

Global optimization and constraint satisfaction

Arnold Neumaier

University of Vienna

Vienna, Austria

In a competitive world, only the best (safest, cheapest, fastest, ...) is good enough.

This is why optimization (and often global optimization) is very frequent in application.

Global optimization is one of the oldest of sciences, part of the art of successful living.

maximize	service (or money? or happiness?)
s.t.	gifts and abilities
	hopes and expectations (ours; others)
	bounded stress

Global optimization is one of the oldest of sciences, part of the art of successful living.

maximize	service (or money? or happiness?)
s.t.	gifts and abilities
	hopes and expectations (ours; others)
	bounded stress

We all know from experience that the optimum of a constrained problem is typically on the boundary.

Especially the last constraint is all too often active

Thousands of years of experience resulted in the following algorithmic framework recommended by St. Paul (ca. 50 AD):

“Consider everything. Keep the good.
Avoid evil whenever you recognize it.”
(*1 Thess. 5:21–22*)

In modern terms, this reads:

Do global search by branch and bound!

My personal global optimization problem is a never ending challenge:

“Be perfect,
as our father in heaven is perfect.”
(Jesus, ca. AD 30)

On the mathematical level, the quest for perfection is

rigorous global optimization

... though as long as we can't do it we must be content with achieving less demanding partial goals.

In a **constraint satisfaction problem (CSP)**,
one asks about **existence**, and one or several examples:

Can 13 balls of radius r touch a ball of radius R ?

For $r = R$, No! (van der Waerden, ~ 1953)

In a **global optimization problem (GOP)**,
one asks about an **extremum**,
and a configuration where it is achieved.

What is the smallest possible R ?

It is $R \approx 1.09r$.

In a **constraint projection problem (CPP)**,
one asks about **exhaustion** of the solution set,
and display of a suitable low-dimensional projection of it.

What is the set of possible (r, R) ?

$\Sigma = \{(r, R) \mid R \geq 1.09r\}$ (apart from roundoff).

Dichotomies

- heuristic \Leftrightarrow guaranteed
... important for reliability
- linear \Leftrightarrow nonlinear
... important for realistic modeling
- local \Leftrightarrow global
... important for jumps in quality
- black box \Leftrightarrow structured access
... important for efficient analysis

Degrees of rigor

- incomplete (heuristics; e.g., smoothing techniques)
- asymptotically complete (no means to know when a global minimizer has been found; e.g., multiple random start, pure branching)
- complete (knows after a finite time that an approximate global minimizer has been found to within prescribed tolerances, assuming exact calculation).
- rigorous (complete, with full rounding error control)

(Often, the label deterministic is used to characterize the last two categories of algorithms; however, this label is slightly confusing since many incomplete and asymptotically complete methods are deterministic, too.)

Complexity

Already in the case where only bound constraints are present, global optimization problems and constraint satisfaction problems are

- undecidable on unbounded domains (WENXING ZHU 2004), and
- NP-hard on bounded domains.

This implies natural limits on the solvability of such problems in practice.

In particular, methods which work with direct attack (analytic transformations without problem splitting) lead to transformed problems of exponentially increasing size, while branch-and-bound methods split the problems into a worst case exponential number of subproblems.

It is very remarkable that in spite of this, many large-scale problems can be solved efficiently.

This is achieved by carefully balancing the application of the available tools and using the internal structure which realistic problems always have.

The great success of the current generation of complete global solvers is due mainly to improvements in our ability to analyze global optimization problems mathematically.

For history, a thorough introduction, a comprehensive overview over the techniques in use, and extensive references see my survey

A. Neumaier

Complete search in continuous global optimization and constraint satisfaction,

pp. 1-99 in: Acta Numerica 2004,

Cambridge Univ. Press 2004.

Complete search techniques I

Direct attack is feasible for polynomial systems of moderate degree (up to about 20)

- Gröbner basis methods
- resultant-based techniques
- eigenvalue methods (Stetter-Müller)
- semidefinite relaxations (GloptiPoly, SOStools, SparsePOP)

Gröbner bases and resultants serve to compute the solution set of a system of algebraic equations $F(x) = 0$, the most basic of all continuous constraint satisfaction problems.

These are primarily for exact computations in rational arithmetic, and are often numerically unstable when done in finite precision arithmetic.

For approximate computations one therefore uses a reformulation of the system of equations as an eigenvalue problem on the vector space of cosets of the ideal generated by the constraints.

In important cases, this coset space has finite dimension, allowing one to extract the (in this case finitely many) solutions.

