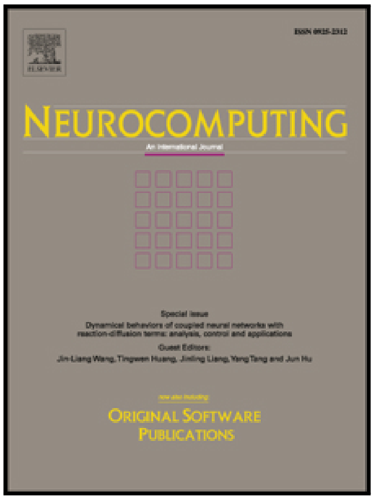
Communicated by Dr W Gao

Accepted Manuscript



A Two-Stage Imitation Learning Framework for the Multi-Target Search Problem in Swarm Robotics

Jie Li, Ying Tan

PII: S0925-2312(19)30055-4
DOI: https://doi.org/10.1016/j.neucom.2019.01.035
Reference: NEUCOM 20339

To appear in: *Neurocomputing*

Received date: 31 March 2018
Revised date: 5 November 2018
Accepted date: 16 January 2019

Please cite this article as: Jie Li, Ying Tan, A Two-Stage Imitation Learning Framework for the Multi-Target Search Problem in Swarm Robotics, *Neurocomputing* (2019), doi: https://doi.org/10.1016/j.neucom.2019.01.035

Highlights

- For the multi-target search problem in swarm robotics, most existing work is about strategic design while this article focuses on strategy imitation.
- To our knowledge, it is the first time that the combination of deep learning technologies and evolutionary algorithms is used for the problem.
- The strategy obtained from the framework is close to the target strategy on multiple indicators.
- The two-stage imitation learning framework can also be used for other swarm tasks.
- The network design and evolutionary algorithm settings can be a good reference.

A Two-Stage Imitation Learning Framework for the Multi-Target Search Problem in Swarm Robotics

Jie Li[a,1], Ying Tan[a,]

[a]*Key Laboratory of Machine Perception (Ministry of Education), Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, P.R. China.*

Abstract

As a distributed system with a large number of individuals, swarm robotics is particularly suitable for multi-target search problems. Most existing work is about strategic design while this article focuses on strategy imitation. Sometimes we can observe the behavior of individuals and obtain a large amount of data, but we do not know the specific details of the strategy behind the behavior. Imitating the self-organizing behavior of organisms is of great significance for us to design e□cient swarm strategies and to reveal the underlying mechanisms. The actual strategy adopted by individuals can be called the target strategy, and in this article, a two-stage imitation learning framework is proposed to approach the target strategy. In the first stage, a deep neural network is trained using the behavioral data of individuals, and in the second stage, the parameters of the neural network are further fine-tuned using the evolutionary algorithm. After two stages of learning and evolution, the resulting strategy RNSE is very close to the target strategy in terms of multiple indicators, including search e□ciency, stability, parallel processing capability, and collaborative processing capability. In addition to multi-target search, the framework can also be used for other collective tasks such as aggregation and dispersion. In this paper, the design of neural networks and the settings of the evolutionary algorithm are

*Corresponding author
Email address: ytan@pku.edu.cn (Ying Tan)
[1]Email address: ustblijie@126.com

discussed in detail, which is of great significance for the migration application of the framework.

Keywords: Swarm robotics, Multi-target search, Swarm intelligence optimization, Deep learning, Evolutionary algorithm, Strategy imitation

1. Introduction

Swarm robotics is a relatively new field, inspired by natural self-organizing behaviors [1] such as ants foraging, bees nesting, and bird migration. In these swarms, the ability of each individual is simple, but through local interactions among individuals and between individuals and the environment, the system can emerge complex behaviors at the swarm level. Compared to single robot or multi-robot systems, swarm robotics has some unique advantages, such as robustness, scalability, and flexibility [2, 3]. In addition, due to the simplicity and low cost of the individuals, swarm robotics has potential applications in many fields, such as post-disaster rescue [4], intrusion detection [5], human interaction [6], and planetary exploration [7], etc.

