Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 613–621
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.20653.2188

New algorithm for robot localization based on BrunsVigia optimization algorithm

Manizheh GhaemiDizaji, Chitra Dadkhah*

Faculty of Computer Engineering, K.N. Toosi University of Technology, Tehran, Iran

(Communicated by Madjid Eshaghi Gordji)

Abstract

Two problems with Particle Filters (PF) are particle impoverishment and degeneracy. Resampling is introduced to solve degeneracy problem which happens when the majority of the particles have very small weight and a few particles have large weights and sample's weight variance is too high. Resampling ignores the less informative particles by replacing them with the better ones but it can results in sample impoverishment or diversity loss problem in the particles if there is no controlling mechanism. BrunsVigia Optimization Algorithm (BVOA) is applied in this paper as an extra step to Pf in order to avoid these problems. Operators of BVOA balance between exploration and exploitation and as the result the optimized PF will put much emphasize on more informative particles while keeping the diversity among them. The optimized PF using BVOA, namely BVOA_PF, is tested in localization problem in a simulating environment. Application of BVOA_PF in localization and comparing the simulation results with two well-known optimization algorithms as PSO and GWO verify the efficiency of BVOA in real applications like robot localization.

Keywords: Brunsvigia Optimization Algorithm, Evolutionary particle filter, Robot Localization
2020 MSC: 65Y04, 90C31

1 Introduction

With recent advancements in hardware industry and availability of high speed instruments, using methods with proven theoretical capability in solving problems seems a good choice for computer society. The progress in accuracy and performance of such devices removes burdens from engineering teams when they want to focus on more efficient algorithms. But one should consider the economical cost for frequent infrastructure upgrades. In other words, a good designer should consider the trade off between employing new methods or algorithms and the cost for hardware replacement. This paper is an attempt to optimize an estimation method so that it can work more efficiently with the available hardware. More precisely, our focus is to optimize the distribution of the particles in particle filter so that the estimation will need less particles for solving mobile robot localization and SLAM as two well-known estimation problems in robotics.

Estimation is the problem of guessing the value of variables with the help of some available noisy measurements [13]. In other words, if the value of interest can not be directly measured, some other information which may be noisy will be used to estimate the values indirectly. If the probability of a variable is calculated from the probabilities of

*Corresponding author
Email addresses: m_ghaemi@email.kntu.ac.ir (Manizheh GhaemiDizaji), dadkhah@kntu.ac.ir (Chitra Dadkhah)

Received: June 2020 *Accepted:* March 2021

earlier time of the same kind, the estimation algorithm is named as Filter. Prior to estimation some assumptions may be made but this will have its pros and cons. Gaussian motion and observation model assumptions and linearity are two assumptions that were made initially to solve estimation problems with KF [9]. Although KFs is applied for many problems but they are reported with weak performance in symmetric places. Also KF family are not successful if the robot is kidnapped and positioned in an unknown place or initially the robot is not provided a knowledge with its initial position to start with. Another limiting assumption for KF family is the Gaussian assumption in motion and observation model noises. In addition, the simulations and studies have shown that KF family perform better in open areas and they lack accuracy in narrow places and when the observation sensor is an ultrasound one. These limitations leads us to methods with less restrictive assumptions like particle filters (PFs).
Particle filter is alternative to Gaussian filters like KF and it is a non-parametric method for estimation. Particle filter uses finite number of values instead of fixed form of functions for states and it doesn't make assumption about the form of the posterior density; instead it relies on the drawn samples from the posterior density. In PF the samples drawn from the posterior density function are called particles and they are shown as the following where M is the number of particles: $X_t := x_t^{[1]}, x_t^{[2]}, ..., x_t^{[M]}$ Each member of particles ($x_t^{[i]}$) is a hypothesis about the true state variable. The number of particles is often chosen to be a large number like 1000 and this article is an attempt to reduce this number to a very small value.
The important part of PF resides in importance resampling. In this step, the particles are sampled with replacement from the sample set for M times where the probability of drawing each sample is proportionate to its weight. After performing this step, the number of the particles remains fixed but their distribution changes. It is obvious that the resulting sample set will include duplicates of some particles. Resampling implements the survival of the fittest where the particles are forced to high probability regions. One of the considerations about PF is that its performance depends to the number of particles.

