Pattern Recognition Letters 116 (2018) 195–199

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Autonomous vehicle routing problem solution based on artificial potential field with parallel ant colony optimization (ACO) algorithm ☆

Fanlei Yan [a,b,*]

[a] Key Laboratory of Advanced Electrical Engineering and Energy Technology, Tianjin Polytechnic University, Tianjin 300387, China
[b] School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin 300387, China

ARTICLE INFO

Article history:
Available online 16 October 2018

Keywords:
Pilot satisfaction
Mapreduce
Parallel ACO
Autonomous flight

ABSTRACT

An autonomous vehicle routing problem solution (VRPS) algorithm with parallel ant colony optimization (ACO) considering pilot satisfaction is proposed in this paper to solve the problem in traditional autonomous vehicle routing, such as the service quality is affected by longer flying time of pilot due to too much emphasis on cost factors and ignoring of delivery time. First, the flying time of pilot is considered to improve traditional autonomous flight model so as to comprehensively consider autonomous cost and pilot satisfaction. Then, for NP problems in autonomous flight, a parallel ACO algorithm is used for global optimization and design Mapreduce improvement method of parallel ACO, to improve optimization performance of the algorithm and to obtain optimal global solution. At last, a contrast experiment is conducted and the results show that, the algorithm proposed herein can effectively reduce the flying time of pilot and help to improve the service quality.

1. Introduction

Vehicle routing problem solution (VPRS) is essential in autonomous flight. Traditional VRPS problem means a delivery optimization process for customers in different geographical locations by a team of flight from one or more warehouses [1]. Lenstra et al. believed that the VRPS problem is substantially an uncertained multinomial NP problem. For large sized problems in autonomous flight, since precise solution requires higher computation cost which does not meet with real time requirements, currently, most of VRPS problems turn to use heuristic and macro optimization algorithms. For example, Arunkumar et al. [2] used a random fuzzy logic guided genetic algorithm to optimize VPRS problem while taking minimizing transport cost as evaluation indicator. Ravichandran et al. [3] used a method in solving the rotation angle with individual mean value in population to improve the quantum evolutionary algorithm and apply it to VRPS problem solution. Du et al. [4] used ACO algorithm to solve the VRPS problem.

In references mentioned above and most of VRPS problems, minimizing the transport distance by flight is always taken as the evaluation indicator, considering the cost [5,6]. However, with the development and ever-increasing competition of logistics industry, the impact of service quality on competitiveness of enterprises cannot be underestimated. Pilot satisfaction is one of key factors affecting the service quality and is mainly affected by flying time during delivery. Many people think that a shorter transport distance means a shorter transport time, but in reality this is not the case. Pilot satisfaction (transport time) should not only affect the decision-making result as a constraint, but should be the most direct decision-making result for target function [7–10].

For selection of optimization algorithms, as a new and intelligent optimization algorithm, harmony search algorithm has been widely applied in various fields, but less applied in the field of optimizing VRPS problems. In addition, harmony search algorithm presents premature convergence. In order to solve the problem like this, a parallel ACO algorithm with Mapreduce improvement is given, and an autonomous VRPS algorithm with parallel ACO and consideration of pilot satisfaction is proposed in this Paper [11–13].

2. QDEMA 2E vehicle routing problem (2E-VRP)

2.1. Problem description

2E vehicle routing problem solution (2E-VRP) is shown in Fig. 1, where is S_1–S_3 Level 1 transfer station and c_1–c_9 is Level 2 user point, the 2E-VRP is distinguished by different line types (solid line for 1E and dotted line for 2E).

2E-VRP mathematical model equals to the minimum delivery time problem, namely minimum total time required for delivery

☆ **Conflicts of interest:** None declared.

* Correspondence to: Key Laboratory of Advanced Electrical Engineering and Energy Technology, Tianjin Polytechnic University, Tianjin 300387, China.
E-mail address: xlcxie99@163.com

https://doi.org/10.1016/j.patrec.2018.10.015
0167-8655/

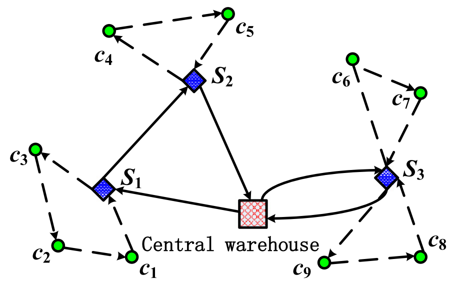

Fig. 1. 2E-VRP problem.

