Effect of Spatial Locality on an Evolutionary Algorithm for Multimodal Optimization

Ka-Chun Wong, Kwong-Sak Leung, and Man-Hon Wong

Department of Computer Science & Engineering
The Chinese University of Hong Kong, HKSAR, China
{kcwong,ksleung,mhwong}@cse.cuhk.edu.hk

Abstract. To explore the effect of spatial locality, crowding differential evolution is incorporated with spatial locality for multimodal optimization. Instead of random trial vector generations, it takes advantages of spatial locality to generate fitter trial vectors. Experiments were conducted to compare the proposed algorithm (CrowdingDE-L) with the state-of-the-art algorithms. Further experiments were also conducted on a real world problem. The experimental results indicate that CrowdingDE-L has a competitive edge over the other algorithms tested.

1 Introduction

Real world problems always have different solutions. Unfortunately, most traditional optimization techniques focus on solving for a single optimal solution. They needs to be applied several times; yet all solutions are not guaranteed to be found. Thus multimodal optimization problem was proposed. In this problem, we are interested in not only a single optimal point, but also the others. Given an objective function, an algorithm is expected to find all optimal points in a single run. With strongly parallel search capability, evolutionary algorithms are shown to be particularly effective in solving this type of problems [7].

2 Background

The work by De Jong [3] is one of the first known attempts to solve multimodal optimization problems by an evolutionary algorithm. He introduced a technique called "crowding": An offspring replaces the most similar individual. As a result, it can maintain different types of individuals in a single run to increase the chance for locating multiple optima. Since then, researchers have proposed different genetic algorithms for multimodal optimization problems. In particular, crowding [3], fitness sharing [8], and speciation [9,19] techniques are the most popular techniques. They have also been integrated in differential evolution [17] and demonstrated promising results [18,10].

Differential Evolution was proposed by Storn and Price [17]. Without loss of generality, a typical strategy of differential evolution (DE/rand/1) [6] is briefly described as follows: For each individual $indiv_i$ in a generation, the algorithm randomly selects three individuals to form a trial vector. One individual forms a base vector, whereas the value difference between the other two individuals forms a difference vector. The sum

C. Di Chio et al. (Eds.): EvoApplications 2010, Part I, LNCS 6024, pp. 481–490, 2010.

of these two vector forms a trial vector, which recombines with $indiv_i$ to form an offspring. In a comparison to crossover and mutation operations, it can provide differential evolution a self-organizing ability and high adaptability for choosing suitable step sizes which demonstrated its potential for continuous optimization in the past contests [2].

2.1 CrowdingDE

To extend the capability of differential evolution, Thomsen [18] incorporates the crowding technique [3] into differential evolution (CrowdingDE) for multimodal optimization. Although an intense computation is accompanied, it can effectively transform differential evolution into an algorithm specialized for multimodal optimization. To determine the dissimilarity (or distance) between two individuals, the dissimilarity measurement proposed in Goldberg et al. [8] and Li et al. [9] is adopted. The dissimilarity between two individuals is based on their Euclidean distance. The smaller the distance, the more similar they are and vice versa.

2.2 Locality of Reference

Locality of Reference [15] (or **The Locality Principle** [5]), is one of the most fundamental principles widely used in computing. The principle was originated from memory management methods in order to predict which memory entries would be referenced soon. The main idea is to make use of neighborhood relationships for prediction, optimizing the throughput. To define the neighborhood relationship, time and space are typically taken as the proximity measures. If time is taken, it is called **temporal locality**. If space is taken, it is called **spatial locality**.