PF relieves the limitations of KF but it needs considerations to avoid two problems of PF namely particle degeneracy and particle impoverishment. The former problem occurs when most of the particles have very small weight and they will have nearly no impact in the estimation process [1]. To state the problem in other way, if the likelihood resides in the tail of the transition probability density of the system state, the obtained samples fail to accurately show the true posterior probability density and few samples get large weight while the majority get small weight and this is what degeneracy problem means [11]. One cure is to resample the particles based on the idea for evolutionary algorithms. This way less important particles will be replaced with important ones. This is done by sampling with replacement according to the weight value of the particles. Sample impoverishment occurs if resampling is done without considering the diversity in the particles. In other words, applying resampling without a controlling mechanism will decrease the samples' weight variance to nearly zero and there will be no diversity in the particles. Different methods for resampling have been introduced so far and more details can be found in [11]. Other solution is particle distribution optimization with the help of evolutionary and swarm algorithms like GA, PSO, and BVOA which is the aim of this article.

BrunsVigia Optimization Algorithm (BVOA) [4] is a newly introduced optimization algorithms. BVOA is compared with some other algorithms like CA, ICA, AAA, ABC, KH, MVO, WOA and the simulation results in many test functions showed their acceptable performance. BVOA has information sharing and angular movements as its main steps. The pseudo code of BVOA is presented in Table 1.

Particle filters like evolutionary/swarm intelligence (EA/SI) optimization algorithms aim to find the best solution. It is the best estimate for PF and the best near-optimal solution for EA/SI. Other similarity is their population based search and the share the survival of the fittest principal. As the result PF suffers from the same problem in EA which is getting stuck in some points as the result of diversity loss in the population. These similar behaviors will help us to employ the constructive strategies in one of them to improve the other or they can be combined for the better results.

The overall structure of the paper is as follows. Section 2 summarizes some available methods for localization problem. Section 3 introduces the optimized particle filter and its application in robot localization problem. Simulation results of the optimized PF in localization problem and comparisons with PSO and GWO based PFs are presented in Section 4 and Section 5 is the conclusion and future work section.

2 Overview of localization problem in robotics

The localization problem is to estimate the posterior distribution over the space of all poses of the robot based on the available data [9]. In other words, localization is the problem of tracking the robot pose with noisy sensors and noisy motor for moving. Sensor data includes data from motion sensors showing the change of the situation like odometer readings and perception sensors like camera images, laser range scans, and ultrasound measures.There are

Table 1: Pseudo Code of BVOA [4]

Procedure BVOA(*num_moves*, vicinity)
Initialize parameters
While stop condition is not satisfied do:
  Share information between the individuals:
  for i=1: $Num\_population/2$
   Randomly choose two individuals
   Exchange the values of some variables between them
  end for
  Angular movement:
  for each population member
   Choose an initial angle
   Choose two random dimensions
   Move the individual according to this angle
   Locally search the vicinity of the best solution
  end for
  Population limiting
end While
Return the best so far solution

different types of localization problem with different levels of difficulties. The most basic one is dead reckoning which is localization based on just robot estimate of its motion like encoder data. Second type of localization makes use of map data to localize itself in the environment. The limitation of this type is the need for apriori map which is not applicable most of the times and such a map will be a static map not reflecting the dynamic changes.

As localization is a filtering problem, so Bayes filter with variants like KF, PF and histogram filter can be applied to estimate the desired values through noisy data. It is obvious that due to the partial observable world stochastic environment and dynamic world, we won't be able to be sure about the next states and the estimation methods just provide a beliefabout the state. Bayes filter for localization tries to find $bel(x_t) = p(x_t|z_{1,t})$ but calculating this formula is time consuming as the time passes and instead a function is used which relies only on the previous belief and the current observation. In other words, Markov assumption is made for both transition model and sensor model. The Markov assumption about transition model states that given the robot pose at time t, all poses at time t+j (x_{t+j}), $j >= 1$ are independent of $x_{0:t-1}$ i. e. $p(x_{t+1}|x_{0:t}) = p(x_{t+1}|x_t)$. The same can be said about the measurements, i.e. $p(z_{t+1}|x_{t+1}, z_{0:t}) = p(z_{t+1}|x_{t+1})$.
$bel(x_t)$ can be calculated through two steps: 1. Prediction step or predicting the pose based on the previous belief and the transition model shown as $\overline{bel}(x_{t+1})$ 2. Correcting or updating the belief based on the current observation model. These two steps are carried out based on (2.1) and (2.2) recursively.

$$\overline{bel}(x_{t+1}) = \int_{x_t} p(x_{t+1}|x_t)bel(x_t) \tag{2.1}$$

$$bel(x_{t+1}) = \eta p(z_{t+1}|x_{t+1})\overline{bel}(x_{t+1}) \tag{2.2}$$

Different filters are a way of representing this belief. That is KF shows the belief with the help of Gaussian distribution and PF shows it with a number of weighted particles or samples.

