RESEARCH

A Monte Carlo localization method based on differential evolution optimization applied into economic forecasting in mobile wireless sensor networks

Miao Qin[1] and Rongbo Zhu[2*]

Abstract

The localization of sensor node is an essential problem for many economic forecasting applications in wireless sensor networks. Considering that the mobile sensors change their locations frequently over time, Monte Carlo localization algorithm utilizes the moving characteristics of nodes and employs the probability distribution function (PDF) in the previous time slot to estimate the current location by using a weighted particle filter. However, it also has the problem of insufficient number of valid samples, which further affects the node's localization accuracy. In this paper, differential evolution method is introduced into the Monte Carlo localization algorithm. The sample weight is taken as the objective function, and differential evolution algorithm is implemented in sample stage. Finally, the node position is estimated by making the sample close to the actual location of the node instead of being filtered out. The simulation results demonstrate that the proposed algorithm provides a better position estimation with less localization error.

Keywords: Economic forecasting, Wireless sensor networks, Valid sample, Localization, Differential evolution

1 Introduction

In the era of big data, economic forecasting is crucial. The nodes' localization in wireless sensor networks (WSNs) refers to the process of obtaining their own or monitoring the geographic position of the object in a certain way [1]. It is important to obtain the sensor node's location, and the monitoring data will be meaningless without location information. For instance, in precision agriculture, the sensor nodes can gather data of light intensity, humidity, and temperature, which must be accompanied by the coordinates of the collectors [2]. Without such positional information, the observer cannot match the data with the region and make an appropriate decision. Meanwhile, the exact location information of sensor nodes are of great help in improving the efficiency of network routing [3]. Since the wireless sensor network consists of a large number of sensors and the topology often change especially for the

environment with mobile nodes, each sensor nodes need to be equipped with a positioning system, such as GPS. Due to the high cost of GPS, it is not suitable for low-power and low-cost requirements of sensor nodes [4]. In addition, for some special application scenarios (such as shopping malls), the positioning performance of GPS will be affected [5]. Generally, there are two types of nodes in wireless sensor networks, which are called as anchor node and blind node. Anchor nodes, which are usually configured manually or equipped with a GPS receiver to obtain their location information, can obtain position coordinates by themselves. However, the proportion of anchor nodes in all sensor nodes is relatively small. Comparatively, blind node can only acquire its position information by using the localization algorithm.

The rest of the paper is structured as follows. The motivation for this work is discussed in Section 2. In Section 3, we derive Monte Carlo localization methods based on differential evolution optimization (MCL-DE) for valid samples in mobile wireless sensor networks. In Section 4, a comparative performance evaluation is

* Correspondence: rbzhu@mail.scuec.edu.cn
[2]College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China
Full list of author information is available at the end of the article

carried out. Finally, concluding remarks and future work are given in Section 5.

2 Related work

So far, the research of node's localization algorithm in wireless sensor networks has been widely carried out. The main purpose of sensor localization is to determine the location of sensors in WSNs via noisy measurements, and most of the methods for localization can be classified into geometrical techniques, multidimensional scaling, stochastic proximity embedding, convex and nonconvex optimization, and hybrid. In range-based measurement localization, the major task is to find the accurate position in non-line-of-sight (NLOS) paths. These range-based measurements may include time-of-arrival (TOA) [6], time-difference-of-arrival (TDOA) [7], angle-of-arrival (AOA) [8], and received signal strength (RSS). After evaluating the distance between the nodes, the position of the blind node can be obtained based on three edge-measuring or maximum likelihood methods [9].

Range-based localization requires additional hardware and power consumption, so nodes can achieve accurate positioning resolution. However, the demand to reduce hardware dependency and energy cost has been the focus of academia and industry, and some researchers also proposed a range-free localization algorithm [10]. Usually, range-free location algorithms demonstrate poor performance in the aspect of positioning accuracy than the range-based localization algorithm, but it does not need additional hardware support and can meet many requirements in the scenarios with rough localization effect. In [10], an indoor localization strategy for mini-UAV in the presence of obstacles is proposed, in which the signal propagation state is identified according to the prior probability and statistics of TDOA and RSS measurements. In [11], a NLOS identification and weaken algorithm with machine learning is proposed to identify and weaken the NLOS error by means of support vector machine (SVM), which can employ a large number of data samples to train the SVM classifier. A voting matrix is constructed to weaken the error of non-line-of-sight and obtain the candidate position in accordance with the error characteristics of LOS measurements and NLOS measurements. Then, the residual weighted method is used to obtain the final positioning results. Base on the range distance in each sampling period, Cui et al. [12] use multidimensional scaling localization algorithm to evaluate the location of the target and fit the result of the estimation by polynomial. The estimation results of current position can be corrected effectively, and the method is proven to achieve high positioning accuracy in indoor environment.

