

# A Novel Multi-objective Bionic Algorithm Based on Plant Root System Growth Mechanism

Lianbo Ma<sup>1</sup>(✉), Xu Li<sup>2</sup>, Jia Liu<sup>3</sup>, and Yang Gao<sup>4</sup>

<sup>1</sup> College of Software, Northeastern University, Shenyang 110819, China  
mlb\_vip@hotmail.com

<sup>2</sup> Benedictine University, Lisle, IL, USA

<sup>3</sup> College of Management, Shenzhen University, Shenzhen 518060, China  
1191610412@qq.com

<sup>4</sup> Academy of Information Technology, Northeastern University,  
Shenyang 110819, China

**Abstract.** This paper proposes and develops a novel multi-objective optimization scheme called MORSGO based on iterative adaptation of plant root growth behaviors. In MORSGO, the basic local and global search operators are designed deliberately based on auxin-regulated tropism of the natural root system, including branching, regrowing of different types of roots. The fast non-dominated sorting approach is employed to get priority of non-dominated solutions obtained during the search process, and the diversity over archived individuals is maintained by using dynamical crowded distance estimation strategy. Accordingly, Pareto-optimal solutions obtained by MORSGO have merits of better diversity and lower computation cost. The proposed MORSGO is evaluated on a set of bio-objective and tri-objective test functions taken from the ZDT benchmarks in terms of two commonly used metrics IGD and SPREAD, and it is compared with NSGA-II and MOEA/D. Test results verify the superiority and effectiveness of the proposed algorithm.

**Keywords:** Multi-objective algorithm · Plant root system · Optimization

## 1 Introduction

Many real-world optimization problems often involve simultaneously optimizing over two or more mutually conflicting objective functions with some uncertain constraints instead of transforming them into a single objective, and accordingly they are generally named as multi-objective optimization problems (MOPs) [1, 2]. These MOPs are more difficult to be handled than the single-objective issues due to the fact that decision makers would find a set of Pareto-optimal solutions (PS) or non-dominated solutions with a trade-off among objectives [3]. Inspired by different backgrounds, an increasing number of multi-objective evolutionary algorithms (EAs) or swarm intelligence (SI) algorithms have been proposed and extended to tackle the MOPs, prominent examples being non-dominated sorting genetic algorithm II (NSGA-II) [2], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [3], strength Pareto evolutionary

algorithm (SPEA2) [4], and multi-objective particle swarm optimization (MOPSO) [5]. However, how to improve the diversity of population or overcome the local convergence of algorithms is still a challenging issue in MO optimization [6, 7]. In this paper a new multi-objective bionic algorithm is proposed derived from plant root growth behaviors and optimal foraging, namely multi-objective root system growth optimizer (MORSGO), which adopts the branching, regrowing, mortality and tropism operations of the root system. Intuitively, the basic search technique of MORSGO can acquire appropriate balance of local search and global search and dynamical variance of population size [6, 7]. Generally, this elite retention strategy can effectively eliminate the infeasible individuals and keep feasible individuals throughout the search process [2]. Accordingly, Pareto-optimal solutions obtained by MORSGO have better diversity and lower computation cost.

## 2 Multi-objective Root System Growth Optimizer

### 2.1 Root System Growth Model (RSGO)

#### 2.1.1 Auxin Information

Assume that the total sum of auxin concentration  $F_i$  is defined as 1 in the root system, and each root's  $F_i$  value can be calculated as:

Assume that the total sum of auxin concentration  $F_i$  is defined as 1 in the root system, and each root's  $F_i$  value can be calculated as:

$$f_i = \frac{fit_i - f_{worst}}{f_{best} - f_{worst}} \quad (1)$$

$$N_i = \frac{Nutrition_i - Nutrition_{worst}}{Nutrition_{best} - Nutrition_{worst}} \quad (2)$$

$$F_i = \frac{f_i}{\sum_{j=1}^N f_j} * \xi + \frac{N_i}{\sum_{j=1}^N N_j} (1 - \xi), \quad \xi \in (0, 1) \quad (3)$$

where  $i$  is the position of the growing point,  $\xi$  is a uniform random quantity,  $N$  is the total number of the points,  $fit(\cdot)$  is the fitness value of the point and  $f_{worst}$  and  $f_{best}$  are the maximum and minimum of the current points, respectively.  $Nutrition_i$  is the current nutrient concentration of individual  $i$  and it can be expressed as:

$$Nutrition_i^{t+1} = \begin{cases} Nutrition_i^t + 1 & \text{if } f_i^{t+1} < f_i^t \\ Nutrition_i^t - 1 & \text{if } f_i^{t+1} > f_i^t \end{cases} \quad (4)$$

In each growing cycle, all individuals are sorted by their auxin concentrations in descending order [7, 8]. That is, the strong individuals with higher auxin concentrations

can be selected as main roots to branch. In our model, half of current sorted roots are selected as main roots:

$$S_m^t = P^t / 2 \quad (5)$$

where  $S_m^t$  is the number of selected main roots,  $P^t$  is the size of current population.

