MEL: Efficient Multi-Task Evolutionary Learning for High-Dimensional Feature Selection

Xubin Wang, Haojiong Shangguan, Fengyi Huang, Shangrui Wu and Weijia Jia*, *Fellow, IEEE*

APPENDIX

A. Representative Classic Meta-heuristic Optimization Algorithms

To demonstrate the effectiveness of our approach, we employed a total of 18 evolutionary computation algorithms in the first phase of the experiment. Following the classification methods outlined in the literature [1], we carefully selected four swarm-based methods, four nature-inspired methods, two evolutionary algorithms, four bio-stimulated methods, and four physics-based methods. These choices are depicted in Figure 1. These meta-heuristic optimization methods that we have chosen are considered to be the most classical, representative, and extensively utilized techniques in the field. In this section, we will provide a brief introduction to each of these methods.

1) **Artificial Bee Colony (ABC)** [2]: It simulates the biological behaviour of honeybees cooperate with each other to collect honey through individual division of labor and information exchange.
2) **Ant Colony Optimization (ACO)** [3]: ACO is inspired by the foraging behavior of ant colonies to find their way around food.
3) **Particle Swarm Optimization (PSO)** [4]: The concept of PSO arose from the study of bird feeding behavior. By simulating the behavior of bird flocks flying for food, the birds collaborate with one another to achieve the group's optimal goal.
4) **Monarch Butterfly Optimization (MBO)** [5]: MBO algorithm simulates the migration and adaptation behavior of monarch butterfly.
5) **Bat Algorithm (BAT)** [6]: BAT algorithm is a random search algorithm that simulates bats in nature using a kind of sonar to detect prey and avoid obstacles.
6) **Cuckoo Search Algorithm (CS)** [7]: CS algorithm is an optimization algorithm by simulating the incubation parasitism of cuckoos and Levy flight mechanism.
7) **Firefly Algorithm (FA)** [8]: FA is a heuristic algorithm for information exchange, mutual attraction and danger warning based on flashing behavior of fireflies.
8) **Flower Pollination Algorithm (FPA)** [9]: FPA simulates the process of plant cross-pollination by birds and bees using Levy's flight mechanism.
9) **Differential Evolution (DE)** [10]: DE is an intelligent optimization algorithm generated by the cooperation and competition between individuals within a group.
10) **Genetic Algorithm (GA)** [11]: GA is an optimization algorithm that simulates the natural selection and genetic mechanism of biological evolution.
11) **Fruit Fly Optimization Algorithm (FOA)** [12]: FOA mimics the process of fruit flies that use their keen sense of smell and vision to hunt.
12) **Grey Wolf Optimizer (GWO)** [13]: GWO is an optimization search method inspired by the prey hunting activities of grey wolves, which has strong convergence performance, few parameters and easy implementation.
13) **Harris Hawks Optimization (HHO)** [14]: HHO is an intelligent optimization algorithm that simulates the predatory behavior of the Harris Hawk.
14) **Whale Optimization Algorithm (WOA)** [15]: WOA mimics the hunting behavior of whales in nature and has the advantages of being easy to implement and having fewer parameters.
15) **Simulated Annealing (SA)** [16]: Simulated annealing algorithm comes from solid annealing principle and is based on Monte-Carlo iterative solution strategy. To avoid falling into local optimality, the search process is endowed with a time-varying probability of jumping to zero.
16) **Harmony Search (HS)** [17]: HS algorithm is a simulation of the process that musicians achieve the most beautiful harmony by repeatedly adjusting the tones of different instruments, so as to achieve the purpose of global optimization.
17) **Gravitational Search Algorithm (GSA)** [18]: GSA is an optimization method based on the law of gravitation and Newton's second law, which seeks the optimal solution through the continuous movement of gravitation between particles within a population.
18) **Multi Verse Optimizer (MVO)** [19]: MVO simulates the motion behavior of objects with high expansion rate tends to low expansion rate under the combined action of white holes, black holes and wormholes in the multiverse population, and tends to the optimal position in the search space by means of gravity.