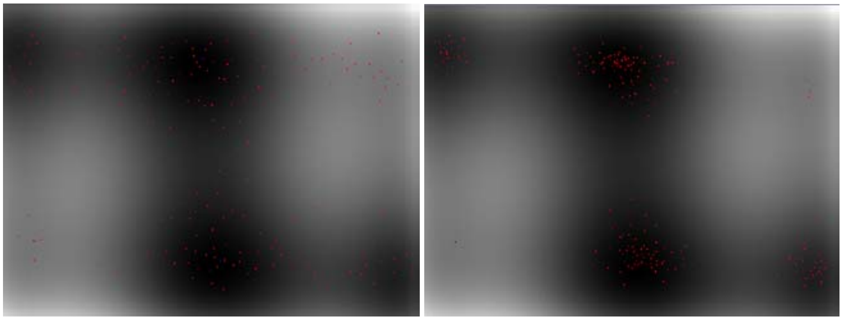
3 Proposed Method

3.1 Motivation

If we do not apply any specific technique to maintain diversity, most evolutionary algorithms will prematurely converge and get stuck in a local optimum. To cope with the problem, the algorithms for multimodal optimization are usually equipped with their own local operations for diversity maintenance. In CorwdingDE, its local operation is the crowding technique. Thinking this technique more deeply, it is to propose a restriction on the individual replacement policy such that an individual gets replaced only when a fitter offspring is generated within the same niche. Thus the choice of the offspring generation method becomes a critical performance factor. Unfortunately, the offspring generations in CrowdingDE mainly relies on random trial vector generations. They are too random to offer enough feasible replacement schemes for all individuals. Thus we propose a new method for trial vector generation, in order to increase the chances for successful replacements.

3.2 CrowdingDE-L

Close individuals tend to have similar characteristics. For instance, two population snapshots of CrowdingDE are shown on Fig. 1. For each snapshot, the population can



(a) 5000 function evaluations (b) 20000 function evaluations

Fig. 1. Population snapshots of CrowdingDE [18] on F5 with population size = 200

be seen to be divided into different niches. Within each niche, the individuals exhibit similar positions and step-sizes for improvement. After several generations, the difference between niches may be even larger. It will be a disaster if a single evolutionary strategy is applied to all of them regardless of their niches. Luckily, it is a two-edged sword. Such property also gives us spatial locality: crossovers between close individuals can have higher chances to generate better trial vectors.

Thus a local operation is proposed to take advantage of it: the individuals which are closer to a parent should be given more chances to be involved in the parent's trial vector generation. In other words, to bring such neighborhood idea into the trial vector generation, the distances between the parent and its candidate individuals are first computed. Then the distances are transformed into the proportions of a roulette-wheel [4]. Within the roulette-wheel, larger proportions of the roulette-wheel are given to closer candidate individuals. It follows that closer individuals are given higher chances for trial vector generations. For the sake of clarity, the local operation is outlined in Algorithm 1.

Combined with the local operation, Crowding Differential Evolution (CrowdingDE) is reformulated as a hybrid algorithm which takes advantages of spatial locality. Thus it is named **Crowding D**ifferential **E**volution using Spatial **L**ocality (**CrowdingDE-L**). A trial vector generation can be tailor-made for each individual. Fitter trial vectors are more likely to be generated. More feasible replacement schemes can thus be provided.

Mathematically, a function is needed to transform distances to proportions of a roulette-wheel. Thus two transformation functions are proposed in this paper: Since closer individuals are given higher values (proportions), the transformation function must be a monotonically decreasing function over the interval [0,MaxDistance], where MaxDistance is the maximum distance between a parent and all of its candidate individuals. Thus a simple function and Gaussian function are proposed for the transformation. The simple function is based on the formula: $Proportion = (\frac{MaxDistance - distance}{MaxDistance})^a$ where a is a scaling constant. On the other hand, the Gaussian function is based on the formula: $Proportion = exp(-(\frac{distance^2}{2 \times SD^2}))$ where $SD = \frac{MaxDistance}{3}$. Since spatial locality is normal in nature [5], the Gaussian function is adopted in CrowdingDE-L for the transformation if not specified explicitly.

Algorithm 1. Trial Vector Generation Using Spatial Locality

P: Parent individual
$trial$: Trial vector
D_G: Genome dimension
CR: Crossover probability
$I[i]$: The gene value of individual I at dimension i
$rand()$: A random number in the interval $[0,1]$

procedure NEWTRIALVECTORGENERATION(P)

1. Transform the distances between P and all candidate individuals to proportions using a transformation function;
2. Prepare a roulette-wheel based on the transformed proportions;
3. Use the roulette-wheel to pick 3 different individuals I_1, I_2, I_3 where $P \neq I_1 \neq I_2 \neq I_3$;

```
trial = P;
i ← ⌊rand() × D_G⌋;
for ( k = 0; k < D_G; k = k+1 ) do
    if rand() < CR then
        trial[i] = I_1[i] + F × (I_2[i] − I_3[i]);
        i = (i+1) mod D_G;
    end if
end for
return trial;
end procedure
```

4 Experiments

We implemented all the algorithms using Sun's Java programming language. The development was based on the EC4 framework [4]. Experiments to compare the performance among CrowdingDE-L and other algorithms were conducted on ten benchmark functions. The other algorithms include: Crowding Genetic Algorithm (CrowdingGA) [3], CrowdingDE [18], Fitness Sharing Genetic Algorithm (SharingGA) [8], SharingDE [18], Species Conserving Genetic Algorithm (SCGA) [9], and SDE [10]. The first five benchmark functions are widely adopted in literatures: F1 is Deb's 1st function [19], F2 is Himmelblau function [1], F3 is Six-hump Camel Back function [12], F4 is Branin function [12] and F5 is Rosenbrock function [16]. The remaining five benchmark functions were derived from [19,11].

