# How does selecting a benchmark function suite influence the estimation of an algorithm's quality?

Iztok Fister*, Suash Deb†,‡, Dušan Fister*, Iztok Fister Jr.*
*University of Maribor,
Faculty of Electrical Engineering and Computer Science
Koroška cesta 46, 2000 Maribor,
Slovenia
iztok.fister@um.si, dusan.fister1@um.si, iztok.fister1@um.si
†Victoria University,
Decision Sciences and Modeling Program,
Melbourne, Australia
‡IT & educational Consultant, Ranchi, Jharkhand, India
suashdeb@gmail.com

***Abstract*—This paper is focused on answering the question how the selection of a testbed on which the newly proposed algorithms are evaluated influence the estimation of an algorithm's quality. New algorithms are usually tested on well-known benchmark function suites, where the goal is to achieve the best results of the algorithm in the shortest time. A lot of questions have arisen when looking for the most suitable testbed, for instance, which benchmark to take, and which version of it is the most representative for determining the best algorithms. In this study, the newly proposed algorithms introducing the coalition game concept for solving global optimization were tested by solving two different benchmark function suites, i.e., CEC-14 and CEC-18, in order to show that selecting the different CEC benchmark suites does not have a crucial impact on estimating the algorithm's quality.**

***Index Terms*—CEC benchmark function suites, coalition game concept, global optimization**

## I. Introduction

The motivation of writting this paper was the comments of reviewers who criticized the outdated CEC-14 function benchmark suite used for evaluating the newly proposed algorithms in a paper published for the Congress of Evolutionary Computation (CEC'19) [1]. According to reviewers, the more novel the benchmark, the more relevant the evaluation of the tested algorithm's quality. In our opinion, the truth of this thesis is slightly questionable, and, therefore, it encouraged us to verify if this really holds in practice.

The newly developed algorithms can be tested on many problem testbeds, such as, for example [2]. However, these testbeds are devoted for solving continuous optimization problems and, in general, represent an easy task for modern stochastic population-based nature-inspired algorithms. Nowadays, there are two conferences that issue sets of testbed problems for promoting the development of new algorithms, which can even compete against each other: The Genetic and Evolutionary Computation Conference (GECCO), and The Conference of Evolutionary Computation (CEC). The former organizes The Black Box Optimization Competition (BBComp) [3], which tries to hide the problem (i.e., black box) to be optimized from the experimenter, while, in the latter case, the original problems are known in advance, but for the purpose of competition, they are modified using shifting, rotation, hybridization, and composition of more functions, in order to mask the original optimum and thus make searching for this much harder [4]–[6].

Interestingly, both mentioned problem testbeds are changed from year to year to prevent experimenters from tuning their algorithms in a specific way, and, thus, outperform the other algorithms. In our opinion, both types of problem testbeds introduced at both mentioned conferences are too complex, and, therefore, the potential possibility of cheating is minimal. However, the problem could represent issuers of the new problem testbeds almost every year who could take advantage by raising their reputation.

The purpose of the paper is to show that the selection of the CEC problem testbed variants does not influence determining an algorithm's quality too crucially. This means that an algorithm achieving good results by solving the older CEC function suites should not be bad by solving the later function suites as well. To show this fact, a new measure was proposed for comparing the quality of algorithms by solving different benchmark suites. The experimental work was conducted using the new Differential Evolution (DE) algorithms for global optimization [7], [8] that introduce a cooperation game theory concept in the solving process [1]. The main advantage of these algorithms is that they have many parameters, which can be dependent on the benchmark used.

The game theory provides mathematical tools for interactive decision-making. Typically, several decision makers (also players) play the games and, thereby, have different goals, with which they affect the outcome of all the decision makers. The purpose of this theory is to predict the behavior of the players, and to support the decision makers with specific strategies that can enable them to win. Nowadays, the foundations of this theory are applied to various scientific domains, like

theoretical economics [9], [10], communication networks [11], political [12] and military sciences [13], [14], inspection games [15], and biology (e.g., survival of the fittest) [16]–[18].

The concept of marginal contributions of a player in a coalition game represents the basis of the new algorithms. This concept is necessary for calculating the Shapley value that is the well-known solution concept in coalition game theory. However, the introduction of this concept has the aim of manipulating a population's diversity. The population diversity is a prerequisite for the open-ended evolution [19], and enables stochastic population-based nature-inspired algorithms to operate in the dynamic conditions of the environment.