- KF for Localization KF shows the belief as a normal (Gaussian) distribution with mean as the expected state (robot pose) and a covariance matrix as the error/uncertainty in the estimation. AS stated before, in order to apply KF some assumptions should hold in addition to the Markov assumption. First assumption states that the transition model is a linear Gaussian like (2.3). A is a matrix and Δ is the translation vector with ϵ as the uncertainty.

  $$x_{t+1} = A_{t+1}x_t + \Delta_{t+1} + \epsilon_{t+1} \tag{2.3}$$

  The second assumption is the normally distributed observation model as (2.4). ζ is the unpredictable Gaussian sensor noise.

  $$z_{t+1} = B_{t+1}x_{t+1} + \zeta_{t+1} \tag{2.4}$$

  The third assumption is that the initial belief ($bel(x_0)$) is normally distributed. KF computes a value named as Kalman gain which shows how important the new measurement is. If the measurement uncertainty is high so

the Kalman gain is low and vice versa. As the result if Kalman gain is low so the new measurement will have low impact in the new belief calculation. The assumptions for KF can be relieved while using Particle filter so that the distributions can be any arbitrary one.

- Particle filter for Localization Particle filter is an effective estimation algorithm for localization where the linearity of the system is violated. This way the belief can be shown with a set of weighted samples called particles each considered as the guess of the actual state. This filter acts like an evolutionary algorithm so that a sample will survive as a member of particle set if it shows the correct state according to the measurement.

Different categories of algorithms have been introduced for localization problem namely multi-hypothesis KFs, grid-based localization and Monte Carlo Localization methods [9]. Multi-hypothesis KF considers mixture of Gaussian distributions.Grid-based method uses a set of point probabilities to show distributions of all possible poses. In other words the state space is divided to some grids and the probability of being in each cell is computed. Monte Carlo localization which is the focus of this paper represents distributions with a set of samples while working on raw sensor data and considering arbitrary noise distributions and non-linear space state.
Like other engineering domains, EA/SI have been applied in robotics problems too. One of the uses is the combination of GA in robot localization [9] in which GA assists the non-linear Extended KF (EKF) to localize the robot. Other example is conducted by authors in [15] who integrated PSO in a tracking system and it is used for both local search to introduce initial robot pose and also for tracking robot pose. Authors applied COA in [5] to improve the performance of path planning algorithm. Another robot related applications with EA/SI can be found in [10] who have proposed a modified Grey wolf optimization algorithm based particle filter and compared their results with PSO-based PF, firefly algorithm-based PF, and Spider monkey optimization assisted particle filter. Ant Colony Optimization (ACO) has been also applied prior to the update stage of PF to address the sample impoverishment problem [16]. In [7] ACO is applied to introduce an improved estimator. Other example is Improved Flower Pollination Algorithm (IFPA) based PF [2], firefly optimization based PF (FAPF) [3], ICS-PF or improved cuckoo search (ICS) based PF [14]. A cure for sample degeneracy problem has been introduced in [1] by applying GA with the particle weight as the fitness value for the GA. The same methodology can be found in [9] which introduces evolutionary aided localization filter or Evolutive Localization Filter(ELF). In all of these researches, EA/SI conducts efficient search in the state space to find the most probable estimate for the robotics problem.

3 The proposed optimized particle filter using BVOA: BVOA_PF

In contrary to KFs which consider Gaussian and linear model, Particle Filters (PF) are employed when dealing with non-Gaussian and nonlinear dynamic models [10]. Each state in PF is shown with a weighted sample set and theoretically there should be an infinite number of samples or particles to infer the true state of the system which is practically impossible and there is always a limit on the particle size. One solution is to put emphasis on more informative particles and ignore the rest. This is what evolutionary algorithms do by applying some operations on a population of potential solutions for some iterations.
This paper employs the newly introduced BVOA algorithm [4] to optimize the particle distribution and the optimized PF is applied for solving localization problem in simulation environment. To do so, the particle set will be the initial population for the optimization algorithms and it will run for limited number of iterations. Applying optimization algorithms to assist the particle filter reduces the number of needed particles and saves the memory and speeds up the computations while increasing the estimation accuracy. The block diagram for the whole process is shown in Figure 1. As it is obvious from Figure 1, BVOA is combined with PF as an extra step before the resampling step is performed. This combination will give the informative particles more chance to be improved or be selected for the next step. Pseudo code of the proposed optimized particle filters is shown as Table 2. In the following the proposed optimized PF is applied in robot localization problem and its performance is evaluated in a simulating environment.

