A Novel Strategy to Control Population Diversity and Convergence for Genetic Algorithm

Dongyang Li[1], Weian Guo[2](✉), Yanfen Mao[2], Lei Wang[1], and Qidi Wu[1]

[1] School of Infomation and Electronical Information, Tongji University, Shanghai 201804, China
[2] Sino-German College of Applied Sciences, Tongji University, Shanghai 201804, China
guoweian@163.com

Abstract. Genetic algorithm (GA), an efficient evolutionary algorithm inspired from the science of genetics, attracts the worldwide attention for several decades. This paper tries to strengthen the search ability of the population in GA in the way of improving the distance among individuals by introducing a new solution updating strategy based on the theory of Cooperative Game. The simulation is done using fourteen benchmark functions, and the results demonstrate that this modified genetic algorithm works efficiently.

Keywords: GA · Solution distance · Cooperative game · Solution updating strategy

1 Introduction

Genetic algorithm (GA), a powerful heuristic algorithm which is established by mimicking the processes of inheritance evolution of biological, was first proposed by Holland in 1975 [1]. Since then, a number of studies have been done and prove that GA exhibits excellent optimization performance, not only on various kinds of numerical benchmarks, but also on high-dimensional and multi-objective optimization problems. In original GA, the algorithm has already outperformed many other evolutionary algorithms for some benchmarks. However, it shows a poor performance on the multi-peak functions usually. In another word, GA tends to a fast convergence to a local minimum. Therefore, it need to be improved in many of its aspects.

Since the advent of GA, many studies have been done to improve the searching ability of GA by improving the diversity of its population. For the multi-mode resource-constrained project scheduling problem, Peteghem and Vanhoucke utilize a strategy of two separate populations to enhance the seeking ability of GA and achieve good results [2]. When solving the problem of flexible job-shop scheduling, Zhang not only modifies the mutation and crossover methods, but also proposes a new population initialization strategy based on the global selection and local selection [3]. Vidal and Crainic equip GA with adaptive diversity management by introducing the efficient local

Y. Tan et al. (Eds.): ICSI 2017, Part I, LNCS 10385, pp. 362–369, 2017.
DOI: 10.1007/978-3-319-61824-1_39

search based improvement procedures and diversity management approaches [4]. In the study of Castro and Soma [5], the set of feasible solutions should be made a division into regions in order to diversify the search that is used on a GA variation. An adaptive method of maintaining variable population size is proposed by Arabas and Michalewicz [6], which is used to improve the diversity of GA. Liu and Zhong [7] turn out that the population of GA which is initialized by ACO instead of randomly generated will strengthen the global search ability of GA and can be used for QoS-aware service composition problem perfectly. In [8], Tsai and Huang make an improvement of GA by consisting of two parallel EGAs along with a migration operator, their results demonstrate that this modified GA takes advantages of better population diversity, inhibiting premature convergence, and keeping parallelism in comparison with conventional Gas. In addition, Wang [9] uses ACO to improve GA to solve the problem of Job-Shop Scheduling. Tang and Pan [10] extend the search ability and convergence of original GA by incorporating an infeasible solution repairing procedure and a local optimization procedure.

Obviously, all these studies mentioned above are just trying to improving the search ability of GA by increasing the diversity of the population. The reason is that in original GA, the regular design that higher crossover probability and lower mutation probability will lead the population to poor diversity and premature convergence. In this paper, in order to improve the searching ability of GA and optimize the population of GA, we propose a new solution updating strategy to improve the search ability of GA by utilizing the theory of Cooperative Game to increase the distance between individuals in the population.

The rest of this paper is organized as follows. In Sect. 2, a brief overview of GA will be introduced. In Sect. 3, the new solution updating strategy will be presented. The simulation results will be compared and discussed in Sect. 4. We end this paper with conclusion in Sect. 5.

2 Genetic Algorithm

2.1 The Principle of Genetic Algorithm

Genetic algorithm (GA) is just established based on the evolution mechanism of natural selection and natural inheritance. The optimization problem is mapped to an evolution process, and it is considered to be the natural environment, a potential solution is seen as a chromosome which is called as an individual, and a set of solutions is defined as a species which we call it population. First, GA generates a population randomly and arranges the fitness values for individuals based on the fitness function set according to the optimization problem. Then a new population will be produced by selecting enough individuals to crosses with others and mutates. Note that, an individual with a higher fitness value should be selected with a higher probability than an individual with a lower fitness value. Through repeating this operation, potential solutions will be more and more close to the optimal solution of the problem, and the approximate optimal solution can be obtained finally.