The structure of the paper is as follows. Section II deals with the basic information necessary for understanding topics in the remainder of the paper. In Section III, the proposed parallel EAs (PEAs) are discussed in detail. The experiments and results are the subjects of Section IV. Section V summarizes the performed work and outlines the directions for future work.

## II. Coalition game concept for global optimization

In multiagent interactions, we are confronted with the problem of how to divide a set of agents $Ag = \{1, \ldots, Np\}$ into subsets of cooperative agents $C = \{C_1, \ldots, C_n\}$ (also called coalitions) such that a division of agents into coalitions (also coalition structure formation) is stable, and a utility obtained by each coalition is shared between agents as fairly as possible. The formation of the coalition structure can be defined formally as a coalition game [20]:

$$G = \langle Ag, v \rangle, \tag{1}$$

where the characteristic function expressed as:

$$v : 2^{Ag} \rightarrow \mathcal{R} \tag{2}$$

assigns to each coalition its real value (also coalition value), in other words:

$$v(C_j) = k, \quad \text{for } j = 1, \ldots, n, \tag{3}$$

where $k$ denotes a utility shared between coalition members.

How the payoff received by the coalition can be shared reasonably between coalition members is only one side of the coin, i.e., the coalition's point of view. The other side refers to the agent's point of view, where each agent is interested in joining the coalition that provides the maximum payoff. However, the first problem is solved by the Shapley value, representing the second solution concept in the coalition game theory [18].

### A. Marginal contribution of agents to a coalition

Fair sharing of the coalition utility between players of a coalition game means that each player is paid according to the contribution he/she brings to a coalition. A Shapley value [21] is one of the fairest, average measures for determining these outcomes. The calculation of this value bases on determining a marginal contribution $\delta_i(C_j)$ obtained after joining agent $i$ to the coalition $C_j$ that is expressed as:

$$\delta_i(C_j) = v(C_j \cup \{i\}) - v(C_j), \quad \text{for } i = 1, \ldots, Np, \tag{4}$$

where $v(C_j \cup \{i\})$ represent the utility obtained after joining the agent $\{i\}$ to the coalition $C_j$, and $v(C_j)$ is the utility of coalition $C_j$.

In our study, the marginal contributions are considered from the agent's point of view as follows: Let us assume that the permutation $\pi \in \Pi(Np)$ is given. Then, each agent $i \in Np$ will be joined to those coalition $C_j$ that ensures the maximum payoff (i.e., the value of marginal contribution $\delta_i(C_j)$), as follows:

$$\forall i \in Np : \max \delta_i(C)\}, \quad \text{for } j = 1, \ldots, n. \tag{5}$$

In other words, each coalition consists of players who ensure the highest payoff for a specific coalition by joining it.

### B. Characteristic functions

The basic characteristic function $v(C)$ in our study measures the diversity of coalition $I(C)$ defined as:

$$I(C) = \sqrt{\sum_{j=1}^{D} (x_{k,j} - s_j)^2}, \quad \text{for } \forall k \in C, \tag{6}$$

where each agent $k$ is represented by a solution vector $\mathbf{x}_k = \{x_{k,j}\}$ for $j = 1, \ldots, D$ and vector $\mathbf{s} = \{s_j\}$ is the centroid, expressed as $s_j = \frac{1}{Np} \sum_{i=1}^{Np} x_{i,j}$. Additionally, two forms of Fitness Distance Correlations (FDC) are used as follows:

$$r(C) = \left(f(i) - \bar{f}\right) / I(C), \text{ and} \tag{7}$$

$$r'(C) = \left(f(i) - \bar{f}\right) \cdot I(C) / \left(s_F \cdot s_D\right), \tag{8}$$

where $f_i$ denotes the fitness, and $s_F$, $s_D$, and $\bar{f}$ are the standard deviations and fitness means, respectively.

Finally, the marginal contribution of agent $i$ to coalition $C$ is expressed as:

$$\delta_i(C) = g(C + \{i\}) - g(C), \quad \text{for } \forall C, \tag{9}$$

where $g(C) \in \{I(C), r(C), r'(C)\}$. Obviously, we are interested for those coalitions $C$, where the marginal contribution is the maximal. Thus, it is expected that the formed coalition could maintain the highest population diversity.

