Elektrik Elektronik Mühendisliği / *Electrical Electronic Engineering* **DOI: 10.21597/jist.722427**
Araştırma Makalesi / *Research Article*
Iğdır Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 10(4): 2451-2460, 2020
Journal of the Institute of Science and Technology, 10(4): 2451-2460, 2020
ISSN: 2146-0574, eISSN: 2536-4618

Performance Analysis of SMDO Method with Benchmark Functions with Matlab Toolbox

Mehmet AKPAMUKÇU[1], Abdullah ATEŞ[1*], Barış Baykant ALAGÖZ[1]

ABSTRACT: SMDO method is a set and trial based optimization algorithm that is developed for online fine-tuning of controller parameters. SMDO method is implemented for several controller tuning applications. It can search parameter space with random backward and forward steps of each parameter. This property reduces risk of testing unstable control system configurations in controller design and thus makes the SMDO method more suitable for online parameter tuning of experimental systems. However, performance of SMDO has not been evaluated previously for benchmark functions in comparison with other well known heuristic optimization methods. This study aims to compare performances of Artificial Bee Colony (ABC), Cuckoo Search Optimization (CK), Particle Swarm Optimization (PSO) and Stochastic Multi-parameters Divergence Optimization (SMDO) methods for benchmark functions. Therefore, a benchmark tests program that is a user-friendly MATLAB GUI is introduced for user. This program can be downloaded from https://www.mathworks.com/matlabcentral/fileexchange/75043-smdo-method-with-benchmark-functions

Keywords: SMDO, optimization, stochastic, benchmark functions

[1] Mehmet AKPAMUKÇU (**Orcid ID:** 0000-0002-3763-5048), Abdullah ATEŞ (**Orcid ID:** 0000-0002-4236-6794), Barış Baykant ALAGÖZ (**Orcid ID:** 0000-0001-5238-6433)), Inonu University, Faculty of Engineering, Computer Engineering Department, Malatya, Turkey

*Sorumlu Yazar/Corresponding Author: Abdullah ATEŞ, abdullah.ates@inonu.edu.tr

This study has been produced from the studies conducted for Mehmet AKPAMUKÇU's Master's thesis.

Geliş tarihi / *Received:* 17-04-2020
Kabul tarihi / *Accepted:* 25-06-2020

INTRODUCTION

There are two main methods while searching solutions for optimization related problems. These are analytic and numerical methods (Frédéric Bonnans et al., 2006; Nocedal and Wright, 2006) . It is known that analytic methods provide exact solutions (Kuhn and Tucker, 2014; Vajda and Dantzig, 1965). These methods supply the power of mathematical instruments to solve the problems. At this point, it must be known that the dynamics of the problem must match with the analytical method which will be used. For instance; the number of the parameters and the number of the equations must be suitable for analytical optimization method. However, the real world engineering problems may not well fit to structure of pure analytic methods. Because sometimes the known parameters are inadequate or there can be distortions which change the dynamics of the problem. It is known that numerical methods are very efficient in this situation (Kakandikar et al., 2018; Mirjalili et al., 2017; Rao, 2009; Yang & Gandomi, 2012). This can be sometimes imitating the behaviors of the animals or sometimes it can be imitating the world of the physics and etc. Solution finding process is started in the solution space with the guidance of the numerical method (Biswas et al., 2013; Chopard & Tomassini, 2018; Emmerich & Deutz, 2018; Yang, 2014). Searching sometimes tries to minimize an error function and sometimes tries to maximize the profit function and sometimes multi-objective structures where these two structures are hybridized are used (Giagkiozis and Fleming, 2015). Sometimes it is fallen into trap in this search which is in this case to fall into local minima. At this point the power of the numerical method is clear. On the other hand, even the most powerful numerical methods do not guarantee the exact solution. Searching generally takes place around the exact solution. A solution which is small epsilon proximity is mostly adequate in the final. At the result it is not wanted to reach local minima solution. An effort is made to find global minimum and global maximum. By the way, all numerical methods also do not match with the dynamics of the optimization problem that is tried to be solved. So the method must be selected meticulously.

