

Low Energy Clustering in BAN Based on Fuzzy Simulated Evolutionary Computation

Jie Zhou

Department of Engineering
Macquarie University
Sydney, Australia
jie.zhou8@students.mq.edu.au

School of Electronics Eng.
Beijing University of
Post and Telecommunications
Beijing, China

Eryk Dutkiewicz
Department of Engineering
Macquarie University
Sydney, Australia
eryk.dutkiewicz@mq.edu.au

Ren Ping Liu

Commonwealth Scientific and
Industrial Research
Organisation (CSIRO)
ren.liu@csiro.au

Gengfa Fang

Department of Engineering
Macquarie University
Sydney, Australia
Gengfa.Fang@mq.edu.au

Yuanan Liu

School of Electronics Eng.
Beijing University of
Post and Telecommunications
Beijing, China
yuliu@bupt.edu.cn

ABSTRACT

A low energy clustering method of body area networks based on fuzzy simulated evolutionary computation is proposed in this paper. To reduce communication energy consumption, we also designed a fuzzy controller to dynamically adjust the crossover and mutation probability. Simulations are conducted by using the proposed method, the clustering methods based on the particle swarm optimization and the method based on the quantum evolutionary algorithm. Results show that the energy consumption of the proposed method decreased compared with the other two methods, which means that the proposed method significantly improves the energy efficiency.

Categories and Subject Descriptors

G.2.1 [Mathematics of Computing]: Discrete Mathematics—Combinatorics; F.2.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity—Nonnumerical Algorithms and Problems

General Terms

Algorithms

Keywords

wireless sensor networks, simulated evolutionary computation, fuzzy controller

1. INTRODUCTION

With the continuous development of wireless sensor technology, more and more researchers pay attention to the

high-density wireless sensor networks. High-density wireless sensor networks have a wide range of applications in the battlefield information collection, security systems, hospital theatres, office automation and target localization, etc [1]. With the improvement of intelligence and the decrease of production costs, the number of sensors in high-density wireless sensor network grows exponentially. In some networks, such as hospital theatres, there can be hundreds of sensors working simultaneously. In these applications, how to reduce the communication energy consumption by clustering is a key issue [2].

The most common clustering protocol for wireless sensor networks includes LEACH and HEED. LEACH (Low Energy Adaptive Clustering Hierarchy) [3] is a low energy consumption clustering scheme that significantly improves network lifetime. The LEACH method uses a duty cycle cluster head selection method instead of the fixed cluster heads. The HEED (Hybrid Energy-Efficient Distributed clustering approach) was first introduced in [4]. In HEED, the reserved energy and density of nodes were considered in cluster head selection.

However, both methods focus on maximizing the network lifetime instead of minimizing energy consumption. For large sensor networks, how to select a certain percentage of cluster heads with the minimum energy consumption for data collection is an important issue. In [5], a method based on the simple genetic algorithm for data collection is proposed for minimum data collection energy consumption. However, the simple genetic algorithm easily falls into local optima, and it often gets high energy consumption clustering results.

Based on the above studies, a Low Energy Clustering method of High-density wireless sensor networks based on fuzzy simulated evolutionary computation (FSEC) is proposed in this paper. In order to reduce communication energy consumption, we also designed a fuzzy controller to dynamically adjust the crossover and mutation probability. Simulations are conducted by using the proposed method, the clustering methods based on the particle swarm optimization (PSO)

and the method based on the quantum evolutionary algorithm (QEA). Results show that the energy consumption of the proposed method decreased compared to the other two methods, which means that the proposed method improves energy efficiency.

2. SYSTEM MODEL

This paper studies a class of high-density sensor network, in particular, the Wireless Body Area Networks in hospital theatres, where nodes are randomly distributed in the monitoring area. The sensor nodes within the monitoring area are divided into a number of clusters, and there is a single cluster head node within each cluster.

In the uplink transmission phase, the randomly distributed sensor nodes observe and check the targets within the monitoring area, and then send the observed data to the cluster head node. The cluster head node collects the data from sensor nodes and then uploads the data to the gateway node directly. The gateway node collects the data from the cluster head nodes and transfers the data to the user for further analysis and processing.

In the downlink phase, users allocate monitoring tasks to the cluster head nodes through the gateway node, and then the cluster head nodes assign the tasks to the sensor nodes within the cluster to complete monitoring tasks. Generally, the amount of uplink data is much larger than the downlink data in monitoring tasks.

In order to reduce the energy consumption with a node communication distance limitation, we must develop an efficient clustering scheme for high-density wireless sensor network. The energy consumption of the wireless sensor network consists of the communication energy, the sensing energy and the microprocessor energy. Studies show that the transmission radio energy and receive energy consumption accounted for more than half of the total energy consumption of wireless sensor networks. Meanwhile, the sensing energy consumption and the microprocessor energy consumption are relatively fixed, which are difficult to reduce through optimization. So we focus on how to reduce the communication energy consumption via optimal clustering in high-density sensor networks.

