

A New Evolutionary Model for Detecting Multiple Optima

Rodica Lung
Babes-Bolyai University
Department of Computer Science
Cluj Napoca, Romania
srodica@cs.ubbcluj.ro

D. Dumitrescu
Babes-Bolyai University
Department of Computer Science
Cluj Napoca, Romania
ddumitr@cs.ubbcluj.ro

ABSTRACT

Multimodal optimization problems consist in detecting all global and local optima of a problem. A new evolutionary approach to multimodal optimization called Roaming technique (RO) is presented. Roaming uses two original concepts in order to detect multiple optima: a stability measure for subpopulations and an external population called archive to store detected optima. Individuals in the archive are refined by evolving them independently. Performance of Roaming is compared by means of numerical experiments with two other evolutionary techniques.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

evolutionary multimodal optimization, stability measure

1. INTRODUCTION

Most real world problems present more than one local optima solution. Such problems occur in pattern recognition, fixed point theory, classification, game theory, etc.

For such problems we are interested not only in finding one or more global optima but in identifying the set of all acceptable solutions. The problems requiring the detection of all local and global optima are called multimodal optimization problems (MMOPs).

From the point of view of the number and distribution of the optima, MMOPs can be classified in two categories:

- MMOPs presenting a discrete set of optima: these problems are addressed in this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...\$5.00.

- MMOPs having an infinite number of optima: for example optima can be displayed in a circular manner similar to the waves of a lake when a stone is thrown in it.

Due to their high adaptability evolutionary algorithms are very good candidates for approaching multimodal optimization. One of their main advantages is that theoretically they may be designed to detect and to maintain a *set* of solutions during one run.

When dealing with multiple optima from an evolutionary optimization point of view several issues arise. The first one is how to assess/decide when or if a local optimum has been detected. Simple comparison of fitness values of two individuals is not relevant: if one of them represents a global optimum and the other a local optimum their comparison may lead to the discarding of the second one. Then the second issue arises: once detected, how to maintain the local optima through the evolution process?

Elitism versus diversity preservation are classical issues in evolutionary computation. To these it is added the one of deciding which are the elitist individuals within a population when the goal is to locate local optima also.

Several evolutionary approaches to multimodal optimization have been proposed. From the output point of view, evolutionary techniques dealing with multiple solutions can be divided in two classes:

- The first class consists of the algorithms that have as output a population containing individuals gathered around the optima. Most classical approaches belong to this class. Another decision maker is needed to extract the local optima from the provided set. The number of optima can not always be deduced from the output population. Among these approaches the most widely used are the Fitness Sharing based techniques [3], the Crowding methods [4, 6, 7].
- In the second class the output of the algorithm consists of a population containing only the optima detected. Sometimes an external population or an archive is used to store these optima. Generally these methods also provide the correct number of optima for the problem. In this class we find somewhat recent approaches such as the Adaptive Elitist Genetic Algorithm [5], Genetic Chromodynamics [2] and Multiresolution Multipopulation Differential Evolution [9].

A new evolutionary approach belonging to the second class, i.e. providing the location and number of detected optima is proposed.

2. ROAMING OPTIMIZATION

A new evolutionary technique for multimodal optimization called *Roaming technique* is proposed. Roaming technique detects multiple optima by using several subpopulations evolving in isolation. Potential optima are saved into an external population called *archive*.

Subpopulations are performing the exploration of the search space while members of the archive perform the exploitation part.

One of the problems facing multimodal optimization techniques is how to decide when an optimum has been detected. Roaming surpasses this problem by introducing a stability measure for subpopulations. This stability measure enables the characterisation of subpopulations as stable or unstable. Unstable subpopulations evolve in isolation until they become stable. The best individual in a stable subpopulation is considered to be a potential local optimum.

The number of subpopulations is a parameter of the algorithm and it is not related to the expected number of local optima. Subpopulations are not restricted to a certain area of the search space. This confers flexibility and robustness to the search mechanism.

Potential optima are saved into the *archive*. Stable subpopulations are spread over the search space in order to detect other optima.

The archive contains individuals corresponding to different optimum regions. The exploitation task is realized by refining the elite individuals in the archive.

The output of the algorithm is represented by the archive - the set of elitist individuals containing local optima.

