Fitness-Distance Balance with Functional Weights: A New Selection Method for Evolutionary Algorithms

Kaiyu WANG†, Sichen TAO†, Rong-Long WANG††, Yuki TODO††, Nonmembers, and Shangee GAO††, Member

SUMMARY In 2019, a new selection method, named fitness-distance balance (FDB), was proposed. FDB has been proved to have a significant effect on improving the search capability for evolutionary algorithms. But it still suffers from poor flexibility when encountering various optimization problems. To address this issue, we propose a functional weights-enhanced FDB (FW). These functional weights change the original weights in FDB from fixed values to randomly generated ones by a distribution function, thereby enabling the algorithm to select more suitable individuals during the search. As a case study, FW is incorporated into the spherical search algorithm. Experimental results based on various IEEE CEC2017 benchmark functions demonstrate the effectiveness of FW.

key words: evolutionary algorithms, fitness-distance balance, functional weights, selection method

1. Introduction

In recent years, with the development of society and the advancement of technology, optimization problems have become more and more complex [1]. Traditional mathematical methods cannot solve these problems in limited time. To achieve a reasonable trade-off between solution accuracy and computational efficiency, many evolutionary algorithms (EAs) have been proposed [2], [3]. An EA treats the problem as a black-box and it can get a quality solution within limited computational time. Representative EAs include genetic algorithm [4], differential evolution [5], [6], gravitational search algorithm [7], [8] and spherical search (SS) [9]. EAs have been applied in various fields and play an important role in scientific research and production practices [10]–[12].

An EA consists of several components (e.g., a population of individuals, selection method, and crossover or mutation operators) that significantly affect its performance and should be designed sophisticatedly. Researchers have improved EA’s performance from many aspects, such as population size and structure [13], [14], and parameter control [15]. However, relatively few studies are performed regarding the effect of a selection method on the algorithm. The selection methods in EAs can be classified as non-deterministic, deterministic and probabilistic [16]. The nondeterministic strategy randomly selects candidate solutions in the population, which ensures the diversity of the population during the search [17]. The deterministic strategy generally selects better fitness solutions as candidate solutions to make promising solutions survive into the next generation, and provide search directions for the subsequent optimization of the population [5]. The probabilistic strategy selects candidate solutions in a probabilistic way, which not only maintains the population diversity, but also archive a certain percentage of current best solutions [18].

Recently, a new selection method based on the balance of fitness and distance (FDB) is proposed [16]. It is widely accepted that an excellent selection method can improve the performance of an EA substantially, and FDB also shows great potential. But it is observed that FDB only uses a fixed weight in its selection rules, which significantly limit its flexibility when facing various problems. It is naturally expected that a selection method can select solutions according to the characteristics of an tackled problem. To realize this, we innovatively propose a new selection strategy based on the FDB. The values of the balance weights for fitness and distance are no longer set to a fixed value, but are generated randomly by a distribution function. This new functional weights generation strategy enables the FDB to select candidate solutions more efficiently. With the aid of FW, more promising solutions can be survived into the next generation and thus make the algorithm to achieve a satisfied trade-off between solution fitness and population diversity.

The main contributions of this study are two-fold:

1) To the best of our knowledge, we are the first to improve the FDB selection method, aiming to achieve a better balance between fitness and distance of solutions.

2) A case study based on SS is performed, and experiments are conducted based on thirty different functions. Extensive performance comparison shows the superiority of FW in comparison with FDB. The results can raise the importance of the selection method in EAs and promote the development of related fields.

2. Fitness-Distance Balance

FDB, as a selection method, is aimed to discover one or more candidate solutions that are more suitable for the population search. According to [16], the distance of FDB is evaluated as:

[Equation]

Copyright © 2021 The Institute of Electronics, Information and Communication Engineers
\[ \forall i = 1, P_i \neq P_{best}, \]
\[ D_{P_i} = \sqrt{(x_{1P_i} - x_{1P_{best}})^2 + \cdots + (x_{nP_i} - x_{nP_{best}})^2} \]  
(1)

where \( P_{best} \) and \( P_i \) indicate the current optimal solution and the \( i \)th solution of the population, respectively. \( x_{nP_i} \) and \( x_{nP_{best}} \) denote the position information of the \( n \)th and \( P_{best} \) solutions in the \( n \)th dimension, respectively.

