Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

A hybrid pigeon inspired optimization algorithm based on Nelder-Mead simplex operations

Yi Zhong, Yaping Zhang, Kaisheng Liao, Zhenghong Zhang
Southwest China Research Institute of Electronic Equipment, Chengdu 610036, P. R. China
E-mail: eric.yi.zhong@outlook.com, 13678168035@163.com

Abstract: As a recent swarm-based intelligent optimization algorithm, Pigeon Inspired Optimization (PIO), is motivated by the natural bio-mechanism of pigeons for their superior skills in destination finding and navigating. The standard PIO has been successfully implemented to solve complex optimization problems. Similar to other swarm intelligent techniques, PIO is suitable to solve global optimization problems for its robustness in adapting to dynamic environments, where the convergence rate is generally limited, since the algorithm does not employ much local information to establish a most promising direction for optima searching. In this paper, we propose a novel hybridized optimization algorithm, Nelder-Mead Pigeon Inspired Optimization (NMPIO), incorporating the global optimization ability of PIO and the capability of fast local convergent of the Nelder-Mead Simplex method. With the implementation of synthesizing a bio-inspired optimization algorithm and a direct local search method, feasible global optimal solution can be found with a faster convergence rate, compared with the original PIO algorithm. Numerical experiments for several well-known benchmarks are conducted to study the performance of this algorithm. The results reveal that our hybridization strategy is effective and efficient for solving global optimization problems.

Key Words: Mathematical Optimization, Pigeon-Inspired Optimization, Nelder-Mead Simplex, Global Optimization

1 Introduction

Mathematical optimization is the procedure of searching the best solution from a collection of alternatives with respect to a given objective function or multi-objective functions, under some constraints. Optimization techniques are extensively utilized in numerous applications like job scheduling, resources allocating, path planning, physical designing, system controlling.

Real-world optimization problems are often NP-hard, which are extremely challenging to solve. Therefore, many problems have to be solved by trials and errors. One essential group of algorithms have been developed by drawing inspiration from population-based intelligence [1], named swarm intelligence, which studies the cooperative behavior among the natural species. Particle Swarm Optimization (PSO) [2] [3] [4] is a popular swarm intelligent algorithm inspired by the social behavior of a moving bird flock or fish school. Individual of the swarm are considered as particles, positions of which represent candidate solutions to the continuous objective function. During the process of PSO, each individual particle determines its next movement by balancing the effects among individual inertial, historical individual fitness values, neighborhood (swarm) fitness values, and some random perturbations. Pigeon Inspired Optimization (PIO) [5] is motivated by the natural bio-mechanism of pigeons for their superior skills in destination finding and navigating. The standard PIO has been successfully implemented to solve complex optimization problems in various scenarios. Compared with standard PSO, PIO possesses faster convergence rate and ability to avoid local extremes [6].

Though swarm-based intelligent meta-heuristic optimization algorithms are much more capable to solve global optimization problems than local search methods [7], the convergence rate of a swarm intelligence algorithm is normally slower compared with a direct local search method.

In this paper, we propose a hybridized optimization method named Nelder-Mead Pigeon Inspired Optimization (NMPIO), for solving unconstrained optimization problems. The major contribution of this work is a novel approach of introducing a fast direct local search strategy in the process of the standard PIO. NMPIO integrates the Nelder-Mead Simplex operations [8] in the position-updating process of candidate solutions, which accelerates the local search efficiency where the ability of global optimizing is maintained. By incorporating the bio-inspired optimization algorithm with local search operations, feasible global optima can be found with a faster convergence rate, compared with the standard PIO algorithm.

This paper is organized as follows. Section 2 discusses some recent hybridized swarm intelligence optimization approaches related to our work. Section 3 presents the proposed NMPIO method after a brief introduction of the standard Nelder-Mead Simplex method and the standard PIO algorithm. Numerical experiments and results are illustrated in Section 4, where the performance of NMPIO working on benchmarks are studied. Section 5 summarizes this work.

