Sensors **2007**, *7*, 2693-2722

sensors
ISSN 1424-8220
© 2007 by MDPI
www.mdpi.org/sensors

Full Research Paper

Hierarchical Wireless Multimedia Sensor Networks for Collaborative Hybrid Semi-Supervised Classifier Learning

Xue Wang *, Sheng Wang, Daowei Bi and Liang Ding

State Key Laboratory of Precision Measurement Technology and Instrument, Tsinghua University,
Beijing 100084, P. R. China
E-mails: wangxue@mail.tsinghua.edu.cn, {wang_sheng00,bdw02,dintliang03}@mails.tsinghua.edu.cn,

* Author to whom correspondence should be addressed.

Received: 24 October 2007 / Accepted: 8 November 2007 / Published: 13 November 2007

Abstract: Wireless multimedia sensor networks (WMSN) have recently emerged as one of the most important technologies, driven by the powerful multimedia signal acquisition and processing abilities. Target classification is an important research issue addressed in WMSN, which has strict requirement in robustness, quickness and accuracy. This paper proposes a collaborative semi-supervised classifier learning algorithm to achieve durative online learning for support vector machine (SVM) based robust target classification. The proposed algorithm incrementally carries out the semi-supervised classifier learning process in hierarchical WMSN, with the collaboration of multiple sensor nodes in a hybrid computing paradigm. For decreasing the energy consumption and improving the performance, some metrics are introduced to evaluate the effectiveness of the samples in specific sensor nodes, and a sensor node selection strategy is also proposed to reduce the impact of inevitable missing detection and false detection. With the ant optimization routing, the learning process is implemented with the selected sensor nodes, which can decrease the energy consumption. Experimental results demonstrate that the collaborative hybrid semi-supervised classifier learning algorithm can effectively implement target classification in hierarchical WMSN. It has outstanding performance in terms of energy efficiency and time cost, which verifies the effectiveness of the sensor nodes selection and ant optimization routing.

Keywords: Wireless sensor networks, target classification, collaborative learning, support vector machine, ant colony optimization.

1. Introduction

With the development of inexpensive multimedia hardware, such as micro-cameras and microphones, wireless multimedia sensor networks (WMSN) have recently emerged as an important technology, which has outstanding performance in multimedia signal acquisition and processing. Compared to a wireless sensor network (WSN), WMSNs can not only enhance existing sensor network applications such as target tracking, classification, home automation, and environmental monitoring, but also enable several new applications, such as multimedia surveillance sensor networks, advanced health care delivery, industrial process control, and so on [1]. The content-rich vision-based information brings more effective way for target tracking and classification, but it also requires efficient distributed processing because the energy and network resources are strictly constrained in WMSNs. Target classification is a big challenge addressed in WMSN. The specific requirements of WMSNs, such as transmission of large amounts of data, various noises and time-varying samples, determine that the energy efficient, robust and online classifier learning algorithm is highly desired.

Support vector machine (SVM) is a well known classification tool, which has been widely used in WSNs [2-4]. However, SVM classifier learning calls applies to solving a quadratic programming problem [5], which is computation expensive and cannot be afforded by single sensor node with limited computing ability and energy. Furthermore, traditional classifier learning is a kind of centralized learning strategy, which also needs to acquire samples from all sensor nodes. The large amount of data transmission will consume much energy, which is impractical for a highly constrained WMSN. Recently, various incremental learning methods were proposed [4-6], which implies that the learning process can be progressively carried out with the collaboration of multiple sensor nodes. Although incremental learning is suitable for target classification in WMSNs, with the consideration of missing and false detection, the incremental learning should be collaboratively implemented according to the contribution of sensor nodes. This requires a new architecture to increase the system scalability with collaborative in-network processing by reducing the energy consumption and gathering the samples from a proper set of sensor nodes. Moreover, the SVM principle determines that the learning process needs a set of independent, identically distributed labeled samples. However, obtaining classification labels is an expensive and infeasible task for unattended WMSN, which determines that the traditional SVM classifier learning algorithm cannot be achieved online in WMSN.

This paper proposes a collaborative semi-supervised classifier learning algorithm for target classification with hybrid computing paradigm in hierarchical WMSN, which is enhanced by the ant optimization routing. The proposed algorithm progressively implements semi-supervised learning process in hybrid computing paradigm with the collaboration of a proper set of sensor nodes, which is so-called collaborative hybrid (CH) learning. The semi-supervised learning is based on the transductive support vector machine (TSVM) algorithm [7]. It can take as much advantage of unlabeled samples as possible, according to the guidance of labeled samples. Thus, the decision function can be effectively constructed based on the labeled and unlabeled samples. Actually, the semi-supervised learning makes it possible to achieve online learning with unlabeled samples extracted from all sensor nodes. Certainly, the proposed learning algorithm also takes the advantages of incremental learning to decrease the energy consumption in data transmission. During the incremental learning process, two metrics, effectiveness metric and access probability are introduced to evaluate the effectiveness and necessity of the samples in specific sensor nodes. According to the evaluation of historical contribution of sensor nodes, the incremental semi-supervised learning is implemented with the collaboration of some purposefully selected sensor nodes. By using the sensor nodes selection strategy, the imprecise sensor nodes will be ignored. Thus the impact of missing detection and false detection can be largely reduced. For further decreasing the energy consumption, the ant optimization routing is adopted to arrange the information transmission of collaborative hybrid learning paradigm in hierarchical WMSN.

Usually, a WMSN is always built on a hierarchical architecture, which comprises several clusters. Each cluster consists of several sensor nodes and a cluster head. In the collaborative hybrid learning paradigm, the in-network signal processing in each cluster is not constructed by client/server paradigm as usual, because the traditional client/server paradigm will greatly deteriorate the quality of service (QoS) of the network. Instead, the progressive distributed computing paradigm [8] is used for the in-network signal processing in each cluster, and a peer-to-peer (P2P) computing paradigm is used for the further signal processing between all cluster heads. The CH learning paradigm has the advantages of collaboration and parallelism. Thus, the CH learning paradigm can reduce the energy consumption and network congestion of data transmission. For investigating the performance of the proposed collaborative hybrid semi-supervised classifier learning algorithm, four learning paradigms, centralized client/server (C-C/S) learning paradigm, distributed client/server (D-C/S) learning paradigm, mobile agent (MA) learning paradigm and collaborative hybrid learning paradigm, are introduced respectively. Then the classification accuracy, energy consumption and time cost of four learning paradigms are compared in real world experiment.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction of background subtraction [9] based target detection and 2-D integer lifting wavelet transform (ILWT) [10] based feature extraction. Section 3 presents the principle of TSVM based target classification in WMSN. Then Section 4 introduces the details of the four different computing paradigms for classifier learning in WMSN and proposes the ant optimization routing method. Section 5 illustrates the experimental results to present the effectiveness of the collaborative semi-supervised classifier learning algorithm, and compares the classification accuracy, energy consumption and time cost of four computing paradigms. And finally, Section 6 summarizes our work.

