Survey of Global Optimization

after "Complete Search in Continuous Global Optimization and Constraint Satisfaction"
by Arnold Neumaier

by Łukasz Stafiniak

Completeness

- An **incomplete** method uses clever intuitive heuristics for searching but has no safeguards if the search gets stuck in a local minimum.
- An **asymptotically complete** method reaches a global minimum with certainty or at least with probability one if allowed to run indefinitely long, but has no means to know when a global minimizer has been found.
- A **complete** method reaches a global minimum with certainty, assuming exact computations and indefinitely long run time, and knows after a finite time that an approximate global minimizer has been found (to within prescribed tolerances).
- A **rigorous** method reaches a global minimum with certainty and within given tolerances even in the presence of rounding errors, except in near-degenerate cases, where the tolerances may be exceeded.

Some Use Domains

- **hard feasibility problems** (e.g., robot arm design), where local methods do not return useful information since they generally get stuck in local minimizers of the merit function, not providing feasible points;
- **computer-assisted proofs** (e.g., the proof of the Kepler conjecture by Hales [128]), where inequalities must be established with mathematical guarantees;
- **safety verification problems**, where treating nonglobal extrema as worst cases may severely underestimate the true risk;
- many problems in **chemistry**, where often only the global minimizer (of the free energy) corresponds to the situation matching reality;

Formulation

A constrained global optimization problem

$$\begin{aligned} & \min f(x) \\ \text{such that} \quad & x \in \boldsymbol{x}, F(x) \in \boldsymbol{F}, x_I \text{ integral} \\ & \boldsymbol{x} = \{x \in \mathbb{R}^n \,|\, \underline{x} \leqslant x \leqslant \bar{x}\} \\ & x_I = (x_{i_1}, \ldots, x_{i_l}) \\ & C = \{x \in \boldsymbol{x} \,|\, x_I \text{ integral}, F(x) \in \boldsymbol{F}\} \end{aligned}$$

where $\boldsymbol{x}$ is a bounded or unbouded box, C is the feasible domain, points $x \in C$ are feasible, a solution (or global minimizer) $\hat{x}$ is a point $\hat{x} \in C$, $f(\hat{x}) = \min_{x \in C} f(x)$. A local (global) solver is an algorithm that finds local (global) minimizers. A constraint satisfaction problem is to decide if the feasible set C is nonempty and find any $\hat{x} \in C$.

- If I is nonempty, the problem is called mixed integer (e.g. linear, nonlinear) program.
- It is simply constrained when $\dim F = 0$,
- separable if $f(x) = \sum_{k=1}^{n} f_k(x_k)$ and $F(x) = \sum_{k=1}^{n} F_k(x_k)$,
- factorable if f, F are polynomials of $f_k(x_k)$,
- DC when f, F are differences of convex functions.

Lagrange Multipliers and Kuhn-Tucker conditions

For every local minimizer $\hat{x}$, there are a number $\kappa \geqslant 0$ and a vector λ, not both zero, such that the vector

$$g^T = \kappa f'(\hat{x}) + y^T F'(\hat{x})$$

satisfies

$$g_i \begin{cases} \geqslant 0 & \text{if } \underline{x}_i = \hat{x_i} < \bar{x}_i, i \notin I, \\ \leqslant 0 & \text{if } \underline{x}_i < \hat{x_i} = \bar{x}_i, i \notin I, \\ = 0 & \text{if } \underline{x}_i < \hat{x_i} < \bar{x}_i, i \notin I, \end{cases}$$

$$\lambda_i \begin{cases} \geqslant 0 & \text{if } \underline{F}_i = F(\hat{x_i}) < \bar{F}_i, \\ \leqslant 0 & \text{if } \underline{F}_i < F(\hat{x_i}) = \bar{F}_i, \\ = 0 & \text{if } \underline{F}_i < F(\hat{x_i}) < \bar{F}_i, \end{cases}$$

For $\kappa = 1$ (i.e. $\kappa > 0$), then λ are called Lagrange multipliers and g is the gradient of the Lagrangian

$$L(x, \lambda) = f(x) + \lambda^T F(x).$$

[Illustration example.]

Methods for Local Optimization

Newton's Method

The Taylor expansion of $f(x)$

$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2$$

attains extremum when Δx solves $f'(x) + f''(x)\Delta x = 0$ and $f''(x) > 0$.