2 Related work

A desired optimization algorithm should have a good balance between the adaptivity of global optimization and the ability of fast local search. Hybridized swarm intelligence optimization algorithms are of significant importance in recent trends of optimization, since a balanced combination of a swarm-based algorithm and a direct local search method will enhance the performance of each searching algorithm.

A modified Ant Colony Optimization (ACO) model underlying the foraging strategy of certain ant species is proposed [9] with incorporating key operations of the tabu-search method in the development of a standard ACO algorithm. The numerical results demonstrate the feasibility and effectiveness of this hybrid algorithm in solving electromagnetic (EM) design problems. Hybridized by PSO and differential evolution (DE), DEPSO [10] provides the bell-shaped mutations with consensus on the population diversity along with the evolution, while keeps the self-organized particle swarm dynamics. The standard PSO algorithm is improved by incorporating a hybridization strategy with the Nelder-Mead Simplex method [11]. With the implementation of combining modified simplex search method and a mutation heuristic strategy to the modified PSO method, the algorithm achieves a faster convergence rate and allows more possibilities to resolve the problem of local convergent.

A recent simple combination of the Nelder-Mead Simplex method and PSO is investigated for a signal source search and localization scenario [12]. The strategy of introducing a fast local search technique in a bio-inspired optimization method speeds up the searching process. Demonstrated by experiments on benchmarks and an interpolated 2.4GHz signal distribution environment, the hybridized algorithm outperforms the standard Nelder-Mead method and PSO regarding the accuracy and the rate of convergence. To handle multi-objective problems, the standard PSO algorithm is extended by introducing a mutation operator. Indicated by the experimental results, this method is highly competitive with respect to some of the best multi-objective evolutionary algorithms [13] [14].

Quantum theory is introduced in the standard PIO process to increase the local search capacity and randomness of population positions [15]. This improved Bloch Quantum-behaved PIO (BQPIO) can avoid the premature convergence problem and suitable for multi-modal applications. PIO algorithm is improved with a modified fitness function to solve complex constrained application problems in hypersonic vehicle trajectory optimization [16]. Hybridized Predator-Prey PIO (PPPIO) algorithm is applied to speed up the convergence rate and to keep global optimization for an automatic landing system of a fixed-wing unmanned aerial vehicle (UAV) in the longitudinal plane [17].

3 Methodology

Direct methods like the Nelder-Mead Simplex algorithm have rapid convergence rates but may easily fall into a local optimum for a global optimization problem. Though swarm intelligence optimization algorithms generally lack the ability of fast local search, they have relatively strong adaptivity of global optimization compared with direct methods.

The idea of our hybrid approach is to combine the advantages of the Nelder-Mead Simplex and the standard PIO algorithm. The goal is to design a balanced combination of a swarm intelligence algorithm and a direct local search method, to enhance the overall performance in the rate of convergence and global optimization. In this section, we start with introducing the Nelder-Mead Simplex algorithm and the standard PIO, then describe our proposed hybrid optimization method NMPIO.

3.1 Nelder-Mead Simplex method

The Nelder-Mead Simplex (NM) algorithm [8] is a direct local search method for nonlinear unconstrained optimization problem without the needing of gradient information. This algorithm iteratively generates a new scale of the simplex formulated on the local behavior of the objective function by using five procedures: ordering, reflection, expansion, contraction and shrinkage, where a simplex is an n-dimensional geometry of nonzero volume that is the convex hull of $n+1$ vertices for an n-dimensional optimization problem. Figure 1 is an illustration of the simplex operations for a two-dimensional optimization problem. For a two-dimensional minimization problem, in each iteration, the worst vertex (red dot) with the largest value will be updated by another position (blue dot pointed by the arrow) to form a new simplex along with the remaining vertices, where the yellow dot denotes the centroid of the simplex. Hence the generated simplex will eventually converge to the optimum value of the objective function [18].

