Global Optimization and Singular Nonlinear Programs: New Techniques

Julie Roy and R. Baker Kearfott

January 23, 2010

General Context and Notation

The general problem we are considering is

$$\begin{aligned} &\text{minimize } \varphi(x) \\ &\text{subject to } c_i(x) = 0,\ i = 1, \ldots, m_1, \\ &\qquad\qquad g_i(x) \le 0,\ i = 1, \ldots, m_2, \end{aligned}$$

where the objective function $\varphi(x) : \boldsymbol{x} \to \mathbb{R}$ and the constraints $c_i, g_i : \boldsymbol{x} \to \mathbb{R}$ are possibly nonlinear, and $\boldsymbol{x} \in \mathbb{R}^n$ is the box where $x_i \in [\underline{x}_i, \overline{x}_i]$ for $i = 1, \ldots, n$ defines the search region in a branch and bound algorithm. The constraints in the problem are active constraints at a point if they hold with equality; for example, an inequality constraint g_i $(i = 1, \ldots, m_2)$ is active at a feasible point x if $g_i(x) = 0$. An inequality constraint g_i is considered to be approximately active if $|g_i(x)| \le \epsilon_g$ for some tolerance $\epsilon_g > 0$, and it is considered to be approximately inactive if $g_i(x) < -\epsilon_g$.

In problems posed in operations research and other fields, lines, planes, hyperplanes, or hypersurfaces that are feasible (or approximately feasible) and that have approximately optimal objective function values often occur. For these problems, traditional software usually finds one approximately optimal point without any indication more solutions exist. Software with rigorous search or automatic result verification fails to complete by taking excessive amounts of time in the branch and bound process, and by returning considerable numbers of small boxes that possibly contain the solution.

Our interest is in computing rigorous enclosures of all of the approximately feasible, approximately optimal points. To do this we first need to define "approximate singular solution set" in a way that it is practical to rigorously enclose.

Approximate Singular Solution Sets

Main Idea

Let $\check{x}$ be a feasible point such that $\varphi(\check{x})$ is approximately optimal. If there are directions from $\check{x}$ in which the objective function φ and the active constraints do not change much, we can construct a region $\boldsymbol{\alpha}$ based on these directions, within which φ is guaranteed to be within some tolerance ϵ_φ of $\varphi(\check{x})$ and within which the constraints are guaranteed to be within some tolerance of feasible. These skewed approximate solution boxes do not necessarily contain all points that are within ϵ_φ of optimal, but by construction, all points in the boxes are within ϵ_φ of optimal.

Rejection of Adjacent Regions

To incorporate the skewed ϵ-approximate solution boxes in a branch and bound process, we also can construct boxes adjacent to these solution boxes within which either the constraints are guaranteed to be infeasible or φ is guaranteed to be greater $\varphi(\check{x}) + \epsilon$, where ϵ can be chosen to be sufficiently less than ϵ_φ for the rejection process to work smoothly.

An illustration of an enclosed ϵ-approximate skewed solution box $\boldsymbol{\alpha}$ and adjacent rejection regions is given in Figure 1. In this illustration, $X^{(C)}$ represents the box in the original coordinate system that contains the skewed box $\boldsymbol{\alpha}$. The direction V represents the direction in which the objective function and active constraints are not changing much, and the direction W represents the direction in which either the objective function value increases or one of the constraints becomes infeasible.

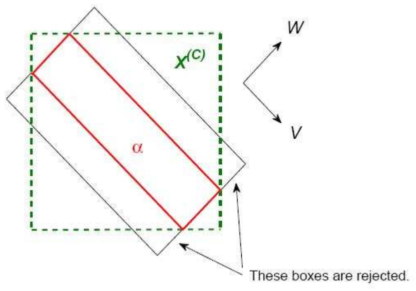

Figure 1: Illustration of enclosed ϵ-approximate solution box and adjacent rejection regions