2. Preliminaries

Target classification is a main application in WMSNs. An autonomous target classification system always consists of three operations: target detection, feature extraction and target classification. The limited bandwidth and energy resources require a computing paradigm for collaborative, distributed and resource-constrained processing that allow for filtering and extraction of effective information at each sensor node. This may decrease the energy consumption of the WMSN, and improve its lifetime accordingly. Thus, during target classification, target detection and feature extraction should be carried out in each sensor node, which can be considered as the preprocessing operations to acquire the samples for classifier learning. Because the computing ability of sensor nodes is strictly constrained, the target detection and feature extraction algorithms should be simple and easy-to-perform.

Background subtraction is a simple algorithm for extracting the minimum boundary rectangle results of targets, which models background scenes statistically to detect foreground objects. The applications in [9] verify that background subtraction is a simple but efficient method for target detection. And it is also successfully applied in our previous work [8]. Please refer to Appendix A for details of background subtraction algorithm.

With the target detection, the minimum boundary rectangle results are acquired, which contain the appearance of target. However, because the data amount of image information is too large for a WMSN, effective feature extraction is highly desired, which refers to transform a data space into a feature space. For WMSN, image compression can be considered as an effective technique for feature extraction, where discrete wavelet transform (DWT) is well established. Appendix B gives the details of DWT algorithm. For images, 2-D wavelet transform leads to a decomposition in approximation and details, where the approximation at a desired level can be considered as the compressed image. To simplify the computation, a lifting scheme (LS) is introduced in DWT [10]. In LS, the input data are split into two signals with evenly and oddly indexed samples. Then two signals are alternately convolved with a primal lifting filter and a dual lifting filter. Here, the primal and dual lifting filters are simple and short FIR filters. Then after some iterations, the *n*th level approximation of input signal is just the desired result. For conquering the disadvantage of lossy image compression, a further improvement is achieved by combining integer wavelet transform with integer LS. In practice, integer lifting wavelet transform is successfully used in embedded image compression in WMSN [11]. Then the compressed image can be used as the compact representation for target classification.

3. TSVM Based Target Classification

3.1 The principle of classical SVM

After feature extraction, a SVM based classifier learning algorithm is adopted in the WMSN. Consider the problem of separating the set of sample vectors belonging to two separate classes in some feature space. Given one set of learning samples,

$$(x_1, y_1), \ldots, (x_l, y_l), \qquad x_i \in R^n, y_i \in \{-1, +1\} \tag{1}$$

where the vectors x_i are the extracted target samples, and y_i are the classification labels. SVM aims to separate the vectors with a hyperplane

$$(w \cdot x) + b = 0 \tag{2}$$

i.e.

$$\begin{cases} (w \cdot x_i) + b \geq 1 & y_i = 1 \\ (w \cdot x_i) + b \leq -1 & y_i = -1 \end{cases} \Leftrightarrow y_i \cdot ((w \cdot x_i) + b) \geq 1 \quad (i = 1, 2, \ldots, l) \tag{3}$$

where w and b are the parameters for positioning the hyperplane.

The hyperplane with the largest margin is the desired one, where the distance between the closest vectors to the hyperplane is given by

$$p(w, b) = \min_{\{x_i | y_i = 1\}} \frac{w \cdot x_i + b}{\|w\|} - \max_{\{x_i | y_i = -1\}} \frac{w \cdot x_i + b}{\|w\|} = \frac{2}{\|w\|} \tag{4}$$

Here, $\|w\|$ denotes the arbitrary norm of w, which is selected as 2-norm in this paper. Hence, the desired hyperplane is the one that minimizes

$$\min\left(\Phi(w)\right)=\min\left(\frac{1}{2}\|w\|^2\right) \tag{5}$$

subject to the constraints of Eq. (3), where $\Phi(w)$ is the measure of distance. With the optimal hyperplane parameter (w_0,b_0), the decision function can be written as

$$f(x)=w_0\cdot x+b_0 \tag{6}$$

Then the test data can be labeled with

$$\text{label}(x)=\text{sgn}\left(f(x)\right)=\text{sgn}\left(w_0\cdot x+b_0\right) \tag{7}$$

Learning vectors that satisfy $y_i\left[\left(w_0\cdot x_i\right)+b_0\right]=1$ are termed support vectors. In the case of linearly non-separable learning data, by introducing slack variables ξ_i, Eq. (5) can be rewritten as

$$\min\left(\frac{1}{2}\|w\|^2+C\sum_i\xi_i\right),\ \text{ s.t. } y_\text{i}\left(w\cdot x_i+b\right)\geq 1-\xi_i,\ \xi_i\geq 0 \tag{8}$$

where C is the parameter for adjusting the cost of constraint violation. In this paper, C is set to 8, which is determined by the comparison between the classification results of different values of C.

3.2 TSVM training with unlabeled samples

However, in WMSNs, obtaining the classification labels is infeasible and expensive, because WMSNs are a kind of unattended system. Normally, large quantities of unlabeled samples are readily available in WMSN, and unlabeled samples are significantly easier to obtain than labeled ones. Thus, the classifiers learning process should take as much advantage of unlabeled samples as possible, which is also the motivation of semi-supervised learning algorithm. TSVM [7] is a useful semi-supervised learning algorithm for SVM based classifier learning. During learning, the decision function is constructed based on all the available data. It is beneficial to incorporate part or all of samples in the classifier learning processes. In TSVM, the learning process often uses a small number of labeled samples and a large number of unlabeled samples, which provide enough information about the distribution of whole sample space. Learning process of TSVM can be described as follows:

Given a set of independent, identically distributed labeled samples

$$\left(x_1,y_1\right),\ldots,\left(x_n,y_n\right),\ x_i\in R^m,\ y_i\in\{-1,+1\} \tag{9}$$

and another set of unlabeled samples,

$$x_1^*,x_2^*,x_3^*,\ldots,x_k^* \tag{10}$$

The learning process of TSVM can be formulated as the following optimization problem

$$\begin{aligned}&\text{Over }\left(y_1^*,y_2^*,\ldots,y_k^*,w,b,\xi_1,\ldots,\xi_n,\xi_1^*,\ldots\xi_k^*\right)\\&\min\left(\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i+C^*\sum_{j=1}^{k}\xi_j^*\right)\\&\text{s.t. } y_i\left(w\cdot x_i+b\right)\geq 1-\xi_i,\ i=1,2,\ldots,n,\ \xi_i\geq 0\\&\qquad y_j^*\left(w\cdot x_j^*+b\right)\geq 1-\xi_j^*, j=1,2,\ldots,k,\ \xi_j^*\geq 0\end{aligned} \tag{11}$$

where $y_1^*, y_2^*, \ldots, y_k^*$ are the predicted labels of the corresponding unlabeled samples, $\xi_1^*, \ldots \xi_k^*$ are the corresponding slack variables, C and C^* are the effect factor of label and unlabeled samples.