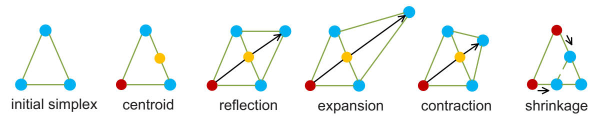

Fig. 1: Illustration of Nelder-Mead Simplex operations

For a two-dimensional minimization problem, denote x_1, x_2, x_3 as the vertices of the initial simplex, and the centroid of the simplex is given as $\bar{x} = \frac{1}{2}\sum_{i=1}^{2} x_i$, denote $f(x_i)$ to be the fitness value on vertex x_i, the NM operations are given as follows.

Step 1 Rank the vertices based on the corresponding fitness value $f(x_i)$ such that $f(x_1) \leq f(x_2) \leq f(x_3)$.

Step 2 Compute the *reflection point* $x_r = (1+\rho)\bar{x} - \rho x_3$. If $f(x_r) \leq f(x_2)$, replace x_3 with x_r and go back to **Step 1**.

Step 3 If $f(x_r) < f(x_1)$, compute the *expansion point* $x_e = (1+\rho\xi)\bar{x} - \rho\xi x_3$. If $f(x_e) < f(x_r)$, replace x_3 with x_e. If $f(x_r) \leq f(x_e)$, then replace x_3 with x_r and go back to **Step 1**.

Step 4 If $f(x_2) \leq f(x_r) < f(x_3)$, compute the *outside contraction point* $x_c = (1+\rho\gamma)\bar{x} - \rho\gamma x_3$. If $f(x_c) \leq f(x_r)$, replace x_3 with x_c and go back to Step 1. Compute the *inner contraction point* $x_{cc} = (1-\gamma)\bar{x} + \gamma x_3$, if $f(x_{cc}) < f(x_3)$, replace x_3 with x_{cc} and go back to **Step 1**.

Step 5 If $f(x_r) < f(x_c)$ or $f(x_3) \leq f(x_{cc})$, compute the *shrinkage points* $x_2 = \sigma x_2 + (1-\sigma)x_1, x_3 = \sigma x_3 + (1-\sigma)x_1$, and go back to **Step 1**.

The values of the Nelder-Mead parameters $\rho, \xi, \gamma, \sigma$ are given in Tabel 2.

3.2 Pigeon Inspired Optimization

Pigeon Inspired Optimization (PIO) method [5] is a recent swarm-based optimization algorithm inspired by the social behavior of destination finding and navigating of the pigeon flock. This algorithm contains two stages of operations. The first is a map and compass operator modeling the bio-mechanism of pigeons shaping the map by using magnetoreception and adjusting the flying direction by regarding the altitude of the sun. The second is a landmark operation simulating the landmark effect of pigeons when they fly close to the destination. Pigeons will fly straight to the destination if they are familiar with the neighboring landmarks, otherwise they will follow those pigeons that can recognize the landmarks.

3.2.1 Map and compass operation

Define the position and velocity of pigeon i at time t as $x_i(t)$ and $v_i(t)$, the updating equations of position and velocity are given in Equation (1) and Equation (2),

$$v_i(t+1) = e^{-rt} v_i(t) + q_1 \left(x_g(t) - x_i(t)\right) \quad (1)$$
$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where $r > 0$ is the map and compass factor, q_1 is a random number uniformly distributed in $(0, 1)$, and $x_g(t)$ is the global best position among all pigeon positions at time t generated by comparing the fitness values obtained by all pigeons.

