Tournament Selection Based Probability Scheme in Spider Monkey Optimization Algorithm

Kavita Gupta and Kusum Deep

Abstract In this paper, a modified version of Spider Monkey Optimization (SMO) algorithm is proposed. This modified version is named as Tournament selection based Spider Monkey Optimization (TS-SMO). TS-SMO replaces the fitness proportionate probability scheme of SMO with tournament selection based probability scheme with an objective to improve the exploration ability of SMO by avoiding premature convergence. The performance of the proposed variant is tested over a large benchmark set of 46 unconstrained benchmark problems of varying complexities broadly classified into two categories: scalable and non-scalable problems. The performance of TS-SO is compared with that of SMO. Results for scalable and non-scalable problems have been analysed separately. A statistical test is employed to access the significance of improvement in results. Numerical and statistical results show that the proposed modification has a positive impact on the performance of original SMO in terms of reliability, efficiency and accuracy.

Keywords Spider monkey optimization · Tournament selection · Unconstrained optimization · Swarm intelligent techniques

1 Introduction

Spider Monkey Optimization (SMO) technique [1] is a newly developed swarm intelligent technique for solving unconstrained real parameter optimization problems. It is a simple and easy to implement swarm intelligent technique with few control parameters. It is inspired from the food searching strategy of Spider monkeys. Spider monkeys belong to the class of fission-fusion social structure

K. Gupta(✉) · K. Deep
Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee 247667, Uttarakhand, India
e-mail: {gupta.kavita3043,kusumdeep}@gmail.com

J.H. Kim and Z.W. Geem (eds.), *Harmony Search Algorithm*,
Advances in Intelligent Systems and Computing 382,
DOI: 10.1007/978-3-662-47926-1_23

based animals which live in a group of 40-50 individuals [4]. The group leader divides the whole group into small subgroups and the members of these subgroups search for their food in different directions. There are mainly two manipulation phases in SMO in each iteration which are responsible for the updation of the swarm. In one of these phases, members of the swarm get chance to update their position based on their probability. This probability is fitness proportionate, which is similar to roulette wheel selection in Genetic Algorithm (GA) [2,3]. Due to the use of fitness proportionate probability scheme in SMO, members of the swarm having higher fitness have better chances of updating their position as compared to the ones having lower fitness value. But sometimes even less fit members may contain some important information which can be very useful, but being not given a chance due to their low fitness, this important information is lost. Aiming at this limitation of SMO, we propose a variation of original SMO in which tournament selection based probability scheme has been used instead of fitness proportionate probability scheme of SMO. Tournament selection is one of the famous operators used in selection phase of GA [5]. Tournament selection based scheme will provide a chance to even less fit individuals to update their position. This modification at improving the search ability and convergence speed of original SMO by favoring more exploration with the help of probability scheme based on tournament selection. To the best of our knowledge, this is the first attempt of using tournament selection for calculating probabilities in SMO. Tournament of size two has been used in the proposed method. The objective of this paper is to study the impact on the performance of SMO in terms of reliability and convergence speed after replacing fitness proportionate probability scheme with tournament selection based probability scheme which is meant to increase the exploration ability of SMO.

The paper is organized as follows. Section 2 gives a brief introduction to Spider monkey optimization technique. In section 3, proposed modification in the original SMO has been discussed. Section 4 deals with experimental settings followed by discussion of experimental results in section 5. Section 6 provides the conclusion and future scope of proposed work.

