BFO: a simple "brute-force" optimizer (yet another direct search algorithm)

Philippe Toint

Department of Mathematics, University of Namur, Belgium

(philippe.toint@fundp.ac.be)

ORBEL 24, January 2010

The problem

We consider the unconstrained nonlinear programming problem:

$$\text{minimize} \quad f(x)$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$.

Important special case: the nonlinear least-squares problem

Additional constraints:

- there may be bounds on the variables
- derivatives are unavailable
- objective function possibly nonsmooth
- some variables can only assume discrete values

Motivation: parameter tuning in algorithm design (Audet-Orban), but many other examples...

Direct methods for optimization

A long history of direct search methods, including:

- Hookes-Jeeves, Nelder-Mead
- Coope-Price
- Dennis, Torczon, Lewis, Trosset (GPS)
- Dennis, Audet, Abramson (MADS).

This is one more of them

A very simple version:

BFO, a Brute-Force Optimizer

BFO basics

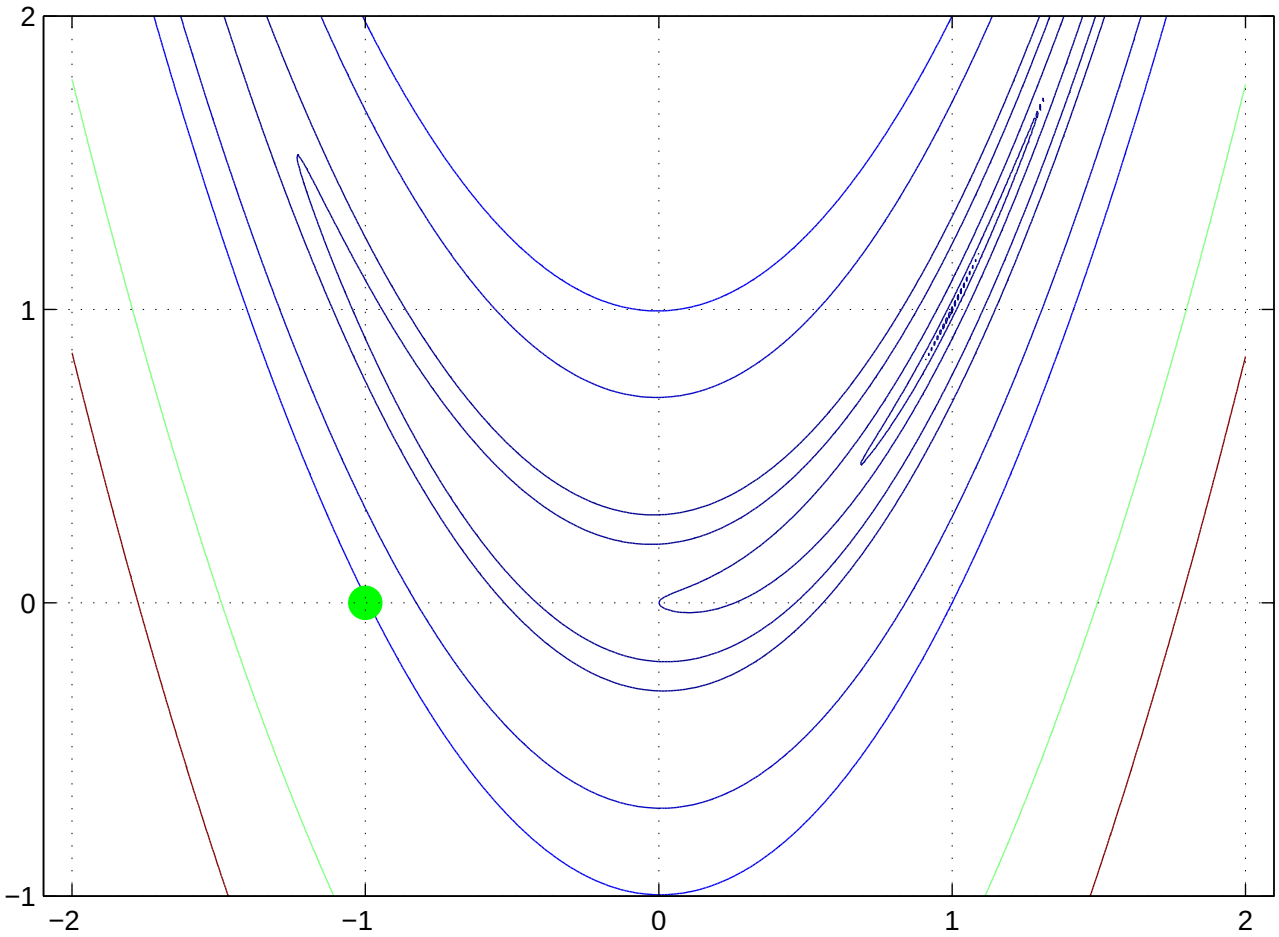
Main features: (not really original)

- Minimizes on progressively finer grids
- Some grid spacing remains fixed to user-specified discrete values
- Only for Local minimization...

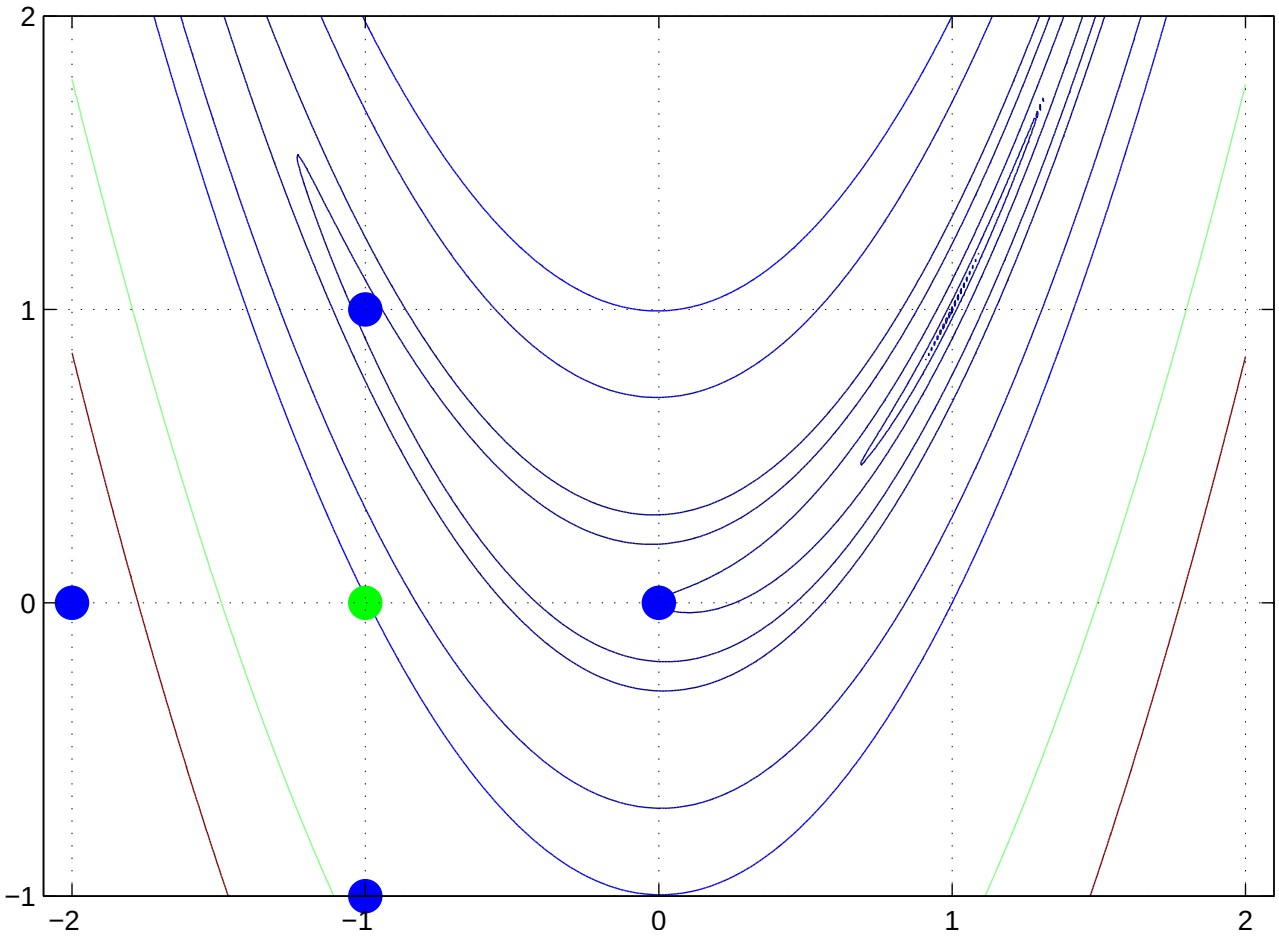
But also

- Attempts to learn from convergence history
- Includes some elements of random search
- User-friendly interface

