A Modified APSODEE for Large Scale Optimization

1st Dongyang Li
Department of Electronics and Information Engineering
Tongji University
Shanghai, China
lidongyang0412@163.com

2nd Weian Guo*
Sino-German College of Applied Sciences
Tongji University
Shanghai, China
guoweian@tongji.edu.cn

3rd Lei Wang*
Department of Electronics and Information Engineering
Tongji University
Shanghai, China
wanglei@tongji.edu.cn

4th Qidi Wu
Department of Electronics and Information Engineering
Tongji University
Shanghai, China
wuqidi@tongji.edu.cn

2021 IEEE Congress on Evolutionary Computation (CEC) | 978-1-7281-8393-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEC45853.2021.9505003

***Abstract*—The balance of exploration and exploitation of particle swarm optimization (PSO) is still a great challenge in large scale optimization. In our previous work, an adaptive particle swarm optimizer with decoupled exploration and exploitation (APSODEE) has been proposed to solve this problem. However, it still shows room for further improvements. First, APSODEE guides particles with the best individual in each sub-swarm, which is adverse to swarm diversity preservation especially in case of large scale optimization. Second, APSODEE fails to lead particles moving to sparse areas which are also potentially promising. To address these issues, two modifications are incorporated into APSODEE including a partial updating strategy and a quality restrained local sparseness diversity measurement. The former is proposed to further enhance the algorithm's swarm diversity preservation ability while the latter is designed to help guiding updated particles moving towards the areas which are both sparse and potentially promising, resulting in the modified APSODEE. The experiments are conducted based on CEC 2013 benchmarks with 1000 dimensionality. The results show the competitiveness of the proposed algorithm.**

***Index Terms*—particle swarm optimization, balance of exploration and exploitation, large scale optimization, partial updating strategy, quality restrained local sparseness diversity**

I. Introduction

Particle swarm optimization (PSO) is famous of its simplicity and efficiency since its advance [1], [2]. The position and velocity of each particle in PSO are iteratively updated according to

$$\begin{aligned} v_i^d(t+1) =& \omega v_i^d(t) + c_1 r_1 (pbest_i^d(t) - p_i^d(t)) \\ &+ c_2 r_2 (gbest^d(t) - p_i^d(t)) \end{aligned} \tag{1}$$

$$p_i^d(t+1) = p_i^d(t) + v_i^d(t+1), \tag{2}$$

Corresponding author: Weian Guo, Lei Wang. Email: guoweian@tongji.edu.cn, wanglei@tongji.edu.cn.

where $v_i^d(t)$ and $p_i^d(t)$ are the dth dimension of the ith particle's velocity and position at generation t, respectively; $gbest(t)$ denotes the best position at generation t; $pbest_i(t)$ is the best position achieved by the ith particle so far at generation t; ω is termed as inertia weight; c_1 and c_2 are acceleration coefficients set by users; r_1 and r_2 are two random number generated within $(0, 1)$. Due to its simplicity and efficiency, PSO has been widely applied to solve various kinds of optimization problems [3]–[9].

However, large scale optimization problem (LSOP) is still challenging for PSO and its variants [10], [11]. The reason is that the areas around the local optima will be sharply extended and the number of the local optima will be exponentially increased with the growing of the dimensionality. The current studies on LSOP mainly focus on proposing new updating or learning strategies and incorporating PSO into the cooperatively coevolutionary (CC) framework.

To name a few. For the methods with new updating or learning strategies. Zhao $et\ al.$ propose DMS-PSO using multi-swarm technique to improve the swarm diversity [12]. Cheng $et\ al.$ propose FBE based on a dual-swarm learning strategy and a mutation strategy [13]. Cheng $et\ al.$ propose CSO, where a pairwise competition learning is proposed to enhance the diversity preservation ability of CSO [14]. Cheng $et\ al.$ propose SLPSO which allows each dimension of a particle to learn from different superior individuals [15]. Yang $et\ al.$ propose DLLSO, where a level-based learning strategy is proposed to balance the exploration and exploitation [11]. They further propose a distributed PSO variant DEGLSO [16]. Li $et\ al.$ propose APSODEE, where the exploration and exploitation are decoupled into different components [10].

