Differential Evolution with a Species-based Repair Strategy for Constrained Optimization

Chenyang Bu^{1,2}, Wenjian Luo^{1,2,*} and Tao Zhu^{1,2}

¹School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, Anhui, China ²Anhui Province Key Laboratory of Software Engineering in Computing and Communication, University of Science and

Technology of China, Hefei 230027, Anhui, China

bucy1991@mail.ustc.edu.cn, wjluo@ustc.edu.cn, zhuta@mail.ustc.edu.cn

Abstract—Evolutionary Algorithms (EAs) with gradient-based repair, which utilize the gradient information of the constraints set, have been proved to be effective. It is known that it would be time-consuming if all infeasible individuals are repaired. Therefore, so far the infeasible individuals to be repaired are randomly selected from the population and the strategy of choosing individuals to be repaired has not been studied yet. In this paper, the Species-based Repair Strategy (SRS) is proposed to select representative infeasible individuals instead of the random selection for gradient-based repair. The proposed SRS strategy has been applied to ε DEag which repairs the random selected individuals using the gradient-based repair. The new algorithm is named SRS- ε DEag. Experimental results show that SRS- ε DEag outperforms ε DEag in most benchmarks. Meanwhile, the number of repaired individuals is reduced markedly.

I. INTRODUCTION

Constrained real-parameter optimization problems are frequently encountered in scientific studies and practical engineering. Traditional methods and EAs are different types of methods, respectively behaving well for different types of problems. Traditional methods are often efficient in problems which are unimodal, strongly convex, etc [1]. EAs, however, have advantages in solving complex problems which are multimodal, discontinuous, non-differentiable and others that could not be well solved by traditional methods [1].

Recent works [2]–[5] show that EAs in combination with traditional methods perform better. In [2], a basic Genetic Algorithm (GA) with a gradient-based repair method which utilizes the gradient information of the constraints set obtains competitive results by finding feasible regions quickly. Also a similar method called the gradient-based mutation is applied in ε DEg [3] and ε DEag [4] which achieve excellent performance. In this paper, the referred repair strategy is the gradient-based mutation [4].

Although the gradient-based repair method [2]–[4] decreases meaningless search in infeasible spaces, it is timeconsuming if all infeasible individuals are repaired. Thus only a part of infeasible individuals are repaired for saving computing time under normal conditions. However, in existing works [2]–[4], the infeasible individuals to be repaired are randomly selected and the problem of how to select individuals to be repaired has not been studied yet. Therefore, in this paper, the SRS strategy is proposed to select representative infeasible

*Corresponding author.

individuals for gradient-based repair. Firstly, the population is divided into species using the clustering method proposed in [6]. Then for every species, the infeasible individuals with best objective values are repaired according to the proportion of the feasible solutions in this species. The characteristics of the proposed strategy are given as follows.

- By only repairing a small number of the individuals in every species, the situations that the neighbouring individuals are repaired simultaneously are reduced. If the individuals to be repaired are neighbouring, the post-repair ones may close to each other too, which may be redundant and limit the diversity of the postrepair individuals.
- 2) For every species, the number of the individuals to be repaired is decided in accordance with the proportion of the feasible solutions in this species. When there are enough feasible individuals in one species, no individuals is repaired in this species.
- 3) The infeasible individuals with best objective values in every species are regarded as the promising ones.

The proposed strategy has been applied to ε DEag [4], which randomly selects the individuals for gradient-based repair. The new algorithm is named SRS- ε DEag. The experimental results show that for most test functions SRS- ε DEag performs better than ε DEag. Meanwhile, the SRS- ε DEag reduces the number of repaired individuals significantly.

The rest of this paper is organized as follows. Section II is the backgrounds. The proposed method is presented in section III. Section IV details the experiments and discusses the results. The last section concludes this paper and introduces the future work.

II. BACKGROUNDS

A. Constrained Optimization

Constrained optimization is formalized as follows.

Minimize
$$f(X), X \in S$$

$$S = \{ (x_1, x_2, \dots, x_n) | l_i \le x_i \le u_i, 1 \le i \le n \}.$$

Subject to :

$$g_i(X) \le 0, i = 1, \dots, M,$$

 $h_i(X) = 0, j = M + 1, \dots, N.$

Tel.: +86 - 551 - 63602824

In most of the cases, the equality constraints are transformed into the form of inequalities [7]. That is,

$$|h_j(X) - \epsilon| \le 0, j = M + 1, \dots, N,$$

where $\epsilon = 0.0001$.

The value of f(X) is called the objective value. If X do not satisfy the constraints, it is an infeasible individual. For an infeasible individual X, the constraint violation $\phi(X)$ is used to indicate how much the constraints are violated, where [3]

$$\phi(X) = \sum_{i=1}^{M} \max\{0, g_i(X)\} + \sum_{j=M+1}^{N} |h_j(X)|.$$
 (1)

If X meets the requirements of all constraints, X is feasible and $\phi(X) = 0$.

B. Gradient-based Repair

The basic idea of the gradient-based repair [2]–[4] is to direct the infeasible individuals toward the feasible region utilizing the gradient information of the constraints set. The theoretical basis of the gradient-based repair is Taylor's formula:

$$f(X_0 + \Delta X) \approx f(X_0) + f'(X_0)\Delta X$$

$$\Rightarrow f(X_0 + \Delta X) - f(X_0) \approx f'(X_0)\Delta X.$$

Let V(X) denotes the vector of constraint functions.

$$V(X) = (g_1(X), \dots, g_M(X), h_{M+1}(X), \dots, h_N(X))^T.$$

The vector of constraint violations $\Delta V(X)$ is given as follows.