4.1 Performance Measurements

For multimodal optimization, there are several performance metrics proposed [11,10,9,18]. The focuses of this paper are on the ability of the algorithms to locate the optima and the accuracy of the optima found by the algorithms. Hence we adopted the Peak Ratio (PR) and Average Minimum Distance to the Real Optima (D) [11,19] as the performance metrics.

As different algorithms perform different operations in one generation, it is unfair to set the termination condition as the number of generations. Alternatively, it is also unfair to adopt CPU time, since it substantially depends on the implementation techniques for different algorithms. For instance, the sorting techniques and the programming languages used. In contrast, fitness function evaluation is always the performance bottleneck. Thus the number of fitness function evaluations was set as the termination condition in the following experiments. All algorithms were run up to a maximum of 40000 fitness function evaluations. The above performance metrics were obtained by taking the average and standard deviation of 50 runs.

4.2 Parameter Setting

Sun's Java Double (double-precision 64-bit IEEE 754 floating point) was used as the real number representation for all algorithms. All populations were initialized randomly. The random seed was 12345. For all DE algorithms, the crossover probability (CR) was 0.9 and F was 0.5. The common GA parameter settings of CrowdingGA, SharingGA and SCGA for all benchmarks were the same as Table 5 in [19]. For all crowding algorithms, population size was set to 100 for Peaks2, Peaks3 and Peaks4. 50 was set for the remaining benchmark functions. The parameter settings of SharingDE, SharingGA, SCGA and SDE for different benchmarks are tabulated in Table 1. For SharingDE and SharingGA, σ and α denote the niche radius and scaling factor respectively. The parameters have been tuned in a few preliminary runs with manual inspections for all algorithms.

Table 1. Parameter setting of SharingDE, SharingGA, SCGA and SDE for different benchmarks

		SharingDE [18]		SharingGA [8]		SCGA [9] and SDE [10]
Benchmark	Population Size	α	σ	α	σ	Species Distance*
F1	100	1	0.001	1	0.001	0.01
F2	100	1	0.03	5	0.1	3
F3	100	1	0.01	2	40	0.5
F4	100	1	0.01	1	0.1	6
F5	100	3	30	3	30	10
Peaks1	200	1	100	1	50	50
Peaks2	200	1	100	2	50	25
Peaks3	200	1	5	1	0.5	3
Peaks4	200	1	5	1	0.5	3
Peaks5	200	1	300	1	300	150

4.3 Results

Table 2 shows the experimental results. Mann-Whitney U tests (two-tailed) have been conducted to reject the null hypothesis that the performance metrics of CrowdingDE-L and those of another algorithm are sampled from the same population. For D in all

Species Distance $= \sigma_s/2$ in [9].

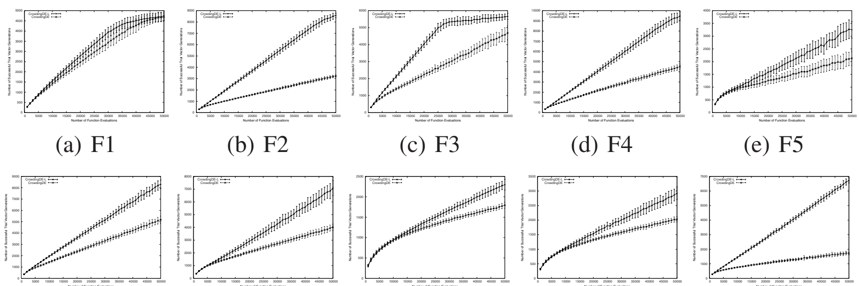
Table 2. Experimental Results for all algorithms tested (averaged over 50 runs)