After normalization, the fitness and distance values of candidate solutions are used in the score calculation, shown as:

\[ \forall i = 1, S_{P_i} = \omega \cdot \tilde{F}_{P_i} + (1 - \omega) \cdot \tilde{D}_{P_i} \]  
(2)

where \( \tilde{F}_{P_i} \) indicates the normalized fitness of solutions, whose fitness function is expressed as \( F \). \( \tilde{D}_{P_i} \) is the normalized distance of \( D_{P_i} \). \( \omega = 0.5 \) is a fixed coefficient.

Different from traditional selection methods which only take solution fitness into consideration in EAs, e.g., the roulette wheel selection, and elite selection [4], FDB can simultaneously consider the effect of solutions’ fitness values and their distances. Crowding solutions that result in a small \( D_{P_i} \) value are not likely to be selected, thus FDB tends to retain solutions with better fitness values from different regions of the search space. By using Eq. (2), the trade-off between solution quality and population diversity can be achieved to some extend.

### 3. Fitness-Distance Balance with Functional Weights

In FDB, the coefficient \( \omega \) balances the trade-off between solution quality and population diversity. Obviously, different optimization problems possess distinct search space. Thus, such trade-off should be adjustable according to the characteristics of the handled problem. Based on this motivation, we proposed a novel functional weights selection method (FW) in this paper. The functional weights strategy changes the limitation in FDB that the value of \( \omega \) is fixed. This strategy enhances the performance and ability to cope with complex problems of EAs.

In Eq. (2), we attempt to randomly generate the functional weights \( \omega \) according to Gaussian and Cauchy distributions, as shown in Eqs. (3) and (4), respectively.

\[ \omega \sim N(\mu, \delta^2) \]  
(3)

\[ \omega \sim \text{Cauchy}(\gamma, x_0) \]  
(4)

A fixed value of weight \( \omega \) may be valid, but cannot always be the best for all kinds of problems. Using a distribution function allows the value of weight \( \omega \) to generate more possibilities, thus possessing more capacity to tackle various problems. As the search space of a tackled problem may have different landscapes, Gaussian and Cauchy distributions tend to manipulate the scenarios with approximated normal distribution and fat-tailed properties, respectively [5], [19]. The values of weights \( \omega \) of a Gaussian distribution have less variance during the entire iterations of the population, and more values are generated around the mean. The values of weights \( \omega \) with a Cauchy distribution are more distributed at the ends of tails, and the variance is larger. It is notable that the generation of \( \omega \) will be re-sampled until it is located within the interval \([0, 1]\). Both generation strategies are expected to enable EAs to possess a more flexible trade-off between fitness and distance of solutions.

### 4. Experimental Results

To test the FW selection method, as a case study we applied it to one of the recent proposed algorithms, i.e., SS. The SS is a new algorithm proposed by Kumar et al. in 2019 [9], which has shown its superior performance in comparison with other state-of-the-art algorithms including LSHADE [20] and EBOwithCMAR [21]. Regarding SS, more details can be referred in [9].

To verify the performance of FW, four kinds of strategies on FDB are comprehensively studied, as summarized in Table 1. In Strategy 1, \( \omega \) is set to a fixed value. \( \omega = 0.5 \) is corresponding to the original FDB [16], which means that the fitness and distance values of solutions are equally treated. \( \omega = 0.75 \) denotes that more attention is paid on the solutions’ fitness, while \( \omega = 1 \) indicates only solution fitness is considered in the selection. FDB2SS, FDB3SS and FDB4SS denote that the weights are randomly selected from a specific group of given weights at each iteration. Preliminary experimental results suggests that \( \omega = 0.75 \) is the best choice among all fixed settings. In Strategy 2, the value of weight is uniformly generated in a given interval. In Strategy 3, the value of weight varies with the number of function evaluations, where \( nFES \) and \( MFES \) mean the current and maximum number of function evaluations, respectively. It is widely recognized that in the early stages of evolution, EAs tend to be more exploratory, while in the later stages of evolution they tend to be more exploitative [22]. Inspired by this, we set weights \( \omega \) to increase with the number of function evaluations for granted. In Strategy 4, \( \omega \) is generated according to a distribution function. The original FDB set \( \omega \) as 0.5, thus it is natural to using a Gaussian distribution with

