논문 17-42-09-18 The Journal of Korean Institute of Communications and Information Sciences '17-09 Vol.42 No.09
https://doi.org/10.7840/kics.2017.42.9.1843

2-D 거리 기반 무선 위치 측위를 위한 Dragonfly Algorithm 분석

필립 또비안또 다엘리•, 신 수 용°

Analysis of the Dragonfly Algorithm for 2-D Range-Based Wireless Localization

Philip Tobianto Daely•, Soo Young Shin°

요 약

본 논문에서 2-D 시나리오에서의 거리 기반 무선 위치 측위를 위해 Dragonfly Algorithm (DA)의 무리 행동을 분석하였다. DA는 잠자리의 무리 행동을 기반으로한 메타휴리스틱 알고리즘이다. 무선 위치 측위는 최적화 문제로서 다뤄지는데 이를 해결하기 위해 DA를 적용하였다. 또한 본 논문에서는 이전에 진행되었던 위치 측위를 위한 생체모방 메타휴리스틱 알고리즘 연구 또한 다루었다. 무리 행동에 대한 분석은 수치적으로 제시되고 논의하였다. 무선 위치 측위에 DA를 적용하였을 때 성능 향상이 있음을 확인하였다.

Key Words : localization, Dragonfly Algorithm, analysis, swarming behavior, optimization

ABSTRACT

In this paper, we provide analysis of the swarming behaviors in Dragonfly Algorithm (DA) on account of being used for range-based wireless localization in 2-D scenario. DA is a metaheuristic algorithm based on the swarming behaviors of dragonfly. Wireless localization itself can be treated as optimization problem, which make DA applicable for it. This paper also covers several previous works related to application of bio-inspired metaheuristic algorithm in localization. The analysis for each swarming behavior is presented numerically and discussed. The results show that there are ways of improvement for DA when applied for wireless localization.

Ⅰ. Introduction

Wireless localization is a method performed with intention to locate one or more wireless nodes without known location information and positioning device. Generally, a wireless node is equipped with a positioning device such as Global Positioning System (GPS) or Global Navigation Satellite System (GLONASS) to get the current location over time. The information of location is necessary, especially for a moving wireless node, to maintain the integrity of data from the wireless node.

Over several years, many studies have been done in order to localize a wireless node without assist from positioning device. Basically, those studies result in techniques that can be divided into two categories: range-based and range-free localization. The difference between both of them is that the

• First Author : Kumoh National Institute of Technology, Department of IT Convergence Engineering, 20156130@kumoh.ac.kr, 학생회원
° Corresponding Author : Kumoh National Institute of Technology, wdragon@kumoh.ac.kr, 종신회원
논문번호 : KICS2017-04-125, Received April 28, 2017; Revised July 11, 2017; Accepted July 14, 2017

range-based localization depends on measurement of distances or ranges between the unknown node (node with unknown location) and several beacon nodes (nodes with known location) while the range-free localization utilizes the locations of the beacon nodes or counts the number of hops with neighboring beacon nodes. In this paper, we only cover the range-based localization in wireless network.

Range-based localization uses multiple distance measurement to locate the unknown node. The distances can be derived from techniques that exploit the wireless network properties such as angle of arrival (AOA)[1], time of arrival (TOA)[2], time difference of arrival (TDOA)[3], or received signal strength (RSS)[4]. AOA uses the angle of arriving signal, which can be converted to distance by using trigonometry laws. TOA and TDOA use the propagation time of the signal to calculate the distance. While time synchronization is necessary in TOA, it is not the case with TDOA. RSS is another technique that convert the signal strength to distance, as signal strength is inversely proportional to square of distance.

In ideal case, the distance measurement is not affected by uncertainty, making the measured distances are exactly same with actual distances, thus can be used directly to pinpoint the unknown node's location by using multilateration principles. But in real environment, there is noise that can tamper the distance measurement and resulting in no exact solution. Therefore, we try to look at this problem as an optimization problem, where there is no exact solution, but instead several solutions called local optima, in a search area with their own fitness value. The optimization algorithm needs to find the global optimum, which is a local optimum with the best fitness value, compared to other local optima.

In this paper, we propose Dragonfly Algorithm (DA) as an optimization algorithm in order to find or estimate the location of unknown node, as shown in [5]. The work in [6] explains about the relation between the localization precision with the change in iteration number and swarm size, but it does not cover the effect of each behavior factor in DA toward the localization performance. This paper tries to further analyze the effect of each factor and gives our insight regarding the results.