In high-density sensor networks, the transmission radio energy can be shown as:

$$cost_s(k, d) = E_{elec} \cdot k + \varepsilon_{amp} \cdot k \cdot d^n \quad (1)$$

where $cost_s(k, d)$ is the transmission radio energy, d is the distance between two nodes, k is the length of sending bits, ε_{amp} is the power amplification parameter, and E_{elec} is the electronics energy parameter. Depending on the communications environment, the general value of n is between 2-4. The value of n is higher when the communication environment is worse.

The receive energy of k bits data can be shown as:

$$cost_r(k) = E_{elec} \cdot k \quad (2)$$

where $cost_r(k)$ is the receiver dissipated energy for receiving k bits.

3. DESIGN OF THE FUZZY CONTROLLER

Simulated evolutionary computation specifically includes evolution programming, evolution strategies, genetic program-

ming and genetic algorithms. As the genetic algorithm is most widely used among these numerical optimization methods, in this paper, we use fuzzy based genetic algorithm to develop an efficient clustering scheme for high-density wireless sensor networks.

3.1 Adjusting the Algorithm Parameter

Previous studies show that adaptive adjust crossover and mutation probability can significantly improve the diversity of the population and improve the convergence rate. To solve this problem, Srinivas proposed an adaptive genetic algorithm [7]. The adaptive genetic algorithm adaptively adjusts crossover and mutation probabilities with the individual fitness. The adjusting method is simple, however, the accelerating effect is not obvious.

In FSEC, a fuzzy controller is designed to automatically adjust the crossover probability and mutation probability. When the average fitness of the population is too high, the fuzzy controller adjusts crossover probability and mutation probability to a low value to avoid a population diversity decline. Similarly, when the average fitness of the population is low, the fuzzy controller adjusts the crossover probability and mutation probability to a high value to increase the population diversity.

Meanwhile, for excellent individuals, the fuzzy controller adjust its crossover probability and mutation probability to a lower value to keep the individual structure to the next iteration. Similarly, for bad individuals, the fuzzy controller will increase the crossover probability and mutation probability to improve the fitness of the individual. We should notice that for the clustering problem, individuals with low fitness have lower energy consumption, which means the individuals are better.

3.2 The Input and Output of the Fuzzy Controller

In the design of the fuzzy controller that adjusts the crossover probability and mutation probability, first we need to determine the input and output of the fuzzy controller, and design the membership function for the fuzzy input and output.

In order to adjust the of crossover probability and mutation probability, first we need to do normalization for the average fitness of the population \bar{f} , the individual for mutation f_t and the individual with smaller fitness value in the crossover operation f_b . The normalization can be shown as

$$f_c = \frac{f_{\max} - f_b}{f_{\max} - f_{\min}} \quad (3)$$

$$f_m = \frac{f_{\max} - f_t}{f_{\max} - f_{\min}} \quad (4)$$

$$f_a = \frac{f_{\max} - \bar{f}}{f_{\max} - f_{\min}} \quad (5)$$

where f_c is the normalized fitness of the individual with smaller fitness value in the crossover operation, and its value range is $f_c \in [0, 1]$, f_m is the normalized fitness of the individual for mutation, and its value range is $f_m \in [0, 1]$. f_a is the normalized average fitness of the population, and its value range is $f_a \in [0, 1]$. f_{\max} is the fitness of the individual with the largest fitness value in the population, and f_{\min} is the fitness of the individual with the smallest fitness value in the population.

In the high-density sensor network clustering, the individual fitness is smaller when the network communication energy is smaller. For the controller of crossover probability P_c , the fuzzy controller input parameters includes the normalized fitness of the individual with a smaller fitness value in the crossover operation f_c and the normalized average fitness of the population f_a .

When f_c is greater, it indicates that the individual with a smaller fitness value in the crossover operation is better. Similarly, when f_a is greater, it indicates that the average fitness of individuals in the population \bar{f} is smaller. The output value range of the controller of crossover probability is $P_c \in [0.65, 0.95]$.

For the controller of crossover probability P_m , the fuzzy controller input parameters includes the individual in the mutation operation f_m and the normalized average fitness of the population f_a . When f_m is greater, it indicates that the individual in the mutation operation is better. Similarly, when f_a is greater, it indicates that the average fitness of individuals in the population \bar{f} is smaller. The output value range of the controller of crossover probability is $P_m \in [0.05, 0.26]$.

3.3 The Membership Function

For crossover probability P_c , we use the triangular membership function and the Mamdani membership function for input value fuzzification.