| Table 1 | Strategies for FDB. |
|---------|---------------------|
| Strategy | Abbreviation of EAs | Setting of \( \omega \) |
| 1       | FDBSS              | \( \omega = 0.5 \) |
|         | FDB1SS             | \( \omega = 0.75 \) |
|         | SS                 | \( \omega = 1 \) |
|         | FDB2SS             | \( \omega \sim [0.2, 0.4, 0.6, 0.8] \) |
|         | FDB3SS             | \( \omega \sim [0.5, 0.6, 0.7, 0.8, 0.9] \) |
|         | FDB4SS             | \( \omega \sim [0.5, 0.75, 1] \) |
| 2       | FDB5SS             | \( \omega = \text{rand}(0, 1) \) |
|         | FDB6SS             | \( \omega = \text{rand}(0.5, 1) \) |
| 3       | FDB7SS             | \( \omega = 0.2 + 0.6 \cdot (nFES/MFES) \) |
|         | FDB8SS             | \( \omega = 0.5 + 0.5 \cdot (nFES/MFES) \) |
|         | FDB9SS             | \( \omega = nFES/MFES \) |
| 4       | FWn1SS             | \( \omega = N(1/2, (1/6)^2) \) |
|         | FWn2SS             | \( \omega = N(3/4, (1/12)^2) \) |
|         | FWc1SS             | \( \omega \sim \text{Cauchy}(1, 4, 3/4) \) |
|         | FWc2SS             | \( \omega \sim \text{Cauchy}(1/5, 3/4) \) |
|         | FWc3SS             | \( \omega \sim \text{Cauchy}(1/6, 3/4) \) |
a mean value of 0.5 (i.e., FW15SS) for comparison. After observed that $\omega = 0.75$ is the most effective setting when using a fixed value, we conducted several further researches by using Gaussian and Cauchy distributions with mean of 0.75.

To verify the performance of these algorithms, we conduct experiment based on the IEEE CEC2017 benchmark functions. These functions include unimodal functions (F1-F3), simple multimodal functions (F4-F10), hybrid functions (F11-F20) and composition functions (F21-F30). As a default setting, the function dimension ($D$) is set as 30, $mFES = 10^4D$ and each function was run for 30 independent times. Figure 1 illustrates Friedman test [23] results

| Table 2 | Experiment results on IEEE CEC2017. |
|---------|------------------------------------|
|         | FW2SS  | FW2SS  | FBSSS  | SS      |
|         | Error  | Std    | Error  | Std    | Error  | Std    |
| F1      | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| F2      | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| F3      | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| F4      | 3.56E+01 | 2.67E+01 | 3.56E+01 | 2.67E+01 | 3.56E+01 | 2.67E+01 |
| F5      | 7.50E+00 | 1.28E+00 | 9.44E+00 | 1.75E+00 | 8.78E+00 | 1.42E+00 |
| F6      | 4.56E-09 | 2.50E-08 | 3.64E-08 | 1.99E-07 | 1.94E-08 | 4.73E-08 |
| F7      | 3.82E+01 | 1.53E+00 | 3.85E+01 | 1.69E+00 | 3.84E+01 | 1.59E+00 |
| F8      | 8.56E-00 | 1.55E+00 | 9.37E+00 | 1.47E+00 | 8.88E+00 | 1.70E+00 |
| F9      | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| F10     | 1.63E+00 | 1.93E+00 | 1.67E+00 | 2.33E+00 | 1.71E+00 | 2.08E+00 |
| F11     | 2.10E+01 | 2.30E+01 | 2.17E+01 | 2.25E+01 | 2.15E+01 | 2.30E+01 |
| F12     | 8.89E+02 | 2.89E+02 | 8.95E+02 | 2.94E+02 | 8.98E+02 | 2.96E+02 |
| F13     | 1.54E+01 | 6.28E+00 | 1.69E+01 | 6.81E+00 | 1.64E+01 | 5.99E+00 |
| F14     | 2.73E+01 | 1.33E+00 | 2.21E+01 | 3.79E+00 | 2.16E+01 | 4.04E+00 |
| F15     | 2.34E+00 | 1.42E+00 | 4.60E+00 | 1.90E+00 | 2.74E+00 | 1.08E+00 |
| F16     | 1.79E+02 | 8.88E+01 | 2.39E+02 | 1.18E+02 | 1.74E+02 | 9.41E+01 |
| F17     | 3.37E+01 | 6.02E+00 | 3.94E+01 | 4.69E+00 | 3.56E+01 | 6.41E+00 |
| F18     | 2.16E+01 | 1.04E+00 | 2.12E+01 | 6.95E-01 | 2.23E+01 | 1.29E+00 |
| F19     | 5.31E+00 | 1.80E+00 | 6.77E+00 | 1.30E+00 | 5.02E+00 | 1.49E+00 |
| F20     | 4.78E+01 | 8.60E+00 | 5.49E+01 | 2.06E+00 | 5.06E+01 | 1.17E+01 |
| F21     | 3.46E+02 | 1.54E+00 | 3.09E+02 | 1.78E+00 | 2.08E+02 | 1.85E+00 |
| F22     | 3.46E+02 | 2.85E+00 | 3.46E+02 | 3.26E+00 | 3.45E+02 | 3.02E+00 |
| F23     | 4.22E+02 | 2.02E+00 | 4.23E+02 | 2.25E+00 | 4.22E+02 | 2.07E+00 |
| F24     | 3.87E+02 | 2.32E-02 | 3.87E+02 | 3.46E-02 | 3.87E+02 | 2.92E-02 |
| F25     | 8.88E+02 | 2.77E+01 | 9.17E+02 | 3.93E+01 | 8.89E+02 | 3.71E+01 |
| F26     | 5.01E+02 | 9.54E+00 | 5.00E+02 | 7.63E+00 | 5.05E+02 | 6.67E+00 |
| F27     | 3.15E+01 | 5.17E+01 | 3.36E+02 | 5.17E+01 | 3.11E+02 | 3.26E+01 |
| F28     | 4.37E+02 | 7.05E+00 | 4.46E+02 | 7.61E+00 | 4.39E+02 | 7.40E+00 |
| F29     | 2.05E+03 | 6.95E-01 | 2.03E+03 | 4.83E+01 | 2.08E+03 | 7.89E-01 |
| W/T/L   | NULL    | 2/280  | 17/121 | 13/170  |

