A Biological Inspired Improvement Strategy for Particle Filters

J. P. Zhong, Y. F. Fung
Department of Electrical Engineering
The Hong Kong Polytechnic University
joni.zhong@polyu.edu.hk

Abstract-Particle Filters (PF) is a model estimation technique based on simulation. But two problems, namely particle impoverishment and sample size dependency, frequently occur during the particle updating stage and these problems will reduce the accuracy of the estimation results. In order to avoid these problems, Ant Colony Optimization is incorporated into the generic particle filter before the updating stage. After the optimization, particle samples will move closer to their local highest posterior density function and better estimation results can be produced.

I. INTRODUCTION

Particle Filters (PF), widely used for solving non-linear and non-Gaussian state estimation problems [1], are based on point mass particles representing the probability densities. PF are often recognized as an alternative to the Extended Kalman Filter (EKF) [2] or the Unscented Kalman filter (UKF) [3] in state estimation problems. With sufficient number of samples, PF can approach the Bayesian optimal estimate [4], rather than the EKF or UKF. However, particle impoverishment and particle size dependency are inevitably induced due to the random particles generation and uniform re-sampling applied in generic PF [5]. Some algorithms employ different sampling strategies to minimize the impoverishment, these include Binary Search [6], Systematic Resampling [7] and Residual Resampling [8]. These algorithms achieve their targets by improving the efficiency of particles. However, in the mean while, the robustness of the filtering is lost, because the diversity of particles is reduced in a certain extent [9].

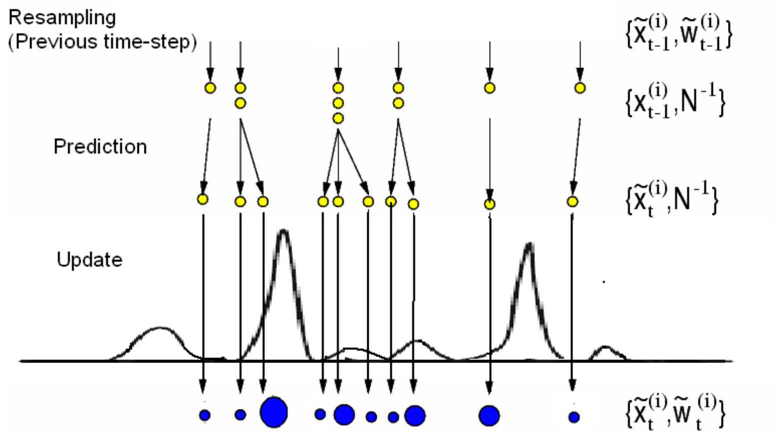
In this paper, the research about the effect of Ant Colony Optimization (ACO) in eliminating these two problems is presented. We will first introduce the particle filters mechanism in next section. The ACO assisted Particle Filter together with a brief introduction of ACO will be discussed in Section III. In Section IV, experiment conducted to study the performance of the ACO assisted PF and algorithm optimizations will be presented. The conclusions are given in Section V.

II. PARTICLE FILTERS

Particle filters include an algorithm to perform recursive Bayesian estimation using Monte Carlo simulation and importance sampling, in which the posterior density is approximated by the relative density of particles in a neighborhood of state space. Since this sampling stage in PF is merely a suboptimal solution, therefore a major short-coming in the generic particle filters, namely particle impoverishment, is induced.

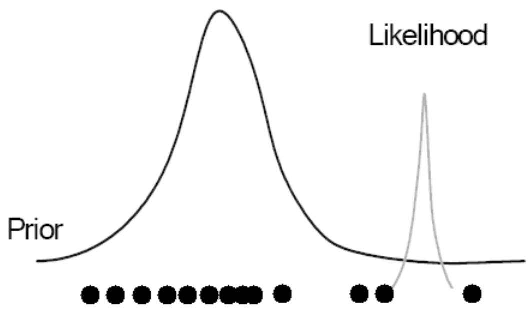
A. Particle impoverishment

Particle impoverishment [5] occurs when the likelihood is so narrow that the overlapping region of likelihood and the prior distribution is quite small as depicted in Fig 1.a. As a result, particle weight of particles which is far away from the region of likelihood become relatively small. Another reason for causing the problem is that the likelihood lies in the tail of the prior distribution (Fig 1.b). If such a situation occurs repeatedly, all but one sample will have negligible weights.