 $\Delta V(X) = (\Delta a_{1}(X) - \Delta a_{2}(X)) h_{1}(X)$

$$(\Delta g_1(X),\ldots,\Delta g_M(X),h_{M+1}(X),\ldots,h_N(X))^T$$

where $\Delta g_i(X) = \max\{0, g_i(X)\}.$

The gradient matrix $\nabla_x V$ derived from V(X) could be acquired accurately by:

$$\nabla_x V = (\nabla_x g_1(X), \dots, \nabla_x g_M(X), \nabla_x h_{M+1}(X), \dots, \nabla_x h_N(X))^T,$$

where

$$\nabla_x g_i(X) = \left(\frac{\partial g_i(X)}{\partial x_1}, \dots, \frac{\partial g_i(X)}{\partial x_n}\right), \forall i = 1, \dots, M$$
$$\nabla_x h_j(X) = \left(\frac{\partial h_j(X)}{\partial x_1}, \dots, \frac{\partial h_j(X)}{\partial x_n}\right), \forall j = M + 1, \dots, N.$$

It can also be estimated by the forward difference formulae.

$$\nabla_x V = \frac{1}{\eta} (V(X + e_1) - V(X), \dots, V(X + e_n) - V(X)),$$
⁽²⁾

where η is a small value ($\eta > 0$) and e_i is a vector of which the *i*-th element is η and other elements are 0.

A feasible individual X_{new} , whose constraint violation is zero, is expected to be acquired by repairing the infeasible individual X_{old} . Therefore, the vector of constraint violations $\Delta V(X)$, the gradient matrix $\nabla_x V$ and changes of solution vector ΔX should satisfy the following equation according to Taylor's formula.

$$\begin{split} V(X_{old} + \Delta X) - V(X_{old}) &\approx \nabla_x V * \Delta X \\ \Rightarrow &- \Delta V(X_{old}) \approx \nabla_x V * \Delta X \end{split}$$

Thus

$$\Delta X \approx -\nabla_x V^{-1} * \Delta V(X_{old}). \tag{3}$$

The inverse $\nabla_x V^{-1}$ can be approximated by the Moore-Penrose inverse or pseudoinverse $(\nabla_x V^+)$ [2], [4], [8] as $\nabla_x V$ is usually not invertible.

Consequently,

$$X_{new} = X_{old} + \Delta X.$$

The adjustment (ΔX) will be repeated until X_{new} is feasible or reaching the maximum repeated times (R_g) . Even if X_{new} is not feasible finally, it is expected to be closer to the feasible region than X_{old} . In original papers [2]–[4], the repair operation is executed with a probability.

The repair strategy proposed in [2] is called the gradientbased repair, and in [3], [4], it is named the gradient-based mutation. Interested readers are encouraged to refer [2]–[4] for details.

In this paper, the gradient-based mutation [4] is adopted, and it is also called the gradient-based repair.

C. Differential Evolution

Differential Evolution (DE) [9]–[11] is a widely used evolutionary algorithm, which has four steps: initialization, mutation, crossover and selection. Firstly, a number of individuals generated uniformly from the whole search space form the initial population. For each individual in the population (i.e. the target vector), a new individual (i.e. the trial vector) is generated through both mutation and crossover. The *i*-th target vector in the *t*-th generation of the population is denoted as $X_{i,t}$, and the corresponding trial vector is represented by $U_{i,t}$. In the selection step, $X_{i,t}$ will be replaced by $U_{i,t}$ if $U_{i,t}$ is better than $X_{i,t}$. The main two steps (i.e. mutation and crossover) are described briefly as follows.

1) Mutation: The mutant of $X_{i,t}$ obtained by the mutation operation is represented by $V_{i,t}$. In a basic DE,

$$V_{i,t} = X_{r_1,t} + F * (X_{r_2,t} - X_{r_3,t}).$$

Note that $r_1 \neq r_2 \neq r_3 \neq i$. *F* is a scaling factor. 2) Crossover: $U_{i,t}$ is generated from $V_{i,t}$ and $X_{i,t}$:

$$U_{j,i,t} = \begin{cases} V_{j,i,t} & \text{if } j \in J, \\ X_{j,i,t} & otherwise. \end{cases}$$

J is the set of cross bits and $U_{j,i,t}$ represents the j-th component of the trial vector $U_{i,t}$.

D. $\varepsilon DEag$

The ε DEag [4] is a DE with the ε -level comparison [12], the gradient-based mutation and an archive [13].

1) The ε -level comparison: The ε -level comparison is proposed in [12], which is defined to compare a pair of individuals with relaxed constraints. If the constraint violation values (calculated by Equation (1)) of the two individuals are equal or both less than a real number ε ($\varepsilon \ge 0$), the individual with the better objective value wins, otherwise the one with the smaller constraint violation wins.

The parameter ε is controlled as follows [4].

$$\varepsilon(0) = \phi(X_{\theta}),\tag{4}$$

$$\varepsilon(t) = \begin{cases} \varepsilon(0)(1 - t/T_c)^{cp} & 0 < t < T_c \\ 0, & t \ge T_c. \end{cases}$$
(5)

The X_{θ} represents the top θ -th individual. Parameter t is the iteration of the population and T_c is the control generation. The cp controls the reducing speed of ε .

