ACoS AIDIC CONFERENCE SERIES

A publication of AIDIC
The Italian Association of Chemical Engineering
Online at: www.aidic.it/cet

VOL. 11, 2013
Chief Editor: Sauro Pierucci

ISBN 978-88-95608-55-6; **ISSN** 2036-5969

DOI: 10.3303/ACOS1311022

The Reformulation-based αGO Algorithm for Solving Nonconvex MINLP Problems – Some Improvements

Andreas Lundell*, Tapio Westerlund

Center of Excellence in Optimization and Systems Engineering,
Åbo Akademi University, Biskopsgatan 8, 20500 TURKU, Finland
andreas.lundell@abo.fi

In this paper, an extension to the *α*-reformulation (*α*R) technique, utilized in the *α* global optimization (*α*GO) algorithm, is presented. The algorithm can be used to solve any nonconvex mixed-integer nonlinear programming (MINLP) problem containing twice-differentiable functions to global optimality as a sequence of reformulated problems overestimating the feasible region of the nonconvex one in a convex extended variable space. In the algorithm, convex underestimators for nonconvex functions are obtained by adding quadratic spline functions overpowering the nonconvexities and subtracting a linearization of the added functions. Here, it is shown how it is possible to reduce the approximation error by utilizing a piecewise quadratic spline function defined on smaller subintervals and in the process increasing the efficiency of the method. This is illustrated by applying the new underestimator on some test problems.

1. Introduction

The *α*GO algorithm can be used to solve any MINLP problem containing nonconvex twice-differentiable functions to global optimality. In contrast to many other global optimization techniques, no branch and bound strategy is used, instead reformulated convex MINLP problems providing an increasing sequence of lower bounds for the original nonconvex problem is solved, *cf.* Lundell and Westerlund (2012a). No upper bounds are needed or considered in this method.
The *α*GO method is based on the signomial global optimization (SGO) algorithm as presented in, *e.g.*, Lundell et al. (2009) and Lundell and Westerlund (2012a), In the SGO algorithm, power and exponential transformations in combination with piecewise linear functions (PLFs) are used for convexifying nonconvex signomial functions. The result is a convex overestimation of the nonconvex problem in an extended variable space. By iteratively tightening the PLF approximations, the overestimation is reduced and the lower bound of the objective function is increased until the solution fulfills all constraints in the nonconvex problem, hence being the global optimum. The SGO algorithm also contains an optimization step for selecting the power and exponential transformations as detailed in Lundell and Westerlund (2007).
The *α* reformulation (*α*R), on which the *α*GO algorithm is based, is a combination of the algorithmic framework of the SGO algorithm and the *α*BB convex underestimator as described in, *e.g.,* Floudas (2000). In the *α*BB underestimator a quadratic function

$$\alpha\left(x-\underline{x}\right)\left(x-\overline{x}\right),\ \ \alpha>0, \tag{1}$$

is added to a nonconvex function *g(x)* on the interval $[\underline{x},\overline{x}]$ to convexify it, in the process underestimating the function in the entire interval since *α* is positive. To guarantee convexity, the value of *α* should be large enough. However, a larger *α*-value results in a less tight convex underestimator, so ideally the smallest possible value should be selected. According to elementary convexity theory, for a univariate function the minimal *α* in the interval $[\underline{x},\overline{x}]$ is found by taking the second derivative of the proposed convex underestimator $g(x)+\alpha\left(x-\underline{x}\right)\left(x-\overline{x}\right)$, *i.e.,*

$$g''(x)+\ 2\alpha \tag{2}$$

and then searching for the minimum positive value *α* fulfilling

Please cite this article as: Lundell A., Westerlund T., 2013, The reformulation-based αsgo algorithm for solving nonconvex minlp problems – some improvements, AIDIC Conference Series, 11, 211-220 DOI: 10.3303/ACOS1311022

$$\alpha \geq -\left(\frac{1}{2}\right) g''(x), \quad \forall x \in [\underline{x}, \overline{x}]. \tag{3}$$

In the multivariate case, the optimal value on α is more difficult to determine exactly and is generally replaced by a valid overestimation. The α-values should be such that they move the eigenvalues of the Hessian matrix of $g(\mathbf{x})$ into the right half of the complex plane, thus making the Hessian of the convexified function positive semidefinite, a requirement for convexity. Methods for determining α-values are described in Floudas (2000). In this paper the scaled Gerschgorin method, based on calculating the interval Hessian matrix and considering the worst-case scenario for the location of the eigenvalues, is used.
In Meyer and Floudas (2005), an extended version of the αBB underestimator utilizing a quadratic spline function, was proposed and this underestimator is used for the reformulations in this paper. The spline underestimator has the benefit of allowing for different α-values in different parts of the domain considered. This paper is an extended version of the conference publication Lundell and Westerlund (2013), where the spline underestimator in a reformulation framework was extended by defining the splines on smaller subintervals.