The SMDO method, which is one of the numerical methods, is in our focus to analyze in this paper. It is a useful algorithm, adequacy of which is proven in real engineering problems. The details on the algorithm are discussed in the section of A Brief Introduction of Stochastic Multi Parameter Divergence Optimization Method. The SMDO algorithm has been used especially in online controller tuning problems. Several online and offline applications of SMDO in tuning of fractional order and integer order controllers have been shown in the literature. The most important feature of this algorithm is the tuning of controller coefficients with a reduced probability of testing an unstable control system, therefore it is useful for the online tuning of experimental systems as well as the offline tuning of simulation models. The SMDO method is used in several studies, that are, a PID controller is designed according to fractional order reference model in (Alagoz et al., 2013), a fractional order controller is designed online for experimental TRMS system (Yeroğlu and Ateş, 2014), optimize controller parameters according to reference model for fractional order PID controller in (Ateş, A. Alagoz, 2013), design fractional controller for two degree of freedom systems with SMDO algorithm (Ateş, A. Alagoz, 2013; Ateş and Yeroglu, n.d.), design of fractional order controller according to the master slave structure in (Abdullah Ates et al., 2014), using fractional order two degrees of the freedom structure for a real-time system in (Ates, A and Yerolgu C, 2016), modeling of the Receptor-Ligand Complexes structure using a fractional order circuit approach using the SMDO algorithm in (Abdullah Ates et al., 2019), designing fractional order controller using different distribution functions in (A. Ates et al., 2020).

Effective applications of SMDO method have been especially shown for PID and fractional order PID controllers desing works. Advantage of SMDO method for these applications comes from its property that allows a cautious progress in parameter search space when searching minima. It progresses one parameter each time, and in case of worsening controller performance, it promptly takes back to the best performance. Via this cautious searching, real systems can not be driven into extreme points that may lead to instability of system and especially the damage of the experimental system. Other search algorithms do not generally be cautious about testing undesired points similar to SMDO. In general, majority of the search algorithms such as swarm based search methods do not purposely designed for real world tuning applications, they are designed for quickly reaching minima without worrying about the risk of undesired tests and developed by considering high performance on the benchmark functions. This success does not guarantee that these algorithms will be appropriate to implement in a real world engineering problem. Although SMDO is developed for controller tuning problems, its performance for benchmark functions is not evaluated. This point is the main motivation of this paper. A performance comparison of SMDO method with other popular methods will be useful to demonstrate applicability of SMDO method as general purpose search method.

In this paper, we introduce the benchmark functions that are used frequently in academic studies. With these functions, performance analysis and comparison of the heuristic search algorithms can be performed. In this study, the benchmark functions were resolved under the same conditions and the analyses were carried out fairly. On the other hand, the benchmark functions presented below can be easily selected from the Matlab GUI, which was developed for testing of SMDO method. This GUI is available on the Mathworks site (https://www.mathworks.com/matlabcentral/fileexchange/75043-smdo-method-with-benchmark-functions) for download and freely use.

The rest of the paper is organized as follows; Section Benchmark Functions presents the benchmark functions that we will use in this analysis. In Section Analyzing of Stochastic Multi Parameter Divergence Optimization Method; SMDO method is briefly explained and besides, its usage details are explained. In section Comparisons with Diffrerent Methods Using Different Benchmark Functions, the solutions of different methods with different benchmark functions are compared. These different methods are found in the literature and are brought together to make comparisons. These comparisons constitute the backbone of this paper. And finally conclusion is presented.

MATERIALS AND METHODS

Benchmark functions

The benchmark functions that are used in this paper are listed in Table 1(*Benchmark Function*, 2020). These functions are mostly used benchmark functions in academic studies. With these functions, performance analysis and comparisons of search algorithms can be carried out before using them in engineering problems. The determined benchmark functions were resolved under the same conditions and comparisions are carried out fairly. These benchmark functions in Table 1 can be easily applied by using the developed Matlab GUI.