With the TSVM algorithm, the classifier learning process can be outlined as following pseudo-code:

```
Procedure  TSVM Algorithm
  1. Initialization
    Specify the parameter $C$ and $C^*$
    Execute an initial learning using all labeled samples, and produce an initial classifier
    Specify a number $N$ as the estimated number of unlabeled samples which will be positive-labeled
  2. Assign label for the unlabeled samples
    Compute the decision function values of all the unlabeled samples with the initial classifier.
    Label $N$ samples with the largest decision function values as positive, and the others as negative.
    Set a temporary effect factor $C_{tmp}^*$.
  Repeat
  3. Retrain the SVM classifiers
    Repeat
      Retrain the SVM classifiers over all labeled samples.
      Use new SVM classifiers to classify all the unlabeled samples.
      Switch labels of one pair of different-labeled unlabeled samples using a certain rule to make the value of the objective function in (11) decrease as much as possible.
    Until no pair of samples satisfying the switching condition is found.
  4. Adjust the value of $C_{tmp}^*$
    Increase the value of $C_{tmp}^*$ slightly
  Until $C_{tmp}^* \geq C^*$
```

The label switching operation in Step 3 guarantees that the objective function will decrease after switching, which improves the classification performance. Then Step 4 can pursue a reasonable error control by slightly increasing the impact of the unlabeled samples, until $C_{tmp}^* \geq C^*$. However, the TSVM algorithm also has a disadvantage that the number of unlabeled samples must be manually specified before training. If the estimated number is inconsistent with the intrinsic property of the unlabeled samples, the performance of the final classifier will be largely deteriorated. Actually, it is difficult to accurately estimate the number, *N*, even if the intrinsic property of the labeled samples is known.

To solve this problem, a new transductive learning method, so-called progressive transductive support vector machine (PTSVM) is proposed for SVM training [12]. In this method, all unlabeled samples are labeled gradually for approximately satisfying the requirement of Eq. (11). Instead of labeling all unlabeled samples in one time, in PTSVM, the labeling process is iteratively carried out. In each iteration, only one positive sample and one negative sample are labeled, which satisfies

$$i_1 = \arg\max_{j:0<\left|f\left(x_j^*\right)\right|<1} \left|f\left(x_j^*\right)\right| \tag{12}$$

$$i_2 = \underset{j:-1<\left|f\left(x_j^*\right)\right|<0}{\arg\max} \left|f\left(x_j^*\right)\right| \tag{13}$$

And the corresponding samples are labeled as follows:

$$\operatorname{label}\left(x_{i_1}^*\right) = \operatorname{sgn}\left(w \cdot x_{i_1}^* + b\right) \tag{14}$$

$$\operatorname{label}\left(x_{i_2}^*\right) = \operatorname{sgn}\left(w \cdot x_{i_2}^* + b\right) \tag{15}$$

This process is so-called pairwise labeling. Pairwise labeling is then iteratively carried out with the unlabeled samples, until all the unlabeled samples are outside the margin band of the separating hyperplane. During progressive learning, the new hyperplane will indicate that some earlier labeling is wrong, and then these labels should be canceled and the corresponding samples are restored as unlabeled ones. This process is so called dynamical adjusting, which makes PTSVM easier to recover from some early classification errors [12]. However, in PTSVM, the pairwise labeling is carried out pair by pair, which will largely increase the computation complexity. The computation complexity of normal SVM is $O\left(l^2\right)$ [5], where l is the number of samples. Obviously, the computation complexity of PTSVM is $O\left(N_1\left(n+k\right)^2\right)$, where N_1 is the number of iterations, n is the number of original labeled samples and k is the number of original unlabeled samples. For decreasing the computation complexity, more than a pair of positive and negative sample can be selected in each iteration. It means that a new region labeling [13] can be used, which can be illustrated as follows:

$$V_{\max} = \max_{j:0<\left|f\left(x_j^*\right)\right|<1} \left(f\left(x_j^*\right)\right) \tag{16}$$

$$V_{\min} = \min_{j:-1<\left|f\left(x_j^*\right)\right|<0} \left(f\left(x_j^*\right)\right) \tag{17}$$

For all remaining unlabeled samples $x_i^*, i = 1, \cdots, k'$, where k' is the number, if

$$V_{\max} - r_1 \le f\left(x_i^*\right) \le V_{\max} \tag{18}$$

or

$$V_{\min} \le f\left(x_i^*\right) \le V_{\min} + r_2 \tag{19}$$

The corresponding samples are labeled as follows

$$\operatorname{label}\left(x_i^*\right) = \operatorname{sgn}\left(w \cdot x_i^* + b\right) \tag{20}$$

where r_1 ($0 < r_1 < V_{\max}$) and r_2 ($0 < r_2 < abs\left(V_{\min}\right)$) are the predefined parameters for adjusting the scale of labeling region, which will directly determine the computation complexity and final performance. The computation complexity of the improved PTSVM algorithm is $O\left(N_2\left(n+k\right)^2\right)$, where N_2 is the number of iterations during learning process. Because $N_2 < N_1$, the computation complexity of the improved PTSVM is obviously less than PTSVM algorithm.

4. Collaborative SVM Learning Method

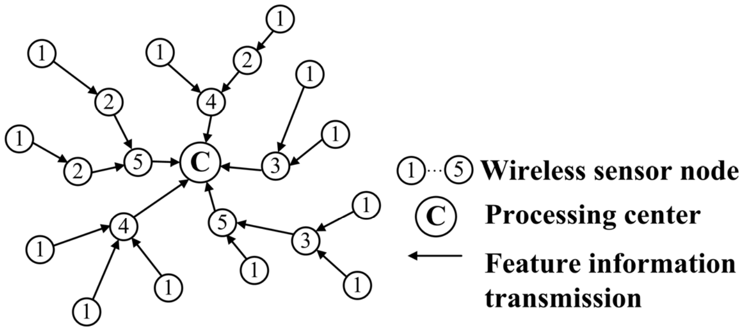
Computing mechanism refers to the information processing model deployed at the application layer of the protocol stack in the context of sensor networks. In this section, four computing paradigms: C-C/S, D-C/S, MA and CH, will be introduced to carry out centralized, distributed and collaborative semi-supervised classifier learning paradigms in WMSN, respectively.