3.2.2 Landmark operation

Pigeons rely more on their familiar neighboring landmarks for navigation when they get close to the destination. In the landmark operation, select those multi-agents with better fitness values as leading pigeons that are familiar with the landmarks, followed by the rest pigeons. Denote NC as an iteration threshold of PIO operations, in other words, when $i < NC$, do the map and compass operation, otherwise perform the landmark operation, where i is the current number of iterations. In each iteration of the landmark operation, half of the pigeons with worse fitness values will be reduced to accelerate the searching process of the entire pigeon flock. The center position of the remaining pigeons is calculated in Equation (3), and identified as the reference flying direction for the pigeon flock in the process of position updating shown in Equation (4),

$$C(t+1) = \frac{\sum_{N_p} x_i(t) fitness((x_i(t)))}{\sum fitness(x_i(t))} \quad (3)$$
$$x_i(t+1) = x_i(t) + q_2 \left(C(t+1) - x_i(t)\right) \quad (4)$$

where q_2 is a random number uniformly distributed in $(0, 1)$, $C(t)$ is the central position of the remaining pigeons, $N_p(t)$ denotes the current number of population at time t, $N_p(t+1) = \frac{1}{2} N_p(t)$, $fitness(x_i(t))$ is chosen as $(f_{min}(x_i(t)) + \epsilon)^{-1}$ for a minimization problem, and $fitness(x_i(t)) = f_{max}(x_i(t))$ for a maximization problem.

3.3 NMPIO method

The motivation of this proposed hybridized algorithm, NMPIO, originates from the dilemma between fast convergence and global optimization when considering either performing NM or PIO exclusively. To resolve the conflict, we attempt to design an algorithm that incorporates the superiorities of those two optimization techniques.

Shown in Figure 2, NMPIO algorithm can be considered as a combination of two components: *Group 1* and *Group 2*.

Group 1 is a *NM component* formed by the best $(D+1)$ ordered agents driving the M agents moving to the direction where the NM algorithm calculated. M denotes the number of agents for a D-dimensional minimization problem.

Group 2, form by the remaining $(M - D - 1)$ agents, can be considered as a linear combination of three PIO subcomponents in Equation (1) - (4): an *inertia component* to

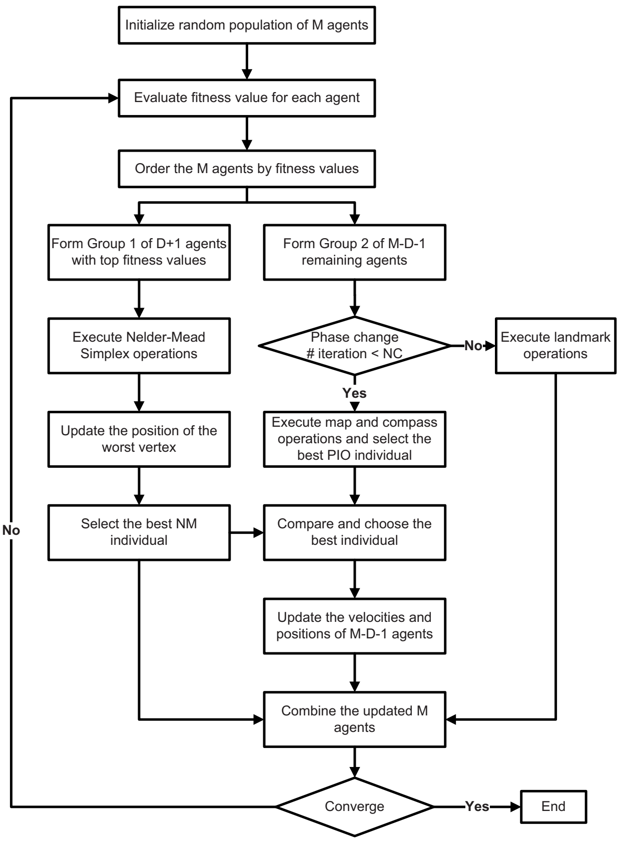

Fig. 2: Flowchart of NMPIO algorithm

keep the agents moving along their previous directions, a *social component* which encourages the agents to move to the best position that the swarm has found so far, and a *cognitive component* which causes each agent to return to its individual best position.

Therefore, with the implementation of this hybrid approach, the drawback of relatively slow convergence in PIO is improved by introducing the *NM component*. Simultaneously, the limitation of local optimization within the NM algorithm is ameliorated by the *social component*.

