A Short Survey on Deterministic Global Optimisation

Prof Jörg Fliege
Department of Mathematical Sciences
University of Southampton, UK

ESTEC, November 2017

Introduction

Example: design of a control system.

1. Choose a parametric form of the control law.
2. Implement a function which can evaluate the performance of any control law.
3. Optimise this function with respect to the control law parameters (using some optimisation tools).

Introduction

More formally:

p: parameters of the control, to be chosen by us.

ℓ: local loss function

$$
\begin{aligned}
\min_{p} \quad & \frac{1}{N}\sum_{i=1}^{n} \ell(x_t, p) \\
\text{subject to} \quad & x_{t+1} = f(x_t, p), \qquad \text{(dynamics)} \\
& g(p) \le 0, \qquad \text{(bounds on parameters)} \\
& h(p) = 0, \qquad \text{(coupling between parameters)}
\end{aligned}
$$

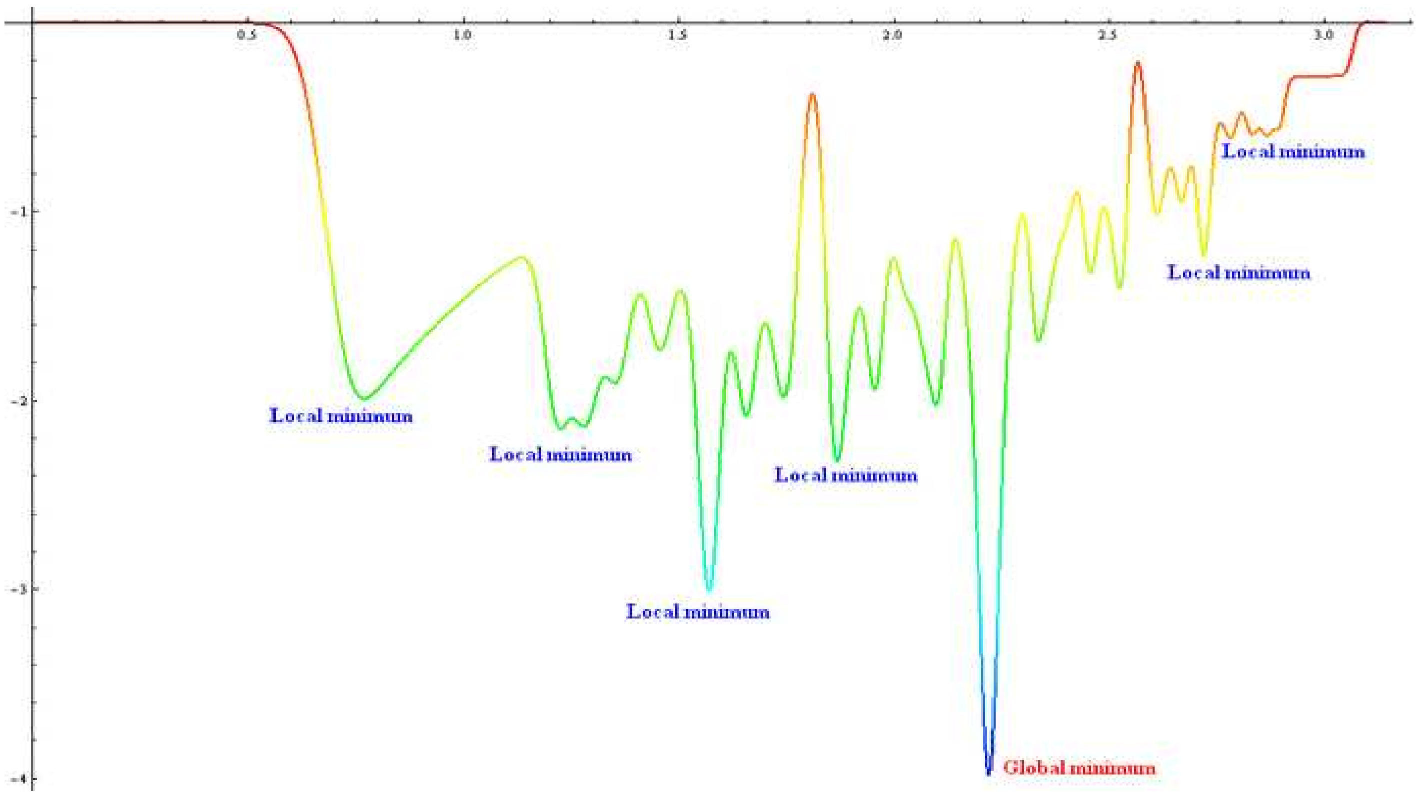
Introduction

Further examples:

- Supervised learning. (Linear regression, logistic regression, support vector machines, neural networks, . . .)
- Supervised learning. (Expectation-maximisation, radial basis function networks, topological data analysis, clustering, . . .)
- Trajectory optimisation.
- Material design, spacecraft design, . . .