4.1 Centralized learning paradigm

Classical SVM learning is a kind of centralized learning paradigm, which requires the samples of all wireless sensor nodes. In WMSN, centralized semi-supervised classifier learning process is implemented by processing center, which is a supernode with powerful signal processing ability. During the procedure of learning, all wireless sensor nodes carry out background subtraction and ILWT for target detection and feature extraction, and transmit all feature information to processing center. And then processing center implements centralized semi-supervised classifier learning based on the unlabeled samples. Essentially, centralized semi-supervised classifier learning is a sample of centralized client/server computing paradigm.

The centralized client/server computing paradigm is one of the most popular computing paradigms in WMSNs. For acquiring enough samples for classifier learning, at each time instant, wireless sensor nodes will autonomously transmit the sample information to the processing center for further processing. Although the C-C/S computing paradigm is widely used, this paradigm is not appropriate for data aggregation in WSN, especially, WMSN. Because the data amount of multimedia information is large, the data transmission will consume too much scarce resources such as battery power and network bandwidth, although feature extraction is used for compression. Furthermore, the data transmission is autonomously triggered by each sensor node. The confused data transmission in a short period will significantly deteriorate the QoS of network. Figure 1 illustrates the computing scenario of the centralized learning paradigm, where the number in each wireless sensor node indicates the approximate relative data amount sending by corresponding sensor node. Obviously, the closer a sensor node is to processing center, the more energy the sensor node will consume, because these sensor nodes have to be an intermediate sensor node to route the packets from other sensor nodes. It means that the sensor nodes closer to the computing center will die much more rapidly than other sensor nodes, which will accordingly decrease the lifetime of WMSN [14].

Figure 1. Computing scenario of centralized learning paradigm.



4.2 Distributed learning paradigm

4.2.1 Distributed client/server learning paradigm.

Recently, some distributed SVM-based classifier learning methods have been proposed for achieving target classification in WSN. The motivation of distributed learning methods is as follows. In SVM, only the samples which lie close to hyper-plane receive non-zero weights. Such samples are so-called support vectors (SVs). The intrinsic property of SVM algorithm determines that the number of SVs can be small compared to the total number of samples. Thus, SVs can provide a compact representation of all samples. Instead of transmitting the samples, only SVs need to be transmitted for learning, which can reduce energy consumption and time delays of data transmission. Distributed learning methods imply that the learning process can be carried out in D-C/S computing paradigm.

The D-C/S paradigm is also a well-known computing paradigm for hierarchical WMSNs, where wireless sensor nodes are divided into several clusters. Each cluster head acquires samples from the sensor nodes in its own cluster and achieves local learning. Afterwards, each cluster head transmits the SVs to processing center. Then further learning is implemented in processing center with all SVs to acquire the final SVs. Obviously, only SVs need to be transmitted, so data transmission energy can be largely reduced. Moreover, the learning process is carried out in distributed manner, which can decrease the computation complexity of semi-supervised classifier learning. But there is still a big drawback in D-C/S learning, which is based on the assumption that the distribution of the samples from all sensor nodes is consistent. Thus, if the distribution of one batch of data is inconsistent with other data, the SVs which are acquired from this batch of data just have little influence in the final results. The reason is that SVM is robust against data outliers, so the SVs acquired from the inconsistent samples will be considered as outliers since it is a desired property of the SVM algorithm, when the final learning is based on the SVs from all samples. In practice, the SVs acquired from the inconsistent samples cannot be ignored in the learning process, because this kind of SVs may present a new knowledge or new information, especially in time-varying target and environment. This effect is more accentuated for adaptively updating classifiers in online learning.

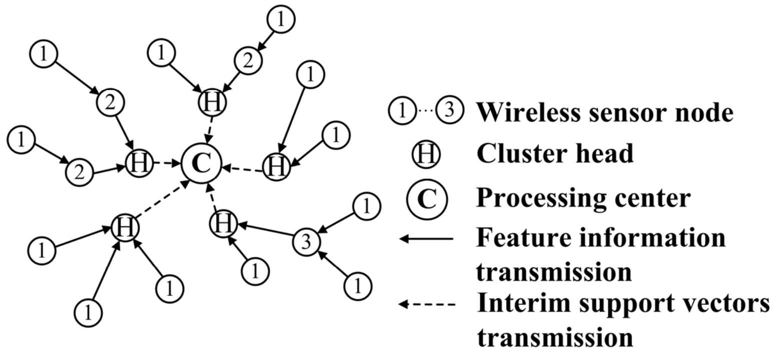
To overcome the drawback of the D-C/S learning, an improved D-C/S learning is adopted, which is extended from the SV-L-incremental SVM learning algorithm [6]. Actually, the cost of the errors on old SVs should be equivalent to the impact of the old samples. To approximate the average error over all samples by the average error over the SVs, the weighting method is used to modify the punishment of the errors on SVs. This can be easily achieved by training SVM with respect to a new loss function. Let $\left(x_i, y_i\right)_{i\in S_k}$ be the SVs acquired from the *k*th cluster head by centralized semi-supervised classifiers learning with the unlabeled samples in *k*th cluster. The improved cost function is defined as follow.

$$\Phi\left(\mathrm{w},\xi\right)=0.5\left\|\mathrm{w}\right\|^2+C\left(\sum_{k=1}^{c}L_k\sum_{i\in S_k}\xi_i\right) \tag{21}$$

where the L_k is the modifying factor of *k*th cluster, and c is the number of clusters. Here, L_k is the number of samples divided by number of SVs in *k*th cluster, $\xi_i\left(i\in S_k\right)$ are the corresponding slack variables and C is the user-specified parameter. It should be noted that the samples which receive zero weight are ignored. Figure 2 illustrates the computing scenario of the D-C/S learning paradigm.