The pseudocodes of NMPIO for a minimization problem are specified in Algorithm 1 and Algorithm 2.

4 Numerical experiments

The goal of the following numerical experiments is to analyze the proposed NMPIO algorithm in comparison with the standard Pigeon Inspired Optimization, Particle Swarm Optimization, and Nelder-Mead Simplex algorithm, with respect to the ability of fast local search and the adaptivity of global optimization.

Three typical two-dimensional benchmark functions are applied due to their differences in complexity and convergence behavior.

- Rosenbrock function

$$f_1 = (1 - x_1)^2 + 100 \left(x_2 - x_1^2\right)^2 \quad (5)$$

Algorithm 1 Initialization of the NMPIO algorithm

Input: numDimension, numPopulation, initialPositions, initialVelocities

Initialize random numPopulation agents, and generate initialPositions and initialVelocities.

for i = 1 to numPositions **do**
  Compute fitness(initialPositions(i)) values $f(\cdot)$
end for
f_{sorted} = sort($f(\cdot)$)
$index_{NM}$ = find($f(\cdot) \leq f_{sorted}(numDimension+1)$)
$index_{PIO}$ = find($f(\cdot) > f_{sorted}(numDimension+1)$)
$position_{NM}$ = initialPositions($index_{NM}$)
$position_{PIO}$ = initialPositions($index_{PIO}$)
Find the position of global best individual $gbest$
Collect the $(numDimension+1)$ agents with $position_{NM}$ as group G_{NM}
Collect the remaining agents with $position_{PIO}$ as Group G_{PIO}

Algorithm 2 Main code of the NMPIO algorithm

Input: numDimension, $position_{NM}$, $position_{PIO}$, $gbest$, maxIterations, compassFactor, NC

Initialize the group of agents, and generate initial positions and velocities.
for N_c = 1 to maxIterations **do**
  if $N_c \leq NC$ **then**
    Form G_{NM} and G_{PIO} according to the fitness values of the numPopulation agents
    Execute the NM operations to G_{NM}
    Update the position of the worst simplex vertex, and find $gbest_{NM}$, the position of best G_{NM} agents
    Execute the PIO map and compass operations to G_{PIO}, and find $gbest_{PIO}$, the position of best G_{PIO} agents
    Get the global best position $gbest$ between $gbest_{NM}$ and $gbest_{PIO}$, where $gbest$ will be applied in the next iteration
    Combine the updated $(numDimension+1)$ simplex vertices and the remaining PIO agents
  end if
  if $N_c > NC$ **then**
    Form G_{NM} and G_{PIO} according to the fitness values of the numPopulation agents
    Execute the PIO landmark operations to G_{PIO}
    Get the global best position $gbest$ between $gbest_{NM}$ and $gbest_{PIO}$, where $gbest$ will be applied in the next iteration
    Combine the updated $(numDimension+1)$ simplex vertices and the remaining PIO agents
  end if
end for

The Rosenbrock function reaches its global minimum at $(1,1)$:

$$min(f_1) = f(1,1) = 0 \tag{6}$$

- Rastrigin function

$$f_2 = 20 + x_1^2 + x_2^2 - 10\left(cos(2\pi x_1) + cos(2\pi x_2)\right) \tag{7}$$

The Rastrigin function achieves its global minimum at $(0,0)$:

$$min(f_2) = f(0,0) = 0 \tag{8}$$

- Ackley function

$$f_3 = 20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{2}\sum_{i=1}^{2} x_i^2}} - e^{\frac{1}{2}\sum_{i=1}^{2} cos(2\pi x_i)} \tag{9}$$

The global minimum of the Ackley function locates at $(0,0)$:

$$min(f_3) = f(0,0) = 0 \tag{10}$$

Swarm-based optimization algorithms will perform different search paths in each run, since random terms are included in their formulas. Consequently, it is inappropriate to compare the performance of these algorithms for a certain run. To deal with this problem, we take 30 independent runs with same random initial positions and measure the averaged performance. In each run, we perform 20 iterations, and measure the L_1-norm error between the computational and real minimum function values. Numerical experiments are conducted on an Intel i7-4770 3.40 GHz desktop computer with 16 GB memory.