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

and in several dimensions

$$x_{n+1} = x_n - [H f(x_n)]^{-1} \nabla f(x_n)$$

usually with a smaller step $\gamma \in (0,1)$ (the Taylor is an approximation)

$$x_{n+1} = x_n - \gamma [H f(x_n)]^{-1} \nabla f(x_n)$$

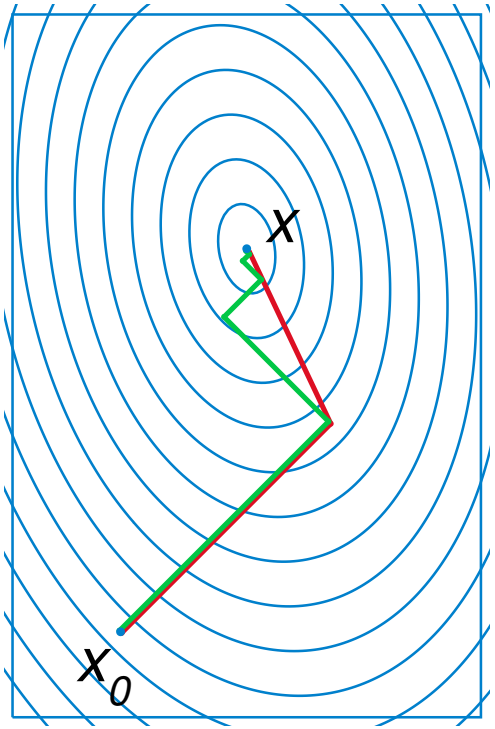
Quasi-Newton Methods

approximate the inverse of Hessian, in each step updating the approximation.

[TODO]

Conjugate Gradients Methods

solve the update equation $H\,f(x_n)x_{n+1} = \nabla f(x_n)$ approximately (using the so-called conjugate gradients). Quasi-Newton methods (e.g., BFGS method) - converge in fewer iterations, although each iteration requires more computation and more memory than a conjugate gradient iteration.



Adaptive Step Methods

[TODO]

Incomplete methods for simple constraints

Besides **simulated annealing** and **genetic algorithms**:

- **multiple random start**: repeat: pick random point and optimize from it (e.g. locally)

- **Smoothing (= homotopy = continuation) methods** are based on the intuition that, in nature, macroscopic features are usually an average effect of microscopic details; averaging smoothes out the details in such a way as to reveal the global picture. The hope is, most or all local minima disappear, and the remaining major features of the surface only show a single minimizer. By adding more and more details, the approximations made by the smoothing are undone, and finally one ends up at the global minimizer of the original surface. The quality of the final local minimizer depends on the homotopy, and frequently is the global or at least a good local minimizer.

- **Response surface techniques** are designed specifically for the global optimization of functions that are very expensive to evaluate. They construct in each iteration an interpolation or approximation surrogate function of known analytic form. The surrogate function is then subjected to global optimization. The resulting optimizers (or some points where the feasible region has been only sparsely explored) are taken as new evaluation points.
- **Clustering methods** first sample the space, retain a fraction of sampled points (e.g. 80%) (with promising values, and recently with positive Hessians), perform clustering using **data mining** techniques, and perform local minimization in each cluster.

In general, incomplete methods tend to fail systematically to find the global optimum on the more difficult problems in higher dimensions, but they frequently give relatively good points with a reasonable amount of effort. Beyond a certain number of function evaluations (that depends on the problem), progress slows down drastically if the global optimum has not yet been located already. This is unlikely to change in the future, although new heuristics and variations of old ones are discovered almost every year.

Reduction to simple constraints

Penalty and barrier formulations.

The reformulation changes the solution: an approximation method, and the result should be used as a starting point for a subsequent local optimization of the original problem.

With soft constraints (some violation is tolerated):

$$
\begin{aligned}
q(x) &= \frac{f(x) - f_0}{\Delta + |f(x) - f_0|} \\
\delta_i(x) &= \begin{cases} (F_i(x) - \underline{F}_i)/\underline{\sigma}_i & \text{if } F_i(x) \leqslant \underline{F}_i \\ (F_i(x) - \bar{F}_i)/\bar{\sigma}_i & \text{if } F_i(x) \geqslant \bar{F}_i \\ 0 & \text{otherwise} \end{cases} \\
r(x) &= \frac{2 \sum \delta_i^2(x)}{1 + \sum \delta_i^2(x)} \\
f_{\text{merit}}(x) &= q(x) + r(x)
\end{aligned}
$$

$f_{\mathrm{merit}} \in (-1, 3)$ and the global minimizer $\hat{x}$ of f_{merit} in $\boldsymbol{x}$ either satisfies

$$\begin{aligned} F_i(\hat{x}) &\in [\underline{F}_i - \underline{\sigma}_i, \bar{F}_i + \bar{\sigma}_i] \text{ for all } i \\ f(\hat{x}) &\leqslant \min\{f(x) | F(x) \in \boldsymbol{F}, x \in \boldsymbol{x}\} \end{aligned}$$

or one of the following holds

$$\begin{aligned} \{x \in \boldsymbol{x} | F(x) \in \boldsymbol{F}\} &= \varnothing \\ f_0 &< \min\{f(x) | F(x) \in \boldsymbol{F}, x \in \boldsymbol{x}\} \end{aligned}$$

Projection penalties.