Introduction to Global Optimisation

In global optimisation, you want to find a global optimum to your optimisation problem, in contrast to a local optimum.



Introduction to Global Optimisation: Caveats

- Classical nonlinear methods only check if a candidate point fulfils necessary conditions of first order (KKT-conditions). Global optimisation aims for more, and is vastly more difficult.
- It is not always clear if one really needs a global optimum all the time: these optima can be very sensitive to changes of parameters —i. e they can vanish under slight perturbations of the parameters. It is sometimes better to accept a local minimum that is more stable.

Introduction to Global Optimisation

- **No free lunch theorem:** No single algorithm will work well on all problems.
- **Curse of dimensionality:** Solving problems with n variables often costs 2^n operations. (And if you use a stochastic method, the probability of finding a solution shrinks quickly as n grows.)
- **But:** Problem-specific knowledge can help tremendously to tailor algorithms.
- **But, but:** This makes algorithms hard to compare.

What is/isn't Deterministic Global Optimisation?

- **What its not:** an approach to generate high-accuracy solutions. (Thats what local optimisation methods are for, employed after a global optimisation step: WORHP, SNOPT, IPOPT, etc.)
- **What its not:** an approach to generate some approximation to a solution with some (unknown!) probability of correctness. (Thats what stochastic algorithms do.)

What is/isn't Deterministic Global Optimisation?

- **What it is:** an approach that provides *mathematically rigorous* bounds on the global optimum, i. e. $3.753 \leq x_1 \leq 3.755$, $2.161 \leq x_2 \leq 2.167$, ...
- **What it is:** an approach that provides *mathematically rigorous* bounds on where global optima are *not* to be found!
- Can handle equality constraints and inequality constraints as well.
- Can handle integer/binary and general continuous variables.
- Can handle multiobjective problems.

Branch-and-Bound (B&B)

Most efficient deterministic global optimisation algorithms use a **branch-and-bound approach**.

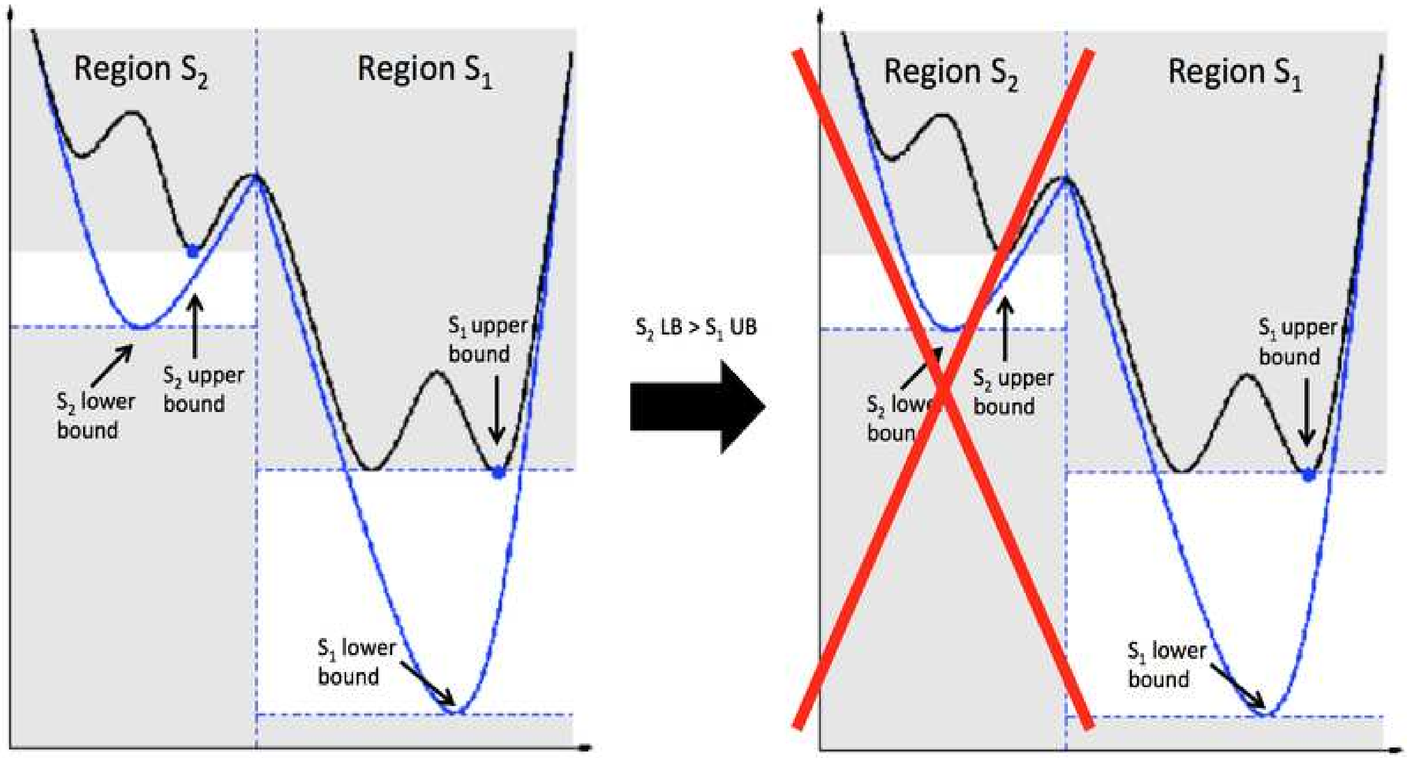
Branch-and-bound algorithms for solving general global problems are not fast (i.e. polynomial-time) in the worst case.