3. BASIC ROAMING TECHNIQUE

Let f be a real valued function $f : S \rightarrow \mathbb{R}$. Consider the following optimization problem:

$$\begin{cases} \text{maximize } f(x), \\ x \in S, \end{cases}$$

where S is the solution space (the set of feasible solutions). Within roaming the fitness value $eval(x)$ of an individual x is calculated by the formula

$$eval(x) = f(x).$$

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

We may consider the following order relation on P .

DEFINITION 3.1. *We say that individual x is better than y , and we write $x \succ y$, if and only if the condition*

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

holds.

3.1 Stability measure

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

By evolving subpopulation P_i for n_{it} generations a new subpopulation P'_i having the same size as P_i is obtained.

The number n_{it} of iterations the subpopulations evolve in isolation until their stability is measured is a parameter of the algorithm called the *iteration parameter*.

Let x_i^* be the best individual in the parent subpopulation P_i . We define operator B as the set of individuals in the offspring in subpopulation P'_i that are better than x_i^* :

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

$$B(x_i^*) = \{x \in P'_i \mid x \succ x_i^*\}.$$

Using the cardinality of the set B the stability measure $SM(P_i)$ of subpopulation P_i is defined.

DEFINITION 3.2. *Stability measure of the subpopulation P_i represents the number $SM(P_i)$ defined as*

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

where x_i^ is the best individual in P_i and $\text{card } A$ represents cardinality of the set A .*

Stability measure of a subpopulation Q has the following properties:

- (i) $0 \leq SM(Q) \leq 1$;
- (ii) If $SM(Q) = 1$ then x^* is a potential local optimum;

where x^* is the best individual in Q .

DEFINITION 3.3. *A subpopulation Q is called σ -stable if the condition*

$$SM(Q) \geq \sigma \tag{1}$$

holds, where $0 \leq \sigma \leq 1$. A 1-stable subpopulation is called a stable subpopulation.

REMARKS 3.1. (i) *A subpopulation Q is called σ -unstable if it is not σ -stable.*

(ii) *A 1 - unstable subpopulation is called unstable.*

4. ARCHIVE

Within roaming, stable subpopulations are considered to contain a potential local optimum, based on the assumption that if no offspring better than their parents is produced the search of that populations may have locally converged. The best individual in a stable subpopulation is regarded as a potential local optimum. An external population called the *archive* is used to store detected potential optima.

Thus the archive is a set of elitist individuals fulfilling the following requirements:

- each individual in the archive corresponds to *one* optimum region;
- to each optimum region there is at most one individual assigned.

Thus for each optimum there should be only one individual in the archive approximating it. The number of individuals in the archive should be less than or equal to the number of optima of the problem. In order to maintain this, several methods for archiving solutions have been proposed:

- The first one (M1) uses a distance parameter δ_s in order to assess whether a new potential optimum is placed on a new optimum region or not.

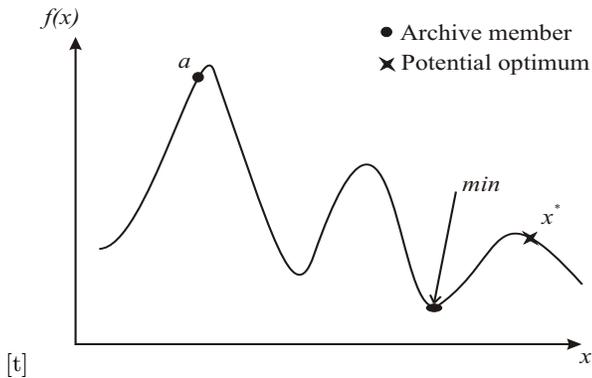


Figure 1: M2. Archiving a solution. x^* is included into the archive.

- The second method (M2) completely eliminates this parameter by using a 'valley detection' scheme.
- The third one (M3) combines the first two in order to take advantage of the robustness of the second method but also to reduce its computational complexity.

4.1 Distance based archiving

The first method (M1) of archiving solutions is based exclusively on the distance between the new potential optimum and the archive members.

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

Setting the value of δ parameter is a difficult, problem dependent, task. Requires a priori knowledge of the fitness landscape and is efficient only for problems with relatively evenly spaced optima.

4.2 Valley Detection Based Archiving

A second method (M2) for archiving solutions is based on a 'valley detection' scheme similar to that proposed by [8].

For each individual a in the archive, the global minimum min of the fitness function is calculated on the domain delimited by x^* and a . If the minimum min indicates there exists a 'valley' between x^* and every member of the archive, then add x^* to the archive. Such a situation is illustrated in figure 1.