Finding the Directions Parallel to the Solution Set

Let $\check{x}$ be an approximately feasible point where the objective function φ is approximately optimal. Let n_a be the number of approximately active inequality constraints at $\check{x}$. Reorder the inequality constraints so that $\{g_i\}_{i=1}^{n_\mathrm{a}}$ are the approximately active inequality constraints at $\check{x}$ and $\{g_i\}_{i=n_\mathrm{a}+1}^{m_2}$ are the approximately inactive inequality constraints at $\check{x}$. We choose directions in the null space of

$$G = \begin{pmatrix} \nabla^T \varphi(\check{x}) \\ \nabla^T c_1(\check{x}) \\ \vdots \\ \nabla^T c_{m_1}(\check{x}) \\ \nabla^T g_1(\check{x}) \\ \vdots \\ \nabla^T g_{n_\mathrm{a}}(\check{x}) \\ \nabla^2 \varphi(\check{x}) \end{pmatrix}.$$

This matrix G is a matrix whose first $m_1 + n_\mathrm{a} + 1$ rows are the transposes of the gradients of the objective function and the active constraints evaluated at $\check{x}$ and remaining n rows consist of the Hessian matrix $\nabla^2\varphi$ evaluated at $\check{x}$. To obtain an orthonormal basis for the null space of G, we can use a singular value decomposition of the matrix (such as with Matlab's `svd` function).

We initially include all of the active g_i $(i = 1, \dots, n_\mathrm{a})$, but if many of the inequality constraints are active, the singular value decomposition of the matrix G may not reveal any directions in which the points remain optimal. In other words, there may not be any lines passing through $\check{x}$ in which the points remain optimal in both directions along the lines. For example, traveling in one direction along the line may correspond to traveling out of the feasible region. This does not mean there are not any other optimal points near $\check{x}$. There may be lines passing through $\check{x}$ along which points remain optimal in only one direction (or along a ray pointing into the feasible region from $\check{x}$). To find these directions, we can look at the singular value decomposition of each reduced matrix G_i $(i = 1, \dots, n_\mathrm{a})$ constructed so that G_i is the matrix G with the i^{th} active inequality constraint gradient removed. For each of these matrices G_i, let V_i denote the matrix of vectors pointing into the feasible region that correspond to the smallest singular values in the singular value decomposition of G_i. The V_i vectors that we need to consider are those which correspond to approximately zero singular values in the singular value decompositions of the respective G_i matrices. The direction in which we can travel the farthest into the feasible region is the direction associated with the active inequality constraint g_i whose gradient we can remove from the G matrix. Therefore, we possibly can use the matrix G_i instead of G to find an approximately singular direction. This process can be continued by computing further reduced matrices corresponding to the deletion of additional active inequality constraint gradients from the matrix G.

In practice, there are some "tuning" considerations. The "approximate null space" is determined by a tolerance that determines when a singular value of G can be ignored as zero. Also ϵ_φ and ϵ in the definition of the approximate solution set affect the method. If they are set too small with inaccuracies in the coefficients of the problem, approximate null sets will not be detected. If they are set too large, the entire search region will be marked, and the result will be meaningless. The appropriateness of the settings depends on the problem.

Examples

A Simple Linear Example

An investment company needs to find out how to invest \$200,000 in four stocks with the expected rates of return and measures of risk given Table 1. The

Table 1: Expected rates of return and risk for the four stocks

stock	A	B	C	D
price per share	\$100	\$50	\$80	\$40
return per share	0.12	0.08	0.06	0.10
risk measure per dollar	0.10	0.07	0.05	0.08

company wants to minimize the risk subject to the following conditions: the annual rate of return must be at least 9%, and no one stock can account for more than 50% of the total investment. This gives the linear program:

$$\begin{array}{ll} \text{Minimize} & 10A + 3.5B + 4C + 3.2D \\ \text{Subject to:} & \\ & 100A + 50B + 80C + 40D \le 200,000, \\ & 12A + 4B + 4.8C + 4D \ge 18,000, \\ & 0 \le 100A \le 100,000, \\ & 0 \le 50B \le 100,000, \\ & 0 \le 80C \le 100,000, \\ & 0 \le 40D \le 100,000. \end{array}$$

Any point along the portion of the line given parametrically by

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \approx \begin{pmatrix} 666.667 \\ 0 \\ 0 \\ 2500 \end{pmatrix} + t \begin{pmatrix} -0.3714 \\ 0 \\ 0.9285 \\ 0 \end{pmatrix},$$

with $0 \le t \le 897.46$, is a solution to this problem. A graph of this solution set is given in Figure 2. The computations for this solution set were done almost instantaneously using Matlab and INTLAB (for information about INTLAB, see [3] and [2]).

