

LETTER

Advanced Antlion Optimizer with Discrete Ant Behavior for Feature Selection

Mengmeng LI[†], Student Member, Xiaoguang REN^{†,††}, Yanzhen WANG^{†,††}, Wei QIN^{†,††},
and Yi LIU^{†,††a)}, Nonmembers

SUMMARY Feature selection is important for learning algorithms, and it is still an open problem. Antlion optimizer is an excellent nature inspired method, but it doesn't work well for feature selection. This paper proposes a hybrid approach called Ant-Antlion Optimizer which combines advantages of antlion's smart behavior of antlion optimizer and ant's powerful searching movement of ant colony optimization. A mutation operator is also adopted to strengthen exploration ability. Comprehensive experiments by binary classification problems show that the proposed algorithm is superiority to other state-of-art methods on four performance indicators.

key words: antlion optimizer, ant colony optimization, feature selection, mutation

1. Introduction

Feature selection is a necessary step in artificial intelligence applications, which could improve the performance of learning models and algorithms significantly [1]. Antlion optimizer (ALO) is a new powerful nature-inspired algorithm that simulates the foraging behavior of antlion [2]. However, it is not fit for solving feature selection problem, as it is a continuous approach and feature selection is a discrete issue [3]. Emary et al. developed several binary versions of ALO called BALO-1, BALO-S and BALO-V to extend ALO to feature selection field [4]. BALO-1 adopted crossover operation between two binary individuals, and performed mutation as a local search around ant lions. BALO-S and BALO-V used *sigmoidal* and *tan* functions to convert continuous values into the corresponding binary solutions. This paper proposes a powerful hybrid modified ALO called Ant-Antlion Optimizer (AAO), which utilizes ant's forceful searching capability in Ant Colony Optimization (ACO) and antlion's sharp hunting skill [5]. AAO employs the discrete way of ant's movement, which finds a path by pheromone values on edges, and still reserves the predation procedure of antlion. And it is the main difference between AAO and BALOs. Besides, mutation operator is employed when AAO gets into local optima to improve its performance. Exhaustive experiments through six binary classification datasets and four measures show the amazing

optimization power of AAO compared with other excellent methods.

2. Description of AAO

ALO has five steps, i.e. random walk of ants, building traps of antlions, falling into traps, catching ants, and rebuilding traps [2]. The walking behavior of ants in ALO is a process where the value changes continuously in source space. However, ALO is not fit for feature selection problem, as it is a combinatorial optimization issue and continuous algorithms are hard to generate effective individuals. ACO is a discrete method, and it has an excellent ability to resolve feature selection problem [6]. Ants select edges one by one in ACO to find sub optimal solutions (paths), which is different from the behavior of ants in ALO.

AAO replaces the action of ants in ALO with the movement way of ants in ACO, which utilizes the clever predation process of ALO and the strong searching ability of ants in ACO simultaneously. The detailed pseudo code of AAO is given in algorithm 1.

AAO uses pheromone matrix to record solution instead of individual itself as the way ACO does. Line one initializes antlions through their initial pheromone matrices and

Algorithm 1 Pseudo-Code of AAO

Input: Population size N , max iterations M , selected features size S , heuristic information.

Output: Sub optimal feature subset.

1. Initialize antlions, calculate their fitness, and update their corresponding pheromone matrices;
2. Regard the best antlion as the elite one;
3. **While** not meet stopping criterion
4. **For** Each ant
5. Select pheromone matrix of an antlion using roulette wheel;
6. Combine the pheromone matrix with pheromone matrix of the elite one;
7. Ant searches a path (solution) based on the combination of pheromone matrix and heuristic information;
8. Calculate the fitness of the new path;
9. Update the pheromone matrix of the selected antlion according to the fitness;
10. **End For**
11. **If** current solutions are not better than elite one
12. Choose the best 1/3 antlions to implement mutation and update their corresponding pheromone matrices;
13. **End If**
14. Update the elite one;
15. **End While**

Manuscript received April 13, 2020.

Manuscript revised August 10, 2020.

Manuscript publicized September 4, 2020.

[†]The authors are with National Innovation Institute of Defense Technology, Beijing, 100010, China.