3.1 Application of the proposed $BVOA\_PF$ on mobile robot localization problem.

PF or Sequential Monte Carlo method can perform better if the number of samples is high or if the samples are optimized. One of the methods to solve the localization problem is applying PF as an estimation algorithm. Particles in localization problem will show the robot location estimates and the aim is to reach to the set of particles which can estimate the location of the robot more accurately and with less error. To do so, $BVOA\_PF$ is employed on mobile robot global localization problem where the robot is equipped with map of the environment and the goal is to localize the robot according to a global reference frame. PFs perform better in low dimensional problems like localization in which the estimation is done in 3 dimensions as (x, y, α). The employed particle filter for localization is as Table 2

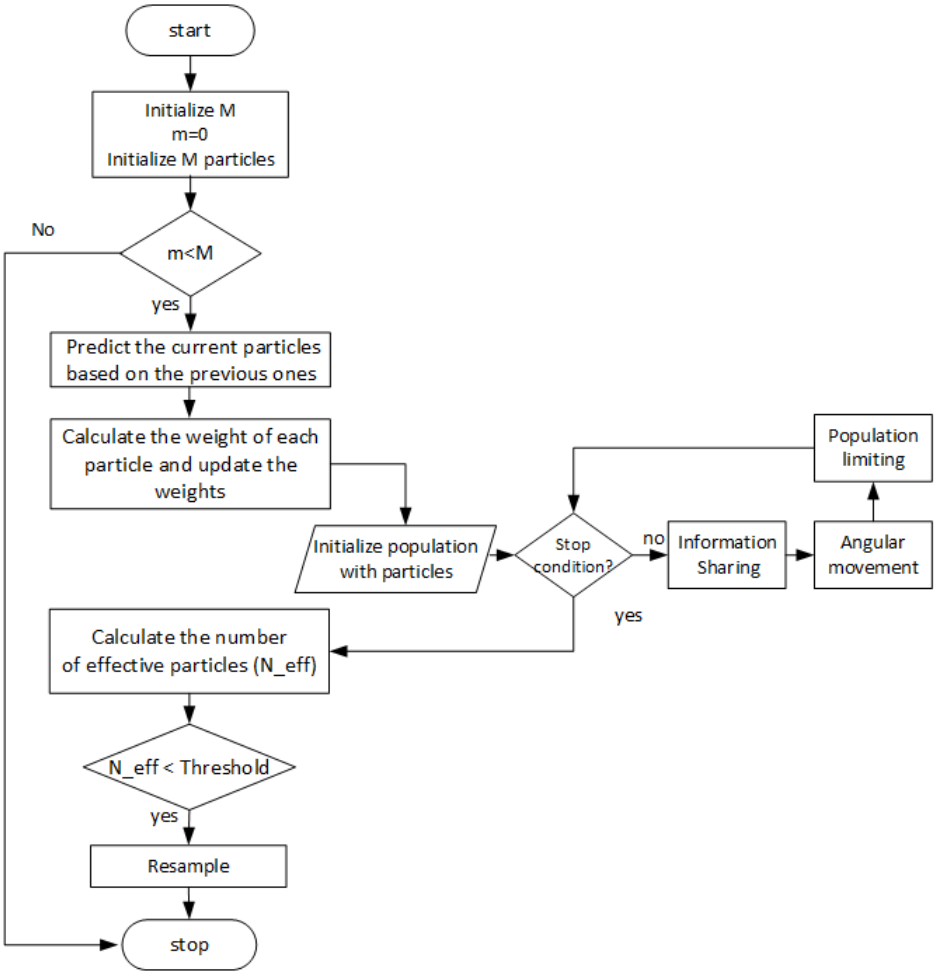

Figure 1: Block diagram of BVOA_PF.