An Approximately Singular Linear Example

The techniques described above also can be applied to problems that are only approximately singular. For example, if the coefficients of the previous problem are perturbed slightly, we can still compute an approximate solution set that is similar to the approximate solution set for the exactly singular problem. To illustrate this, consider the linear program:

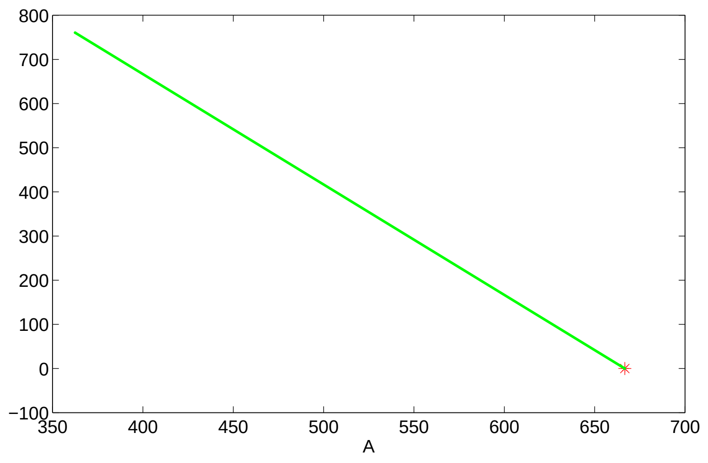

Figure 2: ϵ-approximate solution set for the simple linear example

$$
\begin{array}{ll}
\text{Minimize} & 9.98205404139819A + 3.5B + 4.08C + 3.2D \\
\text{Subject to:} & \\
& 100A + 50B + 80C + 40D \leq 200,000, \\
& 12.0833808879827A + 4B + 4.88C + 4D \geq 18,000, \\
& 0 \leq 100A \leq 100000, \\
& 0 \leq 50B \leq 100000, \\
& 0 \leq 80C \leq 100000, \\
& 0 \leq 40D \leq 100000.
\end{array}
$$

This problem was created by taking a very small random perturbation of both the return per share and the risk measure per dollar for stock A. An approximately optimal solution to this problem occurs when $A \approx 662.066$, $B \approx 0$, $C \approx 0$, and $D \approx 2500$ which is close to one of the solutions to the original problem.

This approximately singular problem has a set of approximate solutions along a portion of the line given parametrically by

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \approx \begin{pmatrix} 662.066 \\ 0 \\ 0 \\ 2500 \end{pmatrix} + t \begin{pmatrix} \text{-}0.3703 \\ 0 \\ 0.9289 \\ 0 \end{pmatrix},$$

for $0 \leq t \leq 52.17$ (where $\epsilon_\varphi = 0.1$). This is a somewhat smaller solution set that is close to the solution set of the exactly singular problem,

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \approx \begin{pmatrix} 666.667 \\ 0 \\ 0 \\ 2500 \end{pmatrix} + t \begin{pmatrix} \text{-}0.3714 \\ 0 \\ 0.9285 \\ 0 \end{pmatrix},$$

for $0 \leq t \leq 897.46$.

An Example from the Standard Netlib Test Problems

The techniques described above for determining an ϵ-approximate solution set were applied to the AFIRO Netlib test problem [1]. This problem has 32 variables and 59 constraints (19 inequality constraints, 8 equality constraints, 32 boundary constraints). One feature of this problem is that many of the constraints are linearly dependent. An approximately feasible, approximately optimal solution set computed using Matlab and INTLAB with $\epsilon_\varphi = 1.0$ is

$$\begin{pmatrix} 80 \\ 25.5 \\ 54.5 \\ 84.8 \\ 63.597 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 18.2143 \\ 45.3827 \\ 67.4128 \\ 500 \\ 475.92 \\ 24.08 \\ 0 \\ 215 \\ 125.3111 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 339.9429 \\ 258.6318 \\ 53.8838 \\ 0 \end{pmatrix} + t \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -0.0494 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.0494 \\ -0.0523 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.6739 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.6739 \\ 0.2898 \\ 0 \end{pmatrix} + s \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -0.5637 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.5637 \\ -0.5975 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -0.0590 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0.0590 \\ -0.0254 \\ 0 \end{pmatrix}$$

for $-128.1484 \leq t \leq 256.1484$ and $-8.1674 \leq s \leq 8.1674$.