In recent years, interacting multiple model (IMM) combined with filtering technology has become a hot research topic. Chen et al. [13] combine IMM and extended Kalman filtering to achieve accurate positioning in NLOS environment. Zhang et al. [14] propose a Kalman filter model based on interacted multiple objectives to filter the measured distance under the LOS/NLOS mixed environment, in which the IMM algorithm is applied to filter the distance, and then the extended Kalman filtering algorithm is used to realize the positioning. Under the IMM framework, Ru et al. [15] employ hidden Markov random field to solve the nonlinear Bayesian estimation problem and improve the positioning accuracy. Nevertheless, the above methods are put forward in the premise of accurate NLOS error parameters. But in the actual environment, the parameters of NLOS error are usually unknown. In [16], an Advanced DV-Hop localization algorithm is proposed to reduce the localization error without requiring additional hardware and computational costs. The hop-size of the anchor node is obtained base on the distance measurement of unknown nodes, and the weighted least square algorithm is introduced to decrease the inherent error in the estimated distance between the anchor and an unknown node. In [17], a mixed localization algorithm for wireless sensor networks based on APIT is proposed to deal with the problem of low localization accuracy with dense distribution of beacon nodes and low coverage ratio in the sparse case.

As a fundamental requirement in Internet of Things (IoT) and other wireless sensor applications, localization awareness is an important part of network design. Due to the complexity of related algorithms, researchers in this area have acquired significant progress but still exists quite a few deficiencies. Most of the solutions are static localization algorithms for node's positioning, but the performance of those algorithms in mobile wireless sensor network is unsatisfactory. In the real scene, such as target tracking, the sensor nodes are usually mobile and the node localization mechanism should be suitable for mobile wireless sensor networks. Considering that the mobile sensors change their locations frequently over time, Monte Carlo localization algorithm utilizes the moving characteristics of nodes and employs the probability distribution function (PDF) in the previous time slot to estimate the current location by using a weighted particle filter. However, it also has the problem of insufficient number of valid samples, which further affects the node's localization accuracy. Therefore, it is necessary to increase the number of valid samples to improve the localization accuracy under low anchor node density.

3 System and network model

3.1 Monte Carlo localization method

Monte Carlo localization method was originally applied to the field of robot localization, and the distinction between the robot localization and the sensor node's positioning in mobile wireless sensor network is very remarkable [18]. In the process of robot walking, the robot's CPU is equipped with a map, and the path guidance will be abided to the prescribed route in the map. However, all sensor nodes will move in a random mode in the designated area. The location method based on Monte Carlo is actually a continuous iterative Bayesian filter, and the basic idea is to make use of some weighted samples to represent the posterior probability density distribution of the estimated state, so as to obtain the solution of the node position. This method can be applied to non-Gauss, nonlinear and multidimensional system, which is beneficial for the characteristics including flexibility, easy to implement, and suitable for parallel processing. Those merits make it very suitable for node localization in wireless sensor networks.

Since that the neighbor nodes within the range of transmission radius can communicate with each other, the known information from anchor nodes can be used to assist blind node's localization [19]. The Monte Carlo localization method is based on the Bayes filtering theory, and the main idea is by utilizing the new observation from the adjacent anchor nodes within the range, the sample and filter steps will be repeated until enough valid samples can be obtained. Then, the blind node can estimate its current location as it completes the movement [20]. Therefore, the resolution of the blind node's localization can be transferred into the posterior probability density function. Let t be a discrete time series, x_t is the state of hidden Markov processes with initial distribution $P(x_0)$, $x_t \in R_x^n$ where n_x is the dimension of state vector. Transfering equation $P(x_t|x_{t-1})$ demonstrates the dynamic features of the state space model. Meanwhile, the observation sequences $\{o_1, o_2, o_3, ..., o_t\}$ are independent of each other at a given node's position $\{x_1, x_2, x_3 ..., x_t\}$, where $O_t \in R_o^n$ and n_0 is the dimension of the observation vector. $P(o_t|x_t)$ is the observation equations, and it denotes the probability of observed values under the condition of a given position x_t.

Suppose that the location of mobile nodes satisfies Markov assumptions in mobile wireless sensor networks, and the observation and node's position are independent. This indicates that the observations only depend on the current position, and the current position x_t lies on the position x_{t-1} at the previous time interval. Thus, the resolution of the location of blind nodes can be converted into a posteriori probability density function $p(x_t|o_{0:t})$.

The posterior probability density function at time t can be approximated by some weighted samples (x_t^i, w_t^i), and we have:

$$p(x_t|o_{0:t}) = \sum_{i=1}^{N} w_t^{i*} \delta(x_t - x_t^i) \tag{1}$$

where δ is the Dirac-delta function, and N represents the number of samples for the node's location. x_t^i is a possible sample of node at time t, and w_t^i is a nonnegative weight.