For the methods with CC framework, researches focus on decomposing the whole dimensionality into several small size segmentations which are independently and simultaneously optimized. Van $et\ al.$ first incorporate PSO into CC framework and propose $\mathrm{CCPSO{-}S_K}$ and $\mathrm{CCPSO{-}S_H}$ [17]. Yao $et\ al.$

propose CCPSO2 based on the Gaussian and Cauchy mutation for further balancing the exploration and exploitation, respectively [18]. Tang $et\ al.$ propose AM-CCPSO which involves more than one context vectors and adopts Gaussian sampling method to enhance the convergence [19]. Furthermore, a huge amount of effort has been put into designing variable grouping techniques to improve the performance of the methods with CC framework. Such as the variable interaction learning [20], meta-modeling decomposition [21], DG [22], XDG [23], GDG [24], and DG2 [25].

However, the existing methods still leave room for further improvements. The methods with new updating and learning strategies still suffer from the balance of exploration and exploitation; on the other hand, the methods with CC framework usually cause unexpected computational burden. In term of the balance of exploration and exploitation for the methods with new updating and learning strategies, this paper proposes a partial updating strategy and a quality restrained local sparseness diversity measure for APSODEE to further balance the exploration and exploitation, resulting in the modified APSODEE (M-APSODEE).

The rest of this paper is organized as follows. Section 2 briefly introduces the implementation of APSODEE. Section 3 presents the details of the proposed method. Section 4 tests the performance of the proposed algorithm. The conclusion and further work are summarized in Section 5.

II. Brief Introduction of APSODEE

The main idea of APSODEE is to conduct the exploration and exploitation in different components. This enables a explicitly management of these two factors. The velocity and position of the ith particle in APSODEE are updated according to

$$\begin{aligned} v_i^d(t+1) =& \omega v_i^d(t) + \phi r_1(p_{exploration,i}^d(t) - p_i^d(t)) \\ &+ r_2(p_{exploitation,i}^d(t) - p_i^d(t)) \end{aligned} \tag{3}$$

$$p_i^d(t+1) = p_i^d(t) + v_i^d(t+1), \tag{4}$$

where $p_{exploration,i}(t)$ and $p_{exploitation,i}(t)$ are the exploration and exploitation exemplars, respectively; $v_i^d(t)$ and $p_i^d(t)$ are the dth dimension of the ith particle's velocity and position, separately; ω, r_1 and r_2 hold the same meaning with that in (1); ϕ is set by users. In (1), $\phi r_1(p_{exploration,i}^d(t) - p_i^d(t))$ and $r_2(p_{exploitation,i}^d(t) - p_i^d(t))$ focus on the exploration and exploitation, respectively.

More specifically, first, APSODEE divides the whole swarm into m sub-swarms at generation t; second, $p_{exploitation,i}(t)$ is the best particle in a sub-swarm and the remaining particles are recorded in a set $P_{update,1}$; third, the local sparseness diversity should be computed for all the particles, then the swarm will be sorted according to the LSD in ascending order. If $Rank(i) < rand \cdot Npop$, the jth particle should be stored in set $P_{update,2}$. Where $Rank(j)$ denotes the ranking of the jth particle in the sorted swarm and $Npop$ is the whole swarm size. Finally, the velocity and position of the kth particle in $P_{update,1} \cup P_{update,2}$ are updated according to (3) and (4).

III. The Proposed Algorithm

A. Motivations

However, APSODEE still cannot find the global optima in many cases as shown by the results in [10], which indicates that the balance of exploration and exploitation in APSODEE still needs to be improved.

First, all the particles in APSODEE learn from the best individual in each sub-swarm. This is potentially adverse to the diversity preservation since the stagnation is more like to happen in LSOP. Second, only the sparseness information is taken into considerations in LSD in APSODEE. However, guiding particles moving to the areas with bad quality might be a waste of computational resources.

To solve the above two issues, a partial updating strategy and a quality restrained LSD are proposed as follows.

B. M-APSODEE

1) Partial Updating Strategy: As discussed above, the exploitation learning strategy in APSODEE will greatly reduce the swarm diversity if the swarm is trapped into stagnation. To alleviate this issue, a partial updating strategy (PUS) is proposed, based on which a PUS-based learning strategy (PUS-LS) is put forward. To be specific, first, similar to APSODEE, the proposed method divides the whole swarm into m sub-swarms according to the adaptive sub-swarm size strategy proposed in APSODEE; second, two empty sets $P_{update,1}$ and P_{e1} are defined to store the particles to be updated and the corresponding exploitation exemplars at generation t, which is the same with APSODEE; third, for the ith particle in the kth sub-swarm, if $rand < 0.5$, then $P_{update,1} \leftarrow P_{update,1} \cup$ the ith particle and $P_{e1} \leftarrow P_{e1} \cup$ the best particle in the kth sub-swarm; finally, the ith particle in $P_{update,1}$ learns from the corresponding exemplar in P_{e1} for exploitation.