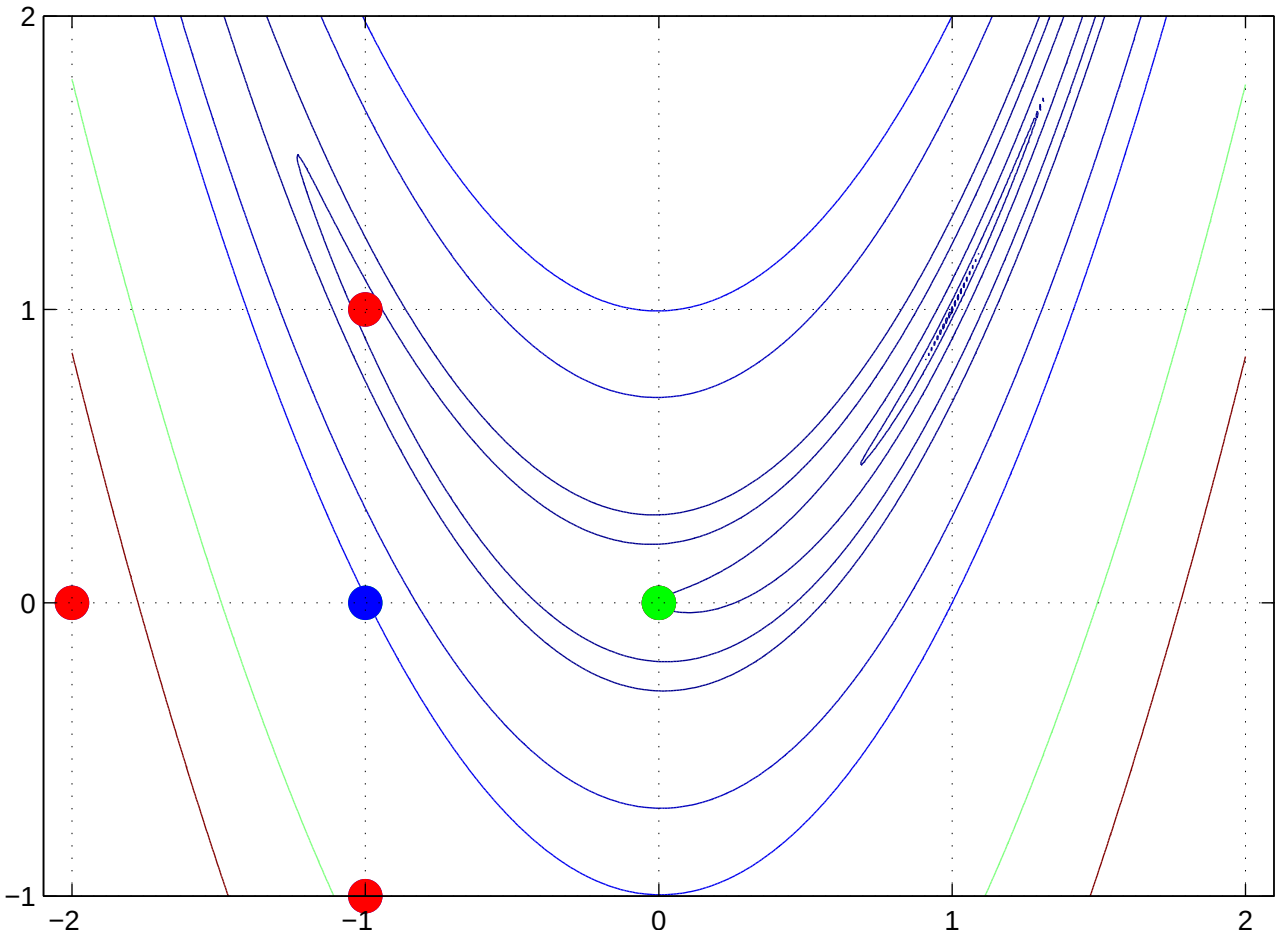
The compass-search on Rosenbrock's function



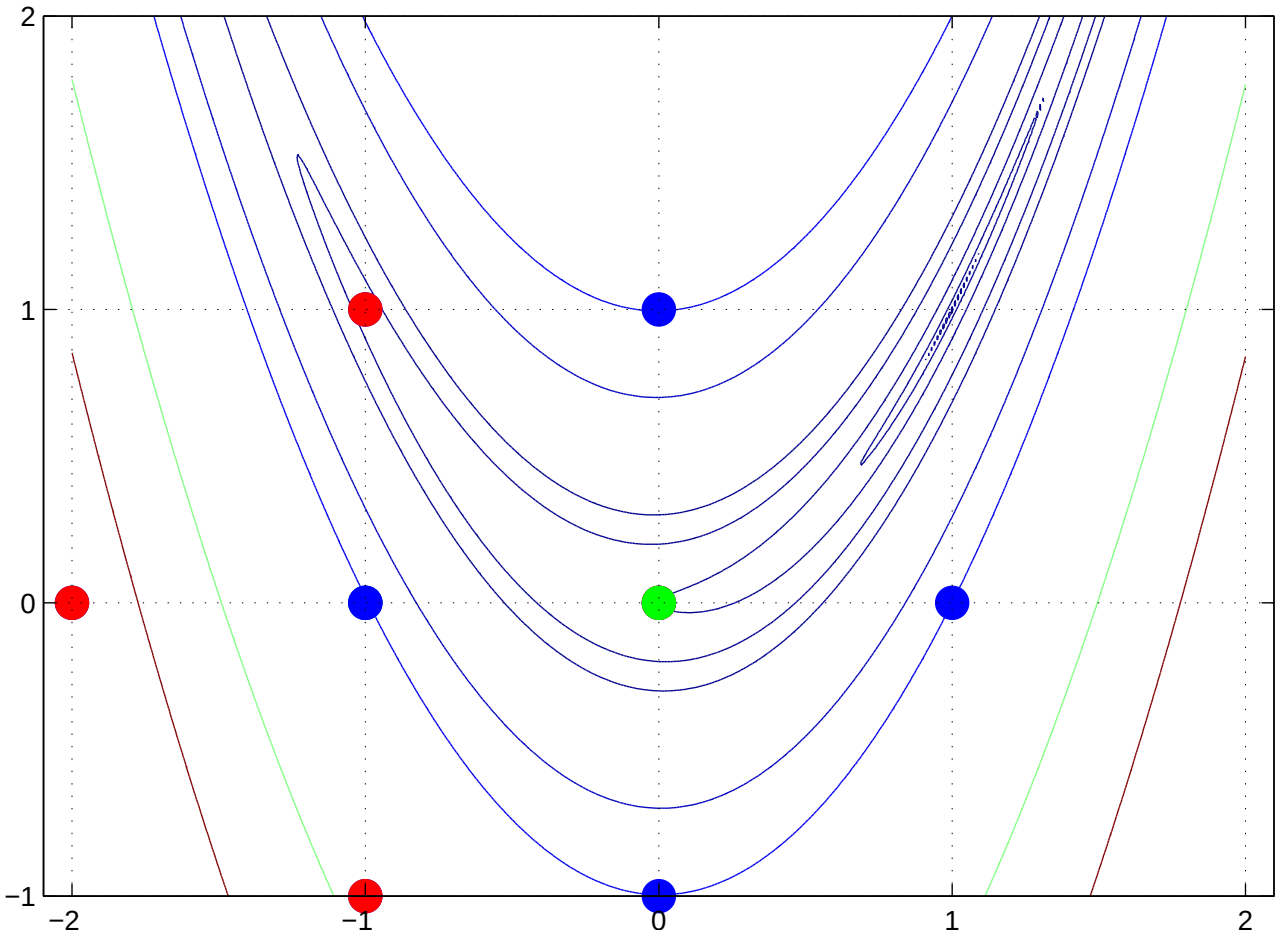
The compass-search on Rosenbrock's function



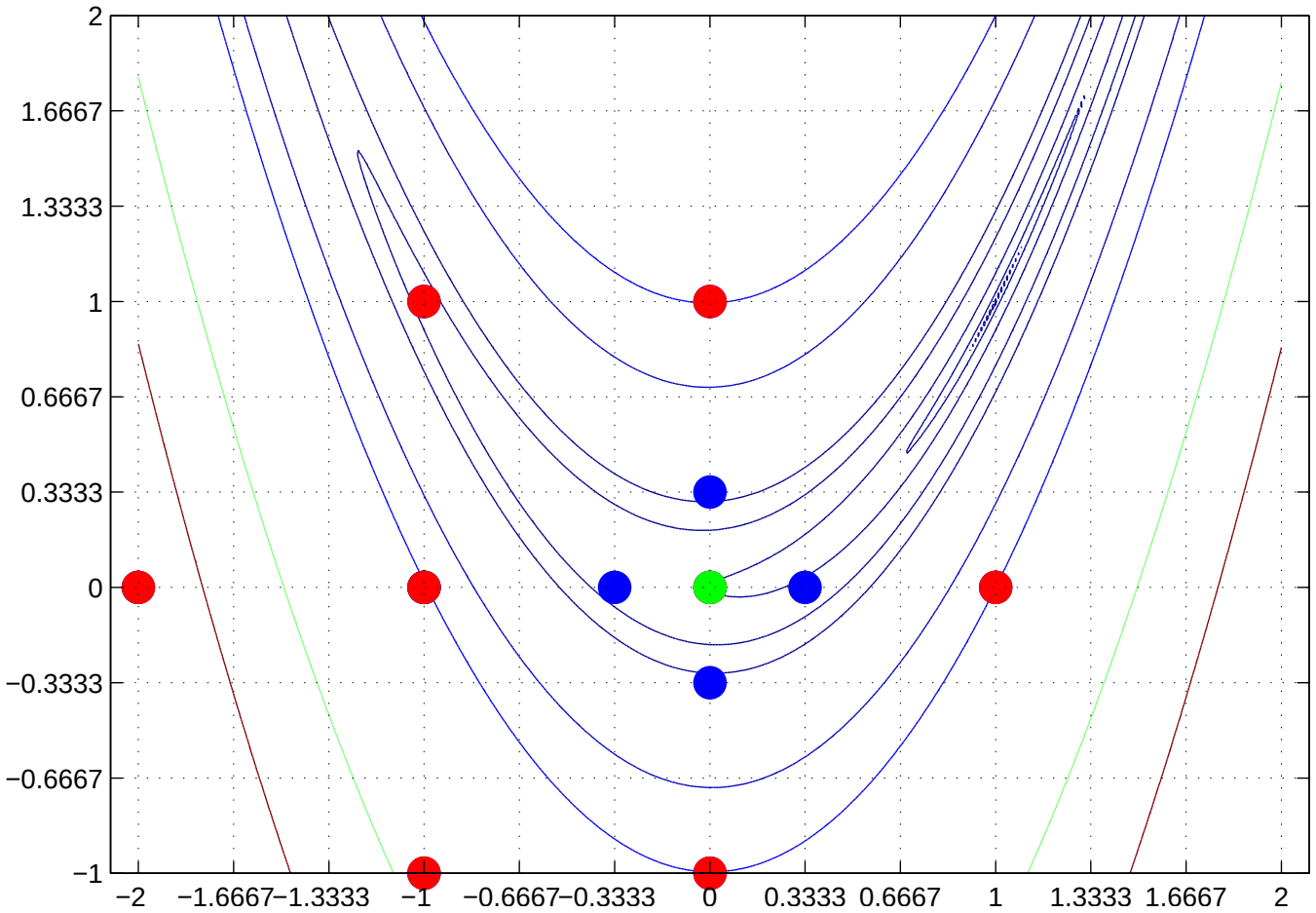
The compass-search on Rosenbrock's function