2.2 Implementation Steps of GA

After a large efforts of scholars, a number of different genetic algorithms have been proposed. The GA proposed by Holland in 1975 is seen as the standard GA (SGA). The main steps are as follows:

(1) Encoding. Encoding is a bridge between problems and algorithms.
(2) Generate an initial population.
(3) Fitness evaluation. A fitness function ought to be defined reasonably to reflect the adaption ability to the environment of individuals.
(4) Selection operator. Selection operator is a reflection of "survival of the fittest". Generally, the probability of being selected of an individual is proportional to its fitness.
(5) Crossover operator. Crossover operator is an important operator of GA. The single point crossover strategy is adopted in SGA.
(6) Mutation operator. A random bit of the chromosome may be changed based on a mutation probability *Pm*.
(7) Ending criteria. An ending principle must be set to terminate GA.

Where *MaxGen* is the maximum of evolution algebra, *N* is the size of the Population while Pc and Pm is the crossover probability and mutation probability respectively.

The ability of dealing with many complex problem of GA is strong has been proved for many years. However, due to its very small probability of mutation, its global exploration too poor to solve the problem with a large solution space or with a number of local minimums. Hence the main drawback of GA is tend to a premature convergence.

3 An Improved GA Based on Cooperative Game

As discussed above, GA does not have a suitable strategy to solve the problem with a large solution space or too many local minimums. In this section, we will provide an brief introduction of Cooperative Game and propose a new solution updating strategy to improve GA.

3.1 Cooperative Game

Cooperative game is first proposed by Rowland [11] in 1944. The focus of Cooperative game is how people share the benefits through cooperation, that is, the issue of benefits distribution. Cooperation and compromise is the core of cooperative game. The purpose of Cooperative game is the interests of both sides in the game have increased, or at least one's benefits of the increase, while the other one is not compromised, and the interests of the whole society will get an increase finally. More details about cooperative game can be found in [11].

A hot topic in GA is the outstanding genes in the parents must be remained to the offspring, while each individual of the offspring also should be better to keep a proper distance to others during the progression of evolution. This can be seen as a cooperative game between individuals of which the goal is to make sure the population has both proper distance among solutions and convergence. Based on this idea, this paper proposed a new solution updating strategy to increase the distance among individuals without breaking its convergence.

3.2 CGGA

As mentioned above, we should give a new solution updating strategy to increase the distance among solutions of population but cannot make GA be a random search algorithm. And this strategy based on cooperative game is introduced as follows.

(1) The *population* is seen as a game alliance S and the benefits of it is defined as $v(S)$, which is evaluated by the rate of increase shown as formula (1).

$$v(S) = w_1 \cdot \frac{D(P_1) - D(P)}{D(P)} + \frac{sum(fit) - sum(fit_1)}{sum(fit)} \tag{1}$$

Where w_1 is the weight of distance between solutions which is calculated by formula (2), *fit* is the sum of fitness of the parents, while fit_1 is the sum of fitness of the offspring, $D(P)$ and $D(P_1)$ represent the distance among individuals of parents and offspring respectively, which is defined as formula (3).

$$w_1 = \frac{popsize}{sum(fit)} \cdot N(MinFit)/20 \tag{2}$$

$$D(P) = Var(Fit) \tag{3}$$

Here, $N(MinFit)$ is the number of consecutive equal values of *MinFit* up to the current generation, and $Var(\cdot)$ is the variance function which is used to evaluate the variance of *Fit* of *P*.

(2) Every two individuals selected to do the crossover and mutation operation will begin a cooperative game and aim to increase $v(S)$ without harming the benefits of each other. Note that a margin is set as 1.2 to avoid a too low probability of evolving. The individual *individual* that has executed the crossover and mutation will be noted as $individual_i$. The benefits of an individual is defined as in formula (4).

$$v(i) = w_1 \cdot \frac{D(P_{1c}) - D(P)}{D(P)} + \frac{sum(fit_i) - sum(fit_{ic})}{sum(fit_i)} \tag{4}$$

Where P_{ic} represents the population that replaces individual *Pi* with using $individual_i$, fit_i and fit_{ic} represent the fitness of $individual_i$ and $individual_{ic}$.

In summary, the idea of the new solution updating strategy is every two selected individuals ready to evolve should try to improve $v(S)$ and cannot reduce the interests of its partner.

4 Simulation and Discussion

In this section, we make a simulation experiment using fourteen benchmark functions. And the results will be shown in this part.

The fourteen benchmark function which is shown in Table 1 are selected to test the modified GA.

Table 1. Benchmark functions.