The parameter cp is controlled as follows [4].

$$cp = \begin{cases} \max\{cp_{min}, \frac{-5 - \log\varepsilon(0)}{\log 0.05}\} & t \le T_{\lambda} \\ 0.3cp + 0.7cp_{min}, & T_{\lambda} < t < T_{c}, \end{cases}$$
(6)

where $T_{\lambda} = 0.95T_c$ and $cp_{min} = 3$.

2) Archive: The archive of ε DEag is used to maintain the diversity of the population. Initially M individuals are randomly generated from the whole search space and top N of them form the population ($M \gg N$). Then the archive consists of the remaining (M - N) individuals. The individuals in the archive could take part in the mutation operation.

3) Details of DE: In mutation step, $X_{r_1,t}$ and $X_{r_2,t}$ are selected randomly from the population P, while $X_{r_3,t}$ is selected from P with probability 0.05 and $P \bigcup A$ otherwise, where A is the archive. The exponential crossover [10] is adopted. To improve the stability, a new child can be produced when the child is not better than its parent.

The parameter F is controlled as follows [4]. If the iteration t satisfies $T_{\lambda} < t < T_c$, then

$$F = \begin{cases} 1 + |rand_G(0, 0.05)|, & \text{if } \mathbf{u}(0, 1) < 0.05\\ 0.3F_0 + 0.7, & otherwise. \end{cases}$$
(7)

And if $0 < t \leq T_{\lambda}$ or $t \geq T_c$, then

$$F = \begin{cases} 1 + |rand_G(0, 0.05)|, & \text{if } \mathbf{u}(0, 1) < 0.05\\ F_0, & otherwise. \end{cases}$$
(8)

 $rand_G(\mu, \sigma)$ represents a random number generator which obeys Gaussian distribution (μ is the mean value and σ is the standard deviation). And u(0, 1) is a number uniformly generated from [0,1]. F is truncated to 1.1.

E. Dividing Population into Species

The clustering method based on speciation proposed in [6] divides the population into species according to the Euclidean distance. The pseudo-code is shown in Algorithm 1.

Algorithm 1 Dividing population into species

- 1: *L_{sorted}* denotes the sorted population in descending order according to the objective values;
- : N denotes the population size of the population;
- 3: r_s denotes the radius;
- 4: $numspe \leftarrow 0$; // The number of the species at present.
- 5: $SpeciesSet \leftarrow \phi$; // The SpeciesSet includes all species founded so far.
- 6: *numProcessed* ← 0; // The number of processed individuals at present.

7: while numPocessed < N do

8: $P \leftarrow L_{sorted}[numProcessed];$

9: $found \leftarrow 0;$

- 10: for i = 1 to numspe do
- 11: s_i denotes the seed of the *i*-th species in the SpeciesSet;
- 12: **if** the Euclidean distance between s_i and P less than r_s **then**
- 13: $found \leftarrow 1$;
- 14: Add P to the *i*-th species in the *SpeciesSet*;

```
15: break;
```

- 16: **end if**
- 17: end for
- 18: **if** found = 0 **then**
- 19: $numspe \leftarrow numspe + 1$; // A new species is found.
- 20: Add P to the new species in SpeciesSet and P is the seed of this species.

21: end if

- 22: $numProcessed \leftarrow numProcessed + 1;$
- 23: end while

III. THE PROPOSED METHOD

The proposed SRS strategy aims to select representative infeasible individuals for gradient-based repair, which attempts to select diverse infeasible individuals that have good objective values.

The population is divided into species firstly using a clustering algorithm. Then for every species, the infeasible individuals with best objective values are repaired using the gradient-based repair. And the number of individuals to be repaired is decided by the proportion of the feasible solutions of this species.

The clustering algorithm based on speciation proposed in [6] (i.e. Algorithm 1) is adopted. The reason of choosing this clustering algorithm is that the time complexity is low [6]. The proposed SRS strategy is only performed every n generation, where n is the number of dimensions. Thus the extra calculation for the SRS strategy is relatively small. The adopted gradient-based repair is the gradient-based mutation presented in [4].

The pseudo-code of SRS- ε DEag which applies the SRS strategy to ε DEag is described in Algorithm 2 and Algorithm 3. The stop condition is that the Maximum Function Evaluations (MAX.FES) are reached.

The SRS strategy is described in Algorithm 3. The parameter ExpFeaRate is set to control the expected proportion of the feasible solutions in the population and

$$ExpFeaRate = InitFeaRate + P_f$$

where InitFeaRate is the estimated proportion of the feasible solutions in the initial population and P_f is a incremental