Corresponding authors (*): Weijia Jia.

Xubin Wang, Shangrui Wu and Weijia Jia are with the Hong Kong Baptist University and the Institute of Artificial Intelligence and Future Networks, Beijing Normal University (Zhuhai campus). Haojiong Shangguan and Fengyi Huang are with Hong Kong Baptist University. E-mail: wangxb19@mails.jlu.edu.cn (Xubin Wang); shangguanhaojiong@uic.edu.cn (Haojiong Shangguan); huangfengyi@uic.edu.cn (Fengyi Huang); wushangrui@uic.edu.cn (Shangrui Wu); jiawj@bnu.edu.cn (Weijia Jia).

Manuscript received March 10, 2023; revised December 22, 2023.

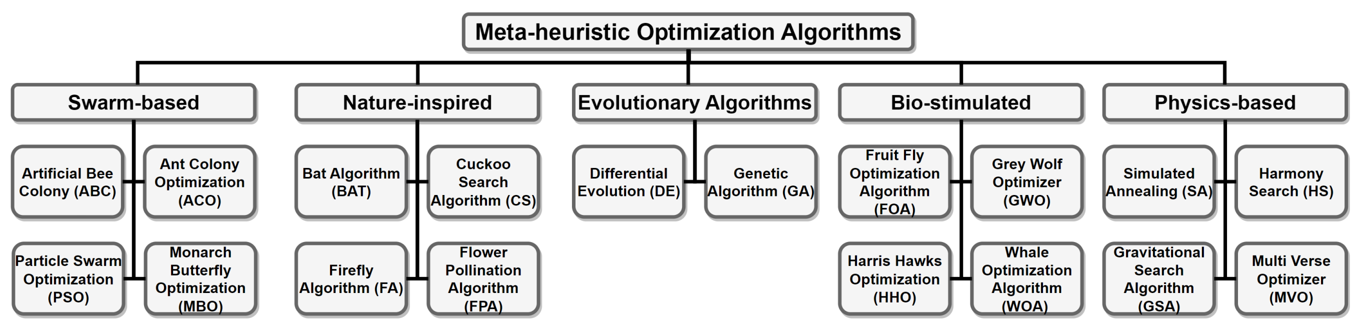

Fig. 1. Representative Classic Methods: 18 Meta-heuristic Optimization Algorithms.

B. Parameter Settings

We used the toolkit[1] for our experiments, and we followed the default settings provided in the toolbox for each method. In this section, we provide a detailed description of the parameters for each method, which can be referred to in Table I.

Methods	Parameters
ABC	lb = 0, ub = 1, $\theta = 0.6$, max_limit = 5.
ACO	tau = 1, eta = 1, alpha = 1, beta = 0.1, rho = 0.2.
PSO	lb = 0, ub = 1, $\theta = 0.6$, c1 = 2, c2 = 2, w = 0.9, Vmax =(ub - lb)/2.
MBO	lb = 0, ub = 1, $\theta = 0.6$, peri = 1.2, p = 5/12, Smax = 1, BAR = 5/12, num_land1 = 4, beta = 1.5.
BAT	lb = 0, ub = 1, $\theta = 0.6$, fmax = 2, fmin = 0, alpha = 0.9, gamma = 0.9, A_max = 2, r0_max = 1.
CS	lb = 0, ub = 1, $\theta = 0.6$, Pa = 0.25, alpha = 1, beta = 1.5.
FA	lb = 0, ub = 1, $\theta = 0.6$, alpha0 = 1, beta0 = 1, gamma = 1, calpha = 0.97.
FPA	lb = 0, ub = 1, $\theta = 0.6$, beta = 1.5, P = 0.8.
DE	lb = 0, ub = 1, $\theta = 0.6$, CR = 0.9, F = 0.5.
GA (Roulette)	CR = 0.8, MR = 0.01.
GA (Tournament)	CR = 0.8, MR = 0.01, Tour_size = 3.
FOA	lb = 0, ub = 1, $\theta = 0.6$.
GWO	lb = 0, ub = 1, $\theta = 0.6$.
HHO	lb = 0, ub = 1, $\theta = 0.6$, beta = 1.5.
WOA	lb = 0, ub = 1, $\theta = 0.6$, b = 1.
SA	c = 0.93, t0 = 100.
HS	lb = 0, ub = 1, $\theta = 0.6$, PAR = 0.05, HMCR = 0.7, bw = 0.2, NP = 20.
GSA	lb = 0, ub = 1, $\theta = 0.6$, G0 = 100, alpha = 20.
MVO	lb = 0, ub = 1, $\theta = 0.6$, p = 6, Wmax = 1, Wmin = 0.2.

