applied sciences

Article

Extended Evolutionary Algorithms with Stagnation-Based Extinction Protocol

Gan Zhen Ye and Dae-Ki Kang *

Department of Computer Engineering, Dongseo University, Busan 47011, Korea; zyzygan@gmail.com
* Correspondence: dkkang@dongseo.ac.kr; Tel.: +82-51-320-1724

Abstract: Extinction has been frequently studied by evolutionary biologists and is shown to play a significant role in evolution. The genetic algorithm (GA), one of popular evolutionary algorithms, has been based on key concepts in natural evolution such as selection, crossover, and mutation. Although GA has been widely studied and implemented in many fields, little work has been done to enhance the performance of GA through extinction. In this research, we propose stagnation-driven extinction protocol for genetic algorithm (SDEP-GA), a novel algorithm inspired by the extinction phenomenon in nature, to enhance the performance of classical GA. Experimental results on various benchmark test functions and their comparative analysis indicate the effectiveness of SDEP-GA in terms of avoiding stagnation in the evolution process.

Keywords: stagnation; extinction; genetic algorithm

Citation: Gan, Z.Y.; Kang, D.-K. Extended Evolutionary Algorithms with Stagnation-Based Extinction Protocol. *Appl. Sci.* **2021**, *11*, 3461. https://doi.org/10.3390/app11083461

Academic Editor: Mauro Castelli

Received: 10 February 2021
Accepted: 8 April 2021
Published: 12 April 2021

1. Introduction

The biological metaphor of evolution was being applied to computation since as early as 1950 [1–3], and the genetic algorithm (GA) is one of the evolutionary algorithms inspired by this process of natural evolution. The theory of evolution was introduced by Darwin in 1858 [4] and together with Weismann's theory of natural selection [5] and Mendel's concept of genetics [6] they formed the neo-Darwinian paradigm [7] which the genetic algorithm was based on. Although it is clear that in the process of evolution, there are fundamental components such as natural selection, reproduction (crossover), and mutation, it is often neglected that extinction plays an important role in this process too. Several evolutionary biologists have raised this question and argued that extinction plays a significant role in evolution [4,8], but little has been done in the field of evolutionary computation where extinction is still being neglected or ignored.

When we take a look at the efforts in improving the performance of GA, we can categorize the efforts into two groups: One tries to improve GA through modeling GA closer to natural evolution by introducing new biological operators or modifying existing ones to closely mimic processes observable in nature [9–11]. The other group mainly designs operators that were tailored to suit specific problems, and these operators have no correspondence to nature of any sort whatsoever [12–14]. Based on the increased performance and accuracy of GAs that were modeled to closely mimic the natural evolution process, we came to a deduction: the closer we model our GA towards natural evolution, the better it will perform. With this in mind, this research aims to introduce a novel genetic algorithm inspired by the extinction phenomenon in nature: stagnation-driven extinction protocol for genetic algorithm (SDEP-GA).

The idea of SDEP-GA is analogous to Dropout regularization [15]. In training the neural network, Dropout operation is randomly omitting a subset of hidden units at each training iteration in the neural network. This random removal of hidden units at each training iteration turns out to be a combination of exponentially many different neural networks which share the same weights. Thus, Dropout regularization basically attempts to average multiple models to improve the performance of the neural network. Srivastava et al. [16]

have shown that dropout reduces overfitting and improves generalization. Removing chromosomes in SFEP-GA is similar to omitting hidden units in dropout, but in SDEP-GA, removing chromosomes with the extinction probability in a random way or a targeted way can allow greater possibility of exploration for new solutions.

The remaining of this paper is organized as follows. Section 2 presents a description of GA, specially focusing on simple GA. Section 3 provides the design and architecture of our proposed algorithm, stagnation-driven extinction protocol for GA (SDEP-GA). Section 4 will cover the experimental design of our proposed algorithm together with the experimental results and discussion of SDEP-GA compared against Simple GA in terms of performance. Finally, Section 5 gives a summary, a review of our contribution, and possible directions for future work.

2. Genetic Algorithm

Genetic algorithm (GA) was invented by Holland in the 1960s and formally introduced by Holland in 1975 [17]. GA is based on ideas from Darwinian evolution and thus it adopts some biological terminology. To assist in the understanding of SDEP-GA, we outline Simple GA algorithm.

2.1. Simple GA

Simple GA is the simplest form of GA, which consists of three types of operators commonly used in other GAs. The three basic operators are as below, and they will be further explained later.

- Selection
- Crossover
- Mutation

Figure 1 show the flowchart of Simple GA process, and it can be interpreted as follows:

1. Generate a random population of n chromosomes.
2. Evaluate the fitness of each chromosome.
3. If the termination criterion is not met, then continue the following steps. Otherwise, return the best chromosome found.
4. Repeat the following until n chromosomes are created:
   (a) Select a pair of parent chromosomes.
   (b) Perform crossover with the selected pair of parent chromosome. A pair of offspring is created in this process.
   (c) Apply mutation operator on the pair of offspring.
5. Replace current population with the newly created n offspring.
6. Go to step 2 .

Each complete iteration (step 1–5 or step 2–5) performed is called a generation, and the entire set of generations is called a run. At the end of each run, the fittest chromosome or the best approximated candidate solution is returned to the user.