Table 2: Pseudo Code of *BVOA_PF*

Procedure *BVOA_PF*
Initialize M particles
for m=1 to M do:
  Predict the current particles considering the previous particle set
  Correct/Update the particles according to observation model.
  Optimize the particles by applying BVOA
  Consider the particles as the initial population of BVOA
  for i=1 to iterations do:
    Information sharing between the individuals
    Angular movement
  end for
  Calculate the number of effective particles (N_eff)
  if N_eff ¡Threshold
    Resample particles
  end if
end for

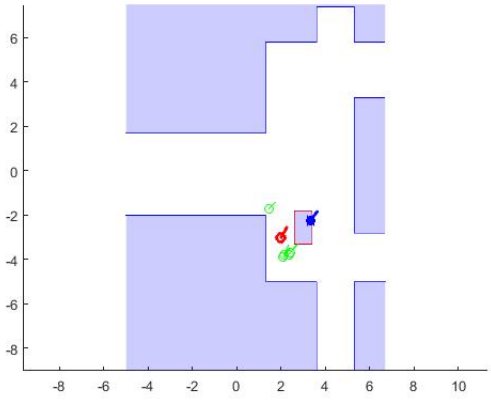

Figure 2: Environment for localization problem with an occupancy grid map. The shaded areas represent occupied cells; the white area represents the free space and the robot is shown with a circle and an arrow as the orientation.

where the input of the algorithm is the map of the environment as well as the control information and the motion and observation models. The output of the algorithm is the robot location estimate. In the following the simulation setup and the results are presented.

4 Simulations and results

BVOA optimization algorithm take the particles of PF for localization as the initial population for optimization and it applies the optimization process on the population for a limited number of iterations. The simulation environment for localization problem is the same setup as in [6]. In the following, simulation environment and parameters are presented in details and results are reported accordingly. All the simulations are performed with MATLAB R2015a and machine with Intel Core i3 CPU (2.4 GHz) with 2GB memory under Windows 10 operating system.
The parameter values will affect the performance of the algorithm. Authors have chosen different values for the number of particles of PF. For example the particle number is 100 in [9] and the optimization algorithm is run for 10 iterations. In [3] and [12] the authors used 20 particles for their Firefly-based tracking system and for SMO-based algorithm respectively. The simulation results for our experiments are for different number of particles like 5, 10, 20, 50, and 100. The comparing algorithms are PSO (Particle Swarm Optimization) and GWO (Gray Wolf Optimization) [8]. The population size for the optimization algorithms can be different than the number of particles. BVOA is reported to have better performance with less population size than others, so for simulations of BVOA half of the particles are used for optimization and for other comparing algorithms the whole particle set is considered as the initial population.

Fitness function: Optimization algorithms need a fitness function to decide on the survival of the individuals in the population. The fitness function for localization problem in this article is the sum of squared error between the estimated robot location and the true location of the robot in the provided map of the environment plus the estimation error in observations defined as Eq.(4.1) [9], p is the true robot position and p' is the estimated position of the robot and $Particle_i$ is the ith particle. Both p and p' have the values for (x,y) in the coordinate system. $z_i s$ are the observations.

$$fitness(Particle_i) = (\overrightarrow{p'} - \overrightarrow{p})^2 + \Sigma_{i=1}^{N}(z_i - z_i')^2. \tag{4.1}$$

Simulation environment: In order to investigate the performance of the optimized particle filter, a simulation environment for localization problem is selected as Figure 2 with known origin and destination as in [6]. MATLAB code of the localization algorithm is taken from
https://github.com/UTSCAS/Robot-Localization-examples. The ground truth for the robot pose is shown with red mark in Figure 2 and the blue circles are the estimated robot pose.

Simulation parameters: The parameter values for the simulations are as follows: control noise is 0.2, laser range is from -90 to 90 degrees, and the scan is done every 30 degrees. Particle range (radius) is 5 Robot initial pose is $(2, -3, \pi/3)$.
BVOA is applied with the following parameter values: Population size is the half of the particles size. The problem

Table 3: Localization Error values of the moving robot for the environment Figure 2 comparing BVOA_PF, PSO_PF and GWO_PF algorithms

