

Performance Analysis of Vegetation Evolution

YU, Jun
JSPS : Research Fellow

TAKAGI, Hideyuki
Faculty of Design, Kyushu University

<https://hdl.handle.net/2324/2545024>

出版情報 : Proceedings IEEE International Conference on Systems, Man, and Cybernetics SMC.
2019, pp.2214-2219, 2019-10-06. IEEE

バージョン :

権利関係 :



Performance Analysis of Vegetation Evolution

Jun YU

JSPS Research Fellow
Graduate School of Design
Kyushu University
Fukuoka, Japan
yujun@kyudai.jp

Hideyuki TAKAGI

Faculty of Design
Kyushu University
Fukuoka, Japan
URL: <http://www.design.kyushu-u.ac.jp/~takagi/>

Abstract—We focus on analyzing the impact of operations of a proposed Vegetation evolution (VEGE) algorithm on its performance rather than compare it with other EC algorithms, i.e., investigate the impact of each component of the VEGE algorithm on its performance. To further analyze the performance of VEGE algorithm, we design a series of controlled experiments to investigate the contribution of each VEGE component by running them on 28 benchmark functions of 3 different dimensions. Subsequently, we summarize some our experiences on setting VEGE parameters to apply the VEGE to optimization tasks. The experimental results reveal that the maturity operation has a critical impact on performance and the number of growth operations of an individual is set as small as possible, while the number of generated seed individuals is not an important factor. Besides, we discover that population size should be gradually increased as the dimension increases. Finally, we point out several potential research directions.

Index Terms—evolutionary computation, vegetation evolution, performance analysis, optimization

I. INTRODUCTION

Evolutionary computation (EC) algorithms have been widely applied to industries thanks to their outstanding advantages, e.g. strong robustness, intelligence, usability, and others. As practical problems have become more complicated, the demand for high performance algorithms has also grown rapidly. Many practitioners try different approaches to improve the performances of EC algorithms. In general, there are three major approaches to improve their performances. The first approach is to propose new EC algorithms inspired by biological evolution, cooperative behavior, natural phenomena, and others [1], [2], [3]. The second one is to introduce efficient mechanisms or strategies into existing EC algorithms to improve optimization performance [4], [5], [6], [7]. The last approach is to use an approximation model of a fitness landscape to accelerate convergence by finding the global/local optimum areas or replacing the original optimization problems with their approximation models to reduce fitness calculation costs [8], [9], [10], [11].

Many new EC algorithms have been proposed in these decades. Although they have similar optimization frameworks, each has its own unique optimization features to ensure their convergence. Vegetation evolution (VEGE) [3] was proposed as a new family member of the EC community. It simulates vegetation growth and seed dispersion repeatedly to switch

back and forth between exploitation and exploration. We compared the proposed VEGE and differential evolution, particle swarm optimization and fireworks algorithm using 28 benchmark functions from CEC 2013 [12]. The experimental results confirmed that the proposed VEGE was effective and potential to simulate the survival process of vegetation cyclically to find the global optimal solution. As a new algorithm, there is still a lot of room to further improve its performance by analyzing the components of the VEGE algorithm. The analytical results can provide guidance to apply it as well as research directions for improving its performance.

The main objective of this paper is to analyze the impact of operations of the VEGE algorithm on its performance rather than compare it with other EC algorithms following its proposal in [3]. In other words, this paper focus on investigating the impact of each component of the VEGE algorithm on its performance by designing a series of controlled experiments using the same benchmark functions. Subsequently, we summarize some general experiences on parameter settings to use the VEGE algorithm for optimization problems. Finally, we provide some open discussion topics and several potential research directions.

Following this introduction section, the remaining paper is organized as following. We briefly summarize the optimization framework of the VEGE algorithm in the Section II and describe a series of detail controlled experiments for analyzing the contribution of each component to the performance of the VEGE algorithm in the Section III. We run these experiments using 28 benchmark functions from CEC 2013 of 3 different dimensions in the Section IV. Finally, we discuss the experimental evaluations in the Section V and conclude in the Section VI.

II. VEGETATION EVOLUTION

We shortly summarize VEGE [3] before analytical experiments. Many natural plants grow from seeds and generate new seeds when they mature for the survival of their population. Generally, many plants do not generate only one seed but hundreds or thousands of seeds to disperse them everywhere. Once the environment is suitable for them to generate, they will open a new round of growth in the rooted environment. The Fig. 1 shows the rough growth cycle of tomato, which may help to understand our proposed VEGE.