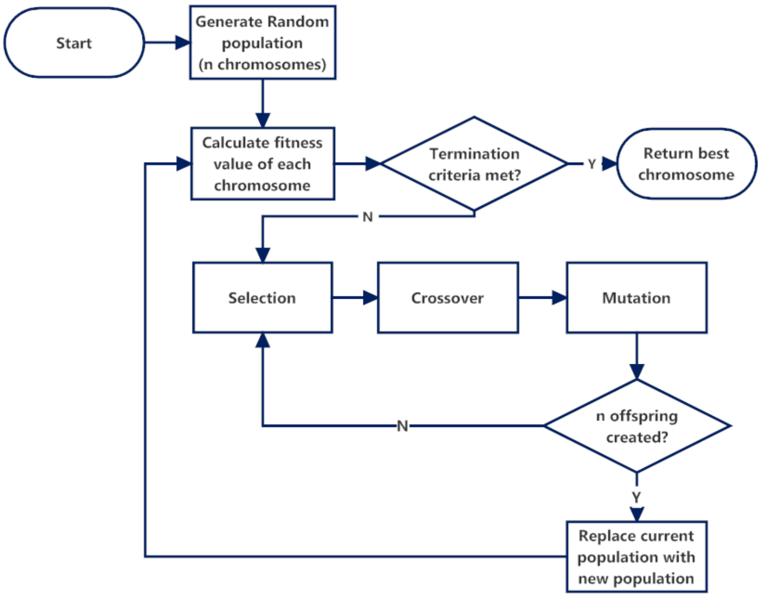

Start
Generate Random population (n chromosomes)
Calculate fitness value of each chromosome
Termination criteria met?
Y
Return best chromosome
N
Selection
Crossover
Mutation
n offspring created?
N
Y
Replace current population with new population

Figure 1. Flowchart of Simple genetic algorithm (Simple GA).

2.1.1. Fitness Evaluation

The first rule required to use GA to solve a problem is that we need to be able to clearly evaluate the fitness of a chromosome, meaning we must have a clear method of measuring the accuracy of each candidate solution. The fitness of a chromosome is judged by its accuracy in solving the given problem. The chromosome that is most accurate in solving the problem will be the fittest chromosome and the least accurate one will be the least fit chromosome.

2.1.2. Selection

The selection operator selects a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done "with replacement", meaning that the same chromosome can be selected more than once to become a parent. There are various types of selection methods, but in this research we will only use stochastic universal sampling method which will be explained in Section 4.

2.1.3. Crossover

With probability *Pc* (the "crossover probability"), we cross over the pair of parents at a randomly chosen point (chosen with uniform probability) to form two offspring. If no crossover takes place (probability of *1-Pc*), we form two offspring that are exact copies of their respective parents. The crossover operator randomly chooses one or more loci in a pair of chromosomes and exchanges the subsequences after the locus to create two offspring. There are two types of crossover operator: single-point crossover and multi-point crossover. Crossover points are selected randomly, so it might happen that during a single-point crossover the locus is located before the first bit or after the last bit of a chromosome. In such cases, the offspring pair will be an exact replica of their parents.

2.1.4. Mutation

Similar to the crossover operator, mutation is done with a probability *Pm* (the mutation probability). There are multiple ways to mutate a chromosome: single-bit mutation or n-bit mutation. For a binary vector case, the mutation operator randomly selects one or more bits in the chromosome and then flips it. Whether or not mutation takes place in the offspring created from crossover operation, the offspring will be placed in the population pool to replace the current population after *n* offspring are created.

3. SDEP-GA

This section introduces SDEP-GA, a GA with an extinction protocol driven by stagnation. The insight of this extinction protocol is to mimic the extinction phenomenon in nature to bring GA closer to how natural evolution takes place. Before going deep into the mechanism of the extinction protocol, it is useful to define what stagnation is and when it occurs.

We define stagnation as a condition when there is no improvement in the best fitness after *K* generations. Stagnation usually takes place when the algorithm is stuck in a local optimum while trying to search for a global optimum. To overcome this problem, we suggest make the chromosomes in the population extinct with a probability of *Pe* (the extinction probability). This will by chance remove some chromosomes that lead towards the stagnation from the population before the selection process takes place. This step will also increase the chances of less fit chromosomes to be selected to reproduce if they have survived the extinction step due to the decrease in the population size. The flowchart of how SDEP-GA works is shown in Figure 2.

SDEP-GA is modeled based on simple GA, with an added operator named SDEP that takes place before the selection operator. Taking into consideration that extinction in nature can either happen to only a certain species (e.g., bird flu) or any species (e.g., starvation, flood, etc.), we proposed two types of extinction protocol:

- Random extinction
- Targeted extinction

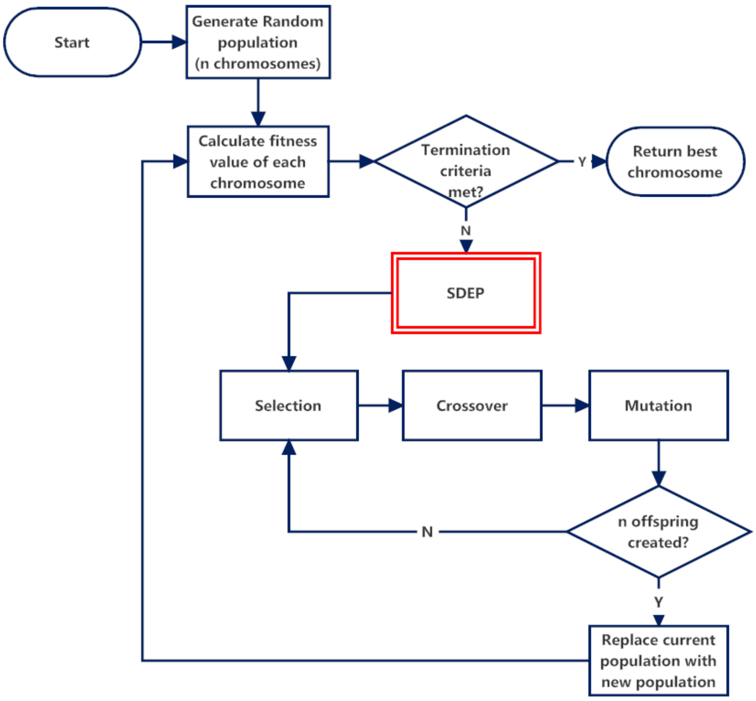

Start
Generate Random population (n chromosomes)
Calculate fitness value of each chromosome
Termination criteria met?
Y
Return best chromosome
N
SDEP
Selection
Crossover
Mutation
n offspring created?
N
Y
Replace current population with new population

Figure 2. Flowchart of SDEP-GA.