## III. The proposed parallel DE/jDE

The proposed PDE/jPDE algorithms operate with a population of agents $i = 1, \ldots, Np$ represented as vectors $\mathbf{x}_i$. These agents are joined to several coalitions, thus forming the stable coalition structure. Obviously, the agents are interested in joining those specific coalitions that increase their diversity the most. As a result, the coalition outcome for an agent is calculated as an increasing of the diversity of coalition by joining.

The pseudo-code of PEAs is illustrated in Algorithm 1 from which it can be seen that it consists of two phases:

**Algorithm 1** The proposed PEAs

```
1: procedure PEA(C, max_coal)
2:     n_coal = 0;
3:     while termination_condition_not_found do
4:         n_coal=FORM_COAL_STRUCT(C, max_coal, n_coal);
5:         for gen = 1 to max_eval do
6:             parfor all coal ∈ C do
7:                 EVOLVE(coal);
8:             end parfor all
9:         end for
10:    end while
11: end procedure
```

- forming the coalition structure (FORM_COAL_STRUCT function),
- evolving the formed coalitions (EVOLVE function).

Let us mention that the first phase is governed by the rules of the coalition game theory, while the second changes the monolithic EAs into Parallel EAs (PEAs), where the particular coalitions are evolved in parallel (**parfor** loop). In the last phase, the evolution of coalition members is governed by traditional DE/jDE algorithms.

The PEAs introduce four new parameters: $max\_coal$, $max\_eval$, $bias$, and $type$ of characteristic function. The first determines the maximum number of coalitions, the second the maximum number of generations permitted in one evaluation cycle, the third the percentage of coalition members that are preserved into the new generation, while the fourth is self-explanatory. All four parameters have a big impact on the performance of the algorithms. They are problem dependent and, therefore, their optimal settings need to be found experimentally.

## IV. Experiments and results

The goal of the experimental study was to show that the selection of CEC benchmark function suites (i.e., problem testbeds) does not have too crucial an impact on the estimation of the algorithm's quality. In line with this, two tests were conducted: (1) Determining the best parameter setting of the PEAs, and (2) Comparing the results of the best PEAs with the other well-known and state-of-the-art algorithms. However, the algorithms were run on two different CEC function benchmark suites (i.e., CEC-14 and CEC-18) in order to demonstrate the correctness of our hypothesis.

The parameters of the DE algorithm during tests were set as: $F = 0.9$ and $CR = 0.5$. The same values of parameters $F$ and $CR$ were used as starting values of the corresponding parameters $F_i^{(0)}$ and $CR_i^{(0)}$ for $i = 1, \ldots, Np$ by the jDE algorithm. However, the parameters of PEAs (i.e., $max\_coal$ and $max\_eval$) are unknown in advance. Therefore, their optimal values were determined during the extensive experimental work.

All algorithms in tests generated the results under the same conditions: They used the same population size $Np = 100$, and termination condition after the same fitness function evaluations $MAX\_FE = 10,000 \cdot D$. Let us mention that two different dimensions of both benchmark functions were taken into consideration, i.e., $D = 10$ and $D = 30$.

The results of the algorithms were evaluated according to five standard statistical measures: *Best*, *Worst*, *Mean*, *Median*, and *StDev* values. Then, the quality of obtained results were estimated using Friedman non-parametric tests [22]. The Friedman test is a two-way analysis of variances by ranks, where the statistic test is calculated and converted to ranks. The lower the value of the rank, the better the algorithm [23]. For estimating the results obtained by the same algorithms run on two different problem testbeds, the Pearson correlation coefficient was used that is defined as follows:

$$p_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \qquad (10)$$

where $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ are Friedman's ranks calculated after comparing the results of algorithms solving the CEC-14 and CEC-18 benchmark suites, respectively. The Pearson coefficient $p_{xy} \in [-1, 1]$ determines the relationships between two random variables $X$ and $Y$, where +1 means that a positive linear correlation exists between them, 0 that variables are not correlated, and -1 that the correlation is opposite. In our case, the high positive correlation shows that there is no difference between the results obtained by different problem testbeds.

In the continuation of the paper, the results of both tests were illustrated in detail.