As a distributed system, swarm robotics is very suitable for tasks involving area coverage, such as collaborative mapping [8], target search [9], area monitoring [10], and so on. And the multi-target search problem of swarm robotics studied in this paper is also such kind of a task [11], in which the entire swarm can search multiple targets in parallel, thereby improving the search e□ciency. Some practical scenarios can be abstracted as the problem, such as shipwreck rescue, submarine search, search and destruction of battlefield targets.

There are various research directions in swarm robotics [12], one of which is the system design. There are roughly two types of design approaches [13]. One is the behavior-based approach, which is the most commonly used. It is a bottom-up approach, achieving the desired collective behavior through the realization, research and improvement of individual behaviors iteratively. And the specific methods including probability finite state machine [14], artificial potential field [15], etc. The other is the learning-based approach (also called

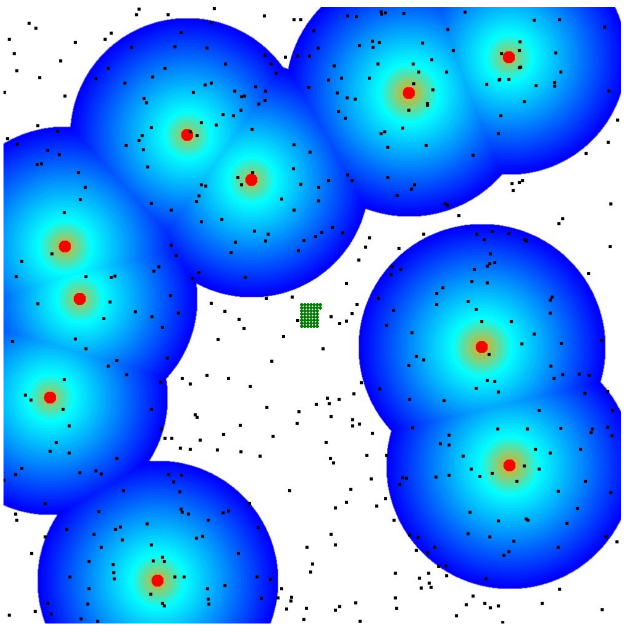


Figure 8: A screenshot of the problem with obstacles at the beginning of a simulation. Black squares stand for the obstacles. Red circles stand for the targets while the central green dot array represents the initial robot swarm.

are represented by black squares (each with a size of 1 × 1). In this paper, a simple avoiding strategy is applied for search strategies. In this strategy [11], the robot will check if it will run into obstacles with current velocity. If so, a small repulsive force perpendicular to the velocity from the obstacle will be added to avoid the collision.

As is shown in Fig. 9, the mI of search strategies in the environment with small obstacles are similar to that in Fig. 6. Although the mI in obstructive environments is a little higher, the overall trend of curves and relative rankings of different search strategies remain unchanged. Therefore, small obstacles basically do not affect the relative performance of search strategies.

5.3. Different Number of Targets

In this section, the search efficiency of various comparison algorithms under different number of targets is investigated. In experiments, each strategy is

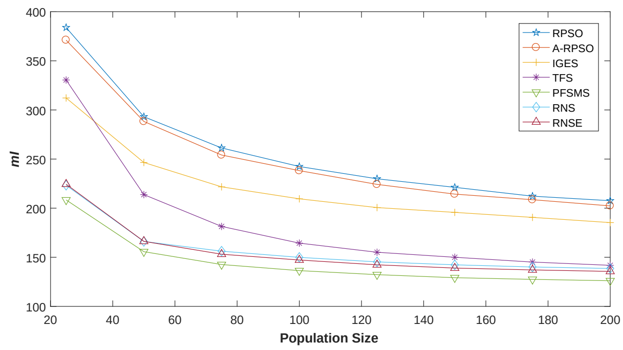


Figure 9: ml of search strategies with various population sizes in the environment with small obstacles.

tested at the target number of 1, 5, 10, 15, 20, 30, 40, 50 in turn, and the map size is 1000*1000, containing 50 robots. The experimental results are shown in Tab. 5, Fig. 10 and Fig. 11.