### 2.1.2 Mainroots Growth Operations

- **Regrowing**

**Step 1.** In each cycle, the group of main roots is constructed by selecting half population sorted according to auxin concentration.

**Step 2.** Considering effect of hydrotropism, select half of current main roots to search towards the optimal position of individuals, given by:

$$x_i^{t+1} = x_i^t + R_3(x_{best}^t - x_i^t) \quad (6)$$

where  $i \subseteq [1, S_m^t/2]$ ,  $R_3$  is random value in the range (0, 1), and  $x_{best}^t$  is the best position in the root tip group.

**Step 3.** Considering gravitropism, the rest of main roots will grow along their original directions as:

$$x_i^{t+1} = x_i^t + R_4 l_{\max}(\phi_i^t) \text{ if } x_i^t > x_i^{t-1} \quad (7)$$

where  $i \subseteq [S_m^t/2, S_m^t]$ ,  $l_{\max}$  is the maximum of root elongation length,  $R_4$  is a normally distributed random number with mean 0 and standard deviation 1;  $H(\phi_i^t)$  is a D-dimensional growth direction of the main root  $i$ ;  $\phi_i^t = (\phi_{i1}^t, \phi_{i2}^t, \dots, \phi_{i(D-1)}^t) \in R^{D-1}$  is a D-1-dimensional growth angle, given by:

$$\phi_i^{t+1} = \phi_i^t + R_5 * \omega_{\max}, \quad 0 < \omega_{\max} < \pi \quad (8)$$

where  $R_5 \in R^{D-1}$  is a uniformly distributed random sequence in the range (0, 1);  $\omega_{\max}$  is the maximum of growing angle, which is limited to  $\pi$ .

- **Branching**

**Step 1.** The nutrient concentration ( $Nutrition_i$ ) of mainroot  $i$  is compared with  $BranchG$  ( $0 < BranchG < 1$ ) to determine whether it performs branching operator:

$$\begin{cases} \text{branching} & \text{if } Nutrition_i > BranchG \\ \text{nobranching} & \text{otherwise} \end{cases} \quad (9)$$

**Step 2.** Calculated the new branching points as:

$$X_i^{t+1} = X_i^t + R_1 * D_i(\varphi_i), \quad (10)$$

where  $X_i^{t+1}$  is the new growing point from  $X_i^t$ ,  $R_1$ , is the elongate-length unit, which is a random varying from 0 to 1,  $\varphi_i$  is the growth angle  $(\varphi_{i1}, \varphi_{i2} \dots \varphi_{i(D-1)})$ .

The growth angle  $\varphi_i$  is calculated as follows:

$$\varphi^{t+1} = \varphi^t + R_2 \alpha_{init} + K * \beta_{max} / S_{max}, \quad (11)$$

where  $R_2$  is a random coefficient varying from 0 to 1,  $\alpha_{init}$  is original growth angle of the initial mainroot as zero degree,  $K$  is randomly parameter selecting the subzone,  $S_{max}$  is subzones number, and  $\beta_{max}$  is the maximum growing turning angle. Empirically,  $\beta_{max}$  is limited to  $\pi$ .

### 2.1.3 Lateral Roots Growth Operation

At the  $t^{\text{th}}$  iteration, each lateral root tip generates a random head angle and a random elongation length, given by:

$$\phi_i^{t+1} = \phi_i^t + R_5 \phi_{max} \quad (12)$$

$$x_i^{t+1} = x_i^t + R_6 l_{max} H(\phi_{max}^{t+1}) \quad (13)$$

where  $i \subseteq [0, S_i^t]$ ,  $R_5$  and  $R_6$  are random values in the range (0, 1),  $\phi_{max}$  is the maximum growing turning angle, and  $l_{max}$  is the maximum of root elongation length.