For PSO, PIO, NM and NMPIO, the selected algorithmic parameters are given in Table 1 and Table 2.

Table 1: Swarm-based parameters for PSO/PIO/NMPIO

Algorithm	**PSO**	**PIO**	**NMPIO**
No. of agents	20	20	20
Iterations	20	20	20
Map and compass factor	-	0.5	0.5
NC	-	6	6
Inertial weight	0.6	-	-
global best weight	2	-	-
personal best weight	2	-	-

Table 2: Nelder-Mead parameters for NM/NMPIO

Algorithm	**NM**	**NMPIO**
Reflection parameter ρ	1	1
Expansion parameter ξ	2	2
Contraction parameter γ	0.5	0.5
Shrinkage parameter σ	0.5	0.5

4.1 Case A: Rosenbrock function

Figure 3 shows that the neighboring region around the optima is quite flat, which means the changes of fitness values are too small to follow for gradient-based optimization methods.

Demonstrated in Figure 4, the NM method has the slowest rate of convergence, since the fitness values vary slightly in the Rosenbrock function, and the updating of the worst vertex in the NM operations intuitively calculate and compare the differences of fitness values acquired at neighboring vertices. Meanwhile, the standard PSO achieves a 10^{-4} magnitude of accuracy in 20 iterations.

Among all the methods, the proposed NMPIO algorithm provides the fastest rate of convergence. Compared with the standard PIO, NMPIO converges faster in the hybridized landmark operations and achieves much better accuracy (10^{-19} magnitude) with 20 iterations.