The sample weight will be updated as the following formula:

$$w_t^i = \frac{p(o_t|x_t^i)p(o_t|x_{t-1}^i)}{\phi(x_t^i, x_{0:i-1}^i, o_{0:t})} w_{t-1}^i \tag{2}$$

where ϕ is the adjustment function being relevant to x_t^i, $x_{0:i-1}^i, o_{0:t}$.

Since the sensor nodes move randomly in a certain area, and the size and direction of motion are unknown. The maximum speed of motion is limited to v_0; the current position of the node must be in the circle area with the center point of the position at the previous moment and a radius of maximum moving speed v_0. Then, the sample probability distribution at the present moment can be expressed as:

$$p(z_t|z_{t-1}) = \begin{cases} 0, & d(z_t, z_{t-1}) \geq v_0 \\ \dfrac{1}{\sqrt{\pi v_0^2}}, & d(z_t, z_{t-1}) < v_0 \end{cases} \tag{3}$$

where $d(z_t, z_{t-1})$ denotes the Euclidean distance between the sample at current and previous time.

3.2 Objective function optimization based on sample weight

The anchor nodes will broadcast the ID identification and location information periodically. Suppose that the broadcast message from the set of anchor nodes within one-hop $S(s)$ and the two-hop $T(s)$ can be received by the blind node at the current time, the samples that does not satisfy the condition can be rejected in the process of filtering prediction with reference to the observation requirement. The eligible samples must be within the communication radius of a neighbor anchor node; meanwhile, the distance between the sample and the two-hop anchor node must be less than two times of the communication radius. Thus, the constraints can be expressed as: $\forall s \in S(s), d(z,s) \leq r \cap \forall s \in T(s), r < d(z,s) \leq 2r$.

According to the N samples of the position and the value of (z_t^i, w_t^i), the current position of the blind node can be estimated as:

$$CurPOS_t = \sum_{i=1}^{N} z_t^i \cdot w_t^i \qquad (4)$$

The sample weights, i.e., the proportion of the samples in the final positioning result, are a measure of the merits of the standard sample. The observation results O_t of ordinary nodes are composed of a one-hop neighbor anchor node set S, two-hop anchor node set T and the set of normal nodes within the transmission range TR, and $O_t = S \cup T \cup TR$. Considering the size of the sample constraint box, the confidence function is introduced into the localization results to reflect the confidence degree of node i at time t, of which the value depends on the size of the constraints from sampling box.

$$w_t^i = p\left(O_t|z_t^i\right) = \left(\prod_{s \in S \cup T} p\left(s|z_t^i\right)\right) \cdot \left\{\sum_{s \in TR} (\gamma(s, t-1), d(z_t^i, s) < r\right\} \qquad (5)$$

where $\gamma(i, t) = \gamma/\gamma_i^t$ and γ_i^t is the diagonal length of the sampling box of node i at time t, and γ is the diagonal length of the entire region.

if $s \in S$,

$$p\left(s|z_t^i\right) = \begin{cases} 1, 0 < d\left(s, z_t^i\right) < r \\ 0, else \end{cases} \qquad (6)$$

if $s \in T$,

$$p\left(s|z_t^i\right) = \begin{cases} 1, r < d\left(s, z_t^i\right) < 2r \\ 0, else \end{cases} \qquad (7)$$

where the sample weight is regarded as the objective function of the optimization.

3.3 The process of sampling optimization

Owing to the similarity of the idea between the differential evolution algorithm and the Monte Carlo algorithm, the individual vector in differential evolution algorithm can be regarded as a particle sample. Moreover, if the population number is equal to the size of the particle set, then the population in the differential evolution algorithm is equivalent to the particle set in the Monte Carlo localization algorithm. Hence, it is convenient to involve differential evolution method into Monte Carlo localization algorithm. In the sampling phase, the sample weight is taken as the objective function, and the differential evolution algorithm is implemented. Then, the actual position of sample can be approached to the node to be positioned actively rather than to be filtered, and the final estimate of node's position can be obtained. The detailed steps are as follows:

1. Initialization: According to the initial variable interval $[z_{min}, z_{max}]$ of the variable given by the

specific problem, the linear transformation can be given as:

$$z_{ij}(0) = z_{min} + rand(0, 1) \cdot (z_{max} - z_{min}) \qquad (8)$$

where $z_{ij}(0)$ denotes the j-th variable of individual i; rand(0,1) represents the random number with the range of [0,1].

2. Mutation: The different individuals $Z_{r1}(g)$, $Z_{r2}(g)$, and $Z_{r3}(g)$ are selected, and the perturbation vectors are generated according to the following method:

$$v_i(g + 1) = z_{r1}(g) + \eta(z_{r2}(g) - z_{r3}(g)) \qquad (9)$$

where η is the control factor to adjust the amplitude of the individual difference, and $Z_i(g)$ denotes the i-th individual in population g.

During the process of evolution, it is necessary to determine whether the variables satisfy the boundary conditions to ensure the validity of the solution. Otherwise, the variable will be generated randomly repeatedly.

