CHEMICAL ENGINEERING TRANSACTIONS Volume 24, 2011
Editor Sauro Pierucci

DOI: 10.3303/CET1124063

Global Optimization with C^2 Constraints by Convex Reformulations

Anders Skjäl, Andreas Lundell, Tapio Westerlund*

Process Design & Systems Engineering, Division of Natural Sciences and Technology, Åbo Akademi University
Biskopsgatan 8, FIN-20500 Turku, Finland, Tapio.Westerlund@abo.fi

The well-known αBB method solves very general smooth nonconvex optimization problems. The algorithm works by replacing nonconvex functions with convex underestimators. The approximations are improved by branching and bounding until global optimality is achieved. Applications are abundant in engineering and science. We present a convex formulation in which the underestimators are improved without directly splitting the domain in a branch-and-bound tree. We show two illustrative examples and discuss some possible gains and drawbacks with the algorithm.

1. Introduction

Twice-differentiable constraints are in general not convex. Therefore, most optimization algorithms cannot guarantee that the global optimum will be found. Many global optimization algorithms are based on convexifying the constraints and objective function. The solution to the modified problem minimizes an underestimate of the objective function (if it is nonconvex) on an overestimation of the feasible set.
The αBB algorithm described in Adjiman et al. (1998a,b) is a general-purpose algorithm for problems with twice continuously differentiable (C^2) constraints and objective function. Any nonconvex part is convexified by adding parabolas $\alpha(x-x_L)(x-x_U)$ in each variable. Sufficiently large α values are readily computed. One of the methods proposed in Adjiman et al. (1998a) uses interval arithmetic and Gershgorin's circle theorem. Finding the best possible (smallest) α values is in general hard.
We have studied the question whether the same type of underestimators can be used in a framework without branching over the domain. We have found constraint formulations to achieve this, and the basic variety is presented in this paper. The algorithm described here was designed with the possibility of solving mixed-integer problems in mind.

2. The Algorithm

The algorithm utilizes (continuous) piecewise linear functions (PLF) in a manner similar to the SGO algorithm in Lundell (2009). A future project would be to integrate this new approach with the SGO algorithm. PLFs are sometimes possible to model with continuous variables, especially if the PLF is convex and a natural part of the minimization trade-off (cf. inventory costs and marginal prices in Skjäl et al., 2009). This is not the case here and we have to introduce binary variables to describe the PLFs.

Please cite this article as: Skjal A., Lundell A. and Westerlund T., 2011, Global optimization of c^2 constraints by convex reformulations, Chemical Engineering Transactions, 24, 373-378
DOI: 10.3303/CET1124063

2.1 Constraints

The formulation is presented here in one dimension for simplicity. The extension to multiple dimensions is intuitive and addressed at the end of the subsection. Minimization of a nonconvex objective function $f(x)$ can be modeled as minimization of μ subject to the constraint $f(x) - \mu \leq 0$. The only difference to other nonconvex constraints is the need for slightly different termination criteria. Therefore we focus the description on constraints.

With given breakpoints $x_0, x_1, \ldots, x_K$ ($x_0 = x_L, x_K = x_U$) the nonconvex constraint $g(x) \leq 0$ will give rise to the following constraints in our model:

$$g(x) + \alpha(x - c_1)(x - c_2) - W \leq 0 \tag{1}$$

$$W = \sum_{k=1}^{K} A_{k-1} b_k + (A_k - A_{k-1}) s_k \tag{2}$$

$$x = \sum_{k=1}^{K} x_{k-1} b_k + (x_k - x_{k-1}) s_k \tag{3}$$

$$s_k \leq b_k, \ k = 1,2,\ldots,K \tag{4}$$

$$\sum_{k=1}^{K} b_k = 1 \tag{5}$$

b_k binary, s_k positive, $k = 1,2,\ldots,K$

The constants $A_0, A_1, \ldots, A_K$ are the values of $\alpha(x\text{-}c_1)(x\text{-}c_2)$ in $x_0, x_1, \ldots, x_K$ respectively. The integer relaxation of this model is convex. In (1) the two first terms are similar to the classical αBB convex underestimator and convex by the same argument. We are free to choose the constants c_1 and c_2, one obvious choice would be $c_1 = c_2 = 0$. The third term in (1) and the rest of the constraints are linear.