The Significance of RNS and RNSE: As is shown in Tab 5 and Fig. 10, the ml s of RNS and RNSE are close to that of PFSMS, and the performance of RNSE is slightly better. In addition, under different target numbers, two-side Wilcoxon rank sum tests (with confidence level 95%) are conducted on the RNS and other algorithms, as well as on the RNSE and other algorithms, which are based on the data of 1000 runs. Statistical tests show that there is no significant difference between TFS, RNS, and RNSE when the target number is 1. And there is no significant difference between RNS and RNSE when the target number is 5. In other cases, RNS is significantly better than other algorithms except for PFSMS and RNSE, and RNSE is significantly better than other algorithms except PFSMS. With the increase in the number of targets, the advantages of RNS and RNSE gradually emerge, showing excellent parallel search capabilities.

As is shown in Tab. 5 and Fig. 10, when there is only one target, all search strategies have similar performance, and there is no statistically significant difference in the performance of TFS, RNS, and RNSE. The excellent local exploitation capability of TFS makes up for its lack of global exploration capabilities. From the results, the performance of RNS and RNSE is especially close,

Table 5: ml and dl of search strategies with various numbers of targets and 50 robots.

Targets	RPSO		A − RPSO		IGES		TFS		PFSMS		RNS		RNSE	
	ml	dl	ml	dl	ml	dl	ml	dl	ml	dl	ml	dl	ml	dl
1	96.0	36.7	98.2	39.0	105.3	43.3	93.8	45.5	84.8	29.3	87.6	32.9	87.3	32.7
5	187.4	46.2	188.5	46.3	180.4	33.8	164.7	52.4	125.4	20.0	133.7	25.2	132.9	24.0
10	289.2	74.7	283.3	71.6	229.1	40.2	211.0	56.0	147.6	20.8	157.6	24.2	155.4	23.2
15	380.0	98.6	366.2	87.6	271.9	41.2	244.3	63.2	163.5	20.0	177.6	27.3	174.1	28.1
20	434.9	97.9	414.1	86.7	302.5	49.2	263.4	63.9	175.7	20.4	195.2	34.5	187.1	26.5
30	548.6	108.7	505.3	93.8	354.0	65.7	306.3	66.2	202.1	25.7	226.9	41.8	215.2	32.8
40	629.0	104.5	563.0	95.0	409.4	60.9	344.3	70.8	223.3	28.3	255.6	45.6	241.3	34.1
50	704.1	121.4	609.2	112.5	436.7	68.3	367.2	67.2	236.5	29.2	284.2	51.2	263.8	38.8

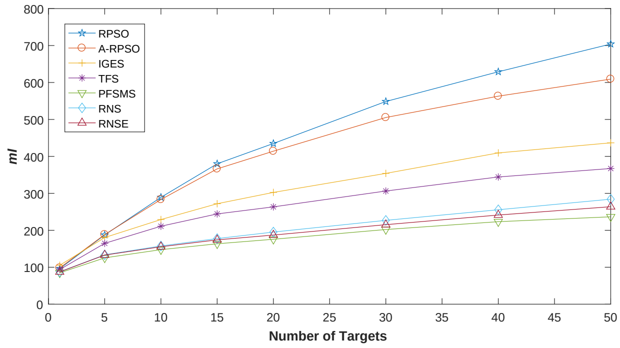


Figure 10: ml of search strategies with various number of targets.

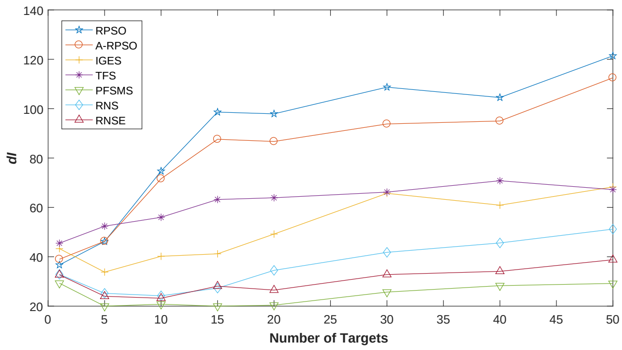


Figure 11: dl of search strategies with various number of targets.

which means that the two strategies have similar local exploitation capabilities, and the diﬀerence in performance lies mainly in the parallel search of multiple targets.

