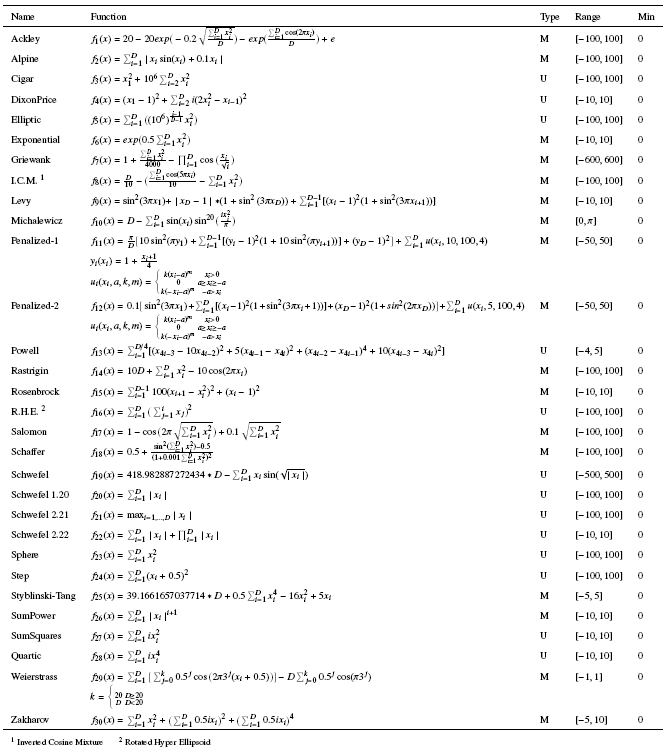
1. **Experimental Study and Discussion**

In this article, a comprehensive experimental study was conducted to test and verify the effect of the proposed FDB selection method. Experimental studies are presented under three sub-sections. Information about test problems, algorithms and parameter settings used in experimental studies are given in the following subsection (3.1. Benchmark functions and parameter settings). In the second subsection (3.2. Test Studies), the results obtained from the experimental studies in which the FDB method was tested are presented. For this purpose, variants of the SOS algorithm are obtained using various combinations of the FDB selection method. In order to determine the most effective FDB-SOS method, comparison studies between developed variants and the original SOS algorithm are performed. In the third subsection (3.3.Validation Studies), the performance of the FDB-SOS method was compared with the latest and most effective MHS algorithms in the literature. As a result of the comparisons, the success of the FDB-SOS method developed in this study was confirmed by experimental study results. Experimental studies were also conducted to examine the effect of FDB selection method on algorithm complexity. For this purpose, the complexity of the algorithm was compared between the SOS variants in which the FDB method was applied and the base SOS algorithm. In addition, the algorithm complexity of all competing algorithms used in experimental study was calculated. In the calculations, the problem dimensions are set to 30, 50 and 100. Besides, to determine whether the results obtained by FDB-based variants of SOS are significantly different from the results generated by the base SOS algorithm and other MHS methods, the nonparametric Wilcoxon rank-sum tests [84-86] have been executed. In Wilcoxon tests, the search performance of the algorithms on 90 test functions and 3 different problem dimensions are compared. To examine the exploitation and exploration abilities of the FDB-based method, convergence curves are given in different dimensions of the functions for the functions in the types of unimodal, multimodal, hybrite and composition.

* 1. **Experimantal settings**

*The set of benchmark functions used in test studies:* A set of classical benchmark problems was prepared to be used in the test studies in which the FDB-SOS method was developed. In this set consisting of the most frequently used problems in the literature, there are 30 unconstrained optimization problems. 17 of these benchmark problems are multi-modal types and others are uni-modal. Note that all the test functions are shown in Table 1, including their names, expressions, types, search space (ranges), and global optimum values respectively.

Table 1. Classical benchmark functions in experiments



*Benchmark test suites used in validation studies:* In order to compare the FDB-SOS method with the powerful and up-to-date MHS techniques in the literature and to verify the search performance, 90 benchmark problems were used. These are 30 classic benchmark problems (see the Table 1), CEC 2014 (30) [87] and CEC 2017 (30) [88] test suites, respectively. In the CEC 2014 test suite, there are 3 unimodal (*f1-f3*), 13 simple multimodal (*f4-f16*), 6 hybrid (*f17-f22*) and 8 composition (*f23-f30*) type functions. In the CEC 2017 test suite there are 3 unimodal (*f1-f3*), 7 simple multimodal (*f4-f10*), 10 hybrid (*f11-f20*) and 10 composition (*f20-f30*) type functions. All test functions are minimization problems defined in Eq. (17) [87]:

(17)

Hybrid functions are used to effectively test both convergence and diversity capabilities of algorithms. A hybrid function created by taking into account the real-world optimization problems. It is a set of multiple functions. In a set of hybrid functions, variables are randomly divided into some sub-components. Different basic functions are used for different sub-components. Hybrid functions are defined in Eq. (18) [87]:

………………………(18)

*F*(*x*) is the hybrid function and gi(*x*) is the ith basic function used to construct the hybrid function and *N* is the number of basic functions.

