Vegetation Evolution for Numerical Optimization

Yu, Jun
Graduate School of Design, Kyushu University

Takagi, Hideyuki
Faculty of Design, Kyushu University

出版情報：2018-08-31. The Japanese Society for Evolutionary Computation
バージョン：

Vegetation Evolution for Numerical Optimization

Jun YU† and Hideyuki TAKAGI††

Graduate School of Design, Kyushu University†, Faculty of Design, Kyushu University††

1 Introduction

Evolutionary computation (EC) is a population-based technique and has developed various novel and effective optimization algorithms inspired by biological evolution and natural phenomena, such as, differential evolution (DE) [1], particle swarm optimization (PSO) [2] and fireworks algorithm (FWA) [3]. Some practitioners have also tried to introduce new search strategies or mechanisms to improve existing EC optimization performance [4, 5]. Still others focus on accelerating EC search using approximate models of fitness landscapes, such as polynomial models, kriging models and neural networks [7]. Our proposed algorithm in this paper is categorized in a new type of evolutionary algorithm.

The main objective of this paper is to summarize a general mechanism from vegetation groth and reproduction, then develop a new population-based evolutionary algorithm by simulating two different periods of vegetation, i.e. growth period and maturity period, to emphasize different optimization capabilities. The secondary one is to analyze the performance of the proposed algorithm as well as its applicability. Finally, we introduce some topics for open discussions.

The remaining paper is organized as following. We briefly review some typical survival patterns of plants in nature in the Section 2. The proposed algorithm is clearly presented in the Section 3. To evaluate the performance of the new algorithm, we compare it with other three widely known EC algorithms using 28 benchmark functions of 3 different dimensions in the Section 4. Finally, we analyze the performance of our proposal and give some open topics in the Section 5 and conclude our work in the Section 6.

2 Vegetation Survival

Many colorful species have emerged on the earth after long-term biological evolution and the changes of the natural environment. Many evolutionary algorithms were inspired by natural selection and cooperative behavior of animals. However, few people have paid their attentions to the evolution and generation of vegetation and gotten some inspirations from them. Actually, vegetation has also evolved many effective and intelligent survival methods to adapt to various environments. It means there is a huge potential to develop new algorithms by extracting and simulating vegetational behaviors.

Vegetation propagates their offspring through a variety of ways, there are four of common ways. (1) Water dispersal; plants generally growing in or on the water use this method to disperse their seeds. (2) Wind dispersal; Wind are everywhere, and blow suspended seeds far away. (3) Autologous dispersal; plants use their own power to disperse their seeds without relying on external forces. (4) Animal carrying; it mainly relies on the animal mobility of carrying their seeds to other places.

Although vegetation have evolved complex mechanisms for survival, we can obtain some inspirations from them and simplify these mechanisms to develop a new optimization algorithm. Let's roughly divid growing periods of plants into two. First, plants need to absorb nutrients to grow up. This process can be seen as an exploitation in a local area. Then, mature plants use various forces to disperse their seeds widely. This process can be viewed as an exploration in a wide range. These processes can form an optimized framework, and we call this new algorithm as vegetation evolution (VEGE).

3 Vegetation Evolution

By learning and modeling the evolution of vegetation, we realize that a simplified evolutionary model can be used to optimize problems. Here, we will focus on how to transform these realistic survival mechanisms into a new evolutionary algorithm, VEGE. In the VEGE, each seed is abstracted as an individual and all individuals form a population. Each individual is divided into two different growing periods, *growth period* and *maturity period*. Each individual experiences the growth period and then the maturity period to undertake different search capabilities. When an indivisual matures, it generates multiple seeds individuals (offspring). All seeds individuals generated by the current population form a temporary seed population. Finally, individuals in the next generation are selected from a mixed population consisting of the current population and the temporary seed population. The above process is repeated until the termination condition is satisfied and outputs eventually found the optimal solution. Fig. 1 illustrates the general process of our proposed VEGE algorithm.

