

Artificial Weed Colonies with Neighbourhood Crowding Scheme for Multimodal Optimization

Ratul Majumdar¹, Ankur Ghosh¹, Aweek Kumar Das¹, Souvik Raha¹,
Koushik Laha¹, Swagatam Das¹, and Ajith Abraham²

¹ Department of E.T.C.E, Jadavpur University, Kolkata, India

² VSB - Technical University of Ostrava, Czech Republic

{majumdar.ratul, ankur708, aveek23, rahasouvik37,
lahakoushik1}@gmail.com, swagatamdas19@yahoo.co.in,
ajith.abraham@ieee.com

Abstract. Multimodal optimization is used to find multiple global & local optima which is very useful in many real world optimization problems. But often evolutionary algorithms fail to locate multiple optima as required by the system. Also they fail to store those optima by themselves. So we have to use other selection scheme that can detect & store multiple optima along with evolutionary algorithms. Hence we use niching which is a very powerful tool in detecting & storing multiple optima. Niching methods were introduced to EAs to allow maintenance of a population of diverse individuals so that multiple optima within a single population can be located. Crowding which is a very primitive branch of niching is used here as the selection scheme with Invasive Weed Optimization (IWO) which is a ecologically inspired algorithms depicting behaviors of plants. For multimodal optimization the total search space is divided into several niches in which separately IWO is applied to find the optima in niches. The niches will also store this optima within themselves.

Keywords: Invasive Weed Optimization, Niching, Neighboring Crowding.

1 Introduction

In practical optimization problems, it is often desirable to simultaneously locate multiple global and local optima of a given objective function. A multimodal optimization task amounts to finding multiple optimal solutions and not just one single optimum, as it is done in a typical optimization study. If a point-by-point classical optimization approach is used for this task, the approach must have to be applied several times, every time hoping to find a different optimal solution.

Evolutionary Algorithms (EAs) [1, 2], due to their population-based approach, provide a natural advantage over classical optimization techniques. They maintain a population of possible solutions, which are processed every generation, and if

the multiple solutions can be preserved over all these generations, then at termination of the algorithm we will have multiple good solutions, rather than only the best solution. *Niching* [3 – 5] is a generic term referred to as the technique of finding and preserving multiple stable *niches*, or favorable parts of the solution space possibly around multiple solutions, so as to prevent convergence to a single solution. the most popular niching techniques used in conjunction with the evolutionary computation community include crowding [7], fitness sharing [6], restricted tournament selection [8], and speciation [9]. Most of existing niching methods, however, have difficulties which need to be overcome before they can be applied successfully to real-world multimodal problems. Some identified issues include: difficulties to pre-specify some niching parameters; difficulties in maintaining discovered solutions in a run; extra computational overhead, and poor scalability when dimensionality is high.

In this paper we propose a simple yet very powerful hybrid EA that synergistically combines the features of two EAs: Invasive Weed Optimization (IWO) [10] and Neighbourhood Crowding technique[19] for multimodal optimization. The reason for employing the IWO in multimodal optimization process is its high explorative power [10] over the bound-constrained search space [11]. Each weed initialized in the search space, produces seeds around it, in a controlled hyper-space thus creating virtual subpopulation of seeds. Hence after a sufficient number of generations the colony is basically spread in sub-regions surrounding promising local and global optima. The crowding scheme is applied along with IWO as described in the proposed algorithm in order to preserve the niches throughout the computation and the maximum number of global peaks can be obtained. [12-17]

2 Evolutionary Multi-modal Optimization Using Niching

If a single-objective optimization problem has more than one optima, it can be considered as multimodal optimization problem. The objective of locating & storing different optima in a single run makes it more complicated than single global optimization. *Niching methods*, formed for extending EAs to multimodal optimization, address this problem by maintaining the diversity of certain properties within the population - and this way they allow parallel convergence into multiple optimal solutions in multimodal domains. The concept of niching is inspired by the way organisms evolve in nature. As Mahoud described [3], “A niching method must be able to form and maintain multiple, diverse, final solutions, whether these solutions are of identical fitness or of varying fitness. A niching method must be able to maintain these solutions for an exponential to infinite time period, with respect to population size.” The process involves the formation of subpopulations within a population. Each subpopulation aims to locate one optimal solution and together the whole population is expected to locate multiple peaks in a single run. Several niching methods were proposed in literature e.g. crowding, deterministic crowding, fitness sharing], derating, restricted tournament selection, parallelization, clustering, clearing and speciation .We shall briefly review only crowding technique below.