3. Cross operation: The crossover between individuals $\{Z_i(g)\}$ of the g-th generation and its mutant intermediates $\{v_i(g + 1)\}$ can be given as:

$$u_{ij}(g + 1) = \begin{cases} v_{ij}(g + 1), & rand(0, 1) \leq \rho \\ z_{ij}(g) \end{cases} \qquad (10)$$

where ρ is the crossover probability.

4. Select operation: Greedy algorithm is applied to select the individuals with high fitness to enter the next generation:

$$z_i(g + 1) = \begin{cases} u_i(g + 1), & f(u_i(g + 1)) \leq f(z_i(g)) \\ z_i(g), & else \end{cases} \qquad (11)$$

3.4 Self-adaption of scaling factor and cross probability

Since the selection of η and ρ is the key to the behavior and performance of differential evolution and affects the convergence of the algorithm directly, η and ρ should be varied with fitness and evolutionary algebra dynamically.

When the fitness of individuals in the population tends to converge or converges to the local optimal solution, the value of η will increase. Otherwise, η decreases as the fitness of population is being dispersed. At the same time, for the individuals whose fitness is greater than the average, the corresponding value of η is large and the solution should be given up. On the contrary, the individual's fitness is lower than the average value, of which is close to the average degree. Therefore, the adaptive zoom factor can provide the optimal solution with respect to the value of η. The adaptive differential evolution algorithm should maintain the diversity of population as

well as to ensure the convergence of difference. Based on the above analysis, η can be adjusted adaptively according to the following formula:

$$\eta = \begin{cases} \ln \dfrac{f_{avg}-f'}{f_{avg}-f_{best}}, & f' < f_{avg} \\ 1, & f' \geq f_{avg} \end{cases} \quad (12)$$

where f indicates the fitness value of individuals to be mutated, f_{avg} is the average fitness value of the population, and f_{best} is the maximum fitness value in the population. At the beginning of the algorithm, the difference between f_{avg} and f_{best} are very large, and there is almost no possibility of local convergence. With the progress of the generation evolution, the gap between f_{avg} and f_{best} will decrease. Meanwhile, f merges a decrease trend and the speed for converging to the optimum solution will continue to be accelerated gradually, which reduce the risk of falling into local convergence.

The probability of crossover demonstrates the possibility that the genes for mutating individuals can be selected to a new individual. Hence, the adaptive function is presented to adjust the value of ρ dynamically according to the senior generation individual. Initially, ρ is set as a relatively small value ρ_0, which ensures the population diversity with a low probability of crossover. With the evolution of the individual, the individuals begin to converge gradually. At this time, the increase of ρ value will not only improve the variation of gene selection's probability but also speed up the convergence rate. When the $\rho = \rho^*$, the value of ρ will increase no longer and remain stable. The value of ρ can be given as:

$$\rho = \begin{cases} \dfrac{G}{\ln(100-G)} + \rho_0, & \rho < \rho^* \\ \rho^*, & \rho \geq \rho^* \end{cases} \quad (13)$$

where G indicates the generation.

The value of sample weights is equal to 0 or 1, and value 1 denotes that the corresponding sample can satisfy with the filtering condition. When the number of sample weights reaches to N or the maximum generation G, the evolutionary algorithm terminates. Finally, the estimation of node's localization can be obtained by using the sampled nodes being selected optimally from the differential evolution algorithm.

3.5 Obtain the positioning results

Before employing the final N samples to calculate the localization results, the weights of the samples should be normalized as:

$$\tilde{w}_t^i = w_t^i / \sum_{i=1}^{N} w_t^i \quad (14)$$

Then, the estimation of the coordinate of the node i at time t can be obtained by the N samples:

$$\begin{cases} x_i^t = \sum_{j=1}^{N} w_t^j \cdot x_t^j \\ y_i^t = \sum_{j=1}^{N} w_t^j \cdot y_t^j \end{cases} \quad (15)$$

Considering that the actual position of the nodes be included in the constraint sampling box, the range of $(x_{\min}, x_{\max}; y_{\min}, y_{\max})$ denotes that all the samples are obtained in the sample box. The sample within the range must satisfy the constraints of the weighted average. Therefore, the actual node's localization error must be less than the maximum difference between the estimation of node's position and the boundary coordinates of sampling box. We have:

$$\begin{cases} err_{x,\,\max}^t = \max(\tilde{x}_t - x_{\min}, x_{\max} - \tilde{x}_t) \\ err_{y,\,\max}^t = \max(\tilde{y}_t - y_{\min}, y_{\max} - \tilde{y}_t) \end{cases} \quad (16)$$

4 Experiments and results

In this section, we will conduct the experiments to compare our algorithm with the traditional method, for example, RMCL [18] and DLS [19] in terms of the localization precision, sample size, maximum velocity, and the density of anchor nodes. In order to verify the indicators objectively, the parameters of the experiments are set to be identical in different scenarios. The specific parameters are set in details as follows: the number of sensor nodes is 320, the deployment area is 500×500 m^2; the transmission radius of sensor node is $r = 50$ m; the number of valid samples is $N = 50$. The motion process of the node employs RWP model [20], and the maximum moving speed $v_{\max} = 0.2r$. Besides, the main parameters of differential evolution are set as: $\eta = 0.8$, $\rho_0 = 0.4$, and $\rho^* = 0.9$.

