A NEW EVOLUTIONARY OPTIMIZATION TECHNIQUE

by
Rodica Ioana Lung, Dan Dumitrescu

Abstract: Roaming optimization is a new evolutionary technique that is used for multimodal optimization. Roaming uses multiple subpopulations evolving in isolation in order to detect local and global optima. Within roaming a stability measure for a subpopulation is defined and used to asses whether a subpopulation has detected an optimum. An external population (archive) it is used to store the optima already found. A new method to add solutions to the archive is presented here. Experimental results prove the efficiency of the algorithm.
Keywords: genetic algorithms, multimdal optimization, roaming optimization.

Genetic algorithms have proven effective in solving a variety of search and optimization problems. Determining the global optima within a fitness landscape has been the subject of much research. The intrinsic parallelism in a GA suggests, however, that this method should be able to locate several optima of a multimodal function. The problem of locating multiple solutions raises in many real world applications where the knowledge of several potential solutions provides the decision maker with a better insight into the nature of the design space and perhaps suggest alternative solutions.

Fitness sharing (Goldberg, Richardson, 1987) is the most popular technique for detecting multiple optima. Other approaches consider parallel subpopulations evolving in isolation or in comunication with each other in order to locate the optima. Some of the most popular parallel models are the island models (Gordon, Whitley and Bohn, 1992).

1. ROAMING OPTIMIZATION

A new evolutionary technique called roaming is proposed. Roaming is inspired by movement of nomad populations trying to find best inhabitation areas by searching different regions. In a similar manner, roaming technique uses multiple subpoputations that are searching the space in order to detect the optima. Suppose one nomad population has found a potential resourcefull area, then the nomads would either populate that area or mark it as a possible dwelling area. The subpopulations used by roaming technique are detecting the potential optima in the same manner and then store them into an external population called the Archive. Just like the nomad populations that will continue to travel to different areas, after saving an optimum the subpopulations will roam to different regions of the search space.

2. ROAMING TECHNIQUE

Roaming technique identifies the local optima using isolated subpopulations and stores them in an external population (called the archive). The number of subpopulations is a parameter of the algorithm and it is not related to the expected number of local optima. This confers flexibility and robustness to the search mechanism. When a local optimum is detected it is stored in the archive.

Let N be the number of subpopulations. At each generation t the population *P(t)* is composed of N subpopulations $P_i(t)$, i = 1, . . . ,N.

We may define an order relation on *P(t)*. Consider the maximization problem.

$$\begin{cases} \max\ eval(x) \\ x \in S \end{cases}$$

where S is the solution space and *eval(x)* the fitness value of individual x . We say that individual x is better then y and we write x >y if and only if the condition

$$eval(x) \geq eval(y)$$

holds.

2.1. Stability measure

A stability measure it is introduced determining whether a subpopulation has located a potential optimum.

By evolving subpopulation $P_i(t)$ for n generations a new subpopulation Pi (t) having the same size as $P_i^{'}(t)$ is obtained. The number n of iterations the subpopulations evolve in isolation until their stability is measured and tested is a parameter of the algorithm called the *iteration parameter*.

Let x_i^* be the best individual in the parent subpopulation $P_i(t)$. We define an operator B as the set of individuals in the offspring in subpopulation $P_i^{'}(t)$ that are better then x_i^* :

$$B : P(t) \longrightarrow \mathcal{P}(P(t))$$

$$B(x_i^*) = \{x \in P_i^{'}(t) \mid x \succ x_i^*\}$$

Using the card of set B a stability measure $SM_i(t)$ of subpopulation $P_i(t)$ may be defined.

Definition 1. Stability measure of the subpopulation $P_i(t)$ is the number $SM_i(t)$ defined as

$$SM_i(t) = 1 - \frac{card\, B(x_i^*)}{card P_i(t)},$$

where x_i^* is the best individual in $P_i(t)$ and *cardA* represents cardinality of the *set A*.

Proposition 1. Stability measure of a subpopulation has the following properties:

(i) $0 \le SM_i(t) \le 1$;

(ii) If $SM_i(t) = 1$ then x_i^* is a potential local optimum;

(iii) $SM_i(t) = 0$ if all offsprings in $P_i^{'}(t)$ are better then x_i^* which means that $P_i^{'}(t)$ is not near d convergence;

The proof is obvious using the definition.

A subpopulation having a stability measure close to 0 is considered to be highly unstable. In Roaming technique unstable subpopulations evolve in isolation until they detect a local optima.

A subpopulation near convergence has a stability measure close to 1 and it is considered to be stable. A high stability measure can also indicate that the subpopulation has detected a false optima.

2.2. Adding a solution to the archive

Roaming technique uses an external population denoted A and called the archive to store the potential optima.

Consider a stable subpopulation $P_i(t)$, i =1, . . . ,N with $SM_i(t) = 0$. This means that after n generation of evolving in isolation no offspring better then x_i^* has been produced. In this case we consider x_i^* as a potential local optimum.

It is reasonable to suppose that a potential optimum can be a local optimum or can be very close to a local optimum.