2 Spider Monkey Optimization Algorithm

Like all other population based algorithms, SMO follows some iterative steps in the process of improving the swarm of randomly generated solutions. In addition to the initialization of the swarm, SMO follows six iterative steps. These are: Local leader phase, global leader phase, local leader learning phase, global leader learning phase, local leader decision phase and global leader decision phase. Detailed description of each iterative step along with their purpose in the algorithm can be found in [1]. Local leader limit, global leader limit, perturbation rate and maximum number of groups are four control parameters of SMO. Pseudocode for SMO has been provided in Fig. 1.

```
begin:
    Initialize the swarm
    Initialize Local leader limit, global leader limit, perturbation rate, maxi-
mum number
    of groups
    Iteration=0
    Calculate fitness value of the position of each spider monkey in the swarm
    Select Global Leader and Local Leaders by applying Greedy Selection
    while (termination criterion is not satisfied) do
            //Local Leader Phase
           //Calculate Probabilities
          //Global Leader Phase
          //Global Leader Learning Phase
          //Local Leader Learning Phase
          //Local Leader Decision Phase
          //Global Leader Decision Phase
            Iteration = iteration +1
     end while
  end
```

Fig. 1 Pseudocode for SMO

3 Tournament Selection Based SMO

TS-SMO is just a variation of SMO replacing fitness proportionate probability scheme used in SMO with tournament selection based probability scheme. The parameter associated with the tournament selection operator is the size of the tournament. This size indicates the numbers of members which will participate in the tournament. In this paper, tournament size is two. In global leader phase, members of the swarm get a chance to update their position based on their probability. Fitness proportionate probability scheme used in SMO provides more chances to highly fit members to make themselves better which sometimes may lead to premature convergence because of attraction of the swarm to highly fit individuals only. Tournament selection based probability scheme facilitates diversity in the population thus avoiding premature convergence. Also, it may happen that even less fit individuals may contain some important information about the optimal solution. But since they do not have high probability, they have very less chances of updating their position. To avoid the loss of important information contained in less fit members of the swarm, it has been decided to use tournament selection based probability scheme in place of fitness proportionate probability scheme so that even less fit individuals may get chance to update their position. Let prob[i] and fit[i] be the probability and fitness respectively of the ith member of the swarm. Pseudocode for calculating probability in original SMO and TS-SMO has been provided in Fig. 2 and Fig. 3 respectively.

```
for i: 1 to n
prob[i] = 0.9 × (fit[i]/maxfit) + 0.1
end for
```

$$prob[i] = 0.9 \times \left(\frac{fit[i]}{maxfit}\right) + 0.1$$

Fig. 2 Pseudocode for calculating probability in SMO

```
for i: 1 to n
 a[i]=0
  for j:1 to n
    if (fit[i]>=fit[j])
      a[i]=a[i]+1
    end if
   end for
 end for
 for i: 1 to n
 prob[i] = a[i] / Σ_{i=1}^{n} a[i]
 end for
```

$$prob[i] = \frac{a[i]}{\sum_{i=1}^{n} a[i]}$$

Fig. 3 Pseudocode for calculating probability in TS-SMO

4 Experimental Setup

The performance of SMO and TS-SMO has been tested over a benchmark set of 46 (1-30 are scalable problems and 31-46 are non-scalable problems) unconstrained optimization problems. All the scalable problems are of dimension 30 and dimension of each non-scalable problem is mentioned in the list of test problems given in the appendix. Below is the parameter setting and termination criterion for the experiment.

Swarm size =150
Perturbation rate (pr) = linearly increasing ([0.1, 0.4])
Maximum number of groups (MG) =5
Local leader limit =100
Global leader limit=50
Total number of runs =100
Maximum number of iteration= 4000
acceptable error =1.0e-05
Stopping criterion = either maximum number of iterations are performed or acceptable error is achieved.

Here error is the absolute difference between the optimal solution and objective function value of the global leader. In order to make a fair comparison between the two algorithms, both the algorithms starts with the same initial swarm..

5 Experimental Results and Discussion

In order to compare the performance of both the algorithms, number of successful runs and the average number of function evaluations of successful runs have been recorded. A run is said to be successful if the error value is less than the acceptable error. Reliability of the algorithms has been measured from the number of successful runs and efficiency is measured with the number of function evaluations of successful runs. Comparison between the two algorithms have been made in the following manner:

First the number of successful runs have been checked and the algorithm with more number of successful runs will be the winner. If both the algorithms have same number of successful runs, then their average number of function evaluations for successful runs have been checked and the algorithm with less number of function evaluations will be the winner. Further, to see if there is really any significant difference between the function evaluations of successful runs, one tailed t-test at a significant level of 0.5 has been employed. "=" indicates there is no significant difference between the average of function evaluations of two algorithms and "+" and "-" indicates that TS-SMO performs significantly better and worse than SMO respectively. T-test is only applied to the problems where number of successful runs for both the algorithms are same. The results of scalable and non-scalable problems have been provided in table 1 and table 2 respectively.