(a) Narrow likelihood

If it is an exact measurement so that the likelihood region is quite small, probably no particles lie within the region of the likelihood. Many particles are wasted in low likelihood area, so particle impoverishment happens



(b) Likelihood lies in the tail of prior distribution

There is another situation that particle impoverishment happens. When new measurements (i.e. the likelihood) appear in the tail of the prior, the particles predicted from the prior density will distribute far from the likelihood.

Fig. 1 Limitations in generic Particle Filter

B. Size Dependency

In addition to particle impoverishment, sample size dependency is another problem. If the particle set size is small, potentially there may not have enough particles that can approach the true state, which will prevent the filter to converge. In generic particle filters, the only way to solve this problem is to enlarge the particle size, but that will increase the computational requirements.

III. ANT COLONY OPTIMIZATION ASSISTED PARTICLE FILTERS

Ant Colony Optimization (ACO) algorithm is a biological inspired system based on agents that simulate the natural behavior of ants [11]. It is derived from the capability of natural optimization of ants. When finding food, they tend to take the best route (or path) between their nest and some external landmark because their particular pheromone trail becomes higher if more ants choose this trail. This iterative process made by ants will achieve suboptimal to optimal trails between endpoints. The ACO algorithm uses the mathematical formulas to simulate this natural optimization process.

A. Ant Colony Optimization in PF

Particle filter is an algorithm to perform recursive Bayesian estimation using Monte Carlo simulation and importance sampling, in which the posterior density is approximated by the relative density of particles in a neighborhood of state space. The PF algorithm is illustrated in the following pseudo-codes.

Algorithm 1: The generic PF

$[\{x_k^i, w_k^i\}_{i=1}^N] = PF[\{x_{k-1}^i, w_{k-1}^i\}_{i=1}^N, y_k]$

Initialization: Generate particle samples $\{x_0^i, w_0^i\}_{i=1}^N$

Prediction:

For i =1:N

---Predict $x_k^i \sim q(x_k^i \mid x_{k-1}^i, y_k)$

---Assign the particle a weight

End For

Measurement update

Calculate total weight: $t = sum[\{w_k^i\}_{i=1}^N]$

For i=1:N

---Normalize: $w_k^i = t^{-1} w_k^i$

End For

Resampling

To optimize the re-sampling step of the generic particle filter, we adopt ACO before the updating step. An ant will replace the randomly-generated particle in the Sequential Monte Carlo concept. They will be relocated based on the following ACO method in order to approach the optimal solution in the resampling step.

The $\tau_j(t)$ as shown in Eq (1), representing the amount of pheromone of particle *j*, which is equal to the weight of particle *j* at the initial stage before the iteration. During an iteration, it is affected when particle *j* is selected as a target of another particle *i to* approach (see Eq.3):

$$\begin{cases} \tau_j(t+1) = (1-\rho)\tau_j(t) + \Delta\tau_i(t) & j \in \text{set of endpoints of movement} \\ \tau_j(t+1) = (1-\rho)\tau_j(t) & j \in \text{set of other particles} \end{cases} \quad (1)$$

where $0 < \rho \leq 1$ is the evaporation rate, $\Delta\tau$ is a enhanced value equaling to p.

Secondly, the heuristic function is defined as the reciprocal of the distance of every two particles (end points):

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \quad (2)$$

The optimization step runs iteratively as follows:

Each particle is recognized as an ant. Before the movement, the probability that a particle *i* will select particle *j* among N-1 particles from the particle set as the moving direction is governed by Eq. (3).