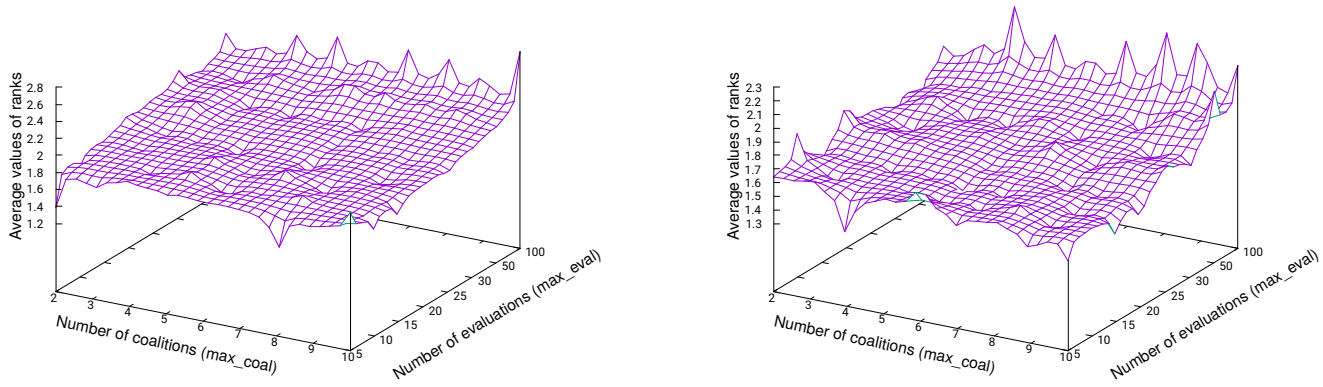
### A. Identifying the best parameter setting

In this test, the best PEAs were identified, obtained by solving different benchmark suites. In line with this, the optimal setting of parameters *max_coal*, *max_eval*, *bias*, and *type* of characteristic function were searched for, where the first was modified within the set $max\_eval \in \{5, 10, 15, 20, 25, 30, 50, 100\}$, the second within the interval $max\_coal \in [2, 10]$ in steps of one, the third within the interval $bias \in [0, 0.75]$ and the fourth has three options. Consequently, we even obtained $8 \times 9 \times 4 \times 3 \times 25 = 21,600$ independent runs per PEA. When two benchmarks are taken into consideration, the number of runs is increased twice (i.e., 43,200).

The average results of each specific algorithm according to five statistical measures obtained after optimizing the 30 functions[1] were composed into statistical classifiers of length $5 \times 30 = 150$. These classifiers were entered into Friedman non-parametric tests, from which the rank is calculated for each algorithm. Each rank of algorithm obtained by one combination of parameters $\langle max\_eval, max\_coal \rangle$ put into 3-dimensional space constitutes the so-called rank landscape with peaks and valleys, where the best values are identified as valleys.

The rank landscapes, where the best parameter setting for jPDE was found for both benchmark suites, are illustrated in Fig. 1, from which it can be seen that the rank landscape is

[1] Actually, the Function 2 is disabled in the CEC-17 benchmark and returns the result 0.




a: CEC-14 benchmark suite.

b: CEC-17 benchmark suite.

Fig. 1: Rank landscape obtained by jPDE optimizing benchmark functions of dimension $D = 30$.

more diverse in the CEC-17 benchmark. Actually, the best rank values were found by the parameter settings as presented in Table I.

TABLE I: The best parameter setting of PEAs.

| Benchmark | $max\_coal$ | $max\_eval$ | $bias$ | Char.fun. |
|---|---|---|---|---|
| CEC-14 | 5 | 8 | 0 % | $r'(C)$ |
| CEC-17 | 5 | 4 | 0 % | $r(C)$ |

### B. Comparing the best PEAs with the other algorithms

This test was devoted to justify our hypothesis. In line with this, the results of the best PEAs discovered in the last subsection were compared with the some well-known algorithms, like DE [7], self-adaptive jDE [8] and SaDE [24], and state-of-the-art algorithms, like jSO [25] and LSHADE [26]. The test was divided into two parts.

In the first part, four Friedman non-parametric tests were conducted with the best ranked PDE and jPDE algorithms using different characteristic functions according to four values of *bias* by solving CEC-14 and CEC-17 benchmark suites together with the other algorithms. Then, the test, where the best value of rank was found, was selected for conducting the Nemenyi post-hoc test [27]. Interestingly, the best results obtained by optimizing benchmark functions of dimension $D = 30$ were found by $bias = 0$ in both cases that are illustrated in Fig. 2, where the characteristic functions $I(C)$, $r(C)$, and $r'(C)$ are denoted as 1, 2, and 3.

