Solving GENOPT Functions with the Reactive Affine Shaker using irace for Parameter Tuning

Learning and Intelligent OptimizatioN Conference LION 10

T. E. KALAYCI

machine Learning and Intelligent OptimizatioN (LION) Research Group
Information Engineering and Computer Science Department
University of Trento

30 May 2016

LION
intelligent-optimization.org

Table of contents

Introduction

- Reactive Search Optimization (RSO): solving complex optimization problems by integrating online machine learning techniques into search heuristics
- Continuous and Cooperative Reactive Search Optimization (CoRSO) (extension of RSO): aims to solve continuous optimization problems by a strategic use of memory and a cooperating team of self-adaptive local searchers
- We have used the Reactive Affine Shaker (RAS) solver, which employs CoRSO methodology, to solve GenOpt functions
- Tuned its parameters using the irace package [1]

[1] The irace Package: Iterated Race for Automatic Algorithm Configuration http://iridia.ulb.ac.be/irace/

The RAS Solver

- A self-adaptive and derivative-free random search algorithm based on function evaluations. Seminal idea was presented for a specific application in neural computation
- An effective strategy during the initial part of the search by analysing the evolution of the search direction in the first iterations
  - when the search succeeds with a very high probability
- Depends on two parameters:
  - multiplier for the size of the initial search region $(0 < \eta \leq 1)$
  - the box expansion/contraction factor $(0 < \rho \leq 1)$

The RAS Solver

- Aggressive local minima searcher: aims to converge rapidly to the local minimizer
- Starts from a random initial point x in the configuration space surrounded by an initial search region $\mathcal{R}$ where the next point along the trajectory is searched for
- In order to keep a low computation overhead, the search region is identified by n vectors $(b_i, ..., b_n)$ which define a "box" around the point x: $\mathcal{R} = \{x + \sum_{i=1}^{n} \alpha_i b_i, \qquad \alpha_i, ..., \alpha_n \in [-1, 1]\}$
- The search occurs by generating points in a stochastic manner, with a uniform probability in the search region

The RAS Solver

- RAS aims at maintaining the search region size as large as possible, while still ensuring that the probability of a success per evaluation will be reasonably close to one
- During the search it uses a "reactive" determination of the search area for the following design objectives:
  - success probability per sample close to one: the area is enlarged if the search is successful, reduced if unsuccessful
  - largest possible step size per successful sample: area is elongated along the last successful direction

The RAS Solver

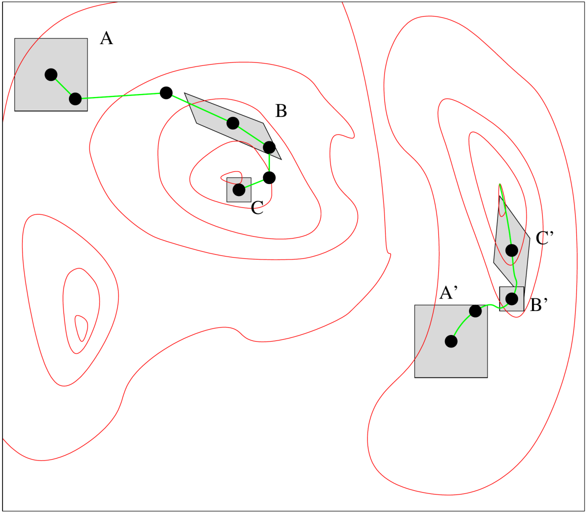

Figure: Two search trajectories leading to two different local minima. The evolution of the search regions is also illustrated

The RAS Solver

- Steps of RAS algorithm:
  1. A new candidate point is generated by **sampling the search region** with a uniform probability distribution and by using the "double shot" strategy
  2. The search region is modified according to outcome of the tentative point
     - It is compressed if the new function value is greater than the current one (unsuccessful sample), it is expanded otherwise (successful sample)
     - The search area defined by vectors b_i undergoes an **affine** transformation
  3. If the sample is successful, the new point becomes the current point, and the search region is translated so that it becomes centred around the new point

The RAS Solver

- As soon as RAS finds a local minimum, the search is continued with a simple way: by restarting from a different initial random point
- We have a population of different searchers
  - they don't have information about each other but working together on the same objective
- Default shaker is **affine shaker**: requires vector-matrix multiplications to update search region
- Our proposed solver can also use **inertial shaker** as an alternative way:
  - The search box is always identified by vectors parallel to the coordinate axes using a single vector *b* without a need to perform matrix multiplications

The RAS Solver

- For the GENOPT contest we used domain limits to define initial random points of searchers
- Searchers start searching around random points of middle of domain limits for each domain variable according to the following equations[2]:

$$X_{m_i} = \frac{(L_i + U_i)}{2} \tag{1}$$

$$\eta_i = \eta * (U_i - L_i) \tag{2}$$

$$X_{s_i} = Rand(X_{m_i} - \eta_i, X_{m_i} + \eta_i) \tag{3}$$

- We take the advantage of domain limits to improve the search quality

[2] L_i is lower boundary and U_i is upper boundary, X_{s_i} is initial point of variable i, where $i \in 1, ..., d$ and d is the number of dimensions, η is taken from user to define initial search region size

The RAS Solver

In summary the proposed solver has following steps:

1. Define a search region for the searchers using η parameter and dimension boundaries
2. Generate random initial X values using the middle points of the domain limits
3. Start the selected searcher (AS or IS) using the region defined and initial X values
4. If the solution is not improved for a number of iterations, restart the searcher (Return to Step 2)
5. Algorithm stopped when the iteration limit is reached