The rest of this paper is organized as follows. Section II reviews past works related to this paper. Section III explains briefly about the range-based wireless localization from optimization problem perspective. Section IV covers the DA and how it can optimize wireless localization problem. Section V shows our simulation results and analysis of DA for wireless localization with varied parameters. Section VI provides our conclusions and possible future works from this paper.

Ⅱ. Related Works

Wireless localization can be assumed as an optimization problem. Optimization algorithms can be categorized as deterministic algorithm or stochastic algorithm[7]. Deterministic algorithm is rigorous, has known inputs, and usually requires the derivative of the problem to be known. Stochastic algorithm on the other hand contains randomness in the process and can provide immediate result with relatively short computational time. Wireless localization is usually embedded to a wireless device with limited computation resource, which is why a stochastic algorithm is preferable.

Metaheuristic algorithm is one kind of stochastic algorithm, a method that can provide quick sufficient solution for most optimization problems. One of the oldest metaheuristic algorithm is Particle Swarm Optimization (PSO), which is inspired by the movement of bird flock and fish school. Several works such as [8] and [9] use PSO for wireless localization. They show that PSO can outperform classical optimization algorithm. Another variant of PSO, which is Binary PSO (BPSO) also has been used to improve the speed of localization, as shown in [10]. Work in [11] shows the localization using Firefly Algorithm (FA) and its modified version. FA is inspired by the way fireflies attracted to each other because of their capability to produce light from their body as a signal. The proposed algorithm shows good performance compared to the other algorithms. Work in [12] and [13] give the results of localization by using Cuckoo Search (CS). CS is based on the behavior of Cuckoo bird leaving its egg to another bird's nest. The comparison with other algorithms show significantly better result. Meanwhile, work in [14] and [15] show another metaheuristic algorithm named Chicken Swarm Algorithm (CSO) for localization. CSO is based on behavior of chicken swarm and hierarchy of each chicken to provide best solution.

All of the previous works reveal that the utilization of metaheuristic algorithm is promising and gain much interest recently. This paper tries to give another perspective from DA, as one of metaheuristic algorithm, in wireless localization. The behavior factors in DA are varied and simulated so that we can see the impact in the performance of localization.

Ⅲ. Range-Based Wireless Localization

Range-based wireless localization is a localization technique to locate unknown node in a wireless network by using knowledge of distances between unknown node and beacon nodes. In this paper, we assume a wireless network consists of unknown nodes and beacon nodes. Every unknown node will measure its distance to each neighboring beacon node.

The measured distance in 2-D space between unknown node u and beacon node b consists of actual distance $d_{u,b}$ and measurement noise n_{dis} as modeled in (1).

$$\check{d}_{u,b} = d_{u,b} + n_{dis} \tag{1}$$

To get more accurate distance, the measurement is performed N times, and then, the average of measurements will be used for localization, as shown in (2).

$$\overline{d}_{u,b} = \frac{1}{N}\sum_{n=1}^{N} \check{d}_{u,b,n} \tag{2}$$

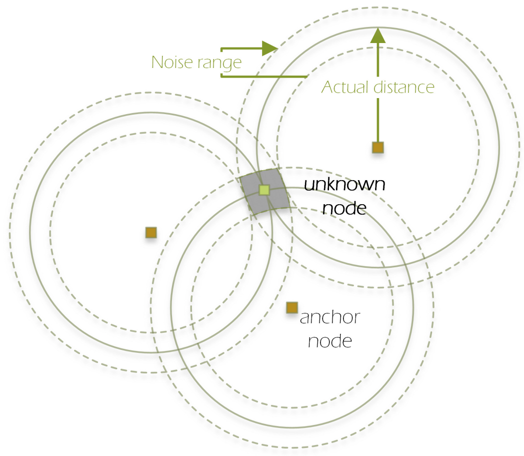

Fig. 1. Illustration of range-based localization, where distance measurement contains noise and results in incapability of exact location estimation (shown by area with grey color around unknown node).

When we look localization problem as optimization problem, we can estimate the location of unknown node u, with B beacon nodes, by using objective function notated in (3), with $(\hat{x}_u, \hat{y}_u)$ denotes the estimated location of unknown node u and (x_b, y_b) denotes the actual location of beacon node b.

$$\arg\min_{\hat{x}_u, \hat{y}_u} \sqrt{\frac{1}{B}\sum_{b=1}^{B}\left[\sqrt{(\hat{x}_u - x_b)^2 + (\hat{y}_u - y_b)^2} - \overline{d}_{u,b}\right]^2} \tag{3}$$

Equation (3) basically tries to minimize the root mean square between estimated distance and measured distance. When the minimum value of objective function is reached, the estimated location will be the final result of localization. The illustration of range-based localization problem is presented in Fig. 1.