TABLE I
PARAMETERS OF DIFFERENT EVOLUTIONARY ALGORITHMS

For ABC, max_limit is the maximum limits allowed. The tau, rat, alpha, beta and rho in ACO are the pheromone value, heuristic desirability, control pheromone, control heuristic and pheromone trail decay coefficient respectively. The c1, c2, w and Vmax in PSO are the cognitive factor, social factor, inertia weight and maximum velocity respectively. The peri, p, Smax, BAR, num_land1 and beta in MBO are the migration period, ratio, maximum step, butterfly adjusting rate, number of butterflies in land 1 and levy component respectively. For BAT, fmax is the maximum frequency, fmin is the minimum frequency, alpha and gamma are two constants, A_max is the maximum loudness and r0_max is the maximum pulse rate. The Pa, alpha and beta in CS are the discovery rate, constant and levy component respectively. The alpha0, beta0, gamma and calpha in FA are the constant, light amplitude, absorbtion coefficient and control alpha respectively. In FPA, beta is the levy component and P is the switch probability. In DE, CR is the crossover rate and F is the constant factor. For genetic algorithm (GA), we use Roulette Wheel and Tournament for selection, which have the same crossover rate (CR) and mutation rate (MR). Tournament also sets the tournament size (Tour_size). The beta in HHO is the levy component. The b in WOA is the constant. For SA, c is the cooling rate and t0 is the initial temperature. In HS, PAR, HMCR and bw are the pitch adjusting rate, harmony memory considering rate and bandwidth respectively. The G0 in GSA is the initial gravitational constant and the alpha is a constant. For MVO, P is the control TDR, Wmax is the maximum WEP and Wmin is the minimum WEP. The lb and ub in the parameter table are the lower boundary and the upper boundary.

In addition, for all algorithms, we set the number of populations (NP) to 20 and the maximum number of iterations (T) to 100. At the same time, we set a threshold value (θ) of 0.6 to convert the value of the real number field into a discrete value, so as to decide whether to choose the feature of the corresponding field. In this study, KNN classifier is employed with K equals to 3. Our classification accuracy is calculated using a five-fold cross-validation average. We performed each experiment ten times and averaged the data to make sure the experimental results were stable.

C. Converge Curves for Physics-based Heuristic Methods

We also conducted a comparison between MEL and four physical-based heuristic methods, namely Simulated Annealing (SA), Harmony Search (HS), Gravitational Search Algorithm (GSA), and Multi-Verse Optimizer (MVO). Figures 2 and 3 illustrate the convergence curves of these physical-based methods.