^{††}The authors are with Tianjin Artificial Intelligence Innovation Center, Tianjin, 300457, China.

a) E-mail: albertliu20th@163.com

DOI: 10.1587/transinf.2020EDL8055

heuristic information, and its procedure is the same as line seven. Line two chooses the best antlion as the elite one. Line three to line fifteen is the main loop of AAO. Line four to line ten is the searching process of an ant. Line five uses roulette wheel to select a pheromone matrix of an antlion. Line six combines the chosen pheromone matrix with pheromone matrix of the elite one to form a new pheromone matrix. Line seven is the process of an ant searches for a path according to transition probability computed by pheromone matrix and heuristic information. Line eight calculates the fitness of the new generated path. Line nine updates the pheromone matrix of the chosen antlion based on its fitness. Line eleven to line thirteen is the strength part of AAO, which makes the best 1/3 antlions to implement mutation, which is the same as the mutation operator in genetic algorithm [7] to obtain better solutions when current solutions are not better. Line fourteen updates the matrix of elite one through better antlions.

The approach of new pheromone matrix generated in line six is described as Eq. (1)

$$phma_n = \omega_c \cdot phma_c + \omega_e \cdot phma_e \quad (1)$$

where $phma_n$ stands for the new pheromone matrix, $phma_c$ and $phma_e$ represent pheromone matrices of chosen antlion and elite one, respectively. ω_c and ω_e indicate the corresponding weights, and we set their values to 1 in this paper.

The transition probability computed in line seven is calculated as Eq. (2)

$$p_i^k = \begin{cases} \frac{[\tau_i]^\alpha \cdot [\eta_i]^\beta}{\sum_{f_j \notin visit_k} [\tau_j]^\alpha \cdot [\eta_j]^\beta} & f_j \notin visit_k \\ 0 & otherwise \end{cases} \quad (2)$$

where p_i^k denotes the probability that ant k selects feature f_i , τ_i is the pheromone value of feature f_i at current iteration, η_i is the statistic expectation heuristic information of selecting feature f_i . $visit_k$ denotes the features which are visited by ant k . α and β are constants to control the relative importance of the pheromone versus the heuristic information.

The update method for the pheromone matrix of selected antlion in lines nine and twelve is shown in Eq. (3)

$$\tau_i(t) = \begin{cases} (1 - \rho)\tau_i(t-1) + F(path^t)/Q & f_i \in path^t \\ (1 - \rho)\tau_i(t-1) & otherwise \end{cases} \quad (3)$$

where ρ is evaporation rate, $path^t$ is the better route at current iteration, $F(path^t)$ is fitness value, and Q is a scaling factor.

3. Experiments and Results

This part compares AAO with other state-of-art algorithms to evaluate its performance, i.e. Particle Swarm Optimization (PSO) [8], Brain Storm Optimization (BSO) [9], ALO and ACO for a comprehensive comparison. Besides, as the two binary ALO approaches, i.e. BALO-1 and BALO-V, have amazing ability according to their experiments, we employ them to compare with our algorithm. We use Preci-

sion, Recall, F1, and classification success rate (Accuracy) indicators to evaluate experiment's results. We adopt Windows 10 operating system, Matlab 2018a, Intel i5-7800HQ, 16 GB RAM as test platform. Six binary classification datasets from UCI machine learning repository [10] are selected for experiments, and their characteristics are shown in Table 1.

In Table 1, instances represent the number of records, dimensions denote for the number of features in each dataset, and features stand for the selected number of features by algorithms for user's preference.

The parameters of AAO are set as follows: population size is 40, the maximum number of iterations is 200, $\rho = 0.05$, $\alpha = 1$ and $\beta = 2$, $Q = 0.1$. Initial pheromone value is 100, mutation probability is 0.1. Besides, we use Fisher discriminant rate of features as the heuristic information of AAO, which is shown in Eq. (4)

$$f(i) = \frac{(\mu_{i1} - \mu_{i2})^2}{\delta_{i1}^2 + \delta_{i2}^2} \quad (4)$$

where μ_{i1} and μ_{i2} represent the mean of the feature f_i in the two classes, respectively. δ_{i1}^2 and δ_{i2}^2 indicate the variance of the feature f_i in the two classes, respectively.