Crowding and restricted tournament selection: In 1975, De Jong introduced the classical crowding method [26] known as De Jong's crowding that tries to maintain population diversity by allowing competition for limited resources among similar individuals in the population. Hence, effectively the competition takes place within each niche. Generally the similarity is measured using Euclidean distance between individuals. The algorithm compares an offspring with some randomly selected individuals from the current population. The most similar individual will be replaced if the offspring is a better solution. A parameter CF called crowding factor is used to control the size of the sample. CF is generally set to 2 or 3. Because of this low CF values, replacement errors is one of the main problem for crowding. Mahoud tried to improve the original crowding by proposing a scheme of deterministic crowding. It eliminates the CF , reduces the replacement errors, and restores selection pressure.

In very similar spirit to crowding, the restricted tournament selection [8] method selects a random sample of w (window size) individuals from the population and determines which one is the nearest to the offspring, by either Euclidean (for real variables) or Hamming (for binary variables) distance measure. The nearest member within the w individuals will compete with the offspring and the one with higher fitness will survive in the next generation.

Apart from the above, several other niching methods have also been developed over the years, including fitness sharing, clearing, speciation, derating, parallelization, and clustering. To the best of our knowledge, IWO has not been applied to solve the multimodal optimization problems till date.

3 IWO and Its Proposed Modification

IWO is a population-based algorithm based on trial and error method that copies the colonizing behavior of weeds. Weed grows its population entirely or predominantly in a geographically specified area which can be substantially large or small without any control of any external factor in a very random manner. Initially a certain number of weeds are randomly spread over the entire search range. These weeds will eventually grow up and execute the following steps and the algorithm proceeds.

3.1 Classical Invasive Weed Optimization

There are four steps in the classical IWO algorithm as described below:

Initialization: A finite number of weeds are initialized randomly in the search space.

Reproduction: Each member of the population is allowed to produce seeds depending on its own, as well as the colony's lowest and highest fitness, such that, the number of seeds produced by a weed increases linearly from lowest possible seed for a weed with the worst fitness to the maximum number of seeds for a weed with the best fitness.

Spatial Distribution: The generated seeds are randomly scattered over the d -dimensional search space by perturbing them with normally distributed random numbers with zero mean and a variable variance. This step ensures that the produced seeds will be generated around the parent weed, leading to local search around each plant. However, the standard deviation (sd) of the random function is made to decrease over iterations. If sd_{\max} and sd_{\min} are the maximum and minimum standard deviations and if pow is a real number, then the standard deviation for a particular iteration can be given as in eqn (1): (iter is current iteration)

$$sd_{ITER} = \left(\frac{iter_{\max} - iter}{iter_{\max}} \right)^{pow} (sd_{\max} - sd_{\min}) + sd_{\min}, \quad (1)$$

Competitive Exclusion: Some kind of competition between plants is needed for limiting maximum number of plants in a colony. Initially, the plants in a colony will reproduce fast and all the produced plants will be included in the existing colony, until the number of plants in the colony reaches a maximum value pop_{\max} . However, it is expected that by this time the fitter plants have reproduced more when compared to weaker plants. From then on, only the fittest plants up to pop_{\max} , among the existing ones and the reproduced ones, are taken in the colony and steps 2 to 4 are repeated until the maximum number of iterations has been reached, i.e. the colony size is fixed from thereon to pop_{\max} . This method is known as competitive exclusion and is also a selection procedure of IWO.

3.2 The Proposed Crowding IWO Algorithm

The crowding IWO algorithm is based on the application of Invasive Weed Optimization (IWO) with the selection scheme used for multimodal optimization is a niching method crowding. At first we initialize some plants or particles within the search space randomly. Then their individual fitness are calculated as per the function used. Now depending on the fitness of the individuals & the best & worst fitness of the colony each plant produces certain number of seeds which linearly decreases from best to worst fitness. Now these seeds are also spread randomly in the search space with a particular standard deviation, functions that decreases with iteration & hence plants with better fitness produces large number of seeds close to it which becomes a potential solution for the test function.

