International Journal of Swarm intelligence and Evolutionary Computation

OPEN ACCESS Freely available online

Research Article

An Improved Multitracker Optimization Algorithm and Multiple Subpopulations

Rizk M. Rizk-Allah[1], Fatma Helmy Ismail[2*], Aboul Ella Hassanien[3]

[1]*Faculty of Engineering, Menoufia University, Shebin El-Kom, Egypt.*

[2]*Faculty of Computer Science, Misr International University, Egypt.*

[3]*Faculty of Computers and Information, Cairo University, Egypt.*

ABSTRACT

Recently, a population-based optimization algorithm called the multitracker optimization algorithm (MTOA) was introduced based on the tracker con- cept. This paper proposes a novel variation of the original MTOA called the migration-based MTOA (MTOA1), which employs multiple subpopulations of trackers to achieve superior performance. The proposed algorithm differs from the traditional MTOA in that it splits the initial population into multiple sub- populations to enhance the search process in different areas of the search space. Furthermore, information is exchanged among the subpopulations in an iter- ative and cyclic manner. The best global trackers in the first subpopulation are used to update the global trackers of the second subpopulation, and this updating process continues for all subsequent subpopulations. Exploration and exploitation are balanced in this cyclic approach for multiple populations. The proposed MTOA1 is validated based on the CEC2017 benchmark problems, and an improvement over the original MTOA is observed. Furthermore, MTOA1 is used to solve the classical welded beam design problem and is compared with eight recently proposed optimization algorithms. The results confirm the superiority of the proposed algorithm.

Keywords: Engineering Optimization Problems, Optimization, CEC2017 Benchmark Functions, Multitracker Optimization Algorithm, Multiple populations.

INTRODUCTION

Many nature-inspired algorithms have been developed in the past few years [1]. For instance, a novel swarm intelligence optimization technique called the dragonfly algorithm (DA) was proposed. The primary concept of the DA rule set originates from the static and dynamic swarming behaviours of dragonflies in nature. The important stages of optimization, exploration and exploitation are designed based on modelling of the social interactions of dragonflies during navigation, food location, and swarm protection considering both dynamic and static methods [2]. As a brand new set of rules, the firefly algorithm (FFA) is a metaheuristic algorithm inspired by the flashing behaviour of fireflies.[3]. A novel optimization rule set known as the sine-cosine algorithm (SCA) was introduced in [4] for optimization tasks. The SCA creates preliminary random candidate answers and requires them to vary outwards or towards the optimal solution through sine and cosine functions. One state-of-the-art algorithm is the salp swarm algorithm (SSA)[5]. The SSA is based on the swarming behaviour of salps while navigating and foraging in oceans. A variation of the SSA was introduced in [6]. The whale optimization algorithm (WOA) is a brand new approach for solving optimization problems. This algorithm consists of three operators to simulate the following behaviours of humpback whales: looking for prey, encircling prey, and bubble net foraging [7]. The grey wolf optimization (GWO) algorithm mimics the hierarchy and search mechanisms of grey wolves in nature, and the processes of looking for prey, encircling prey, and attacking prey are considered in the optimization [8]. The ant lion optimizer (ALO) is the lat- est metaheuristic approach that mathematically models the behaviours of ants and antlions in nature. The ALO was

*Correspondence to: Fatma Helmy Ismail, Faculty of Computer Science, Misr International University, Egypt, Tel: 00201005281302, E-mail: fatma.helmy@miuegypt.edu.eg, fatmahelmy2000@yahoo.com

Received: October 04, 2019; **Accepted:** September 15, 2021; **Published:** September 25, 2021

Citation: Ismail FH (2021) An Improved Multi tracker Optimization Algorithm with Multiple Subpopulations an Editorial. Int J Swarm Evol Comput. 10:p352

developed to solve optimization problems considering the random movements of ants, construction of traps, entrapment of ants in traps, catching prey, and rebuilding traps [9]. The grasshopper optimiza- tion algorithm (GOA) models and mimics the behaviours of grasshopper swarms in nature to solve optimization problems [10]. Recently, a population-based al- gorithm called the multitracker optimization algorithm (MTOA) was proposed.

This method included the term ”multitracker” because there are two types of trackers. The first type includes trackers that concentrate on locating the global optimum, and the second type includes trackers that help the first type to learn its surroundings and avoid being trapped at local optima [11]. It is worth noting that the MTOA is not inspired by nature. This paper proposes a novel variation of the original MTOA named the migration-based MTOA (MTOA1), which em- ploys multiple subpopulations of trackers to achieve superior performance. In the proposed MTOA1 approach, some subpopulations work independently in the area of interest of the search space to simultaneously accelerate the search scheme and identify the global optimal. In this scenario, the local behaviours among subpopulations are considered, and one subpopulation can transmit its best solution to another in a cyclic manner. The proposed MTOA1 approach is validated based on the CEC2017 benchmark problems, and an improvement over the original MTOA is observed. The main contributions of this paper are as follows.

- Population diversity is maintained by cross-subpopulation migration.
- The performance of the introduced multiple population-based method is assessed based on the CEC2017 benchmark problems.
- The proposed multiple population method can be adapted to any swarm algorithm without changing its structure.