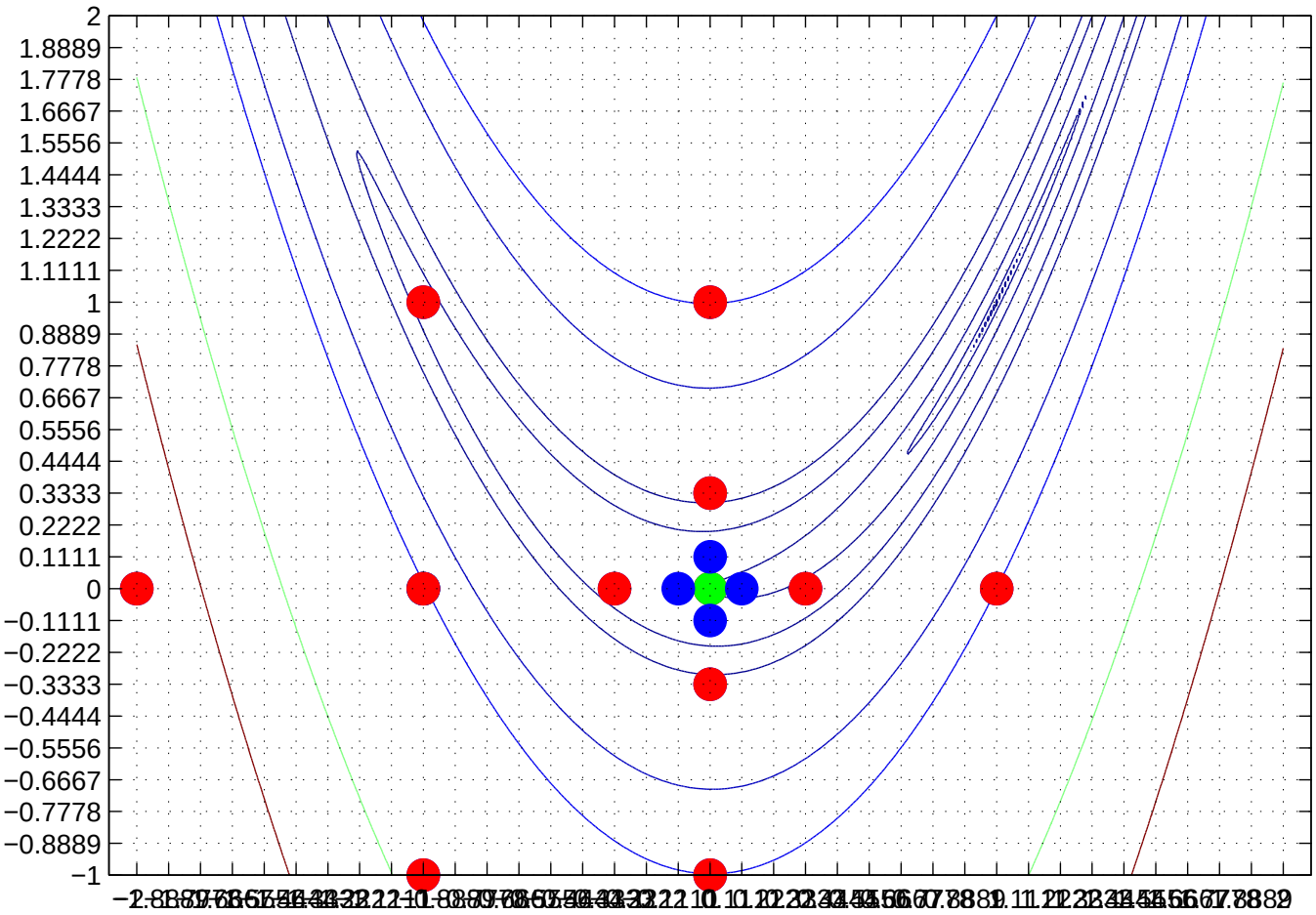
The compass-search on Rosenbrock's function



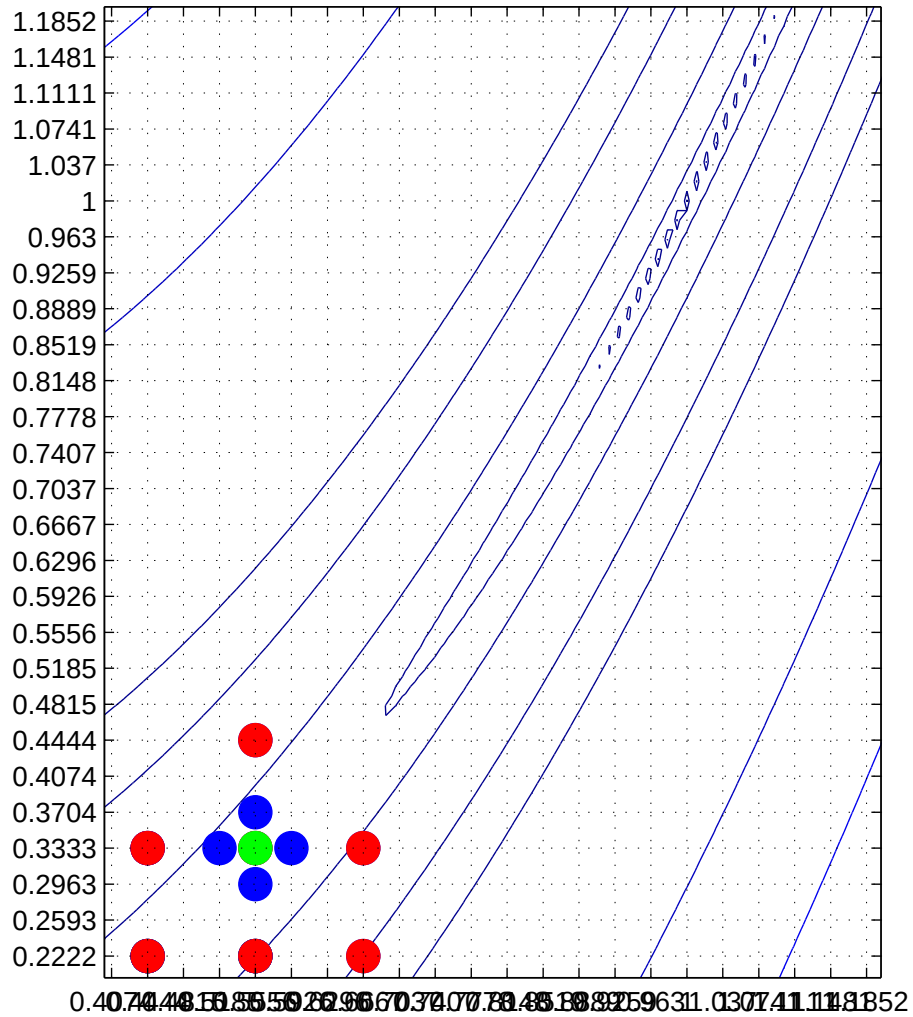
The compass-search on Rosenbrock's function



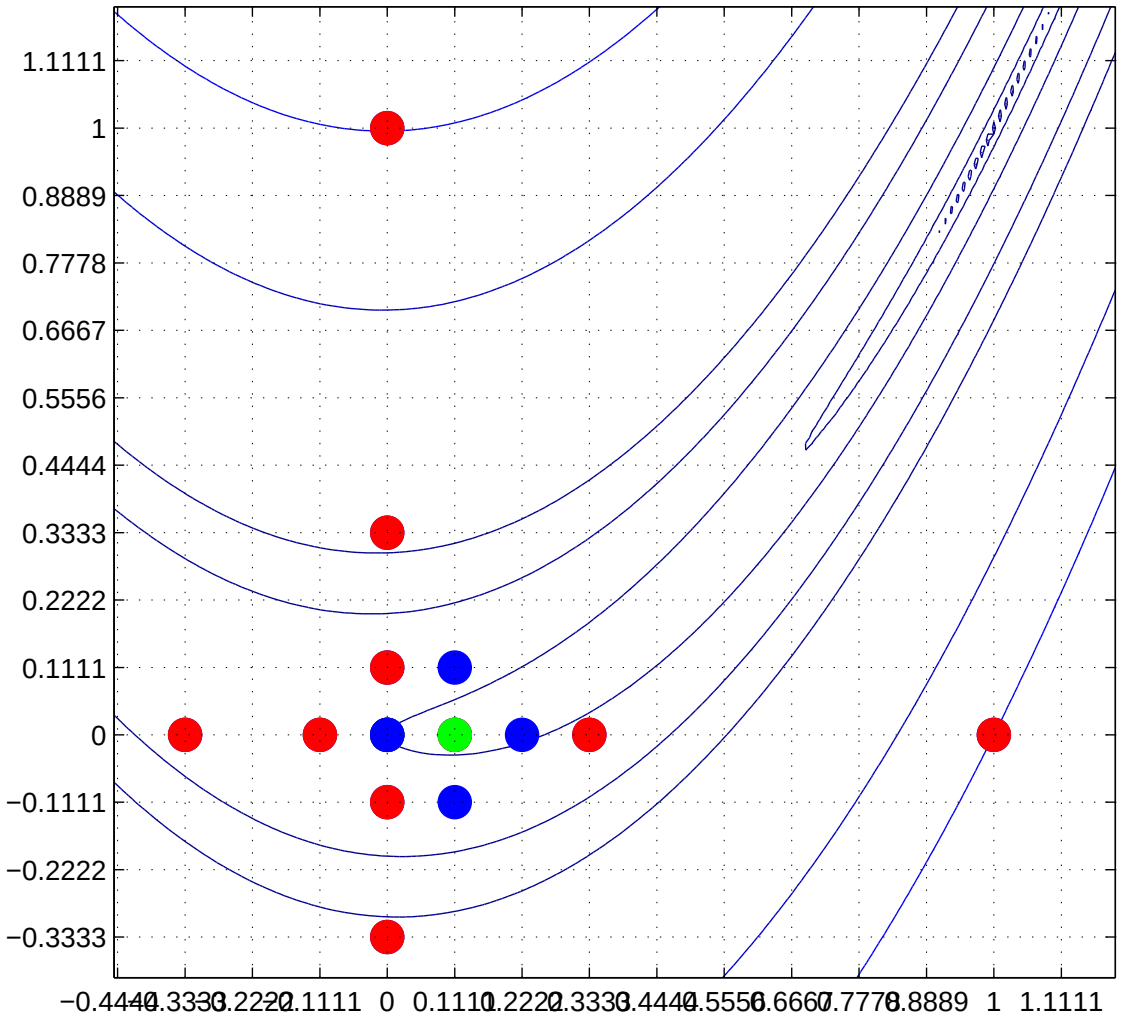
The compass-search on Rosenbrock's function