For the normalized fitness of the individual with smaller fitness value in the crossover operation f_c , we choose the triangular membership function for fuzzification, and select six fuzzy sets, including very small (FVS), small (FS), relatively small (FRS), relatively big (FRB), big (FB), very big (FVB) for the of input value range $f_c \in [0, 1]$. The first letter F is used to distinguish different kinds of normalized fitness of the individuals.

$$\begin{aligned}
FVS(f_c) &= \begin{cases} -5f_c + 1 & 0 \leq f_c \leq 0.2 \\ 0 & f_c > 0.2 \end{cases} \\
FS(f_c) &= \begin{cases} 5f_c & 0 \leq f_c \leq 0.2 \\ -5f_c + 2 & 0.2 < f_c \leq 0.4 \\ 0 & f_c > 0.4 \end{cases} \\
FRS(f_c) &= \begin{cases} 0 & f_c < 0.2 \\ 5f_c - 1 & 0.2 \leq f_c \leq 0.4 \\ -5f_c + 3 & 0.4 < f_c \leq 0.6 \\ 0 & f_c > 0.6 \end{cases} \\
FRB(f_c) &= \begin{cases} 0 & f_c < 0.4 \\ 5f_c - 2 & 0.4 \leq f_c \leq 0.6 \\ -5f_c + 4 & 0.6 < f_c \leq 0.8 \\ 0 & f_c > 0.8 \end{cases} \\
FB(f_c) &= \begin{cases} 0 & f_c < 0.6 \\ 5f_c - 3 & 0.6 \leq f_c \leq 0.8 \\ -5f_c + 5 & 0.8 < f_c \leq 1 \end{cases} \\
FVB(f_c) &= \begin{cases} 5f_c - 4 & 0.8 \leq f_c \leq 1 \\ 0 & f_c < 0.8 \end{cases}
\end{aligned} \tag{6}$$

For the normalized average fitness of the population f_a , we choose the Mamdani membership function for fuzzification, and select six fuzzy sets, including very small (AVS), small (AS), relatively small (ARS), relatively big (ARB), big (AB), very big (AVB) for the of input value range $f_a \in [0, 1]$. For the Mamdani membership function, if the membership value is great than 1, the membership value equal to 1. The first letter A is used to distinguish different kinds of normal-

ized fitness of the individuals.

$$\begin{aligned}
AVS(f_a) &= 1 - e^{-\frac{0.5}{|10f_a|^{2.5}}} & 0 \leq f_a \leq 1 \\
AS(f_a) &= 1 - e^{-\frac{0.5}{|2-10f_a|^{2.5}}} & 0 \leq f_a \leq 1 \\
ARS(f_a) &= 1 - e^{-\frac{0.5}{|4-10f_a|^{2.5}}} & 0 \leq f_a \leq 1 \\
ARB(f_a) &= 1 - e^{-\frac{0.5}{|6-10f_a|^{2.5}}} & 0 \leq f_a \leq 1 \\
AB(f_a) &= 1 - e^{-\frac{0.5}{|8-10f_a|^{2.5}}} & 0 \leq f_a \leq 1 \\
AVB(f_a) &= 1 - e^{-\frac{0.5}{|10-10f_a|^{2.5}}} & 0 \leq f_a \leq 1
\end{aligned} \tag{7}$$

For the output of the fuzzy controller that adjust P_c , we use a Gaussian fuzzy membership function for the defuzzification. The defuzzification includes seven fuzzy sets, including very small (GVS), small (GS), relatively small (GRS), medium (GM), relatively big (GRB), big (GB), very big (GVB) for the of input value range $P_c \in [0.65, 0.95]$. The first letter G is used to distinguish different kinds of normalized fitness of the individuals.

$$\begin{aligned}
GVS(P_c) &= e^{-\frac{(P_c-0.65)^2}{2 \times 0.02^2}} & 0.65 \leq P_c \leq 0.95 \\
GS(P_c) &= e^{-\frac{(P_c-0.7)^2}{2 \times 0.02^2}} & 0.65 \leq P_c \leq 0.95 \\
GRS(P_c) &= e^{-\frac{(P_c-0.75)^2}{2 \times 0.02^2}} & 0.65 \leq P_c \leq 0.95 \\
GM(P_c) &= e^{-\frac{(P_c-0.8)^2}{2 \times 0.02^2}} & 0.65 \leq P_c \leq 0.95 \\
GRB(P_c) &= e^{-\frac{(P_c-0.85)^2}{2 \times 0.02^2}} & 0.65 \leq P_c \leq 0.95 \\
GB(P_c) &= e^{-\frac{(P_c-0.9)^2}{2 \times 0.02^2}} & 0.65 \leq P_c \leq 0.95 \\
GVB(P_c) &= e^{-\frac{(P_c-0.95)^2}{2 \times 0.02^2}} & 0.65 \leq P_c \leq 0.95
\end{aligned} \tag{8}$$

For mutation probability P_m , we also use the triangular membership function and the Mamdani membership function for input value fuzzification.

