Continuous Global Optimization in R

Christoph Bergmeir

Faculty of Information Technology
Monash University

Outline

References

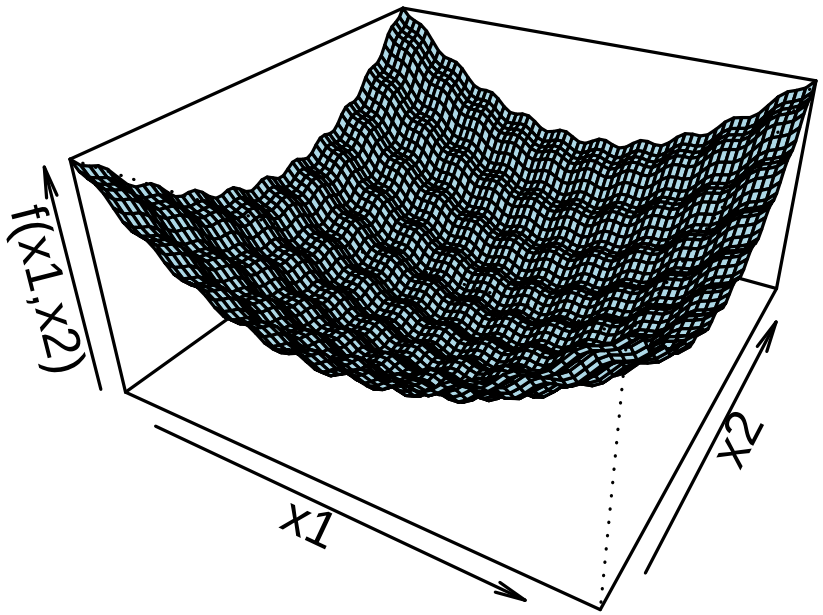
Contents of this talk:

C. Bergmeir, D. Molina, and J.M. Benítez. Memetic Algorithms with Local Search Chains in R: The Rmalschains Package. Journal of Statistical Software, (conditionally accepted for publication).

Journal of Statistical Software, Vol. 60 (2014). Special Volume: Numerical Optimization in R: Beyond optim

Global optimization

- Function minimization (maximization)
- Functions can be convex or non-convex
- Essentially smooth functions
- No constraints, or at most bounds constraints, on the parameters



2 dimensional Rastrigin function

R default capabilities: `optim`

- The function `optim` provides basic optimization capabilities
- It is among the most widely used functions in R
- Methods in `optim` were developed 40 years ago, have known shortcomings
- Nowadays a host of choices exists, see CRAN Task View "Optimization"

Unfortunately [...], the default tools are not best practice, and the model of an aging default collection and an unstructured, largely un-mapped host of contributed packages is at best unattractive. (Nash, 2014)

Methods in `optim` and some shortcomings

- Contains solvers “Nelder-Mead", “BFGS", “CG", “L-BFGS-B", “SANN", (“Brent", for one-dimensional problems only)
- “BFGS", “CG", “L-BFGS-B" use derivatives, solve convex problems.
- “L-BFGS-B”: a newer version of the algorithm was made available by the original authors in the meantime
- “SANN”: *the simulated annealing variant [...] is known to be insufficient in many respects (comment of a reviewer of our paper)*
- “Nelder-Mead": Other implementations of the algorithm exist (at least) in packages **neldermead**, **dfoptim**, **gsl**, **adagio**, and **nloptr**.

State-of-the-art methods in R for global (non-convex) optimization

- `optimx` is a more modern replacement for `optim`.
- CMA-ES: Covariance Matrix Adaptation Evolution Strategy. (packages **cmaes**, **adagio**, and **parma**). Package **cmaes** is arguably a basic implementation that shouldn't be used.
- Differential evolution (packages **DEoptim**, **RcppDE**). **RcppDE** is a port of **DEoptim** from C to C++ (using **Rcpp**). Same results, claimed to be faster.
- Generalized Simulated Annealing (package **GenSA**)
- Genetic algorithms (package **rgenoud**)
- MA-LS-Chains (package **Rmalschains**)

Rmalschains

- **Rmalschains** implements the MA-LS-Chains algorithm family
- Core functionality is in C++, with wrapper code in **Rcpp** and R
- Memetic Algorithms with Local Search Chains (MA-LS-Chains)
- Memetic algorithms combine genetic algorithms with local search.
- MA-LS-Chains: Local search (LS) is applied to individuals for a defined number of iterations. Current state of LS is then saved and possibly continued at a later stage $\Rightarrow$ chaining.
- LS can be applied with more intensity on promising individuals.
- Proved effective in competitions, also for high-dimensional problems.