For every interval $[x_{k-1}, x_k]$ the formulation tries to reproduce the classical αBB underestimator in that interval by removing a linearization of $\alpha(x\text{-}c_1)(x\text{-}c_2)$. The new underestimator when (2)-(5) holds will achieve this except for one detail; α is the same for every interval. A branch-and-bound algorithm studies the intervals separately and can use a smaller α in some intervals where the function g is "less nonconvex". In our formulation we must choose the largest calculated α or we lose convexity in some intervals. A comparison is shown in Fig.1.

It should be noted that the constraints (3)-(5) need only be included once, even if there are many convexified constraints. This "sharing" of the PLFs could plausibly be further exploited by integrating the method with the SGO algorithm for signomial constraints.

In a real situation, the problem would be multi-dimensional. The constraints above are modified by adding a dimension index to (x, α, s, b, A) and summing over this index where appropriate. W does not need to be split; all linearizations can be added in one constraint.

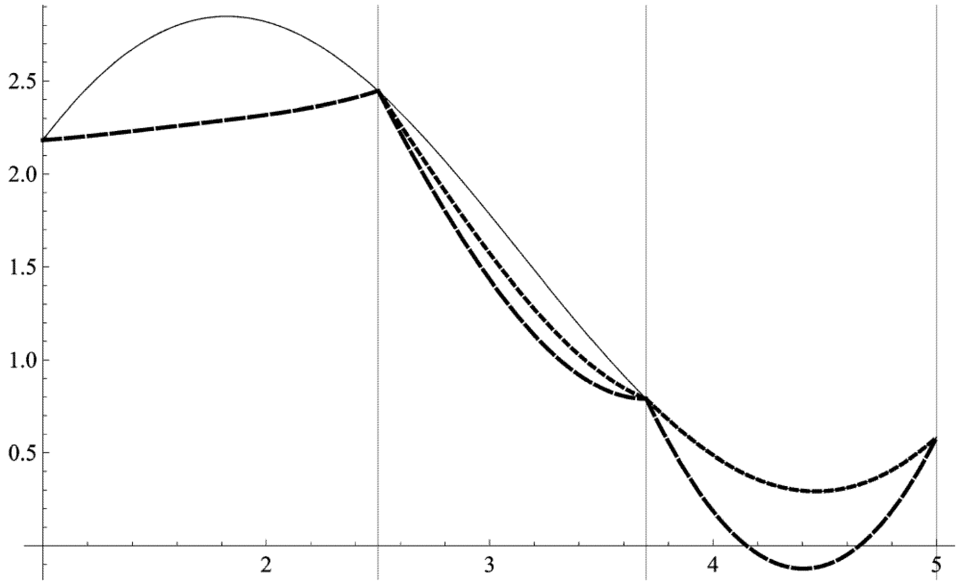

Figure 1: The difference between the classical αBB (short dashes, $\alpha = 1; 0.6; 0$) and the underestimator described here (long dashes, $\alpha = 1$). Note that the classical αBB is used in a branch-and-bound algorithm, one interval at a time.

Other formulations of the same idea are possible. The PLFs can be described as SOS2 (special ordered sets with at most two adjacent variables $\neq 0$) or equality in (2) can be replaced by inequality ($\leq$).

2.2 Iterations

After each iteration, the following tests are done:

- If no solution was found, terminate. The original problem is infeasible.
- If the solution is infeasible (for the original problem), add breakpoints according to some rule and continue with the next iteration.
- If the solution is ε-feasible and the objective function has not been underestimated, terminate. The point is an ε-global solution point.
- If the solution is ε-feasible but the objective function is underestimated, calculate the underestimation gap. If it is less than the specified tolerance, terminate. Otherwise add more breakpoints and continue with the next iteration.

There are many ways to choose the new breakpoints. Lundell (2009) discusses the effects of adding a solution point coordinate, an interval midpoint or the point with the largest underestimation error in the SGO algorithm. The two latter points coincide for our underestimator. Branch-and-bound algorithms usually split domains in two in every step. The algorithm described here solves subproblems of increasing complexity. It seems likely that adding several breakpoints is more effective in this situation. The default strategy employed is to add the midpoint of the solution interval in each dimension. For discrete variables the breakpoints are chosen as integers.