Before adding a candidate solution x^* to the archive the distance between x^* and every solution x in the archive is compared with an archive parameter δ. If the condition $d(x^*, x) < \delta$ holds then only the best fitted from x^* and x enters the archive. δ is a parameter of the algorithm.

Summarizing, a solution x^* is added to the archive if and only if one of the following situations arrize:

(i) the distance to all other solutions in the archive is higher then δ, meaning that x^* represents a new optimum for the archive;

(ii) x^* is better then a solution x in the archive that is at a distance smaller then δ from x^*, in which case x^* replaces x.

The archive is used exclusively to store the potential optima. Solutions preserved in the archive do not participate in further selection processes. Thus the algorithm can not be considered an elitist one.

2.3. Roaming subpopulations

Consider a potential optimum x_i^*, $i \in \{1, . . . ,N\}$ has been added to the archive. To avoid the search process to get stuck, the search performed by the subpopulation P_i has to be redirected to other regions of the search space. Therefore stable subpopulations are selected to be spread in the search space in order to locate new optima.

Selected subpopulations are called *Roaming Subpopulations*. The selection is performed using the stability measure of the subpopulations as an evaluation function.

A parameter $RS \in [0, 1]$ is used as a threshold in order to select the stable subpopulations. Thus if for subpopulation $P_i(t)$ we have $SM_i(t)>RS$ then $P_i(t)$ is selected as a roaming subpopulation.

The roaming subpopulations are spread in the search space in order to detect other optima. We can say that the subpopulations are roaming through the search space. The roaming is realized using genetic operators defined for subpopulations. Here we used a mutation operator for subpopulations that applies strong mutation to each individual of the subpopulation. To ensure complete change of the individuals of the subpopulations, the mutation rate here will be closer to 1. Other types of genetic operators for subpopulations can be defined.

The next generation $P(t+1)$ will be composed of the roamed subpopulations and the offspring Pi (t) of the subpopulations $P_i(t)$ that were not selected to roam.

The algorithm stops after a given number of generations. At the end the archive contains detected local optima.

2.4. Roaming algorithm

Roaming algorithm may be outlined as follows:

Roaming algorithm

Input: N - subpopulations number
Popsize - subpopulation size
Ngen - maximum number of generations
n - iteration parameter
δ - archive parameter
RS - roaming threshold
p_c, p_m -crossover probability and mutation rate

Output: Archive - the set of local optima

Step 1: Initialization a) t := 0;
b) Initialize $P_i(0)$ for each i = 1, . . .N by generating popsize number of individuals;
c) Archive = ϕ;.

Step 2: Evaluate each individual x in each subpopulation $P_i(t)$ by computing its fitness *eval(x)*;
Step 3: Evolve each subpopulation $P_i(t)$ for n iterations applying selection, recombination and mutation .
Let $P_i^{'}(t)$ be the resulting offspring subpopulation.
Step 4: Evaluate each individual x in

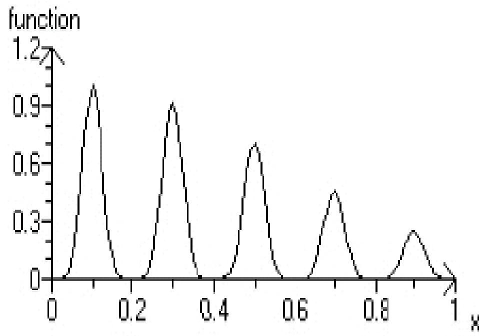

Fig. 1. Decreasing Maxima Function F1

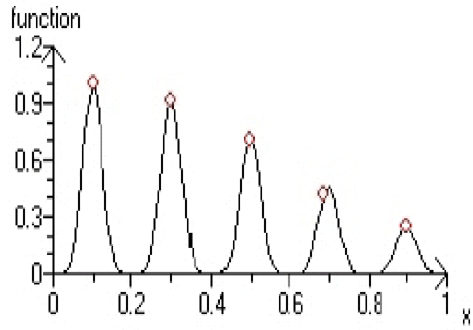

Fig. 2. Archive after 10 generations

the offspring subpopulation $P_i(t)$ by caculating its fitness $eval^{'}(x)$.
Step 5: For each subpopulation $P_i(t)$ calculate:
a) The best individual x_i^*;
b) The stability measure $SM_i(t)$ using Definition 1.
Step 6: For each subpopulation *Pi(t)* having $SM_i(t) = 0$ try to add x_i^* to the *Archive*.
Step 7: For each i = 1, . . .N do
if $SM_i(t) \geq RS$ then consider $P_i(t)$ to be a roaming subpopulation;
Step 8: Migrate all roaming subpopulations using strong mutation with rate = 1

Step 9: Set $P(t+1) = \{P_i(t) \mid P_i(t)$ is a Roaming Subpopulation$\} \cup \cup \{P_i^{'}(t) \mid P_i(t)$ is not a Roaming Subpopulation$\}$;
t = t + 1. If t < *Nrgen* then go to step 2, else stop.

3. EXPERIMENTAL RESULTS

Roaming Algorithm has been tested on several standard test functions.