2. The reformulation technique

The type of MINLP problem considered in this paper is of the following form:

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{x}), \\ \text{subject to} \quad & \underbrace{\mathbf{g}(\mathbf{x}) + \mathbf{q}(\mathbf{x})}_{\mathbf{h}(\mathbf{x})} \leq \mathbf{0} \\ & \mathbf{x} = [x_1, x_2, \ldots, x_N]^T, \ \mathbf{x} \in [\underline{\mathbf{x}}, \overline{\mathbf{x}}]. \end{aligned} \tag{4}$$

where f is a linear or convex nonlinear objective function. The inequality constraints $\mathbf{h} \leq \mathbf{0}$ are composed of twice-differentiable nonconvex functions $\mathbf{g}$ and convex functions $\mathbf{q}$. The variables in $\mathbf{x}$ may be integer- or real-valued and are assumed to be bounded by appropriate explicit lower and upper bounds. Equality constraints are relaxed to corresponding positive and negative inequality constraints. A nonconvex objective function is replaced by a variable μ and an additional constraint $f(\mathbf{x}) - \mu \leq 0$ is included. Thus all nonconvexities are located in the constraints of the problem.
A convex overestimation of the feasible region of the nonconvex problem in Eq. (4) is obtained by replacing the nonconvex functions in the constraints with convex underestimators. The convex relaxation technique is a two-step process, where firstly all functions are convexified by adding functions S convex enough to overpower any nonconvexities to the nonconvex functions. Secondly, the nonconvex functions are underestimated by subtracting a PLF from each S. For the m-th nonconvex constraint this can be formulated as

$$\hat{h}_m(\mathbf{x}) = h_m(\mathbf{x}) + \sum_{i=1}^{N} \left(S_{m,i}(x_i) - \hat{S}_{m,i}\right). \tag{5}$$

If $S_{m,i}$ is convex and $\hat{S}_{m,i}$ is given by a PLF of it, their difference is $S_{m,i} - \hat{S}_{m,i} \leq 0$. Thus the inequality constraint

$$h_m(\mathbf{x}) + \sum_{i=1}^{N} \left(S_{m,i}(x_i) - \hat{S}_{m,i}\right) \leq 0, \quad \text{where} \quad \hat{S}_{m,i} = \mathrm{PLF}\left(S_{m,i}(x_i)\right), \tag{6}$$

will be both a convexified and relaxed constraint.
When replacing the original nonconvex constraints with those in Eq. (6), the result is a convex relaxed MINLP problem in an extended variable space containing the original variables, the variables $\hat{S}$ as well as the variables required for the PLFs. Also, the feasible region of this reformulated problem will contain that of the original nonconvex one.
Initially in Skjäl et al. (2011), the form $S(x) = \alpha x^2$ was proposed for the convexification step. However, in Lundell et al. (2013), the spline version of the αBB underestimator from Meyer and Floudas (2005) was also utilized in the framework. The spline underestimator is a smooth convex piecewise polynomial function of the form

$$S_i(x_i) \begin{cases} \alpha_{i,1} x_i^2 + \beta_{i,1} x_i + \gamma_{i,1}, & \text{if } x_i \in [\omega_{i,1}, \omega_{i,2}], \\ \alpha_{i,2} x_i^2 + \beta_{i,2} x_i + \gamma_{i,2}, & \text{if } x_i \in [\omega_{i,2}, \omega_{i,3}], \\ \quad \vdots & \quad \vdots \\ \alpha_{i,K_i-1} x_i^2 + \beta_{i,K_i-1} x_i + \gamma_{i,K_{i-1}}, & \text{if } x_i \in [\omega_{i,K_i-1}, \omega_{i,K_i}], \end{cases} \tag{7}$$

where $\alpha_{i,k}$, $\beta_{i,k}$ and $\gamma_{i,k}$ are parameters valid in the k-th breakpoint interval of the PLFs of variable x_i, *i.e.*, $[\omega_{i,k}, \omega_{i,k+1}]$, in a specific constraint. The convexity requirement is guaranteed by sufficiently large $\alpha_{i,k}$-values, and the continuity and smoothness of the underestimator is given by the parameters $\beta_{i,k}$ and $\gamma_{i,k}$.

The α-values are calculated for example using the methods presented in Floudas (2000), and the β- and γ-values using the expressions in Meyer and Floudas (2005). In contrast to using the function $S(x) = \alpha x^2$ for convexification, the spline underestimator allows for selecting different α-values in different parts of the domain for the variable. Since α may be zero in convex intervals, the spline function would be linear on such intervals, and the convex underestimator coinciding with the original function, *i.e.,* there would be no underestimation error. Especially when considering nonconvex objective functions, this can greatly reduce the iterations required for finding the global optimum.

3. The αGO algorithm

The αGO algorithm is an extension of the SGO algorithm, as described in, *e.g.,* Lundell et al. (2009) and Lundell and Westerlund (2012a), where the applicable problem type, in addition to signomial functions, now include all nonconvex twice-differentiable functions. In the original SGO algorithm, single-variable power and exponential transformation schemes were used to reformulate nonconvex signomial (including posynomial and polynomial) functions. In the αGO algorithm however, the signomials are regarded as any other twice-differentiable nonconvex function, so no additional transformation schemes are required except for the αR. In Lundell et al. (2013), the αSGO algorithm was introduced, combining the two reformulation techniques. Since it is then possible to transform nonconvex signomials using both the αR as well as the power and exponential transformation schemes, a preprocessing step, selecting an optimized set of transformations for convexifying a given problem, was proposed in Lundell and Westerlund (2012b).
As mentioned earlier, the SGO, αGO and αSGO algorithms share a common framework, where a sequence of reformulated MINLP problems are solved until the global solution to the nonconvex problem is found as the solution to the final subproblem. In each iteration, the overestimation of the feasible region is reduced (in the original variables) by adding breakpoints to the PLFs. Thus the approximation error in the linearization is reduced, as is the overestimation of the feasible region of the reformulated problem. Since the overestimation of the feasible region have a large direct impact on the solution time and number of iterations required, naturally tighter convex underestimators result in a more efficient solution process. This justifies the technique for refining the spline underestimator described in the next section.
It is possible to use the αR technique with or without an iterative procedure such as the αGO algorithm by initially adding a sufficient amount of breakpoints to all PLFs and just solving one or only a few reformulated MINLPs giving the global optimal solution to a specified tolerance. This may be an option for problems with only a few nonconvex functions of a few variables, however, it is often not a viable strategy for medium or large sized problems, since the complexity of the reformulated problem will be too high to be solved within a reasonable time-limit.