Although the D-C/S paradigm can decrease the energy consumption of data transmission between cluster heads and processing centers, the data transmission in each cluster is also based on a client/server paradigm, which will also deteriorate the QoS of network and increase the energy consumption. Similar to C-C/S paradigm, the existence of processing center will also cause the imbalance of energy consumption and decrease the lifetime of WMSN. Obviously, the number of clusters will impact the energy consumption, time cost and performance. If the number of clusters becomes large, the data amount, energy consumption and time cost of data transmission and learning between cluster heads and processing center will increase. And if the number of clusters becomes small, the burden of data transmission and learning in each cluster will also increase. Furthermore, D-C/S learning based on SVs also has a bias with the centralized learning based on the raw samples [6], so if the number of clusters becomes large, the bias between the D-C/S learning and centralized learning will accordingly increase, which may worsen the performance of the classifiers.

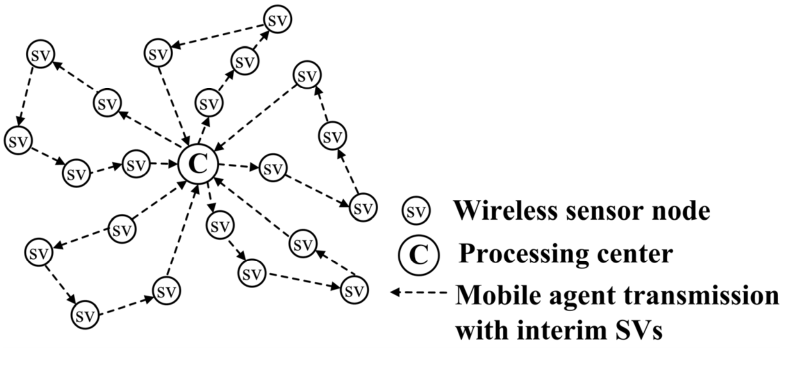
Figure 2. Computing scenario of D-C/S learning paradigm.


①···③ Wireless sensor node
Ⓗ Cluster head
Ⓒ Processing center
Feature information transmission
Interim support vectors transmission

4.2.2 Distributed mobile agent learning paradigm.

Mobile agent is a computational/analytical program which can migrate between different sensor nodes. In mobile agent computing paradigm, the mobile agents which carry the computing methods are dispatched by processing center. The mobile agents migrate among clients, performing local processing using resources available at each sensor node. Qi [15] proposed a mobile agent-based distributed sensor network (MADSN), wherein a MA visits the sensor nodes and incrementally fuses the data for making the final decision. In MADSN, an agent dispatched by cluster head sequentially accesses all sensor nodes to progressively fuse local information inside each cluster, and then processing center makes a global data fusion with the information from all clusters [15]. The computing scenario of distributed mobile agent learning paradigm is illustrated in Figure 3.

Figure 3. Computing scenario of MA learning paradigm.



Similar to D-C/S paradigm, the energy consumption, time cost and performance of MA paradigm are also determined by the number of clusters. Differently, MA paradigm can develop an energy-efficient method to provide progressive accuracy in each cluster, and there is no cluster head. Obviously, the process of sequential learning makes it feasible to replace distributed learning with incremental learning. Different to distributed learning based on all SVs, incremental learning trains SVM on the new data and the SVs from previous learning step [6]. It means that incremental learning is more equivalent to centralized learning than distributed learning. The cost function is as follows.

$$\Phi(\mathrm{w},\xi)=0.5\|\mathrm{w}\|^2+C\left(\sum_{i\in I}\xi_i+L\sum_{j\in S}\xi_j\right) \tag{22}$$

where $(x_i, y_i)_{i\in S}$ denotes old SVs, $(x_i, y_i)_{i\in I}$ denotes new samples, L is the number of samples in the previous batch divided by the number of SVs [6]. Actually, the key challenge of incremental learning is routing. In MADSN, the routing is totally determined by the energy consumption and time cost of transmission [15]. Compared to the confused data transmission of C/S paradigm, the sequential data transmission of MA paradigm can reduce the network congestion and decrease the energy consumption and time cost of data transmission in cluster. Thus, when the scale of cluster becomes larger, the advantage of MA paradigm is more evident. However, MA paradigm still needs a processing center, which is used to restore the computing methods, dispatch mobile agents and carry out the further learning based on the SVs transmitted by each cluster. Furthermore, the multimedia signal processing in WMSN is actually a complex task, so the overhead energy for creating, transmitting and destroying agents will largely increase the energy consumption, which may not be afforded by WMSN.

4.3 Collaborative hybrid learning paradigm

4.3.1 The overview of collaborative hybrid learning paradigm

P2P computing is a new framework for achieving network computing, which differs from the C/S computing paradigm and the MA paradigm in several crucial ways. In P2P computing, every sensor node acts as both a client and a server. During the computing process, sensor nodes can interact with other sensor nodes for improving the performance and decreasing the energy consumption of network. In fact, P2P computing has been applied in WSN [16, 17] and WMSN [4, 18] recently. It has been demonstrated that P2P computing has evident advantages on collaboration, local autonomy, high parallel performance, resource heterogeneity management [19].

In our earlier work [4], P2P computing was used for achieving collaborative target tracking, which was organized as a progressive distributed computing paradigm. That is, one sensor node integrates its result with previous results, and then it selects another sensor node according to some metrics of energy efficiency and predicted contribution, and transmits the integrated results for further processing. Sensor nodes selection and incremental processing is repeated in WSN, until a criterion is satisfied. The dynamic sensor nodes selection and routing ensure that a desired level of performance can be attained by spending the least amount of energy, because just a part of wireless sensor nodes needs to be used in one tracking step. However, different to target tracking, for ensuring the performance of classifiers learning, effective samples must be taken into consideration as much as possible. Thus, if the number of sensor nodes is large, the progressive distributed computing paradigm will spend a lot of time in accessing all sensor nodes.

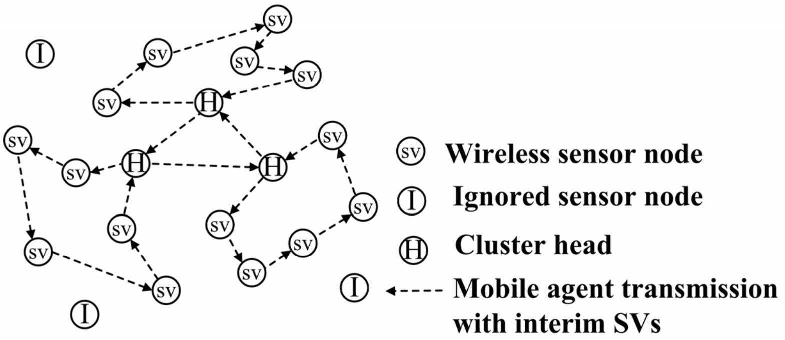
Here, we extend our earlier work on P2P computing paradigm to a hybrid computing paradigm, which adopts the progressive distributed computing in each cluster for achieving primary classifiers learning, and adopts the P2P computing between all cluster heads for further classifiers learning based on the SVs acquired by each cluster. The hierarchical structure is used to carry out the incremental learning in parallel manner, which can solve the problem of linearly increased time cost as the scale of WMSN increases. Furthermore, the P2P computing between all cluster heads avoids the imbalance of energy consumption, which also uses incremental learning to progressively aggravate the SVs from all cluster heads and acquires final classifiers.