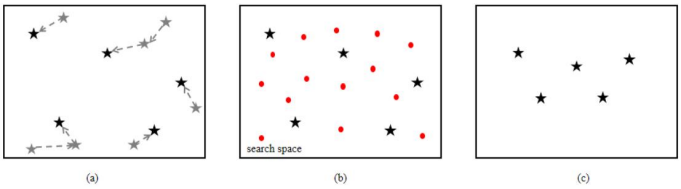

Fig. 1: The search process of our proposed VEGE algorithm. (a) Initial population is randomly generated, Dotted arrows indicate the growth directions of individuals within a local area. (b) Multiple seed individuals are generated by each individual. Red circles indicate seeds individuals and all of them form a temporary seed population. (c) Individuals in the new generation are selected from all individuals in the stpe (b). Steps (b) and (c) are iterated until a termination condition is satisfied.

Growth period

When a plant is rooted in a new environment, it absorbs nutrients to maximize its healthy growth. Inspired by this phenomenon, all individuals in this period compete in exploitation by emphasizing competition among individuals. When a new problem is given, how should we control the local growth of an individual? We introduce two new parameters to solve it: growth cycle and growth radius. They determine the number of local searches of an individual and the radius of each search, respectively. Certainly, these two parameters can be either adaptive or fixed. In our later experiments, we set these two parameters as constants. Algorithm 1 summarizes the general framework of an individual in the growth period.

Algorithm 1 The general framework of an individual searching in the growth period. GC, GR and x_i mean the growth cycle, the growth radius and the i-th individual in the current population, respectively.

```
count1 = 1.
while count1 <= GC do
    An individual generates a new exploitation individual, x_next, using x_next = x_i + GR * rand(-1, 1) in each dimension.
    if the x_next is better than the x_i then
        generated x_next individual replaces the x_i individual.
    end if
    count1 = count1 + 1.
end while
```

Maturity period

When nature plants mature, they generate a large number of seeds though only a few of them can encounter suitable environments to grow up as new plants. Inspired by this reproduction phenomenon, each VEGE individual in this period generate a large number of seeds to achieve exploration in a wide area through cooperation within the current population.

New problem is how to generate seed individuals and how many seed individuals should be generated. Similarly, we introduce two other new parameters, too: the number of seeds individuals generated by the current population and moving scaling. We select two individuals randomly and construct a differential vector to determine the direction to a newly generated offspring. Because a difference vector can adaptively use the distribu-

Algorithm 2 The general framework of an individual searching in the maturity period. MS is a moving scale; x_i is the i-th individual in the current population; x_1 and x_2 are two different randomly selected individuals and are different from x_i.

```
count2 = 1.
while count2 is less than the number of generated seed individuals of the i-th individual do
    An individual generates a new seed individual, x_seed, using x_seed = x_i + MS * (x_1 - x_2).
    Record the generated seed individual into a seed population for selecting the next generation.
    count2 = count2 + 1.
end while
```

tion information of current population.

Moving scale decides a zoom ratio in that direction and is a randomly generated in [-2,2]. Certainly, a generated offspring of each invididual can be either adaptive or fixed. In our later experiments, we set all individuals to generate the same number of offspring. Algorithm 2 summarizes the general framework of an individual in the maturity period.

The last important problem is how to select individuals in the next generation in the maturity period. Because a parent generates an offspring individual, and the better one replaces another one in the growth period, the population size does not change. However, as there are many seed individuals in the maturity period, it is important to select some individuals into the next generation reasonably. There are many famous selection methods, such as tournament selection, truncation selection, ranking selection and others. In this paper, we use greedy selection to pick up PS individuals with the higher fitness rank from a mixted population consisting of the current population and seed population.

The above is the framework of the new algorithm. Algorithm 3 demonstrates the general framework of the completed new algorithm. Note that an individual does not perform in the growth period and the mutured period continuously in one generation, but performs in either the growth period or the mutured period.

Algorithm 3 The general framework of our proposed VEGE algorithm.

```
Initialize population randomly.
Evaluate the population.
if an individual is in the growth period then
    for i = 1, ..., popuration size do
        Perform Algorithm 1 for local search (exploitation) and do not update the current population.
    end for
else
    for i = 1, ..., popuration size do
        Perform Algorithm 2 for wide search (exploration).
    end for
    Mix the current population and seed population, and apply a selection strategy to update the current population.
end if
Output found the optima.
```

4 Experimental Evaluation

To evaluate the performance of our proposed VEGE algorithm, we compare it with the three population algorithms, DE, PSO and enhanced fireworks algorithm (EFWA) [8]. We use 28 functions from the CEC2013 test suite [9] as test bed with three dimensional settings, 2-dimensions (2-D), 10-D, and 30-D. They have several landscape characteristics including shifted, rotated, unimodal and multi-modal.

