j^k, g_{ij}(\hat{x}_j^k))$, $k \in \{1,2,...,K\}$ such that $l_j = \hat{x}_j^1 \le \hat{x}_j^2 \le \ldots \le \hat{x}_j^K = u_j$. We

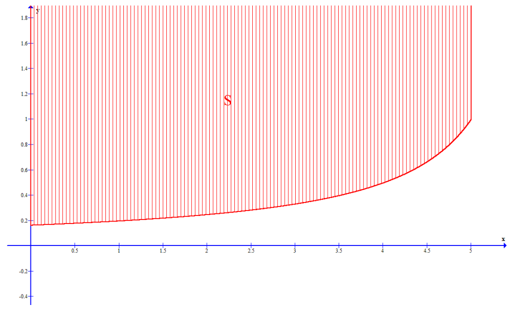

Figure 2: $S = \{(x, y) \in \mathbb{R}^2 \ s.t. \ \frac{1}{6-x} \leq y, \ 0 \leq x \leq 5\}$.

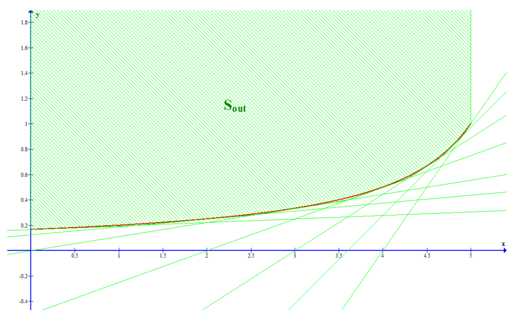

Figure 3: S_{out}.

consider the polyhedral approximation S_{in} of S formed by the constraints:

$$\frac{g_{ij}(\hat{x}_j^{k+1}) - g_{ij}(\hat{x}_j^k)}{\hat{x}_j^{k+1} - \hat{x}_j^k} x_j + \frac{g_{ij}(\hat{x}_j^k)\hat{x}_j^{k+1} - g_{ij}(\hat{x}_j^{k+1})\hat{x}_j^k}{\hat{x}_j^{k+1} - \hat{x}_j^k} \leq y_{ij}$$

for $k = 1, \ldots, K-1$ and $l_j \leq x_j \leq u_j$. It is trivial to check that $S_{in} \subset S$. Figure 4 illustrates this inner approximation on the constraint of Figure 2.

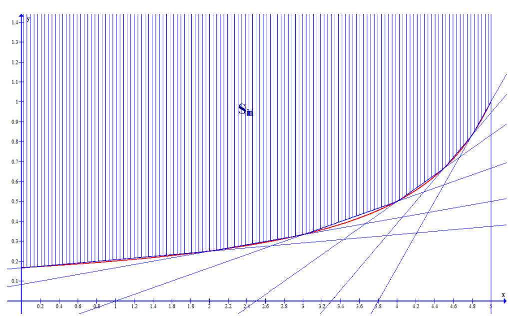

Figure 4: Inner approximation of S with 7 discretization points.

Computing this inner approximation for every nonlinear constraint of the problem, we obtain the Mixed-

Integer Linear Program

$$
\begin{array}{lll}
\min & c^T x & \\
\text{s.t.} & \sum_{j=1}^{n} y_{ij} \leq 0 & i = 1, \dots, m, \\
& \hat{\sigma}_{ij}^k x_j + \hat{\gamma}_{ij}^k \leq y_{ij} & \begin{array}{l} i = 1, \dots, m, \\ j = 1, \dots, n, \\ k = 1, \dots K \end{array} \\
& x \in X & \\
& x_i \in \mathbb{Z} & i = 1, \dots, p,
\end{array}
\tag{IA}
$$

Where $\hat{\sigma}_{ij}^k = \frac{g_{ij}(\hat{x}_j^{k+1}) - g_{ij}(\hat{x}_j^k)}{\hat{x}_j^{k+1} - \hat{x}_j^k}$ and $\hat{\gamma}_{ij}^k = \frac{g_{ij}(\hat{x}_j^k)\hat{x}_j^{k+1} - g_{ij}(\hat{x}_j^{k+1})\hat{x}_j^k}{\hat{x}_j^{k+1} - \hat{x}_j^k}$.

Since the nonlinear constraints are inner approximated, every solution to (IA) is feasible for (sMINLP). We solve (IA) to find an initial feasible solution to (sMINLP).