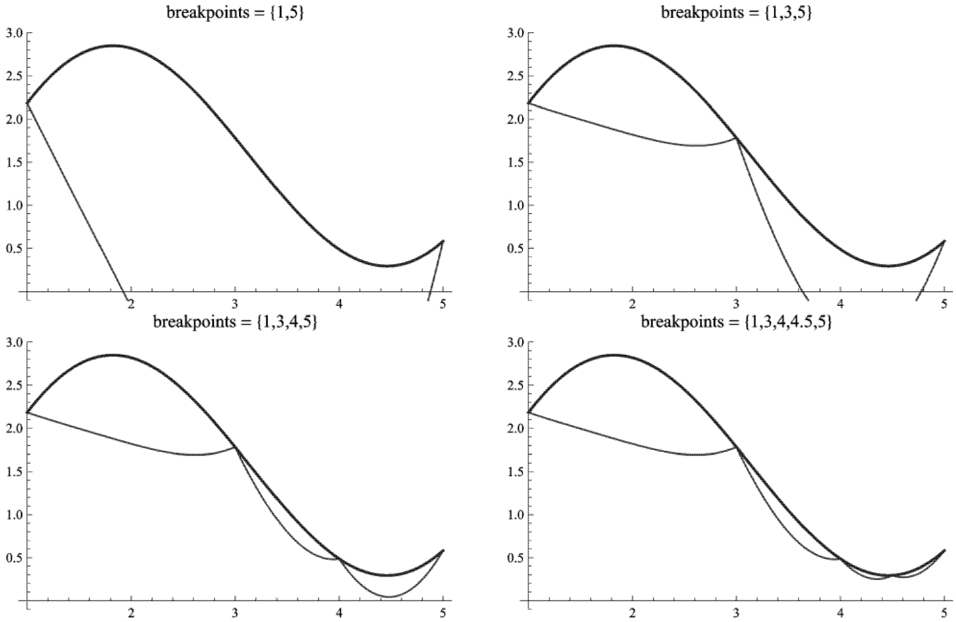

Figure 2: The underestimator of $2\sin(x) + x/2$, $1 \le x \le 5$ *when one new breakpoint is added in each step, in the middle of the interval where the minimum is found.*

3. Two Examples

The underestimation of $2\sin(x) + x/2$, $1 \le x \le 5$ at different levels of refinement is shown in Fig. 2. The new breakpoints are added in the interval where the minimum of the function is found. This corresponds to the behavior when the function is used as the objective. If the function is part of a constraint, say $2\sin(x) + x/2 \ge 5/2$, the choice of breakpoints may be different. If the solution from the second iteration is 2.8 the constraint is not satisfied and the next new breakpoint will be 2, i.e. the midpoint of the interval [1,3].

Consider the following problem in two variables:

$$\text{minimize} \quad (y+3)^2 + e^x + y - x \tag{6}$$

$$\text{subject to} \quad (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \le 100 \tag{7}$$

$$-4 \le x \le 4, \quad -4 \le y \le 4 \tag{8}$$

$$x \text{ integer}$$

The iteration steps for solving this problem are shown in Fig. 3. As x is an integer variable, the x components of the solutions and the new breakpoints in the x direction are all integer. The optimal solution (-2, -3) is found in four iterations. In mixed-integer problems it sometimes happens that the solution "jumps" into the feasible set after a finite number of steps. Another common situation is a solution sequence converging to a feasible point. The process terminates when a found solution is ε-feasible.