If not, then there exist an individual a in the archive such that x^* and a are located on the same peak. In this case x^* is compared with a and only the best of them will remain in the archive. If $x^* > a$ then x^* replaces a in the archive (figure 2). If not, x^* is not added to the archive (figure 3).

The minimum of the fitness function on the domain delimited by x^* and a can be calculated using a simple genetic algorithm or an evolution strategy. It is not even necessary to reach a minimum, the search can stop once a value lower than x^* and a has been reached.

REMARK 4.1. Method M2 does not use any landscape related distance parameter. However this archiving strategy can become computational expensive, since it requires the generation - hence evaluation - of new individuals.

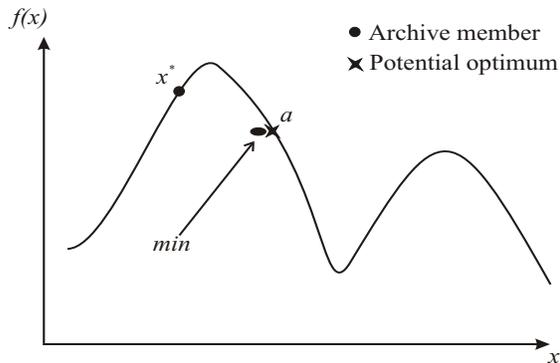


Figure 2: M2. Archiving a solution - x^* is included into the archive.

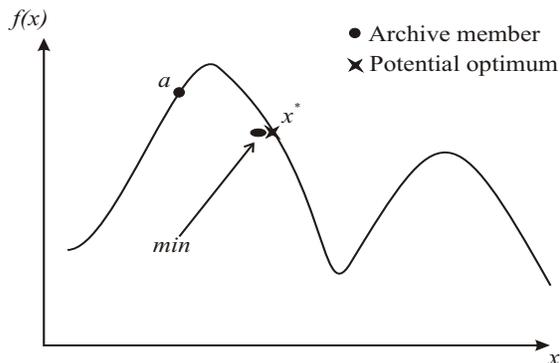


Figure 3: M2. Archiving a solution - x^* is not included into the archive.

4.3 Combination Between Valley Detection and Distance Based Archiving

The third archiving strategy M3 combines M1 and M2 by taking advantage of their strengths and diminishing their drawbacks.

Thus the distance between a potential new optimum x^* and all archive members a_i is computed. If, for some archive member a this distance is below a threshold δ then a 'valley detection' scheme is used to determine if x^* should be archived.

Method M3 diminishes the effect of the distance parameter δ as it is used in M1 and also requires a smaller number of extra fitness function evaluations than M2.

4.4 Refining the archive

Each generation, the archive content is refined in order to improve the quality of the solutions.

The following mutation operator by which offspring off is obtained from a parent p is proposed in order to achieve individuals refining:

$$off = p + \lambda \mu e^{\frac{n}{2t+1}},$$

where n represents the number of dimensions of the search space, t is the current generation number, λ is ± 1 and μ is a random number following a normal distribution between $[-0.5, 0.5]$. Each individual from the archive undergoes mutation and an offspring is created and evaluated. If the fitness

value of the offspring is better than that of the parent than the offspring is considered to be added to the archive - as a new individual - using one of the archiving methods described above. This is a precaution taken to avoid losing an optimum if the offspring is created on a different optimum region than the parent.

5. ROAMING SUBPOPULATIONS

Subpopulations search the space for local optima. Within Roaming, a stable subpopulation is considered to contain a local optimum. After a subpopulation reaches stability and its best individual is archived, the search of that subpopulation is restarted in order to locate a new optimum region.

Subpopulations selected for restarting are called *Roaming Subpopulations*. A parameter $RS \in [0, 1]$ is used as a threshold. RS -stable subpopulations are selected. The RS -unstable subpopulations are called *non-roaming subpopulations*.

6. COMPUTING NEXT GENERATION

Let $P'_i(t)$ be the offspring subpopulation from $P_i(t)$. At the next generation, the population $P(t+1)$ is composed of the restarted roaming subpopulations and the offspring subpopulations $P'_i(t)$ of all non-roaming subpopulations $P_i(t)$.

7. ROAMING ALGORITHM

Roaming algorithm can be described as follows: A set of N subpopulations are randomly generated and evaluated.

