Causes of the Imbalance Between Exploration and Exploitation in Evolutionary Computation

Zhe Chen[1,2](✉) and Chengjun Li[1,2]

[1] School of Computer Science, China University of Geosciences,
Wuhan 430074, Hubei, China
`chenzhe@cug.edu.cn`, `cuglicj@126.com`
[2] Hubei Key Laboratory of Intelligent Geo-Information Processing,
China University of Geosciences, Wuhan 430074, Hubei, China

Abstract. Evolutionary algorithms have been used in more and more research fields. However, it is very usual that an optimal of nontrivial problems cannot be found by an evolutionary algorithm. In fact, only if the balance between exploration and exploitation is achieved in runs, good solutions can be obtained. In this paper, we observe the changing trend of genotype diversity in runs, which cannot obtain the optimal, of different EAs. Then, we illustrate the main cause of the imbalance between exploration and exploitation in different situations.

Keywords: Evolutionary algorithm · Exploration and exploitation · Diversity · Causes

1 Introduction

Evolutionary algorithms (EAs) including genetic algorithm, genetic programming, evolutionary programming, evolution strategy and differential evolution are stochastic search methods and have been used in many field. An EA proceeds in an iterative manner by generating a new population, $P(t+1)$, from an old one, $P(t)(t \in [0, n])$. Every individual in a population is a tentative solution of the current problem encoded in a type of chromosome representation. For indicating suitability of an individual to problem, an evaluation function associates a fitness value to it. The initial population, $P(0)$, is produced randomly, Then, in each generation, crossover, mutation, etc are used to obtain new individuals based on the original ones. These new individuals are collected into a temporary population $P'(t)$. After that, $P(t+1)$ is produced through selection by picking up some individuals from $P(t)$ and some from $P'(t)$. The halting condition of EA is usually set as reaching a preprogrammed number of generations, or obtaining a satisfactory solution [1].

In theory, the task of EAs is to find the optimal of problem. However, for nontrivial problems, it very usual that EAs cannot find an optimal in runs. In

H. Peng et al. (Eds.): ISICA 2018, CCIS 986, pp. 117–131, 2019.
https://doi.org/10.1007/978-981-13-6473-0_11

fact, exploration and exploitation require to be addressed for EAs. Exploration is the process of visiting entirely new regions in search space, while exploitation is the one of visiting the neighborhood of previously visited points [4]. They are two cornerstones in search [5]. Only if the balance between exploration and exploitation is achieved in runs, EAs can obtain good solutions.

In EAs, both exploration and exploitation are realized by operators. Moreover, population size and chromosome representation have important impacts on exploration and exploitation. Nevertheless, it is difficult to measure exploration or exploitation directly [4]. On many occasions, diversity, which can be measured at genotype level or phenotype level, is used to measure exploration and exploitation. In fact, existing methods for the balance between exploration and exploitation are classified according to their effect on diversity in [4]. Although there are methods to achieve the balance between exploration and exploitation, further studies need be done. Provided that causes of imbalance are better known, we can get more ideas for the balance. That is the motivation of this paper.

In this paper, experiments are carried out based on three EAs including a state-of-the-art EA. In our experiments, these EAs do not obtain the optimal in most cases. Thus, it can be inferred that the imbalance between exploration and exploitation occurs in runs of these EAs. Based on experimental results, the changing trend of genotype diversity in runs is shown in figures by us. By observing the trend, we illustrate causes of the imbalance between exploration and exploitation and propose a method to identify the main cause in certain runs.

The rest of this paper is organized as follows. Related works are introduced in Sect. 2. Then, experiments on different EAs are given in Sect. 3. In Sect. 4, we illustrate our finding based on experimental results. Finally, a conclusion and a prospect are dealt with in Sect. 5.