Table 1. Properties of Used Benchmark Functions

Function	Formula	Dim	Range	Optimal Value
Ackley	$f(x) = -20* \exp(\sqrt{\frac{1}{d}\sum_{1}^{d} x_i^2}) - \exp(\sqrt{\frac{1}{d}\sum_{1}^{d}\cos(2\pi x_i)}))+ 20 + \exp(1)$	*30*	[-32.768, 32.768]	0
Beale	$f(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	*2*	[-4.5, 4.5]	0

Table 1. Properties of Used Benchmark Functions (continued)

Bohachevsky	$f_1(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	*2*	[-100, 100]	0
Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	*2*	[-10, 10]	0
Branin	$f(x) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1 - t)\cos(x_1) + s$	*2*	[-5, 10]	0,3978
Dixon Price	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{d} i(2x_i^2 - x_{i-1})^2$	*30*	[-10, 10]	0
Easom	$f(x) = -\cos(x_1)\cos(x_2)\exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	*2*	[-100, 100]	-1
Goldsteinprice	$f(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \text{ x } [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	*2*	[-2, 2]	3
Griewank	$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos(\frac{x_i}{\sqrt{i}}) + 1$	*30*	[-600, 600]	0
Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	*2*	[-10, 10]	0
Perm	$f(x) = \sum_{i=1}^{d} (\sum_{j=1}^{d} (j^i + \beta)((\frac{x_j}{j})^i - 1))^2$	*4*	[-d, d]	0
Powell	$f(x) = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} + x_{4i})^2 + (x_{4i-2} + 2x_{4i-1})^4 + 10(x_{4i-3} + x_{4i})^4]$	*24*	[-4, 5]	0
Rastrigin	$f(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$	*30*	[-5.12, 5.12]	0
Rosenbrock	$f(x) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	*30*	[-5, 10]	0
Schwefel	$f(x) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{\lvert x_i \rvert})$	*30*	[-500, 500]	0
Shubert	$f(x) = (\sum_{i=1}^{5} i\cos((i+1)x_1 + i))(\sum_{i=1}^{5} i\cos((i+1)x_2 + i))$	*2*	[-5.12, 5.12]	-186,73
Zakharov	$f(x) = \sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5ix_i)^2 + (\sum_{i=1}^{d} 0.5ix_i)^4$	*10*	[-5, 10]	0

A brief introduction of stochastic multi parameter divergence optimization method

The SMDO algorithm uses mechanisms of gradient descent method and evolutionary algorithms in addition with parameter wise random perturbation to access optimum points of an objective function including multi-parameters (Alagoz et al., 2013; Yeroğlu & Ateş, 2014). Essentially the algorithm keeps track of the descent of the objective function by successive sets and trials of parameters. It uses stochastic parameter divergences to access to optimum solution. So the divergence steps are in a bidirectional characteristic that they can be forward or backward direction for each parameter. If these steps do not satisfy the intended conditions the parameter will preserve its state. These divergence steps are produced from the random number generator with a uniform distribution in the range [0, 1]. While using these divergences of parameters one by one, the algorithm walks around a divergence directions in the search space. The range of these directions is very important because if the step in the direction is selected too large, the optimal point can be accessed quickly but the correctness of the solution can worsen; on the contrary if steps are selected too small, the optimal point can be accessed slowly.

In SMDO algorithm, by specifying divergence steps randomly, the optimal point can be approximated more correctly rather than fixed steps because the fixed steps may result in more forward or backward steping around the optimal point.

Figure 1 shows a flow chart of the SMDO that initializes by configuring parameters of SMDO algorithm. Firstly, the algorithm moves forward and backward according to the initial conditions. These forward and reverse movements are made according to uniform distribution as mentioned. The algorithm moves according to whether the objective function value of the related movement is larger or smaller than the objective function value of the previous iteration. For example, if a minimization problem is to be solved, the algorithm always acts towards reducing the error or if a maximization problem is to be solved, the algorithm moves towards increasing objective function. Figure 1 shows a flow chart of the SMDO algorithm.