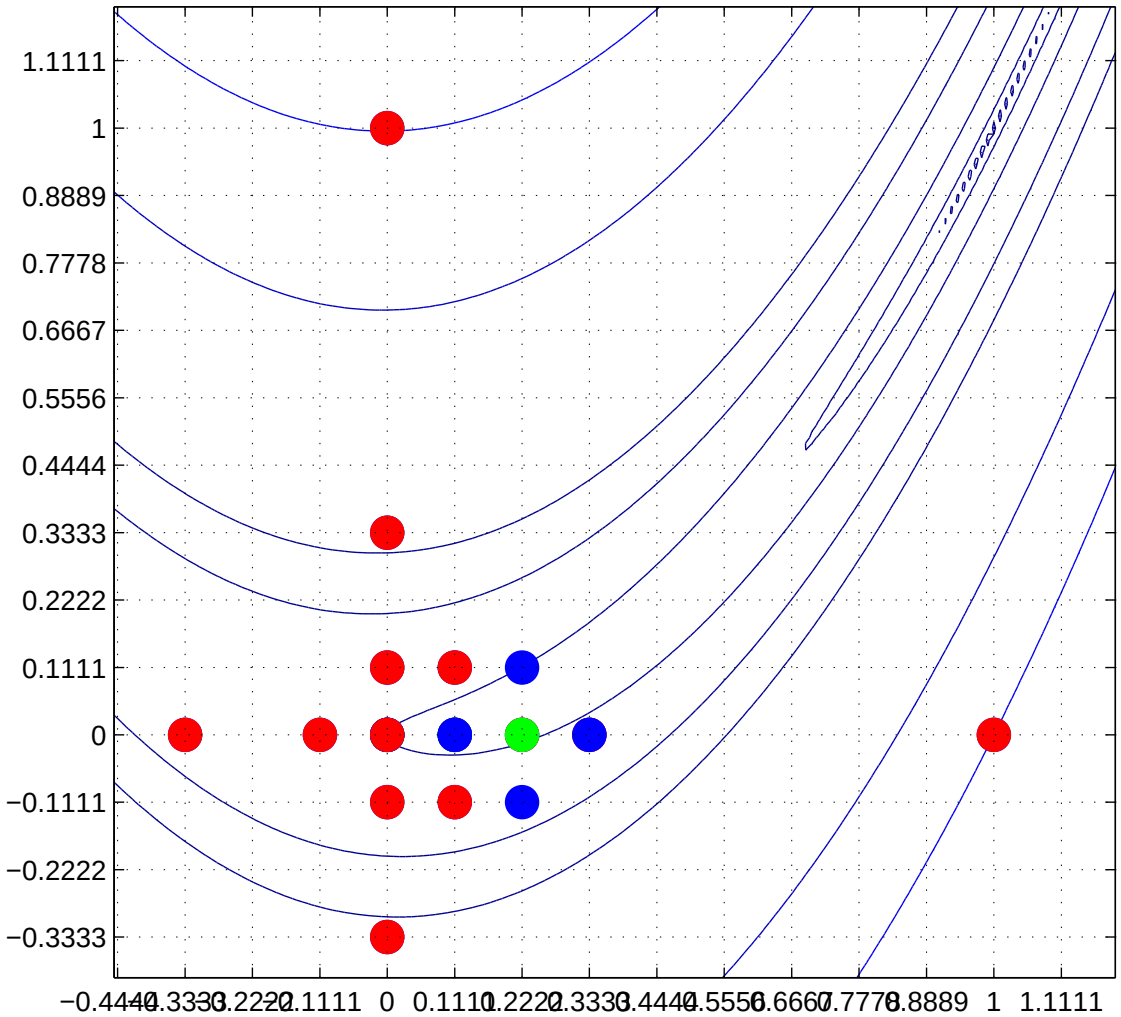
The compass-search on Rosenbrock's function



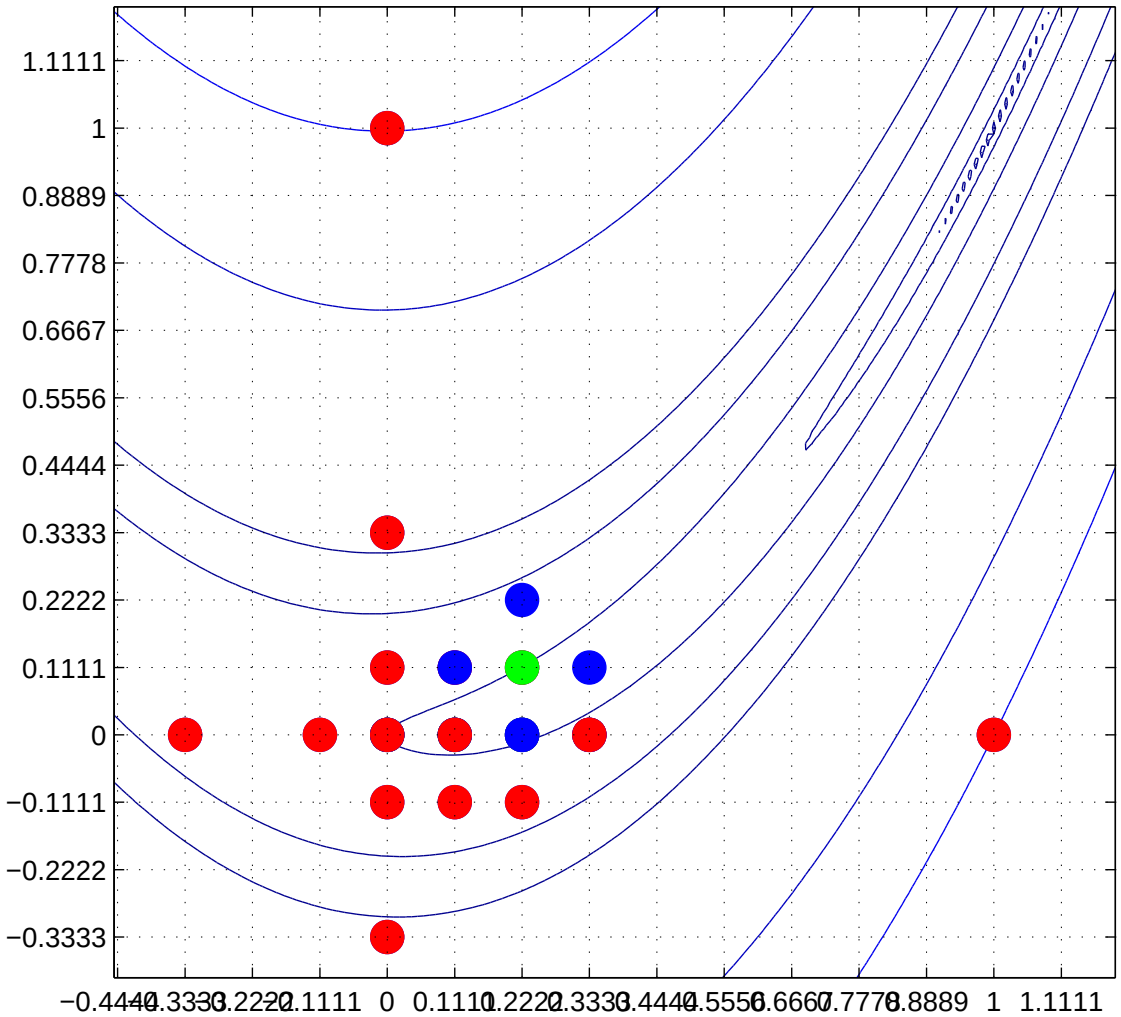
The compass-search on Rosenbrock's function