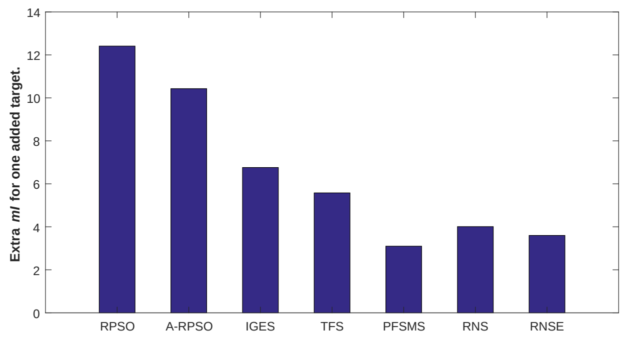


Figure 12: Extra ml required for one added target of different search strategies.

As is shown in Fig. 10, the slope of each curve represents the number of extra iterations needed for one additional target, so a small slope means the algorithm has better parallelism. For each strategy (RPSO, A-RPSO, IGES, TFS, PFSMS, RNS, and RNSE), the slopes of the curves are 12:41, 10:43, 6:76, 5:58, 3:10, 4:01, and 3:60, respectively, as is shown in Fig. 12. RNS and RNSE have similar parallelism with PFSMS, while RNSE is slightly better than RNS. Considering that RNS and RNSE have similar local exploitation capabilities, the evolutionary stage mainly enhances the parallel search capability of the strategy. According to parallelism, the strategies can be sorted as $PFSMS > RNSE > RNS > TFS > IGES > A - RPSO > RPSO$.

With different number of targets, the stability of each search strategy is shown in Fig. 11. As the number of targets increases, the curves of IGES, RNS, RNSE, and PFSMS drop first and then increase, which means that a proper increase in targets helps to reduce the randomness of the problem. As can be seen from the curve trend, compared with other algorithms, the stability of TFS, RNSE and PFSMS is less sensitive to the number of targets. On the other hand, as the number of targets increases, the stability of PRSO and A-RPSO significantly deteriorates, indicating that they are not very good at parallel search for multiple targets.

As is shown in the experimental results, with different number of targets, RNS and RNSE are very close to the target strategy in efficiency and stability, and the performance of RNSE is slightly better than RNS.

5.4. Different Collection Times of Targets

As described in the problem statement, a target requires 10 steps to be processed, which can be done by a single robot in 10 iterations, or 10 robots in one iteration. The “collection times of targets” refers to the number of iterations that a robot needs to collect a target. If the “collection times of targets” is 50, then it takes a robot 50 iterations to collect one target. With different collection times of targets, collaborative collection capabilities of the swarm in different strategies can be investigated, which is complementary to the parallel search capabilities of the swarm. In this section, the search efficiency of various comparison algorithms under different collection times of targets is investigated. In experiments, each strategy is tested at the collection time of 1, 5, 10, 15, 20, 30, 40, 50 in turn, and the map size is 1000*1000, containing 10 targets and 50 robots. The experimental results are shown in Tab. 6, Fig. 13 and Fig. 14.

The Significance of RNS and RNSE: As is shown in Tab 6 and Fig. 13, the mI s of RNS and RNSE are close to that of PFSMS, and the performance of RNSE is slightly better. In addition, under different collection times of targets, two-side Wilcoxon rank sum tests (with confidence level 95%) are conducted on the RNS and other algorithms, as well as on the RNSE and other algorithms, which are based on the data of 1000 runs. Statistical tests show that there is no significant difference between RNS and RNSE when the collection times of targets are 30, 40 and 50. In other cases, RNS is significantly better than other algorithms except for PFSMS and RNSE, and RNSE is significantly better than other algorithms except PFSMS.

As shown by Fig. 13, as the collection time of targets increases, the mI for different search strategies grows approximately linearly. With different collection times of targets, the relative ranking of the efficiency of each strategy has not changed, and the performance of RPSO gradually approaches A-RPSO, indicating that the former has a stronger collaborative collection capability.

Table 6: ml and dl of search strategies with various collection times of targets.