To evaluate the effects on the localization algorithm by setting different parameters, the average localization error and the number of candidate samples are regarded as key indicators. Among them, the number of candidate samples reflects the number of times as the sampling process is executed for obtaining the valid samples. Usually, the lesser the number of candidate samples is, the higher the success rate of sampling can obtain.

Figure 1 shows the localization performance of the samples of different generations with differential evolution.

Once the sampling box and a certain number of initial samples are acquired, the proposed differential evolution
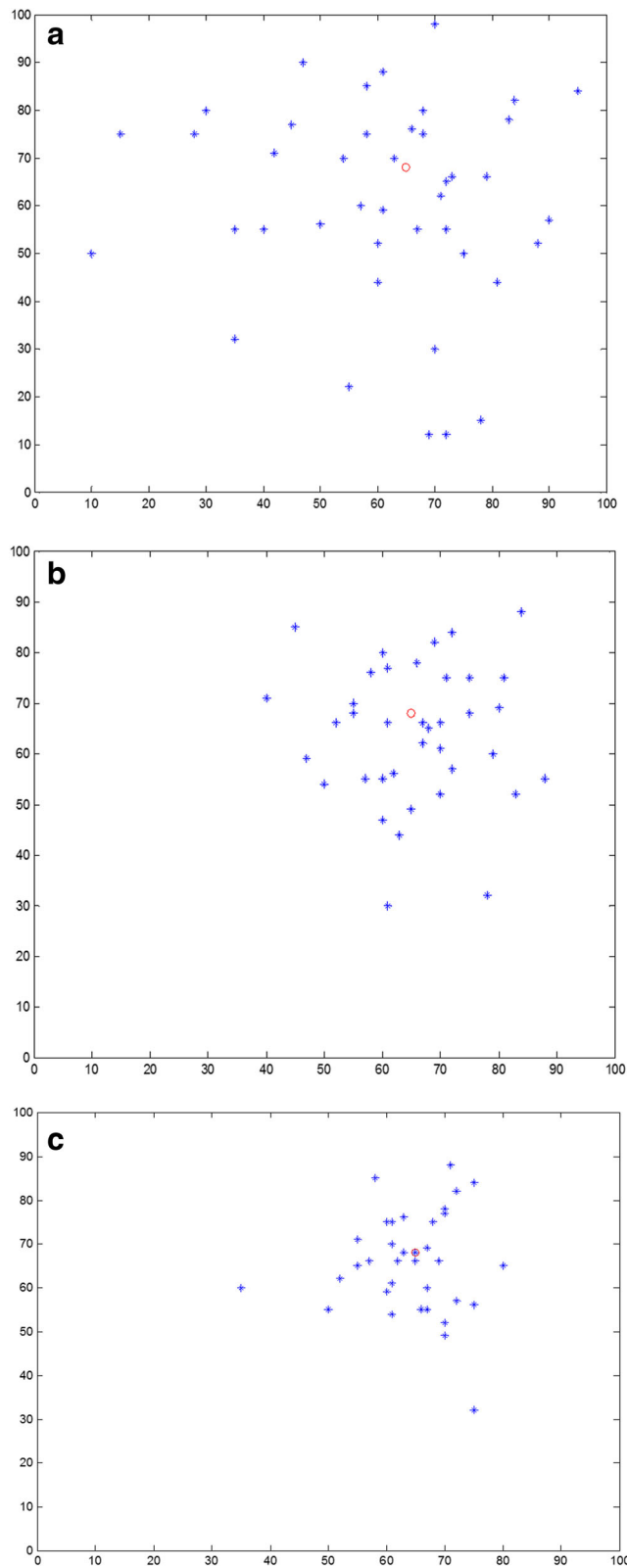
Fig. 1 Localization performance of the samples of different generations with differential evolution. **a** Generation = 0. **b** Generation = 10. **c** Generation = 20

algorithm will be executed, which uses the sample weight as objective function. Since the observation results from the normal nodes only involve anchor nodes, the sample weight is equal to 0 or 1. Next, the derived sample weight is equal to 1 and it can be remained in the next generation. Otherwise, the parents' sample can only reserve in the next generation. After several generations being produced, plenty of samples can be satisfied with the filtering conditions, most of which is close to the actual position of the node. As can be seen from Fig. 1, compared with the initial samples, the samples of subsequent generation with differential evolution are closer to the actual location of the node.