completed by the last vehicle [6]:

$$f = min(C_{\max}) \tag{1}$$

Constraint (2) ensures that each order is processed for once only:

$$\sum_{k=1}^{n} Y_{ik} = 1, i = 1, 2, \ldots, n \tag{2}$$

Constraint (3) ensures that when $\sum_{i=1}^{n} Y_{ik} = 0, k = 1, 2, \ldots, n-1$, $\sum_{i=1}^{n} Y_{i,k+1} \neq 1$, namely if there is no number of order k, there would be no number of order$k+1$.

$$\sum_{i=1}^{n} Y_{i,k+1} \le H \sum_{i=1}^{n} Y_{ik} \tag{3}$$

Constraint (4) ensures that when $\sum_{i=1}^{n} Y_{ik} = 0, y_k = 0$:

$$y_k \le \sum_{i=1}^{n} Y_{ik},\ k = 1, \ldots, n \tag{4}$$

Constraint (5) ensures that when $\sum_{i=1}^{n} Y_{ik} \ge 1,\ y_k = 1$:

$$\sum_{i=1}^{n} Y_{ik} \le H \times y_k,\ k = 1, \ldots, n \tag{5}$$

Constraints (4) and (5) mean, when the order B_k is valid, $y_k = 1$; while when the order B_k is invalid, $y_k = 0$.

Constraints (6)–(8) give limits on flight batches for delivery:

$$\sum_{g=1}^{G} V_{kg} \le 1, k = 1, \ldots, n \tag{6}$$

$$\sum_{g=1}^{G} V_{kg} \le H \times y_k, k = 1, \ldots, n \tag{7}$$

$$\sum_{k=1}^{n} \sum_{g=1}^{G} V_{kg} = \sum_{k=1}^{n} y_k \tag{8}$$

Constraint (6) requires each order is delivered by one vehicle only, and constraint (7) means if there is no order from the customer for the moment, no delivery is required.

Constraint (9) gives limits on flight capacity:

$$\sum_{i=1}^{n} e_i Y_{ik} \le z_g + H\left(1 - V_{kg}\right), k = 1, \ldots, n, g = 1, \ldots, G \tag{9}$$

Constraints (10) and (11) give preparation time for batch B_k:

$$r_k \ge \sum_{k'=1}^{k} \sum_{i=1}^{n} p_i Y_{ik'} - H(1 - y_k), k = 1, \ldots, n \tag{10}$$

$$r_k \le \sum_{k'=1}^{k} \sum_{i=1}^{n} p_i Y_{ik'} - H(1 - y_k), k = 1, \ldots, n \tag{11}$$

Constraints (12)–(14) give limits on departure time u_k:

$$u_k \le H \times y_k, k = 1, \ldots, n \tag{12}$$

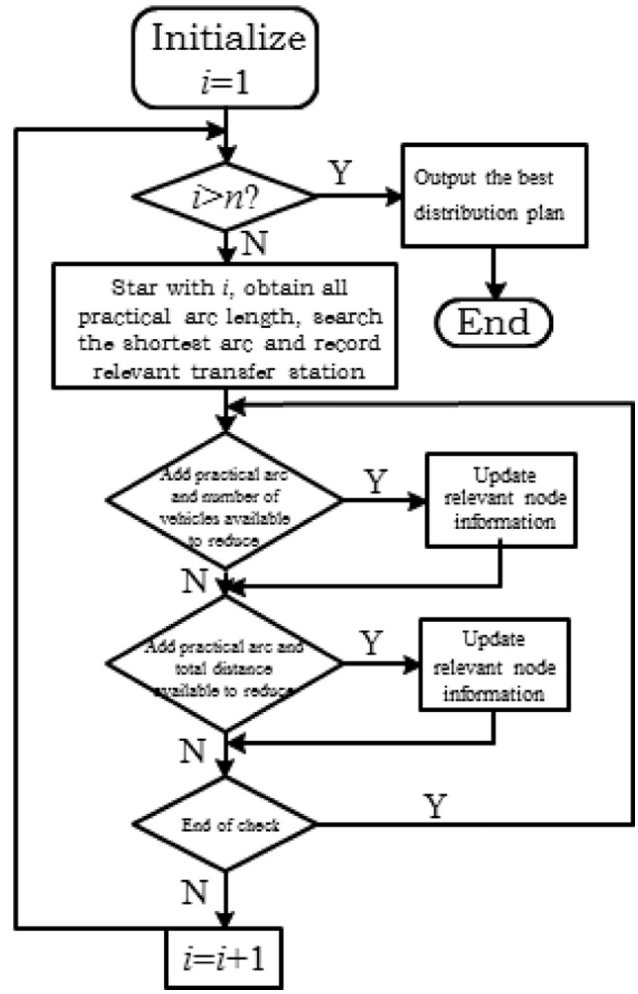

Fig. 2. Best cutting algorithm.