2 Related Works

As above mentioned, in the EA community, diversity is widely used to reflect exploration and exploitation. Diversity refers to differences among individuals. It can be measured at the genotype level or the phenotype one. Genotype diversity reveals differences among genomes within a population, while phenotype one just shows differences among fitness values. Compared with phenotype diversity, genotype diversity is more costly to calculate. So far, for both genotype diversity and phenotype diversity, many different measures have been proposed. Nonetheless, diversity measures are problem-specific [2,6,9]. Consequently, calculation steps of the same measure may be still distinct in EAs if these EAs are based on different chromosome representation. According to [4], most of existing approaches applied in EAs for the exploration and exploitation balance can be explained through their effect on diversity.

Although causes of the imbalance between exploration and exploitation are remained to be studied, phenomena arisen from the imbalance are widely known.

In practice, the imbalance between exploration and exploitation can be judged when no better individual can be obtained any more in a run in spite of an optimal not found. Based on diversity $d \in [0, 1]$, which refers to the differences among individuals, two phenomena can be distinguished from the situation that no better individual can be obtained any more in spite of an optimal not found. One is premature convergence, which features low diversity. The other is stagnation, which features much higher diversity. When premature convergence occurs, there is almost no difference between any two individuals in population. In other words, diversity tends to be zero. When stagnation comes, there are still differences among individuals. That is, diversity value is still high [4].

Here are examples to make these phenomena be understood well. After enough generations, individuals are fairly good in fitness. In this case, the probability of mutated individuals winning in selection may become less and less since their competitors have fairly good fitness. Provided that current diversity level has been low, it is difficult for crossover to generate offspring which are different from their parents since parents are very similar or even identical. Consequently, premature convergence comes soon. If well maintained diversity still can support crossover generating offspring different with their parents, the probability of they selected may become less and less for their competitors is good in fitness. As a result, stagnation happens soon.

3 Experiments to Show the Changing Trend of Genotype Diversity in Different EAs

In this section, three EAs based on different chromosome representations are involved in our experiments to cover the three main types of chromosome representations. In detail, they are a Genetic Algorithm (GA) for the Travelling Salesman Problem (TSP), a GA for the one-max problem and the Differential Evolution (DE) based on the covariance matrix learning and the bimodal distribution parameter setting (CoBiDE) for continuous function optimization. In the three EAs, the CoBiDE is a state-of-the-art one. For different tasks, each EA runs thirty times, respectively. We set enough generations for all algorithms. In each runs, genotype diversity is computed at every interval. Based on these data of genotype diversity, we draw figures showing the changing trend of genotype diversity during runs for each task.

3.1 Experiment on the EA for TSP

The EA for the TSP proposed by [14] and discussed in recent papers, such as [7,8,10], is employed by us. Details of the algorithm can be found in [14]. This EA are based on integer chromosome representation. It uses only one parameter, p, to control both crossover and mutation. In its crossover, a parent plays a major role and can be called the primary one. In crossover, two parents produce only one offspring. Then, in its selection, every offspring competes only with its primary parent. Thus, this algorithm is good at maintaining diversity. Datasets of TSP from TSPLIB [11], which are difficult for this EA, are used in the experiment. Settings this TSP EA are shown in Table 1.

Table 1. Settings of the EA for the TSP

Population size	100
p	0.02
Length of interval	4000 generations
Terminal criterion	400000 generations done

The method for genotype diversity computation in this EA comes from [3]. Let x_i and x_j be two individuals. Matrix M in Formula 1 is connection matrix of TSP tour. In the matrix, k is the number of cities and $a_{lm} \in \{0, 1\}, (0 \leq l \leq k-1, 0 \leq m \leq k-1)$. $a_{lm} = 1$ represents that there is a connection from city $(l+1)$ to city $(m+1)$ in tour, while $a_{lm} = 0$ denotes that such a connection does not exist. Then, a $k \times k$ connection matrix can be built for x_i and x_j, respectively. Let k' be the number of rows which are same in the two matrixes. Then, the distance between x_i and x_j, $D(x_i, x_j)$, can be defined as Formula 2. Further, diversity, PD, can be defined as Formula 3, where NP is population size and C means combination.