Benchmark	Measurement	CrowdingDE-L	CrowdingGA [3]	CrowdingDE [18]	SharingGA [8]	SharingDE [18]	SDE [10]	SCGA [9]
	Mean of D	**2.81E-10**	2.24E-06	3.72E-10	4.08E-03	1.14E-03	1.59E-03	1.09E-03
	StDev of D	**8.00E-11**	4.81E-06	2.03E-10	1.21E-02	4.53E-04	7.87E-03	1.16E-03
	Minimum of D	**8.44E-11**	4.35E-09	1.38E-10	2.34E-05	4.66E-04	8.43E-07	6.06E-05
F1	Median of D	**3.31E-11**	2.24E-06	2.82E-10	1.28E-04	1.33E-03	4.15E-06	1.13E-03
	Mean of Peak Ratio	**1.00**	**1.00**	**1.00**	0.98	**1.00**	0.99	**1.00**
	StDev of Peak Ratio	**0.00**	**0.00**	**0.00**	0.06	**0.00**	0.04	**0.00**
	Mean of D	**2.20E-07**	4.93E-04	3.86E-05	2.06E+00	4.92E-01	1.20E+00	2.59E-01
	StDev of D	**1.53E-06**	5.49E-04	1.51E-05	1.05E+00	7.77E-01	6.36E-01	1.17E-01
	Minimum of D	**2.63E-10**	1.30E-05	1.11E-05	7.39E-03	2.63E-02	2.60E-03	2.34E-02
F2	Median of D	**2.68E-09**	3.03E-04	5.49E-05	7.42E-01	5.01E-02	1.23E+00	1.63E-01
	Mean of Peak Ratio	**1.00**	**1.00**	**1.00**	0.66	0.91	0.78	0.44
	StDev of Peak Ratio	**0.00**	**0.00**	**0.00**	0.17	0.14	0.11	0.18
	Mean of D	**1.66E-09**	2.21E-05	4.86E-07	1.44E-01	1.55E-02	6.22E-03	2.03E-02
	StDev of D	**4.35E-10**	3.38E-05	4.58E-07	2.89E-01	4.96E-03	2.26E-03	2.22E-02
	Minimum of D	**6.07E-10**	3.52E-08	9.41E-08	2.14E-04	4.55E-03	1.49E-03	3.54E-04
F3	Median of D	**1.76E-09**	1.49E-06	5.75E-07	5.85E-04	1.63E-02	5.25E-03	5.62E-02
	Mean of Peak Ratio	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**	0.95
	StDev of Peak Ratio	**0.00**	**0.00**	**0.00**	0.20	**0.00**	**0.00**	0.15
	Mean of D	**4.87E-07**	5.96E-02	2.45E-04	3.39E+00	1.38E+00	2.61E-01	6.63E-01
	StDev of D	**1.50E-06**	1.14E-01	1.25E-04	1.99E+00	1.85E+00	7.81E-01	5.91E-01
	Minimum of D	**1.20E-08**	3.08E-05	6.44E-05	6.09E-03	7.94E-03	3.84E-03	7.72E-02
F4	Median of D	**1.62E-07**	1.22E-01	3.74E-04	5.84E+00	1.96E+00	1.06E+00	4.22E-01
	Mean of Peak Ratio	**1.00**	0.89	**1.00**	0.61	0.88	0.97	0.41
	StDev of Peak Ratio	**0.00**	0.16	**0.00**	0.21	0.16	0.10	0.14
	Mean of D	**7.81E-03**	1.22E-02	2.28E-02	8.59E-03	4.14E-02	4.23E-02	8.76E-03
	StDev of D	**7.54E-03**	2.84E-02	2.03E-02	1.83E-02	1.40E-01	3.07E-02	1.11E-02
	Minimum of D	3.77E-04	2.31E-05	1.11E-03	1.54E-05	1.63E-09	1.29E-03	**9.38E-13**
F5	Median of D	**5.05E-03**	2.57E-02	1.10E-02	6.84E-03	3.63E-02	5.24E-02	7.55E-03
	Mean of Peak Ratio	**1.00**	0.96	**1.00**	**1.00**	0.92	0.94	**1.00**