Compared with the multi-swarm strategy in APSODEE, PUS not only inherits the ability of dynamically adjusting the difference between the updated particles and the exemplar with different sub-swarm size, but also can ensure that more particles can be preserved at each generation, which is beneficial to swarm diversity preservation.

2) Quality Restrained Local Sparseness Diversity: By taking both the sparseness information and particles' quality into consideration, the proposed quality restrained local sparseness diversity (QSLSD) is computed according to

$$con(i) = \frac{l_{i,1} + l_{i,2}}{L} \tag{5}$$

$$dis(i) = \frac{min(l_{i,1}, l_{i,2})}{max(l_{i,1}, l_{i,2})} \tag{6}$$

$$qs(i) = \frac{N - rankf(i) + 1}{N} \tag{7}$$

$$QSLSD(i) = \frac{con(i)}{max(con)} \frac{dis(i)}{max(dis)} qs(i), \tag{8}$$

where $con(i)$ and $dis(i)$ evaluate the congestion and distribution of the ith particle, respectively. $l_{i,1}$, $l_{i,2}$, and L are the corresponding intervals shown in Fig. 1, $rankf(i)$ is the ranking of the $i\mathrm{th}$ particle according to the fitness quality; N denotes the swarm size.

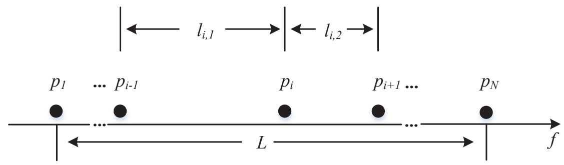


Fig. 1. Computation of $l_{i,1}$, $l_{i,2}$, and L.

To conduct exploration, a QSLSD-based learning strategy (QSLSD-LS) is proposed. First, two set $P_{update,2}$ and P_{e2} should be defined to store the particles to be updated and the corresponding exploration exemplars at generation t, respectively; second, sort the swarm according to the $QSLSD$ in ascending order; third, $P_{update,2} \leftarrow P_{update,2} \cup$ the $i\mathrm{th}$ particle if $rankQSLSD(i) < rand \cdot Npop$, where $rankQDLSD(i)$ is the ranking of the $i\mathrm{th}$ particle in the sorted swarm; forth, randomly select a particle with higher $QSLSD$ and put it into the P_{e2}. Finally, the $j\mathrm{th}$ particle in $P_{update,2}$ updates its velocity and position by learning from the corresponding exemplar in P_{e2} for exploration.

Compared with APSODEE, the modified $QSLSD$ takes both the sparseness information and particles' quality into consideration. This ensures that particles can be guided to the areas which are relative sparse and promising.

C. M-APSODEE

By combining PUS-LS and QSLSD-LS, M-APSODEE is proposed. The particles in M-APSODEE update their velocity and position according to

$$\begin{aligned} v_i^d(t+1) =& \omega v_i^d(t) + \phi r_1(p_{e1,i}^d(t) - p_i^d(t)) \\ &+ r_2(p_{e2,i}^d(t) - p_i^d(t)) \end{aligned} \tag{9}$$

$$p_i^d(t+1) = p_i^d(t) + v_i^d(t+1), \tag{10}$$

where p_i is the $i\mathrm{th}$ particle in $P_{update,1} \bigcap P_{update,2}$; $p_{e1,i}$ and $p_{e2,i}$ are produced by PUS-LS and QSLSD-LS as the exploitation and exploration exemplars, respectively; r_1 and r_2 are randomly generated within $[0,1]$; ϕ is a user defined parameter to balance the exploration and exploitation.

IV. Experiments

In this part, first, the proposed M-APSODEE is compared with seven state-of-the-art algorithms to test its performance on LSOP; second, a comparison between APSODEE and M-APSODEE is presented to show the difference between these two algorithms.

A. Compared Algorithms

To confirm the validation of the experiments, APSO-DEE is compared to 7 peer algorithms including CSO [14], SLPSO [15] and DLLSO [11], APSODEE [10], MA-SW-CHAINs [26], DECC-DG2 [27], and MMO-CC [28]. Where CSO, SLPSO, DLLSO and APSODEE are PSO variants; MA-SW-CHAINs is the winner of the CEC 2010 on LSOP; DECC-DG2 and MMO-CC are two CC framework based EAs.