Fig. 1  Ranking of FW and its competitor algorithms according to the Friedman test results.

![Fig. 1](image-url)

Fig. 2  Box-and-whisker plot graphs of (a) F5, (b) F15 and (c) F29.

![Fig. 2](image-url)
of all compared 16 algorithms. From it, we can find that FWn2SS (Gaussian distribution) and FWc2SS (Cauchy distribution) obtain the best average ranking, suggesting that FW is the most effective strategy for SS.

To give more insights into the differences among SS with/without FDB and FW, Table 2 summarizes the experimental results of FWn2SS, FWc2SS, FDBSS, and SS, where Error denotes the absolute value of the difference between the real optimal value of the objective function and the optimal value found by the algorithm. Wilcoxon rank-sum test with a significance level $\alpha = 0.05$ is performed and “$W/L$” denote the number of functions on which the control algorithm is significantly better (“+”), tied (“=”) or significantly worse (“−”) than its competitor.

From Table 2, it is clear that, although FW with Gaussian and Cauchy distributions has no significant differences on all test 30 functions, FWn2SS indeed outperforms FWc2SS on F14 and F23, which indicating that Gaussian distribution is a better choice for some specific problems. Additionally, both FWn2SS and FWc2SS significantly outperform FDBSS and SS on more than 13 out of 30 functions, which also verifies that FW is more effective than FDB.

Finally, Fig. 2 shows the box-and-whisker plot for three different types of functions: simple multimodal function (F5), hybrid function (F15) and composition function (F29), where the red line indicates the median of error values and the upper and lower limiting black lines indicate the maximum and minimum non-abnormal values, respectively. It can be found that FWn2SS and FWc2SS are better than other algorithms in terms of the robustness and accuracy of solutions.

5. Conclusions

In this paper, we propose an FW selection method, in which the functional weights strategy is added to FDB, thus improving its selection ability. To assess the performance of FW, 16 variants of SS are conducted. Experimental results suggest that the proposed FW is efficient. These results give more guidance for the design of selection methods in EAs. Future work will be done by trying some other distributions in FW.

References

[1] Y. Wang, Y. Yu, S. Cao, X. Zhang, and S. Gao, “A review of applications of artificial intelligent algorithms in wind farms,” Artificial Intelligence Review, vol.53, no.5, pp.3447–3500, Oct. 2020.

[2] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” IEEE Trans. Neural Networks and Learning Systems, vol.30, no.2, pp.601–614, Feb. 2019.

[3] J. Sun, S. Gao, H. Dai, J. Cheng, M. Zhou, and J. Wang, “Bi-objective elite differential evolution algorithm for multivalued logic networks,” IEEE Trans. Cybernetics, vol.50, no.1, pp.233–246, Jan. 2020.