	StDev of Peak Ratio	**0.00**	0.20	**0.00**	**0.00**	0.27	0.24	**0.00**
	Mean of D	**4.95E-07**	6.36E-01	7.79E-06	4.84E+00	2.67E+01	5.00E+01	2.11E+00
	StDev of D	**2.12E-06**	1.18E+00	5.58E-06	7.78E+00	3.12E+01	7.71E+00	9.10E-01
	Minimum of D	**1.08E-10**	1.62E-05	1.16E-06	6.88E-01	2.90E-01	3.74E+01	4.80E-01
Peaks1	Median of D	**1.77E-09**	9.40E-04	1.64E-05	1.87E+01	8.00E+00	5.48E+01	1.92E+00
	Mean of Peak Ratio	**1.00**	0.86	**1.00**	0.01	0.37	0.21	0.34
	StDev of Peak Ratio	**0.00**	0.17	**0.00**	0.05	0.10	0.16	0.05
	Mean of D	1.30E+01	7.43E+00	1.24E+01	3.66E+01	3.60E+01	6.92E+01	**5.76E+00**
	StDev of D	**8.64E-01**	3.52E+00	1.51E+00	4.55E+00	1.01E+01	8.08E+00	1.24E+00
	Minimum of D	9.65E+00	**1.49E+00**	9.29E+00	2.10E+01	1.43E+01	6.58E+01	3.29E+00
Peaks2	Median of D	1.33E+01	**4.68E+00**	1.14E+01	3.59E+01	3.38E+01	6.60E+01	6.11E+00
	Mean of Peak Ratio	**0.70**	0.36	0.66	0.00	0.21	0.19	0.11
	StDev of Peak Ratio	0.03	0.07	0.08	**0.02**	0.04	0.04	**0.02**
	Mean of D	**5.75E-03**	9.15E-02	1.44E-02	1.59E+00	2.24E-01	3.69E+00	4.07E-01
	StDev of D	**1.43E-03**	8.21E-02	2.04E-03	3.66E-01	1.28E-01	6.48E-01	1.09E-01
	Minimum of D	**3.32E-03**	1.89E-02	1.05E-02	9.92E-01	7.05E-02	2.34E+00	2.57E-01
Peaks3	Median of D	**5.64E-03**	6.31E-02	1.44E-02	1.92E+00	2.44E-01	4.72E+00	3.77E-01
	Mean of Peak Ratio	**1.00**	0.84	**1.00**	0.61	0.55	0.34	0.27
	StDev of Peak Ratio	**0.00**	0.09	**0.00**	0.07	0.12	0.05	0.08
	Mean of D	**3.24E-03**	2.69E-01	1.76E-02	2.49E+00	4.90E-01	3.36E+00	5.20E-01
	StDev of D	**8.89E-04**	1.75E-01	2.87E-02	4.32E-01	1.81E-01	6.96E-01	1.19E-01
	Minimum of D	**1.89E-03**	5.56E-02	8.77E-03	1.92E+00	1.77E-01	2.47E+00	3.29E-01
Peaks4	Median of D	**3.95E-03**	1.73E-01	1.37E-02	2.90E+00	4.55E-01	3.37E+00	5.31E-01
	Mean of Peak Ratio	**1.00**	0.69	**1.00**	0.24	0.33	0.33	0.20
	StDev of Peak Ratio	**0.00**	0.08	0.01	0.09	0.12	0.04	0.06
	Mean of D	**7.80E-01**	1.29E+02	6.14E+00	1.84E+02	3.19E+01	2.81E+01	1.13E+02
	StDev of D	1.50E+00	1.58E+01	**1.14E+00**	1.54E+01	2.46E+01	2.85E+01	1.70E+01
	Minimum of D	**2.80E-03**	8.71E+01	2.37E+00	1.40E+02	1.04E+01	1.24E-02	8.39E+01
Peaks5	Median of D	**1.06E-01**	1.25E+02	6.33E+00	1.93E+02	3.41E+01	2.57E+01	1.04E+02
	Mean of Peak Ratio	**0.83**	0.00	0.00	0.00	0.00	**0.83**	0.00
	StDev of Peak Ratio	**0.14**	0.00	0.00	0.00	0.00	0.24	0.00