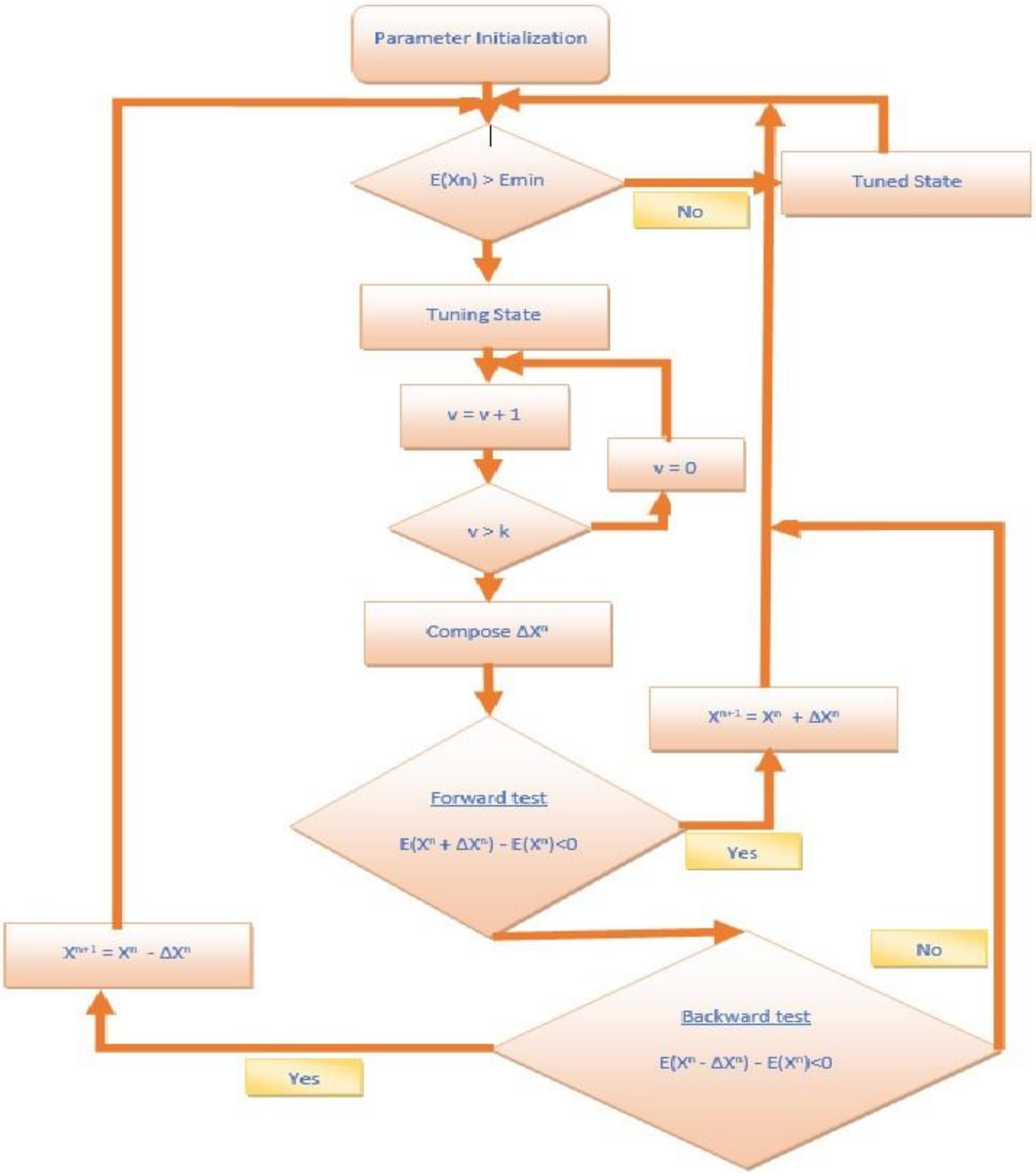


Figure.1. A flow chart of the SMDO Algorithm

RESULTS AND DISCUSSION

Comparisons with diffrerent methods using different benchmark functions

Benchmarking is a method to compare the performance of algorithmic methods because benchmark functions provide standard ways to test the algorithms. However, it is difficult to select an algorithm for an application only by looking at benchmark performances. Because algorithms that are very superior in benchmarking may not produce good results in real engineering problems. Therefore, an algorithm should be examined in many aspects for a real application. Benchmark function analyses should be done for a fair comparision of algorithms. Beacuse they are widely accepted by the algorithm producers and they should be preferred for test and compare the results of their studies in a standard way.

The response of an algorithm to a standard benchmark function can be easily compared with the response of other studies.

Previously, the SMDO algorithm has been applied in control engineering problems however it is tested for benchmark functions. In this study, the performance of the SMDO will be evaluated and compared with other methods.

In Table 2, SMDO algorithm is compared with ABC (Artificial Bee Colony), CK (Cuckoo Search Optimization), PSO (Partical Swarm Optimization) algorithms (Civicioglu & Besdok, 2013). The SMDO algorithm is run several times and the comparisons of the results of related benchmark function according to first and second statistical moments are made. We observed that the SMDO algorithm has satisfactory performance in many benchmark functions.