Algorithm 2 SRS-*\varepsilon* DEag

1:	A denotes the set of the M individuals randomly selected from
	the search space;
2:	NFea is the number of the feasible individuals in A ;
3:	$InitFeaRate \leftarrow NFea/M;$
4:	$ExpFeaRate \leftarrow InitFeaRate + P_f;$
5:	if $ExpFeaRate > 1.0$ then
6:	$ExpFeaRate \leftarrow 1.0;$
7:	end if
8:	P denotes the best N individuals of A according to the $\varepsilon\text{-level}$
	comparison; // P is the population.
9:	$A \leftarrow A - P; // A$ is the archive.
10:	Initialize ε according to Equation (4);
11:	$gen \leftarrow 1$; <i>// gen</i> is the iteration of the population.
12:	while the stop condition is not satisfied do
13:	Set F using Equation (7), (8) and F is truncated to 1.1;
14:	for $i = 1$ to N do
15:	for $k = 1$ to 2 do
16:	Randomly choose X_{r_1} and X_{r_2} from P;
17:	if a random number generated from [0,1] is smaller than
	0.05 then
18:	Randomly select X_{r_3} from P;
19:	else
20:	Randomly choose X_{r_3} from $P \cup A$;
21:	end if
22:	$X_{mutation} \leftarrow X_{r_1} + F * (X_{r_2} - X_{r_3});$
23:	The trial vector X_{child} is generated by performing expo-
	nential crossover on $X_{mutation}$ and X_i ;
24:	if X_{child} is better than X_i according to the ε -level
	comparison then
25:	$X_i \leftarrow X_{child};$
26:	break;
27:	else
28:	replace a random individual in A with X_{child} ;
29:	end if
30:	end for
31:	end for
32:	Set the radius r_s according to Equation (11);
33:	if $gen\%n=0$ then
34:	Conduct the SRS strategy (see Algorithm 3);
35:	end if
36:	Control the ε -level using Equation (5), (6);
37:	$gen \leftarrow gen + 1;$
38:	end while

parameter. If a species has enough feasible individuals, whose proportion of the feasible solutions is over ExpFeaRate, no individuals in this species will be repaired. The value of |numall * ExpFeaRate + 0.5| is the expected number of the feasible individuals in the current species, where |x| is the largest integer which less than x (i.e. |x + 0.5| is the Rounding value of x) and *numall* is the individual number of this species. The parameter *numRepair* is the number of the infeasible individuals needed to be repaired in a species, which is the difference between the expected number of feasible individuals (|numall * ExpFeaRate + 0.5|) and the number of existing feasible individuals (numfea) in this species. As ε DEag has an archive, there are some extra replacement strategies in Algorithm 3 given as follows, where the repaired individual New is acquired by repairing the infeasible individual X_j (i.e. Sinfea[j] in Algorithm 3).

- 1) If New is better than X_j , X_j will be replaced with New.
- 2) Otherwise if the number to be repaired *numRepair*

Algorithm 3 The SRS strategy of SRS- ε DEag

1:	Divide the population P into species using Algorithm 1;
2:	for every species do
3:	numall denotes the individual number in this species;
4:	numfea denotes the number of the feasible individuals in the
	current species;
5:	$numRepair \leftarrow \lfloor numall * ExpFeaRate + 0.5 \rfloor - numfea;$
6:	if $numRepair \ge 1$ then
7:	Sinfea denotes the set of the infeasible individuals in this
	species;
8:	numinfea denotes the number of individuals in Sinfea;
9:	Sort Sinfea according to the objective values in descending
	order;
10:	for $j = 0$ to $(numRepair - 1)$ do
11:	$New \leftarrow Sinfea[j];$
12:	Repairing New using the Gradient-based Repair;
13:	if New is better than $Sinfea[j]$ according to the ε -level
	comparison then
14:	Replace a random individual in A with $Sinfea[j]$;
15:	$Sinfea[j] \leftarrow New;$
16:	else
17:	if $numinfea > numRepair$ then
18:	m is a random integer generated uniformly from
	numRepair to ($numinfea$ -1);
19:	Replace a random individual in A with $Sinfea[m]$;
20:	$Sinfea[m] \leftarrow New;$
21:	else
22:	Replace a random individual in A with New ;
23:	end if
24:	end if
25:	end for
26:	end if
27:	end for

is smaller than the number of the infeasible individuals numinfea in current species, New will replace a random infeasible individual that will not be repaired in this species (i.e. Sinfea[m] in Algorithm 3).

3) The replaced individuals in the population always replace the individuals in the archive randomly.

The setup of the parameter r_s (the radius of the species) appeared in Algorithm 1 is associated with the expected number of the species (i.e. ExpSpeNum) and the volume of the search space (i.e. V). Then the scope that each species should cover is

$$V' = \frac{V}{ExpSpeNum}.$$
(9)

And the relationship between r_s and V' could be given as follows according to the volume formula of an *n*-ball [14].

$$V' = \begin{cases} \frac{\pi^{n/2} * r_s^n}{(n/2)!} & \text{if } n \text{ is even,} \\ \frac{\pi^{n/2} * r_s^n}{\sqrt{\pi * \frac{n!!}{2(n+1)/2}}} & \text{if } n \text{ is odd.} \end{cases}$$
(10)

where n is the number of dimensions and n!! is the double factorial of the n. Therefore,

$$r_{s} = \begin{cases} [\frac{(n/2)!*V}{ExpSpeNum*\pi^{\frac{n}{2}}}]^{\frac{1}{n}}, & \text{if } n \text{ is even,} \\ [\frac{n!!*V}{ExpSpeNum*2^{\frac{n+1}{2}}*\pi^{\frac{n-1}{2}}}]^{\frac{1}{n}}, & \text{if } n \text{ is odd.} \end{cases}$$
(11)

As r_s is only need to be calculated once in a run, the calculation for r_s is small.

IV. EXPERIMENTS

A. Benchmarks and Experimental Setup

The benchmarks proposed in CEC2010 [7] are tested, which have 18 problems, denoted as C01-C18. The benchmarks have two cases with the dimensions (n) of 10 and 30, which named 10D and 30D problems.

The proposed algorithm SRS- ε DEag is compared with the ε DEag proposed in [4]. The results of the ε DEag are from the original paper [4].