4. Refining the spline underestimator

Since the intervals used in the definition of the spline underestimators in Eq. (7) are not connected to those in the PLF approximations of $\hat{S}$ in Eq. (6), it is possible to improve the underestimator by defining the splines over finer intervals. The justification is that when considering smaller intervals, smaller α-values may be obtained due to the function being convex in the interval (resulting in $\alpha = 0$) or since the techniques for obtaining the α-values may give tighter bounds on the parameters due to, *e.g.,* interval arithmetic calculations. An initial partitioning can be done once, and after this, the spline underestimator itself will not be recalculated in subsequent αGO iterations. The default strategy is to calculate the spline underestimator in those intervals defined by the breakpoints in the PLFs, requiring the splines to be recalculated as new breakpoints are added in each iteration. Note however that it is often not enough to calculate the splines to any accuracy to find the global optimum. Instead it is the tightening of the PLFs by adding more breakpoints that guarantees convergence to the global optimum, and thus the approximation of the spline function S will be updated normally by adding additional breakpoints to the PLFs.
Note that, when regarding nonconvex functions that are nonseparable with respect to the variables, the splines must be calculated in hypercubes corresponding to the discretization steps for all variables, and therefore the calculation of the parameters α for the splines are computationally quite costly if a too large number of subintervals are considered. For example, if considering a nonconvex function of two variables with 256 intervals each, the refinement grid will consist of a total of 256 × 256 = 65,536 regions. Therefore, there is a practical limit on how fine the partitioning of the spline parameters should be.
If it is possible to separate the nonconvex functions with respect to the involved variables, individual spline functions can also be used for the individual parts. This can be beneficial for complex multivariate nonconvex functions as it simplifies the calculation of the α's. However, additional transformation variables $\widehat{W}$ are needed although some of the variables used in the PLFs may be reused. Different number of partitions may also be used for different variables.

4.1 An algorithm for calculating the splines

To calculate the splines in N-dimensional subregions the following enumeration algorithm from Westerlund et al. (1995) can be used. The algorithm proceeds through all different combinations of spline intervals for the variables involved in the nonconvex function, only saving the maximum α-value for each interval for each variable. The variables involved in the function $g(x_1,\ldots,x_I)$ are numbered with the index $i = \{1,2,\ldots,I\}$ and the corresponding breakpoint intervals for the variable x_i are numbered with $k_i = \{1,2,\ldots,K_i\}$. The k-th interval for variable x_i is then denoted by $[\cdot]_{i,k}$. The resulting α-value for x_i in the k-th interval is $\alpha(i,k)$.

---

1. Set the index vector $n(i) = 1$ for all $i = \{1,2,\ldots,I\}$. For all valid values, this vector corresponds to a N-dimensional subregion of the domain of the function.
2. Calculate the minimal α-values for each variable x_i in the subregion $[\cdot]_{1,n(1)} \times [\cdot]_{2,n(2)} \times \ldots \times [\cdot]_{I,n(I)}$, corresponding to the index vector n, and save the value for the i-th variabel to $\alpha(i,n(i))$.
3. Set $i = I$, *i.e.*, the total number of variables.
4. If $n(i) + 1 > K_i$
   a. *Set* $n(i) = 1$ and $i = i - 1$.
   b. If $i > 1$ go to 4 else ($i = 0$) go to 5.

   Else
   a. *Set* $n(i) = n(i) + 1$.
   b. Calculate the α-values in the subregion $[\cdot]_{1,n(1)} \times [\cdot]_{2,n(2)} \times \ldots \times [\cdot]_{I,n(I)}$ and save the i-th value to $\alpha'(i, n(i))$. If $\alpha'(i,n(i)) > \alpha(i,n(i))$ save the new value $\alpha'(i,n(i))$ to $\alpha(i,n(i))$ for all i.
   c. Go to 3.
5. Return the vector α.

Note that this algorithm can be easily parallelized by calculating the α-values in even parts of the N-dimensional hypercube on different threads or processors. For example, for a variable x_i the intervals $k_i = \{1, \ldots, K_i / 2\}$ are calculated on one thread and $k_i = \{K_i / 2 + 1, \ldots, K_i\}$ on the other.

4.2 A univariate example

To illustrate the refinement procedure, the reformulation technique is now applied to the function

$$h(x) = x \sin x + x/10, \quad x \in [0,15], \tag{8}$$

assumed to be present in a problem of the type in Eq. (4). A plot of the function is shown in Figure 1. The nonconvex function is replaced with the convex underestimator $h(x) + S(x) - \hat{S}$, where S is a spline function defined as in Eq. (7), and $\hat{S}$ is a PLF of S. If defined on one and two intervals, the spline function will be

$$S(x) = 8.5\,x^2 - 127.5\,x \quad \text{and} \quad S(x) = \begin{cases} 4.75\,x^2 + 85.3125\,x, & 0 \le x \le 7.5, \\ 8.5\,x^2 - 141.563\,x + 210.938, & 7.5 < x \le 15, \end{cases} \tag{9}$$

respectively. So, if the spline is defined on one interval only, we get a variant of the original αBB underestimator, but if instead two intervals are used, a smaller value for α can be used in the first interval, resulting in a tighter underestimator. In Figure 1, the spline functions and their approximations, in the case when the PLF-linearizations are performed in four intervals, are illustrated for different refinement levels of the spline functions. An illustration of the α-values obtained if smaller subintervals are considered for the spline is provided in Figure 2. In the case of only one interval, the largest α-value most be used on the entire interval to guarantee convexity. However, if smaller intervals are considered, different values for the parameter can be utilized. In intervals where the function is convex even zero values are allowed, resulting in the convex underestimator coinciding with the original function, *i.e.,* no underestimation error occurs. In Figure 3, it is illustrated how the underestimator changes as additional breakpoints are added to the PLF approximations.