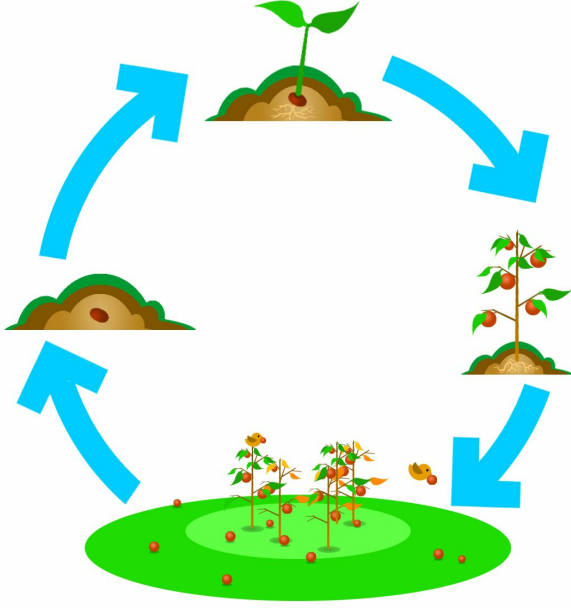


Fig. 1: Rough growth states of tomatoes. We can divide them into two periods, a growth period and a maturity period.

A novel evolutionary framework, VEGE, inspired by the above process was proposed [3]. We regarded vegetation growth and seed dispersal analogously as a local exploitation around itself and a wide exploration, respectively. 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, a *growth period* and a *maturity period* to undertake different search capabilities. Similar to most EC algorithms, VEGE generates initial population randomly. Then, these individuals experience multiple independent growth operations, i.e. a search of an individual within a local area is considered a growth. When a generated offspring is better than its parent, the parent is replaced by the offspring. Otherwise, keep as the same. Once the number of growth reaches the predefined maximum number, an individual enters its mature period. Subsequently, it generates multiple seed individuals (offspring). All seed 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 a termination condition is satisfied and finds the optimal solution. Fig. 2 illustrates the general process of the proposed VEGE algorithm.

The proposed VEGE mainly consists of the following two core operations that simulate the growth and reproduction of vegetation. Due to space limitations, we simply summarized their original intentions. Detailed implementations can be found in the literature [3].

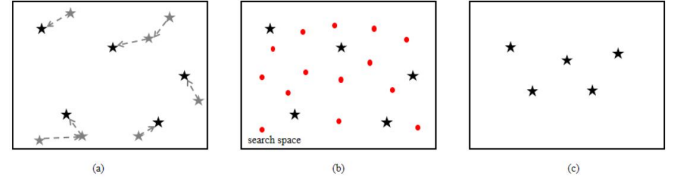


Fig. 2: The search process of the VEGE algorithm. (a) Initial population is randomly generated. Dotted arrows indicate the growth directions of individuals within a local area. (b) Each individual generates multiple seed individuals. Red circles indicate seed individuals and all of them form a temporary seed population. (c) Individuals in the new generation are selected from all individuals in the step (b). Steps (b) and (c) are iterated until a termination condition is satisfied.

Growth period

Each individual in this period searches locally and generates an offspring individual randomly within a small radius, and the winner one survives in the next generation. The parameter, *growth cycle*, is introduced to control the maximum number of growths reasonably. Once the number of growth reaches the predefined *growth cycle*, the individual enters a maturity period.

Maturity period

Each individual in this period searches widely and generates many seed individuals. We use a differential vector among individuals to determine the direction of seed propagation. The parameter, *number of seeds*, is used to control the number of seed individuals generated by each mature individual. Finally, top *PS* individuals are selected to enter the next generation from a mixed population consisting of the current maturity individuals and all seed individuals. Subsequently, a new round of the *growth period* and the *maturity period* start as the next generation.

III. EXPERIMENTAL DESIGN

We design a series of controlled experiments and analyze the contribution of each VEGE component and the effect of parameter settings on its performance to understand the VEGE algorithm deeply. We realize that a *growth period* and a *maturity period* are two core operation units of the VEGE from the above description. After initialization is completed, performing a *maturity period* firstly may generate many potential seed individuals, then top *PS* individuals with better fitness are selected and have the opportunity to perform a *growth period*. Compared to the original VEGE, VEGE with inverse order of two periods may have a high possibility of deep search in potential areas, which may speed up convergence in some cases. Thus, the first controlled experiment is designed to investigate the importance of these two operations as well as their order. Subsequently, we investigate the impact of the number of growth cycles in the *growth period*, the number of generated seeds by each individual in the *maturity period*, and population size on performance, respectively.

Experiment 1

The allocation of resources to two core operation units, a *growth period* and a *maturity period*, influence the performance of VEGE directly. Too much emphasis on either of them is not conducive to performance, but their balance is the key for efficient algorithms.

The Experiment 1 is designed to analyze the effect of two core operation periods and their implementation sequence. Four VEGE variants are compared to analyze the contributions of two operation periods to their performance in this experiment. All their parameters are set to the same. Here, the below numbers represent four VEGE variations.