Function	Name	Domain
F1	Ackley	$-30 \leq x \leq 30$
F2	Fletcher	$-\pi \leq x \leq \pi$
F3	Griewank	$-600 \leq x \leq 600$
F4	Penalty1	$-50 \leq x \leq 50$
F5	Penalty2	$-50 \leq x \leq 50$
F6	Quartic	$-1.28 \leq x \leq 1.28$
F7	Rastrigin	$-5.12 \leq x \leq 5.12$
F8	Rosenbrock	$-2.0481 \leq x \leq 2.048$
F9	Schwefel	$-65.536 \leq x \leq 65.536$
F10	Schwefel2	$-100 \leq x \leq 100$
F11	Schwefel3	$-10 \leq x \leq 10$
F12	Schwefel4	$-512 \leq x \leq 512$
F13	Sphere	$-5.12 \leq x \leq 5.12$
F14	Step	$-200 < x < 200$

More details of these benchmarks can be find in [12–14]. Five other popular EAs are also used to compare with the new algorithm CGGA. They are GA [15, 16], PBIL [17], standard PSO [18–20], ACO [21, 22], and ES [23].

Additionally, for GA and CGGA, we chose the roulette wheel selection, single point crossover, the crossover probability = 1, and the mutation probability = 0.01. And the parameters for other EAs can be found in [24].

In this simulation, the mean optimization results and the best optimization results of each EAs are shown in Tables 2 and 3.

For all of these EAs, Table 2 shows that the mean optimization results of CGGA are better than other EAs in all cases, which is benefit from the healthy distance among individuals in the population. Besides, the best optimization results, which are shown in Table 3, indicate that in the process of the evolution, CGGA performs better in searching the minimum than others in most cases, just for Fletcher, Penalty1, and

Table 2. Mean optimization results

Function	GA	CGGA	PBIL	PSO	ACO	ES
F1	6.96E+00	**2.14E+00**	1.86E+01	1.44E+01	5.89E+00	8.87E+00
F2	2.20E+04	**1.46E+04**	3.44E+05	3.33E+05	4.30E+05	5.52E+05
F3	1.28E+00	**1.10E+00**	1.71E+02	5.17E+01	1.15E+00	9.80E+01
F4	5.40E−02	**2.12E−02**	3.92E+07	1.17E+06	9.07E+07	3.95E+07
F5	6.04E−01	**2.11E−01**	1.18E+08	8.14E+06	1.78E+08	1.13E+08
F6	1.93E−06	**1.16E−06**	1.09E+01	1.49E+00	3.40E−03	1.47E+01
F7	2.63E+01	**2.75E+00**	2.00E+02	1.37E+02	7.39E+01	2.18E+02
F8	3.88E+01	**3.71E+01**	1.29E+03	3.43E+02	8.41E+02	2.48E+03
F9	4.50E+01	**2.54E+01**	4.28E+03	3.63E+03	3.16E+01	2.94E+03
F10	2.90E+03	**6.85E+02**	9.82E+03	5.41E+03	1.96E+03	1.29E+04
F11	3.89E+00	**8.95E−01**	5.37E+01	2.73E+01	2.20E+01	7.54E+01
F12	1.86E+01	**5.97E+00**	5.90E+01	3.52E+01	2.02E+01	2.10E+01
F13	9.23E−02	**8.06E−03**	5.31E+01	1.52E+01	8.02E+00	6.77E+01
F14	1.30E+01	**5.05E+00**	1.91E+04	5.67E+03	2.11E+01	1.59E+04

Table 3. Best optimization results

Function	GA	CGGA	PBIL	PSO	ACO	ES
F1	3.50E+00	**9.66E-01**	1.67E+01	1.09E+01	4.05E+00	6.66E+00
F2	**1.83E+03**	4.61E+03	1.84E+05	1.87E+05	2.62E+05	1.97E+05
F3	1.06E+00	**1.05E+00**	1.02E+02	2.63E+01	1.05E+00	5.97E+01
F4	**4.10E−03**	4.27E−03	4.35E+06	5.25E+04	2.29E+01	2.66E+06
F5	1.23E−01	**1.11E−01**	2.40E+07	1.10E+06	1.35E−32	1.91E+07
F6	4.50E−07	**7.00E−08**	4.68E+00	3.80E−01	1.30E−03	4.87E+00
F7	1.14E+01	**0.00E+00**	1.74E+02	1.09E+02	5.18E+01	1.70E+02
F8	9.84E+00	**7.43E+00**	4.04E+02	1.72E+02	3.94E+02	1.03E+03
F9	**3.83E+00**	6.63E+00	3.14E+03	2.68E+03	9.85E+00	2.30E+03
F10	1.01E+03	**2.20E+02**	5.59E+03	3.09E+03	4.62E+02	7.33E+03
F11	1.40E+00	**4.00E−01**	4.39E+01	1.48E+01	5.70E+00	4.80E+01
F12	7.00E+00	**3.00E+00**	4.26E+01	2.75E+01	6.40E+00	1.29E+01
F13	1.01E−02	**0.00E+00**	2.94E+01	8.64E+00	2.73E+00	3.59E+01
F14	**1.00E+00**	**1.00E+00**	1.27E+04	2.70E+03	9.00E+01	1.02E+04