Enclosing Nonlinear Solutions

For solution sets that are nonlinear, we can find a "tube of solutions," that is, a chain of skewed boxes in different coordinate systems that contain ϵ-approximate optimal points. To do this, we first construct an ϵ-approximate solution box around an initial approximately optimal point by the previously mentioned methods. If the approximate null space has dimension 1, we can use a predictor-corrector type method to construct an approximate solution curve and a set of enclosing boxes. More specifically, we can consider points along the faces of the initial approximate solution box in the directions in which this box is wide, and then use a floating point minimization process to minimize φ along each of these faces of the box. We then can construct new ϵ-optimal solution boxes about these points. This process is continued until either no more new boxes can be constructed or a maximum number of iterations is obtained.

Some Illustrative Nonlinear Examples

The following two examples are simple unconstrained nonlinear examples. These examples are provided to illustrate the method described for finding the enclosures to nonlinear solution sets. Computations for the approximately optimal solution sets only took minutes using Matlab and INTLAB, and would probably be one or two orders of magnitude faster using a compiled language, where there is less overhead in nested loops. The computations would probably take much longer with a general branch and bound algorithm without these techniques.

1. Minimize $\varphi(x) = (\sin(x_1) - x_2)^2$

   The global minimum is $\overline{\varphi} = 0$. All points on the curve $x_2 = \sin(x_1)$ are exact optimal solutions to this problem. A "tube" of approximately feasible, approximately optimal solutions was computed using an initial optimal, feasible point $\check{x} = [2\pi, 0]^T \approx [6.28318530717959, 0]^T$. A graph of the results is given in Figure 3. The central curve represents the exact solution set for this problem (the curve $x_2 = \sin(x_1)$). The two curves on each side of the exact solution set represent upper and lower bounds for all points within ϵ_φ of the exact solution set. The inner boxes containing the central curve represent sets of points that are within $\epsilon_\varphi \approx 0.1$ of optimal (i.e., for all x in the skewed box, $\varphi(x) \leq \overline{\varphi} + \epsilon_\varphi$ where $\overline{\varphi} = 0$ is an upper bound for the global minimum). These skewed boxes do not contain all points that are within ϵ_φ of optimal, but by construction, all points in the boxes are within ϵ_φ of optimal. The outer boxes are guaranteed to have no other solutions other than the central one in them. These outer boxes are boxes that could be rejected in the branch and bound process.