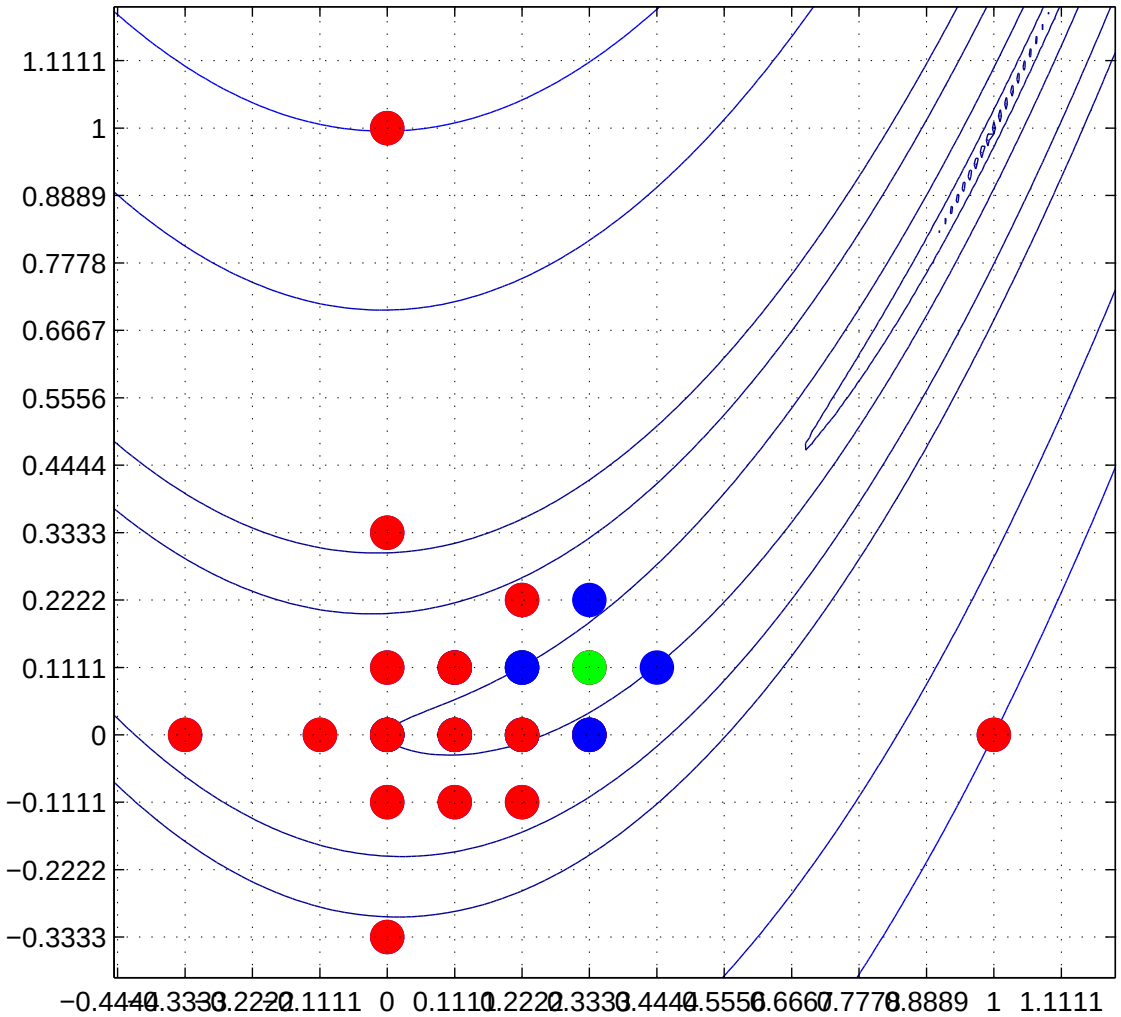
The compass-search on Rosenbrock's function



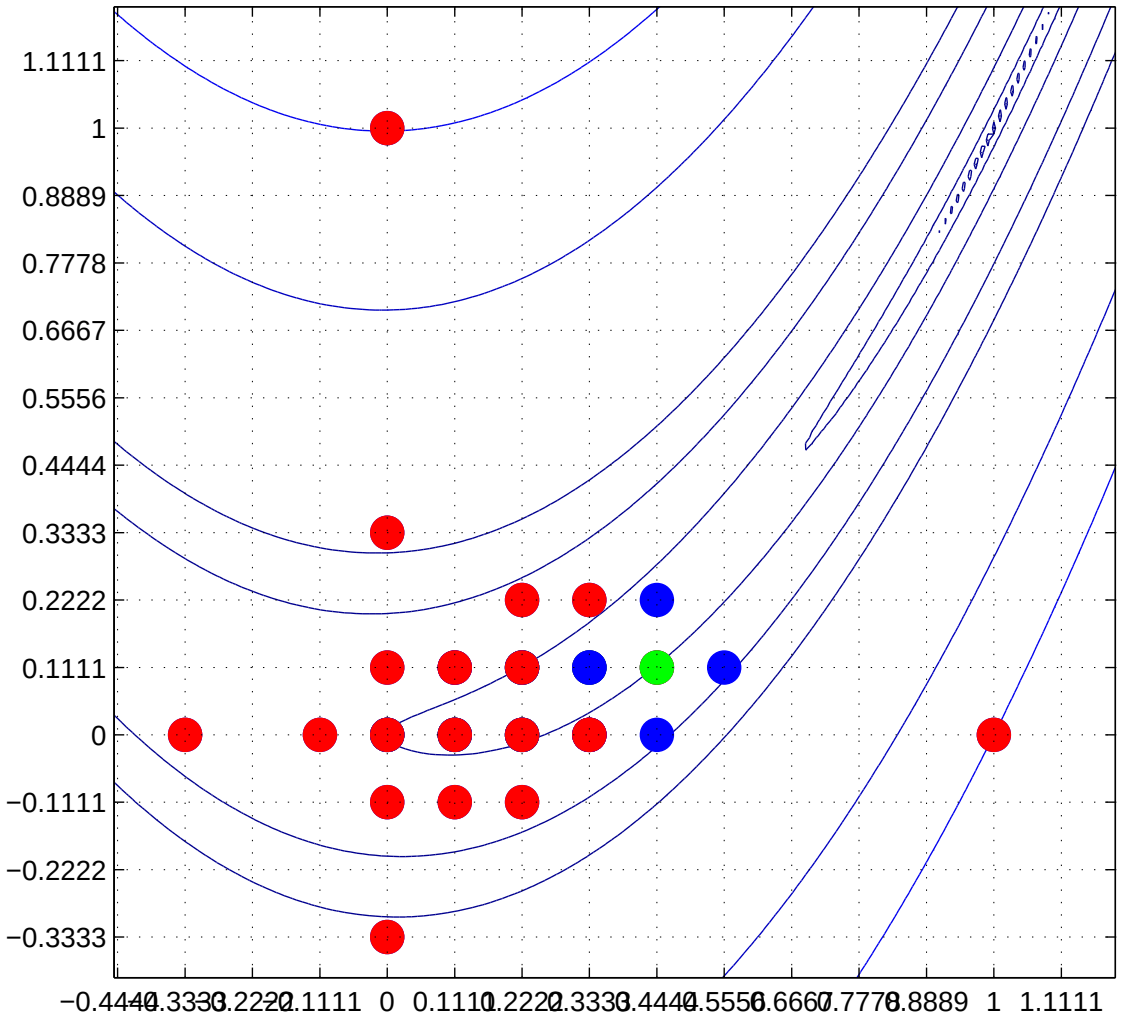
The compass-search on Rosenbrock's function



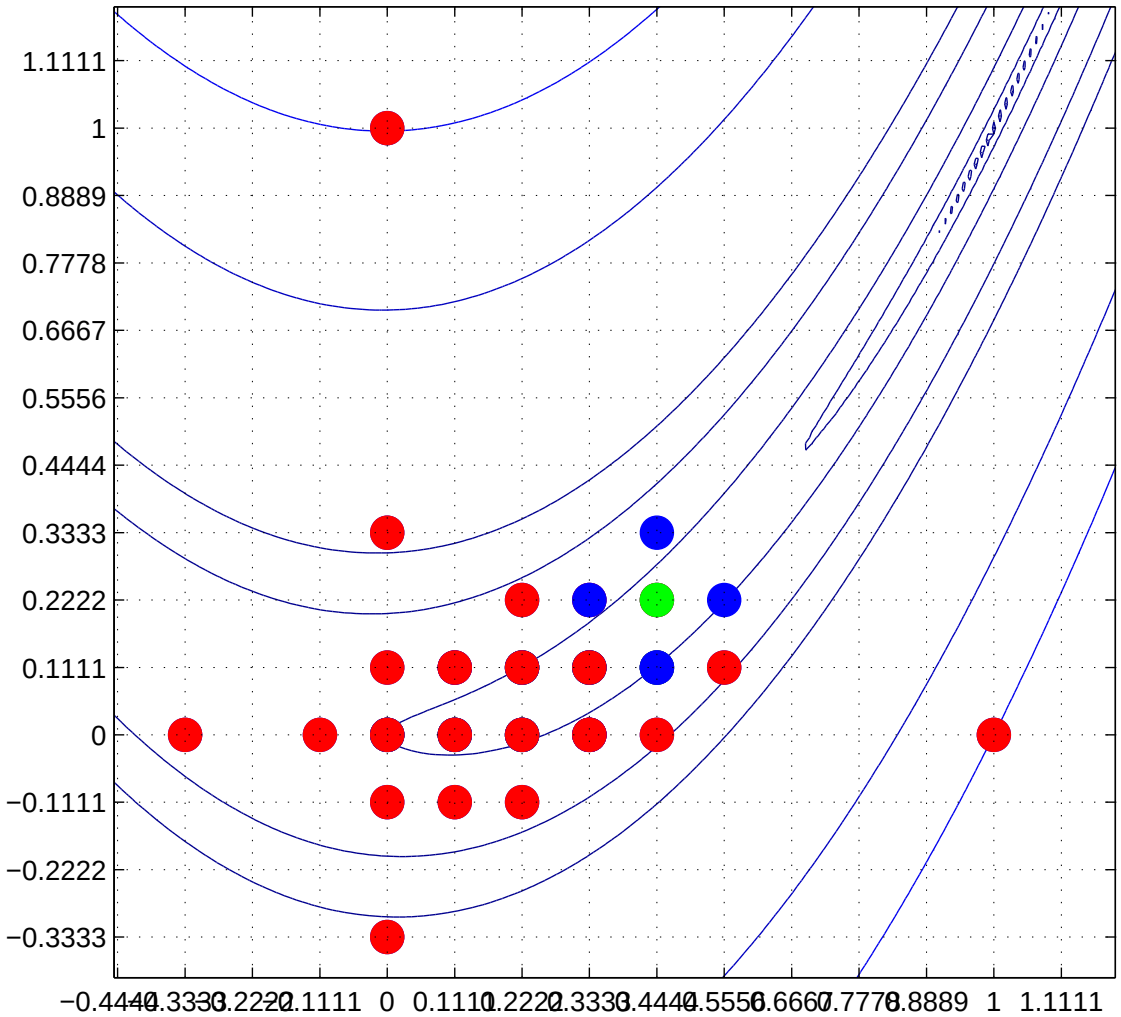
The compass-search on Rosenbrock's function



