Benchmarking Projection-Based Real Coded Genetic Algorithm on BBOB-2013 Noiseless Function Testbed

Babatunde Sawyerr[1] Aderemi Adewumi[2] Montaz Ali[3]

[1]University of Lagos, Lagos, Nigeria
[2]University of KwaZulu-Natal, Durban, South Africa
[3]University of The Witwatersrand, Johannesburg, South Africa

Workshop on Black Box Optimization Benchmarking, 2013

Outline

Outline

1. Introduction
   - Problem Statement
   - Genetic Algorithms
2. Projection-based Real Coded Genetic Algorithm
   - Projection
   - The PRCGA Algorithm
3. Experimental Procedure
   - Experimental Settings
4. Experimental Results
   - Empirical Results
   - Discussion
5. Thank you

The Global Optimization Problem

Real Parameter Optimization

- The task is to `minimize` an objective function f. Given $f : S \rightarrow \Re$ where $S \subset \Re^n$, find $x^* \in S$ for which,

$$f(x^*) \leq f(x), \quad \forall x \in S. \tag{1}$$

- Black Box approach:
  - gradients are not known or not useful.
  - problem domain are rugged and ill-conditioned.
- Goal:
  - To find the global optimum, x^* quickly.
  - With the least search cost (function evaluations).

Outline

1. Introduction
   - Problem Statement
   - Genetic Algorithms
2. Projection-based Real Coded Genetic Algorithm
   - Projection
   - The PRCGA Algorithm
3. Experimental Procedure
   - Experimental Settings
4. Experimental Results
   - Empirical Results
   - Discussion
5. Thank you

Genetic Algorithms

- Developed by John Holland in 1975.
- Goal: Develop robust and adaptive systems.
- Solutions are represented internally as genetic encoding of points.
- Reproduction of offspring via:
  - mutation,
  - recombination.
- Selection methods: initially Fitness-proportional method.
- Model: Generational or Steady state.

Real Coded Genetic Algorithms

- Real valued representation are used as genetic encodings of points.
- They are better adapted to numerical optimization of continuous problems.
- They can also be easily hybridized with other search methods.

Outline

Orthogonal Projection of a vector x on a vector y

For any two n dimensional vectors, the projection of a vector x on another vector y generates a vector, defined by:

$$\hat{y} = \frac{x^T y}{y^T y} y = \frac{x^T y}{\|y\|^2} y = \left(\frac{\|x\| \cos(\theta)}{\|y\|} y \right). \tag{2}$$

Note that the projected vector $\hat{y}$ (the offspring) will be in the same direction as y unless $\frac{\pi}{2} < \theta < \frac{3\pi}{2}$ in which case the angle, θ, between the two vectors is such that $\cos(\theta) < 0$. As a result, the projected vector is in the opposite direction (the reflection of y about the origin).

Orthogonal Projection of a vector *x* on a vector *y*

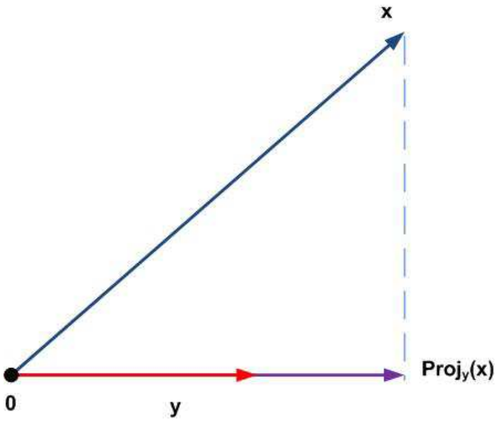


Figure: Projection of vector x on vector y

Outline

The PRCGA Algorithm

- PRCGA was first introduced as RCGA-P in [6, 7].
- The incorporated projection operator showed promising exploratory search capability in some search problems.
- PRCGA is an enhanced version of RCGA-P.

Inputs

- Fitness function f.
- Parameters.

Outputs

- The Best solution x_{best}.
- Fitnss value of x_{best}, $f(x_{best})$.

The PRCGA Algorithm

1. Initialize $P_{t=0}, P_t = \{x_{1,t}, x_{2,t}, \ldots, x_{N,t}\}$ from S
2. $f(x_{i,t}) =$ evaluate(P_t), $\{1 \leq i \leq N\}$
3. While not stopping condition, do steps 4 - 12
4. $\zeta_t = \sigma(f(P_t))$, if $\zeta_t \leq \epsilon$ do step 5 else step 6
5. $\hat{P}_t =$ perturb(P_t)
6. $\hat{P}_t =$ tournamentSelection(P_t)
7. $C_t =$ blend-αCrossover$(\hat{P}_t, p_c)$
8. $M_t =$ non-uniformMutation(C_t, p_m)
9. $\Phi_t =$ projection(M_t)
10. $f(x_{i,t}) =$ evaluate(Φ_t)
11. $P_{t+1} =$ replace(P_t, Φ_t)
12. $t = t + 1$
13. end while

Outline

Computer System and Software

Computer System Configuration

- HP Probook 6545b with AMD Turion(tm) II Ultra Dual-Core mobile M620 CPU processor.
- CPU Speed: 2.5GHz.
- RAM: 2.75GB

Software

- Microsoft Windows 7 Professional service pack 1.
- MATLAB 7.10 (R2010a).
- COmparing Continuous Optimisers (COCO) software.
- Post-Processing Script in Python

Experimental setup

- The experimental setup was carried out according to [3] on the benchmark functions provided in [2, 4].
- Two independent restart strategies were employed
  - Checks for stagnation [1].
  - Maximum number of generations reached without f_{target}.
- For each restart strategy, the genetic run is initiated with an initial population P_0 which is $\sim Unif([-4, 4]^D)$.