Another situation to be compared is the computational complexity of algorithms. For example, the SMDO algorithm is an algorithm with low computational complexity because it does not use swarm intelligence. Single solution is improved by single parameter perturbations at each set and trial, and this feature decreases wasting of inefficient set and trials that it occurs in swarm methods. Therefore, SMDO can use set and trial tasks efficiently and cautious. In this way, it can reach the optimal point with less iteration when it is near to the optimal points. In search of multi-optimum space, SMDO can present satisfactory approximation performance when it is initialized around the global minima. Otherwise, it can find one of local minima.

The swarm methods similar to ABC, CK and PSO algorithms have higher computational complexity since they are nature-inspired algorithms and they perform computations to resemble natural phenomenon. Although the swarm intelligence is very beneficial to find global minima in multi-optimum spaces, additional computational complexity of the swarm causes a disadvantage for running algorithms in real time systems. Because the performance of the hardware will be limited while performing the algorithms, in other words, the hardware limitations are also important for online tuning applications. The SMDO algorithm was implemented for online controller tuning and modeling studies, and the computational many experimental study indicated that complexity of the SMDO is manageable by the hardware (Alagoz et al., 2013; A. Ates et al., 2017, 2020; Abdullah Ates et al., 2014; Yeroğlu & Ateş, 2014).

Consequently, initial configuration, divergence ranges and error limitation are critically important for performance of SMDO method, although they are not critical important for swarm based optimization method such as ABC, CK and PSO. This makes SMDO a more cautious tuning tool and it does not freely walk in the search space. For this reasons SMDO has good performance for fine tuning of the optimization parameters. In addition, SMDO has good performance for controller tuning problem. Because in that type of the problem, parameters search space, divergence vector, upper and lower value of the parameters and error limitation are predictable or can be found some approximate values with analytical method and existing study. If the known number of parameters of the system is inadequate, it is difficult for the SMDO algorithm to reach global or best local point. Because of the dependence to a good configuration and initialization of parameter, SMDO method can be found a local point in multi-optimum space applications. Therefore, it will be advantageous to work with swarm based optimization algorithms for that kind of problems. However, the SMDO algorithm has important advantages for fine tuning the optimization parameters. parameters or using optimized parameters in real-time systems.

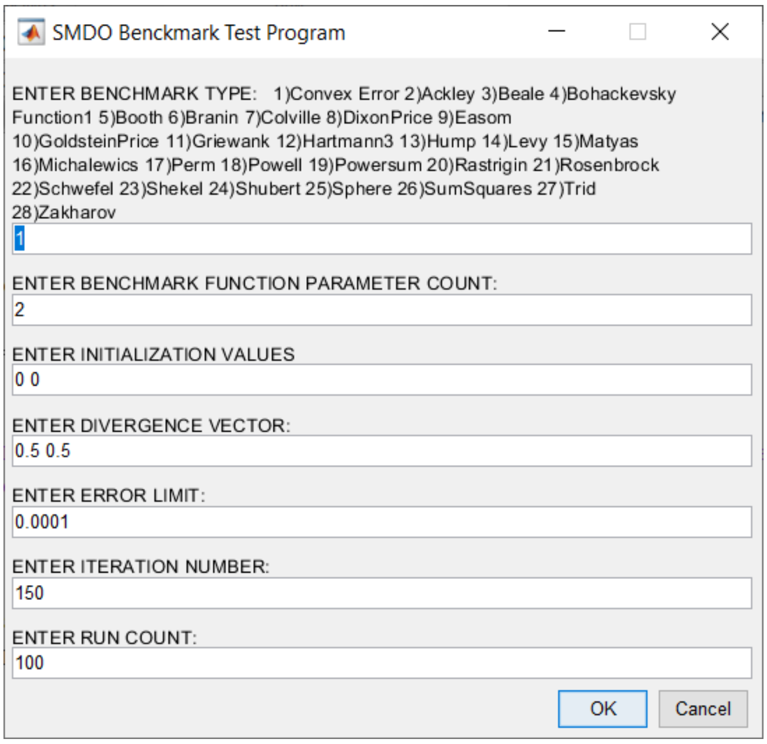
In these experiments the initial conditions are set to zero for all dimensions. The SMDO algorithm is tested 20 times and it is run with 10000 iterations for each test. So it is aimed to the diminish the worsening effects of randomness. The error limit and divergence coefficients are very crucial parts of the SMDO algorithm. Because small divergence provides satisfactory results near the optimal point but then the optimization time gets bigger. Besides that, small error limits again provide facility to get closer near optimum values. But small error limits can be inapplicable in real engineering problems. Since we study with benchmark functions in this work we set small error limits for the algorithm. So for example with Ackley function we set divergence coefficient 0.1x10^{-10} and error limit as 0.1x10^{-16}, with Bohachevsky function we set divergence coefficient 0.1x10^{-10} and error limit as 0.1x10$^{-16,}$ with DixonPrice function we set divergence coefficient 0.1x10^{-2} and error limit as 0.1x10$^{-22,}$ with Goldsteinprice function we set divergence coefficient 0.5 and error limit as 0.1x10^{-2} and similar values with the others.