- 1: original VEGE [3],
- 2: VEGE with inverse order of two periods,
- 3: VEGE without a maturity period,
- 4: VEGE without a growth period.

Experiment 2

The original VEGE controls individuals in a *growth period* to perform local searches. Each individual randomly generates a new offspring individual within a fixed search range, and the winner remains in the next generation. Here, a *growth cycle* plays an important role and controls the number of growth. If it is set large, an individual uses more computational resources for local search; if it is set small, local search does not go deep.

The Experiment 2 is designed to investigate the impact of the *growth cycle* parameter. We set all parameters to the same except the *growth cycle* parameter in this experiment. Here, below numbers represent three VEGE variations.

- 1: VEGE with 2 *growth cycles*,
- 2: VEGE with 6 *growth cycles* (original VEGE [3]),
- 3: VEGE with 10 *growth cycles*.

Experiment 3

Maturity period designed for wide search is other important factor to affect VEGE performance. Each individual in this period generates many seed individuals rather than one-to-one relationship to increase diversity. Top *PS* individuals among them are selected to survive in the next generation according to their fitness. Thus, the number of generated seed individuals is one of factors that affects exploration ability. If it is set large, many search space may be searched, but its convergence speed slows down. Conversely, inadequate search makes it easy to get into local optimal areas.

The Experiment 3 is designed to investigate the effect of the number of generated seed individuals and expect to find appropriate parameter settings. Here, the below numbers represent three VEGE variations.

- 1: VEGE generating 2 seed individuals,
- 2: VEGE generating 6 seed individuals (original VEGE [3]),
- 3: VEGE generating 10 seed individuals.

Experiment 4

Population size is also an important factor affecting VEGE performance. Although many practitioners want to find some generic rules for setting a population size, there is no exact way to guide setting it reasonably so far. Generally, inappropriate setting of a population size may affect convergence speed and even leads to inability to converge.

The Experiment 4 is designed to investigate the effect of population size and obtain some empirical settings hopefully. Here, the below numbers represent three VEGE variations which parameters are the same except population size.

- 1: VEGE with *population size* is 5,
- 2: VEGE with *population size* is 10 (original VEGE [3]),
- 3: VEGE with *population size* is 15.

Four control experiments have been introduced in detail, and each one has different investigative roles. Note that the parameter setting of the original VEGE in all above four experiments are exactly the same to use as a benchmark algorithm and carry out comparative analysis. Its parameter settings are shown in the Table II.

IV. EXPERIMENTAL EVALUATIONS

To evaluate the performance of each VEGE variation, we use 28 benchmark functions from the CEC2013 benchmark test suite [12] in our experiments. They are designed for real parameter single-objective optimization. The Table I shows their types, characteristics, variable ranges, and optimum fitness values. These landscape characteristics include shifted, rotated, global on bounds, unimodal and multi-modal.

TABLE I: Benchmark Function: Uni=unimodal, Multi=multimodal.

No.	Types	Characteristics	Ranges	Optimum fitness value
F_1	Uni	Sphere function	[-100, 100]	-1400
F_2		Rotated high conditioned elliptic function		-1300
F_3		rotated Bent Cigar function		-1200
F_4		Rotated discus function		-1100
F_5		different powers function		-1000
F_6	Multi	Rotated Rosenbrock's function	[-100, 100]	-900
F_7		Rotated Schaffers function		-800
F_8		Rotated Ackley's function		-700
F_9		Rotated Weierstrass function		-600
F_{10}		Rotated Griewank's function		-500
F_{11}		Rastrigin's function		-400
F_{12}		Rotated Rastrigin's function		-300
F_{13}		Non-continuous rotated Rastrigin's function		-200
F_{14}		Schwefel's function		-100
F_{15}		Rotated Schwefel's function		100
F_{16}		Rotated Katsuura function		200
F_{17}		Lunacek BiRastrigin function		300
F_{18}		Rotated Lunacek BiRastrigin function		400
F_{19}		Expanded Griewank's plus Rosenbrock's function		500
F_{20}		Expanded Scaffer's F_6 function		600

TABLE II: VEGE algorithm parameter settings.

population size for 2-D, 10-D, and 30-D search	10
growth cycle GC	6
growth radius GR	a random number in [-1,1]
total seed individuals for 2-D, 10-D, and 30-D search	60
moving scaling MS	a random number in [-2,2]
stop condition; max. # of fitness evaluations, MAX_{NFC} , for 2-D, 10-D, and 30-D search	1000, 10,000 and 40,000

The parameter setting of the original VEGE [3] as our test baseline is shown in the Table II. We run each benchmark

function with 30 trial runs in 3 different dimensional spaces, 2 dimension (2D), 10D and 30D, respectively. The number of fitness calls is used to evaluate convergence against rather than generations for fair evaluations. We apply a statistical test to the fitness values at the stop condition, i.e. the maximum number of fitness calculations, to detect significant differences. The statistical results of four control experiments are shown in the Tables III, IV, V and VI.