The work is cubic in the number of complex solutions, even if there are very few real solutions. This limits the applicability to low dimensions.

Semidefinite relaxations apply to polynomial systems of equations and inequalities, and are based on the insight that testing whether a polynomial is a sum of squares (which implies its nonnegativity) is equivalent to testing the semidefiniteness of an associated moment matrix.

The basic method scales exponentially with the degree of the polynomials used in the relaxation, but recent progress by Kojima's group on sparse formulations has led to the solution of some (tridiagonal, unconstrained) problems of dimension up to 500.

Complete search techniques II

Branch-and-bound methods are the choice for larger problems. Basic approaches use

- constraint propagation
- outer approximation
(linear, convex, conic, semidefinite)
- DC (difference of convex function) techniques
- interval Newton and related methods

Branch and Bound originated as the basic workhorse of integer programming, was later adapted to continuous global optimization, and is now also used for mixed-integer nonlinear programming, which combines the two fields.

A problem is recursively split into subproblems, forming a tree.

Subproblems are simplified or eliminated using range reduction techniques, until one or all solutions are found or the problem is found infeasible.

The work is worst case exponential in the depth of the tree.

Constraint propagation originated in logic.

Deciding the truth of a statement in propositional logic is equivalent in a straightforward way to deciding the feasibility of a discrete constraint satisfaction problem.

The basic range reduction in this case is to eliminate values that are forbidden by considerations of few (often only one) constraint at a time.

In 1847, George Boole discovered that logic can be embedded into real polynomial algebra.

He encoded logical variables as indeterminates x constrained to be idempotent, $x^2 = x$, and the logical operations by polynomial, idempotence preserving formulas:

$$\neg x := 1 - x,$$

$$x \wedge y := xy,$$

$$x \vee y := x + y - xy,$$

$$x \Rightarrow y := 1 - x + xy.$$

This turned logical reasoning into a subfield of real algebra, after a century of discrete logical thinking today (in the context of reformulation-linearization and semidefinite relaxations) again a fashionable approach.

It is interesting to note that logic as operations on the field of 2 elements – what is now called Boolean algebra – was to come much later than Boole's real algebra logic; abstract algebra was foreign to Boole.

For real variables, constraint propagation consists in deducing better bounds for a variable by using the other bounds and one of the constraints. For example, in the constraint satisfaction problem

$$F_1(x) = -x_1 - 2x_2 \leq -2, \quad (1)$$

$$F_2(x) = (x_1 - 1)^2 + (x_2 - 1)^2 = 1, \quad (2)$$

$$x_1 \in [-1, 1], \quad x_2 \in [-1, 1], \quad (3)$$

(1) implies $x_2 \geq 1 - x_1/2 \geq 0.5$ since $x_1 \leq 1$, and $x_1 \geq 2 - 2x_2 \geq 0$ since $x_2 \leq 1$, reducing the bounds to

$$x_1 \in [0, 1], \quad x_2 \in [0.5, 1].$$

Similarly, (2) implies $(x_1 - 1)^2 = 1 - (x_2 - 1)^2 \geq 1 - 0.25 = 0.75$, hence $x_1 \leq 1 - \sqrt{0.75} < 0.14$, giving the improved bound

$$x_1 \in [0, 0.14].$$

This bound improves again x_2 using (1). Alternating use of (1) and (2) shrinks the boxes, converging towards the single feasible point.

Interval techniques became initially known mainly as a tool to control rounding errors. However, it is much less known – and much more important – that their real strength is the ability to control nonlinearities in a fully satisfying algorithmic way.

In the previous example, the gradient of $F_2(x)$ at $x \in \mathbf{x} = ([0, 0.14], [0.5, 1])^T$ is

$$F'_2(x) = (2x_1 - 2, 2x_2 - 2) \in ([-2, -1.72], [-1, 0]) = F'_2(\mathbf{x}).$$

The mean value theorem implies that, for any $\tilde{x} \in \mathbf{x}$,

$$F(x) \in F(\tilde{x}) + F'(\mathbf{x})(x - \tilde{x}) \quad \text{if } x \in \mathbf{x}.$$

Using $\tilde{x} = \hat{x}$ we find for the second component

$$1 \in 1 + [-2, -1.72]x_1 + [-1, 0](x_2 - 1) = [1 - 2x_1, 2 - 1.72x_1 - x_2].$$

The upper bound leads to the new constraint $1.72x_1 + x_2 \leq 1$, a linear relaxation of (2) useful for linear programming techniques.

We illustrate the power of interval methods by considering a nonlinear system $F(x) = 0$.