B. Experimental Settings

For a fair comparison, fitness evaluations (FEs) is adopted as the terminal condition, where the maximum FEs is set to $3E{+}06$ as recommended by [11]. the parameter settings of M-APSODEE is the same with that recommended by APSODEE. For the 7 algorithms in comparison, the parameters are set to the recommended settings in the corresponding papers. Wilcoxon rank sum test is adopted to do the statistical analysis between M-APSODEE and the other algorithms, where the significant level is set to 0.05 and the p value is the significance factor. Each algorithm runs 30 times on each function.

C. Results

Table. I shows the numerical comparison between M-APSODEE and seven peer algorithms. $w/l/t$ at the bottom of Table. I represents that the times APSO-DEE wins/loses/ties in the competitions compared to the corresponding algorithms.

As shown by the comparisons on average performance, M-PSODEE outperforms others for 10 times out of the 15 functions, which indicates the competitiveness of the proposed algorithm. With a deep insight, M-APSODEE wins all other algorithms on 5 times out of the 8 multimodal functions of F_2, F_3, F_5–F_7, F_9, F_{10}, F_{12}; for the 7 unimodal functions, M-APSODEE wins for 5 times. For the competition on the partially separable benchmarks of F_4 to F_{12}, M-APSODEE wins for 6 times. APSO-DEE also performs competitively on the overlapping and non-separable benchmarks.

The stochastic analysis on the bottom of Table. I also shows the competitiveness of M-APSODEE. M-APSODEE outperforms the four PSO variants for 13, 14, 12, 12 times, respectively; compared to MA-SW-CHAINs, M-APSODEE wins for 12 times; finally, M-APSODEE outperforms DECC-DG2 and MMO-CC for 12 and 13 times, respectively.

The convergence of the eight algorithms are also recorded as shown in Fig. 2. It clearly shows that M-APSODEE has a competitive convergence during the optimization process, especially in the middle and latter optimization stage.

In summary, the experimental results demonstrate the competitiveness of M-APSODEE on both performance and convergence.

D. Comparison Between APSODEE and M-APSODEE

To further investigate the exploration and exploitation behaviours of M-APSODEE, the comparisons on diversity and convergence between M-APSODEE and APSODEE are conducted based on F_1 (unimodal function) and F_5 (multimodal function). The results are shown in Fig. 3. As shown by the results, M-APSODEE outperforms APSODEE on convergence on both F_1 and F_5; furthermore, in the comparisons on the diversity, M-APSODEE has a better diversity in the early optimization process and converges faster in the latter stage. These findings indicate that the proposed strategies are effective in balancing the exploration and exploitation, which assists M-APSODEE to better explore the decision space at the beginning and exploit the promising areas in the late optimization stage.