Although incremental learning can balance the impact of SVs and samples by weighting methods, the inevitable imprecise samples still deteriorate its performance. For eliminating the bad effect of imprecise samples, a metric φ_e^i is introduced to evaluate the effectiveness of samples in *i*th sensor node. Essentially, the new samples can be considered to be imprecise if old SVs are almost considered as outliers in current step. Given the total number of old SVs, N_s^i, and the number of old SVs which are considered as outliers, N_o^i, the effectiveness metric of new samples can be defined as

$$\varphi_e^i = N_o^i / N_s^i \tag{23}$$

With the predefined threshold γ, if $\varphi_e^i > \gamma$, new samples are considered to be imprecise. Here, γ is set to 0.7. When the new samples are considered as imprecise samples, old SVs will be transmitted to the next sensor node instead of new SVs for preserving the effective information. Thus, the impact of missing or false target detection can be reduced. However, when some drastic changes occur, the old SVs may be not competent for the current situations. Then old SVs are useless, and φ_e^i is meaningless. For solving this problem, old SVs are discarded if *N* batches of samples are continuously judged as imprecise samples. At this moment, new SVs are inferred from new batch of data.

Figure 4. Computing scenario of collaborative hybrid learning paradigm.



Besides eliminating the bad effect of imprecise samples, the judgment of effectiveness can be also used for sensor nodes selection. That is, during incremental learning, some useless sensor nodes can be ignored to decrease energy consumption. Here, an access probability metric φ_a^i is introduced to evaluate the necessity of *i*th sensor node. φ_a^i is set to 1 in initialization. After a full access of all sensor nodes, φ_a^i is set as the effectiveness metric φ_e^i of *i*th sensor. In next iteration, *i*th sensor is accessed in the probability of φ_a^i. Thus, imprecise sensor nodes can be ignored, which can accordingly reduce the energy consumption. With the collaboration based on eliminating the imprecise samples and ignoring the useless sensor nodes, incremental classifiers learning can accordingly improve the learning performance and decrease the energy consumption. This is the motivation of collaborative classifiers learning. The computing scenario of collaborative hybrid learning paradigm is illustrated in Figure 4.

In WMSNs the performance, energy consumption and time cost of collaborative classifiers learning are directly determined by routing and clustering of hierarchical structure. In this paper, an ant optimization routing method is used to achieve energy-efficient routing and clustering, which will be discussed in following section.

4.3.2 Ant optimization routing for collaborative hybrid learning

Clustering and routing are two important issues addressed in WMSN, which are always discussed separately. Normally, the aim of routing and clustering is to implement data transmission with high energy-efficiency and high performance. As mentioned above, effective samples must be taken into consideration as much as possible to improve the performance of final classifiers. With metrics of effectiveness φ_a^i and access probability φ_a^i, the list of effective sensor nodes can be determined before each iteration of full access. Thus, different to progressive distributed computing paradigm, the routing can also be decided before each full access. Because the performance can be controlled by the sensor nodes selection based on metrics of effectiveness and access probability, only communication energy is used as a common metric in routing, which is one of the most important factors in data transmission. Given standard communication power p_0 along a standard distance l_0, the energy of data transmission from *i*th sensor node to *j*th sensor node φ_{ij} is defined as follows [20].

$$\varphi_{ij} = \frac{p_0 (4\pi)^2 \beta}{G_t G_r \lambda^2} \frac{\left(l_{ij}\right)^2 t}{l_0^2} \tag{24}$$

where l_{ij} is the Euclidean distance between *i*th sensor node and *j*th sensor node, G_t , G_r are transmitting and receiving gain respectively, λ is the wavelength of microwave used in communication and β is the system loss factor and t is the communication time. Moreover, in incremental learning, data amounts are nearly constant, which means that t is nearly a constant. Thus, the absolute energy consumption can be measured by $\left(l_{ij}\right)^2 / l_0{}^2$ [21].

Moreover, for improving the lifetime of WMSN, energy should be evenly consumed among all sensor nodes. Here, entropy H_{ij} is used to measure the evenness of reserved energy [10].

$$H_{ij} = -\sum_k p\left(\tilde{E}_{ij}^k\right) \log p\left(\tilde{E}_{ij}^k\right) \tag{25}$$

where $\tilde{E}_{ij}^k$ is the estimated energy reserved in *k*th sensor node if data transmission is carried out from *i*th sensor node to *j*th sensor node, and $p\left(\tilde{E}_{ij}^k\right)$ is the proportion of energy reserved in *k*th sensor node to the total reserved energy. The combined metric is as follow.

$$\varphi_{ij} = \left(l_{ij}{}^2\right) / \left(l_0^2 H_{ij}\right) \tag{26}$$

If there is only one cluster in WMSN, with the combined metric, the routing problem can be considered as a traveling salesman problem (TSP). But for hierarchical WMSNs, clustering and routing can be considered a combinational optimization problem, which is essentially a multiple traveling salesman problem (MTSP). In MTSP, there are m “salesman” who must visit a set of N “cities”. According to the requirement of the collaborative hybrid learning, all salesmen start traveling from different cities and return to the starting cities after traveling some cities. Every city must be visited exactly once by only one salesman. The objective is to find a routing with the minimum of total distances traveled by all salesmen [22]. Several researchers have been addressed in MTSP [23-25], adopting neural network (NN), simulated annealing (SA) and evolutionary algorithm (EA) solutions, respectively. Ant colony optimization (ACO) has recently emerged as one of the most important metaheuristic algorithm, which has been proved to be more effective for TSP than NN, SA and EA [26, 27]. MTSP is a generalization of TSP, so ACO is adopted in this paper for MTSP.