$$\begin{cases} u_1 = r_1 \\ u_k \ge r_k, \end{cases} k = 2, \ldots, n \tag{13}$$

$$\begin{cases} u_{k'} + t_{01}^{g} + t_{10}^{g} - H\left(2 - V_{kg} - V_{k'g}\right) \le u_k \le \\ u_{k'} + t_{01}^{g} + t_{10}^{g} - H\left(2 - V_{kg} - V_{k'g}\right) \\ k = 2, \ldots, n;\ k' = 1, \ldots, k-1;\ g = 1, \ldots, G \end{cases} \tag{14}$$

Constraint (15) gives limitations on order delivery time T_k:

$$\begin{cases} T_k \le H \times y_k, k = 1, \ldots, n \\ u_k + t_{01}^{g} + t_{10}^{g} - H\left(1 - V_{kg}\right) \le u_k \le \\ u_k + t_{01}^{g} + t_{10}^{g} - H\left(1 - V_{kg}\right) \\ k = 2, \ldots, n;\ g = 1, \ldots, G \end{cases} \tag{15}$$

2.2. Coding mode and initial population

2E-VRP can be regarded as Level 1 SDVRP and Level 2 MDVRP coupling problem, decouple available from Level 2 to Level 1. Firstly, the best cutting method is used to obtain the reasonable delivery plan for Level 1 SDVRP problem, so as to determine the number of transfer stations and used as initial solution of QDEMA algorithm. Then, the delivery plan for Level 2 MDVRP problem is solved to obtain total distance and number of flights. Since QDEMA only optimizes Level 2 customer delivery plan, such practice helps to improve the algorithm efficiency. The procedures for best cutting method are shown in Fig. 2 [7].

3. Parallel quantum ant colony algorithm (QACA)

3.1. Ant colony algorithm (ACA)

When solving VRPS problem with ACO algorithm, the more pheromones gathered in a certain direction, the greater the probability that the direction is selected. In QACA, the pheromones of ants gathered in a direction are in qubit coding and the qubit of ants in the direction is updated with quantum rotation gate. The update of pheromones gathered in the direction coverts to the update of qubit probability.

QACA process [7]:

(a) Initialize quantum ant colony $A(t) = (\propto_1^t, \propto_2^t, \ldots, \propto_n^t)$, the number of ants is n, number of qubit is the number of direction m, $\propto_i^t$ $(i = 1, 2, \ldots, n)$ is the ith quantum ant in the tth iteration of population.

$$\propto_i^t = \left[\begin{array}{c|c|c|c} \propto_1^t & \propto_2^t & \ldots & \propto_m^t \\ \beta_1^t & \beta_2^t & \ldots & \beta_m^t \end{array}\right] \tag{16}$$

In order to make all status of algorithm occur at the same probability during initial searching, the value of all $\propto_i, \beta_i (i = 1, 2, 3, \ldots, m)$ in A(0) is determined as $1/\sqrt{2}$.

(b) Assume the value of parameters $\propto$, β, ρ, and the maximum frequency of iteration NMAX, current iteration frequency is $t = 0$, pheromone $\tau_i(0) = 1$.
(c) Create an individual solution for each ant. For ant $k(k = 1, 2, \ldots, n)$, randomly select a direction and put it into the backpack. Then, calculate the probability of the rest of directions to be selected p_i^k according to probability, to select the direction to put into backpack, until the backpack is unavailable for any direction. The probability of a direction to be selected is shown in Formula (4).

$$p_i^k = \begin{cases} \dfrac{[\tau_i(t)]^\alpha [\eta_i(t)]^\beta}{\sum_{S \in J(k)} [\tau_s(t)]^\alpha [\eta_s(t)]^\beta} & S \in J(k) \\ 0 \quad other \end{cases} \tag{17}$$

In Formula (4), $\tau_i(t)$ means the number of pheromones contained in direction i upon the tth iteration, heuristic function $\eta_i(t)$ means the value of unit quality in direction i, namely $\eta_i(t) = v_i/w_i$, α and β means the number of pheromones contained in a direction and the weighting of unit quality value in a direction respectively, $J(k)$ means the set of directions not selected by ant k, the pheromone update equation is as follows:

$$\tau_i(t+1) = (1 - \rho)\tau_i(t) + \Delta\tau_i(k) \tag{18}$$

$$\Delta\tau_i(k) = Q\left|\beta_i^t\right|^2 \tag{19}$$

where, $\Delta\tau_i(k)$ means the number of pheromones left in direction i by ant k, Q is a constant and ρ means the volatility of pheromone, with $(0 \le \rho \le 1)$.