### 2.1.4 Dead-Roots Elimination

In the proposed root foraging model, it is assumed that  $N_i$  is the current population size,  $N_i$  will increase by one if a root tip splits and reduce by one if a root dies determined by auxin distribution, and it will vary in the searching process [9, 10].

## 2.2 The MORSGO Algorithm

### 2.2.1 Fast Non-dominated Sorting

In each computational iteration, two significant operations need to be calculated, namely domination count  $D_x$ , and the set of solutions  $S_x$  dominated by solution  $x$ .  $D_x$  donates the number of solutions dominated by  $x$  from current population P. Detailed procedures of this process is given in *Algorithm 1*.

**Algorithm 1.** Fast non-dominated sorting

Step 1: For each  $x \in P$ , initialize

$D_x = 0$ ; //Domination counter for solution  $x$

$S_x = \phi$ ; //Set of solutions dominated by  $x$

Step 2: Let  $q \in P$ . for each  $q$ ,

if  $x \prec q$ , let  $S_x = S_x \cup \{q\}$ ; //  $q$  is added into the solution set of  $S_x$

else if  $x \prec q$ , let  $D_x = D_x + 1$ ; //Domination counter for  $x$  is accordingly added

Step 3: If  $D_x = 0$ , //  $x$  falls within the first front set

let  $x_{rank} = 1$  and  $F_1 = F_1 \cup \{x\}$ ; //  $x$  is added into the Pareto front set

Step 4:  $i = 1$ ;

Step 5: Let  $Q = \phi$ , for each  $x \in F_i$ , //  $Q$  is defined to memorize the solutions of the next front

for each  $q \in S_x$ ,

Let  $D_x = D_x - 1$ ;

If  $D_x = 0$ , let  $x_{rank} = i + 1$ ,  $Q = Q \cup \{x\}$ ;

Step 6: Let  $i = i + 1$  and  $F_i = Q$ ,

Step 7: If  $F_i \neq \phi$ , return 5; Else, stop.

**2.2.2 The Dynamical Crowded Distance Estimation Method**

In order to pick out the redundant PS archive and maintain a certain number of solutions, the crowding-distance estimation method [2] is usually used to calculate the average distance of two points on either side of this point along each of the objectives, and it can be formulated as follows:

$$C_i = \sum_{j=1}^M (|f_{i+1j} - f_{i-1j}|) \quad (14)$$

where  $C_i$  is the crowding distance of individual  $i$ ,  $M$  is the number of objective functions,  $f_{i,j}$  is the  $j$ -th objective function value of individual  $i$ .

Then, an improved selection method, namely the dynamical crowded distance estimation method is adopted instead of the traditional crowded distance method [2]. Detailed procedures of this method are given in Algorithm 2.

**Algorithm 2.** The dynamical crowded distance estimation method

Step1: Initialize  $C_i = 0$ ;

Step2: Sort population according to each objective value;

Pre-set the boundary points to an infinite value, and this can ensure the availability of next selection process.

Step3: Calculate the crowding distance of population individuals;

Step4: Determine the minimum individual called  $ID$  in the population, then remove it;

Step5: Re-compute the crowding distance of the individual  $ID+1$  and  $ID-1$  by:

Finally, based on above mechanisms and strategies, the main procedures of MORSGO are listed in Table 1.

**Table 1.** Main procedures of MORSGA

MORSGA algorithm
<b>Step. 1:</b> Initialize Pt and set $t = 0$ ;
<b>Step. 2:</b> Classify current population into main roots group $GR_t$ and lateral roots group $GN_t$ by Eqs.(1)-(5).
<b>Step. 3:</b> Generate new solutions by Eq.(6)-Eq.(8), and Eq.(17) and then form a combined population $GC_t = GR_t \cup GN_t$ .
<b>Step. 4:</b> Calculate the fitness, auxin concentration values of $GC_t$ and sort it by executing Algorithm 1. Then each solution has a domination level. The size of $GC_t$ is $\text{size}(GR_t) + \text{size}(GN_t)$ where $\text{size}(GN_t)$ usually becomes bigger than before.
<b>Step. 5:</b> Select exactly $\text{size}(GR_t)$ best individuals by Algorithm 2 into new $MR_t$ from current $GC_t$ .
<b>Step. 6:</b> Implement lateral roots growth operators and generate new solutions as $LN_t$ and then form a combined population $LC_t = LR_t \cup LN_t$ .
<b>Step. 7:</b> Calculate the fitness, auxin concentration values of $LC_t$ and sort it by implementing Algorithm 1.
<b>Step. 8:</b> Select exactly $\text{size}(LR_t)$ best roots by Algorithm 2 as new $LR_t$ from current main root population $LC_t$ .
<b>Step. 9:</b> Remove the deteriorated roots if the corresponding auxin concentration values are 0.
<b>Step. 10:</b> Set $t = t + 1$ .
<b>Step. 11:</b> If the termination conditions are met, stop; otherwise, return to step 2.