The results of tests performed on function F_1 as defined by Goldberg (1989) are presented. Function F_1 has five decreasing peaks the range [0, 1] and it is defined as: $F_1 : [0, 1] \rightarrow [0, 1]$,

$$F_1(x) = e^{-2\ln(2)\left(\frac{x-0.1}{0.8}\right)^2} \sin^6(5\pi x)$$

The exact positions and values of the maxima as given in [Gan and Warwick] are given in Table1. F_1 is considered to be a deceptive test function because of the fifth peak that can be easily missed by the search algorithm.

Peak	x	Fitness
1	0.1	1
2	0.2994	0.9172
3	0.4988	0.7078
4	0.6982	0.4595
5	0.8976	0.2510

Table 1. F_1 peaks

The results presented here were obtained using the parameters given in Table 2.

Subpopulation number	15
Subpopulation size	10
Generation number	75
Parameter δ	0.1
Search precision	4
Crossover probability	0.5
Mutation rate	0.05
RS parameter	0.7
Iteration parameter n	20
Binary tournament selection	

Table 2. Parameters used for function F_1

Roaming algorithm detects the peaks of the function at early stage of the search process (Figure 2). After 10 generations the Roaming algorithm has already detected all the peak regions. Detected solutions are very close to the real optima. The number of generations needed to refine these solutions depends on the robustness of the variation operators used in Step 2 and also on the number of subpopulations used. Figure 3 shows the archive after 75 generatios.

Numerical results averaged over 10 runs are presented in Table 3.

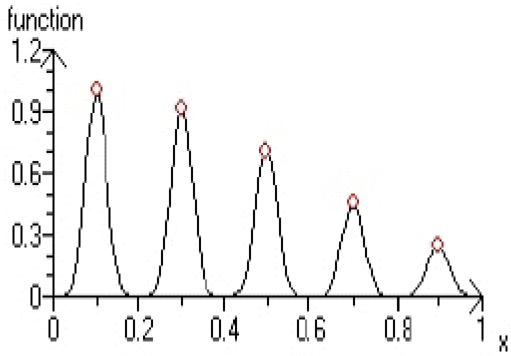

Fig. 3. After 75 generations the Archive contains 5 solutions

Peak	x	Fitness
1	0.1	1
2	0.2992	0.9172
3	0.4984	0.7078
4	0.6981	0.4595
5	0.8973	0.2510

Fig. 4. Numerical results after 75 generations

3.1. Setting Parameters

Roaming technique uses specific parameters such as the *RS* threshold, δ and the number of subpopulations.

The *RS* parameter is chosen so that only stable subpopulations are selected to roam. In our example, if the subpopulation size is 10, then its stability measure can only take the values 0, 0.1, 0.2, ..., 1. If *RS* would be 0.9 then only subpopulations having the stability measure 0 would be chosen to roam.

The value of δ depends on the distribution of the local optima in the fitness landscape. For evenly distributed landscapes, the choice of δ does not represent a problem. However it is almost impossible to choose a suitable value for δ in the case of the functions with unevenly spaced optima. A mechanism to avoid the use of this parameter is the object of current work.

The algorithm works for any number of subpopulations as a parameter. If a small number of subpopulations it is used then in order to detect more optima the number of generations has to be increased. Algorithm convergence depends on the subpopulation number proposed.

4. CONCLUSIONS AND FUTURE WORK

A new evolutionary technique for multimodal optimization called Roaming uses a number of roaming subpopulations in order to detect multiple optima. A stability measure of a sub-population it is introduced in order to asses whether a subpopulation has located an optimum. Subpopulations evolve in isolation until they detect an optimum. Detected optima are saved into an archive and the corresponding subpopulations are spread to other regions of the search space. Numerical examples are presented to prove the efficiency of the algorithm. Roaming algorithm is capable to locate the optima of the function but it needs improvment as far as the accuracy of the solutions is concerned. A solution for this problem might be the use of local search in order to improve the individuals in the archive.

5. REFERENCES

D. Dumitrescu, B. Lazzerini, L.C. Jain, A. Dumitrescu; *Evolutionary Computation*, CRC Press 2000.

J. Gan, K. Warwick; A *Genetic Algorithm with Dynamic Niche Clustering for Multimodal Optimization*, 4th International Conference on Artificial Neural Networks and Genetic Algorithms, 1999.

D.E. Goldberg; Genetic *Algorithms in Search, optimization and Machine Learning*, Addison Weasley, 1989.

D.E. Goldberg, J. Richardson; *Genetic Algorithms with Sharing for Multimodal Function Optimization*, 2nd International Conference on Genetic Algorithms, 41-49

V. S. Gordon, D. Whitley, A. Bohn; *Dataflow parallelism in genetic algorithms*, Manner, R. and Manderick, B., editors, Parallel Problem Solving from Nature 2,(1992) 533 - 542

R.I. Lung, D. Dumitrescu *Roaming optimization: A new evolutionary technique*, Symbolic and Numeric algorithms for scientific computing SYNASC03, 149-156

Authors:

Rodica Ioana Lung, Dan Dumitrescu, ddumitr@cs.ubbcluj.ro, Computer Science Department, “Babes-Bolyai” University, Cluj-Napoca, Romania.