Figure 2 shows the comparison of convergence of the algorithm. As can be seen from the result, the localization accuracy of all algorithms demonstrates absolutely low-quality in the initial stage. With the time goes, the number of valid samples is getting more and more, and the accuracy is corrected by iteration. Therefore, the localization error decreases and fluctuates in a relatively stable range. During the whole convergence phase, we can observe that the average error in MCL-DE shows approximately 21.23% lower than RMCL, about 35.42% lower than DLS. Generally, the speed and direction of the node are unknown except for the maximum velocity. The maximum velocity and previous position can be utilized to predict the constrained sampling area in current time.
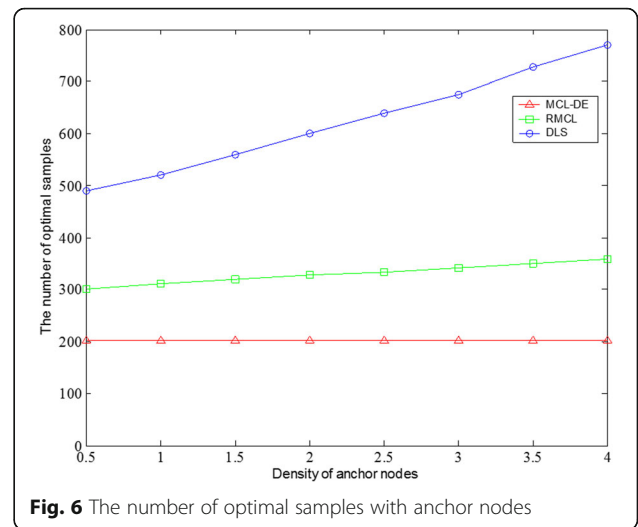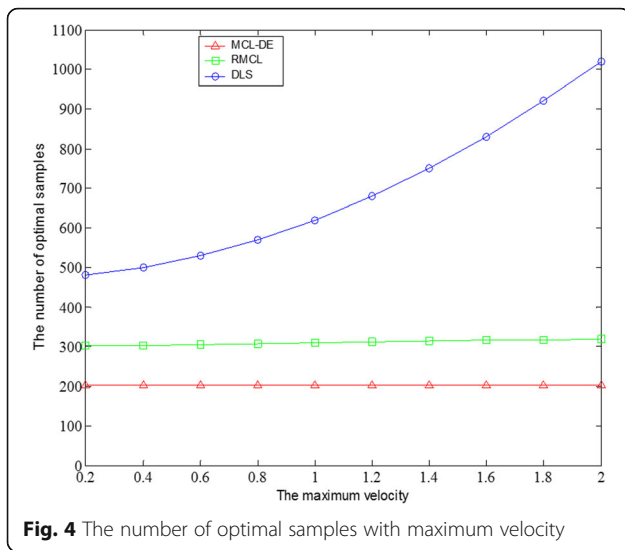
When the maximum moving speed is slow, the size of the sampling box based on the moving velocity is relative small. Apparently, the actual position of the nodes will be of high probability to fall outside of the sampling box and the localization results will be very poor. With the increase of the maximum velocity, the node will be able to establish a better sampling box, which can contain the actual position with high probability, to reduce the localization error. However, excessive value of the maximum velocity of node can constrain the size of sampling box and result in the increase of the average node's localization error and it can be seen from Fig. 3. In the process of the whole trend, the MCL-DE algorithm always keeps the advantage over other methods in localization accuracy.

The number of the candidate samples depends on the size of sampling box, and Fig. 4 demonstrates the relationship between the maximal moving velocity and the number of candidate samples. For DLS, since the effect of sampling is entirely determined by the maximum velocity of the node, the increase of the maximum velocity of movement will extend the range of the sample box as well as increase the number of final candidate samples. However, the sampling constraint box of MCL-DE is mainly composed of one-hop and two-hop neighbor anchor nodes, and the size of the sample box will be primarily restricted by the anchor nodes. That is, when the maximum velocity of motion is increased, the number of candidate samples will not change obviously due to the stable observation results from the anchor nodes.

Next, the experiment is conducted to verify the effect on localization error in different density of anchor nodes. As can be seen from the Fig. 5, the samples being selected optimally can approximate to the actual position, and the results with weighted average calculation make the average localization error as small as possible. From the results, the trend of average localization error in all algorithms shows very clearly. Comparatively, the average location error in different densities of anchor nodes in MCL-DE is less than RMCL and DLS. Especially, when the density of anchor nodes is set to 1, the average localization error of MCL-DE can be less than RMCL about 16.59% and DLS about 33.43%.