$$\mathbf{M} = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0(k-1)} \\ a_{10} & a_{11} & \cdots & a_{1(k-1)} \\ \cdots & \cdots & \cdots & \cdots \\ a_{(k-1)0} & \cdots & \cdots & a_{(k-1)(k-1)} \end{bmatrix} \tag{1}$$

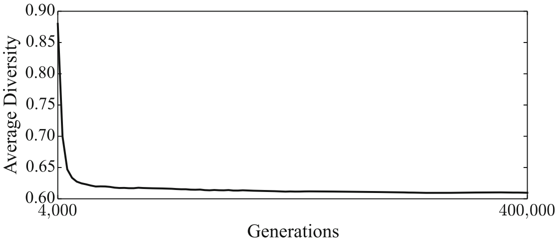
$$D(x_i, x_j) = 1 - \frac{k'}{k} \tag{2}$$

$$PD = \frac{\sum_{i=1}^{NP} \sum_{j=1}^{NP} D(x_i, x_j)}{C_{NP}^{2}} \tag{3}$$

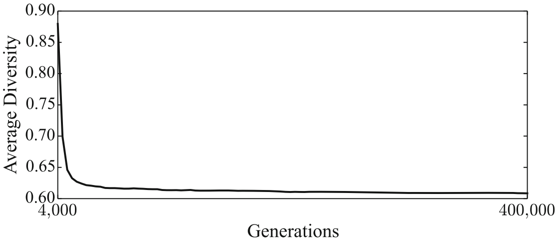
For lin318, linhp318, rd400, fl417, pr439, pcb442 and d493, the TSP EA runs thirty times, respectively. Diversity value is recorded at each interval in runs. Then, for every dataset, the thirty time average of diversity at every interval is plotted in Fig. 1 to show the changing trend of diversity in runs. It can be seen that, for each dataset, the trend in runs is very similar. In detail, after the sharp and short decrease at the initial stage, diversity stays at a low level in the remaining part of run. Such a type of trend shows runs go to stagnation soon after the initial stage.

3.2 Experiment on the EA for the One-Max Problem

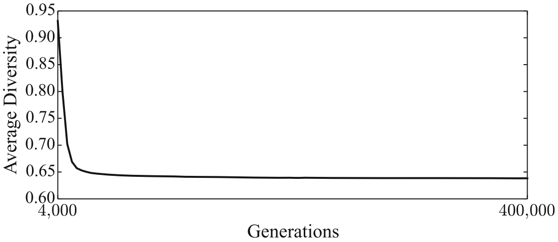
We design a EA with binary chromosome representation for this problem which employ the uniform crossover [13], the bit string mutation and rank-based selecting model. Settings of this EA are listed as below.