Ⅳ. Dragonfly Algorithm

DA is one of swarm intelligence, created based on the swarming behaviors of dragonflies. It employs the exploration and exploitation ability of dragonfly swarm[16]. In exploration phase, the dragonflies will roam around the search area, forming several small swarms, in order to find the food source. When the food source is located, all dragonflies will form one big swarm and start the exploitation of food source.

In exploration phase, DA uses five swarming behaviors of dragonfly to update each search agent over the search area. These behaviors portray how a dragonfly interacts with other nearby dragonflies. These five behaviors are separation, alignment, cohesion, attraction to food, and distraction outward enemy[16]. The illustration of dragonfly's swarming behaviors is shown in Fig. 2. These behaviors are described as follows:

Separation, behavior of dragonfly that try to maintain its distance with neighboring dragonflies. The separation factor for search agent q is formulated in (4), with X_q denotes the position of search agent q, X_k denotes the position of k-th neighboring search agent, and K denotes the number of neighboring search agents.

$$S_q = -\sum_{k=1}^{K}(X_k - X_q) \tag{4}$$

Alignment, behavior of dragonfly that try to match its velocity with the average velocity of neighboring dragonflies. The alignment factor for search agent q is formulated in (5), with V_k denotes the velocity of k-th neighboring search agent.

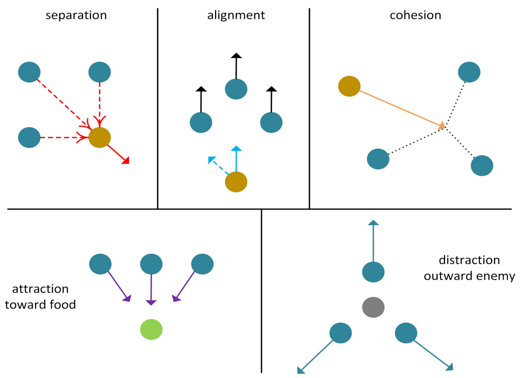

Fig. 2. Five swarming behaviors of dragonfly implemented in DA.

$$A_q = \frac{1}{K}\sum_{k=1}^{K} V_k \tag{5}$$

Cohesion, behavior of dragonfly that try to shift inward the center of mass of neighboring dragonflies. The cohesion factor for search agent q is formulated in (6).

$$C_q = \left(\frac{1}{K}\sum_{k=1}^{K} X_k\right) - X_q \tag{6}$$

Attraction toward food source, behavior of dragonfly that always approach the food source. In DA, the best search agent is considered as food source. The attraction factor for search agent q is formulated in (7), with X_{best} denotes the position of the best search agent.

$$F_q = X_{best} - X_q \tag{7}$$

Distraction outward enemy, behavior of dragonfly that always move away from enemy/predator. In DA, the worst search agent is considered as enemy. The distraction factor for search agent q is formulated in (8), with X_{worst} denotes the position of the worst search agent.

$$E_q = X_{worst} - X_q \tag{8}$$

Each search agent's position update depends on the existence of other neighboring search agents. If search agent q has neighboring search agent(s), it will update the velocity and position as (9) and (10) respectively,

$$V_{q,t+1} = w_S S_q + w_A A_q + w_C C_q + w_F F_q + w_E E_q + w_V V_q \tag{9}$$

$$X_{q,t+1} = X_{q,t} + V_{q,t+1} \tag{10}$$

with w_S, w_A, w_C, w_F, w_E, and w_V are the weight for each behavior factor. If search agent q does not have any neighboring search agent, it will update its position by using random walk based on Lévy flight, as in (11).

$$X_{q,t+1} = X_{q,t} + rand\acute{Levy}() \tag{11}$$

Ⅴ. Simulation Results and Discussion

To see the performance of DA for wireless localization, we simulated it in MATLAB environment. We created a square-shaped area of simulation with length of 100 m. We placed four beacon nodes at each corner of area, and an unknown node in the middle of area. We assumed unknown node is in the vicinity of all beacon nodes, so that it can communicate well with all of them.