The combination of (OA($\mathcal{P}$)) and (IA) gives rise to an Outer-Inner Approximation scheme which hopefully results in a smaller initial gap thanks to good lower and upper bounds. In the next section we present computational experiments aimed at assessing the effectiveness of this scheme.

4 Computational Results

We have implemented the Outer-Inner Approximation in the open source solver Bonmin (Bonami et al., 2008) from COIN-OR (Lougee-Heimer, 2003). Our implementation, called Sepa can be found in the trunk distribution of Bonmin in the sub-directory `experimental/Separable`. Our program takes as input a convex separable MINLP under the form (sMINLP*) (i.e. already reformulated) and applies to it an Outer Approximation decomposition scheme augmented with our improvements:

- the refined initial outer approximation described in Section 3.2,
- the inner approximation scheme described in Section 3.3.

The number of discretization points at which the linear approximations are built is a user-set parameter and is initialized to 20 in our experiments.

Below, we present experiments aimed at comparing our algorithm to the standard Outer Approximation algorithms implemented in Bonmin, called `B-OA`, and when possible to the state of the art solver CPLEX. The codes were compared on three classes of problems : separable quadratic facility location, delay constrained routing, and stochastic service design problems.

All our experiments have been performed on a machine equipped with 4 Intel Quad Core Xeon 2.93 GHz CPUs and 120 GiB of RAM. We use the trunk version of Bonmin with Ipopt 3.8 (Wächter and Biegler, 2006) as the nonlinear programming solver and IBM-CPLEX version 12.1 as the MILP solver. We use the multi-threaded version of CPLEX on 10 threads.

4.1 Separable Quadratic Facility Location problems (SQFL)

SQFL is a variant of the classical uncapacitated facility location problem introduced by Günlük et al. (2007). We are given a set of customers J and a set of candidate locations I. Each customer has a unit demand, and facilities an unlimited capacity. There is a fixed cost for opening a facility plus a shipping cost that is proportional to the square of the quantity delivered. The problem can be modeled as follows:

$$
\begin{array}{lll}
\min & \sum_{i \in I} f_i z_i + \sum_{i \in I, j \in J} q_{ij} x_{ij}^2 & \\
\text{s.t.} & x_{ij} \leq z_i & \forall i \in I,\, j \in J \\
& \sum_{i \in I} x_{ij} = 1 & \forall j \in J \\
& z_i \in \{0,1\},\, x_{ij} \in [0,1] & \forall i \in I,\, j \in J
\end{array}
$$

#	$\lvert I\rvert$	$\lvert J\rvert$	`B-OA`	Sepa	CPLEX	CPLEX SOCP
1	8	30	471	18	0.4	2.63
2	20	100	[∞]	35	6.8	48
3	20	150	[∞]	[1.05%]	315	25
4	30	150	[∞]	2372	152	323
5	35	300	[∞]	[61.2%]	[53.3%]	2156

Table 1: CPU times for SQFL instances.

#	`B-OA`		Sepa		CPLEX	CPLEX SOCP
	#It.	# Nodes	#It.	# Nodes	# Nodes	# Nodes
1	232	33076	2	263	97	8
2	183	71128	2	426	188	31
3	100	24340	6	93104	10929	0
4	83	7826	4	14651	2502	113
5	51	2338	1	37146	36748	3

Table 2: Nodes and iterations for SQFL instances.

Here we compare several different formulations and algorithms for solving this problem.

Our main focus is to compare the solution of this problem by Outer Approximation algorithms. The problem can of course directly be solved by Bonmin's `B-OA`. Alternatively, the univariate formulation can be applied (we note that the problem has a nonlinear objective but the extension to this case is absolutely straightforward). This extended formulation can then be solved by our solver Sepa. For completeness, we compare these two formulations and algorithms with two more classical branch-and-bounds. First, since the initial problem formulation is a standard Mixed-Integer Quadratic Program with a convex quadratic objective and linear constraints, it can be solved directly with an MIQP branch-and-bound. Here, we solve this formulation with CPLEX. Second, we test a tighter formulations for this problem proposed by Günlük and Linderoth (2010) and which uses perspective functions and second-order cone constraints . We solve this SOCP formulation also by branch-and-bound with CPLEX (note that this formulation requires a solver dedicated to SOCP and therefore Bonmin is not a good choice, see Section 6.1 in Günlük and Linderoth 2010).