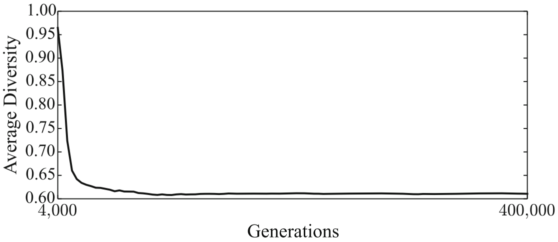
(a) Lin318



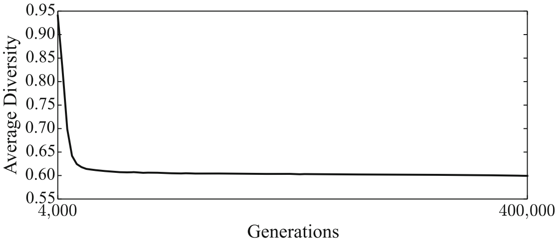
(b) Linhp318



(c) Rd400

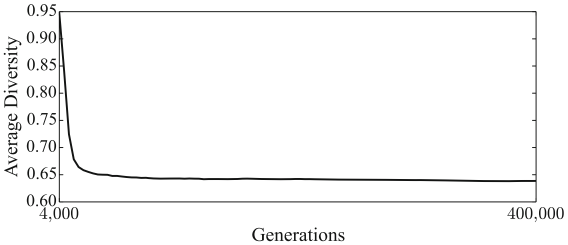


(d) Fl417

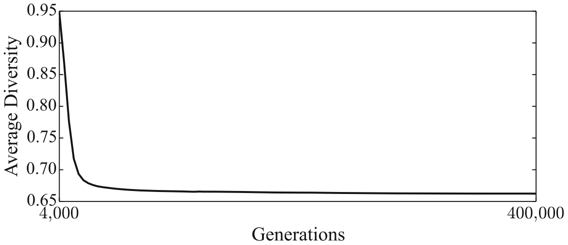


(e) Pr439

Fig. 1. The average diversity during runs of the EA for the TSP



(f) Pcb442



(g) D493

Fig. 1. (*continued*)

Table 2. Settings of the EA for the One-Max Problem

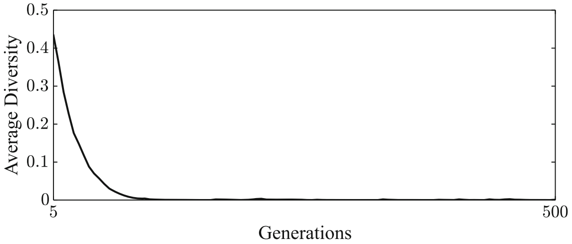
Population size	100
Mutation rate	0.20
Crossover rate	0.80
Length of interval	Five generations
Terminal criterion	500 generations done

In this EA, the method for genotype diversity computation is very simple. Let k' be the number of bits which are same in the two individuals and k be the that of total bits. Then, Formula 2 can be used to compute genotype distance between x_i and x_j. Then, Formula 3 is for diversity computation.

When the scale is set 300 and 500, this EA cannot obtain an optimal. In this case, the EA runs thirty times under the control of given settings, respectively. As the previous experiment, genotype diversity is recorded at each interval in runs. Then, the thirty time average of diversity at every interval is plotted in Fig. 2 to show the changing trend of genotype diversity in runs (Table 2).

For each dataset, it can be seen that the trend in runs is very similar. However, the trend is quietly different with that of the TSP EA runs. In detail, diversity declines sharply and comes to minimum soon.

(a) 300 in scale



(b) 500 in scale

Fig. 2. The average diversity during runs of the EA for the One-Max Problem

3.3 Experiment on CoBiDE

Wang et al. [15] proposed the CoBiDE. The covariance matrix learning based coordinate system is established in the algorithm. Besides, the bimodal distributing parameter setting is employed to control parameters. This algorithm has been tested on benchmark test functions provided by [12], which can be classified into four categories, unimodal functions, basic multimodal ones, expanded multimodal ones and hybrid composition ones. Experimental results in [15] show it has overall better performance compared with some earlier DE variants and other state-of-the-art EAs. The detailed procedure of covariance matrix learning based coordinate system and that of setting two groups of parameter, $F_{i,g}$ and $CR_{i,g}$, can be found in [15]. Settings for the CoBiDE are given in Table 3.

Table 3. Settings of the CoBiDE

Function dimension	30
Population size	60
pb	0.40
ps	0.50
Length of interval	500 generations
Terminal criterion	50000 generations done

The measure to calculate diversity for the CoBiDE is as below. Distance, D, between two individuals, $\boldsymbol{x}_{a,g}$ and $\boldsymbol{x}_{b,g}$, is given in Formula 4, where d denotes the function dimension and g represents generations.

$$D(\boldsymbol{x}_{a,g}, \boldsymbol{x}_{b,g}) = \frac{\sum_{j=1}^{d} |x_{a,j,g} - x_{b,j,g}|}{d} \tag{4}$$

In the formula, $x_{i,j,g}$ is the jth dimension of $\boldsymbol{x}_{i,g}$. Then, diversity, PD, is given in Formula 5 based on Formula 4, where NP still represents the population size and $m_{j,g}$ denotes the median of the jth dimension in the gth generations.

$$PD = \frac{\sum_{i=1}^{NP} \frac{\sum_{j=1}^{d} |x_{i,j,g} - m_{j,g}|}{d}}{NP} \tag{5}$$