As can be seen from the figure, no significant difference can be detected between results presented in diagrams 2.a and 2.b. This means that the results achieved by the jSO and LSHADE algorithms are significantly better than the results of all the other algorithms by solving both the problem testbeds. However, a fact that can surprise us is the relative ranking of both the exposed algorithms: The LSHADE is declared as the best algorithm in diagram 2.a, and the jSO as the second, while the situation in diagram 2.b shows the opposite.

In the second part, the results of all four tests obtained by solving both benchmark suites were composed into two classifiers that were used for calculation of the Pearson correlation coefficient. In summary, there were four calculations according to two different dimensions and two different lengths of classifiers, where the first considers only the results of PEA algorithms, while the second of all the algorithms in the tests. The obtained results are presented in Table II. The values of

TABLE II: Pearson correlation coefficients $p_{xy}$.

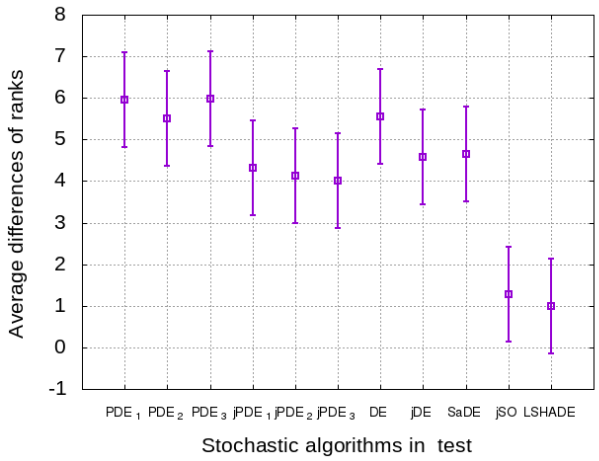
| PEAs | | All algorithms | |
|---|---|---|---|
| $D = 10$ | $D = 30$ | $D = 10$ | $D = 30$ |
| 0.6992 | 0.7669 | 0.9357 | 0.9799 |

Pearson coefficients show the high positive correlation (i.e., $p_{xy} \geq 0.7$). On the macro-level, we can say that there is not a significant difference between results obtained by solving the CEC-14 and CEC-18 benchmark function suites. However, on the micro-level, some differences can be detected in ranking the specific algorithms. However, could we declare that the jSO is better than the LSHADE in general?

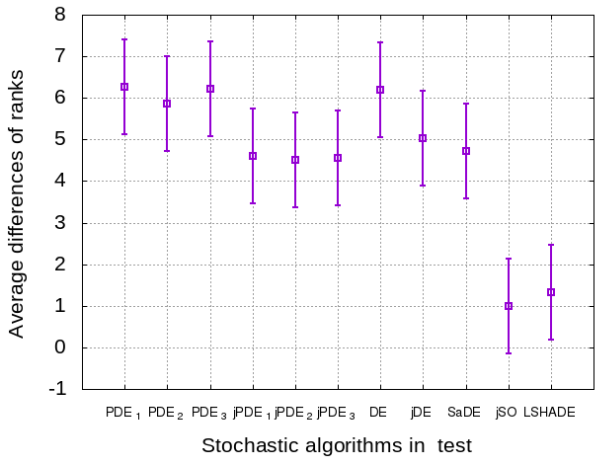
## V. Conclusion

The paper has arisen as the answer to reviewer comments as to why use the outdated CEC-14 benchmark function suite instead of the new CEC-18 by testing results of proposed PEAs introducing the concept of coalition game theory by preserving the diversity of coalitions. In line with this, a huge number of experiments were conducted solving both the problem testbeds, in which the best parameter setting of the PEAs are searched for, on the one hand, and determining how the different algorithms influence the results of optimizing both benchmark function suites, on the other.

The results of analysis were based on Friedman non-parametric tests. Ranks achieved as a result of the tests were




a: CEC-14 benchmark suite.




b: CEC-17 benchmark suite.

Fig. 2: Results of the Friedman non-parametric test by optimizing functions of dimension $D = 30$.

composed into statistical classifiers and served as input data for calculating the Pearson correlation coefficient. The Pearson coefficient shows how the results obtained by optimizing the CEC-14 benchmark function suite correlates with the results achieved by optimizing the CEC-18 suite.

The high positive values of the correlation coefficient obtained in our tests justify that selecting the function benchmark suite does not have crucial influence on the estimation of the algorithm's quality on the macro-level. On the micro-level, some differences in ranking the algorithms can arise, but this fact justifies the well-known No-free lunch theorem.