However, if correctly implemented, they can perform very well in practice for medium-sized problems.

Branch-and-bound

Branch: decompose the design space into two parts or more.

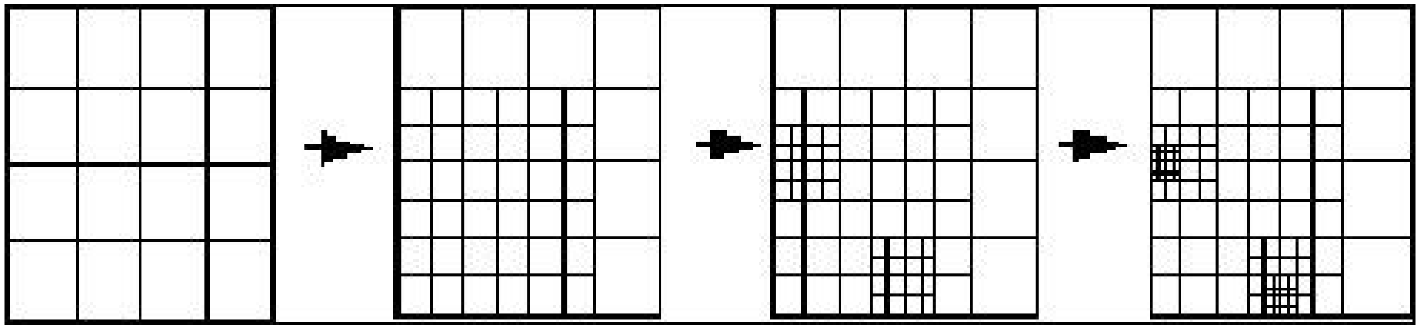
Bounds: find bounds on the optimal solution of smaller problems.