Newton's method approximates

$$0 = F(x) \approx F(z) + F'(z)(x-z) \quad \Rightarrow \quad x \approx z - F'(z)^{-1}F(z).$$

Interval slopes control errors by using instead

$$0 = F(x) = F(z) + F[x, z](x-z) \quad \Rightarrow \quad x = z - F[x, z]^{-1}F(z).$$

If we look for zeros in a box \mathbf{x} and know

$$F[z, x] \in \mathbf{A} \quad \text{for all } x \in \mathbf{x}$$

with a regular interval matrix \mathbf{A} (computable by automatic differentiation like techniques) then

$$x \in \mathbf{x}' := z - \mathbf{A}^{-1}F(z).$$

Properties of the computed box x' :

$x' \cap x = \emptyset \quad \Rightarrow \quad$ there is no zero in x

$x' \in \text{int } x \quad \Rightarrow \quad$ there is a zero in x

In any case, any zero in x is also in $x' \cap x$

Uniqueness proofs are also possible.

Similar properties hold for systems of equations and inequalities, and for optimization problems.

A full account of the theoretical background for interval techniques in finite dimensions is available in my book

A. Neumaier

Interval methods for systems of equations

Cambridge Univ. Press 1990.

The book is still up to date (with a few minor exceptions). While it is officially out of print, if you order it at Cambridge University Press, they'll print an extra copy especially for you.

Outer approximation by convex problems and difference of convex (DC) formulations create convex relaxations that can be handled by local optimization techniques.

In general purpose solvers, they exploit the factorable structure of the global optimization problems.

Factorable functions are more general than polynomials in that they allow the use of arbitrary univariate elementary functions.

For general purpose problems, often interval techniques are used to remove nonlinearities (resulting in linear constraints) or to compute convexifying correction terms (resulting in a DC formulation).

Alternatively, the problem is fully or partially decomposed into constraints involving only a single nonlinear operation, by treating intermediate variables on the same footing as the original variables.

This produces a large, sparse problem formulation for which optimal linear or semidefinite relaxations can be computed analytically.

Complete search techniques III

Further efficiency is gained by

- use of optimality conditions
- multiplier techniques
(duality, Lagrangian relaxation)
- cut generation
- adding redundant constraints
- graph decomposition techniques
- certificates of optimality/infeasibility

Necessary and sufficient global optimality conditions

We consider the polynomial optimization problem

$$\begin{aligned} \min f(x) \\ \text{s.t. } C(x) \geq 0, \quad F(x) = 0, \end{aligned} \tag{4}$$

with $f \in \mathbb{R}[x_{1:n}]$, $C \in \mathbb{R}[x_{1:n}]^m$, and $F \in \mathbb{R}[x_{1:n}]^r$, where $\mathbb{R}[x_{1:n}]$ is the algebra of real polynomials in the variables x_1, \dots, x_n .

Write *SOS* for the set of sums of squares of polynomials in x_1, \dots, x_n . Write B for the vector with components $B_0(x) = f(\hat{x}) - f(x)$ and $B_k = C_k$ for $k > 0$, and B_S for the vector whose entries are the 2^{m+1} polynomials $\prod_{i \in I} B_i$, where I runs over all subsets of $\{0 : m\}$.

Theorem (NEUMAIER & SCHICHL 2005)

For a feasible point \hat{x} of the polynomial optimization problem (4), the following are equivalent:

(i) The point \hat{x} is a global minimum of (4).

(ii) There are a polynomial $y_0 \in SOS$, polynomial vectors $Y \in SOS^{2^{m+1}}$, $Z \in \mathbb{R}[x_{1:n}]^r$, and a positive integer e such that

$$B_0(x)^e + y_0(x) + Y(x)^T B_S(x) + Z(x)^T F(x) = 0 \quad (5)$$

identically in x .

Moreover, any solution of (5) satisfies

$$y_0(\hat{x}) = 0, \quad \inf\{Y(\hat{x}), B_S(\hat{x})\} = 0, \quad F(\hat{x}) = 0, \quad (6)$$

$$\delta_{e1} f'(\hat{x})^T = B_S'(\hat{x})^T Y(\hat{x}) + F'(\hat{x})^T Z(\hat{x}). \quad (7)$$

Complete search techniques IV

Efficiency and reliability also require the use of

- local optimization for upper bounds
- clever box selection heuristics
- adaptive splitting heuristics
- reliable stopping criteria
- combination heuristics
- safeguarding techniques

The COCONUT test set

Number of variables	1 – 9	10 – 99	100 – 999	≥ 1000	
	size 1	size 2	size 3	size 4	total
Library 1 (GLOBALLIB from GAMS)	84	90	44	48	266
Library 2 (CUTE from Vanderbei)	347	100	93	187	727
Library 3 (CSP from EPFL)	225	76	22	6	329
total	656	266	159	241	1322

Detailed test results are available on the COCONUT homepage
(www.mat.univie.ac.at/coconut).