## ACKNOWLEDGMENT

I. Fister acknowledge the financial support from the Slovenian Research Agency (Research Core Founding No. P2-0041). I. Fister Jr. acknowledge the financial support from the Slovenian Research Agency (Research Core Founding No. P2-0057).

## REFERENCES

[1] I. Fister, A. Iglesias, A. Galvez, J. Del Ser, E. Osaba, and I. Fister Jr., "Cooperative game concepts in solving global optimization," in *Evolutionary Computation (CEC), 2019 IEEE Congress on*. Wellington, New Zealand: IEEE, 2019, pp. 1526–1533.

[2] M. Jamil and X. Yang, "A literature survey of benchmark functions for global optimization problems," *CoRR*, vol. abs/1308.4008, 2013.

[3] N. Hansen, T. Tusar, O. Mersmann, A. Auger, and D. Brockhoff, "COCO: The Experimental Procedure," *arXiv e-prints*, p. arXiv:1603.08776, Mar 2016.

[4] J. J. Liang, B.-Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, Tech. Rep., 12 2013.

[5] J. J. Liang, B. Y. Qu, P. N. Suganthan, and Q. Chen, "Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-based Real-Parameter Single Objective Optimization," Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, Tech. Rep., 11 2014.

[6] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization," Technical Report, Nanyang Technological University, Singapore, Tech. Rep., 11 2016.

[7] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec 1997.

[8] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, Dec 2006.

[9] T. Ichiishi, Ed., *Game Theory for Economic Analysis*, ser. Economic Theory, Econometrics, and Mathematical Economics. San Diego: Academic Press, 1983.

[10] C. H. Papadimitriou, "Game theory and mathematical economics: a theoretical computer scientist's introduction," in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, Oct 2001, pp. 4–8.

[11] J. Antoniou and A. Pitsillides, *Game Theory in Communication Networks: Cooperative Resolution of Interactive Networking Scenarios*. Boca Raton, Florida: CRC Press, 2012.

[12] N. McCarty and A. Meirowitz, *Political Game Theory: An Introduction*, ser. Analytical Methods for Social Research. Cambridge University Press, 2007.

[13] O. G. Haywood, "Military decision and game theory," *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 365–385, 1954. [Online]. Available: http://www.jstor.org/stable/166693

[14] L. Ordóñez, *Game Theory and the Decision-Making Process in Military Affairs*, ser. Contributions to Economics. Springer, Cham, 2017.

[15] R. Avenhaus, B. V. Stengel, and S. Zamir, "Inspection games," in *Handbook of Game Theory with Economic Applications*. Elsevier, 2002, vol. 3, pp. 1947–1987.

[16] K. Sigmund and M. A. Nowak, "Evolutionary game theory," *Current Biology*, vol. 9, no. 14, pp. R503–R505, 1999.

[17] T. Börgers, "Recent books on evolutionary game theory," *Journal of Economic Surveys*, vol. 15, no. 2, pp. 237–250, 2001.

[18] S. Durlauf and L. Blume, "Game theory and biology," in *Game Theory*, ser. The New Palgrave Economics Collection. London: Palgrave Macmillan, 2010.

[19] N. Packard, M. A. Bedau, A. Channon, T. Ikegami, S. Rasmussen, K. Stanley, and T. Taylor, "Open-ended evolution and open-endedness: Editorial introduction to the open-ended evolution i special issue," *Artificial Life*, vol. 25, no. 1, pp. 1–3, 2019, pMID: 30933628.

[20] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley Publishing, 2009.

[21] M. Maschler, E. Solan, and S. Zamir, *Game Theory*, 1st ed. Cambridge University Press, 2013.

[22] M. Friedman, "A comparison of alternative tests of significance for the problem of $m$ rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, 03 1940.
[23] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
[24] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *2005 IEEE Congress on Evolutionary Computation*, vol. 2, 2005, pp. 1785–1791 Vol. 2.
[25] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization : algorithm jSO," in *Proceedings of the 2017 IEEE Congress on Evolutionary Computation*, 2017, pp. 1311–1318.
[26] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, July 2014, pp. 1658–1665.
[27] P. Nemenyi, "Distribution-free multiple comparisons," Ph.D. dissertation, Princeton University, Princetown, NJ, 1963.