From Table 1, it can be observed that out of 30 scalable problems, there are 5 problems where both the algorithms do not have even a single successful run. From the rest of 25 problems, there are 21 problems were TS-SMO performs better than SMO. Also, from the results of t-test, significant difference between the function evaluations can be observed.

From Table 2, it is observed that out of 16 non-scalable problems, there are 3 problems where both the algorithms have no successful runs. From the remaining 13 problems, there are 7 problems where SMO performs better than TS-SMO, while there are 6 problems where TS-SMO performs better. Also, t-test results show significant difference in the number of function evaluations.

From the results depicted in Table 1 and Table 2, it can be concluded that whereas TS-SMO performs better than SMO on most of the scalable problems, its performance is not so good on non-scalable problems.

In order to observe the effect of the variation in the objective function value as the iterations proceed Convergence graphs are plotted. Convergence graph of selected problems have been provided in Fig. 4, Fig. 5, Fig. 6 and Fig. 7. From these graphs it can be seen that TS-SMO convergence faster than SMO in most of the problems.

Table 1 No. of successful runs and average number of function evaluations of successful runs for SMO and TS-SMO (Scalable problems)

Fun_no.	Percentage of Success		Function evaluations of successful runs		Outcome of t-test
	SMO	TS-SMO	SMO	TS-SMO	
1	100	100	33192	**32128**	+
2	100	100	26566	**25923**	+
3	100	100	**103625**	108601	=
4	0	0	0	0	N.A.
5	100	100	**229495**	254951	-
6	100	100	63917	**61660**	+
7	100	100	189827	**149630**	+
8	100	100	24027	**19040**	+
9	100	100	37701	**34987**	+
10	100	100	25421	**24404**	+
11	47	**61**	**650570**	682431	N.A.
12	100	100	58502	**56807**	+
13	100	100	32220	**31231**	+
14	100	100	62664	**60888**	+
15	0	0	0	0	N.A.
16	100	100	38287	**37013**	+
17	0	0	0	0	N.A.
18	100	100	14002	**13294**	+
19	100	100	22628	**22437**	+
20	0	0	0	0	N.A.
21	**7**	2	**1051267**	1117825	N.A.
22	83	**85**	354061	**349233**	N.A.
23	100	100	48381	**47167**	+
24	100	100	30839	**29457**	+
25	100	100	30695	**29586**	+
26	100	100	39151	**37892**	+
27	100	100	44665	**43394**	+
28	0	0	0	0	N.A.
29	100	100	106373	**101093**	=
30	100	100	63980	**61938**	+

Table 2 No. of successful runs and average number of function evaluations of successful runs for SMO and TS-SMO for Non-Scalable problems

Fun_no.	Percentage of Success		Function evaluations of successful runs		outcome of t-test
	SMO	TS-SMO	SMO	TS-SMO	
31	100	100	3738	**3636**	=
32	**98**	93	**430014**	465326	N.A.
33	100	100	**18801**	19552	-
34	100	100	3585	**3415**	+
35	100	100	256584	**211382**	+
36	100	100	3304	**3199**	=
37	100	100	131394	**131171**	=
38	100	100	**26119**	33359	-
39	100	100	**2009**	2281	-
40	100	100	3071	**2793**	+
41	0	0	0	0	N.A.
42	100	100	**24508**	45553	-
43	0	0	0	0	N.A.
44	100	100	**13458**	13519	=
45	0	0	0	0	N.A.
46	100	100	**10737**	12525	-

