CHEMICAL ENGINEERING TRANSACTIONS

VOL. 32, 2013

A publication of AIDIC

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

Chief Editors: Sauro Pierucci, Jiří J. Klemeš

ISBN 978-88-95608-23-5; **ISSN** 1974-9791

DOI: 10.3303/CET1332221

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

The α-reformulation (αR) technique can be used to transform any nonconvex twice-differentiable mixed-integer nonlinear programming problem to a convex relaxed form. By adding a quadratic function to the nonconvex function it is possible to convexify it, and by subtracting a piecewise linearization of the added function a convex underestimator will be obtained. This reformulation technique is implemented in the α global optimization (αGO) algorithm solving the specified problem type to global optimality as a sequence of reformulated subproblems where the piecewise linear functions are refined in each step. The tightness of the underestimator has a large impact on the efficiency of the solution process, and in this paper it is shown how it is possible to reduce the approximation error by utilizing a piecewise quadratic spline function defined on smaller subintervals. The improved underestimator is also applied to test problems illustrating its performance.

1. Introduction

The α global optimization (αGO) algorithm is a method for solving mixed-integer nonlinear programming (MINLP) problems containing nonconvex twice-differentiable functions to global optimality. Instead of utilizing a branch and bound strategy, it is based on formulating and iteratively solving reformulated convex MINLP problems providing an increasing sequence of lower bounds for the original nonconvex problem, *cf.* Lundell and Westerlund (2012a). No upper bounds need to be calculated in this method.
The nonconvex functions are convexified and underestimated using a technique called α reformulation (αR). The αR is based on the αBB convex underestimator as described in, e.g., Floudas (2000), where the quadratic function

$$g(x) + \alpha(x-\underline{x})(x-\overline{x}) \tag{1}$$

is used to convexify and underestimate the nonconvex function g on the interval $[\underline{x},\overline{x}]$. This function underestimates g in the entire interval since α is positive. To guarantee convexity, there is a positive lower limit on the value of α. A larger α-value results in a less tight convex underestimator, so ideally the smallest possible value should be selected. For a univariate function, the minimal α in the interval $[\underline{x},\overline{x}]$, is found by taking the second derivative of the function in (1), i.e.,

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

and then searching for the minimum positive value α fulfilling

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

In the multivariate case, the limit value on α is more difficult to obtain and is generally replaced by a valid overestimation. Some methods to obtain such α -values are described in Floudas (2000). In this paper, the scaled Gerschgorin method is used. Newer versions of the αBB underestimator have also been presented, forxample, in Meyer and Floudas (2005) a version utilizing a quadratic spline function was proposed and

Please cite this article as: Lundell A., Westerlund T., 2013, The reformulation-based αgo algorithm for solving nonconvex minlp problems – some improvements, Chemical Engineering Transactions, 32, 1321-1326 DOI: 10.3303/CET1332221

this underestimator is used for the reformulations in this paper. The spline underestimator has the benefit of allowing for different *α*-values in the domain considered.

2. The reformulation technique

The *α*R, as described in this paper, is applicable to the following type of problem

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

where *f* is a linear, convex nonlinear or nonconvex nonlinear objective function. The inequality constraints **h** ≤ **0** are composed of twice-differentiable nonconvex functions **g** and convex functions **q**. The variables in **x** may be integer- or real-valued, and are assumed to be bounded by appropriate explicit lower and upper bounds. Equality constraints are also allowed, but are relaxed to corresponding positive and negative inequality constraints. A nonconvex objective function is replaced by a variable *μ* and an additional constraint *f(**x**)* – *μ* ≤ 0 is included.