Table 4. Results of the CoBiDE

Function	Average (standard deviation)	Average final diversity
F1	0.0000E+00 (0.00E+00)	0.00E+00
F2	1.9390E−28 (1.42E−28)	2.63E−20
F3	1.1434E+02 (1.60E+02)	4.92E−05
F4	2.0600E−28 (1.27E−28)	2.65E−20
F5	7.0031E−12 (9.30E−13)	5.30E−17
F6	1.3829E−26 (3.11E−26)	0.00E+00
F7	2.5449E−03 (5.39E−03)	1.38E−11
F8	2.0001E+01 (2.88E−03)	2.39E−11
F9	0.0000E+00 (0.00E+00)	6.67E−11
F10	4.4942E+01 (1.51E+01)	6.64E−11
F11	6.1589E+00 (3.08E+00)	2.92E−17
F12	3.7778E+03 (4.28E+03)	4.44E−10
F13	1.6766E+00 (4.12E−01)	1.27E−10
F14	1.2356E+01 (4.32E−01)	4.81E−06
F15	4.0667E+02 (5.83E+01)	1.49E−10
F16	9.2541E+01 (6.78E+01)	5.01E−12
F17	7.9355E+01 (2.75E+01)	2.00E−04
F18	9.0423E+02 (8.80E−01)	8.38E−19
F19	9.0429E+02 (1.10E+00)	4.45E−19
F20	9.0411E+02 (5.82E−01)	4.62E−19
F21	5.0000E+02 (0.00E+00)	2.49E−10
F22	8.3259E+02 (2.20E+01)	1.28E−18
F23	5.3416E+02 (1.83E−04)	9.50E−03
F24	2.0000E+02 (0.00E+00)	1.52E−09
F25	2.0962E+02 (5.22E−01)	1.35E−01

We run the CoBiDE 30 times for each of the 25 functions. Results are given in Table 4. According to Table 4, the CoBiDE can obtain an optimal at 100% for F1 and F9. In fact, when we observe the detailed results, we find that the CoBiDE obtains an optimal in a part of runs for F2, F4, F6 and F7. In other words, the CoBiDE cannot obtain an optimal of F3, F5, F8 and F10-F25 in the 25 benchmark test functions under the above settings. Therefore, genotype diversity is recorded at each interval in runs for these functions. Then, the average of diversity at every interval is plotted to show the changing trend of diversity in runs for them. In Figs. 3, 4, 5 and 6, we show the changing trend of genotype diversity of the selected 19 functions.

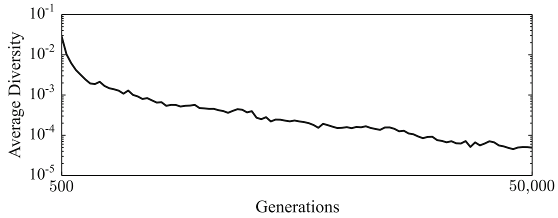
It can be seen that the trend in runs shows difference in different functions. During the whole course, genotype diversity of runs for F3 shows constantly decrease, while that for F23 shows high fluctuates. For remaining functions, the trend in runs consists of one or several declining periods and flat ones. However, details are different. That of F5, F10-F13, F15-F17, F21 and F24-F25 begins with a declining period and then has a flat one. Meanwhile, a declining period and a flat one alternately appear in the trend in runs of F8, F14, F18-F20 and F22. Besides, the trend of these functions except F8 always ends in a flat period. For all functions, diversity never comes to zero.

4 Discussion

In the experiment on the EA for the TSP, runs for all datasets are similar in the changing trend of diversity. In detail, diversity declines only at the beginning and then remains at a level. The value of genotype diversity stable in the later stage shows that runs come to the steady state. On one hand, these runs never find an optimal. On the other hand, diversity value is still high after the steady state coming. It can be inferred that runs of this EA are always trapped into premature stagnation.