This paper is organized as follows: section 2 introduces the MTOA; section 3 in- troduces MTOA1; section 4 presents the simulation results; section 5 includes a case study of the welded beam design optimization problem; and the conclusions and future work are discussed in section 6.

Traditional multitracker optimization algorithm (MTOA)

Unlike optimization algorithms that are inspired by nature, the MTOA was developed to overcome the disadvantages of other optimization algorithms and take advantage of specific features. Similar to other algorithms, the MTOA performs exploration and exploitation processes in the search space. To explore the search space, a random number num of points are distributed and called global trackers (GT). The rank of each point i of a GT is calculated (RKi) based on the corresponding cost value. Rank 1 is assigned to the GTi corresponding to the lowest cost, and the highest rank is assigned to the GTi corresponding to the highest cost. The optimum point is selected as the global optimum point (GOP). To avoid local minima, each GTi is surrounded by a number of points called local trackers (LT) in a specified radius Rsi. The LTs inform each GT of its surroundings to avoid becoming trapped at local minima. After searching using the LTs, GTs start to move and search for the optimal point on a random motion technique. The MTOA process can be divided into two main stages:

1.The first stage of the traditional multitracker process: The first stage is exploration. First, num points are randomly initialized in the search space to search the global optimum. These points are called global track- ers (GT). The rank of each global tracker is calculated according to the fitness function of the problem and whether the problem is a maximiza- tion or minimization problem (in this context, minimization problems are assumed). This calculation ends when the global optimum point (GOP) is reached according to equation 1.

GOP = arg(minimum(Rki))   (1)

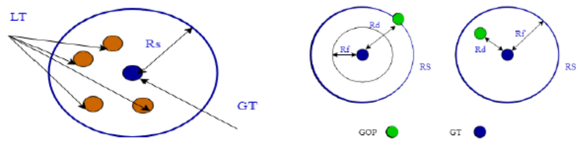
where i=1,2,3,. num

2. The second stage of the traditional multitracker process: The second stage is exploitaion. Each global tracker is informed of its surroundings by a set of points called local trackers. The local trackers are selectd within a certain radius (Rs) of each global tracker point. Based on the received information, global trackers move to better locations in the search space.

The radius around each point i in GT s is determined using equation 2.

$$Rs_i = \left\{\begin{matrix} Rf_i & Rf_i \geq Rd_i \\ Rd_i & Rf_i < Rd_i \end{matrix}\right\} \Bigg| Rf_i = \frac{(Rk_i - 1)}{num - 1}.(RM - Rm) + Rm, \qquad Rd_i = \|G_{Ti} - GP\| \qquad (2)$$

In equation 2, Rm and RM are two constants that present the minimum and maximum radii, respectively; Rki is the rank of the ith GT; and num is the total number of global trackers. Rf varies between Rm and RM; however, if the value of (Rf) is greater than the distance between GTi and GOP ((Rd)), the search radius (Rs) is set to (Rf). Otherwise, the search radius (Rs) is set to (Rd). Therefore, before converging to the optimum solution, the search radii of GT s are large and overlap the GOP. Figures 1a and 1b show the selection of Rs.



(a) Some local trackers (LT) inside the search radius Rs of a global tracker GT (b) if Rd is larger than Rf, then $Rs = Rd$. if Rd is smaller than Rf, then $Rs = Rf$

Figure 1: Selection of Rs

The cost of all local trackers surrounding GTi is calculated and the best point (LPi) is stored to influence the random walk of the global tracker point. The movement of each i point of the global trackers is shown in equation 3.

GTi = ((β)(GOP − GTi) + (1 − β)(LPi − GTi))   (3)\

where β is a value between 0 and 1.

The pseudo code of the MTOA is shown in algorithm 1.

Algorithm 1 Traditional multitracker optimization algorithm (MTOA)

- initialization Initialize the population of Global Trackers (GT). The total number of (GT) is num. Rm and RM are two constants that represent the minimum and maximum radii, respectively. The number of iterations is iterations and iter = 1 initially.
- while iter<iterations do
- //START: The first stage of the traditional multitracker process
- for all point i ∈ GT do
- Determine the cost of point i.
- Determine the rank of point i (RKi)
- end for
- GOP is calculated according to equation 1
- // END: The first stage of the traditional multitracker process
- // START: The second stage of the traditional multitracker process
- for all point i ∈ GT do
- Determine Rsi as in equation 2.
- Determine all Local Trackers (LT)'s for point i within radius Rsi.
- for all point j ∈ LT do
- Determine the cost of point j.
- Determine the rank of point j (RKj).
- end for
- The Local Optimum Point (LPi) = point j with minimum (RKj).
- Move point i according to equation 3.
- end for
- // END: The second stage of the traditional multitracker process 22: iter=iter+1.
- end while
- Select the GOP of the problem.