benchmarks, except that with SCGA in F5, their differences are found statistically significant with α-level=0.01. For PR in all benchmarks, except that with SDE in F1, F5, and Peaks5 and that with CrowdingGA in F5, their differences are found statistically significant with α-level=0.05. On the whole, CrowdigDE-L showed its competitive results with other existing algorithms.

4.4 Effect of Spatial Locality

To demonstrate the effect of spatial locality, experiments between CrowdingDE-L and CrowdingDE were conducted on all the benchmark functions. Figure 2 depicts the results.



(a) F1 (b) F2 (c) F3 (d) F4 (e) F5

(f) Peaks1 (g) Peaks2 (h) Peaks3 (i) Peaks4 (j) Peaks5

Fig. 2. Effect of Spatial Locality (averaged over 50 runs)

Each sub-figure corresponds to one benchmark function. The vertical axis is the number of successful trial vector generation (averaged over 50 runs). A successful trial vector generation is defined as the generation of an offspring, which can replace an individual in a parent population. On the other hand, the horizontal axis is the number of fitness function evaluations. It could be observed that CrowdingDE-L generate a higher number of successful trial vectors than CrowdingDE no matter how the fitness function evaluations were varied from 1000 to 50000. More chances for convergence were provided in CrowdingDE-L.

5 Real World Application

To verify the proposed algorithm in real world optimization, we have adopted an optical system design problem [14,13] as a benchmark problem.

5.1 Varied-Line-Spacing Holographic Grating Design

Holographic gratings have been widely applied in optical instruments for aberration corrections. In particular, the Varied-Line-Spacing (VLS) holographic grating distinguishes itself from others by the high order aberration eliminating capability in diffractive optical systems. It is commonly used in high resolution spectrometers and monochromaters. A recording optical system of VLS holographic grating is outlined in [14].

The objective for the design is to find several sets of design variables (or recording parameters [14]) to form the expected groove shape of G (or the distribution of groove density [13]). Mathematically, the goal is to minimize the definite integral of the square error between the expected groove density and practical groove density [14]:

$$min \;\; J = \int_{-w_0}^{w_0} (n_p - n_e)^2 dw$$

where w_0 is the half-width of the grating, n_p is the practical groove density and n_e is the expected groove density. These two groove densities are complicated functions of the design variables [14].

Theoretically, the above objective is simple and clear. Unfortunately, in practice, there are many other auxiliary optical components, which constraints are too difficult to be expressed and solved in mathematical forms. Single optimal solution is not necessarily a feasible and favorable solution. Thus optical engineers often need to tune the design variables to find as many optimal solutions as possible for multiple trials. Multimodal optimization becomes necessary for the design problem.

5.2 Performance Measurements

As the objective function is an unknown landscape, the exactly optimal information is not available. Thus the previous performance metrics cannot be simply adopted in this section. We propose two new performance metrics in this section. The first one is the best fitness, which is the fitness value of the fittest individual in the last population. The second one is the number of distinct peaks, where a distinct peak is considered found when there exists an individual which fitness value is below a threshold 0.0001 and there isn't an individual within 0.1 distance unit and found as a peak before in the last population. The threshold is chosen to 0.0001 because the fitness values of the solutions found in [14] is around this order of magnitude. On the other hand, the distance is chosen to 0.1 unit because it has already been set for considering peaks found in peak ratio [19,11].

5.3 Parameter Setting

Same as before, all algorithms were run up to a maximum of 40000 fitness function evaluations. The above performance metrics were obtained by taking the average and standard deviation of 50 runs. The groove density parameters followed the setting in [14]: $n_0 = 1.400 \times 10^3 (line/mm)$, $b_2 = 8.2453 \times 10^{-4} (1/mm)$, $b_3 = 3.0015 \times 10^{-7} (1/mm^2)$ and $b_4 = 0.0000 \times 10^{-10} (1/mm^3)$. Half-width w_0 was 90mm. The radii of spherical mirrors M_1 and M_2 were 1000mm. The recording wavelength (λ_0) was 413.1nm. The previous settings were adopted except the algorithm-specific parameters: The species distance of SDE and SCGA was set to 500. The scaling factor and niche radius of SharingDE and SharingGA were set to 1 and 1000 respectively. The population size was set to 50.

5.4 Results

The result is tabulated in Table 3. CrowdingDE-L showed slightly better results among the algorithms.