Now the crowding scheme is used here as the selection scheme where the entire population is divided into several subpopulation. That's how the niches are formed. The number of subpopulation is a niching parameter that has to be given by the user. The plants within a particular niche produces seeds & the seed most identical to its parent plant i.e. which has a minimum Euclidean distance to its parent plant is selected. If the seed has a better fitness than its parent then the seed replaces its parent plant. Now in this way total population is updated & if it

exceeds the maximum number of member of the population then the plants with better fitness are kept & others are removed from the search space. In this way the niches turn out to be potential solutions for multimodal optimization.

3.3 Pseudo Code

```

Initialize a population within search area randomly
Find the fitness value for the given function
While(iter<maxit)
     $\sigma_{iter} = ((\text{maxit}-\text{iter})^{\text{mi}}/(\text{maxit}^{\text{mi}}) * (\sigma_{ini} - \sigma_{fi}) + \sigma_{fi})$ ;
    Find the seeds of the plants
    Seed(1)=(fit(1)-maxfit)*(max no. _seeds-min no._seeds)/(max_fit-min_fit);
    Convert the seeds into plants;
    Update the total no. of plants;
    For j=1 to plants
        For i=1:seeds_of_plants (j)
            Pos_seed (i)=pos_plant(j)+  $\sigma_{iter}$ *randl;
        End
        Update position of plants
    End
    If pos_plant<LB
        Pos_plant=LB;
    End
    If pos_plant>UB
        Pos_plant=UB;
    End
    Find the Euclidean distance between parent plant &its off-springs
    Total population is distributed into sp number of subpopulation;
    Size_subpop=total_pop/sp;
    Within each subpop if the seed with min distance with its parent has a better
    fitness
        Replace the parent with that seed
        Update total population
    If(no._plants>max_plant)
        Eliminate the plants with poor fitness & keep the no_plants maxplant with
        better fitness
        Iter=iter+1;
    End while

```

4 Experiments and Results

The computing was achieved on a Pentium 4 computer running at 3.0 GHz equipped with 1.5 GB of RAM. Results are presented for benchmark functions

given in the tables below where both composite & simple test functions. The results of this algorithm is also represented in a tabular form giving a comparison between the other standard algorithms.

Parametric Setup:

- The following parameters were used while performing the experiments:
- Maximum number of iterations (maxit):1000
- Initial value of standard deviation (σ_{ini}):3
- Final Value of standard deviation (σ_{fi})=.0001
- Maximum number of seeds from each plant (Max no. _ seeds):5
- Minimum number of seeds from each plant (Min no. _ seeds):0
- Number of subgroups formed within the population: sp
- Maximum number of weeds (Max_plant):150
- The composite functions in this experiment are the composite functions in [12]

Table 1. Benchmark Functions

Name	Dim	Test Function	Range	Peaks
Two-Peak Trap	1	$f_1(x) = \begin{cases} \frac{160}{15}(15-x) & \text{for } 0 \leq x < 15 \\ \frac{200}{5}(x-15) & \text{for } 15 \leq x \leq 20 \end{cases}$	$0 \leq x \leq 20$	1
Central Two Peak Trap	1	$f_2(x) = \begin{cases} \frac{160}{10}x & \text{for } 0 \leq x < 10 \\ \frac{160}{5}(15-x) & \text{for } 10 \leq x < 15 \\ \frac{200}{5}(x-15) & \text{for } 15 \leq x \leq 20 \end{cases}$	$0 \leq x \leq 20$	1
Equal Maxima	1	$f_3(x) = \sin^6(5\pi x)$	$0 \leq x \leq 1$	5
Decreasing Maxima	1	$f_4(x) = \exp\left[-2\log(2) \cdot \left(\frac{x-0.1}{0.8}\right)^2\right] \cdot \sin^6(5\pi x)$	$0 \leq x \leq 1$	1
Uneven Maxima	1	$f_5(x) = \sin^6(5\pi(x^{\frac{3}{4}} - 0.05))$	$0 \leq x \leq 1$	5
Uneven Decreasing Maxima	1	$f_6(x) = \exp\left[-2\log(2) \cdot \left(\frac{x-0.08}{0.854}\right)^2\right] \cdot \sin^6(5\pi(x^{\frac{3}{4}} - 0.05))$	$0 \leq x \leq 1$	1
Six Hump Camel Back	2	$f_7(\vec{X}) = -4\left[\left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2\right]$	$-1.9 \leq x_1 \leq 1.9$ $-1.1 \leq x_2 \leq 1.1$	2
Himmelblau's Function	2	$f_8(\vec{X}) = 200 - (x_1^2 + x_2 - 11)^2 - (x_1 + x_2^2 - 7)^2$	$-4 \leq x_1, x_2 \leq 4$	4