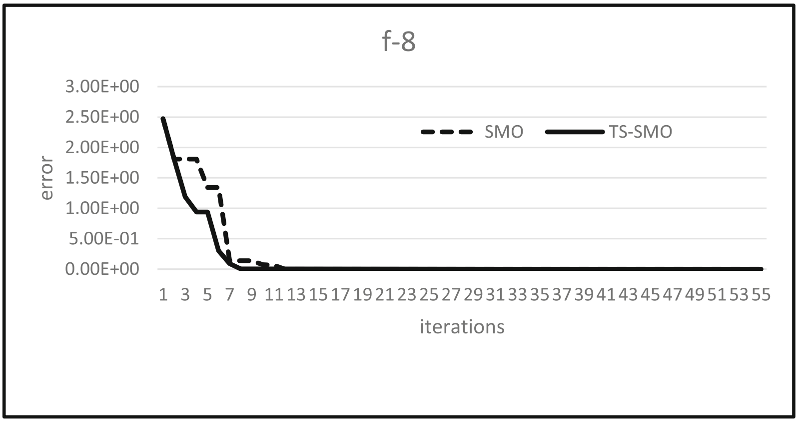

Fig. 4 Convergence graph of Function No. 8

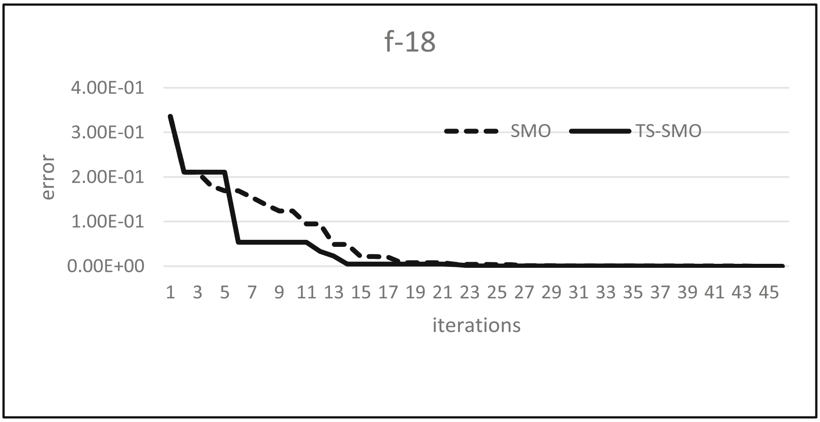

Fig. 5 Convergence graph of Function No. 18

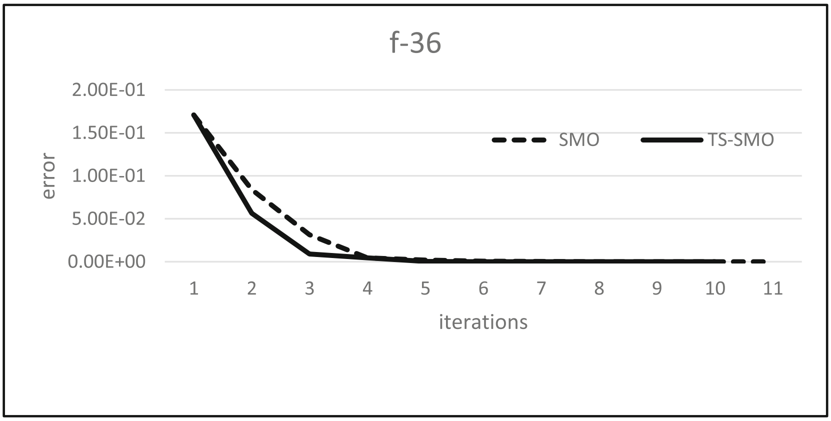

Fig. 6 convergence graph of Function No. 36

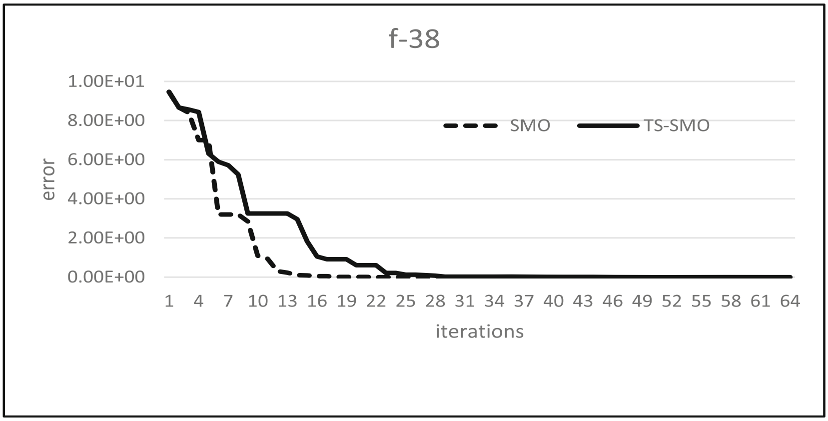

Fig. 7 Convergence graph of Function No. 38

6 Conclusion and Future Directions

In this paper, a new version of SMO abbreviated as TS-SMO has been proposed. The novelty of this new version lies in the use of tournament selection based probability scheme instead of fitness proportionate probability scheme which has been used in original SMO. Maintaining a high level of diversity while preserving convergence speed are two contradictory and necessary features of a metaheuristic technique and the numerical and statistical results show that whereas TS-SMO did a good job in balancing both the features by performing well on scalable problems, it could not perform so well on non-scalable problems. But since the experiments have been performed over a limited number of benchmark problems and no theoretical proof has been provided, making any concrete conclusion will not be justified. Further work needs to be done experimentally and theoretically to make any strong judgement about the superiority of TS-SMO over SMO.

Acknowledgement The first author would like to acknowledge the ministry MHRD, Government of India for financial support.

References

1. Bansal, J.C., Sharma, H., Jadon, S.S., Clerc, M.: Spider Monkey Optimization algorithm for numerical optimization. Memetic Computing **6**(1), 31–47 (2014)
2. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading (1989)
3. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan press, AnnArbor (1975)
4. Symington, M.M.: Fission-fusion social organization inAteles and Pan. International Journal of Primatology **11**(1), 47–61 (1990)