The original factor weights[17] are originally set as random number of uniform distribution with declining distribution range toward the iteration. We then tried to set them as constant one at a time, with varying value. For each value, we ran the localization for 3000 times and compares the results.

For localization with constant w_S, the localization errors is shown in Fig. 3. It shows that $w_S = 2$ has lowest error compared to other constant values. But, as we can see, the original value still has lower error than all other values. The dynamic change of w_S still gives best result for localization.

For localization with constant w_A, the localization errors is shown in Fig. 4. It shows that $w_A = 2$ has the best performance, with the lowest error compared to other values. It also has lower error than original value. It gives an idea that making the value constant and large can give lower error. The increase of w_A should also be limited so that search agents do not go to similar direction and can move to different areas.

For localization with constant w_C, the localization errors is shown in Fig. 5. It shows that $w_C = 1.2$ has the best performance, with the lowest error compared to other values. Decrease and increase from that value show increase of error, which suggest that 1.2 is appropriate or almost appropriate for localization.

For localization with constant w_F, the localization errors is shown in Fig. 6. It shows that $w_F = 2$ has the best performance, with the lowest error compared to other values, except to the original value. But given the error difference between $w_F = 2$ and original value not really much, there is possibility of improvement if we keep increase the value of w_F.

For localization with constant w_E, the localization errors is shown in Fig. 7. It shows that $w_E = 0.2$ has the best performance, with the lowest error

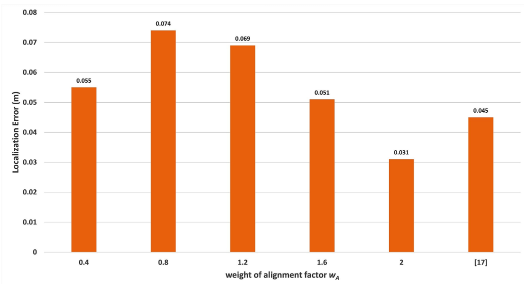

Fig. 4. Comparison of localization error with different value of w_A

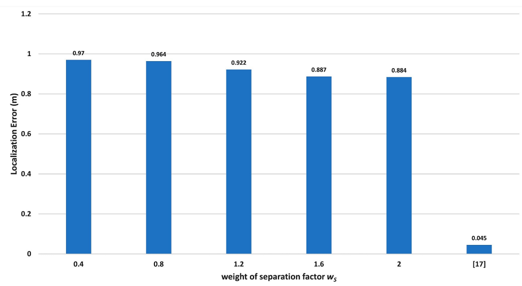

Fig. 3. Comparison of localization error with different value of w_S

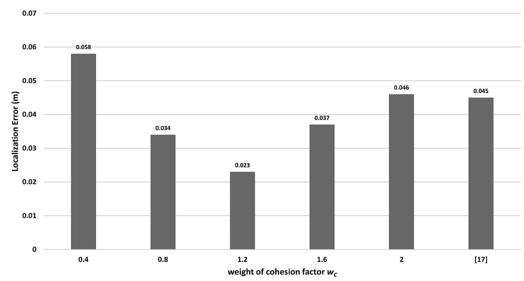

Fig. 5. Comparison of localization error with different value of w_C

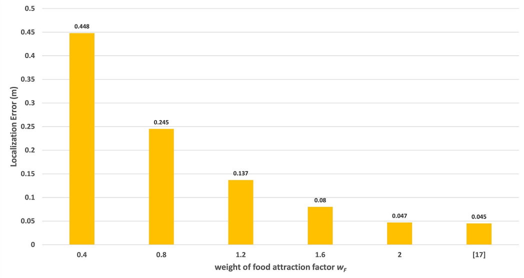

Fig. 6. Comparison of localization error with different value of w_F

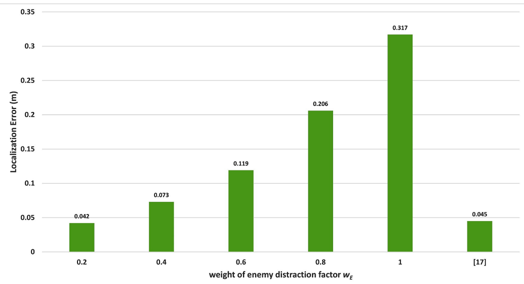

Fig. 7. Comparison of localization error with different value of w_E

compared to other values. Low enemy distraction factor could lead to mitigation of error, but should be keep above 0 so search agents can get out of local minima faster.