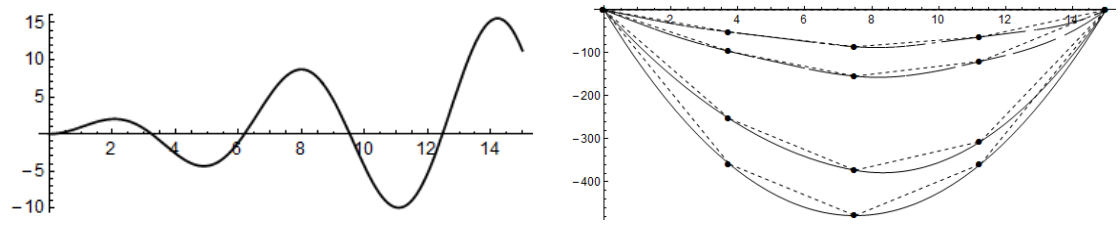

Figure 1: Left: The nonconvex function h(x). Right: The spline functions (solid) and their linearizations (dashed) when the α-values are calculated on one (furthest down), two, 10 and 50 equal subintervals respectively. Five breakpoints are used in each PLF.

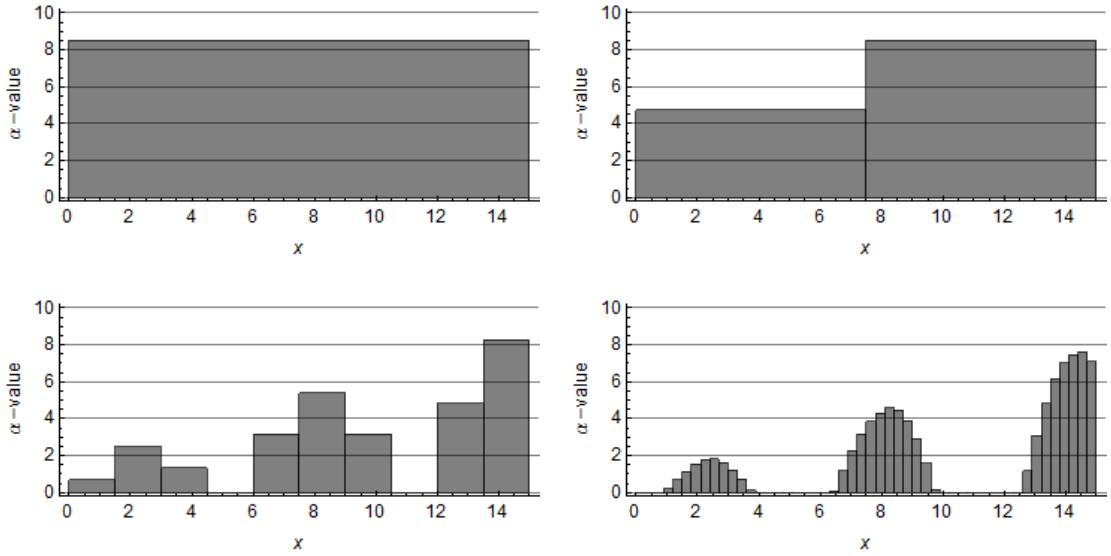

Figure 2: The α-values in the different parts of the domain, when calculating them on finer intervals using the scaled Gerschgorin method. In the initial figure, the same value is used in the whole domain [0,15], but as the grid is made finer, smaller values can be used in the separate intervals. Note the α-values equal to zero in intervals where the function h(x) is convex.

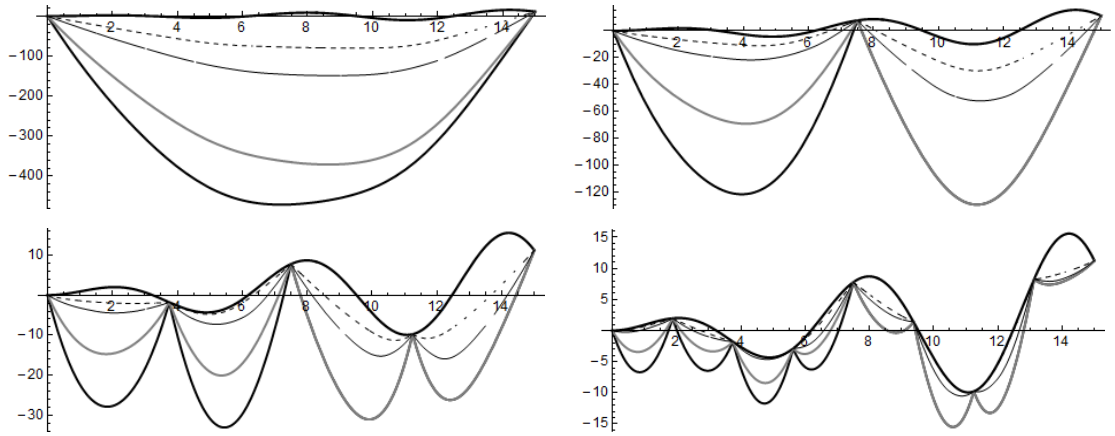

Figure 3: The nonconvex function h(x) as well as the resulting convex underestimators calculated on one (thick), two (grey), 10 (thin) and 50 (dashed) subintervals of equal length with no additional breakpoints (top, left), one additional breakpoint (top, right), three additional breakpoints (bottom, left) and seven additional breakpoints (bottom, right) in the PLFs. Note that the breakpoints used in the PLFs are independent of those in the splines.