The nonconvex problem in Eq (4) is convexified and relaxed by replacing the nonconvex functions in the constraints with convex underestimators, thus obtaining a convex overestimation of the original (nonconvex) feasible set and an underestimation of a nonconvex objective function, if present. The relaxation technique is a two-step process. In the first step, all functions are convexified by adding functions *S* convex enough to overpower any nonconvexities to the nonconvex functions. In a second step, the nonconvex functions are underestimated by subtracting a piecewise linear function from each *S*. For the *m*-th nonconvex constraint this can be formulated as

$$h_m(\mathbf{x}) = h_m(\mathbf{x}) + \ \sum_{i=1}^{N} \left(S_{m,i}(x_i) - \widehat{W}_{m,i}\right). \quad (5)$$

Since $S_{m,i}$ is convex and $\widehat{W}_{m,i}$ a piecewise linear function of it, $S_{m,i} - \widehat{W}_{m,i} \leq 0$, and

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

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 $\mathbf{x}$ 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. However, in Lundell and Westerlund (2012a), the spline version of the *α*BB underestimator from Meyer and Floudas (2005) was utilized in the framework. The spline underestimator is a smooth convex piecewise polynomial function of the form

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

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). This underestimator has the possibility of using different α-values in different parts of the domain for the variable, whereas the original formulation required the largest α-value over the entire domain to be selected for each variable.

3. The *α*GO algorithm

The *α*GO algorithm is a further development of the signomial global optimization (SGO) algorithm as described in e.g., Lundell et al. (2009) and Lundell and Westerlund (2012a). In the 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. (2012), 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).
The αGO, αSGO and SGO 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 PLF approximations. The overestimation of the feasible region have a large impact on the solution time and number of iterations required, so therefore tighter convex underestimators result in a more efficient solution process. This justifies the technique for refining the spline underestimator introduced in this paper.
It is also possible to use the reformulation technique without an iterative procedure such as the αGO algorithm by initially adding a sufficient amount of breakpoints to all PLFs and just solving one reformulated MINLP giving the global optimal solution to a specified tolerance. However, this 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 $\widehat{W}$ 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 technique for obtaining the α-values gives tighter bounds on the parameters. An initial partitioning can be done once, and after this, the spline underestimator itself will not be recalculated in subsequent αGO iterations. The normal 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. The approximation $\widehat{W}$ of the spline function S will however 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, β and γ 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 1,024 intervals each, the refinement grid will normally consist of a total of 1,024 ×1,024 = 1,048,576 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.

4.1 An illustrative example

To illustrate the refinement procedure for the splines, the reformulation technique is now applied to the nonconvex trigonometric function

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

assumed to be present in an inequality constraint in a problem of type Eq 4). The function is shown in Figure 1. The nonconvex function is now replaced with the reformulated variant $h(x) + S(x) - \widehat{W}$, where S is a spline function defined as in Eq (7), and $\widehat{W}$ is a piecewise linear approximation of S. If defined on one and two intervals, the spline function will be

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

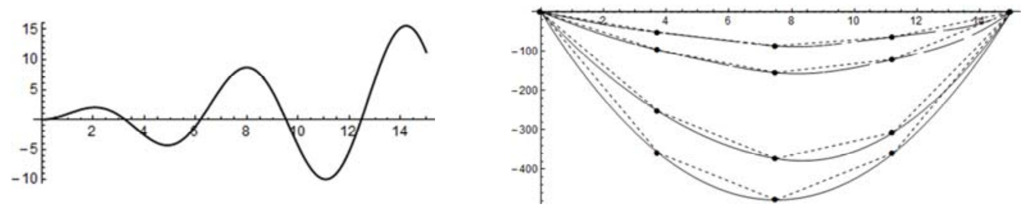

Figure 1: Left: The nonconvex function h(x). Right: The spline functions (solid) and their linearizations (dashed) when the α-values are calculated on one (furthest down), two, 10 and 50 equal subintervals.