Parameter Settings

Parameters

- Population Size = $\min(100, 100 \times D)$,
  where D = dimension.
- Maximum Number of Evaluation = $10^5 \times D$.
- Tournament size = 3.
- Crossover rate p_c = 0.8.
- Mutation rate p_m = 0.15.
- Non-uniformity factor for Mutation β = 15.
- Crafting effort CrE = 0.

CPU Timing Experiment

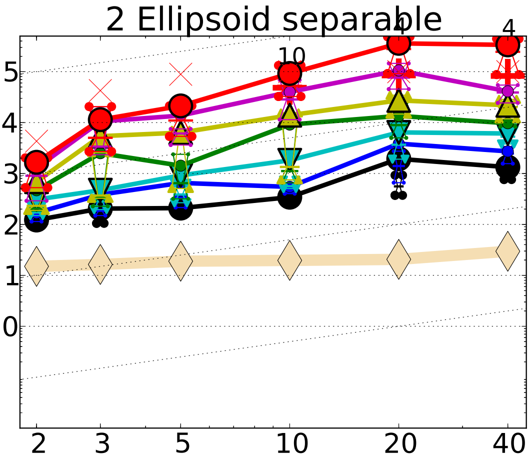
The CPU timing experiment was conducted using the same independent restart strategies on the function f_8 for a duration of 30 seconds.

Time per function evaluation

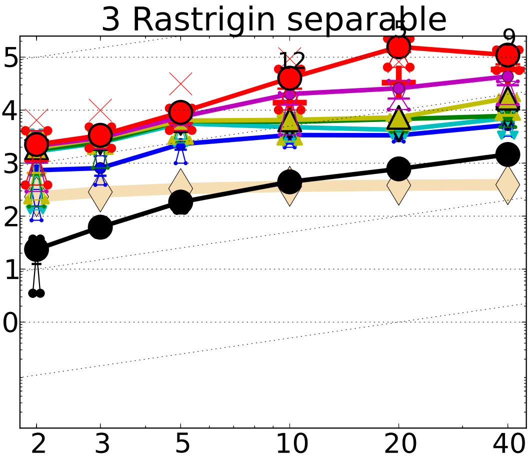
Dimension	2	3	5	10	20	40
Time ($\times 10^{-5}$)	7.1	7.5	6.9	6.9	7.1	8.0

Outline

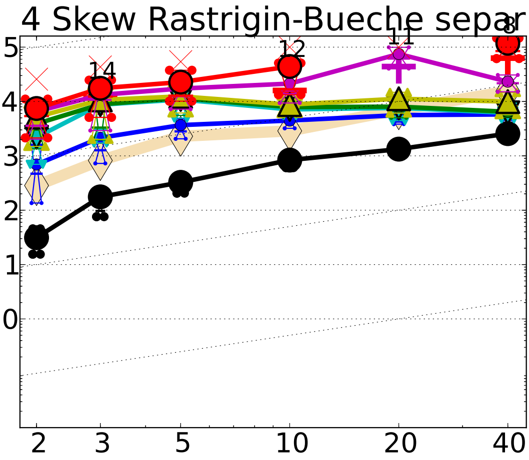
Ellipsoid separable



Rastrigin separable



Skew Rastrigin-Bueche separable



Outline

Discussion

Separable Functions

- PRCGA performed well on separable functions $f_1 - f_4$.
- PRCGA also solved Gallagher's Gaussian 101-me Peaks Function f_{21}, a multi-modal function with weak global structure.
- PRCGA showed some encouraging performance in solving problems $f_6 - f_7$ in dimensions $2 - 10$.

Functions with high conditioning and unimodal

- Functions $f_{10} - f_{14}$ prove to be difficult for PRCGA to solve to the required level of accuracy.

Discussion

Comparison of PRCGA with Previous GAs

- DBRCGA [1] outperformed PRCGA.
- PRCGA performed better than the RCGA in [8].
- PRCGA performed better than the simpleGA in [5].

Conclusion

- The benchmarking of PRCGA on noiseless BBOB function testbed shows the strengths and weaknesses of the algorithm.
- The performance of PRCGA shows that in its current form it cannot compete with state-of-the-art evolutionary algorithms.

Thank You!!!

For Further Reading I

- Y.-C. Chuang and C.-T. Chen.
  Black-box optimization benchmarking for noiseless function testbed using a direction-based rcga.
  In *GECCO (Companion)*, pages 167–174, 2012.

- S. Finck, N. Hansen, R. Ros, and A. Auger.
  Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions.
  Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

For Further Reading II

N. Hansen, A. Auger, S. Finck, and R. Ros.
Real-parameter black-box optimization benchmarking 2012: Experimental setup.
Technical report, INRIA, 2012.

N. Hansen, S. Finck, R. Ros, and A. Auger.
Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions.
Technical Report RR-6829, INRIA, 2009.
Updated February 2010.

For Further Reading III

M. Nicolau.
Application of a simple binary genetic algorithm to a noiseless testbed benchmark.
In *GECCO (Companion)*, pages 2473–2478, 2009.

B. A. Sawyerr.
Hybrid real coded genetic algorithms with pattern search and projection.
PhD thesis, University of Lagos, Lagos, Nigeria, 2010.

For Further Reading IV

B. A. Sawyerr, M. M. Ali, and A. O. Adewumi.
A comparative study of some real coded genetic algorithms for unconstrained global optimization.
Optimization Methods and Software, 26(6):945–970, 2011.

T.-D. Tran and G.-G. Jin.
Real-coded genetic algorithm benchmarked on noiseless black-box optimization testbed.
In *GECCO (Companion)*, pages 1731–1738, 2010.