Reliability analysis for BARON 7.2 (2004)	
	global minimum found/accepted
size 1	524/579 \approx 91%
size 2	210/229 \approx 92%
size 3	88/140 \approx 63%
all	821/950 \approx 86%
	correctly claimed global/accepted
size 1	450/579 \approx 78%
size 2	151/229 \approx 66%
size 3	43/140 \approx 31%
all	644/950 \approx 68%
	wrongly claimed global/claimed global
size 1	14/464 \approx 3%
size 2	7/158 \approx 4%
size 3	8/51 \approx 16%
all	29/675 \approx 4%
	claimed infeasible/accepted and feasible
size 1	3/571 \approx 1%
size 2	1/222 \approx 0%
size 3	0/128 = 0%
all	4/921 \approx 0.4%

In the meantime, BARON improved further. Although commercial, problems can be solved online via the NEOS optimization server. Input is in GAMS; a translator from AMPL to GAMS is available in the COCONUT environment.

The authors of BARON, Nick Sahinidis and Mohit Tawarmalani were awarded the 2006 Beale-Orchard-Hayes Prize of the Mathematical Programming society for their global optimization software.

It is the standard against which new global optimization software must be judged.

Complete, but nonrigorous methods (as used by BARON) are today superior to the best general purpose stochastic methods in small and medium dimensions (< 1000).

But incomplete methods are currently still the only choice available for difficult large-scale problems such as protein folding, radiation therapy planning, optimal design and packing.

Even this might change in the near future.

Rounding errors

Lack of reliability due to rounding errors is still a problem in current codes. In the optimization community, awareness of this problem is only slowly growing.

When I wrote in 1986 my first branch and bound code for covering solution curves of algebraic equations, every now and then a pixel was missing in the pictures. It turned out that the reason was a loss of solutions due to poor handling of roundoff problems.

Geometric visualization software had to cope with the same problem: Today they all use carefully designed algorithms with safe rounding in critical computations.

Not so in optimization codes,
unfortunately!

Even high quality mixed integer linear programming (MILP) codes which have already a long commercial tradition may fail due to roundoff!

CPLEX 8.0 and all but one MILP solver from NEOS failed in 2002 to handle a simple 20 variable MILP problem with small integer coefficients and solution, claiming that no solution exists.

But cheap rigorous safeguards based on interval arithmetic are now available (NEUMAIER & SHCHERBINA 2004, JANSSON 2004, 2005), and will probably be soon part of commercial packages.

The 'innocent' MILP is the case $s = 6$ of the
20 variable integer linear problem

$$\begin{aligned} \min \quad & -x_{20} \\ \text{s.t.} \quad & (s + 1)x_1 - x_2 \geq s - 1, \\ & -sx_{i-1} + (s + 1)x_i - x_{i+1} \geq (-1)^i(s + 1) \text{ for } i = 2 : 19, \\ & -sx_{18} - (3s - 1)x_{19} + 3x_{20} \geq -(5s - 7), \\ & 0 \leq x_i \leq 10 \text{ for } i = 1 : 13, \\ & 0 \leq x_i \leq 10^6 \text{ for } i = 14 : 20, \\ & \text{all } x_i \text{ integers} \end{aligned}$$

FortMP found the solution $x = (1, 2, 1, 2, \dots, 1, 2)^T$,
but...

Five other MIP solvers from NEOS (June 2002), namely GLPK, XPRESS-MP, MINLP, BONSAIG, XPRESS and the solver CPLEX 8.0 (not available through NEOS) claimed no solution exists. (Of course, newer versions handle this particular problem correctly since it was published....)

Most solvers suffer from rounding errors introduced through ill-conditioning. (The solution is a nearly degenerate nonoptimal vertex of the linear programming relaxation.)

WARNING: A high proportion of real life linear programs (72% according to ORDÓÑEZ & FREUND, and still 19% after preprocessing) are ill-conditioned!