5. Miller, B.L., Goldberg, D.E.: Genetic algorithms, tournament selection, and the effects of noise. Complex Systems **9**(3), 193–212 (1995)

Appendix

Table A1 Test problems

Test Problems	Objective function	Search range	Optimal value
Sphere Function	$f_1(x)=\sum_{i=1}^{D} x_i^2$	[-5.12,5.12]	0
De Jong's F4	$f_2(x)=\sum_{i=1}^{D} i x_i^4$	[-5.12,5.12]	0
Griewank	$f_3(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600,600]	0
Rosenbrock	$f_4(x)=\sum_{i=1}^{D}[100(x_i^2-x_{i+1})^2+(x_i-1)^2]$	[-100,100]	0
Rastrigin	$f_5(x)=\sum_{i=1}^{D}(x_i^2-10cos(2\pi x_i)+10)$	[-5.12,5.12]	0
Ackley	$f_6(x)=-20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)$ $-exp\left(\frac{1}{D}\sum_{i=1}^{D}cos(2\pi x_i)\right)+20+e$	[-30,30]	0
Alpine	$f_7(x)=\sum_{i=1}^{D}\lvert x_i sin(x_i)+0.1x_i\rvert$	[-10,10]	0
Michalewicz	$f_8(x)=-\sum_{i=1}^{D} sin(x_i)\left[\frac{sin(ix_i^2)}{\pi}\right]^{20}$	[0,π]	-9.66015
Cosine Mixture	$f_9(x)=\sum_{i=1}^{D}x_i^2-0.1\sum_{i=1}^{D}\cos(5\pi x_i)$	[-1,1]	-D*0.1
Exponential	$f_{10}(x)=-exp\left(-0.5\sum_{i=1}^{D}x_i^2\right)$	[-1,1]	-1
Zakharov	$f_{11}(x)=\sum_{i=1}^{D}x_i^2+\left(\frac{1}{2}\sum_{i=1}^{D}ix_i\right)^2+\left(\frac{1}{2}\sum_{i=1}^{D}ix_i\right)^4$	[-5.12,5.12]	0
Cigar	$f_{12}(x)=x_1^2+100000\sum_{i=2}^{D}x_i^2$	[-10,10]	0
Brown3	$f_{13}(x)=\sum_{i=1}^{D-1}\left[(x_i^2)^{(x_{i+1}^2+1)}+(x_{i+1}^2)^{(x_i^2+1)}\right]$	[-1,4]	0
Schewel Prob 3	$f_{14}(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	[-10,10]	0
Salomon Problem	$f_{15}(x)=1-cos\left(2\pi\sqrt{\sum_{i=1}^{D}x_i^2}\right)+0.1\sqrt{\sum_{i=1}^{D}x_i^2}$	[-100,100]	0
Axis Parallel Hyperellipsoid	$f_{16}(x)=\sum_{i=1}^{D}ix_i^2$	[-5.12,5.12]	0
Pathological	$f_{17}(x)=\sum_{i=1}^{D-1}\left[0.5+\frac{sin^2\sqrt{100x_i^2+x_{i+1}^2}-0.5}{1+0.001({x_i}^2+x_{i+1}^2-2x_ix_{i+1})^2}\right]$	[-100,100]	0
Sum Of Different Powers	$f_{18}(x)=\sum_{i=1}^{D}\lvert x_i\rvert^i$	[-1,1]	0