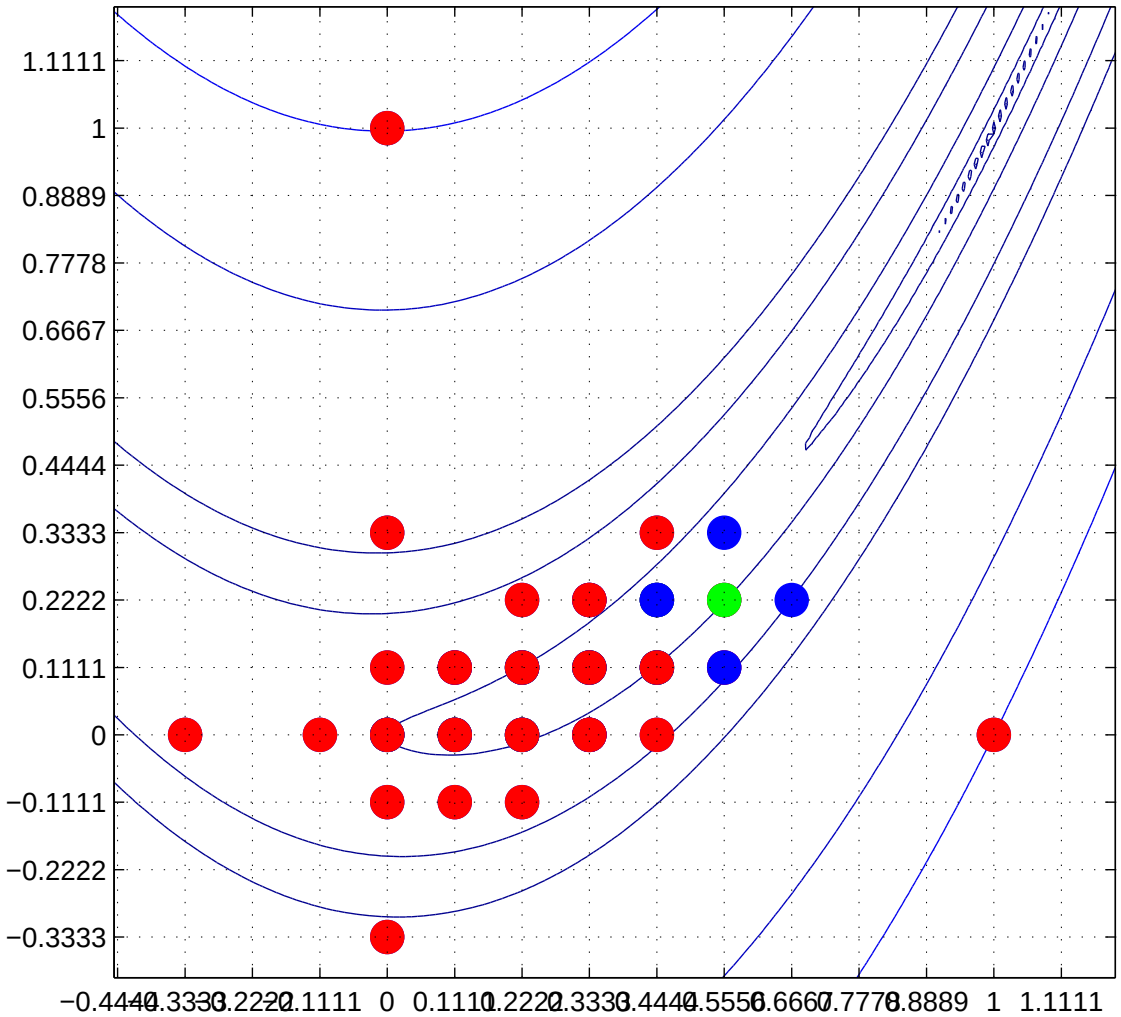
The compass-search on Rosenbrock's function



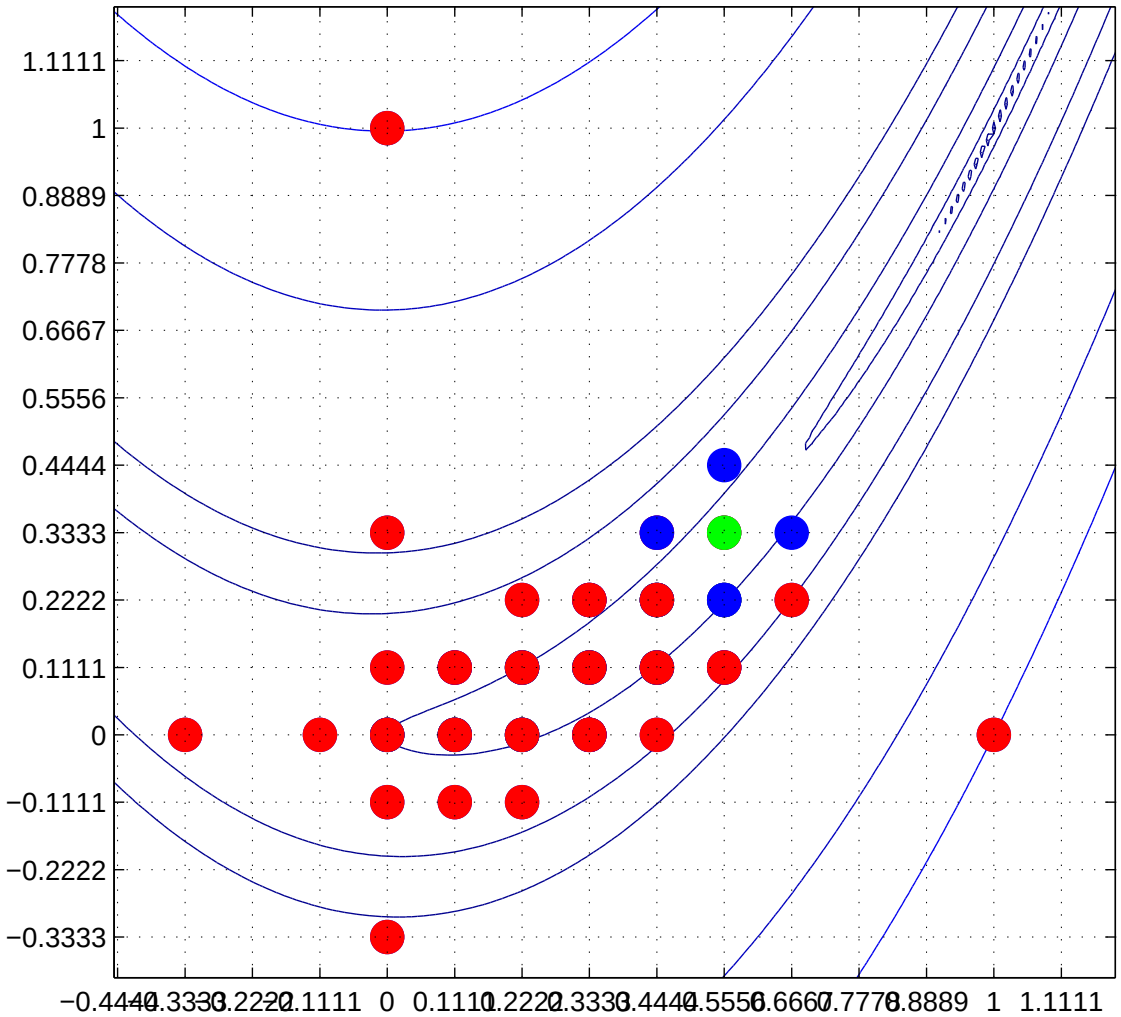
The compass-search on Rosenbrock's function



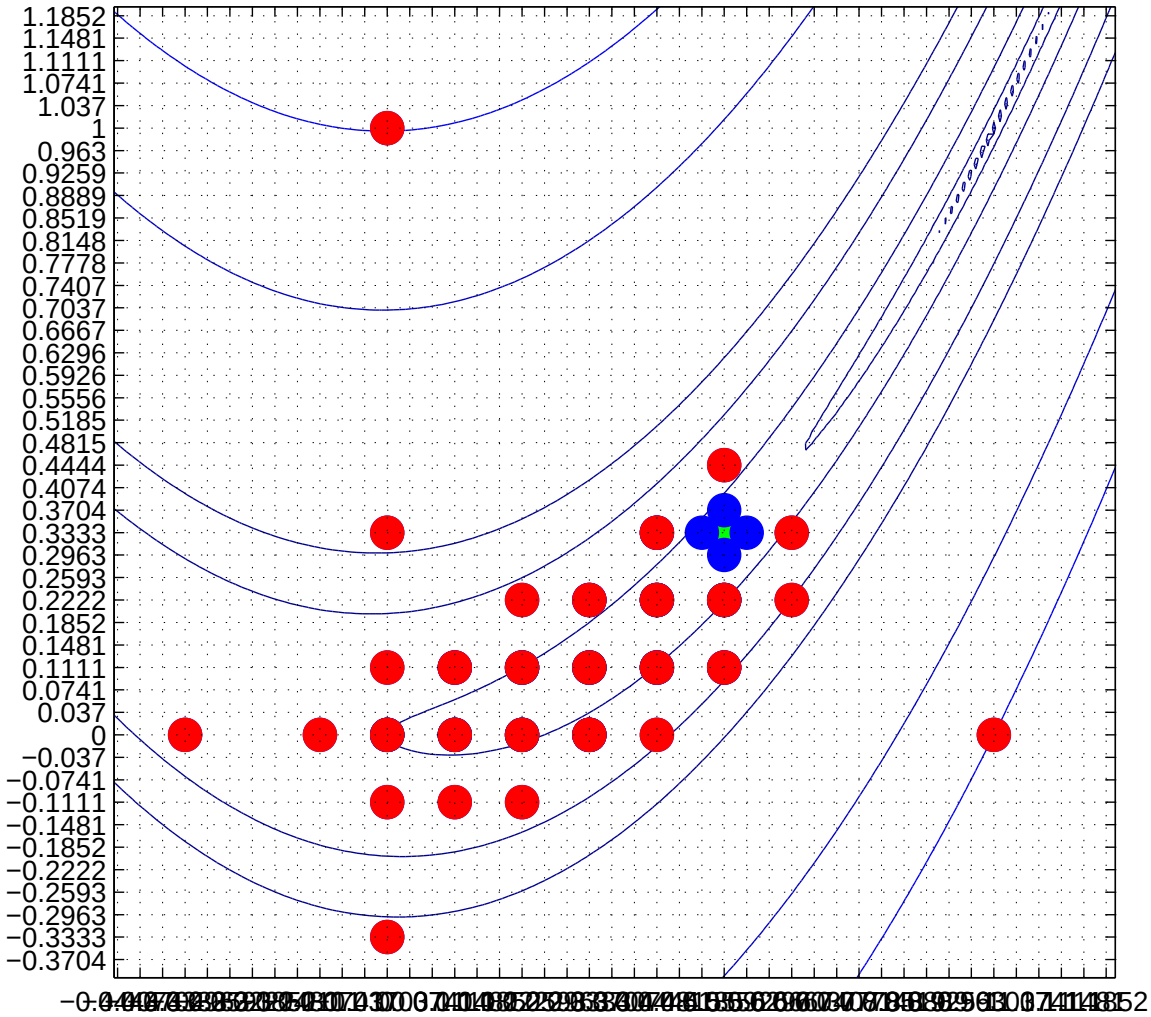
The compass-search on Rosenbrock's function