For each function, 50 runs are performed in this paper. The Maximum Function Evaluations (MAX.FES) are 2×10^5 for 10D problems and 6×10^5 for 30D problems, respectively. The evaluations of the objective function and the evaluations of the constraints violation are treated separately [4]. The stop condition is that either of the evaluations for the objective function and the evaluations for the constraints violation reaches the MAX.FES, which is the same setup as ε DEag. Note that the gradient matrix $\nabla_x V$ is estimated by equation (2), so (n + 1)function evaluations are costed for obtaining the $\nabla_x V$ [4].

We take advantage of the original source codes of ε DEag [4] given by Takahama and Sakai, and our codes may be obtained upon request.

B. Parameter Settings

In order to compare with ε DEag, the settings of the parameters in this paper are similar to those of ε DEag. All parameters are set as follows.

1) Parameters for DE: The population size (N) = 4n. The archive size (M) = 100n, where n is the number of dimensions. Therefore, n is 10 for 10D problems and 30 for 30D problems. The initial scaling factor $F_0 = 0.5$ and the crossover rate $CR_0 = 0.9$.

2) Parameters for the ε -level comparison: $T_c = 1000, \theta = 0.9$.

3) Parameters for the SRS strategy: $P_f = 0.1$. The maximum repeated times of each repair $(R_q) = 3$.

4) Setup of the radius r_s : The search spaces of these benchmarks are hypercubes. So for every test function the volume of the search space can be calculated by

$$V = (upper - lower)^n,$$

where *upper* and *lower* are respectively the upper bound and the lower bound of the problem. As n are even for all the problems, according to Equation (11), the radius r_s in this paper can be calculated as follows.

$$r_s = \pi^{-\frac{1}{2}} * \left[\frac{(n/2)!}{ExpSpeNum}\right]^{\frac{1}{n}} * (upper - lower).$$
(12)

In this paper, the parameter ExpSpeNum is set to 20 for 10D problems and 10 for 30D problems.

C. Experimental Results and Discussion

The comparisons between ε DEag and the proposed SRS- ε DEag are presented in Tables I-VI. The performance measurements are similar to those in [7]. For each function, best, median, worst, mean objective values and standard deviation (std) for 25 runs are reported. The number in the parentheses after the best, median, worst results means the number of the violated constraints. Index c is the number of the violated constraints of the median solutions: the three sequential numbers represent the number of violations which are more than 1.0, 0.01 and 0.0001, respectively. Index \bar{v} is the mean violations of all constraints at the median solution. A feasible run means the best individual of that run is feasible. Feasible Rate indicates the rate of feasible runs over total runs. Eval/Grad Rate respectively shows the ratio of the number of actual function evaluations for objective functions, and the ratio of the number of function evaluations for the gradient matrix over the maximum number of function evaluations.

Tables I-VI show that, for most test functions, better results are achieved by SRS- ε DEag, especially for the 30D problems. The tables also show that the function evaluations of the gradient-based repair are dramatically reduced for all problems. For example, the original *Grad Rate* of C06 for the 30D case is 22.66% while the new value is just 0.0911%.

The reasons of SRS- ε DEag performs better in most functions are analyzed as follows.

Firstly, the number of repaired individuals is decreased significantly. As lot of computational effort is required for repairing an infeasible individual, the number of the individuals to be repaired should be limited and the main goal of the repair is to provide good genetic material [2]. By dividing the population into species and for every species only small number of the individuals are repaired, SRS- ε DEag reduces the redundant repair significantly. This is proved by the significant reduction of the *Grad Rate* for all the problems while the performance for most functions does not degrade.

Secondly, the diversity of the post-repair individuals is encouraged by selecting diverse individuals to be repaired through clustering, which can reduce the probability of falling into a local feasible region and have an advantage in problems with multiple disconnected feasible regions.

Thirdly, the strategy of selecting infeasible individuals with the best objective values in every species to be repaired may also facilitate acquiring better results.

However, there may be limitations owing to just selecting the infeasible individuals with best objective values to be repaired. The setup of the parameter r_s also has an effect on the results. Additionally, the adopted clustering method may have its own limitations.

V. CONCLUSIONS AND FUTURE WORK

The gradient-based repair which utilizes the gradient information of the constraints set has been proved to be effective. In previous work, the repair operation is performed randomly and the problem of how to select individuals to be repaired has not been studied. In this paper, the SRS strategy is proposed to select the representative infeasible individuals instead of the

C01 C02						C02		
	Measures		01		02	0	03	
		ε DEag	SRS- ε DEag	ε DEag	SRS- ε DEag	ε DEag	SRS- ε DEag	
	Best	-7.473104e-01(0)	-7.473104e-01(0)	-2.277702e+00(0)	-2.277711e+00(0)	0.000000e+00(0)	0.000000e+00(0)	
	Median	-7.473104e-01(0)	-7.473104e-01(0)	-2.269502e+00(0)	-2.277711e+00(0)	0.000000e+00(0)	0.000000e+00(0)	
	Worst	-7.405572e-01(0)	-7.292159e-01(0)	-2.174499e+00(0)	-2.239347e+00(0)	0.000000e+00(0)	0.000000e+00(0)	
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	
10D	\bar{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
	Mean	-7.470402e-01	-7.466783e-01	-2.258870e+00	-2.270310e+00	0.000000e+00	0.000000e+00	
	std	1.323339e-03	2.823639e-03	2.389779e-02	1.283211e-02	0.000000e+00	0.000000e+00	
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0	
	Eval/Grad Rate(%)	50.43/2.52	52.07/ 0.00	13.79/16.45	18.54/ 0.3484	97.66/2.08	98.99/ 0.0796	
	Best	-8.218255e-01(0)	-8.217821e-01(0)	-2.169248e+00(0)	-2.212233e+00(0)	2.867347e+01(0)	2.867347e+01(0)	
	Median	-8.206172e-01(0)	-8.206789e-01(0)	-2.152145e+00(0)	-2.185708e+00(0)	2.867347e+01(0)	2.867347e+01(0)	
	Worst	-8.195466e-01(0)	-8.198002e-01(0)	-2.117096e+00(0)	-2.172221e+00(0)	3.278014e+01(0)	2.867347e+01(0)	
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	
30D	\bar{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
	Mean	-8.208687e-01	-8.208098e-01	-2.151424e+00	-2.186866e+00	2.883785e+01	2.867347e+01(0)	
	std	7.103893e-04	6.092129e-04	1.197582e-02	8.884082e-03	8.047159e-01	2.803525e-07	
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0	
	Eval/Grad Rate(%)	53.70/2.26	55.10/ 0.0000	14.17/16.51	14.21/ 0.0801	51.40/5.95	62.59/ 0.0314	