3.1. Random Extinction

In SDEP with random extinction, if the algorithm is stagnant for *K* generations, each of the chromosomes in the population will be removed (extinct) with a probability *Pe* (the extinction probability). This is done by assigning a random value from 0 to 1 to each of the chromosomes in the population pool, and if that value is greater than *Pe*, the chromosome will be removed from the population pool. The flowchart of random extinction protocol is shown in Figure 3 below.

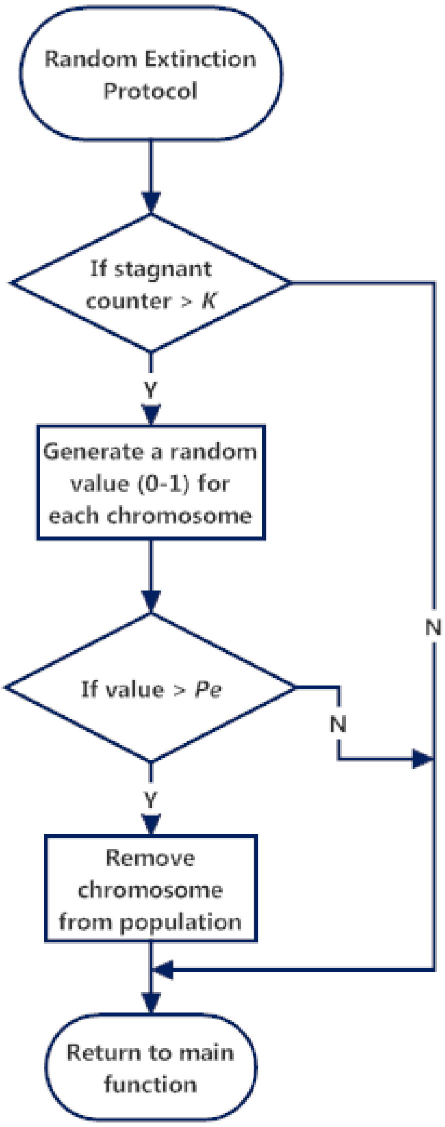

Random Extinction Protocol
If stagnant counter > K
Y
N
Generate a random value (0-1) for each chromosome
If value > Pe
Y
N
Remove chromosome from population
Return to main function

Figure 3. Flowchart of Random Extinction Protocol.

3.2. Targeted Extinction

In SDEP with targeted extinction, if the algorithm is stagnant for K generations, the chromosomes in the population will be removed (extinct) with a probability Pe (the extinction probability) only if their fitness value is below a certain threshold value T. This is done to preserve the elite chromosomes in the population. The threshold value T will be changed after every generation so that only a small percentage of chromosomes from the population pool will be exempted from the extinction process. This is done by ranking each of the chromosomes according to their fitness value and then using the fitness value of chromosome in rank i to be the threshold value. Suppose we have a population of 100 chromosomes and we want to preserve the top 10 percent of the chromosome based on their fitness value, the threshold value T will take the value of the chromosome ranked 10th according to the fitness value of the entire population. The flowchart of targeted extinction protocol is shown in Figure 4 below.

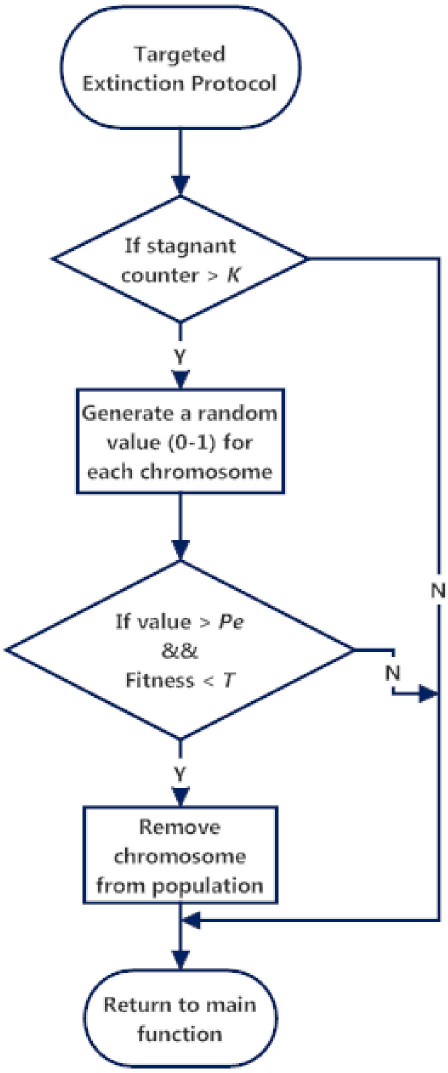

Targeted Extinction Protocol
If stagnant counter > *K*
Y
N
Generate a random value (0-1) for each chromosome
If value > *Pe* && Fitness < *T*
N
Y
Remove chromosome from population
Return to main function

Figure 4. Flowchart of Targeted Extinction Protocol.

4. Experimental Design and Result

With the aim to test the effectiveness of the SDEP-GA in solving multivariable optimization problems, we decided to use some classical test functions widely used among researchers in benchmarking the performance of GA.

4.1. Classical Test Functions

The performance of SDEP-GA is compared against Simple GA using the following classical test functions:

- Rastrigin's function [18]
- Schwefel's function [19]
- Griewank's function [20]

They are all multivariable minimization problems, where the global minimum is known and thus can be used to compare against the solution found by the algorithm.

Rastrigin has many local optima and is highly multimodal. The mathematical expression of the function is as below, together with the lower and upper bound values and the global minimum (shown in Figure 5a when $n = 2$).