The *peak accuracy* measure [20] is calculated as follows: for each global optimum to be found, the closest individual x in the population is taken and the absolute difference in fitness values is computed after maximum number of function evaluations has been elapsed. Then, all these differences are summed and divided by the number of global optima to be found. The peak accuracy calculation is shown below: (Table 3 shows a comparison of peak accuracy values of test functions and the 1st 7 composite functions)

$$peak\ accuracy = \frac{\sum_{i=1}^{\# glo\ peaks} |f(peak_i) - f(\bar{X})|}{\# glo\ peaks} \quad (2)$$

Table 2. Average Number of Peaks found for Test Functions and Comparisons with other Algorithms

Function	ε	r	sp	CIWO	CMA-ES	CDE	SDE	FER PSO	SPSO
f_1	0.05	0.5	150	1	1	1	1	0.72	0.48
f_2	0.05	0.5	150	1	1	1	1	1	0.44
f_3	0.000001	0.01	175	5	4.92	3.84	4.72	4.84	4.88
f_4	0.000001	0.01	150	1	1	0.72	1	1	1
f_5	0.000001	0.01	175	5	4.88	3.96	4.6	5	4.92
f_6	0.000001	0.01	150	1	1	0.6	1	1	1
f_7	0.000001	0.5	200	1.6	1.6	0.04	2	1.96	0.08
f_8	0.005	0.5	160	3.88	3.72	0.32	3.72	3.68	0.84
CF ₁ of [12]	0.5	1	200	1.3	1.08	0	1.8	1.08	0
CF ₂ of [12]	0.5	1	300	1	1.52	1.2	1.2	2	0
CF ₄ of [12]	0.5	1	375	3	0	0	0	0	0
CF ₅ of [12]	0.5	1	200	2	1.12	1.12	1.32	2	0
CF ₇ of [12]	0.5	1	200	1	0	0	1.8	1.52	0
CF ₈ of [12]	0.5	1	250	2	0	0	1.4	1.5	0
CF ₁₀ of [12]	0.5	1	300	2.5	0	0	1.1	1.1	0
CF ₁₃ of [12]	0.5	1	350	1	0	0	0.9	0.3	0
CF ₁₅ of [12]	0.5	1	375	2	0	0	1.6	1.2	0

Table 3. Peak accuracy of simple test functions and first seven composite functions

Function	r	CIWO	CMA-ES	CDE	SDE	FERPSO	SPSO
f_1	0.5	5.32 e-15	9.63e-09	9.46e-08	1.23e-08	5.24e-02	8.74e-02
f_2	0.5	7.23 e-11	1.87e-07	8.76e-06	3.43e-07	9.65e-04	9.45e-02
f_3	0.01	8.65e-11	9.56e-08	7.43e-05	9.53e-07	5.65e-07	3.12e-07
f_4	0.01	1.65 e-12	8.54e-09	9.43e-06	4.03e-09	8.34e-09	2.16e-09
f_5	0.01	3.92 e-10	4.36e-07	5.39e-05	8.27e-07	5.45e-09	9.58e-08
f_6	0.01	6.47 e-09	2.89e-07	8.97e-05	4.51e-07	7.41e-07	2.98e-07
f_7	0.5	1.26 e-06	2.63e-05	3.42e-04	5.33e-08	7.38e-08	3.58e-04
f_8	0.5	2.72 e-05	6.52e-04	4.27e-02	8.57e-04	8.69e-04	5.21e-02
CF ₁ of [12]	1	32.45	35.48	75.68	35.48	30.96	1.05e+02
CF ₂ of [12]	1	26.58	32.16	38.92	32.53	25.54	1.10e+02
CF ₄ of [12]	1	1.2e+02	1.89e+02	1.60e+02	1.39e+02	1.23e+02	2.49e+02
CF ₅ of [12]	1	30.71	42.56	45.62	30.55	39.52	1.07e+02
CF ₇ of [12]	1	27.64	59.87	17.32	25.66	32.62	1.47e+02