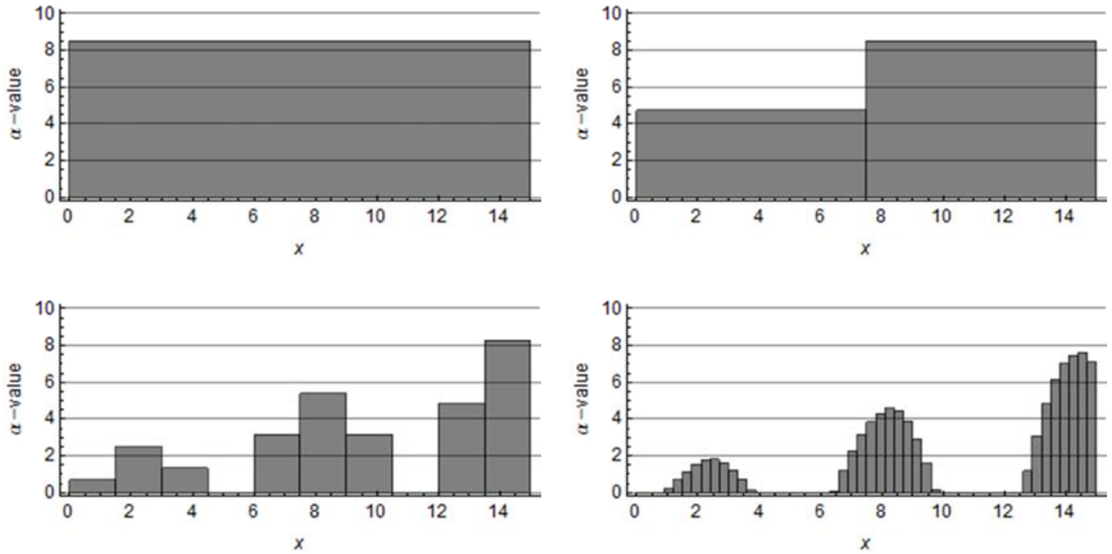

Figure 2: The α-values in the different parts of the domain, when calculating them on finer intervals using the scaled Herschgorin method. In the initial figure, the same value is used in the whole domain [0,15], but as the grid is made finer, smaller values can be used in the separate intervals. Note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

respectively. So, if the spline is defined on one interval only, we get a variant of the original *α*BB 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 is 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. If only one interval is considered, 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.

4.2 Some test problems
In Table 1, results from applying the refinement technique to three nonconvex problems (Problems 8.2.1, 8.2.2 and 8.2.6) from Floudas and Pardalos (1999), as well as, one (Synthes3) from the MINLP Library (http://www.gamsworld.org/minlp/) are given. 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 and GAMS/SBB was used for solving the reformulated convex MINLP subproblems. Refining the splines further gave

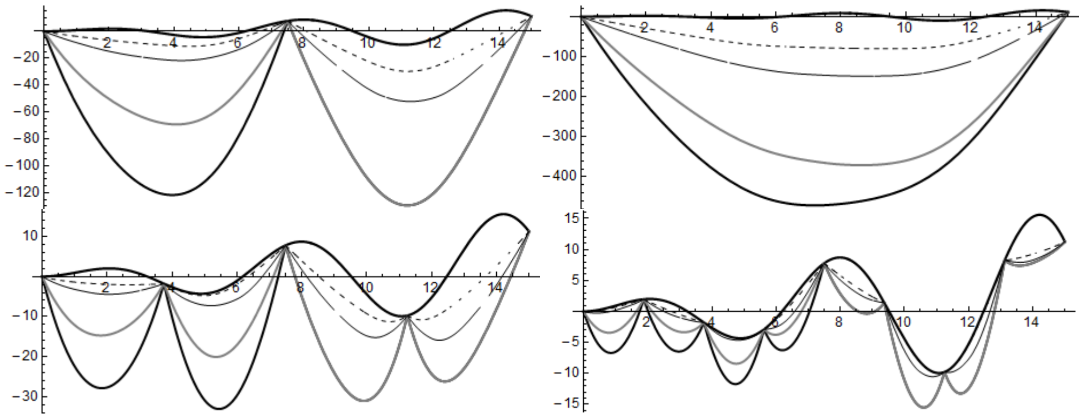

Figure 3: The nonconvex function $h(x) = x \sin x + x/10$ (thick) as well as the resulting convex spline underestimators calculated on one (thick), two (grey), 10 (thin) and 50 (dashed) subintervals of equal length with no additional breakpoints (top, left), one additional breakpoint (top, right), three 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.