4.3 A bivariate example

Now, we consider the following nonconvex MINLP problem:

$$\begin{aligned} \text{minimize} \quad & f(x_1,x_2) = (2\,x_1-4)^2 + (x_2-13/2)^2 \\ \text{subject to} \quad & h(x_1,x_2) = \underbrace{x_1\cos^2 x_2 + x_2\sin^2 x_1 - 3/x_2}_{g(x_1,x_2)} + \underbrace{x_1/2 - 5/2}_{q(x_1)} \le 0, \\ & 2 \le x_1 \le 4,\ 2 \le x_2 \le 8,\ x_1 \in \mathbb{R},\ x_2 \in \mathbb{Z}. \end{aligned} \tag{4}$$

The objective function f is convex. The nonconvex constraint consist of a nonconvex function $g(x_1,x_2)$ and a convex function $q(x_1)$. The variable x_1 is real and x_2 is integer. Since the nonconvex function is a non-separable function of two variables we consider it as a whole, and reformulate it using the quadratic spline underestimator described in Section 4. The result is the following problem:

$$\begin{aligned} \text{minimize} \quad & (\ _1,\ _2) = (2\ \ _1-4)^2 + (\ _2-13/2)^2 \\ \text{subject to} \quad & x_1\cos^2 x_2 + x_2\sin^2 x_1 - 3/x_2 + x_1/2 - 5/2 + S_1(x_1) + S_2(x_2) - \hat{S}_1 - \hat{S}_2 \le 0, \\ & \hat{S}_1 = \mathrm{PLF}\big(S_1(x_1)\big),\ \hat{S}_2 = \mathrm{PLF}\big(S_2(x_1)\big), \\ & 2 \le x_1 \le 4,\ 2 \le x_2 \le 8,\ x_1 \in \mathbb{R},\ x_2 \in \mathbb{Z}. \end{aligned}$$

The reformulated problem is convex in the extended variable space consisting of the original variables x_1 and x_2, as well as those variables needed for the PLFs. The nonconvex function $h(x_1,x_2)$ and the integer-relaxed feasible region of $h(x_1,x_2) \le 0$ are shown in Figure 4.

To illustrate the benefits of defining the spline underestimator on smaller subintervals, a comparison when solving the problem using the αGO algorithm is given for the cases of two and 32 subintervals. The algorithm is implemented in the General Algebraic Modeling System (GAMS), where the reformulated problems were solved using the convex MINLP solver alphaECP. The computer used had a quad core Intel i7 2.8 GHz processor. Since the function $h(x_1,x_2)$ is nonseparable, the splines will be calculated on 2 × 2 = 4 and 32 × 32 = 1,024 regions respectively in the two cases. The spline calculations are performed once, *i.e.,* the spline function itself is not updated in subsequent iterations. The times for calculating the splines is 0.9 s and 1.2 s in the two cases including initialization and calculation of the splines using Wolfram Mathematica. However, if we would calculate the splines on 1,024 subintervals for each variable, 1,048,576 regions would need to be calculated, which takes over 180 s. Increasing the partitioning above 32 for each variable does not seem to have a significant impact on the solution time of the problems in the individual iterations in GAMS.

In Figures 5 and 6, the iterative procedure of the αGO algorithm is exemplified for partitioning levels two and 32. In the first case, seven iterations are required for obtaining the global solution. In the second case, however, only four iterations are required due to the fact that the spline underestimators S_1 and S_2, as well as their linearizations $\hat{S}_1$ and $\hat{S}_1$ are much tighter than in the first case. This leads to a smaller overestimation of the feasible region of the reformulated problem in the second case in each iteration. A comparison of solving the problem to global optimality with different levels of partitioning is given in Figure 7.

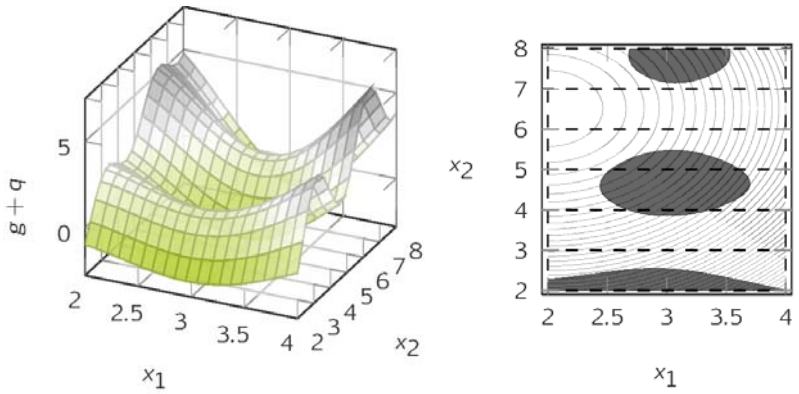

Figure 4. Left: The function in the nonconvex constraint $h(x_1,x_2)$. Right: The integer-relaxed feasible region of the constraint $h(x_1,x_2) \le 0$ (gray region) as well as the contour of the objective function $f(x_1,x_2)$.