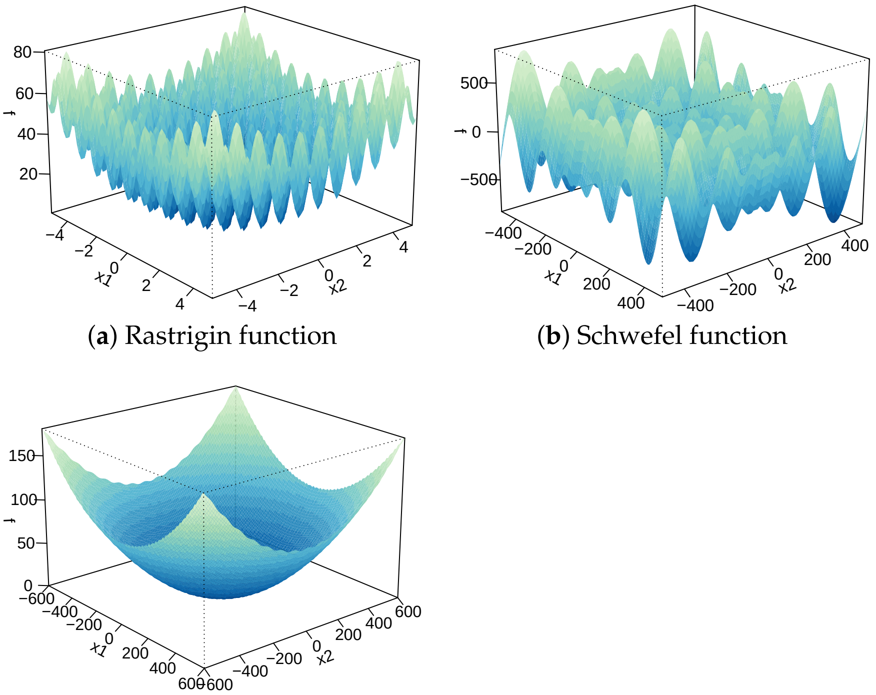
$$\text{Rastrigin}(x) = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i) \right)$$
$$-5.12 \leq x \leq 5.12$$
$$\text{global minimum} : x_i = 0; f(x) = 0$$

Schwefel is a deceptive function where search algorithms are potentially prone to converge towards the wrong direction. The mathematical expression of the function is as below, together with the lower and upper bound values and the global minimum (shown in Figure 5b when $n = 2$).

$$\text{Schwefel}(x) = \sum_{i=1}^{n} \left(-x_i \cdot \sin\left(\sqrt{|x_i|} \right) \right)$$
$$-500 \leq x \leq 500$$
$$\text{global minimum} : x_i = 420.9687; f(x) = -n \cdot 418.9829$$

Griewank is similar to Rastrigin with many widespread local minima. The mathematical expression of the function is as below, together with the lower and upper bound values and the global minimum (shown in Figure 5c when $n = 2$).

$$\text{Griewank}(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$$
$$-600 \leq x \leq 600$$
$$\text{global minimum} : x_i = 0; f(x) = 0$$



(**a**) Rastrigin function

(**b**) Schwefel function

(**c**) Griewank function

Figure 5. Classical test functions.

4.2. Experimental Design

4.2.1. Representation

As mentioned in Section 2, the first step of the experiment is to represent the candidate solution in the form of chromosomes. Rastrigin, Schwefel, and Griewank have 20 variables each. Each variable of Rastrigin will be represented by 10 bits and each variable of Schwefel and Griewank will be represented by 20 bits. Gray coding is used for all these experiments.

4.2.2. Selection

We use a stochastic universal sampling method [21] in the parent selection process where chromosomes are probabilistically selected for reproduction according to their fitness ranking in the current population. The probability of a chromosome being selected for crossover (*Ps*) is as follows:

$$P_s(x_i) = \frac{f(x_i)}{\sum_{i=1}^{n} f(x_i)}$$

where $f(x_i)$ is the fitness of chromosome x_i and $P_s(x_i)$ is the probability of that individual being selected.

4.2.3. Crossover

A single-point crossover will be used with a crossover rate of $P_c = 0.7$. This is to ensure that crossover will more likely to take place so that the offspring created are not a mere replica of the parent pair. The crossover point is selected at random, and the vectors after the selected locus will be swapped.

4.2.4. Mutation

Mutation is done on each element of the chromosome with the mutation probability, P_m, as follows:

$$P_m = \frac{0.7}{Length}$$

Length refers to the length of the chromosome structure, for example, the length is 20 if the chromosome is a 20-bit binary vector. This value is selected as it implies that the probability of any one element of a chromosome being mutated is approximately 0.5 [22]. This means that with a probability of 50 percent, at least one bit of the chromosome will be mutated.

4.2.5. SDEP

For the extinction protocol to take place, in Tables 1 and 2, we set the stagnation counter at 10 (K = 10) generations, and the top 20 percent of the population sorted by their ranks in fitness value will be preserved in the targeted extinction process.

4.2.6. Evaluation Methodology

The performance of SDEP-GA is compared against Simple GA on all three classical test functions. We adopt some parameters used in [23] to evaluate the performance of these two algorithms: mean generation number (*Mgn*) and post-extinction number (*Pen*). The definitions of these two parameters are as follows:

- *Mgn*: the average number of generations for the best result to be obtained in one complete run
- *Pen*: the average number of generations needed to achieve a new best found solution in an entire run

To assist in understanding the way these parameters function, an example will be given below for each of the parameters described above. Suppose we run the experiment for 5 runs, with each run having obtained the best result at 10th, 11th, 20th, 15th, 21st generation, respectively. The *Mgn* will be the average of these five values: $Mgn = (10 + 11 + 20 + 15 + 21)/5 = 15.4$. This value shows that on average the best result is found at the 15.4th generation. Suppose we are working on an optimization problem where we are trying to find the global minimum of a function and these are the best results obtained after each generation for 7 generations: 80, 92, 90, 78, 81, 83, and 74. To calculate the value of *Pen*, we first need to see how many times the best result is updated for one run, which in this case over the duration of 7 generations. The best result is updated at the 1st, 4th, and 7th generation, for a total of 3 times. The value of *Pen* will then be $Pen = 7/3 = 2.33$. This value indicates that on average the algorithm will obtain a better result after every 2.33 generations.

Besides that, we will compare the performance of the two algorithms in terms of accuracy. The average result of 30 runs will be used, where each run will have a total number of 400,000 evaluations. Namely, we used a population size of 400 chromosomes in the population and a maximum number of generations of 1000.

4.3. Results and Discussion

The summarized experimental results of Simple GA (SGA) and SDEP-GA using random and targeted extinction is shown in Table 1.