(a) If n ants create their own solution, shift to Step (e), otherwise to Step (c).
(b) Record the best solutions created by m ants in this iteration.
(c) Apply quantum rotation gate rules [11] to update A(t).
(d) If meet with end conditions, namely $t > N$MAX, output the best solution, otherwise $t = t + 1$, shift to Step (c).

3.2. Mapreduce-based quantum ant colony algorithm (MQACA)

For VRPS problem, the time complexity of QACA is $O(NMAX \cdot m \cdot n)$, with computing mainly concentrated in Step 3, where the ants create solutions independently. MQACA algorithm uses Mapreduce to complete generation evolution in population. Map is used to complete the process of independent solution by ants, from which the ant family's index number is used as the key, the ant's optimal solution and the quantum information as the value. This part can be operated in parallel. Reduce is used to express the process to obtain optimal solution and update quantum ant information. The output information is converted into format of Map input and used as input of the next generation of Map function, to enter into the next cycle.

MQACA steps:

Specific steps are as follows:

(a) Initialize the population and create key value pair (k, v), store them in hadoop file system in the form of files, in which k means the index of ant family and v means the ant's solution and quantum information.
(b) Map function receives (k, v), calculates fitness value of each quantum ant and creates intermediate results list ($k1$, $v1$), in which $k1$ means the index of ant family and $v1$ means the solution by individual ant in the family.
(c) Reduce function receives the key value pair list ($k1$, $v1$) created by Map function, updates quantum ant and global pheromone with quantum rotation gate rules, to determine whether reaching to maximum algebra, if yes, output the optimal value, if not, stores the optimal value and output list ($k2$, $v2$), $k2$ means the index of ant family and $v2$ means the ant's solution and quantum information.

Store list ($k2$, $v2$) in hadoop file system and enter into the next cycle.

3.3. Map stage

The main function of Map function is to make each member of ant family create solution independently, output solution by each ant in family to create intermediate results oflist ($k1$, $v1$). Map function is as shown in Function 1.

```
Function Map(k,v)
{
int n; //size of ant population in this map
int m;//total number of directions
inti = 0;
int j = 0;
Calculate the probability of directions to be selected according to Formula (4):
  while(i < n)
  {
Randomly select a direction to put into the backpack;
    while(j < m−1)
    {
  if (total weight of directions selected by ant < backpack capacity)
    {
Continue selecting directions not put into backpack according to the probability;
Calculate the total weight of directions selected by ants
      }
      Else
      break;
      j++;
  }
    i++;
Update solution L by ant i;
    k1 = k;
    v1 = L;
    Emit(k1,v1);
    }
  }
```

Function 1 Map function of MQACA

3.4. Reduce stage

Reduce function receives the key value pair output by Map function, with main function of decomposing the solution and

Table 1
Simulation results.

Example	Branch and Cut		Multi-start		QDEMA	
	Best	Time	Best	Time	Best	Time
E-n22-k4-s6-17	417	4.94	421	16.12	407	4.21
E-n22-k4-s8-14	384	5.76	384	9.23	379	4.58
E-n22-k4-s9-20	481	17.19	472	21.36	458	4.65
E-n22-k4-s10-14	371	3.64	375	5.87	362	6.07
E-n22-k4-s11-12	427	7.87	444	9.15	418	4.26
E-n22-k4-s12-16	392	6.82	403	8.34	384	4.67
E-n33-k4-s1-9	730	17.92	757	22.64	721	7.64
E-n33-k4-s2-13	714	22.52	733	26.14	702	6.46
E-n33-k4-s3-17	707	35.8	754	39.28	698	8.84
E-n33-k4-s4-5	778	33.25	792	34.18	765	8.51
E-n33-k4-s7-25	756	25.87	756	31.24	745	9.30
E-n33-k4-s14-22	779	10.75	824	19.57	758	4.37
E-n51-k5-s2-17	597	15.27	614	25.31	581	7.36
E-n51-k5-s4-46	530	12.87	533	18.74	521	7.26
E-n51-k5-s6-12	554	18.34	564	26.47	541	8.40