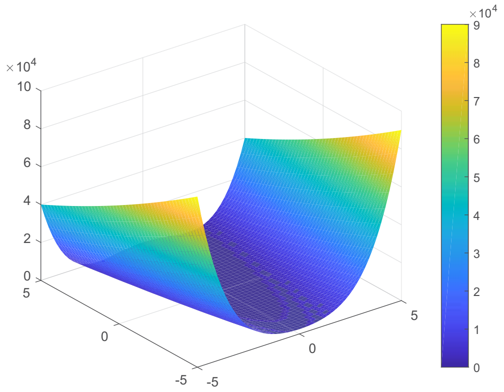

Fig. 3: Surface plot of the Rosenbrock function

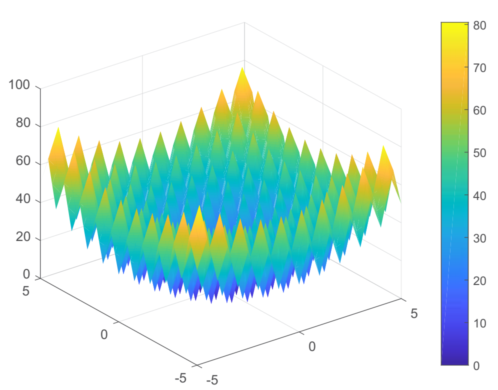

Fig. 5: Surface plot of the Rastrigin function

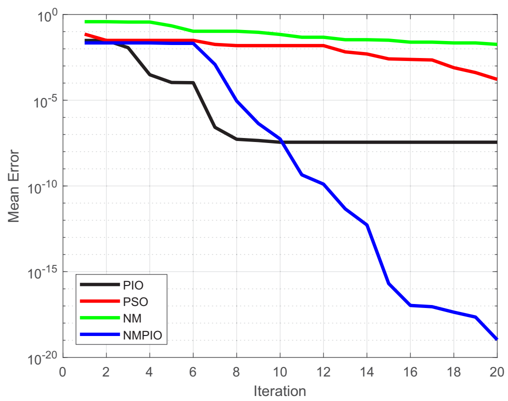

Fig. 4: Convergence results on the Rosenbrock function

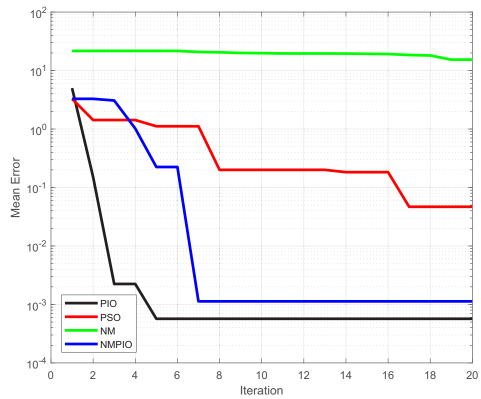

Fig. 6: Convergence results on the Rastrigin function

4.2 Case B: Rastrigin function

Figure 5 shows that there are multiple local extremes located in the surface plot of the Rastrigin function, which makes it quite difficult to solve for local search methods.

Revealed in Figure 6, as we can expect, the NM method converges to a local optima with accuracy over 10^1 magnitude. Though in some special cases when the initial positions are located near the global optima, the NM method can provide a very rapid convergence rate, the overall average performance of it is poorer than swarm-based global optimization algorithms.

The NMPIO and PIO algorithm successfully find the global optima with accuracy of 10^{-3} magnitude, and outperform the standard PSO regarding both the rate of convergence and the accuracy.

4.3 Case C: Ackley function

Similar to the Rastrigin function, the Ackley function has multiple local extremes as shown in Figure 7.

In this case, demonstrated in Figure 8, the NM method actually is capable to find the global optima with the accuracy of 10^{-2} magnitude. This may results from the fact that the fitness values of the Ackley function vary significantly in local neighborhood, which provides more intuition for the NM method to search for the best direction that the function value changes most rapidly. Meanwhile, the NMPIO provides the best performance considering both the accuracy (10^{-9} magnitude) and the rate of convergence, compared with N-M, standard PIO and PSO.

4.4 Summary of numerical experiments

Compared with the standard PIO technique, demonstrated in the results of the numerical experiments, NMPIO is capable to solve complex global optimization problems with a faster convergence rate, while the problem of the premature of numerical solutions with PIO is also ameliorated (See Figure 4 and Figure 8). The *NM component* introduced in the NMPIO method accelerates the local search efficiency compared with the standard PIO. Meanwhile, the *social component* in NMPIO keeps the ability for global optimization.

5 Conclusions

In this work, to solve the dilemma between fast convergence and global optimization, the NM operations are introduced to the standard PIO algorithm to generate a novel hybridized global optimization algorithm, NMPIO.

With the implementation of synthesizing the NM and the PIO operations, feasible global optimal solution can be found. The results of the numerical experiments reveal that the proposed hybrid algorithm NMPIO outperforms the standard PIO, PSO, and the NM method on three typical benchmark functions regarding the rate of convergence and global optimization, indicating that the proposed algorithm is well-balanced for global optimization problems.