In certain cases (e.g. linear and convex quadratic constraints) an exact reformulation as a nonsmooth but Lipschitz continuous simply constrained problem is possible. The idea is to project infeasible points to the feasible domain. For known interior point x_0:

$$\begin{aligned} \bar{f}(x) &:= f(\bar{x}) + \gamma \|\bar{x} - x\|^2 \\ \bar{x} &= \lambda x_0 + (1-\lambda)x \\ \lambda = \lambda_x &\in [0,1] \end{aligned}$$

λ_x is smallest such that $\bar{x}$ satisfies the constraints.

Pure branching methods

Density Theorem

Any method based on local information only that converges for every continuous f to a **global minimizer** of f in a feasible domain C **iff** it produces a sequence of points $x_1, x_2, \ldots$ that is **dense** in C.

No Free Lunch Theorem

The set of all objective functions is Y^X, where X is a finite solution space and Y is a finite poset. The set of all permutations of X is J. Random variable F is distributed on Y^X. For all j in J, $F \circ j$ is a random variable distributed on Y^X, with $\Pr\{F \circ j = f\} = \Pr\{F = f \circ j^{-1}\}$ for all f in Y^X.

Let $a(f)$ denote the output of search algorithm a on input f. If $a(F)$ and $b(F)$ are identically distributed for all search algorithms a and b, then F has an NFL distribution. This condition holds if and only if F and $F \circ j$ are identically distributed for all j in J.

Branching scheme / splitting rules

- To comply with Density Theorem, split boxes so that their diameters tend to zero; the key to efficiency is a proper balance of global and local search.
- DIRECT splits in each round all boxes for which the pair $(v,\, f)$ (where v is the volume and f the midpoint function value) is not dominated by another such pair. Here $(v,\, f)$ is dominated by $(v,\, f)$ if both $v' < v$ and $f' > f$.
- MCS uses domination of pairs $(l,\ f)$, where l is a suitably assigned level, and in addition employs local optimization steps (using line searches and sequential bound constrained quadratic programs) from appropriate candidate points.
- LGO uses lower bounds $L \geqslant \max_{k,l} \|f(x_k) - f(x_l)\| / \|x_k - x_l\|$ on Lipschitz constants L obtained from the previous function evaluations to decide on the promising boxes to split first.

Box reduction

The search is organized into "boxes"; box reduction = reducing the size (and possibly discarding boxes when size = 0) by discarding unfeasible fragments.

- **Constraint propagation** is a very cheap and easily formalizable process that gives important initial range reductions in many otherwise difficult problems. It consists in deducing better bounds for a variable by using the other bounds and one of the constraints.
- **Interval analysis** can be applied in a number of different ways. E.g. to produce linear relaxations of the nonlinear constraint.
- Other **relaxation techniques** to create a convex relaxation of the constraints.
- It might still be the case that there is a second, undiscovered global minimizer. This can be checked with **multiplier techniques**.

Branch and Bound

Branching rules

Select a bisection coordinate j and split the j-th component of the box at bisection point ξ.

- midpoint: $\xi = (\bar{x}_j + \underline{x}_j)/2$
- safeguarded geometric mean (when the interval is wide):

$$\xi = \begin{cases} \operatorname{sign} \underline{x}_j \sqrt{\underline{x}_j \bar{x}_j} & \text{if } 0 < \underline{x}_j \bar{x}_j < \infty \\ 0 & \text{if } \underline{x}_j < 0 < \bar{x}_j \\ \min(\mu, q\bar{x}_j) & \text{if } \underline{x}_j = 0 \\ \max(-\mu, q\bar{x}_j) & \text{if } \bar{x}_j = 0 \\ q^{-1}\underline{x}_j & \text{if } \underline{x}_j > 0 \text{ thus } \bar{x}_j = \infty \\ q^{-1}\bar{x}_j & \text{if } \bar{x}_j > 0 \text{ thus } \underline{x}_j = -\infty \end{cases}$$

where $q \in (0, 1)$ is a fixed constant (e.g. $q = 0.01$).