Collection Times	RPSO		A − RPSO		IGES		TFS		PFSMS		RNS		RNSE	
	ml	dl	ml	dl	ml	dl	ml	dl	ml	dl	ml	dl	ml	dl
1	276.9	75.2	263.3	66.7	211.2	36.5	198.6	54.8	137.2	18.8	145.8	22.6	142.2	20.5
5	286.7	77.4	275.4	70.5	221.3	37.6	207.3	58.7	142.8	19.3	153.0	24.4	149.4	24.0
10	289.4	75.5	283.1	71.0	230.6	39.3	210.3	53.8	147.9	20.6	158.2	24.5	155.1	23.9
15	296.3	74.9	286.6	69.2	235.7	40.5	215.5	59.4	153.1	22.2	163.6	24.2	161.0	23.9
20	298.1	74.3	294.7	72.8	244.8	42.1	219.8	55.8	157.1	22.0	168.3	25.8	164.2	22.8
30	305.8	76.9	302.7	70.8	256.3	44.7	227.5	55.4	165.8	23.2	176.8	27.3	174.0	24.7
40	313.6	77.2	311.7	70.9	265.2	45.0	232.6	55.5	173.9	24.9	185.4	30.3	181.7	25.5
50	320.6	77.3	319.8	73.7	276.0	47.5	242.6	65.4	178.9	23.7	191.7	29.1	189.5	27.0

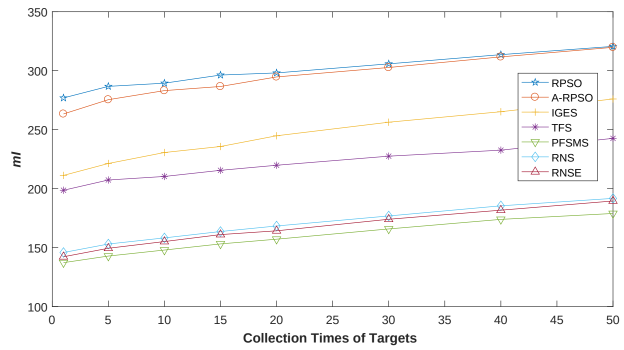


Figure 13: ml of search strategies with various collection times of targets.

Figure 14: dl of search strategies with various collection times of targets.

The slope of each curve in the figure indicates the number of extra iterations needed for one additional collection time of targets, indicating the collaborative

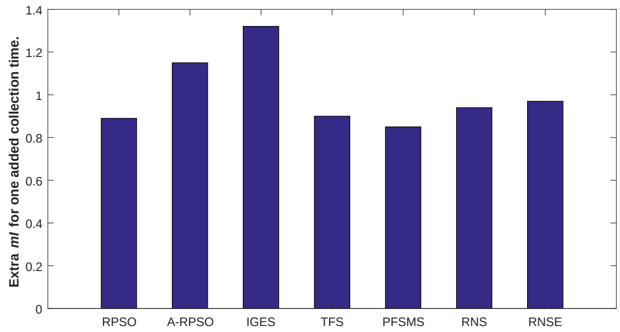


Figure 15: Extra *ml* required for one added collection time of different search strategies.

collection ability of the swarm, thus a small slope means that the search strategy has better collaborative processing capabilities. For each strategy (RPSO, A-RPSO, A-RPSO, IGES, TFS, PFSMS, RNS, and RNSE), the slope of the curve is 0:89, 1:15, 1:32, 0:90, 0:85, 0:94, 0:97, as is shown in Fig. 15. According to the collaborative processing capability, the strategies can be sorted as: $PFSMS > PRSO \approx TFS > RNS > RNSE > A - RPSO > IGES$. Compared with RPSO, niche technology limits the swarm collaboration of A-RPSO and reduces the ability of collaborative processing. In TFS, three-robot formations allow a single target to be processed by at least three robots, resulting in good collaborative collection capabilities. IGES tends to allow the group to spread out slowly, weakening the collaboration among individuals. Compared with the target strategy PFSMS, the RNS and RNSE have a weaker ability for collaborative processing, and the RNSE is slightly weaker than RNS, which means that the evolution of the second stage slightly sacrificed the swarm's collaborative processing capability to enhance parallel search capabilities.