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]_\alpha [\eta_{ij}(t)]_\beta}{\sum_{s \in \text{all particles}} [\tau_{is}(t)]_\alpha [\eta_{is}(t)]_\beta} \quad (3)$$

Every movement velocity is defined as a random number between zero and the difference of original point and target point. It terminates until all particles' positions converge to the high likelihood region (the general or local optimal solution) within a certain threshold, defined in Eq. (4).

$$Threshold^j = \frac{constant\ value}{number\ of\ particles} \quad (4)$$

Therefore, with enough particles in a small region near the true value, we will get a more accurate estimated one.

A pseudo-code description of the optimization algorithm is given below

---

Algorithm 2: The ACO assisted PF

ACO assisted PF

While the distance between particles and their targets are not within a certain threshold (Eq. 4)

---Choose particle *i* whose distance is within the threshold

---Select the moving target based on the probability (Eq. 3)

---Move towards the target with a velocity

---Update the parameters of the ACO, and particle weight

End While

Optimized by the ACO, the particle impoverishment problem is basically alleviated. The particle samples tend to be around high likelihood regions. As a result, most of the particles which scatter far away from the true state will converge to states that represent high probability as shown in Fig. 2. Therefore, when configured with suitable parameters (ρ, τ_0, $\Delta\tau$, constant value in threshold, etc.), ACO has the ability to balance between the diversity and the impoverishment of particle filters.

In addition, since most particles congregate in the high likelihood, the effect of these particles is guaranteed. The sample size dependency problem of the generic particle filter is also improved.

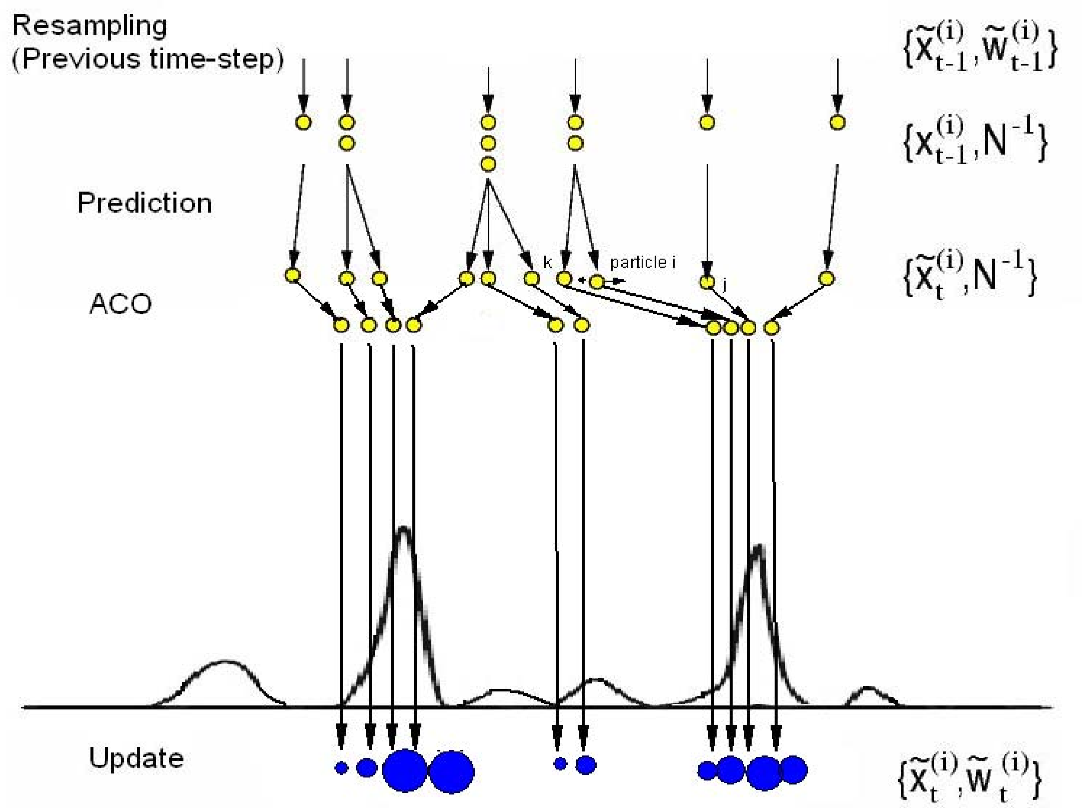


Fig. 2 The ACO Assisted Particle Filter

Because particle j as a moving target has higher weight and shorter distance than the other particles, p_{ij} (denoted by the length of arrow) is larger than other probabilities. Therefore particle i moves towards particle j. Compared to Fig 1.a, particles are closer to the local highest pdf, so that the particle impoverishment is avoided.