$$\begin{aligned}
DVS(f_m) &= \begin{cases} -5f_m + 1 & 0 \leq f_m \leq 0.2 \\ 0 & f_m > 0.2 \end{cases} \\
DS(f_m) &= \begin{cases} 5f_m & 0 \leq f_m \leq 0.2 \\ -5f_m + 2 & 0.2 < f_m \leq 0.4 \\ 0 & f_m > 0.4 \end{cases} \\
DRS(f_m) &= \begin{cases} 0 & f_m < 0.2 \\ 5f_m - 1 & 0.2 \leq f_m \leq 0.4 \\ -5f_m + 3 & 0.4 < f_m \leq 0.6 \\ 0 & f_m > 0.6 \end{cases} \\
DRB(f_m) &= \begin{cases} 0 & f_m < 0.4 \\ 5f_m - 2 & 0.4 \leq f_m \leq 0.6 \\ -5f_m + 4 & 0.6 < f_m \leq 0.8 \\ 0 & f_m > 0.8 \end{cases} \\
DB(f_m) &= \begin{cases} 0 & f_m < 0.6 \\ 5f_m - 3 & 0.6 \leq f_m \leq 0.8 \\ -5f_m + 5 & 0.8 < f_m \leq 1 \end{cases} \\
DVB(f_m) &= \begin{cases} 5f_m - 4 & 0.8 \leq f_m \leq 1 \\ 0 & f_m < 0.8 \end{cases}
\end{aligned} \tag{9}$$

For the normalized fitness of the individual in the mutation operation f_m , we choose the triangular membership function for fuzzification, and select six fuzzy sets, including very small (DVS), small (DS), relatively small (DRS), relatively big (DRB), big (DB), very big (DVB) for the of input value range $f_c \in [0, 1]$. The first letter D is used to distinguish different kinds of normalized fitness of the individuals. The fuzzification process for f_a is the same as the process in the controller for crossover probability P_c .

For the output of the fuzzy controller that adjusts P_m , we use a bell-shaped membership function for the defuzzifica-

tion. The defuzzification includes eight fuzzy sets, including extremely small (OES), very small (OVS), small (OS), relatively small (ORS), medium (OM), relatively big (ORB), big (OB), very big (OVB) for the of input value range $P_m \in [0.05, 0.26]$. The first letter O is used to distinguish different kinds of normalized fitness of the individuals.

$$\begin{aligned}
OES(P_m) &= \frac{1}{1 + \left| \frac{10P_m - 0.5}{0.1} \right|^2} & 0.05 \leq P_m \leq 0.26 \\
OVS(P_m) &= \frac{1}{1 + \left| \frac{10P_m - 0.8}{0.1} \right|^2} & 0.05 \leq P_m \leq 0.26 \\
OS(P_m) &= \frac{1}{1 + \left| \frac{10P_m - 1}{0.1} \right|^2} & 0.05 \leq P_m \leq 0.26 \\
ORS(P_m) &= \frac{1}{1 + \left| \frac{10P_m - 1.4}{0.1} \right|^2} & 0.05 \leq P_m \leq 0.26 \\
OM(P_m) &= \frac{1}{1 + \left| \frac{10P_m - 1.7}{0.1} \right|^2} & 0.05 \leq P_m \leq 0.26 \\
ORB(P_m) &= \frac{1}{1 + \left| \frac{10P_m - 2}{0.1} \right|^2} & 0.05 \leq P_m \leq 0.26 \\
OB(P_m) &= \frac{1}{1 + \left| \frac{10P_m - 2.3}{0.1} \right|^2} & 0.05 \leq P_m \leq 0.26 \\
OVB(P_m) &= \frac{1}{1 + \left| \frac{10P_m - 2.6}{0.1} \right|^2} & 0.05 \leq P_m \leq 0.26
\end{aligned} \tag{10}$$

3.4 The Fuzzy Rules and Fuzzy Implication

The previous experience for the parameter adjustment can be summarized as the following fuzzy rules [8]:

- (I) "IF f_c is bigger, THEN the crossover probability P_c is smaller"
- (II) "IF f_m is bigger, THEN the crossover probability P_m is smaller"
- (III) "IF f_a is bigger, THEN the crossover probability P_c is bigger"

(IV) "IF f_a is bigger, THEN the crossover probability P_m is bigger"
According to the fuzzy rule (I) and (III), we designed a double input single output (DISO) fuzzy rule table, which is shown in Table 1. As the number of the input fuzzy sets for both f_c and f_a is 6, the number of the fuzzy rules in Table 1 increases up to 36 for the output P_c .

Table 1: fuzzy rules for adjust the crossover probability P_c

f_a	f_c					
	FVB	FB	FRB	FRS	FS	FVS
AVS	1GVS	2GVS	3GS	4GRS	5GM	6GRB
AS	7GVS	8GS	9GRS	10GM	11GRB	12GB
ARS	13GS	14GS	15GRS	16GM	17GRB	18GB
ARB	19GS	20GRS	21GM	22GRB	23GB	24GB
AB	25GS	26GRS	27GM	28GRB	29GB	30GVB
AVB	31GRS	32GM	33GRB	34GB	35GVB	36GVB

The numbers in the table are used to indicate the order of rules. To assess the influence of each rule, we use the fuzzy implication $R_i (i = 1, 2, \dots, 36)$ to express i -th rule for adjust the crossover probability P_c . Once all the contributions of the 36 implications are determined, we can aggregate these 36 fuzzy implications into one output fuzzy implication R , namely

$$R = R_1 \cup R_2 \cup \dots \cup R_{36} = \bigcup_{i=1}^{36} R_i \tag{11}$$

Each fuzzy implication can be expressed by the inference "IF f_c is A and f_a is B , THEN P_c is U " that is activated by the input value f_c and f_a , namely $A(f_c) \wedge B(f_a) \rightarrow U(P_c)$.