Primal linear program:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \underline{b} \leq Ax \leq \bar{b}, \end{aligned} \tag{8}$$

Corresponding dual linear program:

$$\begin{aligned} \max \quad & \underline{b}^T y - \bar{b}^T z \\ \text{s.t.} \quad & A^T(y - z) = c, \quad y \geq 0, \quad z \geq 0. \end{aligned} \tag{9}$$

Introduce boxes:

$$\mathbf{b} := [\underline{b}, \bar{b}] = \{\tilde{b} \in \mathbb{R}^n \mid \underline{b} \leq \tilde{b} \leq \bar{b}\},$$

Assume

$$Ax \in \mathbf{b} \quad \Rightarrow \quad x \in \mathbf{x} = [\underline{x}, \bar{x}].$$

From an approximate solution of the dual program we calculate an approximate multiplier $\lambda \approx z - y$, and a rigorous interval enclosure for

$$r := A^T \lambda - c \in \mathbf{r} = [\underline{r}, \bar{r}].$$

Since $c^T x = (A^T \lambda - r)^T x = \lambda^T Ax - r^T x \in \lambda^T \mathbf{b} - \mathbf{r}^T \mathbf{x}$,

$$\mu := \inf(\lambda^T \mathbf{b} - \mathbf{r}^T \mathbf{x})$$

is the desired rigorous lower bound for $c^T x$.

In well-conditioned cases, the bound is quite accurate, while in ill-conditioned cases, it is so poor that it warns the user (or the algorithm) that something went wrong and needs special attention.

Safeguarding the unbounded case and semidefinite problems can also be done (Jansson).

Directed Cholesky factorizations

A strictly convex, quadratic inequality constraint defines an ellipsoid whose interval hull is easy to compute analytically.

However, to cope efficiently with rounding errors is nontrivial.

We devised an algorithm which computes a directed Cholesky factor for a symmetric, positive definite interval matrix A , i.e., a nonsingular triangular matrix R such that $A - R^T R$ is positive semidefinite for all $A \in \mathbf{A}$, and tiny if the components of \mathbf{A} are narrow intervals.

In particular, $x^T A x \geq \|R x\|_2^2$, with little overestimation.

This allows us to write strictly convex, quadratic inequalities in the form

$$\|Rx\|_2^2 - 2a^T x \leq \alpha.$$

By solving $R^T R x_0 = a$, we find bounds

$$\|R(x - x_0)\|_2 \leq \delta := \sqrt{\max\{0, \alpha + a^T x_0\}},$$

from which one can find the componentwise bounds

$$|x - x_0| \leq \delta \operatorname{diag}(R^{-1} R^{-T}).$$

The inversion of R must also be safeguarded against rounding errors.

The computations take $O(n^3)$ operations.

Challenges for the future I

- Ensuring reliability (safe bounds, finite termination analysis, certificates for a posteriori checks)
- Integrating MIP and SDP techniques into a branch-and-bound framework
- unbounded variables
- unconstrained/bound constrained problems (the more constraints the easier the problem!
⇒ bounded residual estimation preferable to least squares)

Challenges for the future II

- Problems with severe dependence
(volume preserving recurrences imply heavy wrapping)
- Problems with symmetries
(optimal design of experiments,
chemical cluster optimization)
- sensitivity analysis
- parametric global optimization
- constraint projection problems

Challenges for the future III

- Differential constraints (optimal control of chemical plants; space mission design)
- Integral constraints (expectations; value at risk; engineering safety factors)
- Semi-infinite constraints (robust optimization; optimal design)
- Other desirables (black box functions; expensive functions; nonsmoothness)

Conclusion

In spite of dramatic progress, there is lots of interesting work to be done at the interface between

Mathematics –

the art and science of unambiguous concepts

Logic –

the art and science of impeccable reasoning

Operations Research –

the art and science of optimal planning

Numerical Analysis –

the art and science of algorithmic approximation

Computer Science –

the art and science of efficient computation

A. Neumaier

Complete search in continuous global optimization
and constraint satisfaction,

pp. 1-99 in: Acta Numerica 2004,

Cambridge Univ. Press 2004.

A. Neumaier, O. Shcherbina, W. Huyer and T. Vinko

A comparison of complete global optimization solvers

Math. Programming B 103 (2005), 335-356.

H. Schichl and A. Neumaier

Transposition theorems and qualification-free optimality
conditions, SIAM J. Optimization, to appear.

<http://www.mat.univie.ac.at/~neum/papers.html#trans>

Global (and Local) Optimization site

`www.mat.univie.ac.at/~neum/glopt.html`

COCONUT homepage

`www.mat.univie.ac.at/coconut`

A. Neumaier

Interval methods for systems of equations

Cambridge Univ. Press 1990.

(individual copies are reprinted upon request)