Speed tweaks: Implement in C/C++

- Many solvers are internally implemented in C/C++ (`optim`, **DEoptim**, **RcppDE**, **Rmalschains**).
- Often, the objective function will also be a C/C++ implementation.
- A lot of performance gets lost by going through R for the function calls of the objective function.
- `optim`, **RcppDE**, and **Rmalschains** allow for direct calls within this process, which can speed up things a lot.
- see **RcppDE** demo “compiled" or **Rmalschains** demo “rastrigin_inline"
- for `optim`, see “Writing R Extensions, Section 6.8". There are C functions: `nmmin`, `vmmin`, `cgmin`, `lbfgsb`, `samin`.

Comparison with other Methods

Test suite:

- Test suite of 19 scalable functions (Rosenbrock, Rastrigin, Schwefel, Sphere, etc.)
- Problem dimensions 2, 10, 30, 50, 100, 200, 500, 1000

A disclaimer:

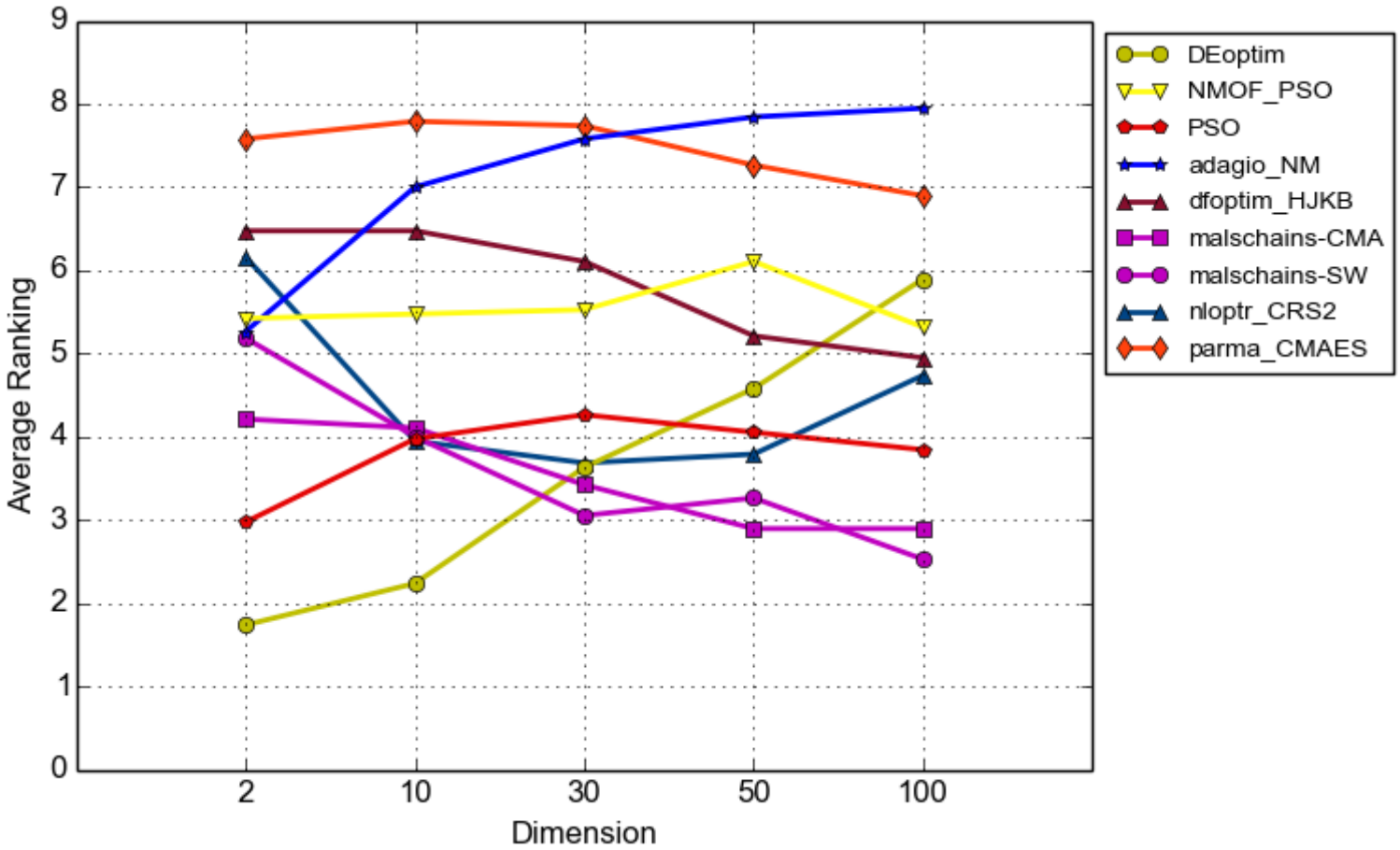
- Methods usually have a host of control setting
- These setting can influence the performance dramatically
- Methods are used with default settings in most comparisons (also here)