Ⅵ. Conclusion and Future Works

We have proposed and analyzed the use of DA for range-based wireless localization. The changes in value of behavior factors definitely have significant impact toward the localization performance. The results show that, with proper adjustment, the error can be lessened and make DA can perform effectively for wireless localization.

Future work could include the hybridization with another algorithm, possibly metaheuristic algorithm, to further improve the accuracy, robustness and computational capability.

References

[1] G. Zhu and J. Hu, "A distributed continuous time algorithm for network localization using angle-of," *Automatica*, vol. 50, no. 1, pp. 56-63, Jan. 2014.

[2] S. Monicaa and G. Ferrarib, "Maximum likelihood localization: When does it fail?," *ICT Express*, vol. 2, no. 1, pp. 10-13, Mar. 2016.

[3] X. Liu, Y. Huang, and J. Wang, "A novel TDOA-Based localization algorithm using sequential quadratic programming," *Int. J. Hybrid Inf. Technol.*, vol. 9, no. 11, pp. 91-98, Nov. 2016.

[4] J. Huang, P. Liu, W. Lin, and G. Gui, "RSS-Based method for sensor localization with unknown transmit power and uncertainty in path loss exponent," *Sensors*, vol. 16, no. 9, Sept. 2016.

[5] P. T. Daely and S. Y. Shin, "Range based wireless node localization using dragonfly algorithm," in *8th Int. Conf. on Ubiquitous and Future Networks*, Vienna, Austria, Jul. 2016.

[6] P. T. Daely and S. Y. Shin, "Analysis of dragonfly algorithm for wireless node localization," in *Proc. KICS Winter Conf.*, Jeongseon, South Korea, Jan. 2017.

[7] X.-S. Yang, *Nature-Inspired Optimization Algorithms*, Elsevier, 2016.

[8] H. Guo, K.-S. Low, and H.-A. Nguyen, "Optimizing the localization of a wireless sensor network in real time based on a low-cost microcontroller," *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 741-749, Mar. 2011.

[9] C.-S. Chen, "A non-line-of-sight error mitigation method for location estimation," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 1, Jan. 2017.

[10] I. Fatihah and S. Y. Shin, "Distributed sensor node localization using a binary particle swarm optimization algorithm," *J. IEIE*, vol. 51, no. 7, pp. 9-17, Jul. 2014.

[11] S. Cao, J. Wang, and X. Gu, "A wireless sensor network location algorithm based on firefly algorithm," in *Asia Simulation Conf. 2012*, Shanghai, China, Oct. 2012.

[12] S. Goyal and M. S. Patterh, "Wireless sensor network localization based on cuckoo search algorithm," *Wireless Personal Commun.*, vol. 79, no. 1, pp. 223-234, Nov. 2014.

[13] S. Sivakumar and R. Venkatesan, "Error minimization in localization of wireless sensor networks using modified cuckoo search with mobile anchor positioning (MCS-Map) algorithm," *Int. J. Comput. Appl.*, vol. 95, no. 6, Jun. 2014.

[14] M. A. Shayokh and S. Y. Shin, "Distributed wireless sensor node localization using chicken swarm optimization," in *Proc. KICS Summer Conf.*, Seogwipo, South Korea, 2016.

[15] P. Chen and Y. Mao, "Wireless sensor network node localization algorithm based on chicken swarm optimization and multi-power mobile anchor," in *3rd Int. Conf. Materials Eng.*, Taiyuan, China, 2016.

[16] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053-1073, May 2016.

[17] S. Mirjalili, *Dragonfly Algorithm Toolbox, MathWorks* (2015), Retrieved Apr., 25, 2017, from https://www.mathworks.com/matlabcentral/fileexchange/51031-dragonfly-algorithm-toolbox.

필립 또비안또 다엘리 (Philip Tobianto Daely)

2015년 2월 : 인도네시아 텔콤 대학교 전자공학과 졸업

2017년 8월 : 국립금오공과대학교 IT융복합공학과 석사

<관심분야> 전자공학, 통신공학, 광통신 공학

신 수 용 (Soo Young Shin)

1999년 2월 : 서울대학교 전기공학부 졸업

2001년 2월 : 서울대학교 전기공학부 석사

2006년 2월 : 서울대학교 전기컴퓨터공학부 박사

2010년 2월 : 국립금오공과대학교 전자공학부 교수

<관심분야> 5G and FRA, wireless communication/network, Internet of Things, signal processing, etc.