### 3 Benchmark Test

#### 3.1 Test Problems and Performance Measures

Five representative benchmarks, including bio-objective ZDT1, ZDT2, and ZDT6, and tri-objective DTLZ1 and DTLZ6, are selected to evaluate the performance of the proposed algorithm [9]. Detailed formulas of these test instances are referred in [9].

Two performance measures are considered: (1) convergence metric  $\mathcal{V}$  based on IGD-metric [10], (2) spread metric  $\Delta$  proposed in [2]. Accordingly the detailed information about the two metrics can refer to [2, 10] respectively.

#### 3.2 Experimental Setup

Experiments are conducted with MORSGO, NSGA-II [2], and MOEA/D [3]. The common parameters for all algorithms are set as following: the population size is 100, the maximum iteration number is set to 1000, and the number of independent algorithmic runs is 20. For other parameters in MOEA/D and NSGA-II, they keep the same with their original Refs. [2, 3].

**Table 2.** Results on 30-D ZDT1, ZDT2 and 10-D ZDT6 by MORSGO, NSGA-II and MOEA/D.

Test problem		ZDT1					ZDT2					ZDT6				
		MO-RSGO	NSGA-II	MOEA/D	MO-RSGO	NSGA-II	MOEA/D	MO-RSGO	NSGA-II	MOEA/D	MO-RSGO	NSGA-II	MOEA/D	MO-RSGO	NSGA-II	MOEA/D
IGD Metric	Mean	1.24E-3	2.49E-2	4.11E-3	6.65E-2	4.22E-3	8.58E-3	4.71E-3	6.62E-2	1.95E-2						
	Median	1.24E-3	2.53E-2	4.2E-03	6.45E-2	3.32E-2	8.87E-3	4.69E-3	2.10E-2	1.95E-2						
	Best	9.80E-4	2.64E-3	3.76E-3	5.26E-2	6.93E-3	8.18E-3	4.15E-03	1.42E-2	1.93E-2						
	Worst	1.37E-3	9.98E-2	5.4133E-03	7.14E-02	1.14E-01	9.09E-3	5.37E-3	8.21E-2	2.03E-2						
Spread Metric	Std	1.35E-4	2.99E-2	4.93E-4	1.15E-02	3.06E-2	2.42E-2	3.70E-4	2.00E-2	6.21E-2						
	Mean	6.86E-1	8.20E-1	8.31E-1	7.76E-2	7.89E-1	7.46E-1	7.29E-1	7.49E-01	7.25E-1						
	Median	6.80E-1	8.23E-1	8.15E-1	2.70E-2	7.94E-1	7.24E-1	7.36E-1	7.47E-1	7.34E-1						
	Best	6.15E-1	7.11E-1	6.99E-1	6.73E-2	7.16E-1	6.36E-01	6.38E-1	6.54E-1	6.25E-1						
Std	Worst	7.72E-1	9.38E-1	1.03	8.73E-2	9.08E-01	9.3262E-01	7.67E-1	8.29E-1	8.27E-1						
	Mean	5.25E-2	6.72E-2	1.19E-1	1.82E-2	5.81E-1	8.81E-2	3.87E-2	5.06E-2	6.68E-2						