Branch-and-bound

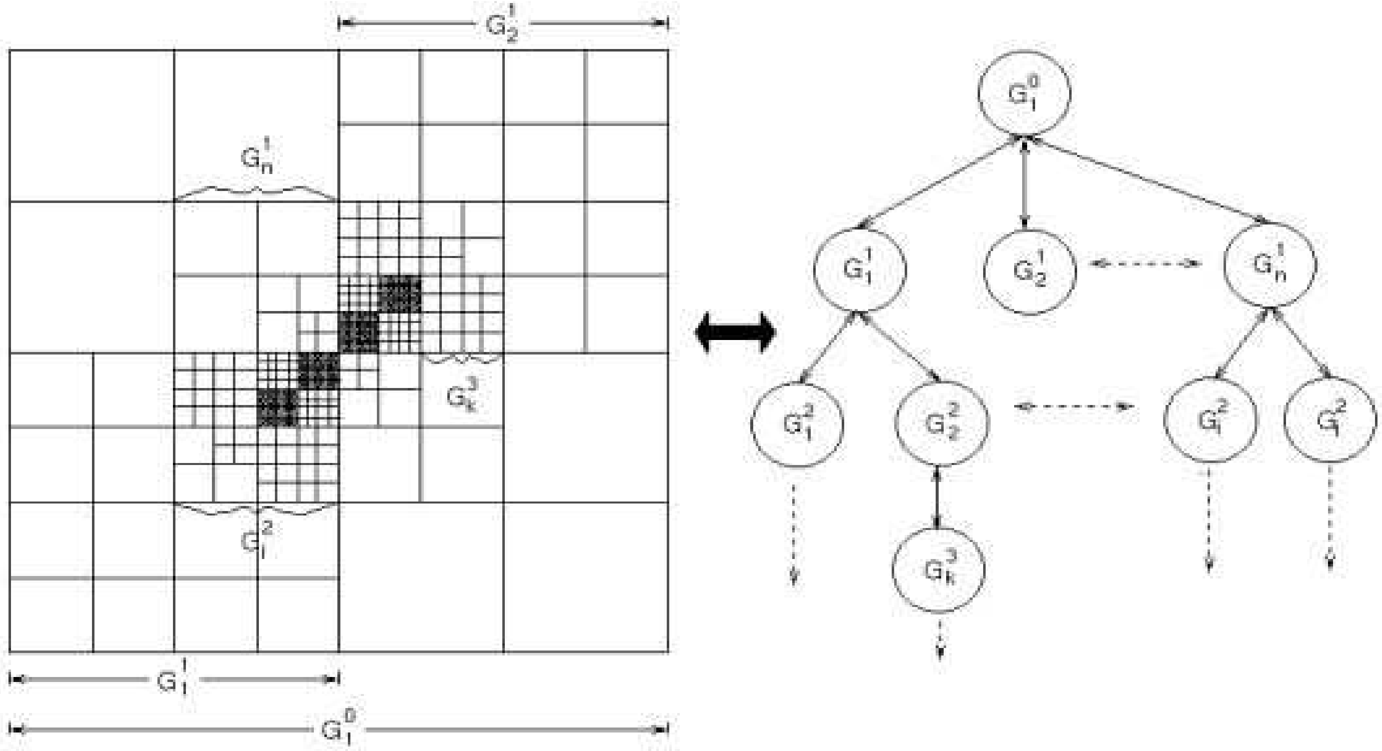
Refinement: employ this process recursively on remaining problems.

Problems considered become smaller and smaller in search space.



Branch-and-bound

The overall process is stored in a tree-like structure. Establishing that a subproblem cannot lead to a global optimum means *pruning* the growth of the tree.



Branch-and-bound

- This search tree can grow exponentially in size —needs good bounds for pruning!
- Branch & Bound is sometimes referred to as the ”prayer algorithm”: start it and pray that you don’t run out of memory!
- Typical failure mode in practice: memory. Global optimum found, but not all remaining branches of the tree pruned.

Bounding

Let S be a region of the design space considered in one the subproblems.

Let f be the objective function.

We need to find good bounds B with

$$\min_{x \in S} f(x) \;\geq\; B.$$

If B is large, we can prune the subproblem over S.

Needs advanced mathematical techniques, because

- Low-quality bounds mean we don't know that we can prune.
- We don't want to solve $\min_{x \in S} f(x)$. Too expensive!

Bounding

Find *good* bounds *quickly*. Approaches:

- Relaxation
- Convexification
- Outer Approximation
- DC Programming
- Lipschitz Optimisation
- Interval Arithmetic
- Duality
- Facets & special cuts
- ...

Which is the right one for your specific problem?

Problem Types, Codes, Problem Sizes

What problems can we solve nowadays?

problem type	some codes	size [vars]
Local optima only	WORHP, SNOPT, IPOPT	$\sim 10,000,000$
mixed-integer linear/ quadratic/conic/SDP	CPLEX, Gurobi, XPress, Mosek	$\sim 100,000$
mixed-integer convex nonlinear	αECP, Bonmin, DICOPT, MINLPBB	$\sim 10,000$
mixed-integer general nonlinear	BARON, KNITRO, Couenne, Lindo	$\sim 1,000$

Caveats: indicative results as in 'orders of magnitude' only. 2-3 hours computation time on standard desktop. Generic problem libraries. YMMV. Solver list incomplete.

Deterministic Global Optimisation: Conclusions

- Problems with $\sim 1,000$ variables reliably solved by off-the-shelf software.
- Still orders of magnitude behind linear/quadratic optimisation.
- Orders of magnitude behind local optimisation: the curse of dimensionality holds. No hope to break it. (Same for stochastic methods.)
- But: problem structures occuring in control and space engineering largely unexplored.
- Large scope for further developments.
- For further details, see report by J. Fliege, W. Coutinho, D. Wassel: *Assessment of Global Optimisation Methods for Space Engineering* (ESA PO 5401001690)