Table 3. Experimental Results for all algorithms tested on the VLS holographic grating design

Measurement	CrowdingDE-L	CrowdingGA [3]	CrowdingDE [18]	SharingGA [8]	SharingDE [18]	SDE [10]	SCGA [9]
Mean of Best Fitness	**4.47E-10**	1.94E-09	2.88E-07	5.05E-03	2.41E+02	1.38E-01	7.81E+03
StDev of Best Fitness	**3.16E-09**	1.06E-08	2.04E-06	1.67E-02	3.45E+02	2.34E-01	1.52E+04
Min of Best Fitness	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	3.18E-02	1.69E-04	9.35E-01
Median of Best Fitness	1.12E-08	**0.00E+00**	**0.00E+00**	5.33E-06	4.08E+01	7.09E-02	8.46E+03
Means of Peaks Found	**50.00**	17.08	**50.00**	2.16	0.00	0.00	0.00
StDev of Peaks Found	**0.00**	4.55	**0.00**	3.47	0.00	0.00	0.00
Min of Peaks Found	**50.00**	10.00	**50.00**	0.00	0.00	0.00	0.00
Median of Peaks Found	**50.00**	15.50	**50.00**	1.50	0.00	0.00	0.00

6 Conclusion

Confirmed by the experimental results, CrowdingDE-L is highlighted with its ability for generating fitter trial vectors, which can successfully replace parent individuals. Extensive experiments have been conducted. The results indicate that CrowdingDE-L has its own competitive edge over the other algorithms tested, in terms of the performance metrics.

The locality principle is proven simple and useful in computing [5]. In a macro-view, the work in this paper can be regarded as a case study for integrating the locality principle into an evolutionary algorithm. The numerical results can also be viewed as a valuable resource for comparing the state-of-the-art algorithms for multimodal optimization.

Acknowledgment

The authors are grateful to anonymous reviewers for their valuable comments. The authors would also like to thank Ling Qing for his source codes and insightful discussions. This research is partially supported by the grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. 414107 and 414708).

References

1. Beasley, D., Bull, D.R., Martin, R.R.: A sequential niche technique for multimodal function optimization. Evol. Comput. 1(2), 101–125 (1993)
2. Bersini, H., Dorigo, M., Langerman, S., Seront, G., Gambardella, L.: Results of the first international contest on evolutionary optimisation (1st ICEO). In: Proceedings of IEEE International Conference on Evolutionary Computation, Nagoya, Japan, May 1996, pp. 611–615 (1996)
3. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, University of Michigan, Ann Arbor (1975); University Microfilms No. 76-9381
4. De Jong, K.A.: Evolutionary Computation. A Unified Approach. MIT Press, Cambridge (2006)
5. Denning, P.J.: The locality principle. Commun. ACM 48(7), 19–24 (2005)

6. Feoktistov, V.: Differential Evolution - In Search of Solutions. Springer Optimization and Its Applications, vol. 5. Springer-Verlag New York, Inc., Secaucus (2006)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
8. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms and their application, pp. 41–49. L. Erlbaum Associates Inc., Hillsdale (1987)
9. Li, J.P., Balazs, M.E., Parks, G.T., Clarkson, P.J.: A species conserving genetic algorithm for multimodal function optimization. Evol. Comput. 10(3), 207–234 (2002)
10. Li, X.: Efficient differential evolution using speciation for multimodal function optimization. In: GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation, pp. 873–880. ACM, New York (2005)
11. Lung, R.I., Chira, C., Dumitrescu, D.: An agent-based collaborative evolutionary model for multimodal optimization. In: GECCO 2008: Proceedings of the 2008 GECCO conference companion on Genetic and evolutionary computation, pp. 1969–1976. ACM, New York (2008)
12. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs, 3rd edn. Springer, London (1996)
13. Qing, L., Gang, W., Qiuping, W.: Restricted evolution based multimodal function optimization in holographic grating design. In: The 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, September 2005, vol. 1, pp. 789–794 (2005)
14. Qing, L., Gang, W., Zaiyue, Y., Qiuping, W.: Crowding clustering genetic algorithm for multimodal function optimization. Appl. Soft Comput. 8(1), 88–95 (2008)
15. Rogers, A., Pingali, K.: Process decomposition through locality of reference. SIGPLAN Not. 24(7), 69–80 (1989)
16. Shang, Y.W., Qiu, Y.H.: A note on the extended rosenbrock function. Evol. Comput. 14(1), 119–126 (2006)
17. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997), `http://www.springerlink.com/content/x555692233083677/`
18. Thomsen, R.: Multimodal optimization using crowding-based differential evolution. In: Congress on Evolutionary Computation, CEC 2004, June 2004, vol. 2, pp. 1382–1389 (2004)
19. Wong, K.C., Leung, K.S., Wong, M.H.: An evolutionary algorithm with species-specific explosion for multimodal optimization. In: GECCO 2009: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pp. 923–930. ACM, New York (2009)