For fair evaluations, we evaluate convergence along the number of fitness calls rather than generations. We test each benchmark function with 30 trial runs. The Kruskal-Wallis test and Holm's multiple comparison test are applied to check whether there is a difference among all algorithms at the stop condition, i.e. the maximum number of fitness calculations. The Table 1 shows the result of these statistical tests.

Table 1: Statistical test results of the Kruskal-Wallis test and Holm's multiple comparison for the average fitness values of 30 trial runs among 4 methods. ($A \gg B$) and ($A > B$) mean that A is significantly better than B with significant levels of 1% and 5%, respectively. ($A \approx B$) means that there is no significant difference among them.

	2-D	10-D	30-D
f_1	VEGE $\gg$ DE $>$ PSO $\gg$ EFWA	VEGE $\gg$ EFWA $\gg$ PSO $\approx$ DE	VEGE $\gg$ EFWA $\gg$ DE $\gg$ PSO
f_2	DE $\approx$ VEGE $\gg$ EFWA $\approx$ PSO	DE $\approx$ PSO $\approx$ VEGE $\gg$ EFWA	VEGE $\approx$ PSO $\approx$ EFWA $\gg$ DE
f_3	DE $\approx$ VEGE $\gg$ PSO $\approx$ EFWA	VEGE $\gg$ DE $\approx$ PSO $>$ EFWA	DE $\approx$ PSO $\approx$ VEGE $>$ EFWA
f_4	DE $\approx$ VEGE $\gg$ EFWA $\approx$ PSO	DE $\gg$ PSO $\gg$ VEGE $>$ EFWA	PSO $\approx$ EFWA $\gg$ DE $\approx$ VEGE
f_5	VEGE $\gg$ DE $>$ PSO $\approx$ EFWA	VEGE $\gg$ PSO $\approx$ DE $\approx$ EFWA	VEGE $\gg$ EFWA $\gg$ DE $\gg$ PSO
f_6	VEGE $>$ DE $\approx$ PSO $\approx$ EFWA	DE $\approx$ VEGE $\approx$ PSO $\approx$ EFWA	DE $\gg$ PSO $\approx$ VEGE $\approx$ EFWA
f_7	VEGE $\gg$ DE $\approx$ PSO $>$ EFWA	DE $\gg$ VEGE $>$ PSO $>$ EFWA	DE $\gg$ PSO $\approx$ VEGE $\gg$ EFWA
f_8	VEGE $\gg$ PSO $\approx$ DE $\approx$ EFWA	PSO $\approx$ VEGE $\approx$ DE $\approx$ EFWA	DE $\approx$ PSO $\approx$ EFWA $\approx$ VEGE
f_9	VEGE $\gg$ PSO $\approx$ DE $\approx$ EFWA	VEGE $\approx$ PSO $\gg$ EFWA $>$ DE	VEGE $\approx$ PSO $\gg$ EFWA $\gg$ DE
f_{10}	VEGE $\approx$ DE $\gg$ PSO $\approx$ EFWA	VEGE $\gg$ PSO $\gg$ DE $\approx$ EFWA	VEGE $>$ EFWA $\gg$ PSO $\approx$ DE
f_{11}	VEGE $\approx$ DE $\approx$ PSO $\approx$ EFWA	VEGE $\approx$ DE $\approx$ PSO $\gg$ EFWA	DE $\gg$ VEGE $\gg$ PSO $>$ EFWA
f_{12}	VEGE $>$ DE $\gg$ PSO $\approx$ EFWA	VEGE $\gg$ PSO $\approx$ DE $\gg$ EFWA	DE $\gg$ VEGE $>$ PSO $\gg$ EFWA
f_{13}	VEGE $\approx$ DE $\gg$ PSO $\approx$ EFWA	DE $\approx$ VEGE $\approx$ PSO $\gg$ EFWA	DE $\gg$ VEGE $\gg$ PSO $>$ EFWA
f_{14}	VEGE $\gg$ PSO $\approx$ DE $\approx$ EFWA	PSO $\approx$ EFWA $\approx$ VEGE $\gg$ DE	EFWA $\approx$ VEGE $\approx$ PSO $\gg$ DE
f_{15}	VEGE $\gg$ DE $\approx$ PSO $\approx$ EFWA	PSO $\approx$ VEGE $\gg$ EFWA $\gg$ DE	VEGE $\approx$ EFWA $\approx$ PSO $\gg$ DE
f_{16}	EFWA $>$ VEGE $\approx$ PSO $\approx$ DE	VEGE $\approx$ EFWA $\gg$ PSO $\approx$ DE	EFWA $\approx$ VEGE $\gg$ DE $\approx$ PSO
f_{17}	VEGE $\approx$ DE $>$ PSO $\approx$ EFWA	VEGE $\gg$ PSO $\approx$ DE $\gg$ EFWA	DE $>$ VEGE $\gg$ PSO $\gg$ EFWA