As we apply the minimum (Mamdani) implication for each $A(f_c) \wedge B(f_a) \rightarrow U(P_c)$, the text form and the corresponding activation degrees of the consequent parts of these 36 rules become:

- (1) IF f_c is FVB and f_a is AVS, THEN P_c is GVS, $R_1(f_c, f_a, P_c) = FVB(f_c) \wedge AVS(f_a) \wedge GVS(P_c)$
- (2) IF f_c is FB and f_a is AVS, THEN P_c is GVS, $R_2(f_c, f_a, P_c) = FB(f_c) \wedge AVS(f_a) \wedge GVS(P_c)$
- ...
- (36) IF f_c is FVS and f_a is AVB, THEN P_c is GVB, $R_{36}(f_c, f_a, P_c) = FVS(f_c) \wedge AVB(f_a) \wedge GVB(P_c)$

According to the fuzzy rules (II) and (IV), we designed a double input single output (DISO) fuzzy rule table, which can be shown in Table 2. As the number of input fuzzy set for both f_m and f_a are 6, the number of fuzzy rules in Table 2 increases up to 36 for the output P_m .

Table 2: fuzzy rules for adjust the mutation probability P_m

f_a	f_m					
	DVB	DB	DRB	DRS	DS	DVS
AVS	1OES	2OES	3OVS	4OS	5ORS	6OM
AS	7OES	8OVS	9OS	10ORS	11OM	12ORB
ARS	13OVS	14OS	15ORS	16OM	17ORB	18OB
ARB	19OVS	20OS	21OM	22ORB	23ORB	24OB
AB	25OS	26ORS	27OM	28ORB	29OB	30OVB
AVB	31ORS	32OM	33ORB	34OB	35OVB	36OVB

To assess the influence of each rule $G_i (i = 1, 2, \dots, 36)$, we use the fuzzy implication $G_i (i = 1, 2, \dots, 36)$ to express the i -th rule for adjusting the mutation probability P_m . Once all contributions of 36 implications are determined, we can aggregate these 36 fuzzy implications into one output fuzzy implication G , namely

$$G = G_1 \cup G_2 \cup \dots \cup G_{36} = \bigcup_{i=1}^{36} G_i \tag{12}$$

Each fuzzy implication can be expressed by the inference "IF f_m is A and f_a is B , THEN P_m is U " that is activated by the input value f_m and f_a , namely $A(f_m) \wedge B(f_a) \rightarrow U(P_m)$. As we apply the minimum (Mamdani) implication for each $A(f_m) \wedge B(f_a) \rightarrow U(P_m)$, the text form and the corresponding activation degrees of the consequent parts of these 36 rules become:

- (1) IF f_m is DVB and f_a is AVS, THEN P_m is OES, $G_1(f_m, f_a, P_m) = DVB(f_m) \wedge AVS(f_a) \wedge OES(P_m)$
- (2) IF f_m is DB and f_a is AVS, THEN P_m is OES, $G_2(f_m, f_a, P_m) = DB(f_m) \wedge AVS(f_a) \wedge OES(P_m)$
- ...
- (36) IF f_m is DVS and f_a is AVB, THEN P_m is OVB, $G_{36}(f_m, f_a, P_m) = DVS(f_m) \wedge AVB(f_a) \wedge OVB(P_m)$

3.5 Defuzzification

According to Zadeh, we use the max-min aggregation operation to get the output fuzzy set of P_c , which can be shown as

$$U^*(P_c) = (\vec{E}^*)^T \circ \bigcup_{i=1}^{36} R_i = \bigcup_{i=1}^{36} U_i(P_c) \tag{13}$$

where U^* is the membership function of the resultant output fuzzy set of P_c , the operator \circ means the minimum implication for the corresponding elements, and R_i is defined