The improved multitracker optimization algorithm (MTOA1)

To address the complex task of optimization, an improved version of the MTOA is proposed and called MTOA1, which is based on the subpopulation concept [12]. This method was developed to solve recent optimization problems such as the CEC2017 problems. The aim of this improvement is to facilitate the sharing of information, and it consequently improves the diversity of the overall solution. Furthermore, the multiple population concept utilizes the adaptive interactions among the different sub populations to enhance the exploration process compared to that of the single-population concept[13]. In the proposed approach, the original population is first split up into m subpopulations. Each subpopulation contains num trackers, where num=T/m. In this case, T is the total number of trackers. Each subpopulation works independently during the MTOA search process, and the local behaviours of all subpopulations are cyclic as shown in figure 2. Thus, the best tracker obtained from the current

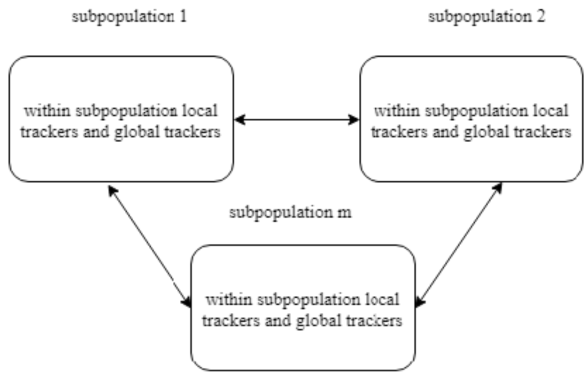


Figure 2: Dividing the population into m subpopulations to exchange information

subpopulation is retained in the next subpopulation. This process of updating the best tracker continues until the final subpopulation is reached. By this mechanism, the best tracker will guide the search process for all trackers in all subpopulations. Note that each subpopulation records the information from its best tracker when the search process begins and compares this information with that received from the previous subpopulation. The MTOA1 steps are as follows.

- Randomly Initialize a population containing T trackers inside the search region.
- Split the population of trackers into m subpopulations each with size num.
- Each subpopulation (subpop) works independently with its own local and global trackers. Each subpopulation identifies its best tracker and retains it in the second subpopulation. This process then continues in a cyclic manner as shown in equation 4.

$$GOP_{subpop} = \begin{cases} arg(minimum(Rk_i)_{i=1,2,3,....num}) & subpop = 1 \\ GOP_1 & subpop = m \\ GOP_{subpop-1} & Cost(GOP_{subpop-1}) < Cost(GOP_{subpop}) \\ GOP_{subpop} & otherwise \end{cases} \quad (4)$$

- If the total number of iterations is reached, then the algorithm stops and the final results are recorded. Algorithm 2 describes the steps of MTOA1. The flowchart of MTOA1 is shown in figure 3.

Simulation Results

CEC2017 test suite includes 29 benchmark functions [14], which are summa- rized in table 1. The benchmark functions can be used to assess the strengths and weaknesses of evolutionary algorithms. A unimodal function (from F1 to F3) is non-separable and direction sensitive with smooth but narrow ridge. Uni- modal problems can be suitably solved with standard exploitation methods. In contrast to unimodal problems, multimodal benchmark problems (from F4 to F10) have many local optima, and the number of optima exponentially increases with the problem dimension. Thus, such problems are appropriate for bench- marking the exploration capability of an approach. In a hybrid function (from F11 to F20), variables are randomly split into subsets and some dissimilar basic functions

[15]. Composition functions (from F21 to F29) combine the properties of certain subfunctions and provide robustness around global or local optima.

Parameter Setting

Most optimization algorithms have parameters that guide the search di- rection towards the global optimal solution. These parameters are extremely

Algorithm 2 the Improved Multitracker Algorithm (MTOA1)

- initialization Initialize the population T . Rm and RM are two constants that
- represent the minimum and maximum radii, respectively. The number of subpopulations is m. The number of iterations is iterations, and iter = 1 initially.
- while iter<iterations do
- Divide the population into m subpopulations.
- The number ofGT (num) in each subpopulation is T /m
- subpop=1.
- while subpop<m do
- start the first stage of the traditional multitracker process.
- if subpop = 1 then
- GOP= arg(minimum(Rki)i=1,2,3,. num).
- else if subpop = m then
- Use GOP of the first subpopulation
- else if Cost(GOPsubpop−1) < Cost(GOPsubpop) then // previous GOP is better
- Use the GOPsubpop−1
- elseUse the GOPsubpop
- end if
- start the second stage of the traditional multitracker process
- subpop=subpop+1.
- end while
- Get the GOP for iteration iter.
- Gather all populations (located at new positions). 21:iter=iter +1.
- end while
- Select the GOP of the problem.