Comparison with other Methods - Execution Time

Algorithm\Dim	5	10	30	50	100	200	500	1000
adagio_NM	68.06	254.56	13954.30	31057.45	123799.00	679064.70	-T-	–
DEoptim	402.30	770.45	2727.22	5138.34	12972.36	37580.78	177020.90	656181.60
RcppDE	287.83	322.06	1044.56	2515.88	4917.35	14383.89	85628.93	361631.10
nloptr_CRS2	322.90	413.01	2450.26	6668.90	29349.60	140109.90	-T-	–
parma_CMAES	844.36	2481.49	11397.93	22843.63	89434.72	-T-	–	–
dfoptim_HJKB	9.51	22.26	59.07	100.07	592.74	1809.29	4615.43	58617.32
malschains-CMA	44.85	137.69	888.95	7188.50	47237.20	352899.50	-T-	–
malschains-SW	29.14	108.08	440.32	1085.85	5693.48	17961.84	121082.20	570921.00
optim_BFGS	1.85	3.61	36.66	88.57	462.28	3144.69	11872.38	-E-
optim_NM	3.60	246.36	2705.37	6336.82	17351.25	43599.793	400229.41	-T-
optim_L-BFGS-B	1.77	4.01	61.94	93.80	404.21	1887.32	-E-	–
PSO	1200.48	1427.22	2002.28	2611.18	3934.63	6655.85	15833.53	35383.74
NMOF_PSO	489.87	1041.79	1858.67	2427.66	3686.72	6498.25	12251.77	26849.33
rgenoud	39695.38	-M-	–	–	–	–	–	–
GenSA	216.62	537.95	-M-	–	–	–	–	–

Time (in ms) for each optimization package. The different errors are: T: time limit was reached. M: memory limit was reached. E: program exited with error.

Comparison with other Methods - Ranking



Rmalschains: Indicators of Use

Included in a comparison of optimization methods for a portfolio optimization problem on `www.portfolioprobe.com`, where it performed pretty well.