Table 1: Results from the comparisons described in Section 4.2. 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, as well as the total time 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 86,400 s.

	Problem 8.2.1					**Problem 8.2.2**				
Spline intervals	αGO iters	Initial LB	Spline time (s)	GAMS time (s)	Total time (s)	αGO iters	Initial LB	Spline time (s)	Solution time (s)	Total time (s)
1	12	-15.8	<0.1	4.3	4.4	53	-762.2	<0.1	22.2	22.4
2	12	-14.4	<0.1	3.9	4.0	47	-539.0	<0.1	18.1	18.2
4	9	-6.8	<0.1	2.3	2.4	32	-271.4	<0.1	10.0	10.1
8	8	-5.3	<0.1	1.8	1.9	29	-171.4	<0.1	8.4	8.5
16	7	-4.7	0.2	1.6	1.9	20	-104.1	<0.1	5.4	5.5
32	7	-4.4	0.8	1.6	2.5	15	-71.3	0.1	3.7	3.9
64	7	-4.2	3.0	1.6	4.7	11	-57.1	0.2	2.9	3.2
128	7	-4.2	11.8	1.6	13.5	11	-50.7	0.3	2.8	3.2
256	7	-4.1	47.7	1.8	49.6	11	-47.7	0.6	3.0	3.7
512	7	-4.1	190.7	2.3	193.2	7	-46.3	1.2	3.4	4.7
Variables	(reals/integers): 2/0, 2 transformed					(reals/integers): 1/0, 1 transformed				
Glob.opt.	-2.02					-1.08				

	Synthes3					**Problem 8.2.6**				
Spline intervals	αGO iters	Initial LB	Spline time (s)	Solution time (s)	Total time (s)	αGO iters	Initial LB	Spline time (s)	Solution time (s)	Total time (s)
1	8	39.8	<0.1	9.3	9.4	-	-2.2E7	<0.1	-	-
2	8	50.8	<0.1	7.8	7.9	-	-4.2E6	<0.1	-	-
4	7	59.2	<0.1	5.5	5.6	-	-4.2E5	<0.1	-	-
8	5	62.1	<0.1	3.5	3.6	-	-1.3E5	0.1	-	-
16	5	63.0	0.2	3.7	3.9	63	-1.6E4	0.4	1000.9	1001.3
32	5	63.6	0.7	3.1	3.9	37	-7.0E3	1.4	129.5	131.0
64	5	63.8	2.4	3.3	5.8	22	-2.3E3	5.6	30.8	36.4
128	5	64.0	9.5	3.4	13.0	13	-1.0E3	23.0	8.7	31.8
256	5	64.1	37.3	4.0	41.4	11	-6.2E2	88.4	5.7	94.2
512	5	64.1	147.6	4.2	152.0	10	-4.5E2	358.2	6.9	365.4
Variables	(reals/integers): 9/8, 6 transformed					(reals/integers): 2/0, 2 transformed				
Glob.opt.	68.01					-10.09				

tighter lower bounds in the first αGO iterations of all the problems, as is clear from the results, and often also less iterations were required to solve the problem to optimum if increasing the refinement level. However, the calculations of the parameters for the splines become more computationally demanding as the refinement grid is increased. This is especially evident in problems of more than one variable and where the variables are nonseparable, since the number of boxes the parameters need to be calculated in is in the worst case 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 αGO iterations. The increase in grid points for the splines did not seem to affect the solution time of reformulated MINLP problem significantly.

5. Conclusions

In this paper, it was shown how the solution process of the αGO 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 more complex functions however, the refinement grid cannot be too fine due to the computational effort required to calculate the spline underestimator.

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., 2012, A reformulation framework for global optimization, Journal of Global Optimization, 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

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

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

Skjäl A., Lundell A., Westerlund T., 2011, Global optimization of C2 constraints by convex reformulations, Chemical Engineering Transactions 24, 373-378. DOI: 10.3303/CET1124063