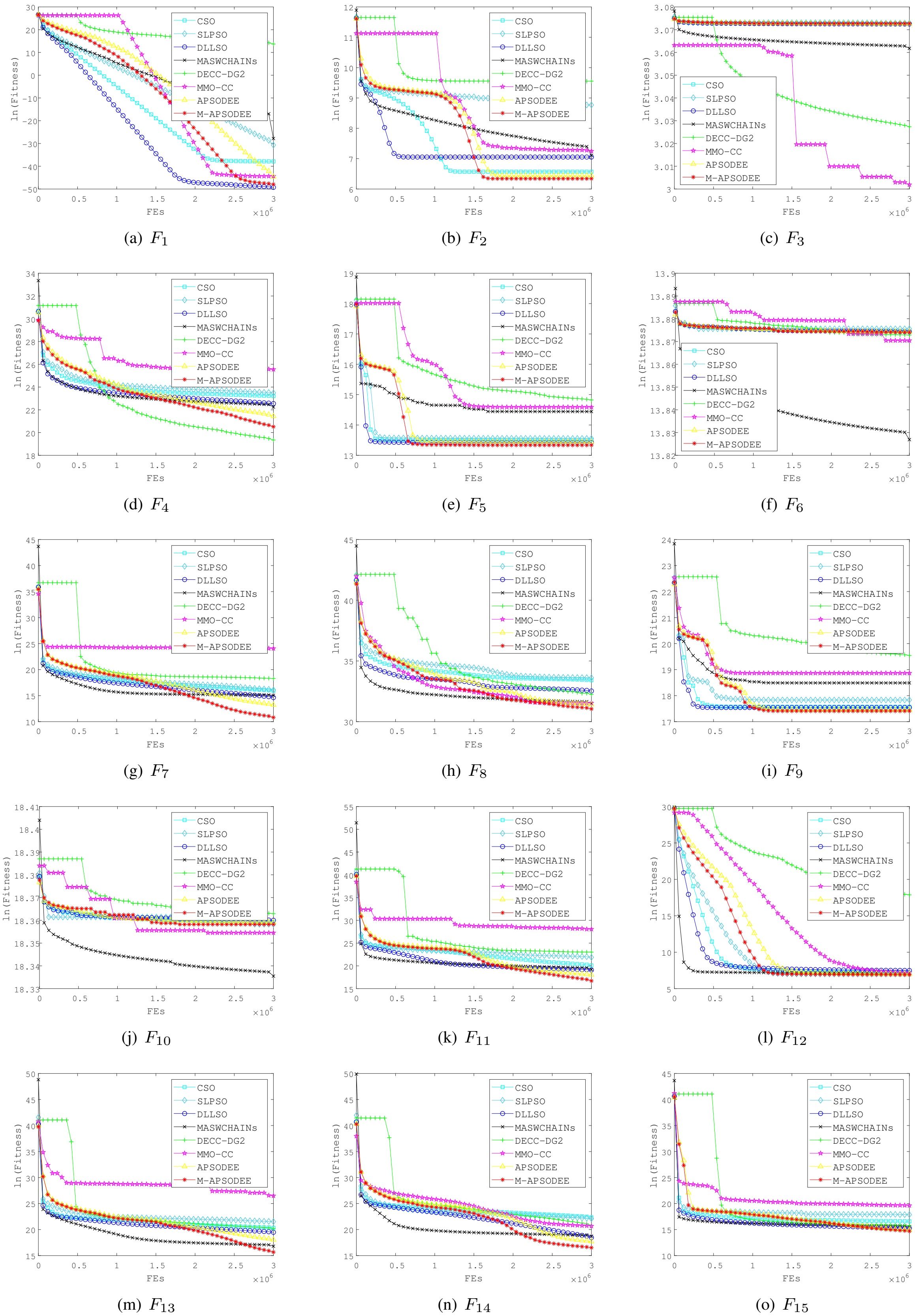


Fig. 2. Convergence profiles of different algorithms obtained on the CEC 2013 test suite with 1000 dimensions. Note that the logarithmic values of the average results obtained by 30 independent runs are shown in the above figures for clarity.

TABLE I
THE EXPERIMENTAL RESULTS OF 1000-DIMENSIONAL IEEE CEC 2013 BENCHMARK FUNCTIONS WITH FITNESS EVALUATIONS OF 3E6.

