Initialization with Repetitions Method for Sequential Niche Techniques

Urszula Boryczka[1] and Leszek Kłosowicz [2]

[1] University of Silesia, Institute of Computer Science, Sosnowiec, Poland,
e-mail: urszula.boryczka@us.edu.pl
[2] e-mail: leszekklosowicz@gmail.com

Abstract. This paper presents a new method of initialization population for sequential niching techniques, in which, evolutionary algorithm (EA) for determining a single local extreme has been employed. Knowledge of the localization of optima, determined in earlier runs of EA has been exploited in this approach. Initialization of a single individual consists in repeating its location until it is placed in the search subspace does not connected with any niche determined earlier. This approach contributes to the enhancement of convergence and to the improvement of achieved results.

1 Introduction

In the age of parallel computing much less attention is attached to sequential optimization methods. Initialization technique of population presented here was named "initialization with repetitions technique" and it is a way to improve those optimization methods. It has been applied in many fields, for example in numerical optimization of multimodal functions by means of sequential methods, which, by using evolutionary algorithm (e.g. simple genetic algorithm), applies population of individuals, for localization of a single extremum. One of such a method is for example a sequential niche technique which used to check initialization with repetition. This initialization technique uses knowledge concerning location of optimums, determined in earlier runs of EA. Because of that, orientation of search procedure takes place at the beginning of EA.

The aim of optimization is as follows: determine p maxima of multimodal function Φ, in k–dimensional search space. Each extreme s_i is connected with a niche, which is a part of solution space, bounded by a hyper sphere (k–sphere) with the centre in given maximum s_i and a niche radius r.

Initialization with repetitions procedure was tested (p.3) on sequential niche (SN) technique (Algorithm 1). This method was thoroughly presented in [1].

Algorithm 1: Sequential niche technique

begin
- Initialization: $M_0(x) = \Phi(x)$;
- Start the EA, memorization the best found individual s_n;
- Modified fitness function update: $M_{n+1}(x) = M_n(x) \cdot G(x, s_n)$;
- **if** $\Phi(s_n) \geq$ *solution threshold* **then**
  - showing s_n as a solution
- **if** *number of found soultions* $< p$ **then**
  - return to the second step

end

Sequential niche algorithm consist of five steps. First step it is an initialization of modified fitness function $M_0(x)$ which take raw fitness function formula $\Phi(x)$. Function $M_n(x)$ is used to evaluate individuals' fitness in evolutionary algorithm. Next an evolutionary algorithm is started and the best individual from the final population is memorized. In the third step function $M_n(x)$ is updated using derating function $G(x, s_n)$. Function $G(x, s_n)$ determines how individual's fitness reduces, if the individual appears in any of localized niche. Next it is checked whether individual s_n is a solution (its raw fitness must exceed solution threshold). Last step checks whether SN algorithm should be finished.

Algorithm 2: Simple evolutionary algorithm

begin
- $t = 0$;
- Initialization P^0;
- Evaluation P^0;
- **while** *termination condition not fulfilled* **do**
  - Reproduction P^t;
  - Genetic operations;
  - Evaluation;
  - Succession;
  - $t = t + 1$;

end

For single local extreme localization sequential niche technique can use simple evolutionary algorithm [5] (Algorithm 2). It consists of the following steps: setting generations counter $t = 0$, initialization of population P^0, evaluation of population P^0, execution of the main program's loop (until the termination condition fulfilment). This loop contains following operations: reproduction of population P^t, application of genetic operations, population evaluation, succession (creation of population P^{t+1}), incrementation of generation counter t.

2 Initialization with repetition method

Typical initialization of population comes down to random choosing individual's location within the search space. However, this approach has some disadvantages in case of sequential niche techniques. It is obvious that niches connected with local extremes, which were found in earlier runs of EA, occupy some part of problem space. So it is not advisable to accept individuals from current run of EA to be located in this part of search space. Fulfilment of this postulate will increase chances of convergence of population to an optimum, which has not yet been localized (probability of finding the same extremes will be smaller). Initialization with repetition (Algorithm 3) consists in the repetition of individual's initial location choice until it is in the subspace not connected with any localized extreme (niche). Unfortunately, some difficulty does exist, since niches of the localized extremes can occupy big part of the search space in the final phase of an optimization process. In this case only a small fragment of the solution space, from which individuals' location should be chosen, remains. It could importantly affect population initialization extension and also unfavourably impress whole the optimization algorithm time. Therefore, a parameter T was introduced for the initialization with repetition, which defines a maximal number of a location choices for a single individual. If, after T location choices the individual is still within the niche area, then it takes the location from the last choice.