value by each member of ant family, to obtain the optimal solution and value. Then, it uses quantum rotation gate rules to update the quantum information of each member of ant family. Update pheromone files according to Formula (5), if meeting with end conditions, output the optimal solution and value, if not, store the key value pair list ($k2$, $v2$) in hadoop file system, in which $k2$ means the index of ant family and $v2$ means the updated solution and quantum ant information. Reduce function is shown in Function 2.

```
Function Reduce(k1, list(v1))
{
int n;//the number of ants in family k1
  for(inti = 0;i < n;i++)
  {
Use quantum rotation gate rules to update quantum ant information A;
Update pheromone according to Formula (5);
    k2 = k1;
    v2 = v1 + A;
    Emit(k2,v2);
}
Write the updated pheromone into pheromone files;
  if(t < NMAX)
  {
Store list(k2,v2)into hadoop file system
  }
  else
Output the optimal value and solution;
}
```

4. Experimental analysis

Experimental conditions settings: the example algorithm selects set2 in Ref. [1,2], with Intel(R)XeonX55502.60 GHz processor, 4GB memory and win 7 system, the simulation software uses matlab2012b. Experimental parameter settings: Set differential evolution population size as $NP = 50$, crossover probability factor as $P_c = 0.8$, frequency of ending iteration of $T_{\max} = 1000$, and maximum algorithm running time of $t_{\max} = 60\,\mathrm{s}$.

Simulation contrast index: The algorithm runs 10 times to get the best target value of delivery plan (*best*), and average running time (*time*). Contrast algorithm uses algorithms Branch and Cut and Multi-start. The simulation results are shown in Table 1 and Fig. 3.

Table 1 gives simulation data of three contrast algorithms on different examples. In the table, the quantity unit of *Best* is kM, and that of *Time* is s. Fig. 3(a) and (b) gives the contrast of three algorithms regarding average target value and running time. Fig. 3(a) shows that, for optimal target value, though with little difference, QDEMA algorithm is always superior to the two contrast algorithms Branch and Cut and Multi-start. While Multi-start is always superior to algorithm Branch and Cut. For running time, Multi-start takes the longest time, followed by algorithm Branch and Cut, while algorithm QDEMA takes the shortest time. Which means, as compared with algorithms Branch and Cut and Multi-start, while Multi-start improves the precision of convergence at the price of computing time, algorithm QDEMA is superior to the two contrast algorithms not only in respect to the precision of convergence, but also in respect to running time. In addition, the algorithm QDEMA has stable average running time.

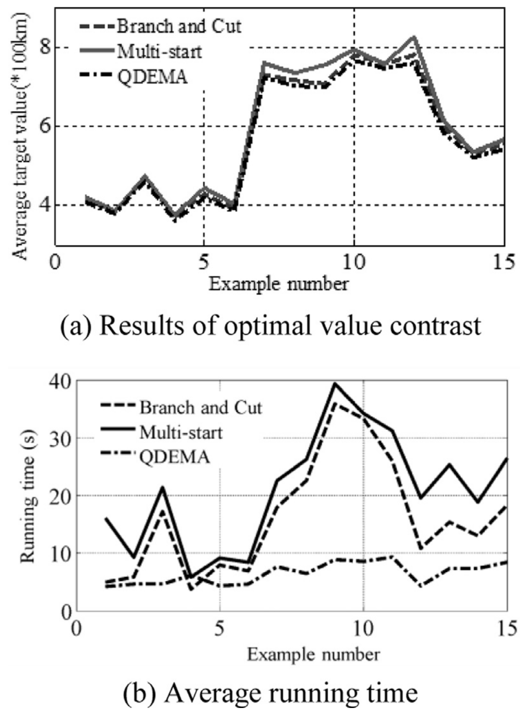

Fig. 3. Results of algorithm contrast.

Some extreme cases, such as Example 4, since the example is much simple and algorithms Branch and Cut and Multi-start have no higher running time, even close to or shorter than running time of algorithm QDEMA. The main reason is that, the example is too simple while algorithm QDEMA still uses the method of population optimization, resulting in longer time. In summary, algorithm QDEMA is suitable for large sized 2E-VRP problems, and in reality, 2E-VRP problems are always large.