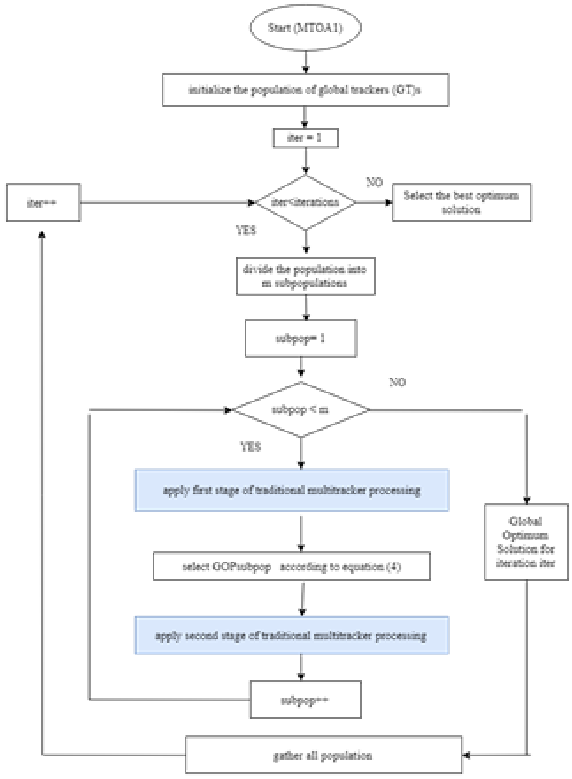


Figure 3: MTOA1 Algorithm Flowchart

CEC 2017			
Type	**No.**	**Description**	**Fi***
	1	Shifted and Rotated Bent Cigar Function	100
Unimodal Function	2	Shifted and Rotated Sum of Different Power Function	200
	3	Shifted and Rotated Zakharov Function	300
	4	Shifted and Rotated Rosenbrock's Function	400
	5	Shifted and Rotated Rastrigin's Function	500
	6	Shifted and Rotated Expanded Scaffer's F6 Function	600

Simple Multimodal Functions	7	Shifted and Rotated Lunacek Bi Rastrigin Function	700
	8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
	9	Shifted and Rotated Levy Function	900
	10	Shifted and Rotated Schwefel's Function	1000
	11	Hybrid Function 1 (N=3)	1100
	12	Hybrid Function 2 (N=3)	1200
	13	Hybrid Function 3 (N=3)	1300
	14	Hybrid Function 4 (N=4)	1400
	15	Hybrid Function 5 (N=4)	1500
	16	Hybrid Function 6 (N=4)	1600
Hybrid Functions	17	Hybrid Function 6 (N=5)	1700
	18	Hybrid Function 6 (N=5)	1800
	19	Hybrid Function 6 (N=5)	1900
	20	Hybrid Function 6 (N=6)	2000
	21	Composition Function 1 (N=3)	2100
	22	Composition Function 2 (N=3)	2200
	23	Composition Function 2 (N=4)	2300
	24	Composition Function 4 (N=4)	2400
	25	Composition Function 5 (N=5)	2500
Composition Functions	26	Composition Function 6 (N=5)	2600
	27	Composition Function 7 (N=6)	2700
	28	Composition Function 8 (N=6)	2800
	29	Composition Function 9 (N=3)	2900

Table 1: Summary of CEC 2017 expensive benckmark problems

Important and unreasonable values can result in divergence. The parameter values are shown in table 2.

Parameter name	**Value**
T	80
m	4
RM	(2)
Rm	1E-8
num	20
iteractions	30

Table 2: List of MTOA1 parameters

Comparison between MTOA1 and Other Algorithms

MTOA1 is compared with the FFA, GOA, WOA, DA, GWO algorithm, ALO, SSA and SCA based on the CEC 2017 benchmark set with 10 dimensions (10D).

Convergence Test

The experimental results show that MTOA1 produces a high-quality so- lution, is not trapped at local minima and yields rapid convergence. Figure 4 shows an example of the convergence curves of ten functions, respectively. MTOA1 outperforms the other algorithms based on the benchmark functions. The obtained solutions indicate that MTOA1 has merit in terms of exploration and exploitation. Based on the obtained convergence performance of MTOA1, it is concluded that MTOA1 can reliably provide high-quality results in a rea- sonable number of iterations, avoid premature convergence in the search process to local optima and provide superior exploration capabilities in the search space. MTOA1 rapidly converges for the following reasons:

- In each iteration, MTOA1 divides the space into m subpopulations with size num. The points in each subpopulation move to better positions in the search space. Each subpopulation transfers its best tracker to the next subpopulation. Meanwhile, the subpopulation moves to the best positions based on the movement of the previous subpopulation. Therefore, each point i in the subpopulation (subpop) is affected by the new positions of a number of points x = num * (subpop - 1)+ (i-1). Some or all points x can be selected as local trackers.
- Conversely, in each iteration, the MTOA has only one subpopulation with size num. Each point i in the search space is affected by the movement of the previous (i-1) points only. Thus, the algorithm converges slowly.
- The number of updated local trackers in MTOA1 is larger than that in the MTOA.