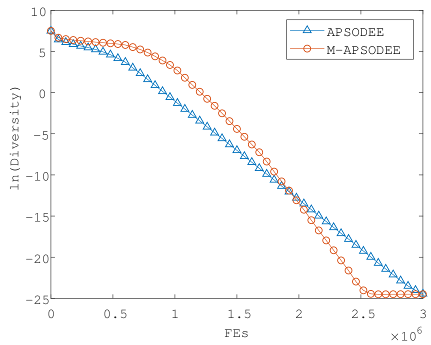
Function	Quality	**CSO**	**SLPSO**	**DLLSO**	**APSODEE**	**MASWCHIANS**	**DECCDG2**	**MMOCC**	**MAPSODEE**
	mean	3.60E-17	3.70E-14	3.99E-22	4.14E-20	8.49E-13	8.65E+05	4.82E-20	1.53E-21
	std	4.69E-19	1.44E-15	2.06E-23	3.62E-21	2.18E-13	2.27E+05	1.30E-21	1.55E-22
F1	p-value	**1.35E-09**	**1.35E-09**	1.35E-09	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	-
	mean	7.14E+02	6.70E+03	1.16E+03	6.31E+02	1.22E+03	1.41E+04	1.51E+03	5.67E+02
	std	6.22E+00	4.98E+01	1.21E+01	4.49E+00	2.28E+01	3.02E+02	8.43E+00	2.26E+00
F2	p-value	**1.35E-09**	**1.35E-09**	**1.35E-09**	**2.46E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	-
	mean	2.16E+01	2.16E+01	2.16E+01	2.16E+01	2.14E+01	2.06E+01	2.01E+01	2.16E+01
	std	1.23E-03	1.14E-03	1.00E-03	8.95E-04	1.12E-02	1.69E-03	2.36E-03	6.99E-04
F3	p-value	9.10E-02	1.92E-04	2.29E-03	3.77E-02	1.35E-09	1.35E-09	1.35E-09	-
	mean	1.20E+10	1.20E+10	6.21E+09	2.05E+09	4.58E+09	2.51E+08	5.15E+11	7.85E+08
	std	3.44E+08	5.54E+08	2.75E+08	5.73E+07	4.91E+08	1.89E+07	9.71E+10	1.34E+07
F4	p-value	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	1.23E-09	**1.35E-09**	-
	mean	7.33E+05	7.58E+05	6.78E+05	6.88E+05	1.87E+06	2.74E+06	2.42E+06	6.18E+05
	std	2.01E+04	2.14E+04	1.94E+04	2.18E+04	6.13E+04	5.66E+04	1.14E+05	1.31E+04
F5	p-value	**1.29E-04**	**1.02E-05**	**2.28E-03**	**2.30E-02**	**1.33E-09**	**1.08E-09**	**1.33E-09**	-
	mean	1.06E+06	1.06E+06	1.06E+06	1.06E+06	1.01E+06	1.06E+06	1.06E+06	1.06E+06
	std	2.08E+02	1.64E+02	1.71E+02	1.82E+02	3.06E+03	4.50E+02	6.41E+02	1.51E+02
F6	p-value	5.34E-01	**5.79E-05**	4.72E-01	3.61E-01	1.35E-09	2.91E-02	**3.15E-03**	-
	mean	8.35E+06	1.73E+07	2.26E+06	5.49E+05	3.45E+06	8.93E+07	1.28E+10	4.93E+04
	std	5.44E+05	1.49E+06	2.01E+05	4.59E+04	2.53E+05	7.16E+06	1.07E+09	2.77E+03
F7	p-value	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.34E-09**	**2.32E-09**	**1.34E-09**	-
	mean	3.22E+14	2.89E+14	1.33E+14	4.40E+13	4.85E+13	1.01E+14	1.54E+14	3.05E+13
	std	1.14E+13	1.75E+13	6.36E+12	2.27E+12	2.03E+12	1.31E+13	4.45E+13	8.46E+11
F8	p-value	**1.35E-09**	**1.35E-09**	**1.35E-09**	**9.81E-07**	**7.41E-08**	**4.03E-08**	**8.71E-05**	-
	mean	4.29E+07	4.44E+07	4.16E+07	3.80E+07	1.07E+08	3.08E+08	1.76E+08	3.63E+07
	std	1.33E+06	1.47E+06	1.52E+06	1.23E+06	3.36E+06	1.39E+07	7.03E+06	1.09E+06
F9	p-value	**2.28E-03**	**5.77E-05**	**4.75E-02**	3.12E-01	**1.34E-09**	**1.32E-09**	**1.34E-09**	-
	mean	9.40E+07	9.43E+07	9.41E+07	9.40E+07	9.18E+07	9.44E+07	9.38E+07	9.39E+07
	std	3.14E+04	3.99E+04	3.73E+04	4.25E+04	2.12E+05	5.82E+04	1.02E+05	3.27E+04
F10	p-value	**4.75E-02**	**2.59E-06**	**1.18E-03**	**3.77E-02**	1.35E-09	**4.14E-06**	3.12E-01	-
	mean	5.40E+08	9.98E+09	1.95E+08	6.85E+07	2.19E+08	9.93E+09	5.66E+12	1.80E+07
	std	1.44E+08	1.82E+09	1.07E+07	2.80E+06	5.96E+06	3.26E+09	1.09E+12	1.73E+06
F11	p-value	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.34E-09**	-
	mean	1.35E+03	1.13E+03	1.80E+03	1.12E+03	1.25E+03	5.81E+07	1.14E+11	9.95E+02
	std	1.64E+01	2.12E+01	2.86E+01	1.27E+01	2.11E+01	1.53E+07	6.32E+10	1.55E+00
F12	p-value	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.01E-09**	**5.96E-07**	-
	mean	6.91E+08	2.05E+09	2.88E+08	6.74E+07	1.98E+07	6.03E+08	1.32E+12	6.38E+06
	std	4.33E+07	2.13E+08	3.85E+07	6.08E+06	3.64E+05	2.69E+07	2.88E+11	1.45E+05
F13	p-value	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.34E-09**	**1.03E-09**	**1.34E-09**	-
	mean	5.15E+09	1.60E+10	1.04E+08	4.94E+07	1.36E+08	1.11E+09	4.12E+11	1.50E+07
	std	5.78E+08	1.62E+09	1.06E+07	3.77E+06	4.22E+06	2.10E+08	1.21E+11	3.73E+05
F14	p-value	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	**1.35E-09**	-
	mean	1.61E+07	6.68E+07	4.32E+06	3.07E+06	5.71E+06	7.11E+06	4.05E+08	2.36E+06
	std	2.26E+05	1.01E+06	5.99E+04	5.37E+04	1.51E+05	2.70E+05	1.91E+07	5.12E+04
F15	p-value	**1.35E-09**	**1.35E-09**	**1.35E-09**	**4.30E-08**	**1.35E-09**	**8.87E-10**	**1.35E-09**	-
w/l/t		13/1/1	14/1/0	12/2/0	12/1/1	12/3/0	12/3/0	13/1/1	-