Fig. 2 Localization error with time steps

Fig. 3 Localization error with maximum velocity

Fig. 4 The number of optimal samples with maximum velocity

Fig. 6 The number of optimal samples with anchor nodes
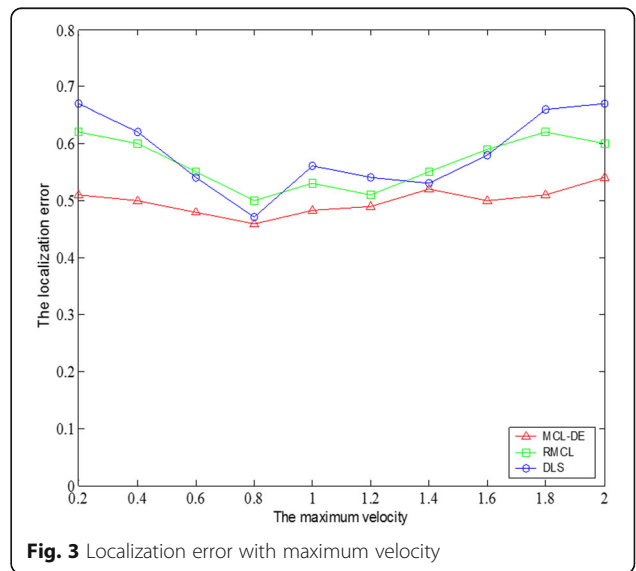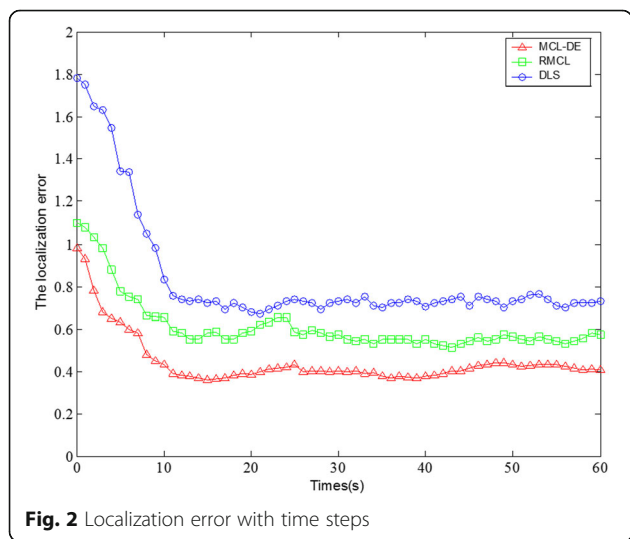
Moreover, the increase of the anchor node's density will bring about more constraint conditions for filtering process. As a result, the filtration rate of samples will increase. As shown in Fig. 6, with the increase of anchor node's density, the number of candidate samples in RMCL and RMCL grows remarkably. In contrast, the growth of DLS algorithm is greater because it does not make use of the anchor node's information at the sampling stage. Due to fixed samples for initial operation, the number of candidate samples remains constantly.

5 Conclusions

After obtaining a certain number of initial samples, we select the weight of the sample as the objective function of the optimization and present the differential evolution algorithm to obtain valid samples rather than perform initial sample filtering and resampling. Finally, the node position is estimated by making the sample close to the

Fig. 5 Localization error with anchor nodes

actual location of the node instead of being filtered out. The simulation result demonstrates that the proposed algorithm provides a better position estimation with less localization error. In the future, we will study and validate the signal strength indicator to improve the performance of our method in aspect of computational and communication costs. And we are also planning to discuss the challenges and open research issues related to the parameters and focus on the localization accuracy of range-free schemes.

Authors' contributions
QM and RBZ contributed to the conception and algorithm design of the study. RBZ contributed to the acquisition of the simulation. QM contributed to the analysis of simulation data and approved the final manuscript. Both authors read and approved the final manuscript.

Authors' information
Qin Miao is an undergraduate student at the School of management in Wuhan University of Technology. His research interests lie in wireless sensor networks, information management, and economic forecast.
Rongbo Zhu received the B.S. and M.S. degrees in Electronic and Information Engineering from Wuhan University of Technology, China, in 2000 and 2003, respectively, and the Ph.D. degree in communication and information systems from Shanghai Jiao Tong University, China, in 2006. From August 2011 to August 2012, he was a research scholar in the Bradley Department of Electrical and Computer Engineering, Virginia Tech, USA. His research interests include mobile computing, protocol design, and performance optimization in wireless networks. He has published over 70 papers in international journals and conferences in the areas of mobile computing and wireless communications and networks. He received the Outstanding B.S. Thesis and M.S. Thesis awards from Wuhan University of Technology in 2000 and 2003, respectively. Dr. Zhu serves as an Editorial Board member of 6 international journals and the Lead Guest Editor for 5 international journals. He has been actively involved in around 30 international conferences.