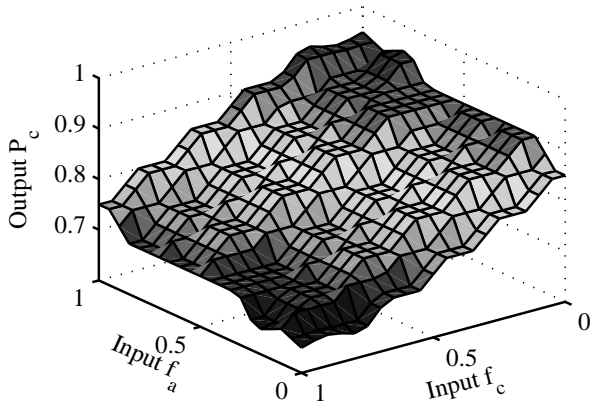


Figure 1: The output P_c

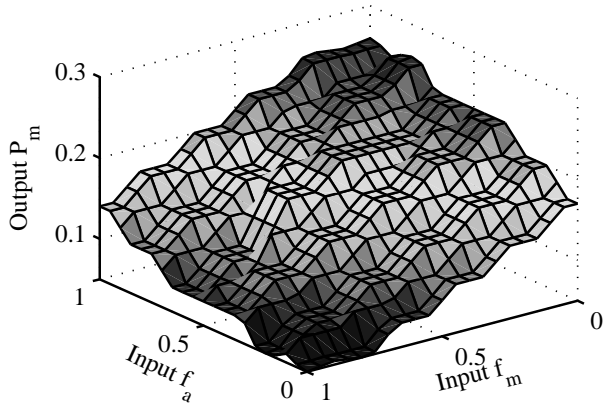


Figure 2: The output P_m

in Section 3.4. $(\vec{E}^*)^T$ is the minor premise that is straightened according to the line. The fuzzy interpretation of the i -th rule can be presented as

$$(1) U_1(P_c) = (FVB(f_{c0}) \wedge AVS(f_{a0})) \circ R_1(P_c)$$

$$(2) U_2(P_c) = (FB(f_{c0}) \wedge AVS(f_{a0})) \circ R_2(P_c)$$

...

$$(36) U_{36}(P_c) = (FVS(f_{c0}) \wedge AVB(f_{a0})) \circ R_{36}(P_c)$$

In the defuzzification, we use the Mean of Maximum (MOM) method to calculate the control output of P_c , and the result can be shown as Fig.1.

From the figure we can see that in the input value range $f_c \in [0, 1]$ and $f_a \in [0, 1]$, the output P_c meets the previous experience (I) and (III) for the parameter adjustment, which are “IF f_c is bigger, THEN the crossover probability P_c is smaller” and “IF f_a is bigger, THEN the crossover probability P_c is bigger”. Similarly, the output fuzzy set of P_m can be shown as

$$U^*(P_m) = (\vec{H}^*)^T \circ \bigcup_{i=1}^{36} G_i = \bigcup_{i=1}^{36} U_i(P_m) \quad (14)$$

where U^* is the membership function of the resultant output fuzzy set of P_m , and G_i is defined in Section 3.4. $(\vec{H}^*)^T$ is the minor premise that it is straightened according to the line. The fuzzy interpretation of the i -th rule can be presented as

$$(1) U_1(P_m) = (DVB(f_{m0}) \wedge AVS(f_{a0})) \circ G_1(P_m)$$

$$(2) U_2(P_m) = (DB(f_{m0}) \wedge AVS(f_{a0})) \circ G_2(P_m)$$

...

$$(36) U_{36}(P_m) = (DVS(f_{m0}) \wedge AVB(f_{a0})) \circ G_{36}(P_m)$$

In the defuzzification, we also use Mean of Maximum (MOM) method to calculate the control output of P_m , and the result can be shown as Fig.2.

From the figure we can see that in the input value range $f_m \in [0, 1]$ and $f_a \in [0, 1]$, the output P_m meets the previous experience (II) and (IV) for the parameter adjustment, which are “IF f_m is bigger, THEN the mutation probability P_m is smaller” and “IF f_a is bigger, THEN the mutation probability P_m is bigger”.

3.6 Computational Complexity

For the computational complexity analysis, the two fuzzy controllers are equivalent to two dual input single output functions $g_1(f_c, f_a) = P_c$ and $g_2(f_m, f_a) = P_m$. The output P_c is calculated in advance and stored in a table for each input (f_c, f_a) that rounds to a certain precision. Similarly, the output P_m is calculated in advance and stored in a table for each input (f_m, f_a) that rounds to a certain precision. In this way, the computational complexity of both controllers is $O(1)$.

4. LOW ENERGY CLUSTERING BASED ON FUZZY SIMULATED EVOLUTIONARY COMPUTATION

In high-density WSNs, the traditional clustering methods lack an overall consideration, so there are some sensors far from the cluster head, which result in energy waste. The clustering method proposed in this paper establishes a global unity mechanism for minimizing communication energy consumption, which considers the position of the cluster heads and the sensor nodes. The main steps of our methods are individual encoding and initial population generation, design fitness function, selection, crossover and mutation, adaptive adjustment of parameters based on the fuzzy controller, etc.

4.1 Population Encoding and Initialization

In FSEC, the encoded solutions are represented by chromosomes. First, we conduct a natural number coding for all the nodes within the region with natural number 1 to L . In this way, each individual can be encoded as a vector of binary numbers, with the same length as the number of nodes L . Each bit of the individual is composed of a Boolean variable stating whether the corresponding sensor node is selected as the cluster head node or not. The “1” represents the node is selected as a cluster head node in the corresponding position, and “0” represents the node is selected as a sensor node in the corresponding position. Each sensor node only clusters with the nearest cluster head. For example, if there

are 8 nodes in the region and No.3, No.5 and No.7 nodes are selected as cluster head nodes, the individuals can be represented as “00101010”.