Thanks to this approach there is a greater chance that the evolution of population will take place outside search space part consisting of determined niches, which will contribute to better solution space exploration.

2.1 Algorithm of initialization with repetition

Algorithm 3: Initialization with repetition algorithm

Input: P^0, $maxima$, r, k, T
Output: Initialized P^0
begin
- **foreach** ind **in** P^0 **do**
  - $\tau = 0$;
  - **repeat**
    - **for** $i = 1$ **to** k **do**
      - $ind.dimensions[i] = Random()$;
      - $i = i + 1$;
    - $EoC = true$;
    - **foreach** max **in** $maxima$ **do**
      - **if** $\sqrt{\sum_{i=1}^{k}(ind.dimensions[i] - max.dimensions[i])^2} < r$ **then**
        - $EoC = false$;
    - $\tau = \tau + 1$;
    - **if** $\tau = T$ **then**
      - $EoC = true$;
  - **until** $EoC = true$;

end

The following input data should be determined for initialization with repetition algorithm:

- set of individuals P^0, comprising initial population,
- individuals set $maxima$, representing determined local maxima,
- niche radius r,
- number of dimensions of solution space k, optimized function Φ,
- maximal choice number T.

Initialized population P^0 is a result of that algorithm. For each individual from the initial population, choice counter τ is reset and the main loop is executed. The loop's task is to designate the individual's location in search space. First instruction in the loop is to set EoC (End of Choice) flag on value *true* (it is assumed that initialization will pass in given loop iteration). Next, individual's location on each search space dimension is chosen. Checking if the Euclidean distance between such initialized individual and any of localized maxima is smaller than niche radius r is the next step of our approach. If this condition is satisfied, then EoC flag is set to *false* (individual location choice will be repeated). Next, choice counter τ is incremented and it is checked, whether τ value reached maximal choice number T. If these two values are equal, then EoC flag is set to *true*. After T unsuccessful choices of individual's localization, initialization of the algorithm stops and individual's location will be taken from the last choice. The exit from loop takes place when while checking of its termination condition, EoC flag has value *true*. In case when maximal choice number $T = 1$, then it is a typical population initialization (additionally, checking of individual's membership of some niche is unnecessary, because choice will not be repeated).

2.2 Time complexity

Initialization with repetition is not a time–consuming algorithm. Assuming, as in [1], that β means time of evaluation of the distance between individual and maximum, n – number of individuals in population P^0, and additionally γ as a time of a single value evaluation (p and k, as it was mentioned earlier, are number of maxima of interest and number of search space dimensions) then the time of initialization with repetition t_{init}, in the worst case, is:

$$t_{init} = nT \cdot (k\gamma + (p-1)\beta)$$

Initialization with repetition has this pessimistic time complexity only in case when EA determines the last p^{th} extreme and for all n individuals location is chosen T times. In other cases initialization time will be shorter, because number of determined extremes will be smaller than $p-1$ or single individual initialization will finish after less then T choice repetitions.

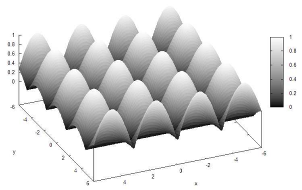
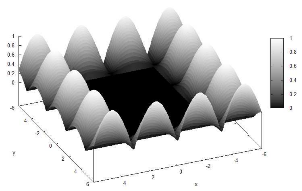
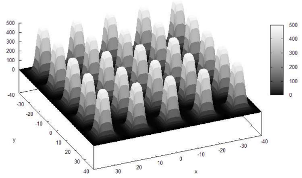
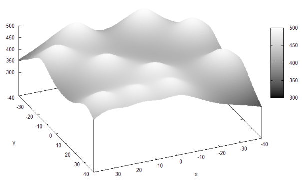
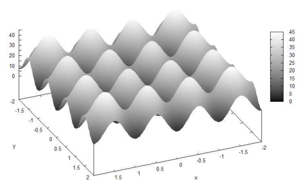
3 Experimental study

The aim of this research work is to determine how initialization with repetition method influences receiving results for numerical optimization tasks. The experimental study consists of examination of the five test functions, presented in Table 1. Initialization with repetition was tested on sequential niche technique with following parameters:

- power law derating function with the parameter $\alpha = 8.0$,
- a niche radius r determined with the help of Deb's technique [2] (it depends on optimizing function and it is normalized to the range $[0, 1]$),

- a solution threshold equals $-\infty$,

Table 1. Test functions

$\Phi_1(x,y) = \lvert\sin x \cdot \cos y\rvert$ $x, y \in [-6, 6]$ $p = 20$ $r = 0.1768$	
$\Phi_2(x,y) = \begin{cases} 0 \text{ if } x \in [-3,3] \wedge y \in [-4.5, 4.5] \\ \lvert\sin x \cdot \cos y\rvert \text{ in other case} \end{cases}$ $x, y \in [-6, 6]$ $p = 14$ $r = 0.1768$	
Shekel's Function $\Phi_3(x,y) = 500 - (0.002+$ $+ \sum_{j=0}^{24} \frac{1}{j+\sum_{i=0}^{1}(x_i - a_{ij}))^6})$ $(a_{ij}) = \begin{pmatrix} -32, -16, 0, 16, 32, -32, -16, 0, 16, 32, \dots \\ 32, 32, 32, 32, 32, 16, 16, 16, 16, 16, \dots \end{pmatrix}$ $x, y \in [-40, 40]$ $p = 25$ $r = 0.1414$	
Modified Shekel's Function $\Phi_4(x,y) = 500 - (0.002+$ $+ \sum_{j=0}^{9} \frac{1}{j+\sum_{i=0}^{1}(x_i - a_{ij}))^2})$ $(a_{ij}) = \begin{pmatrix} 16, -16, -32, 32, 32, -16, 0, -32, 0, -16 \\ 32, 32, 16, 16, 0, 0, -16, -32, -32, -32 \end{pmatrix}$ $x, y \in [-40, 40]$ $p = 10$ $r = 0.1414$	
Rastrigin's Function $\Phi_5(x,y) = 20 + (x^2 - 10 \cdot \cos(2\pi x))+$ $+(y^2 - 10 \cdot \cos(2\pi y))$ $x, y \in [-2, 2]$ $p = 16$ $r = 0.1414$	

- a number of maxima of interest p (depending on optimizing function),
- number of EA runs p.

Simple genetic algorithm with floating point representation and following parameters has been analyzed. It has been run through sequential niching to determine single extreme:

- an individual number $n = 20$,
- proportional reproduction,
- genetic operators: cross-over ($p_c = 0.9$) and mutation ($p_m = 0.01$),
- termination condition: lack of average population fitness improvement in five following generation.

Initialization with repetition research has been performed for maximal choice number T equal to: 1 (typical initialization), 5, 10, 25 and 50. For each test function and each value of parameter T, sequential niche algorithm has been run 1000 times and results were averaged.

Test results for functions Φ_1–Φ_5 are placed in Tables 2–6 respectively. Each table contains the following parameters: EN (Empty Niches) – average number of unoccupied niches per run of sequential niche algorithm; AMI (All Maxima of Interest) – average number of sequential niche technique runs in which all p maxima of interest were found; SAMI (Sum of All Maxima of Interest) – average sum of all values of maxima of interest found on one run of sequential niche technique (counted from these runs of SN algorithm in which all p maxima of interest were found); CT (Convergence Time) – average number of EA generations necessary for population convergence to the local extreme. An unoccupied niche will be understood as a niche connected with a maximum of interest which was not found.

Table 2. Test parameters values for function Φ_1

Function	$\Phi_1(x,y) = \lvert\sin x \cdot \cos y\rvert$				
T	1	5	10	25	50
EN	6.2050%	1.5900%	0.5650%	0.1150%	0.005%
AMI	12.7%	69.0%	88.7%	99.7%	99.9%
SAMI	16.7525	17.9133	18.2100	18.4029	18.4789
CT	19.1051	19.3210	19.4919	19.7027	19.9584

4 Conclusions

After the analysis of the test results of experiments with the use of initialization with repetitions the following conclusions may be reached:

- an increase the quality of the maxima of interest localization is observed, which is the most noticeable in analysis of EN and AMI parameters,
- the greatness of this improvement depends on the landscape of optimized function,
- the greatest gain is noticed for functions with regular landscape in which niches of maxima of interest completely fill the search space and are clearly separated (functions Φ_1, Φ_3 and Φ_5),