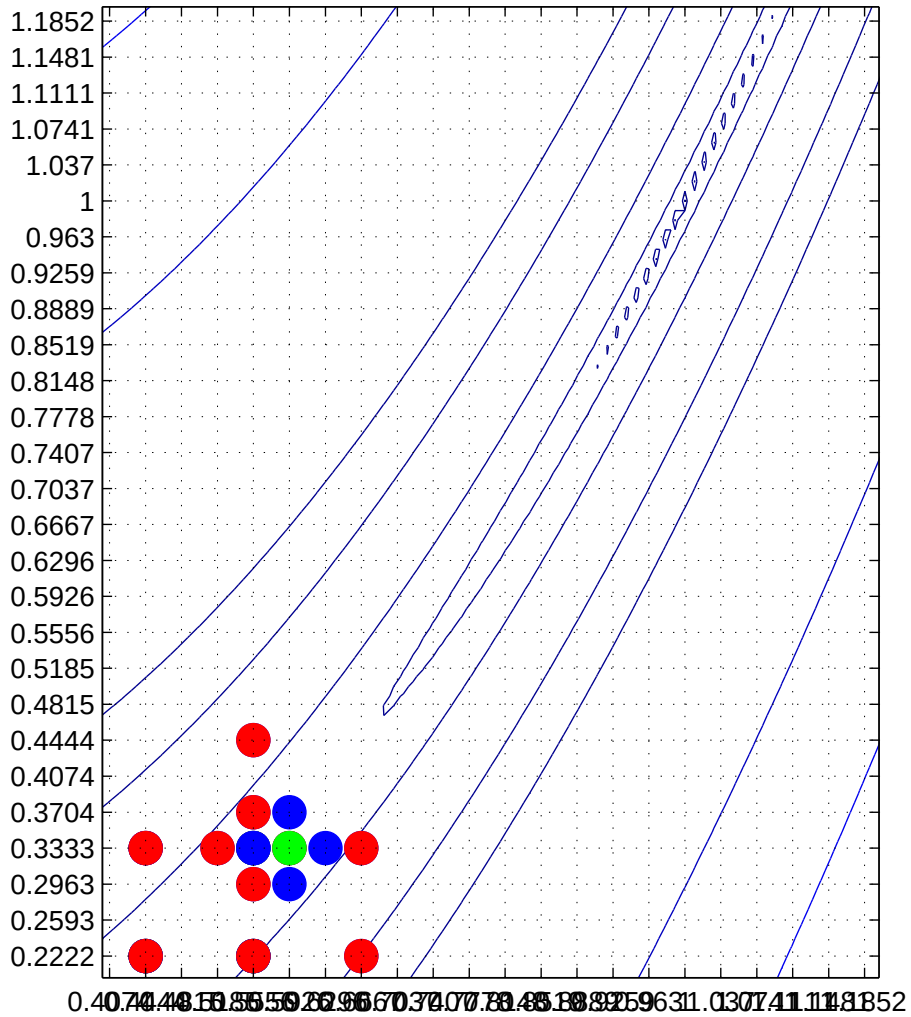
The compass-search on Rosenbrock's function



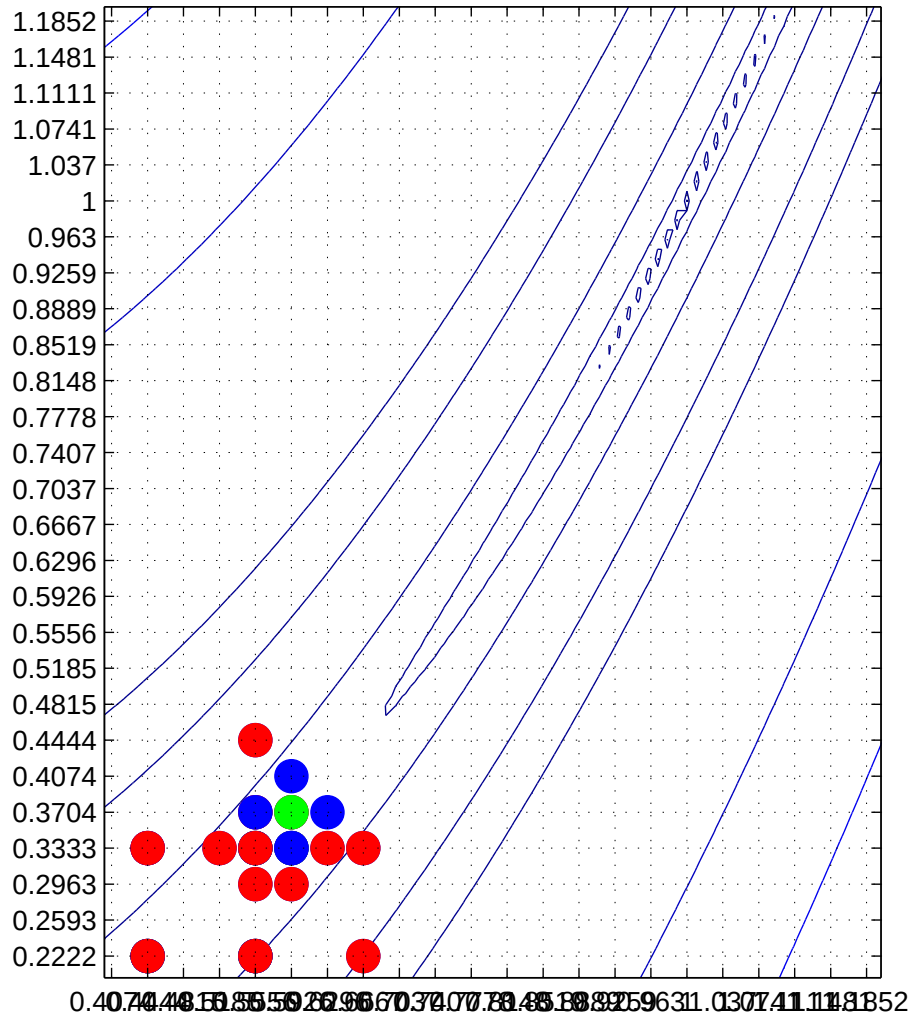
The compass-search on Rosenbrock's function



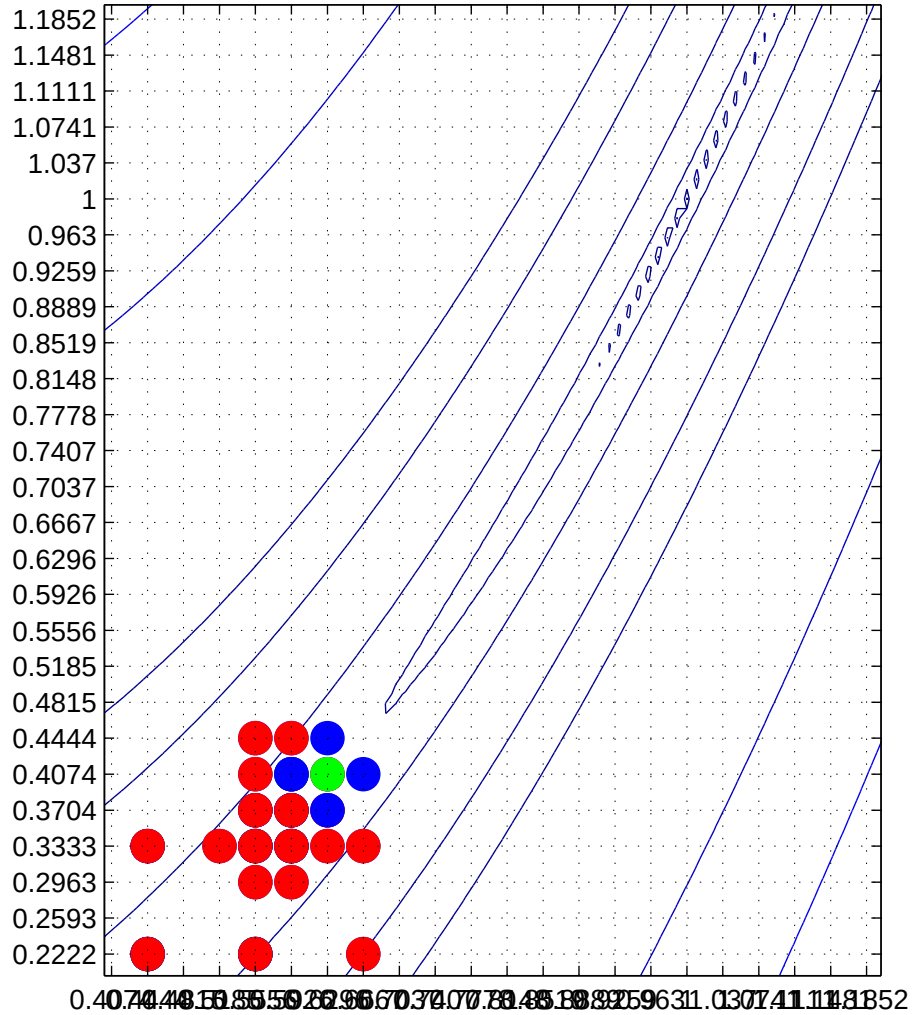
The compass-search on Rosenbrock's function