#par.	Algorithm	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
5	BVOA_PF 100%	**41.52**	70.87	**165.16**	**9.86**	**7.35**	**111.83**	**5.79**
	PSO_PF 100%	76.86	186.64	238.17	227.00	287.70	312.32	137.96
	GWO_PF 100%	59.14	**56.46**	277.56	221.70	349.68	186.35	157.14
10	BVOA_PF 100%	**26.57**	**25.36**	**72.00**	43.41	**36.70**	67.24	**3.63**
	PSO_PF 100%	36.47	35.22	100.17	110.33	167.13	172.55	89.35
	GWO_PF 100%	32.16	46.65	80.70	**29.19**	56.84	**58.08**	5.02
50	BVOA_PF 100%	**14.66**	**8.48**	**19.44**	**2.62**	**0.26**	**0.10**	**0.12**
	PSO_PF 70%	26.86	17.30	32.19	8.22	4.07	3.75	2.72
	GWO_PF 70 %	20.94	18.22	25.00	8.73	2.08	0.45	0.33
100	BVOA_PF 100%	17.28	**2.69**	14.93	**0.16**	**0.04**	**0.08**	**0.10**
	PSO_PF 50%	22.50	8.72	12.45	1.37	0.73	0.29	0.30
	GWO_PF 60%	**16.87**	7.11	**11.38**	3.65	0.14	0.16	0.13

dimension is 2 and the number of iterations is fixed at 3. This will reduce the computation cost. L is 0.5 for angular movement stage of BVOA. Information sharing between two individuals A and B is performed according to Eq. (4.2) and $F = 0.5$. F is a value drawn from Cauchy distribution with (μ, σ) as mean and standard deviation respectively. *best* denotes the best member of the population. *indiv(r1)* and *indiv(r2)* are two randomly selected individuals from the population.

$$\begin{cases} A_i \leftrightarrow B_i & \text{if } rand > 0.5 \\ A_i = A_i + F * (best - A_i) & \\ \quad +F * (indiv(r1) - indiv(r2)) & OW \end{cases} \tag{4.2}$$

For PSO algorithm c1 and c2 parameters are 2 and 1 respectively and the iterations is 3 as for BVOA and GWO. GWO algorithm is inspired from the hunting behavior of wolves. Parameter values for encircling the preys in the optimization process is the same as the original paper. GWO has two parameters to adjust namely a and C. Parameter a is linearly decreased from 2 to 0 to simulate the smooth transition between exploration and exploitation. Parameter C depends on a random value in range [0,1] and is calculated as $C = 2.r$, $r \in [0, 1]$. PSO and GWO use all of the particles of PF as their initial population.

4.1 Simulation results and discussion

This section is devoted to the simulation results of localization problem. Each reported result is the average of 20 independent runs and the best results are highlighted. Robot localization is done with laser scan using particle filter. Grid map of the environment as well as the laser and control data are given to the robot. For comparing the performance of different algorithms, the data for robot true trajectory is also available.
Table 3 shows the estimation error in localization for a simulating robot which moves 7 steps from the initial position to reach its destination. For the simulations, different values for the number of the particles are considered as 5, 10, 50 and 100. Each column in Table 3 shows the average error value for each step of the robot. The fitness value of the best particle is considered in each step and the results are the average of 20 independent runs. The success rate for BVOA_PF algorithm is 100% but PSO_PF and GWO_PF get stuck in local minima and their success rate is written in front of the algorithms in Table 3.

According to the simulation results of Table 3, followings are the contribution of applying BVOA in PF comparing the other algorithms:

- Particle filter can be combined with optimization algorithms for better performance.
- Applying BVOA as an extra step for PF can optimize the distribution of the particle set.
- BVOA reduces the error in state estimation by emphasizing on the more informative particles and ignoring the less informative ones by the survival of the fittest principle implemented in BVOA.
- Applying BVOA is an attempt to solve the particle impoverishment problem of Pf.

- Simulation results shows the better performance of BVOA comparing PSO and GWO optimization algorithms for localization problem in robotics.
- Optimization algorithms like BVOA are iterative procedures that need a controlling mechanisms to keep the computational cost as low as possible.
- In order to reduce the computational cost, BVOA and other optimization algorithms iterate for a small number of iterations like 3.
- The simulation results for BVOA are reported considering the half of the particles in PF as the initial population. The results show that BVOA outperforms PSO and GWO as two well-known algorithms in most of the runs even with less population size. This reduces the memory cost as well.
- With rapid improvement in computational power and hardware technology, applying methods to reach better performance is more welcomed than before.
- According to the results, PSO and GWO get stuck in some cases and their success rate is lower than BVOA.