TABLE III: Statistical test results of Friedman test and Holm's multiple comparison for the average fitness values of 30 trial runs among 4 methods. Numbers in the table have the same meaning with those described in the Experiment 1 sub-section. ($A \gg B$) and ($A > B$) mean that A is significantly better than B with significance levels of 1% and 5%, respectively. ($A \approx B$) means that there is no significant difference among them.

	2D	10D	30D
F_1	$1 \approx 2 \gg 4 \gg 3$	$4 \gg 2 \approx 1 \gg 3$	$2 \approx 1 \approx 3 > 4$
F_2	$2 \approx 1 \approx 4 \gg 3$	$4 \approx 2 \approx 1 \approx 3$	$3 > 4 \gg 1 \approx 2$
F_3	$2 \approx 1 \approx 4 \gg 3$	$2 \approx 1 > 4 \gg 3$	$2 \approx 1 \gg 4 > 3$
F_4	$1 \approx 2 \approx 4 \approx 3$	$2 \approx 1 \approx 4 \approx 3$	$1 \approx 4 \approx 2 \gg 3$
F_5	$4 \gg 1 > 2 \gg 3$	$4 \gg 1 \approx 2 \gg 3$	$4 \gg 2 \approx 1 \gg 3$
F_6	$4 \approx 1 \approx 2 \gg 3$	$4 \approx 1 \approx 2 \approx 3$	$1 \approx 3 \approx 2 > 4$
F_7	$4 \gg 2 \approx 1 \gg 3$	$4 \gg 1 \approx 2 \gg 3$	$4 \gg 1 \approx 2 \gg 3$
F_8	$4 > 2 > 1 \gg 3$	$1 \approx 4 \approx 2 \approx 3$	$1 \approx 4 \approx 2 \approx 3$
F_9	$4 \gg 2 \approx 1 \gg 3$	$4 \gg 2 \approx 1 \gg 3$	$2 \approx 1 \gg 4 \gg 3$
F_{10}	$4 \approx 1 \approx 2 \gg 3$	$4 \gg 2 \approx 1 \gg 3$	$3 \gg 1 \approx 2 \gg 4$
F_{11}	$4 > 2 \approx 1 \gg 3$	$2 \approx 1 \approx 4 \gg 3$	$4 \gg 1 \approx 2 \gg 3$
F_{12}	$2 \approx 1 \approx 4 \gg 3$	$1 \approx 2 > 4 \gg 3$	$2 \approx 1 \approx 4 \gg 3$
F_{13}	$2 \approx 1 \approx 4 \gg 3$	$2 \approx 1 \approx 4 \gg 3$	$2 \approx 1 > 4 \gg 3$
F_{14}	$2 \approx 1 \approx 4 \gg 3$	$4 \approx 2 \approx 1 \gg 3$	$2 \approx 1 > 3 \approx 4$
F_{15}	$2 \approx 4 \approx 1 \gg 3$	$2 \approx 1 \gg 3 \gg 4$	$2 \approx 1 \approx 3 \gg 4$
F_{16}	$3 \approx 1 \approx 4 \approx 2$	$3 > 2 \approx 1 \approx 4$	$3 \approx 1 \approx 2 \gg 4$
F_{17}	$2 \approx 4 \approx 1 \gg 3$	$4 > 1 \approx 2 \gg 3$	$4 \gg 1 \approx 2 \gg 3$
F_{18}	$2 \approx 1 \approx 4 \gg 3$	$1 \approx 2 \approx 4 \gg 3$	$1 \approx 2 \approx 4 \gg 3$
F_{19}	$4 \approx 2 \approx 1 \gg 3$	$4 > 1 \approx 2 \gg 3$	$4 \approx 2 \approx 1 \gg 3$
F_{20}	$4 > 2 > 1 \gg 3$	$2 \approx 1 \approx 4 \gg 3$	$1 \approx 2 \approx 4 \gg 3$
F_{21}	$4 \gg 2 > 1 \gg 3$	$3 \approx 2 \approx 1 \approx 4$	$3 \gg 4 \approx 2 \approx 1$
F_{22}	$4 \approx 2 \approx 1 \gg 3$	$4 > 2 \approx 1 \gg 3$	$4 \approx 1 \approx 2 \gg 3$
F_{23}	$4 \approx 1 \approx 2 \gg 3$	$2 \approx 1 \gg 3 > 4$	$1 \approx 2 \gg 3 \gg 4$
F_{24}	$4 > 1 \approx 2 \gg 3$	$2 \approx 4 > 1 \gg 3$	$4 \gg 2 \approx 1 \gg 3$
F_{25}	$1 \approx 2 \approx 4 \gg 3$	$4 \approx 2 \approx 1 \gg 3$	$4 \gg 1 \approx 2 \gg 3$
F_{26}	$4 \approx 2 \approx 1 \gg 3$	$2 \approx 1 \approx 4 \gg 3$	$2 \approx 1 \approx 4 \approx 3$
F_{27}	$2 \approx 1 \approx 4 \gg 3$	$4 \gg 3 \approx 1 \approx 2$	$4 \gg 3 \approx 2 \approx 1$
F_{28}	$2 \approx 1 \approx 4 \gg 3$	$4 \gg 2 \approx 1 \gg 3$	$1 \approx 2 \approx 4 \gg 3$