IV. EXPERIMENTAL RESULTS

A. *Performance in single variable economic model*

In order to evaluate the performance of the ACO assisted PF, we employed the following nonlinear single variable economic model:

$$x(t) = 6 + \sin(4e - 2\pi t) + 0.5x(t-1) + w(t)$$

$$y(t) = 0.2x(t)^2 + v(t)$$

where w(t) and v(t) stand for the zero-mean Gaussian process noise and measurement white noises respectively. Given the noise measurement, the Extended Kalman Filter, the generic PF and ACO assisted PF were applied to estimate the state sequence x_t for t=1 to 30. The number of particles used in two PFs was 100. In ACO assisted PF, the parameters' values are chosen by random. All the following experiments are executed in Intel Core 2 Duo P8400@2.26G with 2G Ram and Matlab 7.

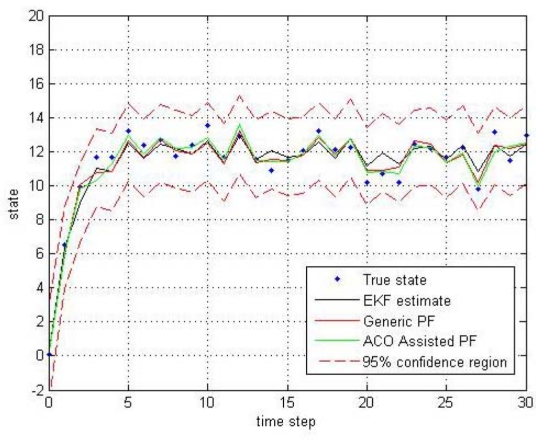


Fig. 3 Tracking of a nonlinear function with different filters

Table I RMS value of error

Filters	RMS Error	RMSE Percentage (KF=100%)
EKF	0.85115	100
Generic PF (N=100)	0.54118	63.58

PF+ACO (N=100)	0.40956	48.12

From Fig.3, we can see that the generic PF cannot track the result from step 9 and 10, it was possible that the particle impoverishment problem mentioned in Section II occurred in these steps. Compared to our ACO assisted PF, based on the same initial particles and observations, tracking performance of the ACO assisted PF is better in the measurement of RMS error (Table I).

In order to investigate the problem of sample size dependency, we used different numbers of particles for the ACO assisted PF and compared the RMS error with the Kalman filter and generic PF with sample size of 100 and the results are given in Table II. As shown in Table II, the RMS error in all cases of the ACO assisted PF are less than that obtained from the original PF. This substantiated that our proposed algorithm can alleviate the problem of sample size dependency and this in turn can reduce the computational time.

Table II

RMS error for different sample sizes

Filters	RMS Error	RMSE Percentage (Generic PF=100%)
EKF	0.8671	136.57
Generic PF (N=100)	0.6349	100
PF+ACO (N=100)	0.4389	69.13
PF+ACO (N=50)	0.6262	98.63
PF+ACO (N=30)	0.6178	97.31

B. Parameters Tuning

In the previous experiments, although our ACO assisted PF only utilized some random parameters, it still performed better than the other two methods. But we need to optimize it for the specific problem shown in Section 4.A by tuning the parameters, i.e. ρ, maximum times of iteration (*mi)*, threshold to terminate the iteration (*ti* divides the total number of particles as Eq. 4). To minimize the effect of the random noise, we enlarged the time step to 1000 as shown in Table III.