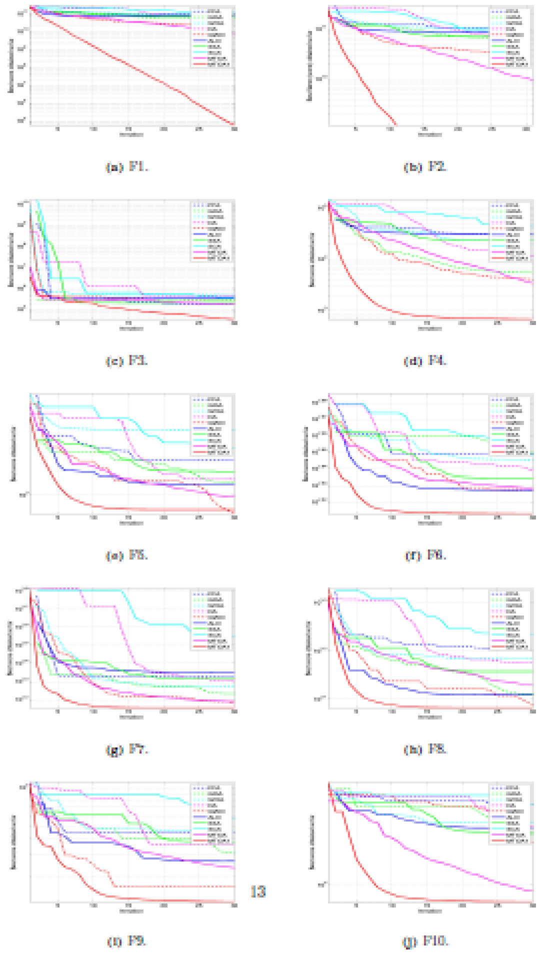


Figure 4: Convergence curves of functions from F1 to F10

Mean and Satandard Deviation

The mean values (the average of 30 runs) obtained are shown in table 3. MTOA1 produces the lowest mean minimization values in the 29 benchmark functions. Table 4 shows the standard deviation of the solutions. The larger the standard deviation is, the larger the fluctuations in the final solution quality. Small standard deviation values indicate that the algorithm can achieve consis- tent and good solutions across almost all runs. MTOA1 achieves the lowest standard deviations for all unimodal functions, two multimodal functions (F8 and F10), six hybrid functions (F11, F12, F13, F14, F17 and F18) and seven composition functions (F22, F24, F25, F26, F27, F28 and F29). Based on the mean values, table 5 shows that MTOA1 always ranks first among all algorithms tested. MTOA1 has the lowest average rank among all other tested algorithms. The calculated p values are reported as metrics of significance as well with an 5% significance level. The main advantages of the MTOA1 can be summarized as follows:

- The MTOA1 has a high convergence rate in comparison with other algo- rithms based on the use of the local trackers. The local trackers surround each global tracker and one of them is nominated as the best local optimum point. Each global tracker changes its location based on that best local optimum point and other factors. Hence, Each global tracker is informed by its surroundings and moves accordingly.

- The MTOA1 cannot be trapped at local optima. Moreover, the probability of reaching the global optimum is higher than that for other algorithms.

	FFA	GOA	WOA	DA	GWO	ALO	SSA	SCA	MTOA	MTOA1
F1	1.12E+11	9.938E+10	1.14E+11	1.01E+11	5.632E+10	1.09E+11	1.21E+11	1.89E+11	6.328E+10	**703.6349**
F2	258469.8561	480631.02	425222.3	2533630.2	432425.25	428888.3	15232653	13988.868	13988.868	**13988.868**
F3	37722.09665	28644.587	42021.63	27594.482	11561.582	36688.65	39673.149	69039.88	13685.472	**610.77977**
F4	1275.585435	1229.16	1327.605	1242.0778	1128.4033	1239.831	1260.2778	1501.8766	1143.8845	**846.12756**
F5	714.4606868	715.06516	726.8988	705.62772	677.84564	698.91556	711.02637	734.57152	686.88666	**657.9504**
F6	2527.028996	2311.8742	2224.539	2295.4634	2042.1339	2649.7449	2439.6262	4829.4287	2410.5776	**1349.0395**
F7	1563.148893	1568.5528	1595.384	1544.2076	1394.884	1516.7322	1560.1189	1766.146	1442.6768	**1123.6578**
F8	53788.77699	53534.753	55139.62	46987.574	32592.96	43803.032	47854.379	84128.047	36677.646	**12365.428**
F9	16789.97325	15018.292	16157.93	16655.382	16542.241	16339.86	16107.963	16842.039	13285.967	**8411.8087**
F10	30845.80439	57294.052	33909.86	43119.752	40054.911	50156.947	31950.454	79619.136	21200.746	**1467.9702**
F11	78161929953	3.884E+10	7.3E+10	5.6E+10	1.192E+10	6.735E+10	7.264E+10	7.906E+10	1.146E+10	**29221328**
F12	41195205851	1.359E+10	2.64E+10	2.498E+10	3.318E+09	4.171E+10	3.116E+10	4.307E+10	2.368E+09	**146978.61**
F13	90093485	31089268	60957531	71764135	15177942	82598489	62557507	65628143	3447268	**113760.76**
F14	8000893691	2.317E+09	6.31E+09	5.154E+09	1.148E+09	4.397E+09	8.529E+09	9.193E+09	117342103	**47858.976**
F15	10459.01382	6642.8259	8109.131	8076.8103	4921.6066	9011.8003	8433.6737	9436.6883	5389.8252	**3650.5596**
F16	13380.12628	5616.5406	7681.596	7098.9074	5946.4499	9241.6025	9293.6858	78510.485	4386.8158	**3573.2825**
F17	169266611.4	85452452	2.57E+08	208692666	18628.412	195667174	149123667	342241483	43118477	**806345.15**
F18	4074360316	422221876	2.41E+09	1.397E+09	60637848	88429205	3.383E+09	5.145E+09	93239296	**46335.318**
F19	4951.94308	4249.9051	4615.909	4960.2521	4589.8696	2.45E+09	4498.1295	4971.0663	3873.8137	**3522.8352**
F20	3294.297495	3169.8381	3240.673	3133.9531	2928.7312	4588.0326	3238.9559	3301.9183	2987.9716	**2624.8363**
F21	18556.88521	16405.995	17654.94	18080.009	18251.887	3189.4838	17170.112	18714.916	15132.789	**9941.6554**
F22	5115.494241	4208.0877	4297.751	4577.0643	3619.0057	17544.06	4319.9036	4331.3313	3682.6933	**3070.6439**
F23	5382.915711	4549.7315	4500.544	4945.0888	3853.3453	5203.2009	4369.4392	4602.8773	4063.3622	**3269.552**
F24	17400.35891	14707.583	16817.85	17991.789	9652.6877	5494.6203	19174.055	38479.926	10798.846	**3116.8746**
F25	18738.91862	18430.864	19635.3	18634.164	12993.722	17674.105	19247.609	22691.786	14827.854	**8295.5743**
F26	7539.976444	5788.2974	6081.985	6910.1698	4810.5981	8466.416	6215.2624	6306.4602	4478.9938	**3827.7939**
F27	13009.99368	12317.728	13362.11	12428.168	8196.4568	14327.717	14273.999	16847.971	8233.9919	**3381.9461**
F28	11317.0522	16287.508	29413.31	58287.009	9707.2332	84154.522	76451.7	58608.116	9223.4218	**5144.172**
F29	755660	2.511E+09	4.27E+09	3.656E+09	792075071	6.009E+09	6.034E+09	8.767E+09	422804760	**12441818**