Table 2. Results of algorithms with Benchmark Functions (Civicioglu & Besdok, 2013)

Benchmark Functions	**Optimization Algorithms**									
	Global Min.	***ABC***		***CK***		***PSO***		***SMDO***		
		Mean	***Best***	***Mean***	***Best***	***Mean***	***Best***	***Mean***	***Best***	***Standard Deviation***
Ackley	0	***0.3e-13***	***2.22e-14***	***4.4e-15***	***4.4e-15***	***0.8e-14***	***0.8e-14***	***1.24e-15***	***8.8818e-16***	***1.09e-15***
Beale	0	***0.8e-15***	***0.4e-15***	***0.0***	***0.0***	***0.0***	***0.0***	***2.79e-07***	***3.3466e-09***	***3.14e-07***
Bohachecsky	0	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***
Booth	0	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***	***9.95e-12***	***1.2242e-13***	***1.04e-11***
Branin	0,3978	***0.3978***	***0.3978***	***0.3978***	***0.3978***	***0.3978***	***0.3978***	***0.3979***	***0.3979***	***1.36e-09***
DixonPrice	0	***2.3e-15***	***1.4e-15***	***0.6667***	***0.6667***	***32.700***	***0.66667***	***0.6667***	***0.6667***	***5.18e-11***
Easom	-1	***−1.0***	***−1.0***	***−0.3***	***−1.0***	***−1.0***	***−1.0***	***-1.21e-08***	***-3.6102e-08***	***8.53e-09***
GoldsteinPrice	3	***2.9999***	***2.9999***	***2.9999***	***2.9999***	***2.9999***	***2.9999***	***3.0004***	***3.0000***	***5.01e-04***
Griewank	0	***0.0***	***0.0***	***0.0***	***0.0***	***0.0092***	***0.0***	***0.0***	***0.0***	***0.0***
Matyas	0	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***	***0.0***	***1.34e-23***	***2.9322e-25***	***7.96e-24***
Perm	0	***0.0204***	***0.0011***	***5.28e-5***	***8.58e-9***	***0.0014***	***8.83e-6***	***0.0541***	***0.0081***	***0.0487***
Powell	0	***0.0004***	***2.71e-4***	***1.5e-7***	***7.97e-8***	***5.18e-5***	***2.1e-5***	***3.25e-23***	***4.4828e-25***	***3.14e-23***
Rastrigin	0	***0.0***	***0.0***	***1.28***	***0.00038***	***28.008***	***13.9294***	***0.0***	***0.0***	***0.0***
Rosenbrock	0	***0.0609***	***0.0002***	***0.0***	***0.0***	***2.3639***	***0.00007***	***3.0982***	***2.8789***	***0.1242***
Schwefel	0	***−12,569.48***	***−12,569.4866***	***−12,505.45***	***−12,569.48***	***−8,927.9796***	***−10,426.97***	***1.25e+04***	***1.2569e+04***	***3.73e-12***
Shubert	-186,73	***−186.73***	***−186.73***	***−186.73***	***−186.73***	***−186.73***	***−186.73***	***-32.4496***	***-64.4725***	***22.6024***
Zakharov	0	***4.88e-14***	***8.3e-15***	***0.0***	***0.0***	***0.0***	***0.0***	***1.93e-25***	***1.6930e-30***	***3.13e-25***