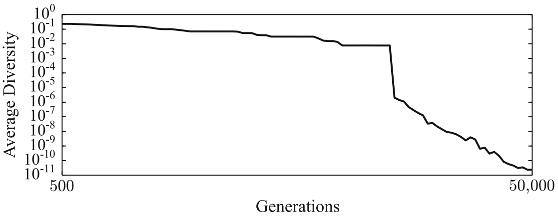
The experiment on the EA for the one-max problem shows that genotype diversity in runs declines to the minimum soon. However, no optimal is obtained in runs. The above phenomena illustrate that runs of this EA fall into premature convergence.

In the experiment on the CoBiDE, the changing trend of genotype diversity is very complicated and needs be further analyzed. Since the trend of runs for F3, F8 and F23 does not finish with a flat period, it can be inferred that they still do not fall into premature stagnation or premature convergence after 50000 generations. Meanwhile, the trend of runs for other functions always ends in a flat period with a value larger than zero. Besides, an optimal is never obtained in these runs. Therefore, these runs fall into premature stagnation. Further, the fact that the trend in runs of F14, F18-F20 and F22 shows more than one flat period can be attributed to the effect of particular procedures applied in the CoBiDE, such as the covariance matrix learning. In detail, though the sign of premature stagnation, a flat period with diversity value higher than zero, appears in runs, the particular procedures may make runs jump out of premature stagnation for one or more times.

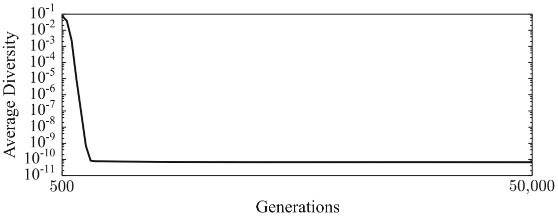


(a) F3

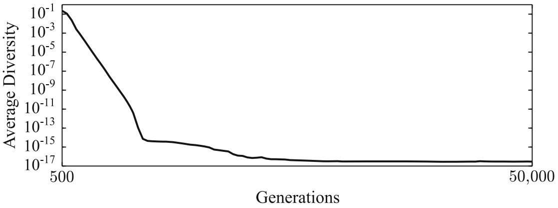
(b) F5



(c) F8

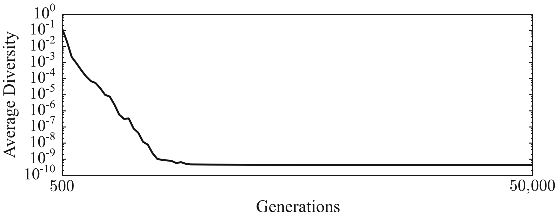


(d) F10

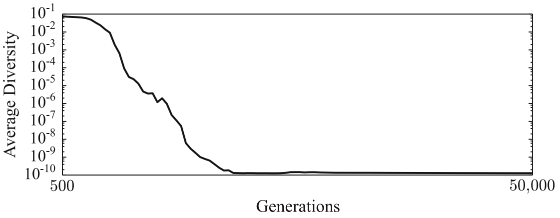


(e) F11

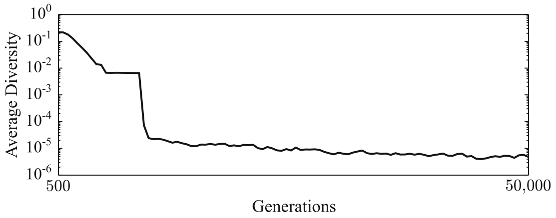
Fig. 3. The average diversity during runs of the CoBiDE (part 1)