The parameters of ALO, PSO, BSO, BALO-1 and BALO-V are the same as those in their corresponding references, and the parameters of ACO which are common with AAO are set the same as those in AAO. Furthermore, population size and maximum number of iterations in compared algorithms are identical with AAO for a fair situation. We adopt fivefold cross validation for each dataset, and every algorithm which uses F1 as the optimization objective is executed 20 times independently to obtain an average value as a result. K -nearest neighbor classifier is employed to classify data, K is set to 5, and Euclidean distance is used here. Table 2 to Table 7 give four index values of every method on each dataset.

From Table 2, we can find out that AAO is better than other six algorithms in the view of Recall, F1 and Accuracy, while AAO gives best values of Precision, F1, and Accuracy

Table 1 Characteristics of datasets.

Datasets	Instances	Dimensions	Features
Sonar	208	60	19
Statlog (heart)	270	13	5
Planning Relax	182	13	8
Ionosphere	351	34	10
Z-Alizadeh Sani	303	56	13
Climate	540	18	7

Table 2 Results on sonar.

Algorithm	Precision	Recall	F1	Accuracy
AAO	0.9465	0.9600	0.9505	0.9566
ALO	0.9682	0.9111	0.9382	0.9469
PSO	0.9473	0.9467	0.9467	0.9518
BSO	0.9109	0.9578	0.9325	0.9373
ACO	0.9068	0.9251	0.9145	0.9229
BALO-1	0.9493	0.8928	0.9175	0.9228
BALO-V	0.9488	0.8690	0.9058	0.9086

Table 3 Results on statlog (heart).

Algorithm	Precision	Recall	F1	Accuracy
AAO	0.8895	0.9346	0.9109	0.8963
ALO	0.7884	0.9623	0.8647	0.8519
PSO	0.8037	0.9138	0.8541	0.8296
BSO	0.7898	0.9251	0.8514	0.8259
ACO	0.8146	0.9007	0.8542	0.8333
BALO-I	0.7967	0.9377	0.8603	0.8370
BALO-V	0.8403	0.9312	0.8821	0.8667

Table 4 Results on planning relax.

Algorithm	Precision	Recall	F1	Accuracy
AAO	0.7917	0.9929	0.8803	0.8021
ALO	0.7397	0.9716	0.8348	0.7351
PSO	0.7465	0.9917	0.8511	0.7622
BSO	0.7465	0.9917	0.8511	0.7622
ACO	0.7556	0.9917	0.8571	0.7731
BALO-I	0.7827	0.9714	0.8639	0.7745
BALO-V	0.8011	0.9502	0.8656	0.7805

Table 5 Results on ionosphere.

Algorithm	Precision	Recall	F1	Accuracy
AAO	0.9720	1	0.9856	0.9800
ALO	0.9068	0.9821	0.9423	0.9257
PSO	0.8761	0.9856	0.9274	0.9095
BSO	0.8766	0.9856	0.9276	0.9095
ACO	0.8869	0.9810	0.9312	0.9151
BALO-I	0.9037	0.9872	0.9432	0.9202
BALO-V	0.8877	0.9956	0.9381	0.9175

Table 6 Results on Z-alizadeh sani.

Algorithm	Precision	Recall	F1	Accuracy
AAO	0.8914	0.9809	0.9338	0.9004
ALO	0.8296	0.9677	0.8924	0.8350
PSO	0.8546	0.9720	0.9088	0.8609
BSO	0.8427	0.9601	0.8959	0.8406
ACO	0.8490	0.9712	0.9059	0.8572
BALO-I	0.7845	0.9688	0.8666	0.7829
BALO-V	0.7867	0.9727	0.8688	0.7904

Table 7 Results on climate.

Algorithm	Precision	Recall	F1	Accuracy
AAO	0.9596	1	0.9794	0.9611
ALO	0.9383	0.9980	0.9672	0.9389
PSO	0.9365	1	0.9671	0.9389
BSO	0.9366	1	0.9671	0.9389
ACO	0.9437	1	0.9709	0.9463
BALO-I	0.9346	1	0.9659	0.9352
BALO-V	0.9364	1	0.9669	0.9370

on Statlog (heart). Furthermore, AAO provides mostly all the best results on Planning Relax, Ionosphere, Z-Alizadeh Sani and Climate, which are shown in Tables 4–7. However, BALO-V has a higher Precision on Planning Relax. It is demonstrated that AAO has a better feature selection performance compared with other six algorithms based on above analysis. On the one hand, AAO is better than BALO-I and BALO-V, which shows that the hybrid way of AAO can produce a better searching ability for feature selection. On the other hand, AAO is more excellent than ALO and ACO as it gives higher indicator values except Precision on Sonar and Recall on Statlog (heart), which shows that our method is effective. At last, we test time cost of seven algorithms on six datasets, and the results are shown in Table 8.