Smdo toolbox

For this study, a SMDO Matlab GUI is developed to test SMDO algorithm for different benchmark test functions. The input parameters of SMDO and test functions can be configured by using user-friendly GUI. This Matlab GUI can be seen in Figure 2. This program can be downloaded from the the link. (https://www.mathworks.com/matlabcentral/fileexchange/75043-smdo-method-with-benchmark functions). In this program, all structures that affect the performance of the SMDO algorithm have been parameterized. For example, with "Enter run count" it is determined how many times the algorithm will be run consecutively. The performance evaluation of stochastic search algorithms should be done according to statistical analysis such the mean performance, the best performance etc. In other words, performance analyses of the algorithms should be carried out by running that algorithm many times consecutively. Therefore, mean and standard deviation values should be given after each multiple run. In this toolbox, this structure has been automated. "Enter iteration number" section indicates how many iterations the algorithm will work. "Enter error limit" section is actually a control mechanism for SMDO. The system tries the values above the given error value and does not take them to the solution. "Enter divergence vector" section determines the speed at which the SMDO algorithm reaches the optimal point. “Enter initialization values” section is one of the most important steps of SMDO. It is especially important in terms of SMDO where the algorithm is started. Therefore, it is very important from which point to start especially in real engineering problems. The “Enter benchmark type” section expects to benchmark type selection and there are 17 different benchmark functions which can be selected in this study. Each benchmark function is assigned a number. By selecting this number, the interested benchmark function is resolved and analyzed according to the specified interval. The “Enter benchmark function parameter count” section expects to parameter count of selected benchmark function. After processing the inputs; the mean and best values are shown with the help of this program to analyze and compare the results.


SMDO Benckmark Test Program

ENTER BENCHMARK TYPE: 1)Convex Error 2)Ackley 3)Beale 4)Bohackevsky Function1 5)Booth 6)Branin 7)Colville 8)DixonPrice 9)Easom 10)GoldsteinPrice 11)Griewank 12)Hartmann3 13)Hump 14)Levy 15)Matyas 16)Michalewics 17)Perm 18)Powell 19)Powersum 20)Rastrigin 21)Rosenbrock 22)Schwefel 23)Shekel 24)Shubert 25)Sphere 26)SumSquares 27)Trid 28)Zakharov
1

ENTER BENCHMARK FUNCTION PARAMETER COUNT:
2

ENTER INITIALIZATION VALUES
0 0

ENTER DIVERGENCE VECTOR:
0.5 0.5

ENTER ERROR LIMIT:
0.0001

ENTER ITERATION NUMBER:
150

ENTER RUN COUNT:
100

Figure.2. Toolbox of SMDO (https://www.mathworks.com/matlabcentral/fileexchange/75043-smdo-method-with-benchmark-functions)

CONCLUSION

This paper presents the comparisons of solutions of different numerical optimization methods by using different benchmark functions. SMDO algorithm has been directly proposed for controller tuning problems. This is an advantage of SMDO in term of being application oriented, because optimization problems are generally designed through benchmark test functions. But if an algorithm does not contribute to a real world problem, it remains restricted in the theoretical ground and one may not use it in a practical way which is not preferred from the engineering point of view. However, there was a need for applying SMDO for benchmark functions for a comparision with other popular methods. Results indicate that SMDO can also provide satisfactory performance when it is configured properly for solving benchmark functions. The study discuses advantages and shortcomings of SMDO method. In addition, as an output of this study, a Matlab GUI was developed and shared for free use.

REFERENCES

Alagoz, B. B., Ates, A., & Yeroglu, C. (2013). Auto-tuning of PID controller according to fractional-order reference model approximation for DC rotor control. *Mechatronics*, *23*(7). https://doi.org/10.1016/j.mechatronics.2013.05.001

Ateş, A. Alagoz, B. B. S. B. C. Y. (2013). Kesir Dereceli PID Kontrolörler İçin Referans Model Tabanlı Optimizasyon Yöntemi. *Türkiye Otomatik Kontrol Milli Komitesi 2013 Malatya*, *1*.