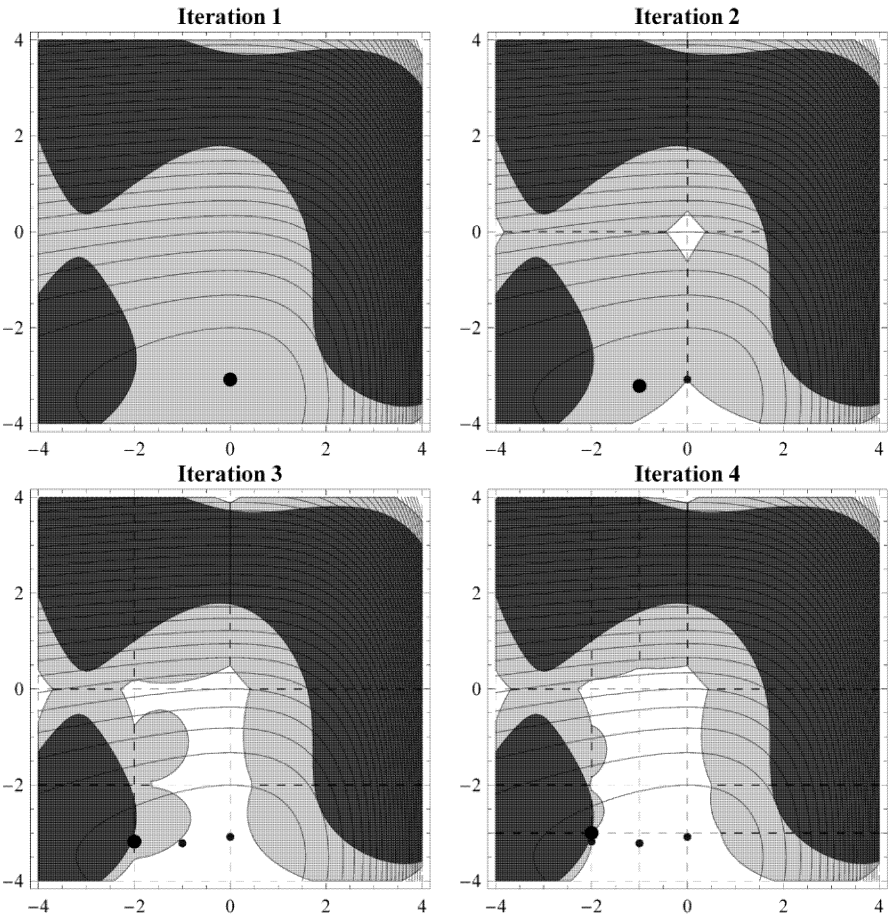

Figure 3: The iteration steps taken when the second example is solved; the level curves of the objective function are visible; the dark areas are the feasible set of the integer-relaxed problem; the lighter area is the overestimated feasible set in the current iteration. The current solution is indicated with the large black point.

4. Discussion and Further Research

Note that branching cannot be avoided altogether, even in the continuous case. The algorithm in this paper merely embeds it in the convex MINLP subproblems and passes it on to the subsolver. A major difference between a branch-and-bound algorithm like αBB and our "non-branching" algorithm is the information acquired during the solution process.

The αBB algorithm knows a lower bound for each studied node. When a feasible solution is found it provides an upper bound on the global minimum. Any branch with a lower bound higher than the upper bound is fathomed and cut from the tree. Nodes with a lower bound below the best found upper bound could contain a better solution and must be branched further. In contrast, the non-branching algorithm obtains lower bounds only for the whole domain. Once a feasible solution is found it is also globally optimal and the algorithm terminates.

Meyer and Floudas (2005) constructed an underestimator by fitting classical αBB underestimators together to form a continuously differentiable spline. The basic variety of this underestimator is applicable also in our framework and lets us relax the requirement of a uniform α in a given dimension. The effect in one dimension would be an underestimator coinciding with the classical αBB on each interval. In multiple dimensions we can choose α based on an interval “slice” along the other dimensions. The extra calculations needed are almost negligible, the improvement is problem-specific.

Akrotirianakis and Floudas (2004a,b) studied an underestimator created with expressions of exponential functions instead of parabolas. This underestimator could also be tried out in our framework. In general, any convex underestimator created by adding univariate convex negative functions could be used.

Acknowledgements

Financial support from the Center of Excellence in Optimization and Systems Engineering at Åbo Akademi University is gratefully acknowledged.

References

Adjiman C.S., Dallwig S., Floudas C.A. and Neumaier A., 1998a, A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical advances, Computers & Chemical Engineering, 22, 1137-1158.

Adjiman C.S., Androulakis I.P. and Floudas C.A, 1998b, A global optimization method, αBB, for general twice-differentiable constrained NLPs – II. Implementation and computational results, Computers & Chemical Engineering, 22, 1159-1179.

Akrotirianakis I.G. and Floudas C.A., 2004a, A New Class of Improved Convex Underestimators for Twice Continuously Differentiable Constrained NLPs, Journal of Global Optimization, 30, 367-390.

Akrotirianakis I.G. and Floudas C.A., 2004b, Computational Experience with a New Class of Convex Underestimators: Box-constrained NLP Problems, Journal of Global Optimization, 29, 249-264.

Lundell A., 2009, Transformation Techniques for Signomial Functions in Global Optimization, PhD Thesis, Åbo Akademi University.

Meyer C. and Floudas C.A., 2005, Convex Underestimation of Twice Continuously Differentiable Functions by Piecewise Quadratic Perturbation: Spline αBB Underestimators, Journal of Global Optimization, 32, 221-258.

Skjäl A., Björk K.-M., Östman J. and Carlsson C., 2009, An Optimization Model for Tactical Planning of Wood Procurement, 42nd Hawaii International Conference on System Sciences.