TABLE I. FUNCTION VALUES OF PROBLEMS C1-C3 WITH MAX.FES OF 2×10^5 FOR THE 10D CASE AND 6×10^5 FOR THE 30D CASE

TABLE II. FUNCTION VALUES OF PROBLEMS C4-C6 WITH MAX.FES OF 2×10^5 FOR THE 10D CASE AND 6×10^5 FOR THE 30D CASE

	Mansuras	C	04	C	05	C	06
	wiedsures	ε DEag	SRS- <i>ɛ</i> DEag	ε DEag	SRS- ε DEag	ε DEag	SRS- <i>ɛ</i> DEag
	Best	-9.992345e-06(0)	-1.000000e-05(0)	-4.836106e+02(0)	-4.836106e+02(0)	-5.786581e+02(0)	-5.786624e+02(0)
	Median	-9.977276e-06(0)	-9.999942e-06(0)	-4.836106e+02(0)	-4.836106e+02(0)	-5.786533e+02(0)	-5.786624e+02(0)
	Worst	-9.282295e-06(0)	-9.999374e-06(0)	-4.836106e+02(0)	-4.836106e+02(0)	-5.786448e+02(0)	-5.786583e+02(0)
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
10D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
	Mean	-9.918452e-06	-9.999881e-06	-4.836106e+02	-4.836106e+02	-5.786528e+02	-5.786619e+02
	std	1.546730e-07	1.601177e-10	3.890350e-13	6.252776e-13	3.627169e-03	9.068002e-04
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0
	Eval/Grad Rate(%)	36.08/17.22	44.91/ 0.5243	59.37/10.46	64.92/ 0.4593	22.33/20.86	25.82/ 0.4344
	Best	4.698111e-03(0)	2.511241e-03(0)	-4.531307e+02(0)	-4.710838e+02(0)	-5.285750e+02(0)	-5.292583e+02(0)
	Median	6.947614e-03(0)	5.294157e-03(0)	-4.500404e+02(0)	-4.636899e+02(0)	-5.280407e+02(0)	-5.289143e+02(0)
	Worst	1.777889e-02(0)	1.149340e-02(0)	-4.421590e+02(0)	-4.575179e+02(0)	-5.264539e+02(0)	-5.282776e+02(0)
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
30D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
	Mean	8.162973e-03	5.701988e-03	-4.495460e+02	-4.637938e+02	-5.279068e+02	-5.288217e+02
	std	3.067785e-03	1.840437e-03	2.899105e+00	3.376524e+00	4.748378e-01	2.542424e-01
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0
	Eval/Grad Rate(%)	20.75/21.91	20.74/ 0.1490	8.68/22.37	9.24/0.0863	14.25/22.66	14.94/0.0911

TABLE III. FUNCTION VALUES OF PROBLEMS C7-C9 WITH MAX.FES OF 2×10^5 FOR THE 10D CASE AND 6×10^5 FOR THE 30D CASE

	Measures	C	07	C08		C	209
	wiedsures	ε DEag	SRS- ε DEag	ε DEag	SRS- ε DEag	ε DEag	SRS- ε DEag
	Best	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)
	Median	0.000000e+00(0)	0.000000e+00(0)	1.094154e+01(0)	1.094154e+01(0)	0.000000e+00(0)	0.000000e+00(0)
	Worst	0.000000e+00(0)	0.000000e+00(0)	1.537535e+01(0)	1.094154e+01(0)	0.000000e+00(0)	1.420783e+02(0)
	С	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
10D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
	Mean	0.000000e+00	0.000000e+00	6.727528e+00	8.390586e+00	0.000000e+00	2.929730e+00
	std	0.000000e+00	0.000000e+00	5.560648e+00	4.489414e+00	0.000000e+00	1.988795e+01
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0
	Eval/Grad Rate(%)	99.79/0.03	99.82/ 0.0009	71.86/1.88	64.85/ 0.1792	97.70/2.03	97.59/ 0.0688
	Best	1.147112e-15(0)	5.547705e-16(0)	2.518693e-14(0)	7.489190e-15(0)	2.770665e-16(0)	5.864695e-21(0)
	Median	2.114429e-15(0)	2.114899e-15(0)	6.511508e-14(0)	4.210234e-14(0)	1.124608e-08(0)	1.051212e-18(0)
	Worst	5.481915e-15(0)	8.463931e-15(0)	2.578112e-13(0)	1.414990e-13(0)	1.052759e+02(0)	8.096873e+01(0)
	С	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
30D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
	Mean	2.603632e-15	2.695924e-15	7.831464e-14	4.903190e-14	1.072140e+01	2.431359e+00
	std	1.233430e-15	1.612093e-15	4.855177e-14	3.092311e-14	2.821923e+01	1.199192e+01
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0
	Eval/Grad Rate(%)	99.68/0.08	99.76/ 0.0002	93.40/0.70	94.09/ 0.0009	50.86/5.02	78.38/ 0.0259