V. DISCUSSIONS

The significant contribution of this paper is to analyze each component of VEGE and the effect of parameter settings on performance. It can help us to have a further understanding of VEGE and tune parameters optimally to adapt to a variety of problems with different characteristics. It can lead to develop more powerful improved versions, i.e. adaptive parameters to apply to multi-modal tasks. Some novel strategies inspired by many plants can be integrated into VEGE to overcome its limitations, e.g. new local search mechanism and seed generation mechanism. Subsequently, detailed analysis through four controlled experiments are described below.

TABLE IV: Statistical test results of the Friedman test and Holm's multiple comparison for the average fitness values of 30 trial runs among 3 methods. Numbers in the table have same meaning with those described in the Experiment 2 sub-section. The used symbols have the same meaning with the Table III.

	2D	10D	30D
F_1	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$	$1 \approx 3 \approx 2$
F_2	$1 \approx 3 \approx 2$	$1 \approx 3 \approx 2$	$3 \approx 2 \gg 1$
F_3	$2 \approx 1 \approx 3$	$1 \gg 2 \approx 3$	$3 > 2 \gg 1$
F_4	$2 > 3 \approx 1$	$1 \approx 2 \approx 3$	$1 > 2 > 3$
F_5	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$	$3 \gg 2 \gg 1$
F_6	$1 \approx 2 \approx 3$	$3 \approx 2 \approx 1$	$2 \approx 3 > 1$
F_7	$1 \gg 2 > 3$	$1 \gg 2 \gg 3$	$1 > 2 \approx 3$
F_8	$1 \gg 2 \approx 3$	$2 \approx 1 \gg 3$	$2 \approx 3 \approx 1$
F_9	$1 \gg 2 \approx 3$	$1 \approx 2 \approx 3$	$2 \approx 3 \approx 1$
F_{10}	$1 > 2 > 3$	$1 \gg 2 \approx 3$	$3 > 2 \gg 1$
F_{11}	$1 \gg 3 \approx 2$	$1 \gg 3 \approx 2$	$3 \approx 1 \approx 2$
F_{12}	$1 \approx 2 \approx 3$	$1 \approx 3 \approx 2$	$1 \approx 2 \approx 3$
F_{13}	$1 \gg 2 \approx 3$	$1 \gg 2 > 3$	$1 \approx 2 \approx 3$
F_{14}	$1 \approx 2 \approx 3$	$1 \approx 2 \approx 3$	$3 \approx 2 \approx 1$
F_{15}	$1 \approx 2 \approx 3$	$1 \approx 2 \approx 3$	$3 \approx 1 \approx 2$
F_{16}	$1 \approx 2 \approx 3$	$3 > 2 \approx 1$	$3 \approx 2 \approx 1$
F_{17}	$1 \approx 3 \approx 2$	$1 > 2 > 3$	$1 \gg 2 \gg 3$
F_{18}	$1 \approx 2 \approx 3$	$1 \approx 2 \approx 3$	$1 > 2 > 3$
F_{19}	$2 \approx 1 > 3$	$1 \approx 2 \approx 3$	$1 > 2 > 3$
F_{20}	$1 \gg 2 \approx 3$	$1 \approx 2 \approx 3$	$1 \approx 2 \approx 3$
F_{21}	$1 \gg 3 \approx 2$	$2 > 3 \gg 1$	$1 > 2 \approx 3$
F_{22}	$1 > 2 \approx 3$	$1 \approx 2 \approx 3$	$1 \approx 2 \approx 3$
F_{23}	$2 \approx 1 \approx 3$	$1 \approx 3 \approx 2$	$1 \approx 2 > 3$
F_{24}	$2 > 1 \gg 3$	$3 \gg 1 > 2$	$1 > 2 \approx 3$
F_{25}	$2 \approx 1 \gg 3$	$1 \gg 2 \approx 3$	$1 > 2 > 3$
F_{26}	$1 \approx 2 \approx 3$	$2 \approx 1 \approx 3$	$1 \gg 3 \approx 2$
F_{27}	$2 > 1 > 3$	$1 > 2 > 3$	$1 \approx 2 \approx 3$
F_{28}	$1 \gg 2 > 3$	$2 > 3 \gg 1$	$3 \approx 2 \gg 1$

A *growth period* and a *maturity period* are two core operation units of the VEGE, and undertake the roles of exploitation and exploration, respectively. A good balance between these two operation periods can make VEGE have high performance. The Experiment 1 (Table III) analyzes the proportion of their contributions. We can obtain following conclusions from the experimental results.