ACO is a kind of swarm intelligence, which is inspired by social insects. In ant colonies, ants communicate by depositing a substance called a pheromone. Paths with higher pheromone levels will more likely be chosen and thus reinforced, while the pheromone intensity of paths that are not chosen is decreased by evaporation. This form of indirect communication provides the ant colony shortest-path finding capabilities. In ACO, path selection is a stochastic procedure based on two parameters, the pheromone and heuristic values. The pheromone value gives an indication of the number of ants that chose the specific trail, while the heuristic value is a problem dependent quality measure. Once the ant arrives at its destination, the solution corresponding to the ant's followed path is evaluated and the pheromone value of the path is increased accordingly. Additionally, evaporation causes the pheromone level to diminish gradually. Hence, trails that are not reinforced gradually lose pheromone and will in turn have a lower probability of being chosen by subsequent ants. However, the ant does not choose its direction based on the level of pheromone exclusively, but also takes the proximity of the nest and of the food source into account. This allows the discovery of new and potentially shorter paths.

In MTSP, one salesman should first travel to a number of cities, and the next salesman travels to a number of unvisited cities. In this way, all the salesmen travel the total cities. The number of cities which are traveled by each salesman is randomly generated in a certain range. Let n_i denote the number of cities traveled by *i*th salesman, then

$$\begin{cases} T_i \geq n_i \geq 2 \\ \sum_{i=1}^{m} n_i = N - m \end{cases} \tag{27}$$

where T_i is the maximum number of cities that *i*th salesman can travel, m is the number of salesman, and N is the number of cities. Let D_i denotes the total distance traveled by *i*th salesmen. The target function is described as follows:

$$\text{Min}\left(\sum_{i=1}^{m} D_i\right) \tag{28}$$

For solving the routing and clustering problem of WMSN, consider the sensor nodes as cities, and let the combined metric, φ_{ij}, denote the distance between *i*th sensor node and *j*th sensor node. During optimization, m ants are used to construct a potential solution together. Then k groups of ants cooperate to search the final best solution. In each group, every ant acts as a salesman who travels between different sensor nodes. Before traveling, the number of sensor nodes which are traveled by each ant is randomly determined according to Eq. (27). Assign a sub-tour list H_j^i to *j*th ant ($j = 1, 2, \cdots, m$) of *i*th group ($i = 1, 2, \cdots, k$), which records the sensor nodes traveled by *j*th ant. When the number of sensor nodes is equal to n_i, *j*th ant stops the migration and returns to its starting sensor node. The set of all sub-tour lists forms a potential solution. In each iteration, k potential solutions can be acquired for finding the best solution.

Route selection for each ant is independently determined by the pheromone and heuristic values of each line. Suppose that the probability of *v*th ant in *u*th group moving from *i*th sensor node to *j*th sensor node at time *t* is $P_{ij}^{uv}(t)$. The following probabilistic formula can be given

$$p_{ij}^{uv}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha_1}\left[\eta_{ij}(t)\right]^{\alpha_2}}{\sum\limits_{s\in J_u}\left[\tau_{is}(t)\right]^{\alpha_1}\left[\eta_{is}(t)\right]^{\alpha_2}} & \text{if } j\in J_u \\ 0 & \text{Otherwise}\end{cases} \tag{29}$$

where $\tau_{ij}(t)$ is the pheromone of the route between *i*th sensor node and *j*th sensor node. $\eta_{ij}(t)$ is the heuristic value denoting the expectation for migrating from the *i*th sensor node to the *j*th sensor node. Normally, $\eta_{ij}(t)$ is considered as the reciprocal of the "distance" between two sensor nodes.

$$\eta_{ij}(t)=1/\varphi_{ij}=\left(l_0^2H_{ij}\right)/\left(l_{ij}{}^2\right) \tag{30}$$

And J_u denotes the sensor nodes that *u*th group of ants have not visited, which is defined as follows.

$$J_u=\{1,2,\cdots,N\}-\bigcup_{v=1}^{m}H_u^v \tag{31}$$

where H_u^v is the set of sensor nodes accessed by *v*th ant in *u*th group. The exponents α_1 and α_2 are positive parameters, which determine the relation between pheromone information and heuristic values.

After all groups of ants return to their starting sensor nodes, the pheromone value of all paths must be updated to reflect the ant's performance and the quality of the potential solutions. According to the suggestion of [26], the initialization value of pheromone, τ_0, is set to $\tau_0=\left(N\cdot L_{nn}\right)^{-1}$, where L_{nn} is tour length produced by the nearest neighbor heuristic [28] and N is the number of cities. The pheromone updating is a key factor of the adaptive learning process in ACO, which can be defined as follows.

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}\quad \rho\in(0,1) \tag{32}$$

where ρ is the parameter for presenting the speed of evaporation, $\Delta\tau_{ij}$ is the adding pheromone to the line between *i*th sensor node and *j*th sensor node in this iteration, which is given as follows.

$$\Delta\tau_{ij}=\begin{cases}Q/D^* & \text{If line } ij \text{ is passed by the optimal group of ants} \\ 0 & \text{Otherwise}\end{cases} \tag{33}$$

where Q is a predefined constant, D^* is the sum of defined distance accessed by the optimal group.

$$D^*=\sum_{j=1}^{m}\sum_{i=1}^{n_i}\varphi_{ij}+\sum_{c=1}^{m}\varphi_{h_c} \tag{34}$$

where φ_{ij} is the combined metric corresponding to the *i*th segment of the routes traveled by the *j*th ant, and φ_{h_c} is the combined metric corresponding to the *c*th segment of the routes between all cluster heads. Here, the starting sensor nodes are considered as the cluster heads of different clusters.

Obviously, pheromone updating considers both the distance of routes and the performance of solutions. The route selection and pheromone updating are repeated for global searching, until some criterion is satisfied. Then the current best solution is considered as a good approximation of the optimal solution. With the best solution, the clustering and the routing in each cluster can be easily determined. With the definition of combined metric, the optimized routing and clustering results determined by ACO can decreasing the energy consumption and prolonging the lifetime of WMSN.

Although ACO is a centralized algorithm in this paper, it has been proved that ACO is a simple but effective routing algorithm for networks [29], especially for WSNs [30]. Usually, the computing ability of multimedia sensor nodes is better than normal sensor nodes. Thus, ACO can be successfully carried out in WMSNs. The procedure of ACO based routing and clustering is illustrated as follows.

