SEA: An Evolutionary Algorithm based on Spherical Inversions

Juan Pablo Serrano Rubio
Center for Research in Mathematics (CIMAT)
Guanajuato, Mexico
jpsr@cimat.mx

Arturo Hernández Aguirre
Center for Research in Mathematics (CIMAT)
Guanajuato, Mexico
artha@cimat.mx

Rafael Herrera Guzmán
Center for Research in Mathematics (CIMAT)
Guanajuato, Mexico
rherrea@cimat.mx

ABSTRACT

This paper introduces the Spherical Evolutionary Algorithm (SEA) for global continuous optimization. Two new geometric search operators are included in the design of the SEA. The operators are named: Inversion Search Operator (ISO) and Reflection Search Operator (RSO). This paper describes the general implementation of SEA and its performance is analyzed through a benchmark of 10 functions.

Track: Evolution Strategies and Evolutionary Programming.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

Evolutionary Algorithm, Explore and Exploit, Geometric Search Operators.

1. INTRODUCTION

In general, an evolutionary algorithm is defined by the reproduction and/or mutation operators to create new individuals. The reproduction and mutation operators define the search mechanism of the algorithm. Some evolutionary algorithms have been studied from a geometric perspective [2, 3]. Therefore, the search space can be considered as a geometric space where the new offspring have a relation of distance with the parents. This paper introduces an evolutionary algorithm with two geometric search operators named Inversion Search Operator (ISO) and Reflection Search Operator (RSO).

2. IMPLEMENTING THE SEA

A pseudo-code for SEA is shown in Algorithm 1. The SEA performs either the ISO or the RSO operators with a probability of 0.5. The geometric search operators are defined over a hyper-sphere which delimit the region of search space to produce a new offspring solution with respect to the position of the parents.

Algorithm 1 Pseudo-code of the SEA.

```
procedure SEA
    /* d: dimension of the problem
       ps: population size
       η: Number of hyper-spheres for inverse points */
    /*1. Initialize a population */
    Pop ← {x^(1), x^(2), ..., x^(ps)}; where the individual x^(i) ∈ ℝ^d
    t ← 0
    while not stopping condition do
        𝒫 ← Select the best η individuals.
        for i ← 1 to ps do
            /*2. Select a center of the hyper-sphere*/
            repeat
                c ← select randomly one from 𝒫
            until c ≠ x^(i)
            if U(0,1) ≥ 0.5 then
                /*3. Inversion Search Operator*/
                y ← InversionSearchOperator(c, x^(i))
            else
                /*4. Reflection Search Operator*/
                y ← ReflectionSearchOperator(c, x^(i))
            end if
            if fitness(x^(i)) > fitness(y) then
                x^(i) ← y /*5. Replace the individual*/
            end if
        end for
        t ← t + 1
    end while
end procedure
```

The ISO is presented in the Algorithm 2. The heart of the ISO is the inversion with respect to a hyper-sphere [1]. The main steps of ISO which furnish and enhance the search capability of the SEA are listed as follows:

- Mutation for the center of the hyper-sphere.
- Calculation of the radius r of the hyper-sphere.
- Calculation of the inverse point in the acceptable region.

The inverse point of x is $x^{\wedge}$, and c is the center of the hyper-sphere. In order to restrict the inversion to an acceptable region (close to the inversion hyper-sphere), we use two more hyper-spheres. These two hyper-spheres have the same center but different radii

$$\alpha = re^{\frac{-1}{r}} \quad \text{and} \quad \beta = re^{\frac{1}{r}}.$$

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2598492.

Algorithm 2 Pseudo-code of the Inversion Search Operator.

```
procedure InversionSearchOperator(c, x)
    /*1. Mutation for the center
    if U(0,1) ≥ 0.5 then
        k ← Select randomly one component of the center
        c_k ← c_k + N(0,1)
    end if
    /*2. Calculate the radius of the hyper-sphere*/
    r ← 2(U(0,1))(‖ c − x ‖^2)
    if ‖ c − x ‖> 0 then
        /*3. Compute the inverse point*/
        x^ ← (r^2/‖c−x‖) ((c−x)/‖c−x‖) + c
        /*4. Calculate the acceptable regions */
        α ← r(e^(−1/r)); β ← r(e^(1/r))
        if ‖ x^ − c ‖< α then
            x^ ← α((c−x^)/‖c−x^‖) + c
        else if ‖ x^ − c ‖> β then
            x^ ← β((c−x^)/‖c−x^‖) + c
        end if
    else
        x^ ← x /* The individual does not change*/
    end if
    return x^
end procedure
```

The pseudo-code of the RSO is presented in Algorithm 3. The RSO redistributes the points on the surface of the hyper-sphere.

Algorithm 3 Pseudo-code of the Reflection Search Operator.

```
procedure ReflectionSearchOperator(c,x)
    v^radius ← x − c
    for j ← 1 to d do
        if U(0,1) ≥ 0.5 then
            v_j^radius ← (−1)(v_j^radius)
        end if
    end for
    v^new ← v^radius + c
    return v^new
end procedure
```

3. EXPERIMENTS

A benchmark of 10 functions is listed in Table 1. The population size of the SEA is set to 129 and the number of hyper-spheres is set to $\eta = 9$. The problem dimensionality is set to 30 dimensions (d). 50 runs were made for each function. The stop criteria is given true when the number of Fitness Function Evaluations (FFEs) reach 1e+06 or the current Best Fitness Value (BFV) is smaller than the objective threshold (1e-10). Table 2 reports the FFEs, BFV, Worst Fitness Value (WFV) and Success Rate (SR) reached by the SEA in the tested functions.