Automatic Parameter Tuning with irace

- The irace package implements the iterated racing procedure
- Automatically configures optimization algorithms by finding the most appropriate settings given a set of tuning instances of an optimization problem
- Offline tuning mechanism with two delimited phases:
  1. an algorithm configuration is chosen with the help of a set of tuning instances representatives of the problem at hand
  2. the chosen algorithm configuration is used to solve unseen instances of the same problem
- Its goal is to find an algorithm configuration that minimizes some cost measure over the set of instances that will be seen during the second phase

Automatic Parameter Tuning with irace

- The following parameters are tuned:
  1. shaker type (categorical): affine shaker (AS) or inertial shaker (IS)
  2. η (numerical, (0, 1]): initial search region size multiplier (considered precision is 2)
  3. ρ (numerical, (0, 1]): box expansion/contraction factor for the selected shaker (considered precision is 2)
- We performed different irace executions to diversify the tested candidates
- Different parameter configurations of different irace runs produced nearly the same results
- Inertial shaker was chosen in all the best configurations
- Best submission: η was equal to 0.01 and best ρ was equal to 0.48
- Found configuration parameters are used to execute RAS over the GENOPT functions

Conclusions

- We've used both offline (irace) and online tuning ("reactive" determination of the search area)
- We succeeded in employing very different parameter values in a short time using irace for automatic parameter configuration
- Small η values: because of using η to identify random initial values of variables
- Using random initial points around middle points gives an advantage to the RAS, because most of the GENOPT functions have their global minimum around center of the dimensions
- We didn't reach best results in all of functions because different functions have conflicting objectives
- We mostly reached better high jump scores, but worse target shooting scores in the submissions

Thanks & Questions

THANKS FOR LISTENING!
ANY QUESTIONS?

Leaderboard

GENOPT 2016 Prize, Biathlon and High Jump: Eduardo Segredo, Eduardo Lalla-Ruiz, Ben Paechter, Emma Hart, Stefan Voß

GENOPT 2016 Prize, Target Shooting: Konstantin Barkalov, Alexander Sysoyev, Ilya Lebedev, Vladislav Sovrasov

Position	Team	High Jump	Target Shooting	Biathlon Score
1	Eduardo Segredo (School of Computing, Edinburgh Napier University, UK) Eduardo Lalla-Ruiz (Institute of Information Systems, University of Hamburg, Germany) Ben Paechter (School of Computing, Edinburgh Napier University, UK) Emma Hart (School of Computing, Edinburgh Napier University, UK) Stefan Voß (Institute of Information Systems, University of Hamburg, Germany)	1.88889	2.3889	2.13889
2	Costas Voglis (Nodalpoint LTD, Athens, Greece)	2.08333	2.3889	2.23611
3	Konstantin Barkalov (Software Department, Lobachevsky State University of Nizhny Novgorod, Russia) Alexander Sysoyev (Software Department, Lobachevsky State University of Nizhny Novgorod, Russia) Ilya Lebedev (Software Department, Lobachevsky State University of Nizhny Novgorod, Russia) Vladislav Sovrasov (Software Department, Lobachevsky State University of Nizhny Novgorod, Russia)	2.23611	2.2778	2.25694
4	Tahir Emre Kalayci (LION lab, Università degli Studi di Trento, Italy)	1.93056	2.9444	2.4375
5	Vasily Ryabov (Software Department, Lobachevsky State University of Nizhny Novgorod, Russia) (Submitted at March 31)	3.875	4.6667	4.27083

References I

Prasanna Balaprakash, Mauro Birattari, and Thomas Stützle.
Improvement strategies for the f-race algorithm: Sampling design and iterative refinement.
In Thomas Bartz-Beielstein, María José Blesa Aguilera, Christian Blum, Boris Naujoks, Andrea Roli, Günter Rudolph, and Michael Sampels, editors, *Proc. of 4th Int. Workshop on Hybrid Metaheuristics (HM 2007)*, pages 108–122. Springer Berlin Heidelberg, 2007.

Roberto Battiti and Mauro Brunato.
The LION way. Machine Learning plus *Intelligent Optimization*.
LIONlab, University of Trento, Italy, 2014.

Roberto Battiti and Giampietro Tecchiolli.
Backpropagation, part iv learning with first, second, and no derivatives: A case study in high energy physics.
Neurocomputing, 6(2):181 – 206, 1994.

References II

Mauro Birattari, Zhi Yuan, Prasanna Balaprakash, and Thomas Stützle.
F-race and iterated f-race: An overview.
In Thomas Bartz-Beielstein, Marco Chiarandini, Luís Paquete, and Mike Preuss, editors, *Experimental Methods for the Analysis of Optimization Algorithms*, pages 311–336. Springer Berlin Heidelberg, 2010.

Mauro Brunato and Roberto Battiti.
Rash: A self-adaptive random search method.
In Carlos Cotta, Marc Sevaux, and Kenneth Sörensen, editors, *Adaptive and Multilevel Metaheuristics*, pages 95–117. Springer Berlin Heidelberg, 2008.

Mauro Brunato and Roberto Battiti.
Corso (collaborative reactive search optimization): Blending combinatorial and continuous local search.
Informatica, An international journal, to appear in.

Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari.
The irace package, iterated race for automatic algorithm configuration.
Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.

G. P. Tecchiolli and R. Brunelli.
On random minimization of functions.
Biological Cybernetics, 65(6):501–506, 1991.