Table 3. Test parameters values for function Φ_2

Function	$\Phi_2(x,y) = \begin{cases} 0 & \text{if } x \in [-3,3] \text{ and } y \in [-4.5,4.5] \\ \lvert \sin x \cdot \cos y \rvert & \text{in other case} \end{cases}$				
T	1	5	10	25	50
EN	8.0143%	4.2571%	3.8714%	3.4929%	3.6571%
AMI	10.0%	45.0%	49.9%	54.0%	51.3%
SAMI	10.9258	11.5580	11.5826	11.6191	11.5568
CT	18.8044	18.9745	20.1765	20.1027	20.8812

Table 4. Test parameters values for function Φ_3

Function	$\Phi_3(x,y) = 500 - \left(0.002 + \sum_{j=0}^{24} \frac{1}{j+\sum_{i=0}^{1}(x_i - a_{ij}))^6}\right)$				
T	1	5	10	25	50
EN	3.6640%	0.4480%	0.0720%	0.0000%	0.0000%
AMI	25.5%	88.8%	98.2%	100.0%	100.0%
SAMI	9245.3200	11039.8000	11429.5000	11614.3000	11677.3000
CT	17.1873	17.1942	17.2389	17.352	17.4905

Table 5. Test parameters values for function Φ_4

Function	$\Phi_4(x,y) = 500 - \left(0.002 + \sum_{j=0}^{9} \frac{1}{j+\sum_{i=0}^{1}(x_i - a_{ij}))^2}\right)$				
T	1	5	10	25	50
EN	27.5600%	25.1800%	25.2300%	25.6200%	25.2500%
AMI	0.5%	0.0%	0.1%	0.2%	0.1%
SAMI	4785.9500	—	4791.0700	4787.6700	4793.29
CT	14.3821	11.8866	11.6236	11.4703	11.5794

- an increase in computation accuracy (SAMI parameter),
- a rise in accuracy is reached mainly due to the increase in maxima accuracy determined in the final phase of sequential niche algorithm,
- the best solutions are reached for parameter $T = 25$,
- an insignificant increase in costs of single local extreme determination (not more then two additional generation for the evolutionary algorithm).

Initialization with repetition technique presented here is a good tool for improving basic component of evolutionary algorithm, which is population initialization. Proposed

Table 6. Test parameters values for function Φ_5

Function	$\Phi_5(x,y) = 20 + (x^2 - 10 \cdot \cos(2\pi x)) + (y^2 - 10 \cdot \cos(2\pi y))$				
T	1	5	10	25	50
EN	3.7938%	0.7188%	0.4125%	0.3813%	0.4188%
AMI	44.7%	88.7%	93.4%	94.0%	93.3%
SAMI	619.9730	635.1420	637.8850	638.3800	638.5210
CT	20.4765	20.4821	20.6034	20.5782	20.4274

approach is especially useful for a niche localization betterment and prevention of multiple population converge to the same extremes.

5 Future Research

Improvement of evolutionary algorithm to incorporate another optimization methods seems persuasive problem for future. There is a definite scope for improvement in parameter selection. Improved parameter selection might result in enhanced algorithm's performance. The study of parameter sensibility of the population dynamics is another area for further research.

Bibliography

[1] David Beasley, David R. Bull, and Ralph R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.

[2] Kalyanmoy Deb. Genetic algorithms in multimodal function optimization. Master thesis, TCGA Report No. 89002, The University of Alabama, Department of Engineering Mechanics, 1989.

[3] Goldberg D. E. *Algorytmy genetyczne i ich zastosowania.* WNT, Warszawa, 2003.

[4] Tomasz D. Gwiazda. *Operator krzyżowania dla problemòw numerycznych*, volume I of *Algorytmy genetyczne. Kompendium.* PWN, Waszawa, 2007.

[5] John H. Holland. *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor, 1975.

[6] Arabas J. *Wykłady z algorytmòw ewolucyjnych.* WNT, Warszawa, 2001.

[7] Samir W. Mahfoud. A comparison of parallel and sequential niching methods. In Larry Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 136–143, San Francisco, CA, 1995. Morgan Kaufmann.

[8] Michalewicz Z. *Algorytmy genetyczne + struktury danych = programy ewolucyjne.* WNT, Warszawa, 2003.

[9] Michalewicz Z. and Fogel D. B. *Jak to rozwiązaè, czyli nowoczesna heurystyka.* WNT, Warszawa, 2006.