Schwefel, the best optimization results of CGGA are the second best. It shows that CGGA works more effective than all the other EAs in finding out the minima of the fourteen benchmarks mentioned above.

5 Conclusions

In this paper, we improve GA based on the theory of cooperative game, and propose a new evolutionary algorithm CGGA. With considering the distance among individuals of population, it uses the cooperation mechanism of cooperative game to equip the solutions with a strong ability of searching. In this way, the distance among individuals in the population increased. A set of fourteen benchmarks is used to test and compare the proposed algorithm with several popular EAs. The results demonstrate the proposed algorithm is effective to deal with optimization problems.

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems, pp. 211–247. MIT Press, Cambridge (1975)
2. Peteghem, V.V., Vanhoucke, M.: A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. Eur. J. Oper. Res. **201**(2), 409–418 (2010)
3. Zhang, G., Gao, L., Shi, Y.: An effective genetic algorithm for the flexible job-shop scheduling problem. ACM Trans. Intell. Syst. Technol. **38**(4), 3563–3573 (2011)
4. Vidal, T., Crainic, T.G., Gendreau, M., et al.: A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Comput. Oper. Res. **40**(40), 475–489 (2013)
5. Castro, J.L.D., Soma, N.Y.: A constructive hybrid genetic algorithm for the flowshop scheduling problem. Int. J. Comput. Sci. Netw. Secur. **9**, 219–223 (2013)
6. Arabas, J., Michalewicz, Z., Mulawka, J.: GAVaPS - a genetic algorithm with varying population size. In: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, vol. 1, pp. 73–78. IEEE Xplore (1994)
7. Liu, H., Zhong, F., Ouyang, B., et al.: An approach for QoS-aware web service composition based on improved genetic algorithm. In: International Conference on Web Information Systems and Mining, pp. 123–128. IEEE Xplore (2010)
8. Tsai, C.C., Huang, H.C., Chan, C.K.: Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. IEEE Trans. Industr. Electron. **58**(10), 4813–4821 (2011)
9. Wang, L., Haikun, T., Yu, G.: A hybrid genetic algorithm for job-shop scheduling problem, pp. 271–274 (2015)
10. Tang, M., Pan, S.: A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. Neural Process. Lett. **41**(2), 211–221 (2015)
11. Rowland, E.: Theory of Games and Economic Behavior. Theory of games and economic behavior, pp. 2–14. Princeton University Press (1944)
12. Back, T.: Evolutionary Algorithms in Theory and Pratice. Oxford University Press, Oxford (1996)
13. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. **3**, 82–102 (1999)
14. Cai, Z., Wang, Y.: A multiobjective optimization-based evolutionary algorithm for constrained optimization. IEEE Trans. Evol. Comput. **10**, 658–675 (2006)
15. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, New York (1992)
16. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
17. Parmee, I.: Evolutionary and Adaptive Computing in Engineering Design. Springer, New York (2001)

18. Onwubolu, G., Babu, B.: New Optimization Techniques in Engineering. Springer, Berlin (2004)
19. Eberhart, R., Shi, Y., Kennedy, J.: Swarm Intelligence. Morgan Kaufmann, San Mateo (2001)
20. Clerc, M.: Particle Swarm Optimization. ISTE Publishing, Amsterdam (2006)
21. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
22. Dorigo, M., Gambardella, L., Middendorf, M., Stutzle, T.: Special section on 'ant colony optimization'. IEEE Trans. Evol. Comput. **6**(4), 317–365 (2002)
23. Guo, W., Wang, L., Ge, S.S., Ren, H., Mao, Y.: Drift analysis of mutation operations for biogeography-based optimization. Soft Comput. **19**, 1881–1892 (2015)
24. Li, D., Wang, L., et al.: Particle swarm optimization-based solution updating strategy for biogeography-based optimization. In: IEEE Congress on Evolutionary Computation (CEC), pp. 455–459 (2016)