Each generation t , subpopulations $P_i(t), i = 1, \dots, N$ evolve independently and offspring subpopulations $P'_i(t), i = 1, \dots, N$ are created. The stability measure for each $P_i(t), i = 1, \dots, N$ is computed. Potential optima from stable subpopulations are archived.

Several archiving strategies are proposed in order to insure that the archive contains only one individual corresponding to each real optima of the problem.

Individuals in the archive are refined. Roaming subpopulations are selected and their search restarts. The next generation is composed of Roaming subpopulations and the offspring subpopulations corresponding to unstable subpopulations.

The algorithm stops after a termination condition is met. The output is represented by the archive - the set of elitist individuals containing local optima.

Figure 4 illustrates an iteration of the algorithm.

The parameters used by Roaming technique are listed as follows:

- N - subpopulations number
- $Popsiz$ - size of the subpopulations
- $Nrgen$ - maximum number of generations
- $Maxeval$ - maximum number of fitness function evaluations allowed
- n - iteration parameter
- RS - roaming threshold
- p_c, p_m - crossover probability and mutation rate

One of the most important features of this method is that not only it locates optima but also provides their number and positions unlike other methods that only give a set containing optima among other individuals.

The outline of the Roaming algorithm is given in Algorithm 1.

Algorithm 1 Outline of the Roaming Optimization Algorithm

```

Parameters setting;
randomly initialize subpopulations;
evaluate subpopulations;
while not termination condition do
    evolve subpopulations independently;
    calculate the stability measure for each subpopulation;
    archive potential optima;
    evolve the archive;
    restart search of roaming subpopulations;
    compute the population for the next generation as
    {Roaming subpopulations} ∪ {offsprings of non-roaming
    subpopulations};
end while

```

8. EXPERIMENTAL RESULTS

Several experimental results concerning Roaming technique are presented in the following. First the experiments set-up is presented. Three estimators for the performance of an evolutionary multimodal optimization algorithm are presented and used to compare results of three methods: Roaming algorithm, Crowding Differential Evolution proposed by [7] and Adaptive elitist Genetic Algorithm proposed by [5].

8.1 Peaks benchmark environment

The peaks benchmark represents an environment for testing multimodal optimization methods derived from the moving peaks benchmark proposed by Branke [1] for optimization in dynamical environments.

The peaks test function with n dimensions and m peaks can be formulated as:

$$F(x) = \max\{B(x), \max_{i=1, \dots, m} P(x, peak_i)\},$$

where $B(x)$ is a „basis” landscape and P is a function defining the peak shape. Each of the m peaks is defined by its location p , height h and width w :

$$peak = (p, h, w)$$

For the next experiment set-up we have used a cone peak function defined in the following manner:

$$P(x, peak) = h - \sqrt{\sum_{j=1}^n (x_j - p_j)^2} \cdot w,$$

where x belongs to the search space and $peak$ is a peak defined as mentioned above.

The following instances of the Peaks problem are used for the presented numerical experiments:

Problem Peaks1.

is defined on $[-100, 100]^2$ and has three peaks of different heights (figure 5). The exact coordinates and heights of the peaks are given in Table 1. No basis function is used.

Problem Peaks2.

has 10 peaks of varying heights (figure 6). The characteristics of the peaks are given in table 2.

The minimum distance between two peaks is 36.055 and the maximum is 197.98. No basis function is used.

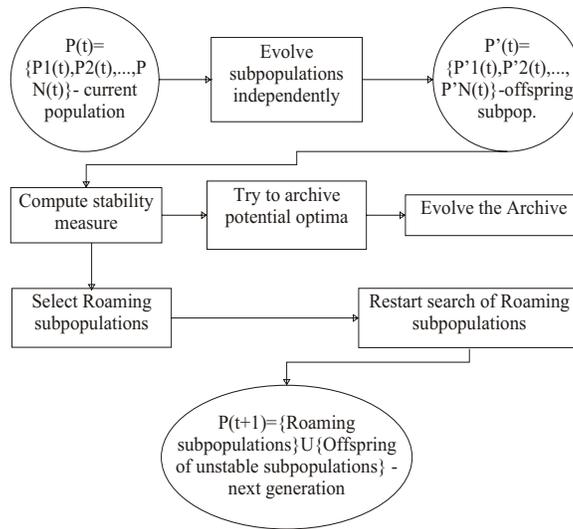


Figure 4: An iteration of Roaming algorithm

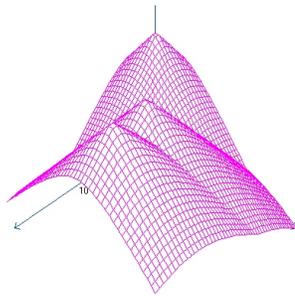


Figure 5: Problem peaks1

Table 1: Peaks1 benchmark - coordinates of the peaks

Peak	Coordinate p	Height h	Width w
1	(-70,-70)	20	0.3
2	(20, 10)	25	0.3
3	(70, 30)	30	0.3