TABLE IV. FUNCTION VALUES OF PROBLEMS C10-C12 WITH MAX.FES OF 2×10^5 FOR THE 10D CASE AND 6×10^5 FOR THE 30D CASE

	Measures	(210	C	211	C	212	
	wieasures	ε DEag	SRS- ε DEag	ε DEag	SRS- ε DEag	ε DEag	SRS- ε DEag	
	Best	0.000000e+00(0)	0.000000e+00(0)	-1.522713e-03(0)	-1.522713e-03(0)	-5.700899e+02(0)	-5.700899e+02(0)	
	Median	0.000000e+00(0)	0.000000e+00(0)	-1.522713e-03(0)	-1.522713e-03(0)	-4.231332e+02(0)	-5.683828e+02(0)	
	Worst	0.000000e+00(0)	2.352812e+02(0)	-1.522713e-03(0)	-1.522713e-03(0)	-1.989129e-01(0)	-1.772472e-01(0)	
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	
10D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
	Mean	0.000000e+00	4.801868e+00	-1.522713e-03	-1.522713e-03	-3.367349e+02	-4.703457e+02	
	std	0.000000e+00	3.293251e+01	6.341035e-11	6.017895e-11	1.782166e+02	1.422306e+02	
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0	
	Eval/Grad Rate(%)	97.68/2.04	96.82/ 0.0838	30.82/6.66	33.25/ 0.2075	21.71/14.97	24.82/ 0.5844	
	Best	3.252002e+01(0)	3.203914e+01(0)	-3.268462e-04(0)	-3.684443e-04(0)	-1.991453e-01(0)	-1.991645e-01(0)	
	Median	3.328903e+01(0)	3.270017e+01(0)	-2.843296e-04(0)	-2.964730e-04(0)	5.337125e+02(1)	1.330206e+02(1)	
	Worst	3.463243e+01(0)	3.389747e+01(0)	-2.236338e-04(0)	-1.920063e-04(0)	5.461723e+02(1)	7.037591e+02(1)	
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,1,0	0,1,0	
30D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	3.240709e-01	3.277013e-01	
	Mean	3.326175e+01	3.287527e+01	-2.863882e-04	-2.992116e-04	3.562330e+02	2.133080e+02	
	std	4.545577e-01	4.747073e-01	2.707605e-05	3.324991e-05	2.889253e+02	2.709754e+02	
	Feasible Rate(%)	100.0	100.0	100.0	100.0	12.0%	6.0%	
	Eval/Grad Rate(%)	28.53/5.71	28.47/ 0.0264	12.33/0.1276	13.22/ 0.0892	13.16/11.11	12.46/ 0.1296	

TABLE V. FUNCTION VALUES OF PROBLEMS C13-C15 WITH MAX.FES OF 2×10^5 FOR THE 10D CASE AND 6×10^5 FOR THE 30D CASE

	Measures	C	213	C14		C15	
		$\varepsilon DEag$	SRS- ε DEag	ε DEag	SRS- <i>ɛ</i> DEag	ε DEag	SRS- ε DEag
	Best	-6.842937e+01(0)	-6.842937e+01(0)	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)
	Median	-6.842936e+01(0)	-6.842936e+01(0)	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)	0.000000e+00(0)
	Worst	-6.842936e+01(0)	-6.351751e+01(0)	0.000000e+00(0)	0.000000e+00(0)	4.497445e+00(0)	6.736439e+02(0)
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
10D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
	Mean	-6.842936e+01	-6.803657e+01	0.000000e+00	0.000000e+00	1.798978e-01	2.209658e+01
	std	1.025960e-06	1.332025e+00	0.000000e+00	0.000000e+00	8.813156e-01	1.050063e+02
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0
	Eval/Grad Rate(%)	43.86/6.25	47.85/ 0.0908	98.90/0.94	99.69/ 0.0208	96.20/2.25	96.25/ 0.0737
	Best	-6.642473e+01(0)	-6.750817e+01(0)	5.015863e-14(0)	1.208848e-14(0)	2.160345e+01(0)	2.160353e+01(0)
	Median	-6.531507e+01(0)	-6.589543e+01(0)	1.359306e-13(0)	7.742921e-14(0)	2.160375e+01(0)	2.160368e+01(0)
	Worst	-6.429690e+01(0)	-6.451542e+01(0)	2.923513e-12(0)	5.123535e-13(0)	2.160403e+01(0)	2.160383e+01(0)
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
30D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
	Mean	-6.535310e+01	-6.587739e+01	3.089407e-13	1.034953e-13	2.160376e+01	2.160368e+01
	std	5.733005e-01	6.081125e-01	5.608409e-13	8.243519e-14	1.104834e-04	6.241290e-05
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0
	Eval/Grad Rate(%)	30.94/7.42	33.08/ 0.0260	98.68/1.15	99.75/ 0.0037	65.61/4.63	67.82/ 0.0095