D. Supplementary Tables for Section V, Subsection G

This part provides the two tables from the seventh sub-section "Experiments with Larger Data Samples" of the "Results and Analysis" section, comparing the subset size and running time. The first table shows the comparison of the average subset sizes generated by different algorithms on 10 larger datasets. It reports the number of selected features by each

[1]https://github.com/JingweiToo/Wrapper-Feature-Selection-Toolbox

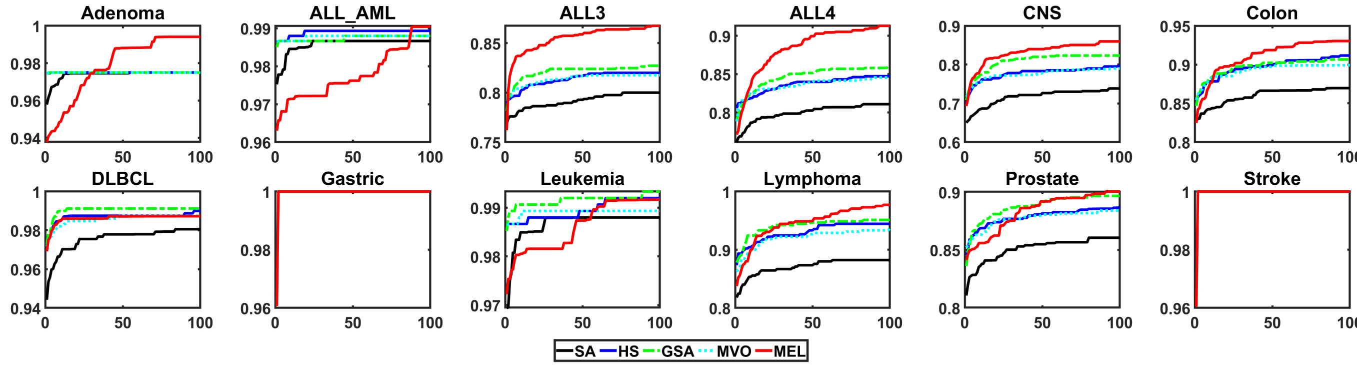

Fig. 2. Convergence Curves of Physics-based EC Algorithms in Terms of Accuracy

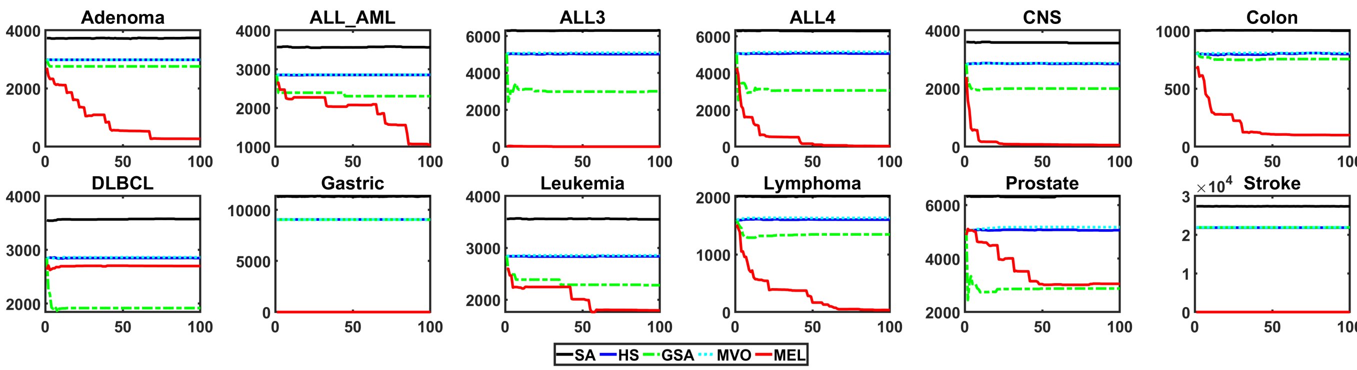

Fig. 3. Convergence Curves of Physics-based EC Algorithms in Terms of the Size of Feature Subset

method. The second table compares the average running times of the algorithms in seconds. It lists the execution time of each algorithm on each dataset. These two tables present a quantitative analysis of how MEL performs relative to other methods in terms of the parsimony of the selected feature subsets and computational efficiency, using larger real-world classification problems. They complement the classification accuracy results discussed in the previous sub-section.