Table 1. Summary of SDEP-GA with random and targeted extinction against SGA; 95% confidence interval is presented after ± sign, and standard deviation is presented inside parenthesis. Minimum results are in bold font.

Test Function	Algorithm	Avg. Best Fitness	Pen	Mgn
Rastrigin	SDEP-GA			
	Random	7.87 ± 1.54 (4.14)	**0.03 ± 0.00** (0.01)	**711.60 ± 83.01** (222.31)
	Targeted	**2.75 ± 0.40** (1.06)	0.04 ± 0.00 (0.00)	887.37 ± 42.85 (114.75)
	SGA	2.97 ± 0.48 (1.29)	0.05 ± 0.00 (0.01)	891.37 ± 46.59 (124.76)
Schwefel	SDEP-GA			
	Random	−4773.83 ± 178.69 (478.55)	**0.16 ± 0.01** (0.03)	**702.16 ± 80.32** (215.11)
	Targeted	**−4780.64 ± 183.33** (490.96)	**0.16 ± 0.01** (0.03)	824.47 ± 59.61 (159.63)
	SGA	−4740.42 ± 191.99 (514.16)	0.16 ± 0.02 (0.04)	835.87 ± 47.22 (126.45)
Griewank	SDEP-GA			
	Random	0.03 ± 0.01 (0.02)	**0.07 ± 0.00** (0.01)	**687.93 ± 67.50** (180.78)
	Targeted	**0.02 ± 0.02** (0.04)	0.08 ± 0.00 (0.01)	785.50 ± 49.51 (132.58)
	SGA	0.04 ± 0.02 (0.04)	0.08 ± 0.01 (0.01)	854.17 ± 72.29 (193.59)

With reference to the third column of Table 1, we can see that SDEP with targeted extinction has a lower fitness value when compared to SGA for Rastrigin, Schwefel, and Griewank. As these three test functions are all minimization problems, the lower fitness value indicates that the algorithm scores better in terms of accuracy. In terms of *Pen*, we can see that SDEP-GA has a lower or equal value when compared against SGA. This indicates that the algorithm is able to find a better result within a shorter number of generations. In other words, SDEP with targeted extinction evolves towards the right direction at a faster rate compared to SGA. The lower value of SDEP-GA in *Mgn* shows that it is able to find a close approximation of the ideal solution faster than SGA.

Figure 6 shows the fitness values of the three algorithms over the generations for Rastrigin, which supports our argument. In Figure 6, note that we reversed the fitness for convenience in understanding the graph. It can be seen that SDEP with targeted extinction reaches near-optimum earlier than SGA and achieves near-optimum points multiple times. These results suggest that SDEP with targeted extinction is better in terms of accuracy as well as the computation time to reach the solution when compared against SGA.

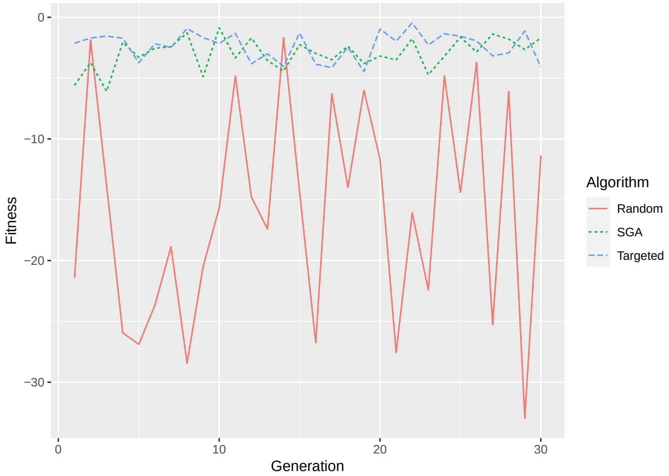


Figure 6. Fitness values over the generations for Rastrigin. Fitness values are inverted for convenience (i.e., higher values are more accurate).

The overall running time of SDEP-GA is a slightly longer than that of SGA because of the extinction overhead. Table 2 shows the running time in milliseconds for Rastrigin, Schwefel, and Griewank functions, respectively. However, for complicated problems like Griewank, there is not much difference in running time among the algorithms, because the additional overhead for random and targeted extinction is relatively small.

Table 2. Overall running time (in milliseconds) of SGA, and SDEP-GA with random and targeted extinction; 95% confidence interval is presented after ± sign.

Test Function	Algorithm	Running Time (in Milliseconds)
Rastrigin	SGA	433.13 ± 5.23
	Random Extinction	628.23 ± 16.79
	Targeted Extinction	913.33 ± 82.83
Schwefel	SGA	455.36 ± 5.25
	Random Extinction	719.36 ± 66.26
	Targeted Extinction	832.13 ± 69.66
Griewank	SGA	830.70 ± 104.79
	Random Extinction	870.00 ± 104.87
	Targeted Extinction	996.8 ± 53.86

Tables 3–5 show the fitness results of SDEP-GA Random Extinction with various parameters on Rastrigin, Schwefel, and Griewank functions, respectively. Note that the optimal value is in bold.

Table 3. Fitness results of SDEP with random extinction on Rastrigin (K is the stagnant counter); 95% confidence interval is presented after ± sign, and standard deviation is presented inside parenthesis. Minimum results are in bold font.