As is shown in Fig. 14, the stability of each strategy is less sensitive to the collection times of targets. The stability of RNS and RNSE is close to PFSMS, which is significantly better than other comparison algorithms. Because this article focuses on the search efficiency of the strategy, rather than the collaborative processing, the collection times of targets in the problem is small (10 times), and only linear acceleration is considered (the collection efficiency of the target is proportional to the number of participating robots), so the performance improvement brought by good collaborative collection capability is not significant. If the collection times of targets increases significantly, the performance of various strategies may be different.

As is shown in the experimental results, with different collection times of targets, RNS and RNSE are very close to the target strategy in efficiency and stability, and the performance of RNSE is slightly better than RNS.

6. Conclusions

The existing multi-target search strategies in swarm robotics are generally behavior-based, that is, the multi-target search behavior of the swarm is achieved by designing individual behaviors. The purpose of this article is not to design new strategies, but to approach a target strategy through imitation and evolution. In this paper, it is assumed that the specific details of the target strategy are unknown, but the individual robot's behavioral data and performance metric can be obtained. This paper proposes a two-stage imitation learning framework to approach the target strategy. In the first stage, the neural network is used to learn the behavioral data of individual robots. In the second stage, the network parameters are optimized by evolutionary algorithms. Experimental results show that the final strategy RNSE is very close to the target strategy PFSMS in terms of efficiency and stability, showing a balance between exploration and exploitation, and a good trade-off between parallelism and collaboration, which validates the effectiveness of the framework. The method section in this article is useful for the design of the network structure (layers, activation functions, initialization techniques, regularization techniques, etc.) and the setting of evolutionary algorithms (stability of the performance metric, learning rate settings) to facilitate the framework promotion and migration applications.

In this paper, the evolutionary algorithm is only used to optimize the last layer of the network, and the study of evolving the entire network can be conducted in the future. In addition, there are certain artificial traces in the construction and selection of the netwotk input features in this paper. Since deep networks have strong feature extraction and representation capabilities, in the future research, the original data can be directly used as the input of the network for training. In addition to the multi-target search, the framework can also be used for other tasks in swarm robotics, such as aggregation, dispersion, object transport, and so on.

Acknowledgment

This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119 and 61673025 and also Supported by Beijing Natural Science Foundation (4162029), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

References

[1] S. Camazine, Self-organization in biological systems, Princeton University Press, 2003.

[2] E. Şahin, Swarm robotics: From sources of inspiration to domains of application, in: International workshop on swarm robotics, Springer, 2004, pp. 10–20.

[3] E. Şahin, S. Girgin, L. Bayindir, A. E. Turgut, Swarm robotics, in: Swarm intelligence, Springer, 2008, pp. 87–100.

[4] M. S. Couceiro, D. Portugal, R. P. Rocha, A collective robotic architecture in search and rescue scenarios, in: Proceedings of the 28th Annual ACM Symposium on Applied Computing, ACM, 2013, pp. 64–69.

[5] J. Clark, R. Fierro, Mobile robotic sensors for perimeter detection and tracking, ISA transactions 46 (1) (2007) 3–13.

[6] J. Alonso-Mora, S. H. Lohaus, P. Leemann, R. Siegwart, P. Beardsley, Gesture based human-multi-robot swarm interaction and its application to an interactive display, in: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE, 2015, pp. 5948–5953.

[7] G. A. Landis, Robots and humans: synergy in planetary exploration, Acta astronautica 55 (12) (2004) 985–990.

[8] W. Wu, A. Song, P. Varnell, F. Zhang, Cooperatively mapping of the underwater acoustic channel by robot swarms, in: Proceedings of the International Conference on Underwater Networks & Systems, ACM, 2014, p. 20.

[9] B. Yang, Y. Ding, Y. Jin, K. Hao, Self-organized swarm robot for target search and trapping inspired by bacterial chemotaxis, Robotics and Autonomous Systems 72 (2015) 83–92.

[10] A. Jaimes, S. Kota, J. Gomez, An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles uav (s), in: System of Systems Engineering, 2008. SoSE'08. IEEE International Conference on, IEEE, 2008, pp. 1–6.