In Tables 1 and 2, we present statistics on the solution of five instances of SQFL of increasing size with these four algorithmic options: Outer-approximation on the initial formulation (`B-OA`), Outer-Inner scheme on the extended formulation (Sepa), CPLEX on the MIQP formulation (CPLEX) and CPLEX on the SOCP formulation (CPLEX SOCP). All algorithms where given a time limit of three hours of CPU time. In Table 1, we give the number of customers and location in each instances and either the total CPU time to solve the instance or, if the instance could not be solved within the time limit, the final relative gap between brackets. In Table 2, we give the number of outer approximation iterations (i.e. number of MILPs solved) for the two outer approximation based algorithms, and the total number of branch-and-bound nodes for all algorithms (note that for the two outer approximation based algorithms, this is a cumulative number for all iterations).

The results show that for this problem branch-and-bound approach are better suited. For all instances, the two CPLEX based solutions give the best result. As should be expected from (Günlük and Linderoth, 2010) the SOCP formulation is superior for the bigger instances. Nevertheless, the comparison between the two outer approximation based algorithms is interesting. First, we note that this problem presents a similar behavior to the one in Example 1. The first problem which has only 8 candidate locations (and therefore 8 binary variables) is the only one solved by the vanilla implementation of the outer approximation algorithm. Its solution requires the solution of 232 MILPs which is not far from the worst case of 2^8. On the other hand, the Outer-Inner code on the extended formulation is able to solve 3 instances out of 5 and, even though it is not competitive with CPLEX, the number of iterations is usually very reasonable.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert K\rvert$	B-OA	CPLEX SOCP	Sepa
rdata1	60	280	100	**11.4**	357.8	23.5
rdata2	61	148	122	0.43	**0.32**	0.66
adata3	100	600	200	291.28	[2.4%]	**260**
rdata4	34	160	946	426.43	[3.7%]	**85.1**
rdata5	67	170	761	6330.91	[∞]	**3285.7**
adata6	100	800	500	881.65	[1.1%]	**393.2**

Table 3: Results for $N = 1$ (mono-routing), 10 candidate paths per commodity and 1% optimality relative gap.

4.2 Delay constrained routing problem

The delay constrained routing problem consists in finding an optimal routing in a telecommunication network that satisfies both a given demand and constraints on the delay of communications. The problem was first introduced by Ben Ameur and Ouorou (2006). We are given a graph $G = (V, E)$ with arc costs w_e and capacities c_e and a set of K demands which are described by the set of paths they can use $P(k) = \{P_k^1, P_k^2, ..., P_k^{n_k}\}$, the quantity of flow v_k that need to be routed and the maximum delay for this demand α_k. A feasible routing is an assignment of a sufficient flow to candidate paths to satisfy demand and such that the total delay on each *active* path P_k^i is smaller than α_k. The delay on an active path P_k^i is given by the nonlinear function $\sum_{e\in P_k^i} \frac{1}{c_e - x_e}$ where x_e is the total quantity of flow going through edge e. Although several formulations of this problem have been proposed (see Ben Ameur and Ouorou, 2006; Hijazi et al., 2010), we only consider here the so called big-M formulation. Denoting by x_e the total quantity of flow routed through edge e, ϕ_k^i the quantity of flow routed along path P_k^i, and z_k^i an indicator variable indicating if path P_k^i is active, the model reads

$$
\begin{aligned}
\min\ & \sum_{e\in E} w_e x_e \\
\text{s.t.} & \sum_{i=1}^{n_k} \phi_k^i \geq 1, && \forall k \in K \\
& \sum_{k\in K} \sum_{P_k^i \ni l} \phi_k^i v_k \leq x_e, && \forall e \in E \\
& x_e \leq c_e,\ \forall e \in E \\
& \sum_{e\in P_k^i} \frac{1}{c_e - x_e} \leq M - z_k^i (M - \alpha_k), && \forall k \in K,\ \forall P_k^i \in P(k). \\
& \sum_{P_k^i \in P(k)} z_k^i \leq N, && \forall k \in K \\
& \phi_k^i \leq z_k^i, && \forall k \in K,\ \forall P_k^i \in P(k) \\
& z_k^i \in \{0, 1\}, && \forall k \in K,\ \forall P_k^i \in P(k) \\
& \phi_k^i \in [0, 1], && \forall k \in K,\ \forall P_k^i \in P(k) \\
& x_e \in \mathbb{R}, && \forall e \in E.
\end{aligned}
$$