(1) A *maturity period* operation played a key role, but the *growth period* operation did not achieve the desired effect. In other words, the exploitation capability of the original VEGE [3] needs to be improved. This is probably because a *maturity period* generates many seed individuals, which increases diverse and potential individuals. Meanwhile, a *growth period* used in the VEGE is only a rough simulation of conceptual plant growth and does not realize the actual plant growth mechanism.

TABLE V: Statistical test results of the Friedman test and Holm's multiple comparison for the average fitness values of 30 trial runs among 3 methods. Numbers in the table have same meaning with those described in the Experiment 3 sub-section. The used symbols have the same meaning with the Table III.

	2D	10D	30D
F_1	$2 \approx 1 \approx 3$	$3 \approx 2 \gg 1$	$2 \approx 1 \approx 3$
F_2	$1 \approx 2 \approx 3$	$1 \approx 3 \approx 2$	$1 \gg 2 \approx 3$
F_3	$3 \approx 1 \approx 2$	$3 \approx 2 \approx 1$	$1 \approx 3 \approx 2$
F_4	$2 \approx 1 \approx 3$	$2 \approx 3 \approx 1$	$3 \approx 2 \gg 1$
F_5	$2 > 3 \approx 1$	$2 \approx 3 \gg 1$	$1 \gg 2 \gg 3$
F_6	$2 \approx 3 \gg 1$	$2 \approx 3 \approx 1$	$2 \approx 1 \approx 3$
F_7	$2 \approx 3 \gg 1$	$3 \approx 2 \gg 1$	$3 \approx 2 > 1$
F_8	$3 \approx 2 > 1$	$2 \approx 1 \approx 3$	$2 \approx 3 \approx 1$
F_9	$2 \approx 3 \approx 1$	$3 \approx 2 > 1$	$2 \approx 3 \approx 1$
F_{10}	$2 \approx 3 \approx 1$	$3 \approx 2 \approx 1$	$1 \gg 2 \gg 3$
F_{11}	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$
F_{12}	$2 \approx 3 \approx 1$	$2 \approx 3 \approx 1$	$3 > 2 \approx 1$
F_{13}	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$	$3 \approx 2 \gg 1$
F_{14}	$2 \approx 3 \approx 1$	$3 \approx 2 \approx 1$	$2 \approx 3 \approx 1$
F_{15}	$2 \approx 3 \approx 1$	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$
F_{16}	$3 \approx 1 \approx 2$	$1 \gg 3 \approx 2$	$1 \approx 2 > 3$
F_{17}	$3 \approx 1 \approx 2$	$3 \approx 2 \gg 1$	$3 > 2 \gg 1$
F_{18}	$3 \approx 2 \approx 1$	$3 \approx 2 \gg 1$	$3 \approx 2 \gg 1$
F_{19}	$3 \approx 2 \approx 1$	$2 \approx 3 \gg 1$	$3 \approx 2 \gg 1$
F_{20}	$3 > 2 \approx 1$	$3 \approx 2 \approx 1$	$2 \approx 3 > 1$
F_{21}	$3 \approx 2 \approx 1$	$1 \approx 3 \approx 2$	$1 \approx 3 \approx 2$
F_{22}	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$	$2 \approx 3 \approx 1$
F_{23}	$2 \approx 3 \approx 1$	$3 \approx 2 > 1$	$2 \approx 3 > 1$
F_{24}	$2 \approx 3 \approx 1$	$3 \approx 2 \gg 1$	$3 \approx 2 \gg 1$
F_{25}	$2 \approx 3 > 1$	$3 > 2 \approx 1$	$3 \approx 2 > 1$
F_{26}	$3 \approx 1 \approx 2$	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$
F_{27}	$3 \approx 2 \approx 1$	$2 \approx 3 \gg 1$	$3 \approx 2 \approx 1$
F_{28}	$3 \approx 2 \approx 1$	$3 \approx 2 \gg 1$	$2 \approx 3 \approx 1$

As the dimension increases, the effect of the *maturity period* becomes weak. It may be because the population is not sufficient to provide sufficient difference vectors to generate diverse seed individuals. Actually, plants have several different ways to grow in nature, and there is great potential to obtain inspiration from them. Thus, how to achieve an efficient local search will be one of research directions to improve the VEGE performance.