Table A1 (*continued*)

Step Function	$f_{19}(x)=\sum_{i=1}^{D}(\lfloor x_i+0.5\rfloor)^2$	[-100,100]	0
Quartic Function	$f_{20}(x)=\sum_{i=1}^{D} ix_i^4+random[0,1)$	[-1.28,1.28]	0
Inverted Cosine Wave Function	$f_{21}(x)=-\sum_{i=1}^{D-1} exp\left(-\left(\frac{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}}{8}\right)\right)*$ $\cos\left(4\sqrt{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}}\right)$	[-5,5]	-D+1
Neumaier3 Problem	$f_{22}(x)=\left\|\sum_{i=1}^{D}(x_i-1)^2-\sum_{i=1}^{D} x_i\, x_{i+1}\right\|$	[-900, 900]	0
Rotated Hyper Ellipsoid Function	$f_{23}(x)=\sum_{i=1}^{D} x_i^2$	[-65.536, 65.536]	0
Levi Montalvo 1	$f_{24}(x)=\frac{\pi}{D}[10sin^2(\pi y_1)+$ $\sum_{i=1}^{D-1}(y_i-1)^2\left(1+10sin^2(\pi y_{i+1})\right)+(y_D-1)^2]$, Where $y_i=1+\frac{1}{4}(x_i+1)$	[-10,10]	0
Levi Montalvo 2	$f_{25}(x)=0.1\left(sin^2(3\pi x_1)+\sum_{i=1}^{D-1}\left[(x_i-1)^2\left(1+sin^2(3\pi x_{i+1})\right)\right]\right)$ $+(x_D-1)^2\left(1+sin^2(2\pi x_D)\right)$	[-5,5]	0
Ellipsoidal	$f_{26}(x)=\sum_{i=1}^{D}(x_i-i)^2$	[-D,D]	0
Shifted Parabola CEC 2005	$f_{27}(x)=\sum_{i=1}^{D} z_i^2+f_{bias}$, z=(x-o), x=$[x_1,x_2,\dots,x_D]$, O=$[o_1,o_2,\dots,o_D]$	[-100,100]	-450
Shifted Schwefel CEC 2005	$f_{28}(x)=\sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2+f_{bias}$, z=(x-o), x=$[x_1,x_2,\dots,x_D]$, O=$[o_1,o_2,\dots,o_D]$	[-100,100]	-450
Shifted Greiwank CEC 2005	$f_{29}(x)=\sum_{i=1}^{D}\frac{z_i^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right)+1+f_{bias}$, z=(x-o), x=$[x_1,x_2,\dots,x_D]$, O=$[o_1,o_2,\dots,o_D]$	[-600,600]	-180
Shifted Ackley CEC 2005	$f_{30}(x)=-20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)$ $-exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right)+20+e+f_{bias}$, z=(x-o), x=$[x_1,x_2,\dots,x_D]$, O=$[o_1,o_2,\dots,o_D]$	[-32,32]	-140