	Extinction Probability (P_e)				
	0.1	**0.3**	**0.5**	**0.7**	**0.9**
K 10	6.82 ± 1.50 (4.01)	11.91 ± 1.83 (4.90)	16.04 ± 2.09 (5.58)	20.60 ± 2.51 (6.74)	33.39 ± 3.16 (8.46)
100	3.31 ± 0.69 (1.86)	4.15 ± 0.67 (1.80)	6.82 ± 1.65 (4.43)	8.32 ± 1.55 (4.15)	19.93 ± 2.82 (7.56)
200	2.92 ± 0.50 (1.34)	3.68 ± 0.72 (1.94)	4.46 ± 1.38 (3.70)	9.17 ± 2.24 (5.99)	14.31 ± 2.74 (7.34)
300	**2.82** ± **0.44** (1.18)	3.72 ± 0.55 (1.48)	3.97 ± 0.72 (1.92)	4.75 ± 0.91 (2.44)	5.98 ± 0.90 (2.40)
400	3.49 ± 0.58 (1.55)	3.16 ± 0.57 (1.53)	3.48 ± 0.61 (1.65)	4.11 ± 0.73 (1.95)	4.22 ± 0.63 (1.69)
500	2.91 ± 0.45 (1.21)	3.35 ± 1.02 (2.73)	5.35 ± 1.64 (4.39)	6.94 ± 2.01 (5.39)	15.99 ± 3.33 (8.92)

Table 4. Fitness results of SDEP with random extinction on Schwefel (K is the stagnant counter); 95% confidence interval is presented after ± sign, and standard deviation is presented inside parenthesis. Minimum results are in bold font.

	Extinction Probability (P_e)				
	0.1	**0.3**	**0.5**	**0.7**	**0.9**
K 10	−4779.91 ± 178.69 (478.55)	−4830.64 ± 192.21 (514.75)	−4739.18 ± 187.37 (501.78)	−4870.65 ± 193.91 (519.30)	−**4903.66** ± **155.85** (417.37)
100	−4812.89 ± 171.71 (459.86)	−4745.69 ± 218.27 (584.53)	−4801.74 ± 193.10 (517.14)	−4793.12 ± 186.45 (499.33)	−4820.04 ± 171.53 (459.37)
200	−4878.61 ± 191.80 (513.66)	−4772.05 ± 180.67 (483.85)	−4669.98 ± 151.52 (405.77)	−4790.46 ± 252.18 (675.35)	−4685.11 ± 195.38 (523.23)
300	−4639.88 ± 216.00 (578.46)	−4821.38 ± 205.53 (550.41)	−4814.87 ± 197.89 (529.96)	−4810.89 ± 156.37 (418.77)	−4740.43 ± 191.03 (511.59)
400	−4849.50 ± 253.70 (679.43)	−4704.33 ± 179.80 (481.50)	−4629.31 ± 203.17 (544.11)	−4788.53 ± 250.73 (671.47)	−4766.24 ± 185.33 (496.31)
500	−4822.09 ± 174.10 (466.26)	−4637.11 ± 179.41 (480.48)	−4803.43 ± 203.50 (544.99)	−4779.93 ± 180.51 (483.41)	−4713.39 ± 217.27 (581.86)

Table 5. Fitness results of SDEP with random extinction on Griewank (K is the stagnant counter); 95% confidence interval is presented after ± sign, and standard deviation is presented inside parenthesis.

	Extinction Probability (P_e)				
	0.1	**0.3**	**0.5**	**0.7**	**0.9**
K 10	0.03 ± 0.01 (0.02)	0.04 ± 0.02 (0.05)	0.03 ± 0.01 (0.02)	0.03 ± 0.01 (0.04)	0.03 ± 0.02 (0.07)
100	0.02 ± 0.01 (0.02)	0.05 ± 0.02 (0.06)	0.02 ± 0.01 (0.02)	0.03 ± 0.01 (0.03)	0.03 ± 0.01 (0.04)
200	0.02 ± 0.01 (0.03)	0.02 ± 0.01 (0.03)	0.02 ± 0.01 (0.03)	0.03 ± 0.02 (0.05)	0.03 ± 0.01 (0.03)
300	0.03 ± 0.02 (0.04)	0.04 ± 0.02 (0.05)	0.03 ± 0.01 (0.03)	0.02 ± 0.01 (0.03)	0.03 ± 0.01 (0.04)
400	0.03 ± 0.01 (0.04)	0.03 ± 0.01 (0.03)	0.03 ± 0.01 (0.03)	0.03 ± 0.01 (0.03)	0.04 ± 0.02 (0.05)
500	0.02 ± 0.01 (0.04)	0.02 ± 0.01 (0.03)	0.03 ± 0.01 (0.04)	0.03 ± 0.01 (0.03)	0.03 ± 0.01 (0.04)

Taking a look at the performance of SDEP using random extinction in Tables 3–5 compared against SGA in Table 1, we can see that, for various values of two parameters (K and P_e), unfortunately, neither algorithm clearly outperforms the other.

However, from Tables 6–8, and the summary table, Table 1, it can be seen that SDEP using targeted extinction generally outperforms SGA. The reason for the poor performance of random extinction when compared to the targeted extinction might be due to the removal of certain crucial chromosomes during the random extinction process. This will set back the overall evolution rate of the algorithm because the efforts of previous evolutions are being wasted.

Table 6. Fitness results of SDEP with targeted extinction on Rastrigin (K is the stagnant counter and fitness threshold $T = 0.5$); 95% confidence interval is presented after ± sign, and standard deviation is presented inside parenthesis. Minimum results are in bold font.

			Extinction Probability (P_e)		
	0.1	**0.3**	**0.5**	**0.7**	**0.9**
K 10	3.04 ± 0.47 (1.25)	2.78 ± 0.34 (0.90)	2.57 ± 0.57 (1.51)	2.87 ± 0.53 (1.42)	3.20 ± 0.39 (1.05)
100	3.04 ± 0.50 (1.33)	2.48 ± 0.53 (1.41)	3.02 ± 0.52 (1.38)	2.99 ± 0.47 (1.26)	2.63 ± 0.42 (1.11)
200	2.95 ± 0.60 (1.61)	2.60 ± 0.45 (1.20)	2.79 ± 0.57 (1.53)	3.06 ± 0.49 (1.31)	2.64 ± 0.51 (1.36)
300	2.41 ± 0.37 (0.97)	2.61 ± 0.44 (1.16)	2.78 ± 0.55 (1.46)	2.95 ± 0.52 (1.38)	3.11 ± 0.52 (1.38)
400	**2.38 ± 0.49** (1.29)	2.95 ± 0.56 (1.49)	2.53 ± 0.41 (1.10)	2.75 ± 0.49 (1.29)	2.46 ± 0.42 (1.11)
500	2.43 ± 0.52 (1.38)	2.50 ± 0.44 (1.17)	2.43 ± 0.33 (0.88)	3.20 ± 0.53 (1.40)	2.43 ± 0.42 (1.10)

Table 7. Fitness results of SDEP with targeted extinction on Schwefel (K is the stagnant counter and fitness threshold $T = 0.5$); 95% confidence interval is presented after ± sign, and standard deviation is presented inside parenthesis. Minimum results are in bold font.