Dataset	SaWDE	FWPSO	DENCA	PSO-EMT	MTPSO	MEL (Ours)
BASEHOCK	2247.9	**1.3**	1514.9	108.5	172.9	1953.9
COIL20	426.9	**3.3**	363.6	48.1	219.5	345.2
HAPTDataSet	255.8	**1.3**	177.3	20.7	79.1	217.2
Isolet	257.0	**1.0**	227.0	35.3	94.1	243.6
madelon	205.0	**1.8**	214.1	10.9	18.7	204.1
MultipleFeaturesDigit	258.5	**1.7**	199.1	18.2	107.6	240.1
Pancancer	770.4	**1.0**	799.7	-	261.6	708.4
PCMAC	1453.7	**1.0**	1083.7	40.9	162.0	1361.4
RELATHE	1929.9	**1.1**	1434.4	49.7	250.6	1746.9
USPS	139.8	**1.0**	97.9	-	38.7	78.9
Average	794.5	**1.5**	611.2	41.5	140.5	710.0

TABLE II
SUBSET SIZE COMPARISON ON DATASETS WITH LARGE SAMPLE SIZE

Dataset	SaWDE	FWPSO	DENCA	PSO-EMT	MTPSO	MEL (Ours)
BASEHOCK	**256.3**	1884.8	2884.9	372330.9	7489.4	862.4
COIL20	**35.2**	268.9	234.7	85504.1	2415.4	121.9
HAPTDataSet	**14.9**	127.9	90.2	22998.8	394.2	58.2
Isolet	**26.9**	221.5	132.3	82507.8	1165.3	105.6
madelon	**43.6**	3687.3	604.1	198703.8	1325.1	169.3
MultipleFeaturesDigit	**14.0**	121.1	78.8	19404.5	442.9	55.0
Pancancer	**543.0**	3693.6	8396.1	-	20684.9	1803.7
PCMAC	**157.3**	1241.6	2370.3	226997.1	5316.9	589.0
RELATHE	**121.3**	897.8	1733.5	164727.4	5028.1	434.3
USPS	**208.1**	1499.1	1788.9	-	14718.1	661.3
Average	**142.1**	1364.4	1831.4	119223.2	5898.0	486.1

TABLE III
RUNNING TIME COMPARISON ON DATASETS WITH LARGE SAMPLE SIZE

REFERENCES

[1] A. Kumar and S. Bawa, "A comparative review of meta-heuristic approaches to optimize the sla violation costs for dynamic execution of cloud services," *Soft Comput.*, vol. 24, no. 6, pp. 3909–3922, 2020.

[2] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *J. Glob. Optim.*, vol. 39, no. 3, pp. 459–471, 2007.

[3] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, "Text feature selection using ant colony optimization," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6843–6853, 2009.

[4] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)*. IEEE, 1998, pp. 69–73.

[5] G.-G. Wang *et al.*, "Monarch butterfly optimization," *Neural. Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019.

[6] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Springer, 2010, pp. 65–74.

[7] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. IEEE, 2009, pp. 210–214.

[8] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *arXiv preprint arXiv:1003.1409*, 2010.

[9] X.-S. Yang, M. Karamanoglu, and X. He, "Flower pollination algorithm: a novel approach for multiobjective optimization," *Eng. Optim.*, vol. 46, no. 9, pp. 1222–1237, 2014.

[10] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[11] C.-L. Huang and C.-J. Wang, "A ga-based feature selection and parameters optimizationfor support vector machines," *Expert Syst. Appl.*, vol. 31, no. 2, pp. 231–240, 2006.

[12] W.-T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowl. Based Syst.*, vol. 26, pp. 69–74, 2012.

[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[14] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019.

[15] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.

[16] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[17] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[18] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: a gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[19] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," *Neural. Comput. Appl.*, vol. 27, no. 2, pp. 495–513, 2016.