Competing interests
The authors declare that they have no competing interests.
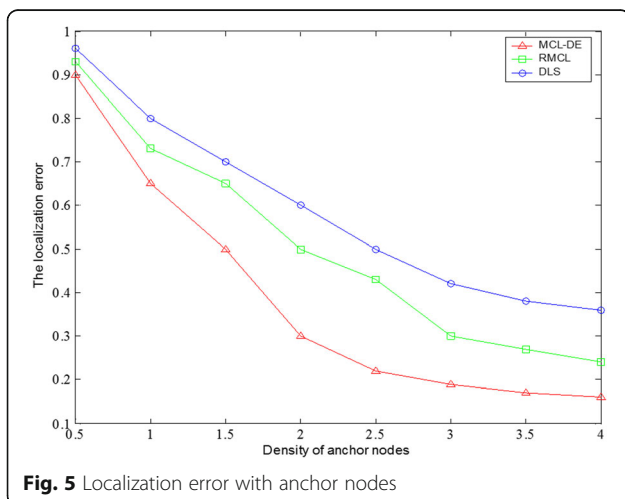
Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]School of management, Wuhan University of Technology, Wuhan 430070, China. [2]College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China.

References

1. R Stoleru, T He, SS Mathiharan, SM George, JA Stan-kovic, Asymmetric event-driven node localization in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **23**(4), 634–642 (2012)
2. A Kumar, A Khosla, JS Saini, SS Sidhu, Range-free 3D node localization in anisotropic wireless sensor networks. Appl. Soft Comput. **34**(2), 438–448 (2015)
3. F Shahzad, TR Sheltami, EM Shakshuki, Effect of network topology on localization algorithm's performance. J. Ambient Intell. Humanized Comput. **7**(3), 445–454 (2016)
4. N Iliev, I Paprotny, Review and comparison of spatial localization methods for low-power wireless sensor networks. IEEE Sensor **15**(10), 5971–5987 (2015)
5. SP Singh, S Sharma, Range free localization techniques in wireless sensor networks: a review. Procedia Comput. Sci. **57**(2), 7–16 (2015)
6. I. GCuvenç, and C. C. Chong, A survey on TOA based wireless localization and NLOS mitigation techniques, IEEE Commun. Surveys Tuts., **11**(3): 107–124, 2009.
7. KC Ho, XN Lu, L Kovavisaruch, Source localization using TDOA and FDOA measurements in the presence of receiver location errors: analysis and solution. IEEE Trans. Signal Process. **55**(2), 684–696 (2007)
8. YS Lee, JW Park, L Barolli, A localization algorithm based on AOA for ad-hoc sensor networks. Mobile Inf. Syst. **8**(1), 61–72 (2012)
9. J Teng, H Snoussi, C Richard, R Zhou, Distributed variational filtering for simultaneous sensor localization and target tracking in wireless sensor networks. IEEE Trans. Veh. Technol. **61**(5), 2305–2318 (2012)
10. I Sharp, K Yu, T Sathyan, Positional accuracy measurement and error modeling for mobile tracking. IEEE Trans. Mobile Comput. **11**(6), 1021–1032 (2012)
11. Z Xiao, H Wen, A Markham, et al., Non-line-of-sight identification and mitigation using received signal strength. IEEE Trans. Wirel. Commun. **14**(3), 1689–1702 (2015)
12. W Cui, CD Wu, W Meng, et al., Dynamic multidimensional scaling algorithm for 3-D mobile localization. IEEE Trans. Instrum. Meas. **65**(12), 2853–2865 (2016)
13. BS Chen, CY Yang, KF Liao, Mobile location estimator in a rough wireless environment using extended Kalman-based IMM and data fusion. IEEE Trans. Veh. Technol. **58**(3), 1157–1169 (2009)
14. JY Ru, CD Wu, ZX Jia, et al., An indoor mobile location estimator in mixed line of sight/non-line of sight environments using replacement modified hidden Markov models and an interacting multiple model. Sensors **15**(6), 14298–14327 (2015)
15. K Shrawan, DK Lobiyal, An advanced DV-hop localization algorithm for wireless sensor networks. Wireless Pers Commun. **71**(2), 1365–1385 (2013)
16. B-N Vo, S Singh, A Doucet, Sequential Monte Carlo methods for multi target filtering with random finite sets. IEEE Trans. Aerosp. Electron. Syst. **41**(4), 1224–1245 (2005)
17. Pezzilli R, Venturi M, Morselli-Labate A M, et al, Differential evolution algorithms solving a multi-objective, source and stage location-allocation problem. Ind. Eng. Manag. Syst. **14**(1),11–21, (2015)
18. Y Wang, Z Cai, Y Zhou, et al., An adaptive tradeoff model for constrained evolutionary optimization. IEEE Trans. Evol. Comput. **12**(1), 80–92 (2008)
19. J-P Sheu, W-K Hu, J-C Lin, Distributed localization scheme for mobile sensor networks. IEEE Trans. Mob. Comput. **9**(4), 516–526 (2010)
20. T Camp, J Boleng, V Davies, A survey of mobility models for ad hoc network research. Wirel. Commun. Mob. Comput. **2**(5), 483–502 (2002)