Problem Peaks3.

is a 15-peaks function defined on $[-10, 10]$ (figure 7). All peaks have the same height. Table 3 presents the characteristics of the peaks. Minimum distance between peaks is 3.18 and the maximum is 21.98. No basis function is used.

Problem Peaks4.

is similar to *peaks3* benchmark, the only difference appearing regarding the peaks heights as presented in Table 3 and illustrated in figure 8.

Problem Peaks5.

is defined on $[-100, 100]^{10}$. It has 4 peaks situated at a minimum distance between them of 125.29 and a maximum one of 333.46 (table 4). The peaks have the same height of 100 and no basis function is used.

Table 2: Peaks2 benchmark - coordinates of the peaks

Peak	Coordinate p	h	w
1	(-80, 10)	30	0.5
2	(90, 90)	10	0.5
3	(-50, -50)	10	0.5
4	(-30, 70)	40	0.5
5	(90, 10)	15	0.5
6	(25, 20)	18	0.5
7	(50, -50)	10	0.5
8	(10, -20)	20	0.5
9	(70, -80)	50	5
10	(0, -70)	25	0.5

Table 3: Peaks3 and Peaks 4 benchmarks - coordinates of the peaks, heights and widths.

Peak	Coordinates	h peaks3	w peaks3	h peaks4
1	(-8.08 ,8.16)	4	2	7
2	(-6.04, 5.72)	4	2	4
3	(-4.64 ,1.36)	4	2	7
4	(3.4 ,5.04)	4	2	4
5	(7.44, 6.24)	4	2	4
6	(4.28, 8.12)	4	2	7
7	(6.28 ,1.4)	4	2	4
8	(-4.76, -2.92)	4	2	4
9	(-7.08, -6.16)	4	2	7
10	(-4.4 , -8.92)	4	2	4
11	(3.16, -8.56)	4	2	4
12	(1.48, -1.6)	4	2	7
13	(7.92, -3.04)	4	2	6
14	(3.64, -4.44)	4	2	7
15	(7.84, -7)	4	2	5

Table 4: Peaks7 benchmark - coordinates of the peaks.

Peak	Coordinates p	Heights h	Widths w
1	(-90 ,0, 30, -40, -70, -60, 70, 80 ,90 ,100)	100	1
2	(-100, -20, 30, 40, -80, 60, -10, 80, 90, -20)	100	1
3	(-20 ,-30, -40, 50, 60, 70, 80, 90, -50, -90)	100	1
4	(-40 ,100, -20, 40 ,100 ,50 ,-30, 70, 80, 40)	100	1

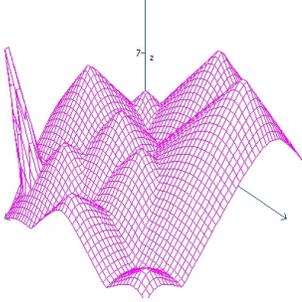


Figure 6: Problem peaks2

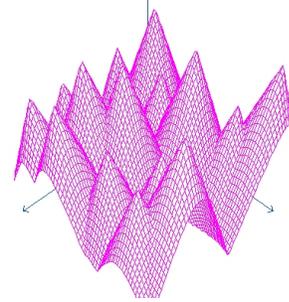


Figure 8: Problem peaks4

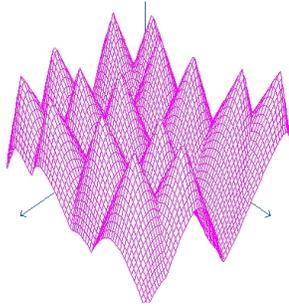


Figure 7: Problem peaks3

$$D = \frac{\sum_{i=1}^m \min_{ind \in Pop} (peak_i, ind)}{m}$$

where m is the number of peaks, ind denotes an individual and Pop denotes a population of individuals.

- standard deviation of the averages minimum distances to real optima SD over a number of runs. It indicates the variations of average minimum distances over different runs.