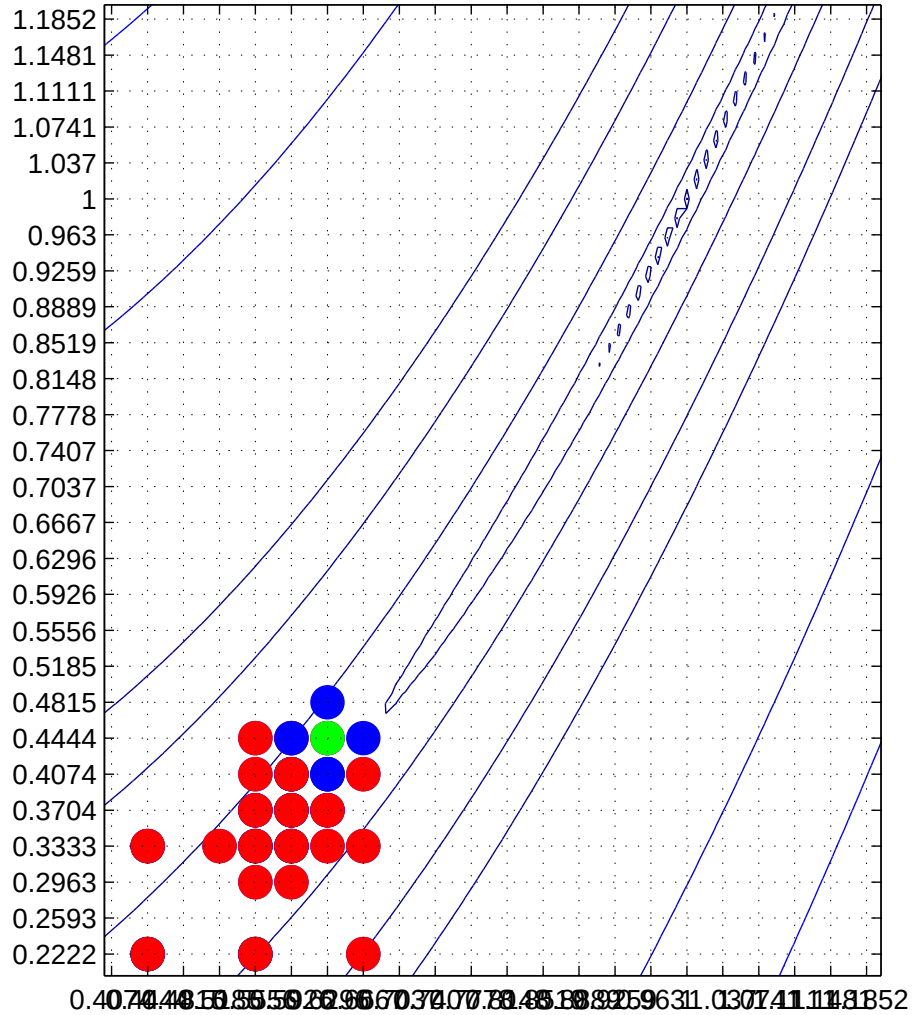
The compass-search on Rosenbrock's function



The compass-search on Rosenbrock's function



The compass-search on Rosenbrock's function



Weaknesses and algorithmic "solutions" (1)

- All compass points evaluated at each iteration
  - stop evaluating as soon as sufficient reduction is obtained
- Painful zigzag crawling when axis not suitable
  - align axis on a grid with progress direction on previous grid (+ random complement)
  - define progress on a user-defined number of past iterations (inertia)
- Compute function at infeasible points
  - keep track of active and nearly-active bound constraints

Weaknesses and algorithmic "solutions" (2)

- Slow progress on fine grids
  - expand the (continuous) grid on "successful iterations"
- Minima in neighbouring discrete subspaces not close to each other
  - recursively explore a depth-first tree of discrete subspaces
  - keep track of record value in each such subspace to avoid re-exploration

Algorithmic parameters

BFO: the Brute-Force Optimizer

User may specify (amongst others):

- grid shrinking/expansion factors
- inertia for defining progress directions
- initial scale in continuous variables

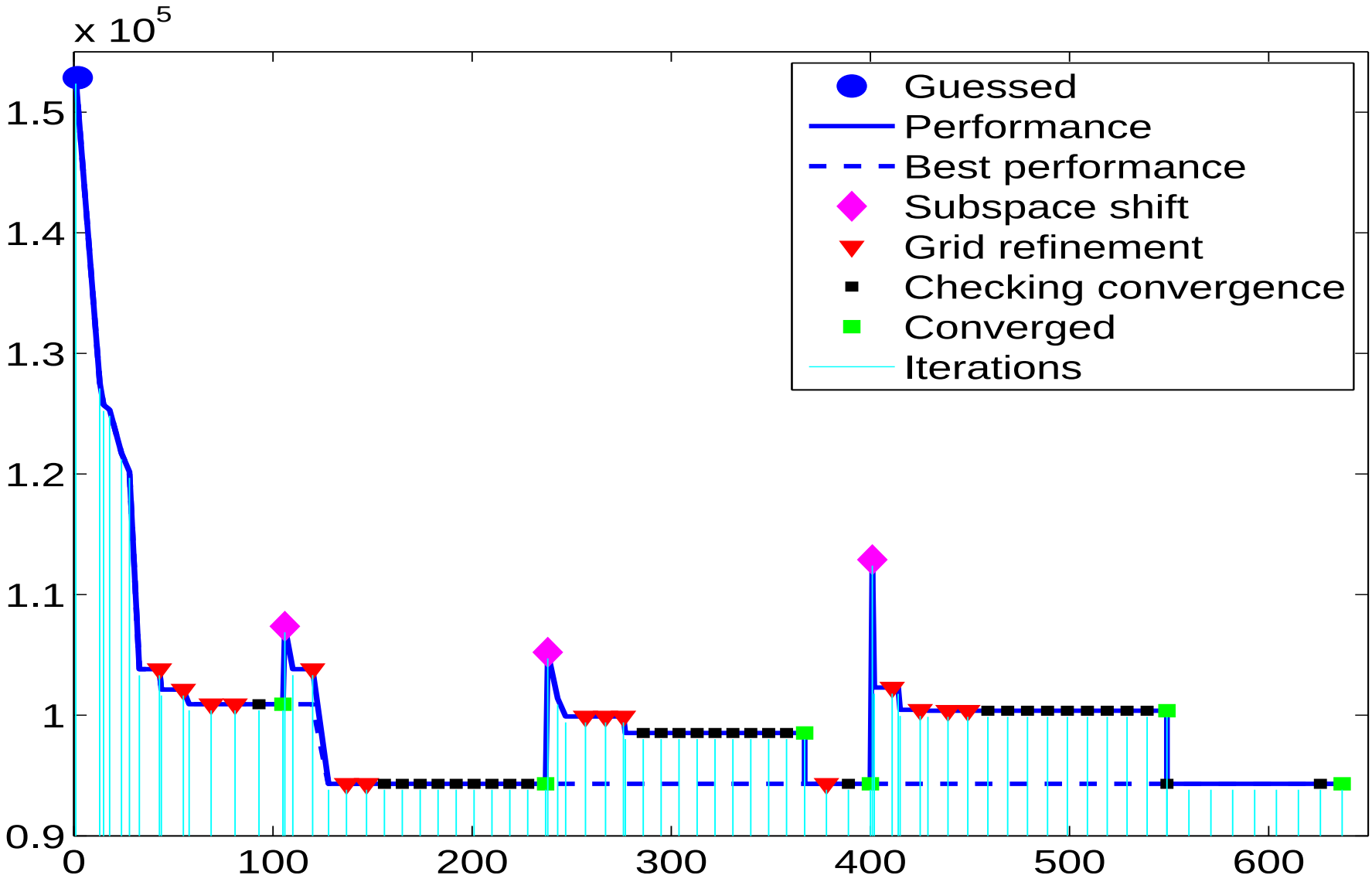
How to choose suitable defaults?

Use BFO!

The setting for algorithm tuning

- a set of 54 test problems
- define objective function value = total number of evaluation to solve all problems
- choose (reasonable) initial default values
- . . . let BFO optimize for finding better values!
  (problem in 5 continuous and 1 discrete variables)

637 iterations of performance optimization



Additional package features

Implementation features:

- check-pointing at user-specified frequency
- norms of vector functions t(f(x))
- allows objective functions with user-defined parameters
- allows partial objective computations
- allows randomized termination test
- provides a FD estimate of the (projected) gradient's norm (in the continuous variables) at termination
- very flexible keyword-based calling sequence

Conclusions

- yet another direct method for mixed integer local optimization
- simple compact implementation
- freely-available easy-to-use Matlab package BFO
- useable for algorithm tuning
  - used on other codes (DFO, ...)
  - beware of overfitting!
- more to do (constraints, use of structure, ...)

Thank you for your attention!