5 Conclusions

Robotics applications are coined with estimation as the gathered data and the instruments are still noisy. As the result optimized estimation algorithms are welcomed to be used in real world applications of robotics. One of the widely considered approaches for optimization in applications is evolutionary/swarm intelligence algorithms as they show promising results with often simple operators. This paper introduced an optimized estimation algorithm namely BVOA_PF which is the combination of PF with the newly introduced BVOA optimization algorithm. The resulting BVOA_PF is tested in localization problem in robotics where the aim is to localize a robot in an environment with a known map. To do so, the robot starts with some initial guesses of PF as its probable initial points and moves in the environment which leads to the correction in its predictions. BVOA is used to optimize the guesses with the help of some operators namely information sharing and angular movement. The simulation results in an simulating environment with a known map verifies the efficiency of the proposed BVOA_PF in robotics application. The results are compared with two other methods based on PSO and GWO optimization algorithms which verify the performance improvement while using BVOA with half of the population of PSO and GWO.
For our future attempt we will test the performance of the introduced BVOA_PF in other robotic estimation problems like mapping and SLAM. Also, decreasing the computational cost of BVOA_PF is our next step.

References

[1] P. Abbaszadeh, H. Moradkhani, and H. Yan, *Enhancing hydrologic data assimilation by evolutionary Particle Filter and Markov Chain Monte Carlo*, Adv. Water Resources **111** (2018), 192–204.

[2] M. Gao, J. Shen, and J. Jiang, *Visual tracking using improved flower pollination algorithm*, Optik **156** (2018), 522–529.

[3] M. L. Gao, L. L. Li, X. M. Sun, L. J. Yin, H. T. Li, and D. Sh. Luo, *Firefly algorithm (FA) based particle filter method for visual tracking*, Optik **126** (2015), no. 18, 1705–1711.

[4] M. Ghaemidizaji, Ch. Dadkhah, and H. Leung, *A New Optimization Algorithm Based on the Behavior of BrunsVigia Flower*, IEEE Int. Conf. Syst. Man Cyber. (SMC). IEEE (2019), 263–267.

[5] S. Hosseininejad and Ch. Dadkhah, *Mobile robot path planning in dynamic environment based on cuckoo optimization algorithm*, Int. J. Adv. Robotic Syst. **16** (2019), no. 2, 1729881419839575.

[6] Sh. Huang and G. Dissanayake, *Robot Localization: An Introduction*, Wiley Encyclopedia of Electric. Electron. Engin. (2016), 1–10.

[7] S. M. Kalami Heris and H. Khaloozadeh, *Ant colony estimator: an intelligent particle filter based on ACOR*, Engin. Appl. Artif. Intell. **28** (2014), 78–85.

[8] S.A. Mirjalili, S. M. Mirjalili, and A. Lewis, *Grey Wolf Optimizer*, Adv. Engin. Software **69** (2014), 46–61.

[9] L. Moreno, S. Garrido, and M.L. Munoz, *Evolutionary filter for robust mobile robot global localization*, Robotics Autonomous Syst. **54** (2006), 590–600.

[10] M. Narayana, H. Nenavath, S. Chavan, and L K. Rao, *Optik intelligent visual object tracking with particle filter based on modified grey wolf optimizer*, Optik **193** (2019), 162913.

[11] Zh. Qiu and H. Qian, *Adaptive genetic particle filter and its application to attitude estimation system*, Digital Signal Process. **81** (2018), 163–172.

[12] R. Rohilia, V. Sikri, and R. Kapoor, *Spider monkey optimisation assisted particle filter for robust object tracking*, IET Comput. Vision **11** (2017), no. 3, 207–219.

[13] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, *Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association*, J. Machine Learn. Res. **4** (2004), no. 3, 380–407.

[14] G. S. Walia and R. Kapoor, *Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search*, Expert Syst. Appl. **41** (2014), no. 14, 6315–6326.

[15] Q. Zhang, P. Wang, and Z. Chen, *An improved particle filter for mobile robot localization based on particle swarm optimization*, Expert Syst. Appl. **135** (2019), 181–193.

[16] J. Zhong, Y. Fai Fung, and M. Dai, *A biologically inspired improvement strategy for particle filter: Ant colony optimization assisted particle filter*, Int. J. Control Automation Syst. **8** (2010), no. 3, 519–526.