	4593

Table 3: Mean test results on CEC'17 benchmark functions.

	FFA	GOA	WOA	DA	GWO	ALO	SSA	SCA	MTOA	MTOA1
F1	9840842654	1.127E+10	8.47E+09	9.385E+09	1.727E+10	1.1E+10	1.218E+10	2.173E+10	8.411E+09	**10163.325**
F2	46071.59987	111174.51	92899.62	4438449.3	85746.909	115188.82	133679.47	40363747	107366.48	**7510.2104**
F3	8183.949885	9134.3023	5678.664	6964.6313	3319.7284	6786.1375	10016.694	18118.08	4093.0267	**26.700102**
F4	31.49057933	82.293074	97.03921	39.478426	60.892	53.372927	37.267684	78.251799	64.727294	67.288879
F5	3.983080818	12.54284	11.68204	7.2913457	9.8935878	8.8514786	6.3425374	11.415056	9.8470165	8.603637
F6	195.1033859	260.49395	163.4051	268.08288	226.14834	124.26773	205.5779	766.72582	271.78636	146.26387
F7	54.7160253	86.887275	110.3552	43.261581	74.628923	48.527761	46.048877	88.200182	36.890257	40.882814
F8	6842.745669	8158.5066	12361.16	5817.533	9318.8538	9625.1351	5278.74	10298.134	9918.1867	**2944.2277**
F9	672.3159677	767.74359	645.5785	635.45934	787.13308	815.68215	803.31986	562.15964	685.72317	904.63306
F10	7986.094043	15189.703	14886.63	14223.03	12349.311	21085.493	3765.5938	28863.502	72048.33	**77.432579**
F11	10069957767	1.577E+10	1.79E+10	1.228E+10	5.644E+09	1.071E+10	1.851E+10	2.074E+10	2.742E+09	**25137829**
F12	10212776961	7.039E+09	7.55E+09	5.909E+09	3.144E+09	1.125E+10	8.918E+09	1.175E+10	1.343E+09	**60286.21**
F13	65101517.62	36354203	60619604	54047607	19521433	75025475	63693064	26752254	3498708.5	**65220.391**
F14	3187270009	1.501E+09	3.8E+09	4.065E+09	1.418E+09	2.275E+09	4.007E+09	3.659E+09	95849502	**27277.152**
F15	1260.029807	526.42717	758.3057	1415.2036	648.71656	1506.9078	994.67509	1481.6381	746.76911	535.33831
F16	10020.62341	1376.2077	1788.69	2373.8146	275.09367	4615.5336	4852.8319	126470.66	620.19614	371.02129
F17	75076302.98	54249684	1.83E+08	134733793	76241937	145173425	99073010	147600076	35177191	**473842.05**
F18	1534558009	287297325	1.1E+09	1.003E+09	32160363	1.666E+09	1.847E+09	1.302E+09	64536156	**37805.016**
F19	266.6478977	271.2021	423.3076	331.41483	259.51688	265.30555	383.20848	220.1185	291.97185	374.45891
F20	105.0551169	99.47139	130.5856	62.906807	47.595856	77.258082	96.753073	68.235813	93.130532	72.220503
F21	308.2406757	896.38373	740.1124	549.24615	586.06872	514.36035	684.83421	423.4752	583.43726	918.27237
F22	253.4693649	251.6055	194.2747	168.45956	90.676314	322.53413	174.35891	85.019097	139.46624	**83.803386**
F23	334.4426924	155.21596	187.8132	262.12461	130.83608	338.47104	235.11805	199.45192	103.50468	173.61462
F24	2881.88497	2983.7554	1765.087	3993.6975	1982.179	2706.2861	2994.1834	10291.634	1481.7921	**48.779344**
F25	1555.819521	1153.7935	2021.367	2407.575	1344.9581	1308.6398	1143.6274	2917.5004	1354.4965	**1004.9281**
F26	675.0174107	575.51844	788.0867	612.33709	339.66151	917.96421	754.21168	620.87339	292.22257	**175.56462**
F27	1145.674112	1070.1589	1318.566	1355.2512	685.16341	1115.3675	1387.9181	1949.7408	1035.9364	**68.813339**
F28	205455.7215	6395.6585	18354.43	104223.89	1757.4602	97247.745	108372.71	71975.086	1210.8136	**581.04505**
F29	259326	1.7E+09	1.6E+09	2.081E+09	812231657	3.787E+09	2.514E+09	2.99E+09	170590813	**5474859.1**

	363 0

Table 4: Standard Deviation test results on CEC'17 benchmark functions.