Table A1 (*continued*)

Step Function	$f_{19}(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	[-100,100]	0
Quartic Function	$f_{20}(x) = \sum_{i=1}^{D} i x_i^4 + random[0,1)$	[-1.28,1.28]	0
Inverted Cosine Wave Function	$f_{21}(x) = -\sum_{i=1}^{D-1} exp\left(-\left(\frac{x_i^2 + x_{i+1}^2 + 0.5 x_i x_{i+1}}{8}\right)\right) * \cos\left(4\sqrt{x_i^2 + x_{i+1}^2 + 0.5 x_i x_{i+1}}\right)$	[-5,5]	-D+1
Neumaier3 Problem	$f_{22}(x) = \left\lvert \sum_{i=1}^{D} (x_i - 1)^2 - \sum_{i=1}^{D} x_i x_{i+1} \right\rvert$	[-900, 900]	0
Rotated Hyper Ellipsoid Function	$f_{23}(x) = \sum_{i=1}^{D} x_i^2$	[-65.536, 65.536]	0
Levi Montalvo 1	$f_{24}(x) = \frac{\pi}{D}[10 sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2(1 + 10 sin^2(\pi y_{i+1})) + (y_D - 1)^2]$, Where $y_i = 1 + \frac{1}{4}(x_i + 1)$	[-10,10]	0
Levi Montalvo 2	$f_{25}(x) = 0.1\left(sin^2(3\pi x_1) + \sum_{i=1}^{D-1}\left[(x_i - 1)^2(1 + sin^2(3\pi x_{i+1}))\right]\right) + (x_D - 1)^2(1 + sin^2(2\pi x_D))$	[-5,5]	0
Ellipsoidal	$f_{26}(x) = \sum_{i=1}^{D} (x_i - i)^2$	[-D,D]	0
Shifted Parabola CEC 2005	$f_{27}(x) = \sum_{i=1}^{D} z_i^2 + f_{bias}$, z=(x-o), x=$[x_1, x_2, \ldots, x_D]$, O=$[o_1, o_2, \ldots, o_D]$	[-100,100]	-450
Shifted Schwefel CEC 2005	$f_{28}(x) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2 + f_{bias}$, z=(x-o), x=$[x_1, x_2, \ldots, x_D]$, O=$[o_1, o_2, \ldots, o_D]$	[-100,100]	-450
Shifted Greiwank CEC 2005	$f_{29}(x) = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{bias}$, z=(x-o), x=$[x_1, x_2, \ldots, x_D]$, O=$[o_1, o_2, \ldots, o_D]$	[-600,600]	-180
Shifted Ackley CEC 2005	$f_{30}(x) = -20 exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D} cos(2\pi x_i)\right) + 20 + e + f_{bias}$, z=(x-o), x=$[x_1, x_2, \ldots, x_D]$, O=$[o_1, o_2, \ldots, o_D]$	[-32,32]	-140

Shekel10	$f_{44}(x) = -\sum_{j=1}^{10}\left[\sum_{i=1}^{4}(x_i - C_{ij})^2 + \beta_j\right]^{-1}$	[0,10]	4	-10.5364
Dekkers and Aarts	$f_{45}(x) = 10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4$	[-20,20]	2	-24777
Shubert	$f_{46}(x) = -\sum_{i=1}^{5} i cos((i + 1)x_1 + 1) \sum_{i=1}^{5} i cos((i + 1)x_{2+1})$	[-10,10]	2	-186.7309