We use a population with a fixed number of individuals. The population can be shown as

$$P = \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,L-1} & e_{1,L} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,L-1} & e_{2,L} \\ \vdots & & & & \vdots \\ e_{N-1,1} & e_{N-1,2} & \cdots & e_{N-1,L-1} & e_{N-1,L} \\ e_{N,1} & e_{N,2} & \cdots & e_{N,L-1} & e_{N,L} \end{bmatrix} \quad (15)$$

$$= \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_{N-1} \\ E_N \end{bmatrix} \quad (e_{n,l} \in \{0, 1\})$$

where L is the number of nodes in sensor networks, N is the number of individuals in the population, M is the number of cluster heads. $e_{n,l}=1$ represents the n -th node of l -th individual is the cluster head node, and $e_{n,l}=0$ otherwise.

We assume that there is a fixed number of cluster head nodes in sensor networks, so the constraint can be shown as

$$\sum_{l=1}^L e_{n,l} = M \quad (n \in \{1, 2, \dots, N\}) \quad (16)$$

where M is the number of cluster head nodes.

4.2 Fitness Function

Each individual in the population is awarded a score depending on the communication energy consumption. The fitness function is defined as

$$Fit(E) = \sum_{l=1}^L (cost_s + cost_r) \quad (17)$$

where $E_n = [e_{n,1} \ e_{n,2} \ \cdots \ e_{n,L-1} \ e_{n,L}]$ is the fitness of the n -th individual of the population, and the fitness value equals to the communication energy consumption of the n -th clustering scheme. By this definition, the individuals with smaller fitness in the population are the better ones and are more likely to be parents in the next generation.

4.3 Selection

We use the roulette wheel proportionate selection strategy for the selection operator. The selection operator selects an individual from the current population for the next population with the probability inversely proportional to its fitness value, which can be shown as

$$P_{SELECT}(E_n) = \frac{\frac{1}{Fit(E_n)}}{\sum_{n=1}^N \frac{1}{Fit(E_n)}} \quad (18)$$

In this way, the individuals that have a lower communication energy consumption will have a higher probability to be selected as parents.

4.4 Crossover

In the crossover operation, two new offspring individuals are generated from each pair of selected parent individuals. In order to keep the number of cluster head nodes fixed, we designed a crossover operator based on Boolean operation.

Firstly, we apply the logical AND operation to each pair of selected parent individuals, and obtain an intermediate binary vector E' . The length of binary vector E' is equal to the number of sensor nodes L . For example, when parent individuals are $E_1 = [00101010]$ and $E_2 = [00100101]$, the intermediate binary vector is $E' = [00100000]$.

Secondly, we apply the logical XOR operation to each pair of selected parent individuals, and obtain another intermediate binary vector E'' . The length of binary vector E'' is equal to the number of sensor nodes L . For example, when the pair of selected parent individuals are $E_1 = [00101010]$ and $E_2 = [00100101]$, the other intermediate binary vector is $E'' = [00001111]$.

Finally, we average random assign the “1” in individual E' to the corresponding position of E'' to generate two new offspring individuals. The average means the number of “1” assigned to each individual is equal, and the random means the possibilities of the distribution being equal. For example, when $E' = [00100000]$ and $E'' = [00001111]$, the possible two new offspring individuals can be $E_1^{new} = [00101100]$ and $E_2^{new} = [00100011]$.

4.5 Mutation Operation

In order to add some diversity to the population and keep the number of cluster head nodes fixed, some random mutations are applied to the individuals in the population. The mutation operation is done by randomly exchanging the position of “1” and “0” in an individual with a certain probability. For example, the individual $E = [00101100]$ can be mutated to $E = [00101001]$, which means the “1” in the 6th bit and the “0” in 8th bit exchanged.

4.6 Fuzzy Adjust the Algorithm Parameters

In the traditional genetic algorithm, crossover and mutation probabilities of selection have a great influence on the performance of the algorithm. We use the adjust method presented in Section 3.

5. SIMULATION RESULTS AND DISCUSSION

In this section, the proposed FSEC is tested with different sensor nodes and cluster head proportions to investigate the low energy clustering problem in high-density WSNs. Simulation experiments were conducted to verify the communication energy reduction of the proposed clustering method. In the simulations, the monitoring area is $100\text{m} \times 100\text{m}$, and nodes in high-density sensor network are randomly distribute within the simulating hospital floor area. The gateway node is located at (50, 50). As the communication energy consumption of uplink is much larger than downlink in most WSNs, we only consider the uplink communication energy consumption. In the simulations, we set $k = 1\text{Mbps}$, $n = 3$, $E_{elec} = 50\text{nJ/bit}$, $\varepsilon_{amp} = 100\text{pJ/bit/m}^2$ [6].