2. Minimize $\varphi(x) = (x_1^2 - 3x_2)^2$

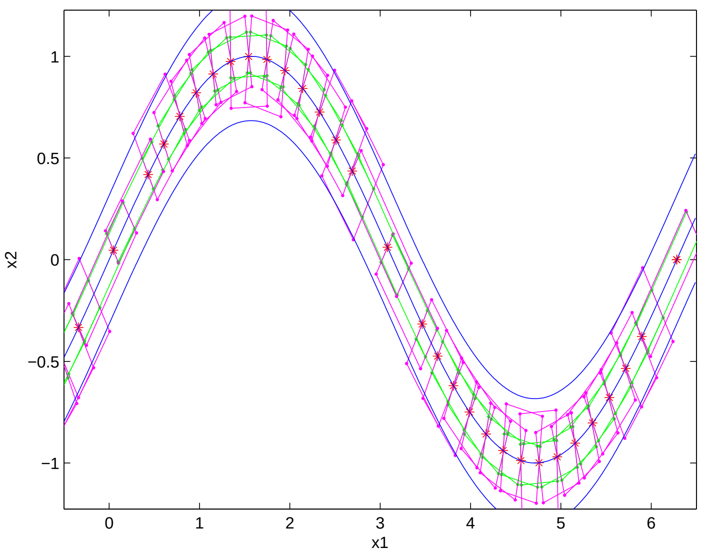

Figure 3: $\varphi(x) = (\sin(x_1) - x_2)^2$

The global minimum is $\overline{\varphi} = 0$. All points on the parabola $x_2 = x_1^2/3$ are exact optimal solutions for this problem. Using an initial optimal, feasible point $\check{x} = [3, 3]^T$, the "tube" of approximately feasible, approximately optimal solutions is shown in Figure 4 and in Figure 5. As in the previous example, the curves represent the exact solution set and the upper and lower bounds for the ϵ-approximate set where $\epsilon \approx 0.1$. The inner boxes are boxes constructed to be guaranteed to lie within the ϵ-approximate set where $\epsilon \approx 0.1$. The outer boxes are guaranteed to have no other solutions other than the central one in them.

Future Work

Presently, we are investigating heuristics to best determine the dimension of the approximate optimizing set, and also to reject regions adjacent to the approximate optimizing set. The goal is to incorporate these techniques into a branch and bound method for global optimization.

References

[1] *Netlib Repository*, `http://www.netlib.org/lp/data/afiro`.

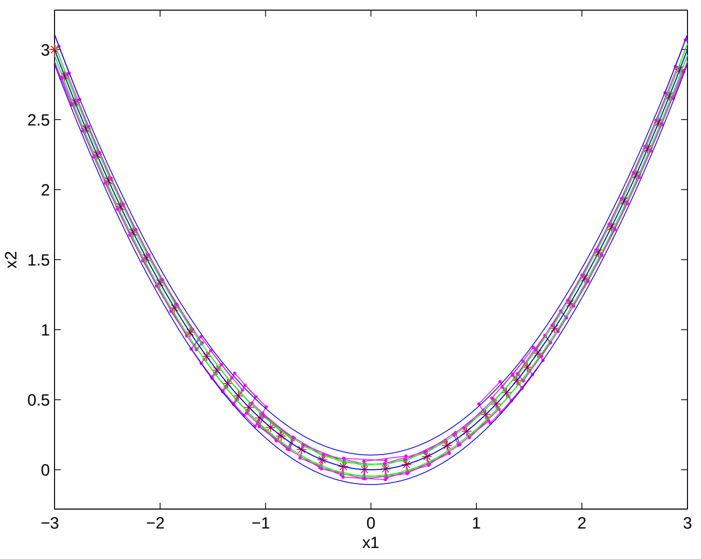

Figure 4: $\varphi(x) = (x_1^2 - 3x_2)^2$

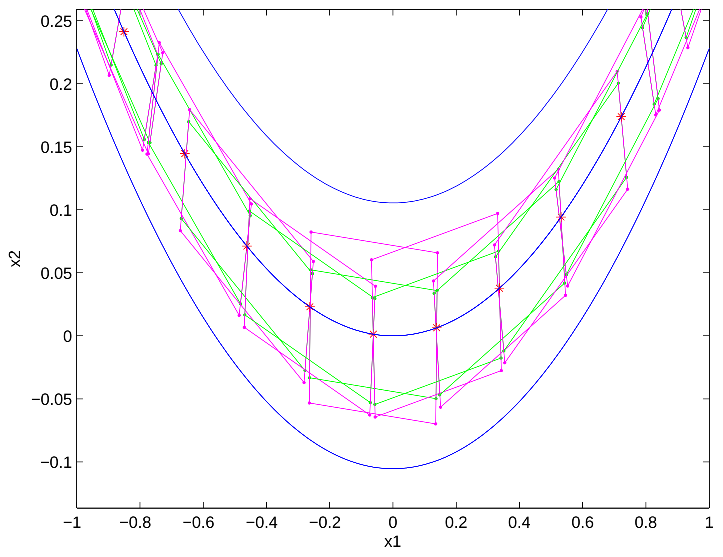

Figure 5: $\varphi(x) = (x_1^2 - 3x_2)^2$ – a closer look

[2] S.M. Rump, *INTLAB - INTerval LABoratory website*, `http://www.ti3.tu-harburg.de/rump/intlab/`.

[3] S.M. Rump, *INTLAB - INTerval LABoratory*, Developments in Reliable Computing (Tibor Csendes, ed.), Kluwer Academic Publishers, Dordrecht, 1999, `http://www.ti3.tu-harburg.de/rump/`, pp. 77–104.