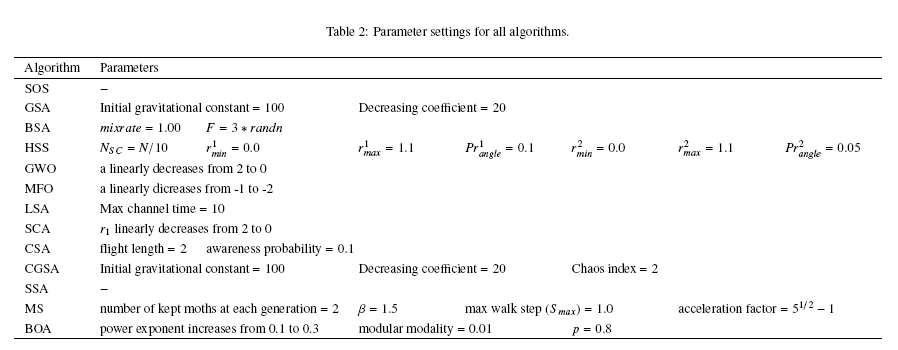
Composition functions are used to test the balance between the explotation and exploration ability of the algorithms. The composition functions have massive number of local optima, and are defined in Eq. (19) [87]:

……………………………………………….(19)

*F*(***x***): composition function, *g*i(***x***): *i*th basic function used to construct the composition function, *N*: number of basic functions, *biasi*: defines which optimum is global optimum, used to control each *g*i(***x***)’s height, : the normalized weight value for each *g*i(***x***).

*Dimensions of test functions and search range:* All of the test functions used in experimental studies are scalable as given in Eq. (17). In order to test how the performance of FDB changes with scaling the search space, we test all functions in 30, 50 and 100 dimensions. In this way, the effect of FDB selection method on low-, middle- and high-dimensional functions can be examined in detail. The search range for classical benchmark problems is given in the "range" column of Table 1. The search range for CEC 2014 [87] and CEC 2017 [88] test suites is taken as [-100, 100]D, as indicated in the description documents of these problems.

*Competing algorithms and parameter settings:* In experimental studies, 13 well-known MHS techniques were used to compare the search performance of competing algorithms. The abbreviations of these algorithms and the years they were developed are GSA (2009) [52], BSA (2013) [64], SOS (2014) [48], HSS (2014) [74], GWO (2014) [53], MFO (2015) [59], LSA (2015) [67], SCA (2016) [77], CSA (2016) [62], CGSA (2017) [58], SSA (2017) [21], MS (2018) [60], and BOA (2018) [63] respectively. In experimental studies, the parameter settings in the original papers of the competing algorithms were followed. Accordingly, the parameter settings of the algorithms are given in Table 2. All compared algorithms are implemented in [MATLAB](https://www.sciencedirect.com/topics/mathematics/matlab)®R2016b and run on an Intel(R) Core(TM) i7-4770K CPU @ 3.50 GHz and 16 GB RAM and x64-based processor.



*Termination criteria and statistical analysis:* To make a fair comparison, the parameters of all competing algorithms are set to be the same as in their original papers. Furthermore, the maximum number of function evaluations (*maxFEs*) is used as the termination condition, which was set to 1000\**D.* Convergence to the global optimum point to 10-8 is also accepted to solve problems [88].The mean and the standard deviation of the best-of-run errors for 51 independent runs of all of the contestant algorithms on the benchmark problems for D = 30, D = 50 and D = 100 are given in Tables 4-6 and Tables 9-17. The error is the [absolute value](https://www.sciencedirect.com/topics/mathematics/absolute-value) of the difference between the real [optimum value](https://www.sciencedirect.com/topics/mathematics/optimum-value) of the [objective function](https://www.sciencedirect.com/topics/engineering/objective-function) (*f*opt) and the best result *f* (*Pbest*). In order to test the statistical validity of the results obtained from test and validation studies, non-parametric pair wise Wilcoxon test has been applied. The test has been conducted for 5% level of significance. The results are presented in Tables “+”, “=”, and “-” denote that the performance of the corresponding algorithm is better than, similar and worse than to that of FDB-SOS, respectively. We also show the total number of the aforementioned cases at the end of the Tables 4-6 and Tables 9-17 of each dimension, for each of the competitor algorithms as (+/=/–).