The first criteria of our parameter choice was RMS error, but also we did not hope that the time cost exceed 200 seconds, which probably increases our experimental cost. Table III shows that *ρ* and *ti* have more significant influence than *mi*, because the process terminates before executing the maximum number of iteration.(*mi).* Considering both the time cost and the error RMS, we chose the parameters in Group 10 in the problem.

V. CONCLUSIONS

Particle Filters, although are useful to solve the tracking problem in a noisy environment, are easy to cause particle impoverishment and sampling dependency. In this paper, we proposed a new algorithm that utilizes the Ant Colony Optimization to re-arrange the particles before the updating step of particle filters. After the ACO optimization, particles approach the higher likelihood in the *pdf* and this can minimize both problems of particle impoverishment and sample dependency. Our experimental results reflect that the ACO assisted PF produces better tracking performance in solving the nonlinear single variable nonlinear estimation problem. Further more, some optimization methods were tested and validated based on the experiment results.

Table III

Results for parameters change

Parameter	*ρ*	*ti (/N)*	*mi*	*Time Cost (s)*	*Error RMS*
1	**0.1**	100	1000	196.191	0.60205
2	**0.3**	100	1000	202.297	0.60375
3	**0.5**	100	1000	195.761	0.57257
4	**0.8**	100	1000	216.656	0.58317
5	0.5	**100**	1000	203.124	0.59231
6	0.5	**300**	1000	193.441	0.58828
7	0.5	**400**	1000	181.897	0.59982

8	0.5	**500**	1000	220.094	0.62044
9	0.5	300	**1000**	184.56	0.58891
10	0.5	300	**3000**	184.759	0.58542
11	0.5	300	**5000**	184.395	0.59213

ACKNOWLEDGEMENT

The work presented in this paper is supported by the Department of Electrical Engineering of the Hong Kong Polytechnic University.

REFERENCES:

[1] N. J. Gordon, D.J. Salmond and A. F. M. Smith. "Novel approach to nonlinear/non-Gaussian Bayesian state estimation." *IEE Proc-FI,* vol. 140(2), pp. 107-113, 1993.

[2] S. F. Schmidt. "The Kalman filter - Its recognition and development for aerospace applications." J Guid. *Contol Dynm.*, vol. 4(1), pp..4-7, 1981.

[3] E. A. Wan and R. Van der Merwe., "The unscented Kalman filter for nonlinear estimation.." *IEEE Proc Sym. Adapt Sys. Sign*, pp. 153-158, 2000

[4] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners." IEEE Transactions on Signal Processing, *IEEE T Signal Proces.*, vol. 50(3), pp. 736-746, 2002

[5] M.G.S. Bruno and A. Pavlov, "Improved particle filters for ballistic target tracking." *IEEE Proc-Acoust. Speech*, vol. 2, pp. 705-708, 2004

[6] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering." *Stat Comput.*, vol. 10(3), pp. 197-208, 2000

[7] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking." *Int. J. Comput. Vision*, vol. 29(1), pp. 5-28, 1998

[8] J. U. Cho, et al, "A real-time object tracking system using a particle filter." *IEEE Proc-Intell. Robot Syst.*, pp. 2822-2827, 2007.

[9] C. Stachniss, G. Grisetti and W. Burgard, "Recovering particle diversity in a Rao-Blackwellized Particle Filter for SLAM after actively closing loops." *IEEE Proc-Robotic Autom..*, pp. 18-22, 2005.

[10]G. Tong, Z. Fang, X. Xu. "A Particle Swarm Optimized Particle Filter for Nonlinear System State Estimation." *IEEE Proc-Evol. Comput.*, pp. 438-442, 2006

[11] O. Cordon, F. Herrera and T. Stutzle. "A review on the ant colony optimization metaheuristic: basis, models and new trends." *Math. Soft Comp.*, vol. 9, pp. 141-175, 2002

[12] A.Colorni, M. Dorigo and V. Maniezzo, "Distributed optimization by ant colonies." *Proc. Eur. Art. Life*, pp. 134-142, 1992.