PortfolioProbe

Investment techi
century

Home About Key Features Demo or buy User Area Contact Blog

← US market portrait 2012 week 30 R Inferno-ism: order

A comparison of some heuristic optimization methods

Posted on 2012/07/23 by Pat

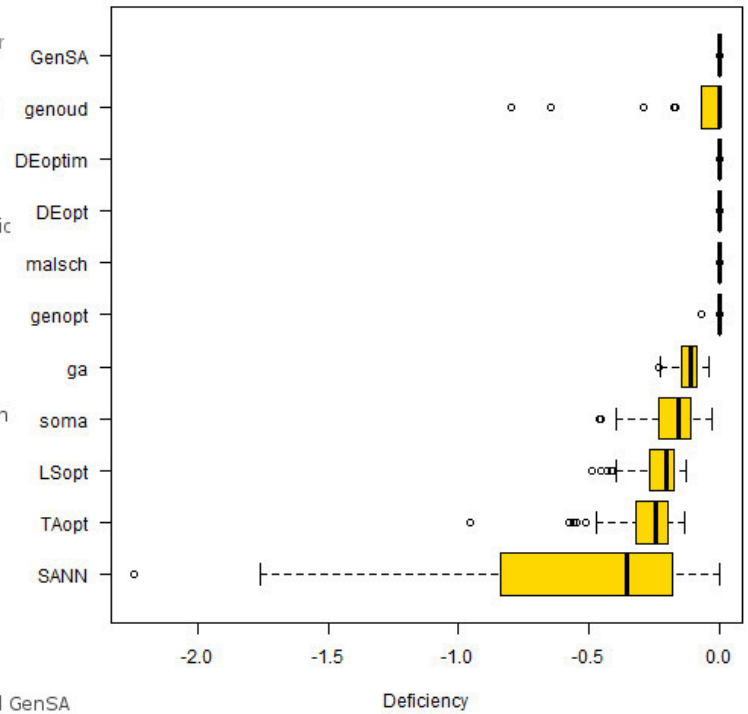
A simple portfolio optimization problem is used to look at several R functic use randomness in various ways to do optimization.

Orientation

Some optimization problems are really hard. In these cases sometimes th

Summary

If your problem is anything like this problem, then the Rmalschains and GenSA packages are worth test driving.



Rmalschains: Indicators of Use

Included in a comparison of optimization methods for a portfolio optimization problem on `www.portfolioprobe.com`, where it performed pretty well.

PortfolioProbe

Investment techi
century

Home About Key Features Demo or buy User Area Contact Blog

← US market portrait 2012 week 30 R Inferno-ism: order

A comparison of some heuristic optimization methods

Posted on 2012/07/23 by Pat

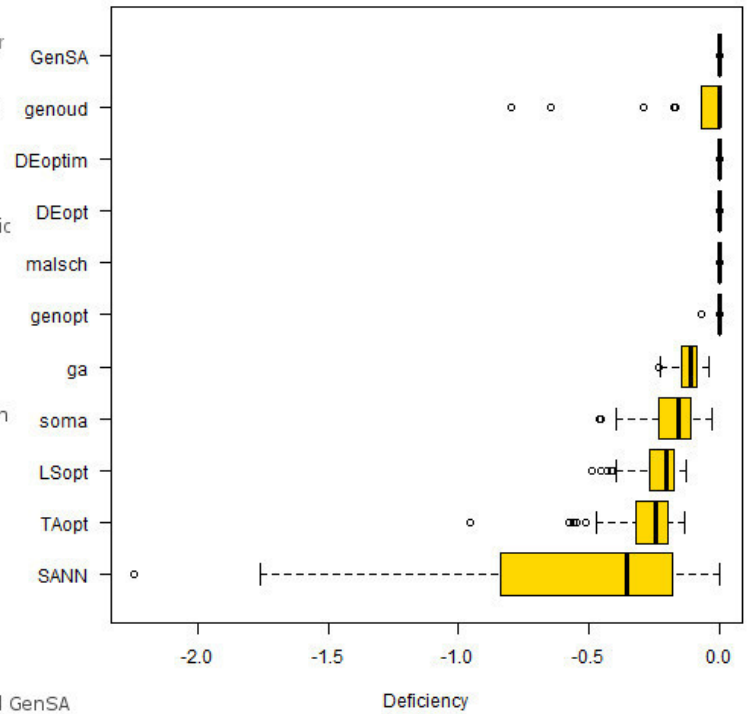
A simple portfolio optimization problem is used to look at several R functic use randomness in various ways to do optimization.

Orientation

Some optimization problems are really hard. In these cases sometimes th approach is to use randomness to get an approximate answer

Summary

If your problem is anything like this problem, then the Rmalschains and GenSA packages are worth test driving.



Conclusions

- `optim` not considered state of the art nowadays.
- Especially for non-convex optimization, a host of other choices is available
- See, e.g., **optimx**, parma::cmaes, **GenSA**, **RcppDE**
- We implemented the package **Rmalschains**, which is also good choice, especially for high-dimensional problems

Thank you

Christoph Bergmeir

christoph.bergmeir@monash.edu

Comparison with other Methods - Ranking (2)