V. CONCLUSION AND FUTURE WORK

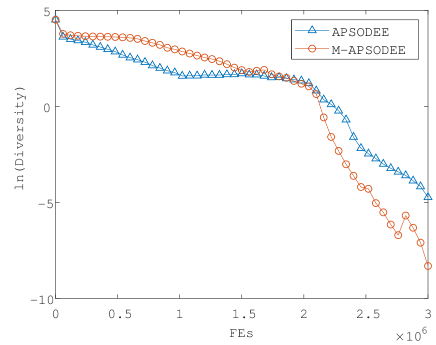
In conclusion, this paper proposes the partial updating strategy and the quality restrained local sparseness diversity to further balance the exploration and exploitation for APSODEE. The experimental results show that the proposed strategies are able to improve both the exploration and exploitation of APSODEE, resulting in a better balance of these two factors. The numerical comparisons also demonstrate the competitiveness of M-APSODEE in comparison to seven peer algorithms. Furthermore, the proposed partial updating strategy and the quality restrained local sparseness diversity are independent to the algorithm and benchmarks. Therefore, in our future work, we will test the proposed strategies' performance in combinations with other kinds of algorithms. In addition, the combination of M-APSODEE and other kinds of optimization techniques is also an interesting study.

ACKNOWLEDGMENT

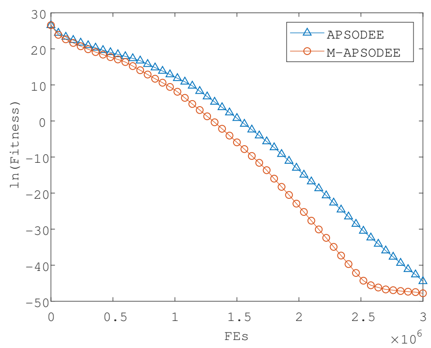
This work is supported by the National Natural Science Foundation of China under Grant Number 71771176 and 61503287, Natural Science Foundation of Shanghai, China under Grant Number 19ZR1479000, 20692191200 Fundamental Research Funds for the China Central Universities under Grant Number 22120190202, China Scholarship Council under File Number 201906260030, Joint Fund of Zhejiang Provincial Natural Science Foundation under Grant Number LHY20F030001



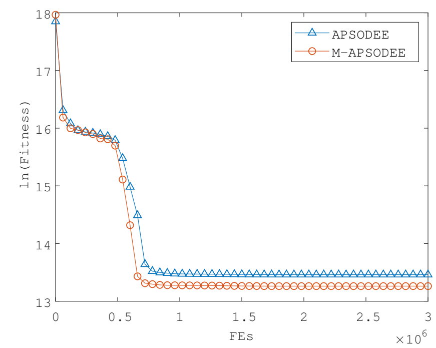
(a) F_1



(b) F_5



(c) F_1



(d) F_5

Fig. 3. Diversity and convergence comparisons between APSODEE and M-APSODEE on F_1 and F_5 with 1000 dimensionality. Note that the logarithmic values of the average results obtained by 30 independent runs are shown in the above figures for clarity.

REFERENCES

[1] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Proceedings of ICNN'95-International Conference on Neural Networks. Vol. 4. IEEE, 1995.

[2] Eberhart, Russell, and James Kennedy. "Particle swarm optimization." Proceedings of the IEEE international conference on neural networks. Vol. 4. Citeseer, 1995.

[3] Kashyap, Abhishek Kumar, and Dayal R. Parhi. "Particle swarm optimization aided PID gait controller design for a humanoid robot." ISA transactions (2020).

[4] HA, Mahmoud Pesaran, et al. "A hybrid genetic particle swarm optimization for distributed generation allocation in power distribution networks." Energy 209 (2020): 118218.

[5] Radha, R., and R. Gopalakrishnan. "A medical analytical system using intelligent fuzzy level set brain image segmentation based on improved quantum particle swarm optimization." Microprocessors and Microsystems 79 (2020): 103283.