TABLE VI. FUNCTION VALUES OF PROBLEMS C16-C18 WITH MAX.FES OF 2×10^5 FOR THE 10D CASE AND 6×10^5 FOR THE 30D CASE

	Maggurag	C	216	C	C17		C18	
	wiedsuies	ε DEag	SRS- <i>ɛ</i> DEag	ε DEag	SRS- ε DEag	ε DEag	SRS- <i>ɛ</i> DEag	
	Best	0.000000e+00(0)	0.000000e+00(0)	1.463180e-17(0)	5.546678e-32(0)	3.731439e-20(0)	1.523574e-20(0)	
	Median	2.819841e-01(0)	1.625153e-02(0)	5.653326e-03(0)	1.824505e-11(0)	4.097909e-19(0)	2.187970e-19(0)	
	Worst	1.018265e+00(0)	1.043905e-01(0)	7.301765e-01(0)	6.438400e-01(0)	9.227027e-18(0)	1.292241e-15(0)	
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	
10D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
	Mean	3.702054e-01	2.337096e-02	1.249561e-01	4.152941e-02	9.678765e-19	5.789320e-17	
	std	3.710479e-01	2.641648e-02	1.937197e-01	1.243568e-01	1.811234e-18	2.236009e-16	
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0	
	Eval/Grad Rate(%)	32.09/21.64	41.98/ 0.2157	35.19/16.84	47.02/ 0.4114	91.02/4.64	95.38/ 0.1687	
	Best	0.000000e+00(0)	0.000000e+00(0)	2.165719e-01(0)	8.546849e-03(0)	1.226054e+00(0)	1.241343e-01(0)	
	Median	0.000000e+00(0)	0.000000e+00(0)	5.315949e+00(0)	1.169026e-01(0)	2.679497e+01(0)	9.631557e+00(0)	
	Worst	5.421011e-20(0)	0.000000e+00(0)	1.889064e+01(0)	3.712959e+00(0)	7.375363e+02(0)	2.402905e+02(0)	
	с	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	
30D	\overline{v}	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
	Mean	2.168404e-21	0.000000e+00	6.326487e+00	4.829979e-01	8.754569e+01	3.946860e+01	
	std	1.062297e-20	0.000000e+00	4.986691e+00	7.767602e-01	1.664753e+02	6.238992e+01	
	Feasible Rate(%)	100.0	100.0	100.0	100.0	100.0	100.0	
	Eval/Grad Rate(%)	64.44/7.22	67.79/ 0.0221	35.55/16.92	37.96/ 0.1157	11.23/14.35	12.00/ 0.0390	

random selection for gradient-based repair. The population is divided into species firstly by the clustering method proposed in [6]. Then for every species, a small number of the infeasible individuals are repaired in accordance with the proportion of feasible solutions in this species. In order to examine the effectiveness, the strategy has been applied to ε DEag which repairs the random selected individuals using the gradientbased repair. The new algorithm is named SRS- ε DEag. Results show that SRS- ε DEag outperforms ε DEag in most test functions. Meanwhile the number of the function evaluations for the gradient-based repair is reduced significantly.

In the future, it is interesting to explore the effectiveness of the SRS in combination with other state-of-the-art EAs, such as the SAMO-DE [15] and the SAMO-GA [15], etc. And the idea of reducing computing time by repairing representative individuals could also be used to hill-climbing strategies. Through only applying hill-climbing strategies to representative individuals, the calculation cost is expected to be reduced and the quality of final results could not degrade. Moreover, other kinds of infeasible individuals, such as the ones with least constraint violation, can be considered in the selection of the individuals to be repaired. Other clustering methods should be tested. And the self-adaptive setup of the parameter r_s needs to be studied.

ACKNOWLEDGMENT

This work is partly supported by the 2006-2007 Excellent Young and Middle-aged Academic Leader Development Program of Anhui Province Research Experiment Bases.

REFERENCES

- [1] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. IOP Publishing Ltd., 1997.
- [2] P. Chootinan and A. Chen, "Constraint handling in genetic algorithms using a gradient-based repair method," *Computers & Operations Research*, vol. 33, no. 8, pp. 2263–2281, 2006.
- [3] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites," in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation(CEC2006)*, pp. 308–315.
- [4] —, "Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation," in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation(CEC2010)*, pp. 1680–1688.
- [5] F. Neri, C. Cotta, and P. Moscato, *Handbook of Memetic Algorithms*. Springer, 2011, vol. 379.
- [6] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation.* ACM, 2005, pp. 873–880.
- [7] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization," *Nanyang Technological University, Singapore*, 2010.
- [8] S. L. Campbell and C. D. Meyer, Generalized Inverses of Linear Transformations. SIAM, 2009, vol. 56.
- [9] 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.
- [10] K. V. Price, R. M. Storn, and J. A. Lampinen, "Differential evolution a practical approach to global optimization," *Berlin, Germany:Springer-Verlag*, 2005.
- [11] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

- [12] T. Takahama and S. Sakai, "Constrained optimization by ε constrained particle swarm optimizer with ε -level control," in *Soft Computing as Transdisciplinary Science and Technology*. Springer, 2005, pp. 1019–1029.
- [13] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [14] X. Wang, "Volumes of generalized unit balls," *Mathematics Magazine*, vol. 78, no. 5, pp. 390–395, 2005.
- [15] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Multi-operator based evolutionary algorithms for solving constrained optimization problems," *Computers & Operations Research*, vol. 38, no. 12, pp. 1877–1896, 2011.