			Extinction Probability (P_e)		
	0.1	**0.3**	**0.5**	**0.7**	**0.9**
K 10	−4645.65 ± 196.66 (526.66)	−4612.80 ± 245.48 (657.41)	−4820.76 ± 215.67 (577.58)	−4839.10 ± 199.85 (535.20)	−4849.53 ± 233.43 (625.12)
100	−4888.51 ± 188.37 (504.47)	−4727.33 ± 241.21 (645.97)	−4857.66 ± 178.47 (477.94)	−4716.80 ± 207.79 (556.46)	−4842.42 ± 248.00 (664.15)
200	−4669.41 ± 189.33 (507.03)	−4778.05 ± 186.31 (498.95)	−4887.34 ± 164.99 (441.85)	−4779.97 ± 189.52 (507.53)	−4750.35 ± 197.91 (530.02)
300	−4676.04 ± 179.96 (481.94)	−4727.22 ± 207.94 (556.85)	−4798.06 ± 228.48 (611.89)	−4821.99 ± 151.12 (404.71)	−4908.28 ± 189.64 (507.86)
400	−4699.78 ± 201.81 (540.46)	−4780.60 ± 212.11 (568.02)	−4723.42 ± 214.60 (574.69)	−4675.31 ± 251.11 (672.48)	−4816.82 ± 194.22 (520.12)
500	−4712.69 ± 189.71 (508.05)	**−4939.95 ± 171.60** (459.55)	−4873.40 ± 192.05 (514.32)	−4853.73 ± 213.47 (571.69)	−4746.26 ± 204.25 (546.99)

Table 8. Fitness results of SDEP with targeted extinction on Griewank (K is the stagnant counter and fitness threshold $T = 0.5$); 95% confidence interval is presented after ± sign, and standard deviation is presented inside parenthesis.

	Extinction Probability (P_e)				
	0.1	**0.3**	**0.5**	**0.7**	**0.9**
K 10	0.02 ± 0.01 (0.03)	0.04 ± 0.04 (0.09)	0.02 ± 0.01 (0.03)	0.02 ± 0.02 (0.04)	0.02 ± 0.01 (0.03)
100	0.03 ± 0.01 (0.03)	0.03 ± 0.02 (0.05)	0.02 ± 0.01 (0.03)	0.03 ± 0.02 (0.04)	0.03 ± 0.02 (0.03)
200	0.03 ± 0.02 (0.05)	0.02 ± 0.02 (0.03)	0.02 ± 0.01 (0.03)	0.03 ± 0.02 (0.05)	0.02 ± 0.01 (0.03)
300	0.03 ± 0.02 (0.05)	0.03 ± 0.02 (0.04)	0.02 ± 0.01 (0.03)	0.02 ± 0.02 (0.05)	0.02 ± 0.01 (0.03)
400	0.02 ± 0.01 (0.03)	0.03 ± 0.02 (0.05)	0.02 ± 0.02 (0.04)	0.03 ± 0.02 (0.03)	0.03 ± 0.02 (0.05)
500	0.03 ± 0.02 (0.04)	0.02 ± 0.01 (0.03)	0.02 ± 0.01 (0.03)	0.02 ± 0.01 (0.03)	0.04 ± 0.04 (0.09)

To investigate further the statistical significance in comparing the algorithms, we perform a Friedman test [24] on our experimental results. Figure 7 shows a box and whisker plot with Friedman test results. To alleviate multiple comparison errors, p-values are adjusted using the Bonferroni method [25]. Note that a p-value < 0.05 means that the experimental result in Table 1 is statistically significant.

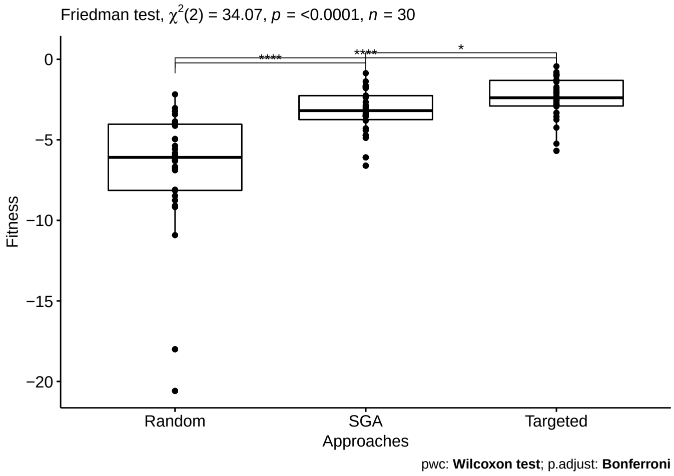


Figure 7. Box and whisker plot of Rastrigin results with Friedman test results. Note that a p value < 0.05 means that the experimental result is statistically significant.

We also perform pairwise Wilcoxon signed-rank tests [26] to identify which pairs were significant. Table 9 shows the pairwise Wilcoxon signed-rank test of the Rastrigin results. It can be seen that SDEP with targeted extinction outperforms SGA, while SGA outperforms SDEP with random extinction.

Table 9. Pairwise Wilcoxon signed-rank test of Rastrigin fitness performance among SGA, SDEP with random extinction, and SDEP with targeted extinction. The results in bold are statistically significant with a 95% confidence level. p-values are adjusted using Bonferroni method.