8.2 Performance measures

Several performance measures for multimodal search operators have been used in literature. The following performance indicators are considered throughout this work:

- peaks ratio - considered as the number of optima as reported by the algorithm divided by the real number of optima of the problem:

$$p_r = \frac{\text{number of reported peaks}}{\text{real number of peaks}}.$$

- real peaks ratio [7] - considered as the number of optima actually detected by the algorithm divided by the real number of optima. An optimum is considered to be detected if there exists one individual in the output population within 0.5 distance to it. Thus real peaks ratio is computed by the following formula:

$$rpr = \frac{\text{number of detected peaks}}{\text{real number of peaks}}.$$

- average minimum distance to the real optima - using the following formula:

Most evolutionary approaches only provide a population with individuals concentrated around optimum regions. It is up to the human decision maker to establish if all optima are detected and if all concentrations represent optimum regions or not.

Although some population may contain local optima, it may be difficult to assess which are these optima among the other individuals in the populations. It is therefore important for a multimodal evolutionary algorithm to provide not only a distinct set of optima but also their number. That is why we have considered two indices connected to the number of optima: the peaks ratio (p_r) and the real peaks ratio (rpr). Although they have similar expressions, one is computed taking into account the number of optima reported by the algorithm, while the other the number of optima detected (number of individuals within 0.5 distance to optima).

In the following section, the above indicators are averaged above 30 runs for each experiment. The Wilcoxon sum rank test has been used to analyze the results obtained.

8.3 Parameter settings

Roaming.

The results presented are obtained using the M2 archiving strategy for solving the five benchmarks presented above. All three algorithms were run up to maximum 50000 fitness

Table 5: Roaming : Setting parameters for Peaks1-5 tests

Problem	Size of subpop.	Valley detection
Peaks1	10	10
Peaks2	3	30
Peaks3	3	30
Peaks4	3	30
Peaks5	20	15

Table 6: Roaming: Common parameters for Peaks1-5

Parameter	Value
RS	0.8
Number of iterations	1
Number of subpopulations	10
Mutation rate	0.05

function evaluations. Average and standard deviations of distances to optima are presented as well as for the peaks ratio and real peaks ratio.

Parameter settings for the seven benchmarks are presented in tables 5 and 6.

AEGA.

AEGA used parameters presented in table 7. Mutation rate was set to 0.03 for all experiments.

Crowding DE.

Crowding DE used a population of 100 individuals, $CF = 100$, for all the five benchmarks.

8.4 Results

Tables 8 - 12 presents descriptive statistics related to the five benchmarks, i.e. the values for the mean of the average distances to optima over the 10 runs, the corresponding standard deviation, the minimum and median of the average distances to optima, the mean and standard deviation of the peaks ratio and the mean and standard deviation of the real peaks ratio.

Roaming and AEGA share a common feature: they evolve a population of elitist individuals corresponding to local optima. Peak ratio values are presented for this methods. Results show that Roaming peaks ratio - except for problem peaks2 - is 1 meaning that Roaming is capable to detect the regions optima are located in.

Average distances to optima obtained by Roaming are also better than the other two methods: Wilcoxon sum rank test is used to draw statistical conclusions about the results. W -values, p -values and difference in location are presented in table 13.

Table 7: Parameters used to run the AEGA algorithm for the seven peaks benchmarks

Benchmark	Pop. size	δ
peaks1	20	50
peaks2	20	50
peaks3	20	5
peaks4	20	5
peaks7	20	50

Table 8: Descriptive statistics related to problem Peaks1

Indicator	RO	AEGA	CRDE
Mean of D	0.246	0.915	0.468E-4
StDev of D	0.779	1.272	0.1441E-4
Min of D	3.907E-5	0.058	2.710E-4
Median of D	6.805E-5	0.266	4.468E-4
Mean of rp_r	0.966	0.8	1
StDev of rp_r	0.105	0.23	-
Mean of p_r	1	1.966	-
Stdev of p_r	-	0.554	-

Table 9: Descriptive statistics related to problem Peaks2

Indicator	RO	AEGA	CRDE
Mean of D	8.414	13.77	13.23
StDev of D	4.000	6.419	0.0188
Min of D	0.752	5.304	13.19
Median of D	8.53	12.36	13.24
Mean of rp_r	0.75	0.46	0.7
StDev of rp_r	0.108	0.09	-
Mean of p_r	0.86	1.12	-
Stdev of p_r	0.069	0.285	-