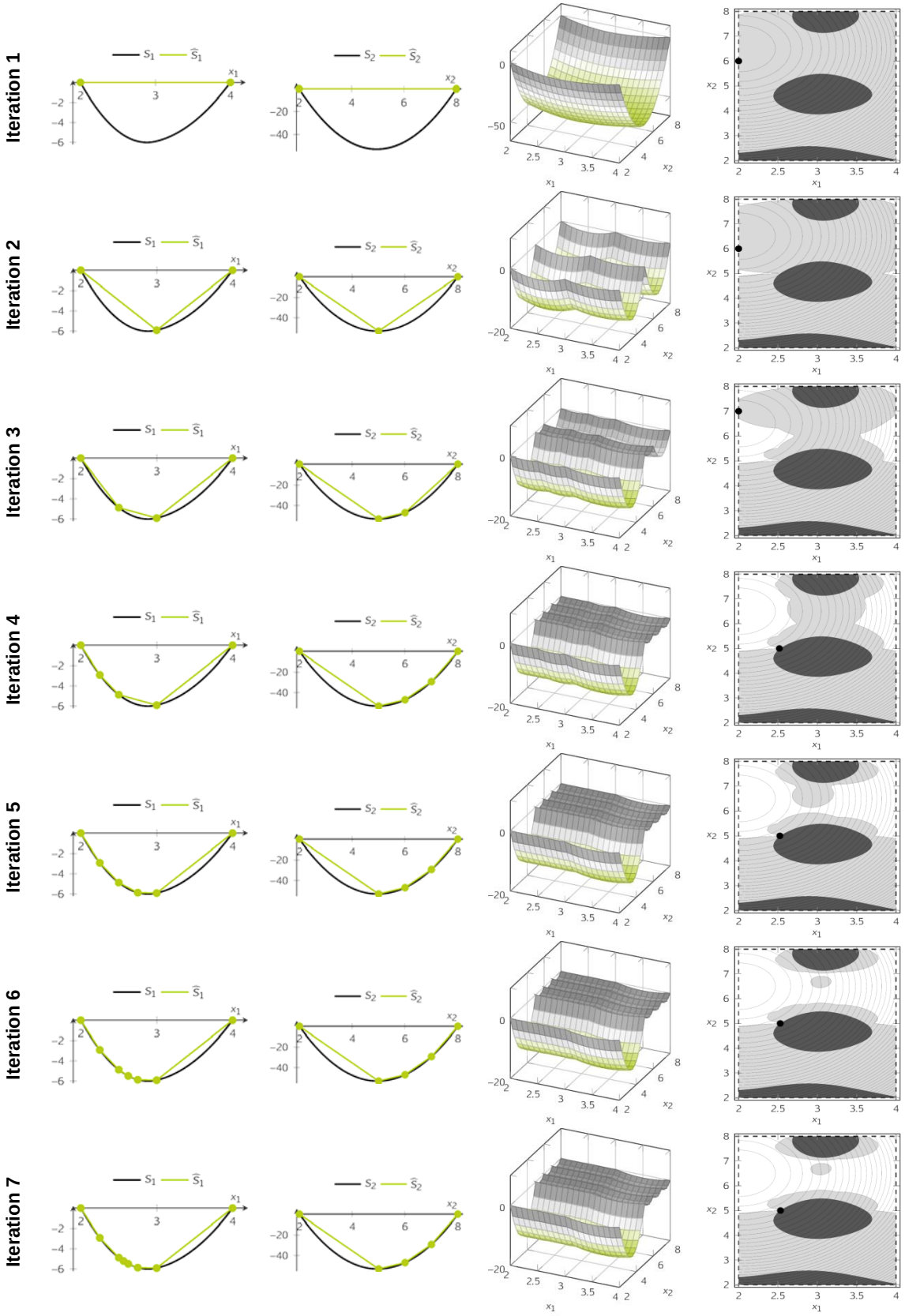

Figure 5. An illustration of the reformulated problems in each iteration of the αGO algorithm when calculating the splines on 2 × 2 = 4 subregions. The first two columns show the spline functions and their linearizations, the third column the piecewise convex underestimator and the fourth the overestimation of the feasible region in the reformulated problem, as well as the solution point of the reformulated problem in each iteration.