	SGA	SDEP with Random Extinction
SDEP with targeted extinction	**0.03**	**0.0000000783**
SDEP with random extinction	**0.00000207**	

5. Conclusions

In this paper, we presented the results of an investigation aimed to explore a new GA operator borrowed from nature, i.e., the stagnation-driven extinction protocol with random extinction and targeted extinction. We defined SDEP-GA based on those two operators, and tested their performance against SGA on three classical test functions for GAs benchmarking purpose. The achieved result suggests that SDEP-GA using targeted extinction is comparable or sometimes advantageous over SGA in terms of accuracy and also computation time. However, the same cannot be said for SDEP-GA using random extinction. These results encourage us to further investigate forms of GAs closer to natural evolution, especially in the aspect of extinction process.

The pros and cons of our proposed algorithms are as follows:

- Random extinction randomly removes chromosomes, which expedites more exploration for the population; however, it is prone to degrading the performance due to removal of near-optimal solutions.
- Targeted extinction selectively removes chromosomes, which at once enables exploration and preserves exploitation. However, it takes longer time due to internal sorting and population management overhead.

From a theoretical point of view, we plan to investigate further the optimum value of threshold values for SDEP, i.e., the number of generations of stagnation (K) to trigger extinction protocol and the chromosomes to preserve in targeted extinction. We also plan to improve SDEP-GA by mimicking how extinction takes place in nature and how it affects the evolution of species. SDEP can also be added to other forms of GAs such as elitist GA.

From a practical point of view, we plan to measure the effectiveness and robustness of the algorithm when dealing with real-world problems.

Author Contributions: Conceptualization, G.Z.Y.; methodology, G.Z.Y.; software, G.Z.Y.; validation, G.Z.Y.; formal analysis, G.Z.Y.; investigation, G.Z.Y.; resources, G.Z.Y.; data curation, G.Z.Y.; writing—original draft preparation, G.Z.Y.; writing—review and editing, D.-K.K.; visualization, G.Z.Y.; supervision, D.-K.K.; project administration, D.-K.K.; funding acquisition, D.-K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1A02050166).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors wish to thank members of the Dongseo University Machine Learning/Deep Learning Research Lab. and anonymous referees for their helpful comments on earlier drafts of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bremermann, H.J. Optimization through evolution and recombination. In *Proceedings of the Conference on Self-Organizing Systems—1962*; Yovits, M.C., Jacobi, G.T., Golstine, G.D., Eds.; Spartan Books: Washington, DC, USA, 1962; pp. 93–106.
2. Friedberg, R.M. A learning machine: Part I. *IBM J. Res. Dev.* **1958**, *2*, 2–13. [CrossRef]
3. Friedberg, R.M.; Dunham, B.; North, J.H. A learning machine: Part II. *IBM J. Res. Dev.* **1959**, *3*, 282–287. [CrossRef]

4. Darwin, C. *On the Origin of Species by Means of Natural Selection*; Murray: London, UK, 1859.
5. Weismann, A. *The Germ Plasm: A Theory of Heredity*; W. Scott: London, UK, 1893.
6. Mendel, G. Versuche über Pflanzen-Hybriden. *Verh. Nat. Ver. Brünn* **1866**, *42*, 3–47.
7. Keeton, W.T. *Biological Science*; W. W. Norton & Company Ltd.: London UK, 1996.
8. Raup, D.M. The role of extinction in evolution. *Proc. Natl. Acad. Sci. USA* **1994**, *91*, 6758–6763. [CrossRef] [PubMed]
9. Falco, I.D.; Cioppa, A.D.; Tarantino, E. Mutation-based genetic algorithm: Performance evaluation. *Appl. Soft Comput.* **2002**, *1*, 285–299. [CrossRef]
10. Falco, I.D.; Iazzetta, A.; Tarantino, E.; Cioppa, A.D.; Iacuelli, A. Towards a Simulation of Natural Mutation. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999), Orlando, FL, USA, 13–17 July 1999; Banzhaf, W., Daida, J.M., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M.J., Smith, R.E., Eds.; Morgan Kaufmann: Burlington, MA, USA, 1999; pp. 156–163.
11. Falco, I.D.; Iazzetta, A.; Tarantino, E.; Cioppa, D. On biologically inspired mutations: The translocation. In Proceedings of the Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference (GECCO'00), Las Vegas, NV, USA, 8–12 July 2000.
12. Jones, T. Crossover, macromutation, and population-based search. In Proceedings of the Sixth International Conference on Genetic Algorithms, Pittsburgh, PA, USA, 15–19 July 1995; pp. 73–80.
13. Keenan, M. Novel and combined generation, selection, recombination and mutation operators for evolutionary computing. In Proceedings of the Ninth International Conference on Computer Applications in Industry, San Francisco, CA, USA, 7–9 March 1996; ISCA Publisher: Winona, MN, USA, 1996.
14. Falco, I.D.; Cioppa, A.D.; Iazzetta, A.; Tarantino, E. A new mutation operator for evolutionary airfoil design. *Soft Comput.* **1999**, *3*, 44–51. [CrossRef]
15. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
16. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
17. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
18. Törn, A.A.; Zilinskas, A. *Global Optimization*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1989; Volume 350.
19. Schwefel, H.P. *Numerical Optimization of Computer Models*; John Wiley & Sons, Inc.: New York, NY, USA, 1981.
20. Griewank, A. *A Generalized Descent Method for Global Optimization*; Australian National University: Canberra, Australia, 1977.
21. Baker, J.E. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*; L. Erlbaum Associates Inc.: Hillsdale, NJ, USA, 1987; pp. 14–21.
22. Hesser, J.; Männer, R. Towards an optimal mutation probability for genetic algorithms. In *Parallel Problem Solving from Nature*; Schwefel, H.P., Männer, R., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1991; Volume 496, pp. 23–32. [CrossRef]
23. Jaworski, B.; Kuczkowski, L.; Śmierzchalski, R.; Kolendo, P. Extinction Event Concepts for the Evolutionary Algorithms. *Electr. Rev.* **2012**, *88*, 252–255.
24. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]
25. Dunn, O.J. Multiple Comparisons among Means. *J. Am. Stat. Assoc.* **1961**, *56*, 52–64. [CrossRef]
26. Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*; Kotz, S., Johnson, N.L., Eds.; Springer: New York, NY, USA, 1992; pp. 196–202. [CrossRef]