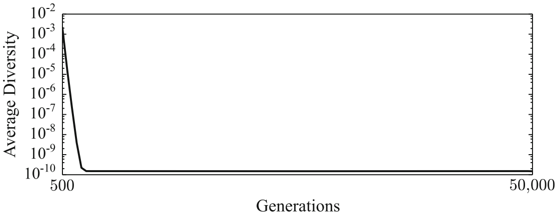
(a) F12



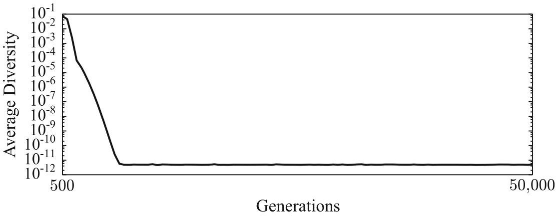
(b) F13



(c) F14

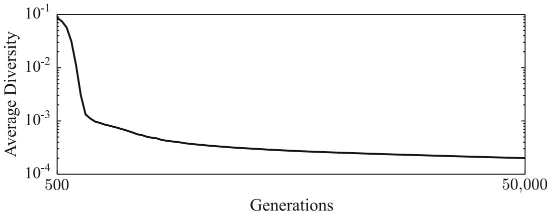


(d) F15

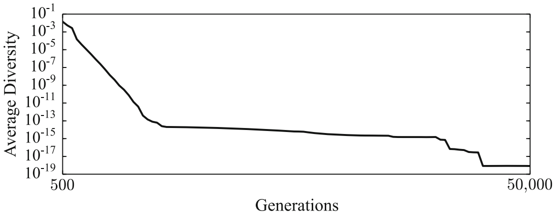


(e) F16

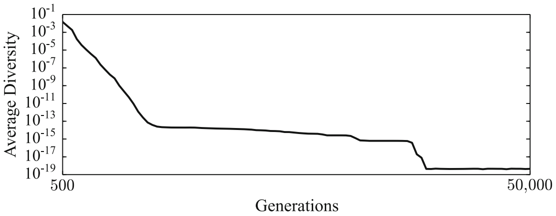
Fig. 4. The average diversity during runs of the CoBiDE (part 2)



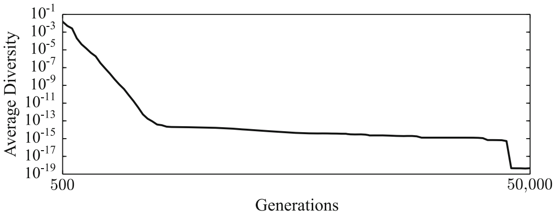
(a) F17



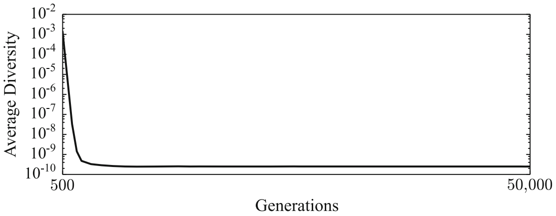
(b) F18



(c) F19

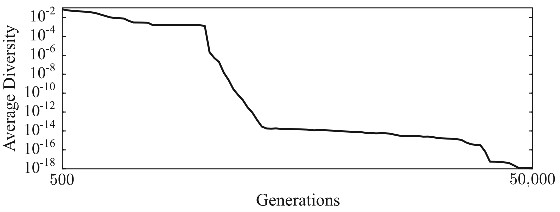


(d) F20

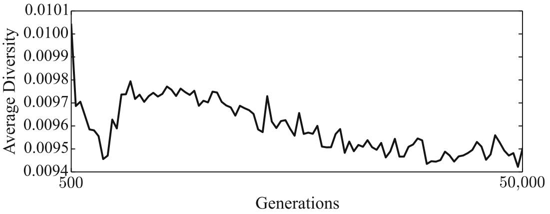


(e) F21

Fig. 5. The average diversity during runs of the CoBiDE (part 3)

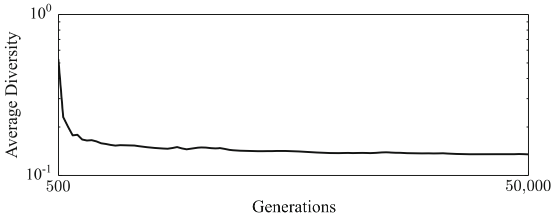


(a) F22



(b) F23

(c) F24



(d) F25

Fig. 6. The average diversity during runs of the CoBiDE (part 4)