The model is clearly separable since the only nonlinear function is $\sum\limits_{e\in P_k^i} \frac{1}{c_e - x_e}$. Also, as shown in (Hijazi, 2010), the delay constraints are second order cone representable, and the problem can be solved using CPLEX SOCP branch-and-bound. In Tables 3 and 4, we report computational experiments of solving the natural formulation with B-OA, the extended formulation with Sepa and the SOCP formulation with CPLEX. Like before, Table 3 reports computing time to 1% of optimality (or final gap after 3 hours of computations) and Table 4 reports number of nodes and of outer approximation iterations.

Table 4: Number of nodes + number of Master MIPs for 10 candidate paths.

Instance	B-OA		CPLEX SOCP	Sepa	
	# Nodes	# It.	# Nodes	# Nodes	# It.
rdata1	2250	4	15259	**962**	**1**
rdata2	**0**	**0**	**0**	**0**	**0**
adata3	16256	6	38302	**7833**	**1**
rdata4	1840	3	172849	**733**	**1**
rdata5	**37748**	5	142256	48197	**1**
adata6	4654	8	104947	**1884**	**1**

First, we note that, the SOCP formulation solved by CPLEX is not efficient in practice for this problem. The extended model Sepa solved using the Outer-Inner approximation algorithm gives the best performance with respect to time. Let us emphasize that, in this experiment, the initial linear approximation derived from Sepa is sufficient to obtain a 1% optimality gap at the first iteration while the classical outer approximation algorithm needs an average of 4.3 outer approximation iterations to get within this gap.

4.3 Stochastic Service Systems Design Problems (SSSD)

SSSD consist in configuring optimally the service levels of a network of M/M/1 queues. It was first proposed by Elhedhli (2006). We are given a set of customers J , a set of facilities I and a set of service levels K. Each customer has a mean demand rate λ_j. The goal is to determine service rates for the facilities so that the customer demand is met. There is a fixed cost for operating facility j at rate k, a fixed cost for assigning customer i to facility j. A binary variable x_{ij} indicates if customer i is served by facility j, and a binary variable y_{jk} indicates if facility j is operated at service level k. Service level k in facility j has a predetermined mean service rate μ_{jk}. A convex MINLP model for the problem reads:

$$
\begin{aligned}
\min \ & \sum_{i\in I, j\in J} c_{ij}x_{ij} + t\sum_{j\in J} v_j + \sum_{j\in J, k\in K} f_{jk}y_{jk} \\
\text{s.t.} & \sum_{i\in I}\lambda_i x_{ij} = \sum_{k\in K}\mu_{jk}y_{jk} && \forall j\in J,\\
& \sum_{j\in J} x_{ij} = 1 && \forall i\in I,\\
& \sum_{k\in K} y_{jk} \le 1 && \forall j\in J,\\
& z_{jk} - y_{jk} \le 0 && \forall j\in J,\, k\in K,\\
& z_{jk} - \frac{v_j}{1+v_j} \le 0 && \forall j\in J,\, k\in K,\\
& z_{jk}\ge 0, v_j\ge 0,\ \forall j\in J,\, k\in K\\
& x_{ij}, y_{jk}\in\{0,1\}\ \forall i\in I,\, j\in J,\, k\in K
\end{aligned}
$$

(where z_{jk} and v_j are auxiliary variable aimed at ensuring convexity of the model.)

The initial model for this problem is already in a form where not more than one univariate function appears in each constraint. Therefore, there is no point here in applying the extended formulation. In this case, comparing B-OA and Sepa will give us a feeling of how useful to solve the problem in practice are the refined outer approximation and the inner approximation steps.

Table 5 summarizes the results for solving 4 instances of SSSD with B-OA and Sepa in the same conditions as before. In the table, we report the number of facilities in the instance $|I|$, the number of facilities $|J|$ and for each algorithm, the total CPU time (or the final relative gap if the instance could not be solved), the number of outer approximation iterations and the total number of nodes. The results show a clear improvement induced by the modifications made in Sepa.

		B-OA			Sepa		
		time	#It.	# Nodes	time	#It.	# Nodes
4	15	1.2	6	1496	0.7	**1**	**102**
8	15	24	13	57220	6.8	**1**	**1259**
10	50	191	12	67689	28	**1**	**4435**
20	50	[∞]	1	5.8×10^7	4510	**1**	**2**$.1 \times 10^6$

Table 5: Time, nodes and iterations for SSSD instances.