Bounding rules

Usually a convex relaxation: a convex (and often linear) optimization problem whose feasible set contains the feasible set of the subproblem (outer approximation) and whose objective function is at no feasible point larger than the original objective function (underestimation). If the convex problem is infeasible or provides a lower bound larger than best so far feasible $f(x)$, the subproblem can be discarded. It is equivalent to adding a cut $f(x) \leqslant f^{\mathrm{best}}$ (for known $f(x^{\mathrm{found}}) = f^{\mathrm{best}}$). Cuts can be more complicated, linear cuts are called cutting planes and using cuts is called branch and cut.

Reduction rules approximate the feasible set to reduce the size of the box.

Use of Local Optimization

Tunneling tries to find a better solution starting from a local minimum. One technique is to minimize the sum of squares of constraint violations, with additional constraint $f(x) \leqslant f^{\text{best}} - \Delta$.

Rules of thumb for problem difficulty, solving a problem of comparable size and sparsity structure:

time needed to solve a linear program	LP
convex quadratic program	$\text{QP} = 5 * \text{LP}$
local min. of a nonconv. quadratic program	$\text{QP}' = 10 * \text{LP}$
for a convex nonlinear program	$\text{SQP} = 30 * \text{QP}$
local minimizer of a nonconv. nonlin. program	$\text{SQP} = 200 * \text{QP}$
global min. of a nonconv. nonlin. program	$\text{GLP}_f = 100 * \text{SQP}$
verifying that it is a global minimizer	$\text{GLP}_v = 1000 * \text{SQP}$

Pure constraint satisfaction dominates global search at initial stage: find first feasible point, and at final stage: show no feasible points smaller than found minimizer.

Constraint Propagation

Examples of theorems useful for box reduction:

Let J_k be subsets of indices, let q_k be real-valued functions on $\boldsymbol{x}_k$. If for suitable $\bar{q}_k, \bar{s}$

$$\bar{q}_k \geqslant \sup \{q_k(x_{J_k}) | x_{J_k} \in \boldsymbol{x}_{J_k}\}, \bar{s} \geqslant \sum_k \bar{q}_k$$

then, for arbitrary $\underline{a}$,

$$x \in \boldsymbol{x}, \underline{a} \leqslant \sum_k q_k(x_{J_k}) \Longrightarrow q_k(x_{J_k}) \geqslant \underline{a} - \bar{s} + \bar{q}_k \, (\forall k)$$

Dually for $\underline{q}_k \leqslant \inf$. It is easily generalized to double-sided inequalities useful in **interval arithmetic**.

More difficult example: **semiseparable constraints**. We have a semiseparable inequality of the form

$$\sum_k q_k(x_k) + (x - x^0)^T H (x - x^0) \leqslant \bar{a}$$

with possibly nonsymmetric H, and (modified Cholesky factorization)

$$H + H^T = R^T R - D$$

with a nonnegative diagonal D. Then

$$\left\| R(x - x^0) \right\|_2^2 \leqslant 2(\bar{a} - \underline{s})$$

Conditioning can be used to separate variables: rewrite $F_i(x) \in \boldsymbol{F}_i$ as

$$F_i(\xi) + F'(\boldsymbol{x})(x - \xi) \cap \boldsymbol{F}_i \neq \varnothing$$

(e.g. use with $q_k(x_k) = \boldsymbol{\alpha} x_k$). Going second-order may let use the result for semiseparable constraints.

Shaving: trying to discard a small slab of an interval $[\underline{x}_i, \bar{x}_i]$. While expensive, it reduces the overestimation in the processing of constraints which contain a variable several times.

The cluster problem

When programming a simple branch and bound algorithm for global optimization, it is fairly easy to eliminate boxes far away from the global minimizer, while, especially in higher dimensions, there remains a large cluster of tiny boxes in a neighborhood of the global minimizer that is difficult to eliminate. Often, algorithms try to avoid the cluster problem by providing only a Δ-optimal solution; i.e., the program stops when it has shown that there is no feasible point with an objective function value of $f^{\text{best}} - \Delta$. However, when Δ is small then the cluster problem is still present.