Table 10: Descriptive statistics related to problem Peaks3

Indicator	RO	AEGA	CRDE
Mean of D	3.518E-4	10.74	7.857E-3
StDev of D	5.219E-5	1.391	9.922E-4
Min of D	2.598E-4	8.459	5.849E-3
Median of D	3.509E-4	10.88	8.065E-3
Mean of rp_r	1	0.066	1
StDev of rp_r	-	-	-
Mean of p_r	1	0.066	-
Stdev of p_r	-	-	-

Table 11: Descriptive statistics related to problem Peaks4

Indicator	RO	AEGA	CRDE
Mean of D	1.919E-3	10.76	6.474E-3
StDev of D	3.196E-3	2.599	1.133E-3
Min of D	3.114E-4	7.642	4.409E-3
Median of D	4.409E-4	10.30	6.346E-3
Mean of rp_r	1	0.066	1
StDev of rp_r	-	-	-
Mean of p_r	1	0.066	-
Stdev of p_r	-	-	-

Table 12: Descriptive statistics related to problem Peaks5

Indicator	RO	AEGA	CRDE
Mean of D	8.565	18.32	59.21
StDev of D	7.323	8.294	0.130
Min of D	0.044	7.907	58.85
Median of D	11.00	15.07	59.26
Mean of rp_r	0.775	0.05	0.25
StDev of rp_r	0.184	0.105	-
Mean of p_r	1	118.9	-
Stdev of p_r	-	33.08	-

Table 13: Wilcoxon sum-rank test values

Problem	RO-CRDE			RO-AEGA		
	W	p	diff	W	p	diff
Peaks1	10	1.505E-3	-0.37E-4	10	1.505E-3	-0.171
Peaks2	20	0.02	-4.69	27	0.08	-4.516
Peaks3	0	1.083E-5	-7.7E-3	0	1.083E-5	-10.884
Peaks4	18	0.01469	5.779E-3	0	1.083e-05	-10.299
Peaks5	0	1.083E-5	-48.0548	21	0.02881	-10.913

p -values less than 0.05 indicate that the alternative hypothesis, i.e. results obtained by Roaming differ significantly from the others, can be accepted.

9. CONCLUSIONS AND FURTHER WORK

A new evolutionary approach to multimodal optimization called Roaming is presented.

Roaming uses subpopulations evolving in isolation to locate optima. Subpopulations are not geographically constrained to certain optimum regions. A stability measure for subpopulations is introduced in order to decide if a subpopulation has converged or not.

An external population called archive is used to store and evolve elitist individuals corresponding to local optima. Best individuals of stable subpopulations are archived using an archiving strategy.

The output of the method is the archive containing detected local optima.

Numerical experiments show that Roaming is an efficient technique compared to other evolutionary approaches to multimodal optimization.

10. REFERENCES

- [1] Jürgen Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, 2001.
- [2] D. Dumitrescu. Genetic chromodynamics. *Studia Universitatis Babeş-Bolyai, Informatica*, XLV(1):39–49, 2000.
- [3] D.E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In Grefenstette, editor, *Genetic algorithms and their Applications ICCGA '87*, pages 41–49, 1987.
- [4] K. A. D. Jong. *An analysis of the behaviour of a class of genetic adaptive systems*. PhD thesis, University of Michigan, 1975.
- [5] K. S. Leung and Y. Liang. Adaptive elitist-population based genetic algorithm for multimodal function optimization. In *GECCO 2003, LNCS 2723*, pages 1160–1171. Springer-Verlag Berlin Heidelberg, 2003.
- [6] S. W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana Champaign, Illinois Genetic Algorithm Laboratory, 1995.
- [7] Rene Thomsen. Multimodal optimization using crowding-based differential evolution. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 1382–1389, Portland, Oregon, 20–23 June 2004. IEEE Press.
- [8] Rasmus K. Ursem. Multinational evolutionary algorithms. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the Congress of Evolutionary Computation (CEC-99)*, volume 3, pages 1633–1640, Mayflower Hotel, Washington D.C., USA, 6–9 July 1999. IEEE Press.
- [9] Daniela Zaharie. Extensions of differential evolution algorithms for multimodal optimization. In Dana Petcu et al., editor, *Symbolic and Numeric Algorithms for Scientific Computing*, pages 523–534, September 2004.