Comparisons are made with PSO and QEA by only considering the best solution in each iteration. In QEA, we use the same lookup table as [9]. In PSO, the cognitive and social parameters are set to 2, and the maximum velocity is set to 6.

Fig.3 and Fig.4 show the communication energy consumption of the high-density WSN using FSEC, the PSO and QEA with 300 and 400 nodes respectively, where the energy consumption is obtained directly from the formula (1) and (2). The cluster head proportion is 10%. The results are the average over 10 runs with different random nodes distributions. According to these comparisons, the proposed FSEC

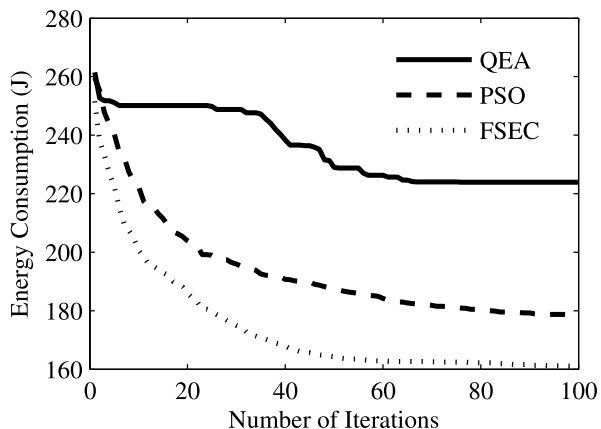


Figure 3: The communication energy consumption by iteration with 300 nodes and cluster head proportion of 10%

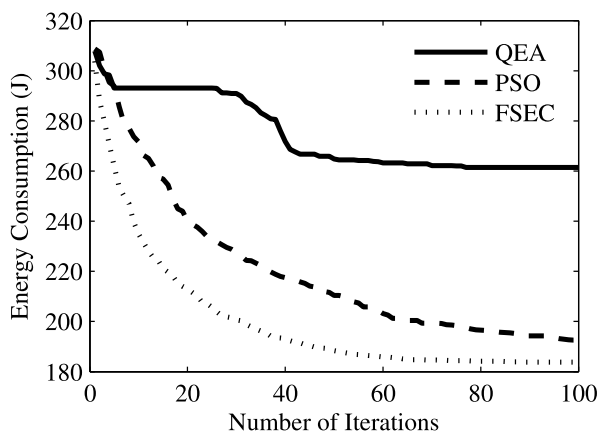


Figure 4: The communication energy consumption by iteration with 400 nodes and cluster head proportion of 10%

has a lower communication energy consumption than that of PSO and QEA methods with different numbers of nodes after 100 iterations.

As it may be observed in these figures, in the beginning, communication energy consumption of all the algorithms decreased. However, after only few iterations the QEA fall into evolutionary stagnation. From the figures, we can also see that PSO is much slower in convergence compared with FSEC. More specifically, the communication energy consumption of FSEC is 9.78% and 4.57% less than that of PSO, and 28.08% and 29.69% less than that of QEA for 300 and 400 nodes respectively.

6. CONCLUSION

In this paper, we propose a fuzzy simulated evolutionary computation clustering method to reduce communication energy consumption for body area networks. In order to reduce communication energy consumption, we also designed a fuzzy controller to dynamically adjust the crossover and mutation probability. Simulation results show that the communication energy consumption of the proposed method decreased compared to the other two methods, which means that the proposed method significantly improves the energy efficiency.

7. ACKNOWLEDGMENTS

Jie Zhou is a recipient of a Macquarie University Research Excellence Scholarship and is under a cotutelle PhD arrangement between Macquarie University and Beijing University of Posts and Telecommunications. This work was supported in part by the National Natural Science Foundation of China (No.61170275), the National Science and Technology Major Project (No.2012ZX03001001-002), The research project of Guangdong (No.2011B090400433).

8. REFERENCES

- [1] M. Li, Z. Li, and A. V. Vasilakos. A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues. *Proceedings of the IEEE*, 101(12):2538 – 2557, 2013.
- [2] O. Demigha, W. K. Hidouci, and T. Ahmed. On energy efficiency in collaborative target tracking in wireless sensor network: A review. *IEEE Communications Surveys & Tutorials*, 15(3):1210 – 1222, 2013.
- [3] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, 2002.
- [4] O. Younis and F. Sonia HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, 2004.
- [5] X. Li. Research on clustered-network in wireless sensor networks. *Tianjin University, China*, PhD thesis, 2006.
- [6] S. Lindsey and C. S. Raghavendra. Pegasus: Power-efficient gathering in sensor information systems. In *Proceedings of the Aerospace Conference*, pages 3-1125–3-1130. IEEE, March 2002.
- [7] M. Srinivas and L. M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 24(4):656–667, 1994.
- [8] Q. Li, D. L. Zhen and Y. Tang A new kind of fuzzy genetic algorithm. *Journal of University of Science and Technology Beijing*, 23(1):85–89, February 2001.
- [9] H. W, C. A, and B. H. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 1–10. IEEE, January 2000.