**Table 3.** Results on 7-D DTLZ1 and 22-D DTLZ6 by MORSGO, NSGA-II and MOEA/D

Test problem		DTLZ1					DTLZ6				
		MORSGO	NSGA-II	MOEA/D	MORSGO	NSGA-II	MOEA/D	MORSGO	NSGA-II	MOEA/D	
IGD Metric	Mean	3.5094	1.40E + 01	2.65E + 01	2.2848E-2	1.28E-1	6.060E-2				
	Median	1.5798	1.47E + 01	2.680E + 01	2.2339E-2	1.03E-1	6.171E-2				
	Best	8.002E-1	9.298	2.18E + 01	1.8376E-2	5.86E-2	4.012E-2				
	Worst	1.187E + 1	2.013E + 1	3.5285E + 1	2.905E-2	3.1055E-01	7.662E-2				
	Std	4.3211	3.7561	4.306	3.27E-3	7.45E-2	1.286E-2				
Spread Metric	Mean	4.999E-01	5.948E-1	5.131E-01	6.3503E-1	6.78E-1	6.685E-1				
	Median	4.816E-01	5.565E-1	5.030E-01	6.301E-1	6.84E-1	6.569E-01				
	Best	4.426E-01	5.095E-01	4.385E-01	5.4812E-1	6.30E-1	5.316E-01				
	Worst	5.711E-01	7.413E + 02	6.2429E-01	7.0279E-1	7.36E-01	8.032E-01				
	Std	5.036E-02	9.017E-2	5.375E-2	4.6848E-2	3.1935E-02	8.91E-2				

### 3.3 Results and Analysis

Tables 2 and 3 give computational results obtained by MORSGO, NSGA-II and MOEA/D in 20 runs on bi-objective and tri-objective ZDTs. It is seen from Table 2 that MORGO is able to find better mean, best and standard deviation values of the IGD metric in all bio-objective test functions except ZDT2. MOEA/D and NSGA-II obtain a little better performance than MORSGO on ZDT2 only in terms of best and median values of IGD-metric. For the  $\Delta$ -metric, in all cases with MORGO, the standard deviation and mean values are significantly satisfactory, except in NSGA-II with ZDT2 and in MOEA/D with ZDT6. This means that MORGO can get a better spread of non-dominated solutions than other algorithms. From Table 2, it is clearly seen that MORSGO yields similar performance in terms of diversity and convergence to the bio-objective test case. Specially, MORSGO is able to obtain the best spread of solutions on DTLZ1 and DTLZ6. For the IGD metric, on DTLZ1, MORGA obtains the best mean and standard deviation values, but NSGA-II also obtains the best mean values. On DTLZ2, MORGA obtain better performance in terms of mean, best and median values than the others, but MOEA/D obtains the best standard deviation.

## 4 Conclusions

In this paper, a new plant-inspired algorithm called MORSGO is designed to handle multi-objective optimization problems. In MORSGO, several basic search operators are designed and developed inspired from the plant root growth behaviors. To effectively handle non-dominated solutions, the fast non-dominated sorting approach is employed to get priority of non-dominated solutions obtained during the search process, and the dynamical crowded distance estimation strategy is used to maintain diversity of Pareto optimal solution.

A set of test functions including bio-objective and tri-objective instances have been employed to evaluate the computational performance of the proposed algorithm. Experimental results show that MORSGO has a promising ability of maintaining better population diversity and accordingly obtains better convergence, which indicates that MORSGO has potential ability of effectively tackling real-world problems in the near future.

**Acknowledgements.** This work is supported by National Natural Science Foundation of China under Grant No. 61503373; Natural Science Foundation of Liaoning Province under Grand 2015020002.

## References

1. Yi, X., Zhou, Y.: A dynamic multi-colony artificial bee colony algorithm for multi-objective optimization. *Appl. Soft Comput.* **35**, 766–785 (2015)
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)



3. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: Proceedings of Congress on Evolutionary Computation (CEC 2009), Norway, pp. 203–208 (2009)
4. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm. In: Proceedings of EUROGEN 2001: Evolutionary Methods Design Optimization Control with Applications to Industrial Problems, Athens, Greece, pp. 95–100 (2002)
5. Coello Coello, C.A., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **8**(3), 256–279 (2004)
6. Ma, L., Hu, K., Zhu, Y., Chen, H.: A hybrid artificial bee colony optimizer by combining with life-cycle. *Powell's Search Crossover Appl. Math. Comput.* **252**, 133–154 (2015)
7. Hodge, A., Berta, G., Doussan, C., Merchan, F., Crespi, M.: Plant root growth, architecture and function. *Plant Soil* **321**(1–2), 153–187 (2009)
8. Ma, L., Zhu, Y., Liu, Y., Tian, L.: A novel bionic algorithm inspired by plant root foraging behaviors. *Appl. Soft Comput.* **37**, 95–113 (2015)
9. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of Congress on Evolutionary Computation, pp. 825–830 (2002)
10. Zhou, A.M., Jin, Y.C., Zhang, Q.F., Sendhoff, B., Tsang, E.: Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In: Proceedings of Congress on Evolutionary Computation, pp. 3234–3241 (2001)