(2) The combination of two period operations further improve performance in some cases. Because it can own the characteristics of two operations and avoid over-emphasizing either operation not to fall into local areas. The implemented order of two period operations slightly affects the VEGE performance though it is not obvious. Thus, how to balance the two operations would be also a promising topic. Adaptively

TABLE VI: Statistical test results of the Kruskal-Wallis test and Holm's multiple comparison for the average fitness values of 30 trial runs among 3 methods. Numbers in the table have same meaning with those described in the Experiment 4 sub-section. The used symbols have the same meaning with the Table III.

	2D	10D	30D
F_1	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$
F_2	$3 \approx 2 \approx 1$	$1 \approx 3 \approx 2$	$1 \gg 2 > 3$
F_3	$1 \approx 3 \approx 2$	$2 \approx 1 \gg 3$	$1 \approx 2 \gg 3$
F_4	$2 \approx 3 \approx 1$	$2 \approx 1 \approx 3$	$1 \gg 2 \approx 3$
F_5	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$
F_6	$3 \approx 2 \approx 1$	$1 > 3 \approx 2$	$2 \approx 1 > 3$
F_7	$2 \gg 1 \gg 3$	$2 > 3 \approx 1$	$2 \approx 3 \approx 1$
F_8	$1 \gg 2 \approx 3$	$2 \approx 1 \approx 3$	$2 \approx 3 \approx 1$
F_9	$1 \gg 2 > 3$	$1 \approx 2 > 3$	$1 \approx 2 \approx 3$
F_{10}	$1 \approx 2 \gg 3$	$1 \gg 2 \gg 3$	$1 \gg 2 \gg 3$
F_{11}	$3 \approx 2 \approx 1$	$2 \approx 3 > 1$	$3 \approx 2 \approx 1$
F_{12}	$2 \approx 3 \approx 1$	$3 \approx 2 \approx 1$	$2 \approx 3 \approx 1$
F_{13}	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$	$3 \approx 2 \gg 1$
F_{14}	$2 > 3 \approx 1$	$2 \approx 3 > 1$	$2 \approx 1 \approx 3$
F_{15}	$2 \approx 3 \approx 1$	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$
F_{16}	$1 \approx 2 \approx 3$	$1 \approx 3 \approx 2$	$2 \approx 1 \approx 3$
F_{17}	$1 > 3 \approx 2$	$1 \approx 2 \gg 3$	$2 \approx 1 \gg 3$
F_{18}	$1 \approx 3 \approx 2$	$1 > 2 > 3$	$2 \approx 1 > 3$
F_{19}	$2 \approx 1 \approx 3$	$1 > 2 \gg 3$	$3 \gg 2 \approx 1$
F_{20}	$1 \gg 2 \approx 3$	$2 \approx 1 \approx 3$	$2 \approx 3 \approx 1$
F_{21}	$3 > 2 > 1$	$3 \approx 2 \approx 1$	$1 \approx 2 > 3$
F_{22}	$2 \approx 3 \approx 1$	$3 \approx 2 \approx 1$	$2 \approx 3 \approx 1$
F_{23}	$3 \approx 2 \approx 1$	$3 \approx 2 > 1$	$2 \approx 3 \gg 1$
F_{24}	$2 \approx 3 \approx 1$	$2 \approx 1 \approx 3$	$2 \approx 1 \approx 3$
F_{25}	$3 \approx 2 \approx 1$	$2 \approx 1 \approx 3$	$2 \approx 1 \gg 3$
F_{26}	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$	$3 \approx 2 \approx 1$
F_{27}	$3 \approx 2 \approx 1$	$2 \gg 1 \approx 3$	$1 \approx 2 \approx 3$
F_{28}	$3 \approx 2 \approx 1$	$2 > 3 \gg 1$	$1 \gg 2 \gg 3$

assigning the frequency of the two operations may be one of good choices to balance their search ability.

The Experiment 2 (Table IV) further analyzes the *growth period* operation. Initially, it was designed to find more potential individuals within a local area, and its number is mainly determined by the number of local searches in the *growth period* and a search strategy. The first experiment confirmed that the random search strategy used in the *growth period* was not satisfactory. Thus, this experiment was redesigned to confirm the effect of the number of random searches in the *growth period*, a *growth cycle*. The experimental results revealed that the *growth cycle* parameter has an impact on the VEGE performance. Generally, the smaller the *growth cycle* parameter, the better the performance. As the dimension increases, the *growth cycle* also need to increase appropriately. Because a *maturity period* plays a key role in VEGE, its effects

may be obscured and cannot be reflected. It also needs further verification in our future work. Anyway, we recommend that the *growth cycle* parameter should not be set too large and should be also gradually increased as the dimension increases. Adaptive tuning *growth cycle* may be a promising solution to achieve better performance.