[11] J. Li, Y. Tan, The multi-target search problem with environmental restrictions in swarm robotics, in: Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on, IEEE, 2014, pp. 2685–2690.

[12] L. Bayindir, E. Şahin, A review of studies in swarm robotics, Turkish Journal of Electrical Engineering & Computer Sciences 15 (2) (2007) 115–147.

[13] M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm robotics: a review from the swarm engineering perspective, Swarm Intelligence 7 (1) (2013) 1–41.

[14] O. Soysal, et al., Probabilistic aggregation strategies in swarm robotic systems, in: Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, IEEE, 2005, pp. 325–332.

[15] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, The international journal of robotics research 5 (1) (1986) 90–98.

[16] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, Vol. 1, MIT press Cambridge, 1998.

[17] S. Nolfi, D. Floreano, Evolutionary robotics (2000).

[18] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, nature 529 (7587) (2016) 484–489.

[19] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.

[20] G. Klambauer, T. Unterthiner, A. Mayr, S. Hochreiter, Self-normalizing neural networks, in: Advances in Neural Information Processing Systems, 2017, pp. 972–981.

[21] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: Advances in neural information processing systems, 2007, pp. 153–160.

[22] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[23] J. Li, S. Zheng, Y. Tan, The effect of information utilization: introducing a novel guiding spark in the fireworks algorithm, IEEE Transactions on Evolutionary Computation 21 (1) (2017) 153–166.

[24] M. S. Couceiro, R. P. Rocha, N. M. Ferreira, A novel multi-robot exploration approach based on particle swarm optimization algorithms, in: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, IEEE, 2011, pp. 327–332.

[25] H. N. Ataei, K. Ziarati, M. Eghtesad, A bso-based algorithm for multi-robot and multi-target search, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2013, pp. 312–321.

[26] K. Krishnanand, D. Ghose, A glowworm swarm optimization based multi-robot system for signal source localization, in: Design and control of intelligent robotic systems, Springer, 2009, pp. 49–68.

[27] Z. Zheng, Y. Tan, Group explosion strategy for searching multiple targets using swarm robotic, in: 2013 IEEE Congress on Evolutionary Computation, IEEE, 2013, pp. 821–828.

[28] G. M. Viswanathan, V. Afanasyev, S. Buldyrev, E. Murphy, P. Prince, H. E. Stanley, Lévy flight search patterns of wandering albatrosses, Nature 381 (6581) (1996) 413.

[29] O. Bénichou, C. Loverdo, M. Moreau, R. Voituriez, Two-dimensional intermittent search processes: An alternative to lévy flight strategies, Physical Review E 74 (2) (2006) 020102.

[30] V. Gazi, K. M. Passino, Stability analysis of social foraging swarms, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34 (1) (2004) 539–557.

[31] L. Xie, G. Yang, J. Zeng, Z. Cui, Swarm robots search based on artificial physics optimisation algorithm, International Journal of Computing Science and Mathematics 4 (1) (2013) 62–71.

[32] J. Li, Y. Tan, Triangle formation based multiple targets search using a swarm of robots, in: International Conference in Swarm Intelligence, Springer, 2016, pp. 544–552.

[33] M. S. Couceiro, P. A. Vargas, R. P. Rocha, N. M. F. Ferreira, Benchmark of swarm robotics distributed techniques in a search task, Robotics and Autonomous Systems 62 (2) (2014) 200–213.

[34] M. Dadgar, M. S. Couceiro, A. Hamzeh, Rdpso diversity enhancement based on repulsion between similar ions for robotic target searching, in: Artificial Intelligence and Signal Processing Conference, 2018, pp. 275–280.

[35] M. Dadgar, S. Jafari, A. Hamzeh, A pso-based multi-robot cooperation method for target searching in unknown environments, Neurocomputing 177 (2016) 62–74.

[36] X. Yang, J. Yuan, J. Yuan, H. Mao, A modified particle swarm optimizer with dynamic adaptation, Applied Mathematics and Computation 189 (2) (2007) 1205–1213.

[37] Y. Tan, Y. Zhu, Fireworks algorithm for optimization, in: International Conference in Swarm Intelligence, Springer, 2010, pp. 355–364.