5. Conclusions

Considering pilot satisfaction in vehicle routing problem solution (VRP) of autonomous flight, an improved autonomous flight optimization model is proposed in this Paper to improve the service quality of delivery. Mapreduce distribution and step length control parameters are introduced, and the ACO algorithm is improved. The theoretical analysis is given and routing of autonomous flight model mentioned above is designed and optimized through the improved parallel ACO algorithm. The experimental results show that, the algorithm proposed herein can realize expected goals while giving consideration to delivery cost and pilot satisfaction.

Acknowledgement

Tianjin City High School Science & Technology Fund Planning Project, China (No.20140808).

Reference

[1] Y. Zhang, E. Mintzer, E.K. Uhrich, Synthesis and characterization of PEGylated bolaamphiphiles with enhanced retention in liposomes, J. Colloid Interface Sci. 482 (2016) 19–26.

[2] N. Arunkumar, V.S. Balaji, S. Ramesh, S. Natarajan, V.R. Likhita, S. Sundari, Automatic detection of epileptic seizures using independent component analysis algorithm, in: Proceedings of the International Conference on Advances in Engineering, Science and Management, IEEE, 2012, pp. 542–544.

[3] S Ravichandran, W Kiratitanavit, J Kumar, R Nagarajan, Evaluation of performance of polycardanol based flame-retardants in polyolefins, Abstracts of Papers, 243, American chemical society, NWWASHINGTON, D.CUSA, 2012 20036.

[4] X. Du, Y. Zhu, Z. Peng, Y. Cui, Q. Zhang, Z. Shi, Y. Guan, X. Sha, T. Shen, Y. Yang, X. Li, Z. Wang, X. Li, G. Liu, High concentrations of fatty acids and beta-hydroxybutyrate impair the growth hormone-mediated hepatic JAK2-STAT5 pathway in clinically ketotic cows, J. Dairy Sci. 0302 (18) (2018).

[5] Z. Xia, W. Kiratitanavit, S. Yu, P. Facendola, S. Thota, J. Kumar, R. Mosurkal, R. Nagarajan, Modified tannic acid-a bioinspired fire resistant char forming additive for polyamide, Abstracts of Papers, 252, American chemical society, NWWASHINGTON, D.CUSA, 2016 20036.

[6] W. Kiratitanavit, Z. Xia, S. Yu, P. Facendola, R. Ramanathan, R. Mosurkal, R. Nagarajan, Bio-derived phytic acid complex as a char forming additive for polypropylene, Abstracts of Papers, 252, American chemical society, NWWASHINGTON, D.CUSA, 2016 20036.

[7] Y. Zhang, A. Algburi, N. Wang, V. Kholodovych, O.D. Oh, M. Chikindas, K.E. Uhrich, Self-assembled cationic amphiphiles as antimicrobial peptides mimics: role of hydrophobicity, linkage type, and assembly state, nanomedicine: nanotechnology, Biol. Med. 13 (2) (2017) 343–352.

[8] N. Arunkumar, K. Ramkumar, V. Venkatraman, Entropy features for focal EEG and non focal EEG, J. Comput. Sci. (2018), doi:10.1016/j.jocs.2018.02.002.

[9] C. Liu, N. Arunkumar, Risk prediction and evaluation of transnational transmission of financial crisis based on complex network, Cluster Comput. (2018), doi:10.1007/s10586-018-1870-3.

[10] G. Meng, N. Arunkumar, Construction of employee training program evaluation system of three exponential forecast based on sliding window, Cluster Comput. (2018), doi:10.1007/s10586-017-1652-3.

[11] X. Chen, L. Pang, P. Guo, X. Sun, Z. Xue, N. Arunkumar, New upper degree of freedom in transmission system based on wireless G-MIMO communication channel, Cluster Comput. (2017) 1–9, doi:10.1007/s10586-017-1513-0.

[12] R. Hamza, K. Muhammad, N. Arunkumar, G.R. González, Hash based encryption for keyframes of diagnostic hysteroscopy, IEEE Access (2017), doi:10.1109/ACCESS.2017.2762405.

[13] S.L. Fernandes, V.P. Gurupur, N.R. Sunder, N. Arunkumar, S. Kadry, A novel nonintrusive decision support approach for heart rate measurement, Pattern Recognit. Lett. (2017), doi:10.1016/j.patrec.2017.07.002.

Fanlei Yan received the Ph.D. degree in textile engineering from Tianjin Polytechnic University, Tianjin, China, in 2016. He is currently an assistant Professor in Tianjin Polytechnic University, Tianjin. His research interests include computer vision, mobile robots, and UAV mission planning.