[6] Che, Gaofeng, Lijun Liu, and Zhen Yu. "An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle." Journal of Ambient Intelligence and Humanized Computing 11.8 (2020): 3349-3354.

[7] Jahandideh-Tehrani, Mahsa, Omid Bozorg-Haddad, and Hugo A. Loiciga. "Application of particle swarm optimization to water management: an introduction and overview." Environmental Monitoring and Assessment 192 (2020): 1-18.

[8] Hua, Zhoudong, Junfeng Chen, and Yingjuan Xie. "Brain storm optimization with discrete particle swarm optimization for TSP." 2016 12th International Conference on Computational Intelligence and Security (CIS). IEEE, 2016.

[9] Zhang, Wan-qiu, Yong Zhang, and Chao Peng. "Brain storm optimization for feature selection using new individual clustering and updating mechanism." Applied Intelligence 49.12 (2019): 4294-4302.

[10] Li, Dongyang, et al. "An adaptive particle swarm optimizer with decoupled exploration and exploitation for large scale optimization." Swarm and Evolutionary Computation 60: 100789.

[11] Yang, Qiang, et al. "A level-based learning swarm optimizer for large-scale optimization." IEEE Transactions on Evolutionary Computation 22.4 (2017): 578-594.

[12] Zhao, Shi-Zheng, et al. "Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization." 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). IEEE, 2008.

[13] Cheng, Ran, Chaoli Sun, and Yaochu Jin. "A multi-swarm evolutionary framework based on a feedback mechanism." 2013 IEEE Congress on Evolutionary Computation. IEEE, 2013.

[14] Cheng, Ran, and Yaochu Jin. "A competitive swarm optimizer for large scale optimization." IEEE transactions on cybernetics 45.2 (2014): 191-204.

[15] Cheng, Ran, and Yaochu Jin. "A social learning particle swarm optimization algorithm for scalable optimization." Information Sciences 291 (2015): 43-60.

[16] Yang, Qiang, et al. "A distributed swarm optimizer with adaptive communication for large-scale optimization." IEEE transactions on cybernetics (2019).

[17] Van den Bergh, Frans, and Andries Petrus Engelbrecht. "A cooperative approach to particle swarm optimization." IEEE transactions on evolutionary computation 8.3 (2004): 225-239.

[18] Li, Xiaodong, and Xin Yao. "Cooperatively coevolving particle swarms for large scale optimization." IEEE Transactions on Evolutionary Computation 16.2 (2011): 210-224.

[19] Tang, Ruo-Li, Zhou Wu, and Yan-Jun Fang. "Adaptive multi-context cooperatively coevolving particle swarm optimization for large-scale problems." Soft Computing 21.16 (2017): 4735-4754.

[20] Chen, Wenxiang, et al. "Large-scale global optimization using cooperative coevolution with variable interaction learning." International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 2010:300-309.

[21] Mahdavi, Sedigheh, Mohammad Ebrahim Shiri, and Shahryar Rahnamayan. ”Cooperative co-evolution with a new decomposition method for large-scale optimization.” 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2014: 1285-1292.

[22] Omidvar, Mohammad Nabi, et al. ”Cooperative co-evolution with differential grouping for large scale optimization.” IEEE Transactions on evolutionary computation 18.3 (2013): 378-393.

[23] Sun, Yuan, Michael Kirley, and Saman Kumara Halgamuge. ”Extended differential grouping for large scale global optimization with direct and indirect variable interactions.” Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. 2015.

[24] Mei, Yi, et al. ”A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization.” ACM Transactions on Mathematical Software (TOMS) 42.2 (2016): 1-24.

[25] Omidvar, Mohammad Nabi, et al. ”DG2: A faster and more accurate differential grouping for large-scale black-box optimization.” IEEE Transactions on Evolutionary Computation 21.6 (2017): 929-942.

[26] Molina, Daniel, Manuel Lozano, and Francisco Herrera. ”MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization.” IEEE Congress on evolutionary computation. IEEE, 2010.

[27] Omidvar, Mohammad Nabi, et al. ”DG2: A faster and more accurate differential grouping for large-scale black-box optimization.” IEEE Transactions on Evolutionary Computation 21.6 (2017): 929-942.

[28] Peng, Xingguang, Yaochu Jin, and Handing Wang. ”Multimodal optimization enhanced cooperative coevolution for large-scale optimization.” IEEE transactions on cybernetics 49.9 (2018): 3507-3520.