In Table 8, bold values indicate the shortest time overhead. We can find that PSO takes less time than other compared algorithms in most cases, while ACO runs faster on Statlog (heart) and Climate. AAO takes more time than other methods especially ALO. It is because that when current solutions are not better than elite one up to now, AAO may be considered to fall into local optima and should execute mutation operations by the best 1/3 antlions, which improves approach's exploration ability but consumes more time partly. Furthermore, AAO may execute mutation operations more times if the problems are harder to optimize, which could take more time but probably find better solutions.

4. Conclusions

A new algorithm called AAO is proposed to resolve feature selection problems. AAO utilizes ant's excellent searching capability in ACO and antlion's fantastic hunting skill in ALO simultaneously. Furthermore, a mutation operator is also adopted in AAO to improve its performance when it gets into local optima. Experiments prove the effectiveness of the algorithm as it provides higher indicator values than ALO and ACO. Besides, AAO is also better than other compared methods in terms of searching aspect. Next, we will further reduce the time overhead of AAO and generalize it to practical engineering problems.

Acknowledgments

This work was partially supported by the National

Table 8 Time cost (s).

Dataset	Sonar	Statlog (heart)	Planning Relax	Ionosphere	Z-Alizadeh Sani	Climate
AAO	49.42	51.85	51.32	53.78	49.70	55.13
ALO	44.88	41.07	40.36	44.79	44.20	44.59
PSO	37.25	39.72	39.15	40.97	37.88	42.62
BSO	37.68	40.19	39.41	43.60	38.09	42.68
ACO	38.14	39.62	39.26	41.20	38.62	41.34
BALO-I	38.92	40.01	39.17	38.38	38.66	41.73
BALO-V	45.32	42.96	41.56	52.03	45.55	44.20

Key Research and Development Program of China (No. 2017YFB1001905), the National Natural Science Foundation of China (No. 91648204 and 61532007), the National Science Foundation for Young Scientists of China (Grant No. 61802426), and the National Science and Technology Major Project.

References

- [1] X.-L. Deng, Y.-Q. Li, J. Weng, and J.-L. Zhang, "Feature selection for text classification: A review," *Multimedia Tools and Applications*, vol.78, no.3, pp.3797–3816, 2019, doi: 10.1007/s11042-018-6083-5.
- [2] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol.83, pp.80–98, 2015, doi: 10.1016/j.advengsoft.2015.01.010.
- [3] Y. Liu, Y.-Z. Wang, X.-G. Ren, H. Zhou, and X.-C. Diao, "A classification method based on feature selection for imbalanced data," *IEEE Access*, vol.7, pp.81794–81807, 2019, doi: 10.1109/ACCESS.2019.2923846.
- [4] E. Emary, H.M. Zawbaa, and A.E. Hassanien: "Binary ant lion approaches for feature selection," *Neurocomputing*, vol.213, pp.54–65, 2016, doi: 10.1016/J.NEUCOM.2016.03.101.
- [5] M. Dorigo and G.D. Caro, "Ant colony optimization: A new meta-heuristic," *Proc. 1999 Congress on Evolutionary Computation*, CEC99, vol.2, pp.1470–1477, 1999, doi: 10.1109/CEC.1999.782657.
- [6] R.J. Manoj, M.D.A. Praveena, and K. Vijayakumar, "An ACO-ANN based feature selection algorithm for big data," *Cluster Computing*, vol.22, pp.3953–3960, 2019, doi: 10.1007/s10586-018-2550-z.
- [7] O. Kramer, "Genetic algorithm essentials," *Studies in Computational Intelligence*, pp.13–15, 2017, doi: 10.1007/978-3-319-52156-5.
- [8] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. ICNN '95 International Conference on Neural Networks*, vol.4, pp.1942–1948, 1995, doi: 10.1109/ICNN.1995.488968.
- [9] Y.-H. Shi, "An optimization algorithm based on brainstorming process," *International Journal of Swarm Intelligence Research*, vol.2, no.4, pp.35–62, 2011, doi: 10.4018/ijisir.2011100103.
- [10] <http://archive.ics.uci.edu/ml/datasets.php>