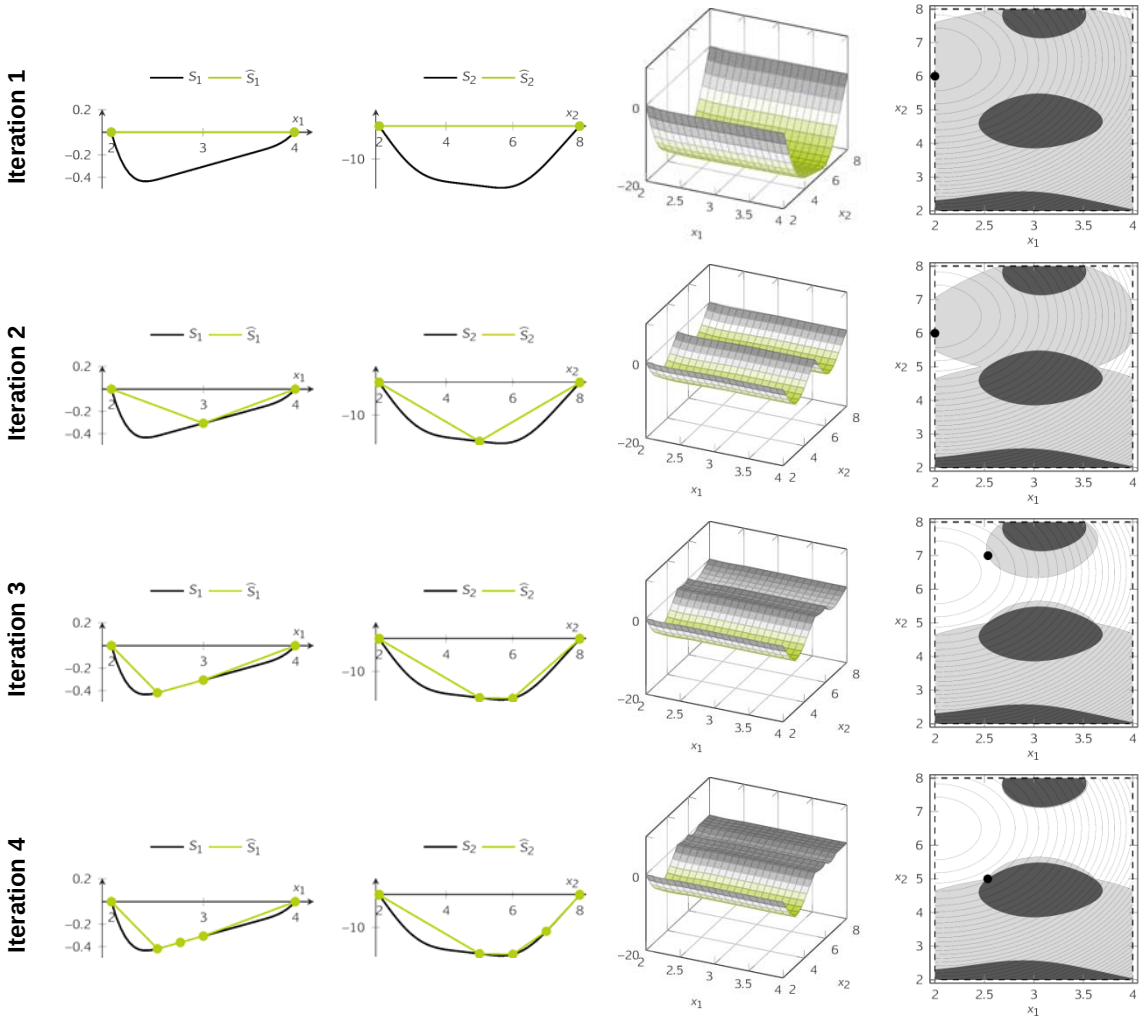

Figure 6. An illustration of the reformulated problems in each iteration of the αGO algorithm when calculating the splines on 32 × 32 = 1,024 subregions. The first two columns show the spline functions and their linearizations, the third column the piecewise convex underestimator and the fourth the overestimation of the feasible region in the reformulated problem, as well as the solution point of the reformulated problem in each iteration.

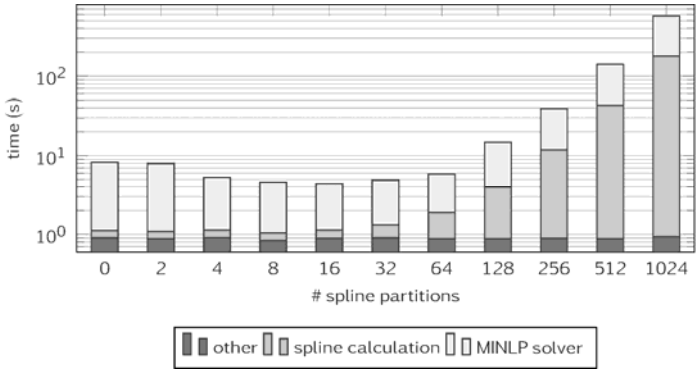

Figure 7. A comparison of the impact on the number of spline partitions on the solution time of the problem in Section 4.3. Note that the scale on the y-axis is logarithmic.

4.4 Some test problems

In this section, the refinement technique is applied to three nonconvex test problems from Floudas and Pardalos (1999) and the results are given in Table 1. Note that these problems are the same as in the paper Lundell and Westerlund (2013), however an error in the implementation gave the wrong results there. The *α*GO algorithm is used to solve the problems, and the spline calculations are performed once, *i.e.,* the spline function itself is not updated in subsequent iterations. The computer used for the comparisons had a quad core Intel i7 2.8 GHz processor. The spline calculations were done with the scaled Gerschgorin method implemented in Wolfram Mathematica and for solving the reformulated MINLP problems, the GAMS/SBB solver was used. Refining the splines further gave tighter lower bounds in the first *α*GO iteration of all the problems, as is clear from the results, and often also less iterations were required to solve the problem to optimum when increasing the refinement level. A drawback is, however, that the calculations of the parameters for the splines become computationally more demanding as the refinement grid is increased. This is especially evident in problems of more than one variable, where the variables are nonseparable, since the number of boxes the parameters need to be calculated in the product of the number of subintervals for all variables. Therefore, there is a trade-off between the number of subintervals and the resulting number of *α*GO iterations as can be seen in Table 1. The increase in grid points for the splines did not seem to affect the solution time of the reformulated MINLP problem significantly as also can be seen from the table.