f_{18}	VEGE $\approx$ DE $\approx$ PSO $\approx$ EFWA	VEGE $\gg$ PSO $\approx$ DE $>$ EFWA	DE $>$ VEGE $\gg$ EFWA $\gg$ PSO
f_{19}	VEGE $\approx$ DE $\gg$ PSO $\approx$ EFWA	VEGE $\gg$ PSO $\approx$ EFWA $\approx$ DE	VEGE $\approx$ DE $\approx$ PSO $\gg$ EFWA
f_{20}	VEGE $\gg$ DE $\approx$ PSO $\approx$ EFWA	VEGE $\approx$ PSO $\approx$ DE $\gg$ EFWA	DE $>$ VEGE $\gg$ EFWA $\gg$ PSO
f_{21}	VEGE $\gg$ DE $\approx$ PSO $\approx$ EFWA	VEGE $\gg$ EFWA $\gg$ DE $\approx$ PSO	VEGE $\gg$ DE $\approx$ EFWA $\gg$ PSO
f_{22}	VEGE $\approx$ PSO $\approx$ DE $\approx$ EFWA	PSO $\approx$ EFWA $\approx$ VEGE $\gg$ DE	EFWA $\approx$ VEGE $\approx$ PSO $\gg$ DE
f_{23}	VEGE $\gg$ DE $\approx$ PSO $\gg$ EFWA	VEGE $\gg$ PSO $>$ DE $\approx$ EFWA	VEGE $\gg$ PSO $\approx$ EFWA $\gg$ DE
f_{24}	VEGE $\gg$ DE $\approx$ PSO $>$ EFWA	DE $>$ VEGE $>$ PSO $\gg$ EFWA	DE $\gg$ VEGE $>$ PSO $\gg$ EFWA
f_{25}	VEGE $\gg$ PSO $\approx$ DE $>$ EFWA	DE $\approx$ VEGE $>>$ PSO $>$ EFWA	DE $\gg$ VEGE $\gg$ PSO $\approx$ EFWA
f_{26}	VEGE $\gg$ PSO $\approx$ DE $\gg$ EFWA	VEGE $\approx$ DE $\approx$ PSO $\gg$ EFWA	DE $\approx$ VEGE $\approx$ PSO $>$ EFWA
f_{27}	VEGE $\approx$ PSO $\approx$ DE $\gg$ EFWA	VEGE $\approx$ PSO $\approx$ DE $\approx$ EFWA	DE $\approx$ VEGE $\approx$ PSO $\gg$ EFWA
f_{28}	VEGE $\gg$ PSO $\approx$ DE $\approx$ EFWA	DE $\approx$ VEGE $\gg$ PSO $>$ EFWA	DE $\approx$ VEGE $\gg$ PSO $\approx$ EFWA

5 Discussion

We begin our discussion from an explanation of the superiority of our VEGE algorithm. The algorithm realizes different search capabilities by simulating the different generation periods of vegetation. All individuals in the growth period are responsible for exploitation. There is no communication among them, and they compete independently to search for more potential areas. They move to the mature period gradually according to the progress of search. Thus, the whole population generates a large number of diversity seed individuals widely to be responsible for exploration through the cooperation among individuals.

A certain number of promising individuals are selected as individuals in the next generation, and they continue to repeat the growth trajectory. The VEGE algorithm emphasizes exploitation and exploration alternately to achieve a better balance rather than focusing on either one ability. It guarantees that its individual has a strong local search ability, at the same time uses generated seed individuals to jump out from a local area to avoid premature. Besides, the proposal VEGE can change the length of the differential vectors adaptively because they can keep the distribution information of population. Overall, the balance between exploitation and exploration can be maintained well, and it changes adaptively according to the convergence level of population.