4. CONCLUSIONS

In this paper, we proposed the SEA. According to numerical experiments the ISO and RSO furnish and enhanced the search capability of the algorithm. The ISO computes the inverse point with respect to the hyper-spheres and an acceptable region is implemented to guide the search. RSO mutates an individual repositioning it on the hyper-sphere. The nonlinear geometric nature of the ISO enhances the search capability of the algorithm.

Table 1: Benchmark functions.

Sphere Model: $f_1(0) = 0$ $f_1(x) = \sum_{i=1}^{d} x_i^2$, $-100 \leq x_i \leq 100$.
Generalized Rosenbrock's Function: $f_2(1) = 0$. $f_2(x) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$, $-30 \leq x_i \leq 30$.
Generalized Rastrigin's Function: $f_3(0) = 0$ $f_3(x) = \sum_{j=1}^{d} \left[x_i^2 - 10cos(2\pi x_i) + 10\right]$, $-5.12 \leq x_i \leq 5.12$,
Ackley's Function: $f_4(0) = 0$ $f_4(x) = -20exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - exp\left(\frac{1}{d}\sum_{i=1}^{d} cos(2\pi x_i)\right)$ $+20 + e$, $-32 \leq x_i \leq 32$
Ellipsoid Function: $f_5(0) = 0$ $f_5(x) = \sum_{i=1}^{d} 10^{6\left(\frac{i}{d-1}\right)} x_i^2$, $-10 \leq x_i \leq 5$
Cigar Function: $f_6(0) = 0$ $f_6(x) = x_1^2 + \sum_{i=2}^{d} 10^6 x_i^2$, $-10 \leq x_i \leq 5$
Tablet Function: $f_7(0) = 0$ $f_7(x) = 10^6 x_1^2 + \sum_{i=2}^{d} x_i^2$, $-10 \leq x_i \leq 5$.
Cigar Tablet: $f_8(0) = 0$ $f_8(x) = x_1^2 + \sum_{i=2}^{d-1} 10^4 x_i^2 + 10^8 x_d^2$, $-10 \leq x_i \leq 5$.
Different Powers: $f_9(0) = 0$ $f_9(x) = \sum_{i=1}^{d} \lvert x_i \rvert^{2+10\frac{i-1}{d-1}}$, $-10 \leq x_i \leq 5$.
Parabolic Ridge: $f_{10}(5, 0, 0, ..., 0) = -5$ $f_{10}(x) = -x_1 + 100\sum_{i=2}^{d} x_i^2$, $-10 \leq x_i \leq 5$.

Table 2: Descriptive statistics fitness function evaluations. Threshold to reach $\leq$ 1e-10.

f	SEA Values					
	FFEs		BFV	WFV	BFV	SR
	Mean	Std.Dev.	Mean	Mean	Std.Dev.	
f_1	5.37e+04	1.36e+03	8.86e-11	9.98e-11	7.14e-12	100%
f_2	6.16e+05	5.91e+04	9.45e-11	1.00e-10	6.72e-12	100%
f_3	1.86e+05	2.03e+04	8.82e-11	9.98e-11	9.24e-12	100%
f_4	9.45e+04	3.95e+03	9.39e-11	9.97e-11	4.28e-12	100%
f_5	6.44e+04	1.81e+03	8.79e-11	9.78e-11	7.81e-12	100%
f_6	7.28e+04	1.81e+03	9.03e-11	9.95e-11	6.83e-12	100%
f_7	5.34e+04	1.49e+03	8.64e-11	9.89e-11	6.96e-12	100%
f_8	6.78e+04	1.83e+03	8.70e-11	9.97e-11	7.84e-12	100%
f_9	3.09e+04	1.65e+03	8.12e-11	9.90e-11	1.17e-11	100%
f_{10}	6.17e+04	1.89e+03	-5.00e+00	-5.00e+00	1.10e-11	100%

5. ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support from the National Council for Science and Technology of Mexico (CONACyT) and from the Center for Research in Mathematics (CIMAT). The third author would like to thank the International Centre for Theoretical Physics (ICTP) for its hospitality and support.

6. REFERENCES

[1] H. Eves. *College Geometry.* Jones and Bartlett, 1995.

[2] A. Moraglio, C. Di Chio, and R. Poli. Geometric particle swarm optimisation. In *Proceedings of the 10th European Conference on Genetic Programming*, EuroGP'07, pages 125–136, Berlin, Heidelberg, 2007. Springer-Verlag.

[3] A. Moraglio and C. G. Johnson. Geometric generalization of the nelder-mead algorithm. In *Proceedings of the 10th European Conference on Evolutionary Computation in Combinatorial Optimization*, EvoCOP'10, pages 190–201, Berlin, Heidelberg, 2010. Springer-Verlag.