Ates, A., Alagoz, B. B., Chen, Y. Q., Yeroglu, C., & HosseinNia, S. H. (2020). *Optimal Fractional Order PID Controller Design for Fractional Order Systems by Stochastic Multi Parameter Divergence Optimization Method with Different Random Distribution Functions*. 9–14. https://doi.org/10.1109/iccma46720.2019.8988599

Ates, A., Alagoz, B. B., & Yeroglu, C. (2017). Master–slave stochastic optimization for model-free controller tuning. *Iranian Journal of Science and Technology - Transactions of Electrical Engineering*, *41*(2). https://doi.org/10.1007/s40998-017-0029-1

Ateş, A., & Yeroglu, C. (n.d.). *SMDO Algoritması ile İki Serbestlik Dereceli FOPID Kontrol Çevrimi Tasarımı Two Degrees of Freedom FOPID Control Loop Design via SMDO Algorithm*. 6–11.

Ates, Abdullah, AlagOz, B. B., Tepljakov, A., Petlenkov, E., Yeroglu, C., Kuznetsov, A., & Sobolev, I. (2019). Fractional Order Model Identification of Receptor-Ligand Complexes Formation by Equivalent Electrical Circuit Modeling. *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*. https://doi.org/10.1109/IDAP.2019.8875913

Ates, Abdullah, & YEROĞLU, C. (2016). Online Tuning of Two Degrees of Freedom Fractional Order Control Loops. *Balkan Journal of Electrical and Computer Engineering*, *4*(1), 5–11. https://doi.org/10.17694/bajece.52491

Ates, Abdullah, Yeroglu, C., Alagoz, B. B., & Senol, B. (2014). Tuning of fractional order PID with master-slave stochastic multi-parameter divergence optimization method. *2014 International Conference on Fractional Differentiation and Its Applications, ICFDA 2014*. https://doi.org/10.1109/ICFDA.2014.6967388

Biswas, A., Mishra, K. K., Tiwari, S., & Misra, A. K. (2013). Physics-Inspired Optimization Algorithms: A Survey. *Journal of Optimization*. https://doi.org/10.1155/2013/438152

Chopard, B., & Tomassini, M. (2018). Particle swarm optimization. In *Natural Computing Series*. https://doi.org/10.1007/978-3-319-93073-2_6

Civicioglu, P., & Besdok, E. (2013). A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-011-9276-0

Emmerich, M. T. M., & Deutz, A. H. (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing*. https://doi.org/10.1007/s11047-018-9685-y

Frédéric Bonnans, J., Charles Gilbert, J., Lemaréchal, C., & Sagastizábal, C. A. (2006). Numerical optimization: Theoretical and practical aspects. In *Numerical Optimization: Theoretical and Practical Aspects*. https://doi.org/10.1007/978-3-540-35447-5

Giagkiozis, I., & Fleming, P. J. (2015). Methods for multi-objective optimization: An analysis. *Information Sciences*. https://doi.org/10.1016/j.ins.2014.08.071

Kakandikar, G. M., Nandedkar, V. M., Kakandikar, G. M., & Nandedkar, V. M. (2018). Engineering Optimization. In *Sheet Metal Forming Optimization*. https://doi.org/10.4324/9781315156101-4

Kuhn, H. W., & Tucker, A. W. (2014). Nonlinear programming. In *Traces and Emergence of Nonlinear Programming*. https://doi.org/10.1007/978-3-0348-0439-4_11

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*. https://doi.org/10.1016/j.advengsoft.2017.07.002

Nocedal, J., & Wright, S. (2006). Numerical optimization, series in operations research and financial engineering. In *Springer*.

Rao, S. S. (2009). Engineering Optimization: Theory and Practice: Fourth Edition. In *Engineering Optimization: Theory and Practice: Fourth Edition*. https://doi.org/10.1002/9780470549124

Unconstained. (n.d.). Retrieved April 16, 2020, from http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm

Vajda, S., & Dantzig, G. B. (1965). Linear Programming and Extensions. *The Mathematical Gazette*. https://doi.org/10.2307/3612922

Yang, X. S. (2014). Nature-Inspired Optimization Algorithms. In *Nature-Inspired Optimization Algorithms*. https://doi.org/10.1016/C2013-0-01368-0

Yang, X. S., & Gandomi, A. H. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computations (Swansea, Wales)*. https://doi.org/10.1108/02644401211235834

Yeroğlu, C., & Ateş, A. (2014). A stochastic multi-parameters divergence method for online auto-tuning of fractional order PID controllers. *Journal of the Franklin Institute*, *351*(5). https://doi.org/10.1016/j.jfranklin.2013.12.006.