Rank	Unimodal		Multimodal		Hybrid		Composition	
	Algo.,	Avg., Rank	Algo.,	Avg., Rank	Algo.,	Avg., Rank	Algo.,	Avg., Rank
1	MTO A1	1	MTO A1	1	MTO A1	1	MTO A1	1
2	MTO A	3.33	MTO A	3.14	MTO A	2.2	MTO A	2.55
3	FFA	5.33	FFA	7.14	FFA	8.6	FFA	8.22
4	GOA	5.66	GOA	6.28	GOA	3.9	GOA	4.11
5	WOA	7.33	WOA	7	WOA	6.8	WOA	5.88
6	DA	6	DA	5.71	DA	6	DA	6.77
7	GWO	2.33	GWO	3.28	GWO	3.2	GWO	4.33
8	ALO	6.33	ALO	5.71	ALO	6.9	ALO	8
9	SSA	7.66	SSA	5.71	SSA	6.8	SSA	6.77
10	SCA	10	SCA	10	SCA	9.6	SCA	8.66
p value		0.002		0.002		0.002		0.002

Table 5: Average rank table and p value of MTOA1 and other algorithms

Welded Beam Design Optimization Problem

In this section, MTOA1 is used to solve the classical welded beam design problem. Welded beam design problem is a minimization problem with four variables namely the weld thickness (h) , length of bar attached to the weld (l), bar height (t), bar thickness (b) as shown in figure 5. The constraints included in this problem are the bending stress (θ), beam deflection (δ), shear stress (τ), buckling load (P) and other side constraints. The mathematical formulas related to this problem are represented as follows [16]:

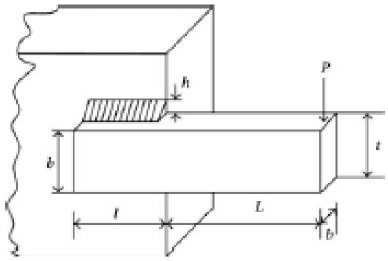
$$\text{Consider}\,\vec{x} = [x_1 x_2 x_3 x_4] = [hltb],$$

$$\text{Minimize}\, f(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2),$$

$$\text{Subject to}\, g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0,$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \le 0,$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \le 0,$$



$$g_4(\vec{x}) = x_1 - x_4 \le 0,$$

$$g_5(\vec{x}) = P - P_c(\vec{x}) \le 0,$$

$$g_6(\vec{x}) = 0.125 - x_1 \le 0,$$

$$g_7(\vec{x}) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \le 0$$

variable range

$$0.1 \le x_1 \le 2,$$

$$0.1 \le x_2 \le 10,$$

$$0.1 \le x_3 \le 10,$$

$$0.1 \le x_4 \le 2,$$

where

$$\tau(\vec{x}) = \sqrt{\tau'^2 + 2\tau'\tau'' \frac{x_2}{2R} + \tau''^2}$$

$$\tau' = \frac{P}{\sqrt{2} x_1 x_2}$$

$$\tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

Figure 5: welded beam optimization problem

$$J = 2\left\{\sqrt{2} x_1 x_2 \left[\frac{x_2^2}{4}\left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}$$

$$\delta(\vec{x}) = \frac{4PL^3}{E x_3^2 x + x_4}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000\ lB, \qquad L = 14\ in.$$

$$\delta_{\max} = 0.25\ in/, \qquad E = 30 \times 10^6\ psi,$$

$$G = 12 \times 10^6\ psi, \qquad \tau_{\max} = 13,600\ psi,$$

$$\sigma_{\max} = 30,000\ psi$$

Table 6 shows a comparison of all the simulation results of the studied opti- mization algorithms for the welded beam design problem. In terms of the best result, MTOA1 outperforms the MTOA. Additionally, the mean obtained by MTOA1 for this problem is better than that obtained by all other algorithms.

Algori thm	Best,	Mean	Std.de v	x1	x2	x3	x4
FFA	1.9953 5	1.38E +19	7.53E +19	0.1259 83	5.5598 73	9.9900 81	0.2018 8
GOA	1.8374 9	1.51E +17	8.28E +17	0.1294 49	5.4208 86	9.0368 15	0.2057 38

WOA	2.0579 7	3.12E +17	1.71E +18	0.1479 82	4.7719 71	8.8105 31	0.2441 29
DA	1.8591 8	2.1939 58	0.2572 98	0.125	5.6409 51	9.0109 47	0.2069 15
GWO	1.7085	1.7591 06	0.0577 85	0.2007 86	3.3773 66	9.0427 06	0.2060 98
ALO	1.7124 2	2.6768 84	0.7474 85	0.1906 44	3.5410 38	9.0465 94	0.2056 8
SSA	1.8444 8	2.4032 3	0.6895 74	0.1294 94	5.4008 92	9.0979	0.2054 26
SCA	1.8730 8	2.1478 53	0.1388 32	0.1917 28	3.7163 26	9.0421 53	0.2234 57
MTO A	1.8223 3	1.9919 07	0.1552 77	0.1380 68	5.0763 28	8.9880 62	0.2079 59
MTO A1	1.6957 6	1.7104 95	0.0199 99	0.2051 95	3.2626 74	9.0366 25	0.2057 3

Table 6: Comparative results of welded beam design problem.