5 Conclusions

In this paper we proposed a multimodal evolutionary optimization technique that summarizes concepts powerful modern optimizing technique IWO and a selection technique of niching called crowding. The crowding IWO algorithm was tested for the optimization of fifteen benchmark functions (including five composite functions). To justify its development, results were directly compared with nine state-of-the-art evolutionary multi-modal optimizers based on the performance metrics like average number of peaks found and peak accuracy. The results of our experimental studies suggest that crowding IWO can provide a statistically superior and more consistent performance than the other standard multimodal optimization algorithms.

References

1. Bäck, T., Fogel, D., Michalewicz, Z.: Handbook of Evolutionary Computation. Oxford Univ. Press (1997)
2. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
3. Mahfoud, S.W.: Niching methods for genetic algorithms, Technical Report (TR No.: 95001), Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign (1995)
4. Sareni, B., Krahenbuhl, L.: Fitness sharing and niching methods revisited. IEEE Transactions in Evolutionary Computation 2(3), 97–106 (1998)
5. Singh, G., Deb, K.: Comparison of multimodal optimization algorithms based on evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1305–1312. ACM Press, Seattle (2006)

6. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms, pp. 41–49 (1987)
7. Thomsen, R.: Multimodal optimization using Crowding-based differential evolution. In: Proceedings of the IEEE 2004 Congress on Evolutionary Computation, pp. 1382–1389 (2004)
8. Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: Proceedings of the 6th International Conference on Genetic Algorithms, San Francisco, pp. 24–31 (1995)
9. Petrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: Proc. of 3rd IEEE Congress on Evolutionary Computation, pp. 798–803 (1996)
10. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics* 1, 355–366 (2006)
11. Chakraborty, P., Roy, G.G., Das, S., Panigrahi, B.K.: On population variance and explorative power of the invasive weed optimization algorithm. In: Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), December 9–11, pp. 227–232. IEEE Press, Coimbatore (2009)
12. Qu, B.Y., Suganthan, P.N.: Novel multimodal problems and differential evolution with ensemble of restricted tournament selection. In: IEEE Congress on Evolutionary Computation, Barcelona, Spain (July 2010)
13. Shir, O.M., Emmerich, M., Bäck, T.: Adaptive niche radii and niche shapes approaches for niching with the CMA-ES. *Evol. Comput.* 18(1), 97–126 (2010)
14. Li, X.: Niching without niching parameters: particle swarm optimization using a ring topology. *IEEE Transaction on Evolutionary Computation* 14(1) (February 2010)
15. Li, X.: Efficient differential evolution using speciation for multimodal function optimization. In: Proceedings of the Conference on Genetic and Evolutionary Computation, Washington DC, USA, pp. 873–880 (2005)
16. Li, X.: Multimodal function optimization based on fitness-euclidean distance ratio. In: Proc. Genet. Evol. Comput. Conf., pp. 78–85 (2007)
17. Gan, J., Warwick, K.: A variable radius niching technique for speciation in genetic algorithms. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2000), San Francisco, USA, pp. 96–103 (2000)
18. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. Doctoral Dissertation, University of Michigan (1975)
19. Li, X.: Niching without niching parameters: particle swarm optimization using a ring topology. *IEEE Transaction on Evolutionary Computation* 14(1) (February 2010)
20. Thomsen, R.: Multimodal optimization using Crowding-based differential evolution. In: Proceedings of the IEEE 2004 Congress on Evolutionary Computation, pp. 1382–1389 (2004)