[4] S. Forrest, “Genetic algorithms,” ACM Comput. Surv. (CSUR), vol.28, no.1, pp.77–80, March 1996.

[5] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, “Chaotic local search-based differential evolution algorithms for optimization,” IEEE Trans. Syst. Man Cybern.: Systems, vol.51, no.6, pp.3954–3967, June 2019. doi:10.1109/TSMC.2019.2956121.

[6] Y. Yu, S. Gao, Y. Wang, and Y. Todo, “Global optimum-based search differential evolution,” IEEE/CAA J. Automatica Sinica, vol.6, no.2, pp.379–394, March 2018.

[7] Z. Lei, S. Gao, S. Gupta, J. Cheng, and G. Yang, “An aggregative learning gravitational search algorithm with self-adaptive gravitational constants,” Expert Systems with Applications, vol.152, p.113396, Aug. 2020.

[8] Y. Wang, S. Gao, Y. Yu, Z. Wang, J. Cheng, and T. Yuki, “A gravitational search algorithm with chaotic neural oscillators,” IEEE Access, vol.8, pp.25938–25948, 2020.

[9] A. Kumar, R.K. Misra, D. Singh, S. Mishra, and S. Das, “The spherical search algorithm for bound-constrained global optimization problems,” Applied Soft Computing, vol.85, p.105734, Dec. 2019.

[10] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, and J. Cheng, “A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models,” Energy Conversion and Management, vol.230, p.113784, Feb. 2021.

[11] J. Cheng, G. Yuan, M. Zhou, S. Gao, C. Liu, H. Duan, and Q. Zeng, “Accessibility analysis and modeling for IoV in an urban scene,” IEEE Trans. Veh. Technol., vol.69, no.4, pp.4246–4256, April 2020.

[12] J. Wang, Y. Sun, Z. Zhang, and S. Gao, “Solving multi-trip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms,” IEEE/CAA J. Automatica Sinica, vol.7, no.4, pp.1134–1153, July 2020.

[13] Y. Wang, S. Gao, Y. Yu, Z. Cai, and Z. Wang, “A gravitational search algorithm with hierarchy and distributed framework,” Knowledge-Based Systems, vol.218, p.106877, April 2021.

[14] Y. Wang, S. Gao, M. Zhou, and Y. Yu, “A multi-layered gravitational search algorithm for function optimization and real-world problems,” IEEE/CAA J. Automatica Sinica, vol.8, no.1, pp.94–109, Jan. 2020.

[15] A.E. Eiben and S.K. Smit, “Parameter tuning for configuring and analyzing evolutionary algorithms,” Swarm and Evolutionary Computation, vol.1, no.1, pp.19–31, March 2011.

[16] H.T. Kahraman, S. Aras, and E. Gedikli, “Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms,” Knowledge-Based Systems, vol.190, p.105169, Feb. 2020.

[17] L. Cui, G. Li, Z. Zhu, Q. Lin, K.C. Wong, J. Chen, N. Lu, and J. Lu, “Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism,” Information Sciences, vol.422, pp.122–143, Jan. 2018.

[18] D.T. Le, D.K. Bui, T.D. Ngo, Q.H. Nguyen, and H. Nguyen-Xuan, “A novel hybrid method combining electromagnetism-like mechanism and firefly algorithms for constrained design optimization of discrete truss structures,” Computers & Structures, vol.212, pp.20–42, Feb. 2019.

[19] A.F. Kamaruzaman, A.M. Zain, S.M. Yusuf, and A. Udin, “Levity flight algorithm for optimization problems—a literature review,” Applied Mechanics and Materials, vol.421, pp.496–501, Sept. 2013.

[20] R. Tanabe and A.S. Fukunaga, “Improving the search performance of shade using linear population size reduction,” 2014 IEEE Congress on Evol. Comput. (CEC), pp.1658–1665, IEEE, 2014.

[21] A. Kumar, R.K. Misra, and D. Singh, “Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retread phase,” 2017 IEEE Congress on Evol. Comput. (CEC), pp.1835–1842, IEEE, 2017.

[22] B. Morales-Castañeda, D. Zaldivar, E. Cuevas, F. Fausto, and A. Rodríguez, “A better balance in metaheuristic algorithms: Does it exist?,” Swarm and Evol. Comput., vol.54, p.100671, May 2020.

[23] J. Carrasco, S. García, M. Rueda, S. Das, and F. Herrera, “Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review,” Swarm and Evol. Comput., vol.54, p.100665, May 2020.