The convergence curves for the welded beam design problem are shown in figure 6 . The performance of MTOA1 is examined and compared to that of the other algorithms through 20 independent runs. Based on the reported perfor- mance, it is clear that MTOA1 provides faster convergence and achieves better outcomes than the other algorithms. In addition, the box plots are provided in figure 7 to illustrate the stability and the distribution of the obtained results of MTOA1 and the other comparative algorithms over 20 independent runs. Based on the obtained results from the box plots, it is noted that MTOA1 is associ- ated with the smallest span among the distributions of the solutions. This result verifies the superior performance of MTOA1 among the compared algorithms.

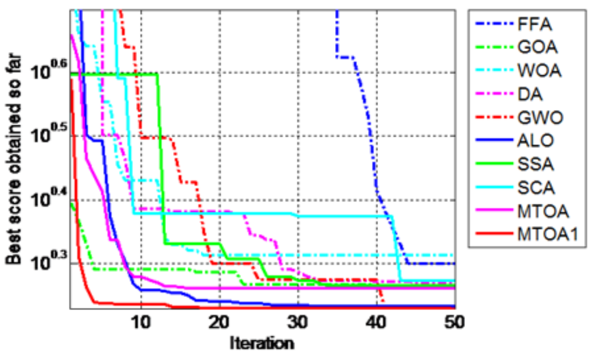


Figure 6: Convergence curves of welded beam design problem

CONCLUSION AND FUTURE WORK

In this paper, an improved MTOA is developed with multiple subpopulations to solve the CEC2017 function optimization problem. The algorithm starts by splitting a single population into several subpopulations. In each subpopulation, the best tracker with the best fitness value is recorded, and this information is transferred to the following subpopulation. All trackers within the same subpop- ulation are updated according to the updating formula. The simulation results show the superiority of MTOA1 to different optimization algorithms. In future work, the improved MTOA with multiple subpopulations may be implemented

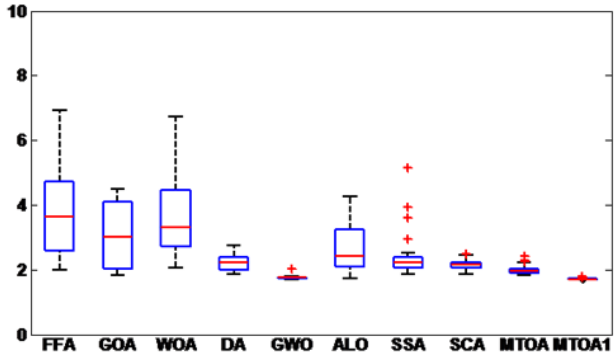


Figure 7: Comparative Box plots between MTOA1 and other algorithms

To solve multi-objective optimization problems. The multipopulation framework offered here can be extended to other varieties of optimization algorithms, such as many-objective optimization, large-scale optimization, and complex system optimization.

REFERENCES

1. X.-S. Yang, Nature-Inspired Optimization Algorithms, 1st Edition, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, (2014).
2. S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization tech- nique for solving single-objective, discrete, and multi-objective problems, Neural Computing and Applications (2016), 27 (4) 1053–1073.
3. X. Yang, Firefly algorithm, stochastic test functions and design opti- misation, Int. J. Bio-Inspired Comput. (2010), 2: 78–84
4. S. M. Mirjalili, Sca: A sine cosine algorithm for solving optimization prob- lems, Knowl.-Based Syst. (2016) 96: 120–133.
5. S. Z. M. S. S. H. F. S. M. M. Seyedali Mirjalili, Amir H.Gandomi, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Advances in Engineering Software (2017) 114 163–191.
6. H. A. E. Rizk-Allah, R.M., A new binary salp swarm algorithm: Develop- ment and application for optimization tasks, Neural Computing and Ap- plications (2018) 1–23.
7. S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in En- gineering Software (2016), 95 51–67.
8. A. L. Seyedali Mirjalili, Seyed Mohammad Mirjalili, Grey wolf optimizer, Advances in Engineering Software (2014) 69: 46–61.
9. A. E.S.Ali, S.M.Abd Elazim, Ant lion optimization algorithm for optimal location and sizing of renewable distributed generations, Renewable Energy (2017) 101: 1311–1324.
10. A. L. Shahrzad Saremi, Seyedali Mirjalili, Grasshopper optimisation al- gorithm: Theory and application, Advances in Engineering Software (2017) 105: 30–47.
11. Y. B.-L. A. Z. Ehsan Zakeri, Seyed Alireza Moezi, Multi-tracker optimiza- tion algorithm: A general algorithm for solving engineering optimization problems, Iranian Journal of Science and Technology, Transactions of Me- chanical Engineering (2017) 41: 315–341.

12. Wei-DerChang, A modified particle swarm optimization with multiple sub- populations for multimodal function optimization problems, Applied Soft Computing (2015) 33: 170–182.

13. H. Ma, M. Fei, Z. Jiang, L. Li, H. Zhou, D. Crookes, A multipopulation- based multiobjective evolutionary algorithm, IEEE Transactions on Cyber- netics (2018) 1–14.