For the cluster problem for unconstrained global optimization (Kearfott & Du), the source of the problem is the limited accuracy with which the function values were bounded: it disappears if, for x in a box of diameter $O(\varepsilon)$, one can bound the overestimation of $f^{\text{best}} - f(x)$ by $O(\varepsilon^3)$. For pure constraint satisfaction problems, a similar cluster effect is present (Schichl & Neumaier), but with $O(\varepsilon^2)$ sufficient. Several techniques to deal with these cases. If after using them

we are still left with a box of significant size, we must have been close to a degeneracy; splitting would probably not improve this and lead to an exponential number of boxes; thus it is preferable to put this box in the list of output boxes to indicate that a low resolution candidate for a solution has been found. Usually a large region (box) around the output box can be excluded (basin of the minimum) (backboxing).

Linear and convex relaxations

By introducing new variables, **factorable** problems can be rewritten with unary $z = \varphi(x)$ and binary $z = x \circ y$ constraints only. Unary elementary functions can be easily linearly over/under-estimated (e.g. secant/tangent). Only binary operations that need to be analyzed are products and quotients. $x \in \boldsymbol{x}, y \in \boldsymbol{y}, z = x\,y$:

$$\underline{y}\,x + \underline{x}\,y - \underline{x}\;\underline{y} \leqslant z \leqslant \underline{y}\,x + \bar{x}\,y - \bar{x}\;\underline{y}$$

$$\bar{y}\,x + \bar{x}\,y - \bar{x}\,\bar{y} \leqslant z \leqslant \bar{y}\,x + \underline{x}\,y - \underline{x}\,\bar{y}$$

Semidefinite relaxations (aka **convex conic relaxations**) are a recent and efficient approach (not yet fully integrated with global optimization software).

Relaxations without extra variables

The first possibility is to write the constraints as a difference of convex functions (DC representation). The package αBB separates in each inequality constraint a recognizable linear, convex or concave parts from a "general" remainder. Linear and convex parts are kept, concave parts are overestimated by secant type constructions, and general terms are made convex by adding a nonpositive separable quadratic function.

The second possibility is to use centered forms, e.g. based on first-order Taylor form $f(x) \in \boldsymbol{f} + c^T(x - z)$.

Semilinear constraints and MILP

A constraint is semilinear if, for arguments x in a bounded box $\boldsymbol{x}$, it is equivalent to a finite list of linear constraints and integer constraints. The objective function $f(x)$ is called semilinear if the inequality $f(x) \leqslant x_0$, where x_0 is an additional variable, is semilinear. A semilinear program is an optimization problem with a semilinear objective function and a bounded feasible domain defined by semilinear constraints only. They are equivalent to **mixed integer linear programs** for which efficient solvers exist.

Examples of semilinear constraints

$z \in \{0,1\}$	$z \in [0,1], z \in \mathbb{Z}$
Binary Special Ordered Set	$\sum_{k \in K} x_k = 1, x_k \in \{0,1\} (k \in K)$
$A_1 x \in \boldsymbol{b}_1 \vee \ldots \vee A_d x \in \boldsymbol{b}_d$	z is a BSOS, $F_k(x) \geqslant \underline{F}_k(1 - z_k)$ $F_k(x) = \begin{pmatrix} A_k x - \underline{b}_k \\ \bar{b}_k - A_k x \end{pmatrix}$
$P_1 \vee \ldots \vee P_K$	$x_1 + \ldots + x_K \geqslant 1$
$P_1 \wedge \ldots \wedge P_K$	$x_k = 1$ for $k = 1 \ldots K$
$P_1 \Leftrightarrow P_2$	$x_1 = x_2$
$P_1 \Rightarrow P_2$	$x_1 \leqslant x_2$
$P_1 \vee \ldots \vee P_K \Rightarrow P_{K+1} \vee \ldots \vee P_L$	$x_k \leqslant x_{K+1} + \ldots + x_L$ for $k = 1 \ldots K$
$A\,x \in \boldsymbol{a}$ if $B\,x < b$	$A\,x \in \boldsymbol{a} \vee (B\,x)_1 \geqslant b_1 \vee \ldots \vee (B\,x)_n \geqslant b_n$
$a^T x \leqslant \min_{i=1 \ldots d} (A\,x - b)_i$	$a^T x \leqslant (A\,x - b)_i$ for $i = 1 \ldots d$
$a^T x \geqslant \min_{i=1 \ldots d} (A\,x - b)_i$	$a^T x \geqslant (A\,x - b)_1 \vee \ldots \vee a^T x \geqslant (A\,x - b)_d$
the components of x_K are distinct integers	$x_k \in \mathbb{Z}, \lvert x_j - x_k \rvert \geqslant 1$ for $j, k \in K, j \neq k$

The End