Table 1: Results from the comparisons described in Section 4.4. Initial LB is the solution to the MINLP problem in the first iteration. The times for calculating the α-, β- and γ-values in the spline (with Wolfram Mathematica), solving the MINLP subproblems with GAMS/SBB (including compilation), as well as the total solution times are given. The instances were solved to the global optimal solution indicated for each problem. However, for Problem 8.2.6 the instances indicated with (-) were prematurely terminated at a time-limit of 3,600 s.

	Problem 8.2.1					**Problem 8.2.2**				
Spline intervals	*α*GO iters	Initial LB	Spline time (s)	GAMS time (s)	Total time (s)	*α*GO iters	Initial LB	Spline time (s)	GAMS time (s)	Total time (s)
1	12	-15.8	0.9	15.2	17.2	53	-762.2	1.0	93.7	95.8
2	12	-13.6	0.9	12.0	14.0	47	-539.0	1.0	66.9	69.0
4	8	-6.2	1.0	6.3	8.4	32	-271.4	1.0	33.3	35.3
8	7	-4.7	1.0	3.8	5.8	29	-170.8	1.0	27.5	29.6
16	6	-4.1	1.0	3.0	5.1	20	-95.9	1.0	15.2	17.2
32	6	-3.9	1.2	3.1	5.4	10	-51.3	1.0	5.0	7.1
64	6	-3.7	1.7	3.3	6.0	8	-32.0	1.0	4.2	6.3
128	6	-3.7	3.7	3.1	7.9	8	-26.2	1.0	3.6	5.8
256	6	-3.6	11.9	3.1	16.1	7	-23.8	1.1	3.1	5.4
512	6	-3.6	43.7	3.3	48.0	7	-22.7	1.4	3.1	5.5
Variables	Reals: 2, 2 transformed					Reals: 1, 1 transformed				
Glob.opt.	-2.02					-1.08				

	Problem 8.2.6				
Spline intervals	*α*GO iters	Initial LB	Spline time (s)	GAMS time (s)	Total time (s)
1	-	-2.2E7	0.9	-	-
2	-	-4.2E6	1.0	-	-
4	-	-4.2E5	1.0	-	-
8	-	-1.3E5	1.0	-	-
16	-	-1.6E4	1.1	-	-
32	36	-7.0E3	1.3	3577.7	3580.1
64	23	-2.3E3	2.4	193.6	197.0
128	13	-1.0E3	6.4	39.9	47.3
256	11	-6.1E2	21.4	22.8	45.2
512	10	-4.5E2	82.1	20.3	103.4
Variables	Reals: 2, 2 transformed				
Glob.opt.	-10.09				

5. Conclusions

In this paper, it was shown how the solution process of the *α*GO algorithm could be improved by defining the spline convex underestimator on a finer grid than the regular iteratively added breakpoints used in the algorithm. The technique works very well for functions where the nonconvex functions are separable. For nonseparable functions however, the refinement grid cannot be too fine since the computational effort required to calculate the spline underestimator increases. Therefore, the trade-off between the number of subintervals and the resulting number of *α*GO iterations need to be considered.

Acknowledgements

Support from the Foundation of Åbo Akademi University, as part of the grant for the Center of Excellence in Optimization and Systems Engineering, is gratefully acknowledged.

References

Floudas C., Pardalos P., 1999, Handbook of test problems in local and global optimization. Nonconvex optimization and its applications, Kluwer Academic Publishers, Dordrecht, the Netherlands.

Floudas, C.A., 2000, Deterministic Global Optimization. Theory, Methods and Applications, Number 37 in Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Dordrecht, the Netherlands.

Lundell A., Skjäl A., Westerlund T., 2013, A reformulation framework for global optimization, Journal of Global Optimization 57(1),115-141. doi: 10.1007/s10898-012-9877-4.

Lundell A., Westerlund J., Westerlund, T., 2009, Some transformation techniques with applications in global optimization, Journal of Global Optimization 43(2), 391-405. doi: 10.1007/s10898-007-9223-4.

Lundell A., Westerlund T., 2007, Optimization of power transformations in global optimization, Chemical Engineering Transactions 11, 95-100.

Lundell A., Westerlund T., 2012a, Global optimization of mixed-integer signomial programming problems, In Lee, J., Leyffer, S., eds, Mixed Integer Nonlinear Programming. Volume 154 of The IMA Volumes in Mathematics and its Applications, Springer, New York, 349-369.

Lundell A., Westerlund T., 2012b, Finding an optimized set of transformations for convexifying nonconvex MINLP problems. In Karimi, I., Srinivasan, R., eds.: Proceedings of the 11th International Symposium on Process Systems Engineering, Singapore, 1497-1501.

Lundell A., Westerlund T., 2013, The reformulation-based *α*GO algorithm for solving nonconvex MINLP problems – some improvements, Chemical Engineering Transactions 32, 1321-1326. doi: 10.3303/CET1332221.

Meyer C.A., Floudas C.A., 2005, Convex underestimation of twice continuously differentiable functions by piecewise quadratic perturbation: Spline *α*BB underestimators, Journal of Global Optimization 32(2), 221-258.

Skjäl A., Lundell A., Westerlund T., 2011, Global optimization of C^2 constraints by convex reformulations, Chemical Engineering Transactions 24, 373-378. doi: 10.3303/CET1124063.

Westerlund T., Isaksson J. and Harjunkoski, I., 1995, Solving a production optimization problem in the paper industry, Report 95-146-A, Process Design Laboratory, Åbo Akademi University.