The Experiment 3 (Table V) is to confirm how many generated seed individuals are adequate to obtain a high VEGE performance. It showed that a *maturity period* operation was crucial to the VEGE performance and explore as wide areas as possible by generating many seed individuals. It bears the responsibility of wide search by simulating the propagation of many generated seed individuals. The Experiments 1 and 2 indicate that the seed generation strategy can provide potential seeds to the next cycle. The experimental results in the Table V showed that generating too few seed individuals may result no potential search and is not recommended though it is not a key factor affecting VEGE performance. Conversely, generating too many seed individuals may create unnecessary waste of fitness calls. Because only a small number of seed individuals can remain in the next generation, and many seed individuals die out during selection competition. Thus, how to improve the utilization of seed individuals would be also one of our future works. In a way, the parameter, *number of seeds* is also related to the population size. Balancing the resource allocation between population size and total generated seed individuals is one of our next works. It is also a potential topic that each individual determines the number of its generated seeds adaptively based on its own optimization process, i.e. fitness, potential, and others.

The Experiment 4 (Table VI) is to find some rules for setting a reasonable population size for optimal performance. Population size is a long-lasting topic in EC community, and many researchers tried to propose its general tuning rules. We can roughly conclude that it is proportional to the dimension of tasks from the experimental results. It may be due to the fact that large population size can provide sufficient number of difference vectors to spread the diversity of seed individuals in high dimensional search space. Suppose that the population size is set to N , then it can provide the $(N^2 - N)$ directions of propagation. Thus, it should be increased according to the increase of the task dimensions.

We are going to focus on the relationship between population sizes and the characteristics of tasks and develop a dynamic population strategy combining the task characteristics and optimization processes.

VI. CONCLUSION

We designed a series of experiments to analyze performance of VEGE and obtain reasonable parameter settings rules. From these experiments, we have found that a *maturity period* operation had a major contribution to the VEGE performance, while the effect of a *growth period* operation was not obvious.

The results of statistical test confirmed that the parameter, *number of seeds* individuals, does not cause huge fluctuations in performance, while the *growth cycle* parameter affects the

VEGE performance; the smaller it, the better the performance. We do not recommend setting them too large. Finally, we have found that insufficient population size might hinder convergence and it should have been gradually increased as the dimension of tasks increases.

In our future works, we intend to further improve performance against existing defects based on our investigations. We will try to adopt a new local search strategy to enhance its exploitation and introduce some efficient mechanisms to enhance its optimization capabilities.

ACKNOWLEDGMENT

This work was supported in part by Grant-in-Aid for Scientific Research (18K11470, 19J11792).

REFERENCES

- [1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks - Conference Proceedings*, vol. 4, Perth, Western Australia, 1995, pp. 1942–1948.
- [2] M. Dorigo, V. Maniezzo, and A. Colomni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
- [3] J. Yu and H. Takagi, "Vegetation evolution for numerical optimization," in *JPNSEC International Workshop on Evolutionary Computation*, Shenzhen, China, 2018, pp. 49–54. [Online]. Available: https://catalog.lib.kyushu-u.ac.jp/opac_download_md/2230938/ECmeeting083118.pdf
- [4] T. Back, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.
- [5] R. J. 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] J. Yu and H. Takagi, "Acceleration for fireworks algorithm based on amplitude reduction strategy and local optima-based selection strategy," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10385 LNCS, Fukuoka, Japan, 2017, pp. 477–484.
- [7] J. Yu, Y. Pei, and H. Takagi, "Competitive strategies for differential evolution," in *The 2018 IEEE International Conference on Systems, Man, and Cybernetics*, Miyazaki, Japan, 2018, pp. 268–273.
- [8] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.
- [9] N. Murata, R. Nishii, H. Takagi, and Y. Pei, "Analytical estimation of the convergence point of populations," in *2015 IEEE Congress on Evolutionary Computation*, Sendai, Japan, 2015, pp. 2619–2624.
- [10] J. Yu, Y. Pei, and H. Takagi, "Accelerating evolutionary computation using estimated convergence points," in *2016 IEEE Congress on Evolutionary Computation, CEC 2016*, Vancouver, Canada, 2016, pp. 1438–1444.
- [11] J. Yu, Y. Tan, and H. Takagi, "Accelerating fireworks algorithm with an estimated convergence point," in *The Ninth International Conference on Swarm Intelligence*, Shanghai, China, 2018, pp. 263–272.
- [12] 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," 2013. [Online]. Available: http://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimization/CEC2013_RealParameterOptimization_TechnicalReport.pdf