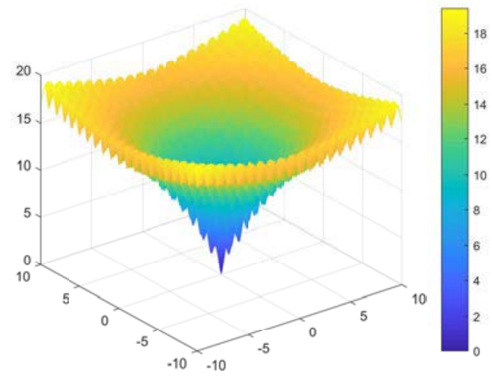

Fig. 7: Surface plot of the Ackley function

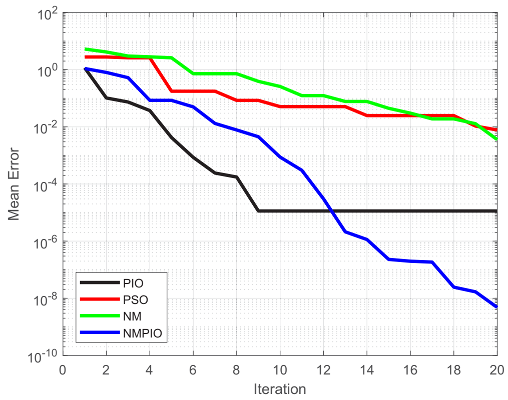

Fig. 8: Convergence results on the Ackley function

In the future, we will analyze the convergence of NMPIO, and apply this algorithm for some more computationally expensive optimization benchmarks with different dimensions, like CEC2014 and CEC2017 problem sets.

References

[1] I. Fiste, X. Yang, J. Brest, D. Fister, A brief review of nature-inspired algorithms for optimization, *arXiv preprint arXiv:1307.4186*, 2013.

[2] R. Eberhart, J. Kenney, A new optimizer using particle swarm theory, In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995: 39–43.

[3] J. Kennedy, Particle swarm optimization, *Encyclopedia of machine learning*, 2010: 760–766.

[4] Y. Shi, R. Eberhard, Empirical study of particle swarm optimization, In *Proceedings of the 1999 Congress on Evolutionary Computation*, 1999: 1445–1450.

[5] H. Duan, P. Qiao, Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning, *International Journal of Intelligent Computing and Cybernetics*, 7(1): 24–37, 2014.

[6] S. Zhang, H. Duan, Multiple UCAVs target assignment via bloch quantum-behaved pigeon-inspired optimization, In *Proceedings of the 34th Chinese Control Conference (CCC)*, 2015: 6936–6941.

[7] X. Yang, S. Deb, S. Fong, X. He, Y. Zhao, From swarm intelligence to metaheuristics: nature-inspired optimization algorithms, *Computer*, 49(9): 52–59, 2016

[8] J. Nelder, R. Mead, A simplex method for function minimization, *The Computer Journal*, 7(4): 308–313, 1965.

[9] S. Ho, S. Yang, G. Ni, J. Machado, A modified ant colony optimization algorithm modeled on tabu-search, *IEEE Transactions on Magnetics*, 42(4): 1195–1198, 2006.

[10] W. Zhang, X. Xie, DEPSO: hybrid particle swarm with differential evolution operator, In *Proceedings of 2003 IEEE International Conference on Systems, Man and Cybernetics*, 2003: 3816–3821.

[11] E. Zahara, Y. Kao, Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems, *Expert Systems with Applications*, 36(2): 3880–3886, 2009.

[12] Y. Zhong, K. Liao, Y. Zhang, A Multi-Agent Based Hybrid Optimization Method for Signal Source Search and Localization, In *Proceedings of the 2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*, 2019: 126–131.

[13] C. Coello, G. Pulido, M. Lechuga, Handling multiple objectives with particles swarm optimization, *IEEE Transaction on Evolutionary Computation*, 8(3): 256-279, 2004.

[14] K. Deb, *Multi-objective optimization using evolutionary algorithms*. Chichester: John Wiley & Sons, 2001.

[15] H. Li, H. Duan, Bloch quantum-behaved Pigeon-inspired optimization for continuous optimization problems, In *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, 2014: 2634–2638.

[16] Y. Zhang, P. Sun, Y. Li, Y. Liu, Hypersonic vehicle trajectory optimization based on improved pigeon-inspired optimization algorithm, *Flight Dynamics*, 35(4): 60-64, 2017.

[17] H. Duan, M. Huo, Z. Yang, Y. Shi, Q. Luo, Predator-Prey Pigeon-Inspired Optimization for UAV ALS Longitudinal Parameters Tuning, *IEEE Transactions on Aerospace and Electronic Systems*, 55(5): 2347-2358, 2019.

[18] J. Lagarias, J. Reeds, M. Wright, P. Wright, Convergence properties of the Nelder–Mead simplex method in low dimensions, *SIAM Journal on Optimization*, 9(1): 112–147, 1998.