Secondly, we would like to point out that the performance of our proposal is mainly influenced by three operations: a growth operation, a mature

operation and a selection operation. Growth operation controls the local search ability of individuals with the newly introduced parameter, the growth cycle. The larger the growth cycle is, the stronger ability to explore locally an individual has. However, once the the growth cycle is set too largely, the population is easy to fall into local areas. On the other hand, if it is too small, it may not be able to search deeply.

Mature operation can generate many distributed seed individuals widely to increase diversity and can ease the pressure of local search by using the newly introduced parameter, seeds individuals. Although many matural plants may generate thousands of seeds to survive, corresponding to the VEGE algorithm, it is unwise to generate too many seed individuals because it may increase a lot of computational cost. Also, generating too few seed individuals may not achieve an extensive search and hence cannot jump out from local areas. It is necessary to further study and observe the generation mechanism of vegetation and apply obtained konwledge to our proposal with a more rational way.

The last but also important is the selection operation. Although an individual experiences two periods, growth and maturation, many seed individuals are generated in only the maturity period. So, it is important to design how to select individuals for the next generation rationally in the mature period. In this paper, we adopted fitness ranking to select individuals. Although this operation can maintain a faster convergence speed, it also has the risk of losing diversity of population rapidly. To overcome this risk, we intend to develop a new adaptive selection operation based on the current optimization situation in our forthcoming works. In short, too much emphasis on one of them may weaken the performance of our proposal.

Finally, we apply the Kruskal-Wallis test and Holm's multiple comparison test at the stop condition among four EC algorithms to analyze the performance of our proposal. The statistical results shown in the Table 1 confirm that our proposed VEGE algorithm is effective and potential compared to the other three well-known algorithms. Besides, we can see that it performs best in all algorithms except for F_{16} in 2-D, but it is also in the second place in F_{16} in 2-D. As the dimension increases, our proposal is not ranked first in F_4, F_7 and F_{24} in 10-D, it is also not at the bottom of the ranking. However, when the dimension is up to 30, our proposal is worse than DE for about two-third of the functions, but it is basically better than PSO and EFWA. It is probably because, the number of individuals is too small along to the increased dimensions to generate sufficient number of differential vectors. The total number of possible differential vectors is $(PS-1)*(PS-2)$, where PS is a population size. In our experiments, the number of differential vectors in each generation is 72, because the PS was set to 10, while the PS of DE is 6 times of our proposal; DE could use sufficient diversity of differential vectors. Besides, due to the use of greedy selection, it might further accelerate the loss of diversity in the population. It indicates that small population size for high-dimensional problems is not recommended. It is also a good choice to use outstanding individuals from past generations to generate sufficient differential vectors. It is also needs to further investigat the relationship between spatial dimensions and population size.

6 Conclusion

We proposed a new population-based algorithm balancing exploitation and exploration by simulating the generation and propagation of natural vegetation. Each individual search widely through interindividual cooperations after performing multiple local searches through independent competition. The controlled experiment revealed that this mechanism was effective in companion to other popular EC algorithms.

In future work, we will further analyze the VEGE algorithm's convergence theoretically, improve our initial version and develop more powerful version, and expand its usage scenarios.

Acknowledgment

This work was supported in part by Grant-in-Aid for Scientific Research (18K11470, 17H06197).

References

1) R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

2) J. Kennedy and R. Eberhart, "Particle swarm optimization," *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

3) T. Ying and Z. Yuanchun, "Fireworks algorithm for optimization," *Lecture Notes in Computer Science*, no. PART 1, pp. 355–364, 2010.

4) B. Thomas, H. Ulrich, and S. Hans-Paul, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.

5) R. Mullen, D. Monekosso, S. Barman, and P. Remagnino, "A review of ant algorithms," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9608–9617, 2009.

6) Z. Jingqiao and S. A. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

7) Yaochu Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.

8) Z. Shaoqio, A. Janecek and T. Ying, "Enhanced Fireworks Algorithm," IEEE Congress on Evolutionary Computation, Cancun, Mexico, pp. 2069–2077, 2013.

9) J. Liang, B. Qu, P. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," `http://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf`, 2013.