[38] Z. Zheng, J. Li, J. Li, Y. Tan, Improved group explosion strategy for searching multiple targets using swarm robotics, in: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2014, pp. 246–251.

[39] J. P. Hecker, M. E. Moses, Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms, Swarm Intelligence 9 (1) (2015) 43–70.

[40] J. Gomes, P. Urbano, A. L. Christensen, Evolution of swarm robotics systems with novelty search, Swarm Intelligence 7 (2-3) (2013) 115–144.

[41] M. Duarte, V. Costa, J. Gomes, T. Rodrigues, F. Silva, S. M. Oliveira, A. L. Christensen, Evolution of collective behaviors for a real swarm of aquatic surface robots, PloS one 11 (3) (2016) e0151834.

[42] J. Foerster, I. A. Assael, N. de Freitas, S. Whiteson, Learning to communicate with deep multi-agent reinforcement learning, in: Advances in Neural Information Processing Systems, 2016, pp. 2137–2145.

[43] J. Foerster, N. Nardelli, G. Farquhar, P. Torr, P. Kohli, S. Whiteson, et al., Stabilising experience replay for deep multi-agent reinforcement learning, arXiv preprint arXiv:1702.08887.

[44] Q. Li, X. Du, Y. Huang, Q. Sykora, A. P. Schoellig, Learning of coordination policies for robotic swarms, arXiv preprint arXiv:1709.06620.

[45] Y. Bengio, et al., Learning deep architectures for ai, Foundations and trends® in Machine Learning 2 (1) (2009) 1–127.

[46] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning?, Journal of Machine Learning Research 11 (Feb) (2010) 625–660.

[47] Z. Zheng, J. Li, J. Li, Y. Tan, Avoiding decoys in multiple targets searching problems using swarm robotics, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 784–791.

[48] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, 2015, pp. 448–456.

[49] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[50] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, H. China, Benchmark functions for the cec 2013 special session and competition on large-scale global optimization, gene 7 (33) (2013) 8.

[51] J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.

[52] N. Hansen, The cma evolution strategy: A tutorial, arXiv preprint arXiv:1604.00772.

[53] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1-3) (2006) 489–501.

[54] Q. Chen, B. Liu, Q. Zhang, J. Liang, P. Suganthan, B. Qu, Problem definition and evaluation criteria for cec 2015 special session and competition on bound constrained single-objective computationally expensive numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, China and Nanyang Technological University, Singapore, Technical report.

Jie Li, received a Bachelor of Intelligence Science and Technology from University of Science and Technology Beijing, China, in 2013. He is currently majoring in Computer Science and working towards the PhD degree at Key Laboratory of Machine Perception (Ministry of Education) and Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include Swarm Robotics, Evolutionary Computation, Machine Learning and Artificial Neural Networks.

Ying Tan, is a full professor and PhD advisor at the School of Electronics Engineering and Computer Science of Peking University, and director of Computational Intelligence Laboratory at Peking University. He received his B.Eng, M.S., and PhD from Southeast University, in 1985, 1988, and 1997, respectively. He is the inventor of Fireworks Algorithm (FWA). He serves as the Editor-in-Chief of International Journal of Computational Intelligence and Pattern Recognition (IJCIPR), the Associate Editor of IEEE Transaction on Cybernetics (CYB), IEEE Transaction on Neural Networks and Learning Systems (NNLS), International Journal of Swarm Intelligence Research (IJSIR), International Journal of Artificial Intelligence (IJAI), etc. He also served as an Editor of Springers Lecture Notes on Computer Science (LNCS) for 20+ volumes, and Guest Editors of several referred Journals, including IEEE/ACM Transactions on Computational Biology and Bioinformatics (IEEE/ACM TCBB), Information Science, Softcomputing, Neurocomputing, IJSIR, IJAI, B&B, CJ, etc. He is an IEEE senior member and a member of Emergent Technologies Technical Committee (ETTC) of IEEE Computational Intelligence Society since 2010. He is the founder general chair of the ICSI International Conference series. He won the 2nd-Class Natural Science Award of China in 2009.