Based on our experimental results, we can explain causes of the imbalance between exploration and exploitation as follow. In general, the combination of operators and settings of an EA is the main cause of the imbalance. When it is the main cause, runs of an EA for different tasks show the similar changing trend of diversity. For example, runs of the GA for the TSP for the seven datasets show premature stagnation, while those of the GA for the one-max problem for the two scales show premature convergence. However, a state-of-the-art EA can provide a more appropriate ratio of exploration and exploitation by its combination of well-designed operators and dynamic settings. In this situation, the combination of operators and settings cannot be the main cause of the imbalance. Instead, fitness landscapes of the current task becomes the more important cause of the imbalance than the combination of operators and settings. Therefore, runs for different functions which are distinct in fitness landscapes show difference in the changing trend of genotype diversity.

Hence, we can give a method to judge the main cause of the imbalance between exploration and exploitation. The changing trend of genotype diversity in runs for different task can be gotten by plotting average of diversity at every interval just as we do. Provided that the changing trend is similar in runs for different tasks, the main cause of the imbalance is the combination of operators and settings of this EA. In this case, modifying algorithm may lead to improvement on solutions for all tasks. However, if the trend is different in runs for different tasks, the main cause is fitness landscapes of the tasks. In this case, to improve solutions, algorithm need be modified for a special type of tasks.

5 Concluding Remark

In this paper, we executed experiments on three EAs. In experiments, we select datasets, scales and functions to make runs fall into stagnation or premature convergence. Based on experimental data, we used figures to show the changing trend of genotype diversity in runs. Based on the figures, we drew a conclusion as follow. The changing trend of diversity similar in runs illustrates that the main cause of the imbalance between exploration and exploitation is the combination of operators and settings of the current EA, while the trend showing different in runs for different tasks reveals that fitness landscapes of tasks is the main cause, instead. In this way, the main cause can be detected. Our work can help to find some more pointed measures based on features shown in runs for achieving the balance for further improving solutions. In the future, we will focus on proposing such measures.

References

1. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE Trans. Evol. Comput. **6**(5), 443–462 (2002)
2. Burke, E.K., Gustafson, S., Kendall, G.: Diversity in genetic programming: an analysis of measures and correlation with fitness. IEEE Trans. Evol. Comput. **8**(1), 47–62 (2004)
3. Chang, P.C., Huang, W.H., Ting, C.J.: Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems. Expert Syst. Appl. **37**(3), 1863–1878 (2010)
4. Črepinšek, M., Liu, S.H., Mernik, M.: Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput. Surv. (CSUR) **45**(3), 35 (2013)

5. Eiben, A.E., Schippers, C.A.: On evolutionary exploration and exploitation. Fundamenta Informaticae **35**(1–4), 35–50 (1998)
6. Galván-López, E., McDermott, J., O'Neill, M., Brabazon, A.: Towards an understanding of locality in genetic programming. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 901–908. ACM (2010)
7. Izzo, D., Getzner, I., Hennes, D., Simões, L.F.: Evolving solutions to TSP variants for active space debris removal. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 1207–1214. ACM (2015)
8. Mavrovouniotis, M., Müller, F.M., Yang, S.: Ant colony optimization with local search for dynamic traveling salesman problems (2016)
9. Paenke, I., Jin, Y., Branke, J.: Balancing population-and individual-level adaptation in changing environments. Adapt. Behav. **17**(2), 153–174 (2009)
10. Pan, G., Li, K., Ouyang, A., Li, K.: Hybrid immune algorithm based on greedy algorithm and delete-cross operator for solving TSP. Soft Comput. **20**(2), 555–566 (2016)
11. Reinelt, G.: TSPLIB—A traveling salesman problem library. ORSA J. Comput. **3**(4), 376–384 (1991)
12. Suganthan, P.N., et al.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. KanGAL Rep. **2005005**, 2005 (2005)
13. Syswerda, G.: Uniform crossover in genetic algorithms (1989)
14. Tao, G., Michalewicz, Z.: Inver-over operator for the TSP. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 803–812. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0056922
15. Wang, Y., Li, H.X., Huang, T., Li, L.: Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. Appl. Soft Comput. **18**, 232–247 (2014)