5 Conclusion

In this paper, we have shown how separability in MINLPs can be used to improve Outer Approximation schemes, both theoretically and in practice.

Although our improvements have been applied to the Outer Approximation decomposition scheme they could as well be applied in the context of an Outer Approximation based branch-and-cut.

As a conclusion and suggestion for extensions, we outline a few ideas to apply similar schemes to more general classes of MINLP. First, note that the transformation we have applied to obtain (sMINLP*) is similar to the classical Mc-Cormick reformulation applied in global optimization and thus similar inner-approximation schemes could be tried for very broad classes of MINLPs. Second, the scheme could be very simply applied to problems with quadratic constraints. It would suffice to use spectral decomposition to make each constraint separable in a similar fashion as Saxena et al. (2009). Of course, separable MINLPs present a very simple structure and we cannot presume that these experimental findings can be generalized to more complex classes of MINLPs, but the positive results we obtained lead us to a certain optimism.

Acknowledgments

First author's research is currently supported by grants: Digiteo Emergence PASO, Digiteo Chair 2009-14D RMNCCO and Digiteo Emergence 2009-55D ARM, a major part of this work was accomplished at Orange Labs R&D. Second author's research was supported by ANR Grant BLAN06-1-0375 and Google Research Award. We thank Jeff Linderoth and Mustafa Kılınç for sharing the UFL and SSSD instances with us.

References

Abhishek, K., S. Leyffer, J.T. Linderoth. 2010. FilMINT: An Outer Approximation-Based Solver for Convex Mixed-Integer Nonlinear Programs. *INFORMS JOURNAL ON COMPUTING* ijoc.1090.0373doi:10.1287/ijoc.1090.0373.

Ben Ameur, W., A. Ouorou. 2006. Mathematical models of the delay constrained routing problem. *Algorithmic Operations Research* **1** 94–103.

Bonami, P., L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, A. Wächter. 2008. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization* **5** 186–204.

Bonami, Pierre, Mustafa Kilinç, Jeff Linderoth. 2009. Algorithms and Software for Convex Mixed Integer Nonlinear Programs. Technical Report #1664, Computer Sciences Department, University of Wisconsin-Madison, 2009.

Duran, M. A., I. Grossmann. 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming* **36** 307–339.

Elhedhli, Samir. 2006. Service System Design with Immobile Servers, Stochastic Demand, and Congestion. *Manufacturing & Service Operations Management* **8** 92–97. doi:10.1287/msom.1050.0094.

Fletcher, R., S. Leyffer. 1994. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming* **66** 327–349.

Grossmann, Ignacio, Jagadisan Viswanathan, Aldo Vecchietti. Ramesh Raman, Erwin Kalvelagen. 2001. GAMS/DICOPT: A discrete continuous optimization package. *Math. Methods Appl. Sci* **11** 649–664.

Günlük, O., J. Lee, R. Weismantel. 2007. MINLP strengthening for separaable convex quadratic transportation-cost UFL. Tech. Rep. RC24213 (W0703-042), IBM Research Division.

Günlük, O., J. Linderoth. 2010. Perspective relaxation of mixed integer nonlinear programs with indicator variables. *Mathematical Programming* **124** 183–205.

Hijazi, Hassan. 2010. Mixed Integer NonLinear Optimization approaches for Network Design in Telecommunications. Ph.D. thesis.

Hijazi, Hassan, Pierre Bonami, Grard Cornujols, Adam Ouorou. 2010. Mixed integer nonlinear programs featuring on/off constraints: convex analysis and applications. *Electronic Notes in Discrete Mathematics* **36** 1153 – 1160. ISCO 2010 - International Symposium on Combinatorial Optimization.

Lougee-Heimer, R. 2003. The common optimization interface for operations research. *IBM Journal of Research and Development* **47** 57–66. `http://www.coin-or.org`.

Quesada, I., I. E. Grossmann. 1992. An LP/NLP based branch–and–bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering* **16** 937–947.

Saxena, Anureet, Pierre Bonami, Jon Lee. 2009. Convex relaxations of non-convex mixed integer quadratically constrained programs: Projected formulations. *Mathematical Programming, Series A* To appear.

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

Wächter, A., L. T. Biegler. 2006. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming* **106** 25–57.