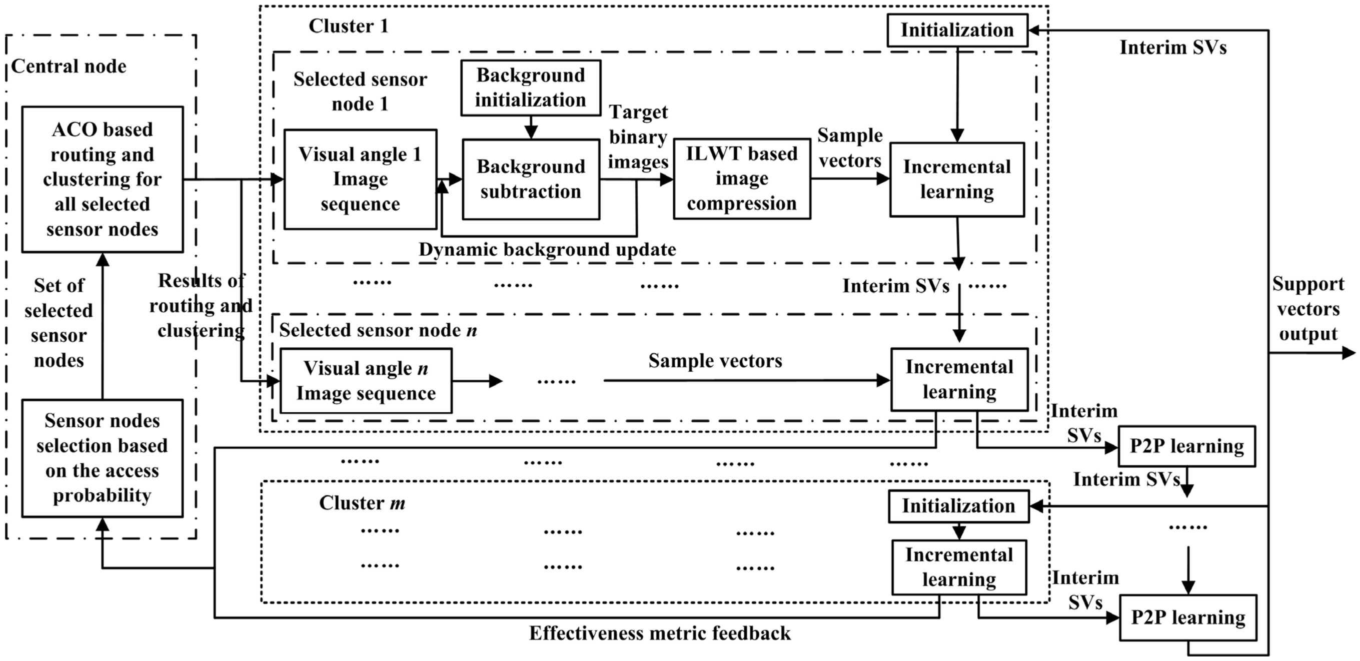
```
Procedure  ACO Based Routing and Clustering in WMSNs

1. Initialization
   Initialize the pheromone of lines between all sensor nodes, $\tau_{ij}$ $(i, j = 1, 2, \cdots, n)$
   Initialize the ant colony which contains $k$ groups of ants. And each group consists of $m$ ants.
Repeat
2. Assign the number of sensor nodes to every ants
   For each group $i \in [1, \cdots, k]$ and each ant $j \in [1, \cdots, m]$:
       Randomly select a sensor node as starting sensor nodes for current ant.
       Randomly assign a number of sensor nodes, $n_j^i$, to jth ant in ith group.
   Endfor
3. Route Selection
   For each group $i \in [1, \cdots, k]$ and each ant $j \in [1, \cdots, m]$:
       If the number of sensor nodes in sub-tour list $H_j^i$ is less than $n_j^i$
           Then generate the probability of potential routes by Eq. (29) and select the next sensor node
       Else current ant returns to its starting sensor node
       EndIf
   Endfor
4. Pheromone Updating
   For each group $i \in [1, \cdots, k]$:
     Calculate the total distance, $D_i^*$, accessed by current group of ants.
   Endfor
   Find the optimal group of ants with the smallest $D_i^*$
   Update the pheromone by Eq. (32) and Eq. (33)
Until stopping condition is satisfied (usually when the number of iterations reaches a predefined
threshold, or when the best solution does not update for a certain number of iterations.)
5. Output results
   Output the routing and clustering results indicated by current best solution
```

The overview of collaborative hybrid learning paradigm is illustrated in Figure 5. For implementing the optimization of routing and clustering in WMSNs, a specific sensor node called central node is used to carry out sensor nodes selection and ACO algorithm, which has powerful computing and wireless communication ability. Last cluster head in P2P computing paradigm return the current SVs and effectiveness metrics of all sensor nodes to central node. Then the central node selects the sensor nodes according to the access probability. If the set of selected sensor nodes changes, the central node will optimize the routing and clustering according to current set of selected sensor nodes and transmits the result to each sensor node by flooding for a new iteration of collaborative hybrid learning.

With four different learning paradigms, the classifiers will have different performance in target classification, and the learning processes will have different energy consumption and time cost. For investigating the performance of four different learning paradigms, an online learning and classifying experiment will be illustrated in the following section, and a simulation experiment will be also implemented to present the advantages of ACO based routing and clustering.

Figure 5. Overview of collaborative hybrid learning paradigm.



5. Experiments

5.1 Simulated routing and clustering experiments

For comparing the performance of routing and clustering, some simulation experiments were carried out, where SA, GA and ACO based MTSP algorithms are used. Moreover, a method which independently implements routing and clustering was used for investigating the difference between TSP and MTSP, where maximum entropy inference [31, 32] is used for clustering and ACO is used for routing. In this experiment, starting temperature and final temperature of SA are set to 1000 and 0.01, and the temperature decreasing parameter is set to 0.95 [24]. And in GA algorithm, the number of individuals is 50, the probabilities of reproduction, crossover and mutation are 0.1, 0.6 and 0.3. And the parameters of ACO in both TSP and MTSP are set as following: $\beta = 2$, $Q = 1$, $\alpha = \rho = 0.1$, $\tau_0 = \left(N \cdot L_{nn}\right)^{-1}$. Here, N is the number of sensor nodes, which is set to 200. The number of ants in TSP and the number of groups of ants in MTSP are both set to 10. Maximum iterations of three algorithms are all set to 300. All sensor nodes are randomly deployed in a square field with area of $100m \times 100m$. The number of clusters, m, changes from 3 to 10 for investigating the impact of the number of clusters. The maximum number of sensor nodes in each cluster, T_i, is set to $1.5N/m$.

Moreover, in simulation, the reserved energy can not be acquired. Thus, it is assumed that the energy consumption between two sensor nodes is measured by $\left(l_{ij}\right)^2/l_0^{\,2}$, where l_{ij} is the Euclidean distance between two sensor nodes, and l_0 is a standard distance, which is set to 1m. Simulation experiments are achieved in a wireless multimedia sensor node with 200 MHz embedded processor (ARM9).

Figure 6 illustrates the routing and clustering results of ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP where the number of sensor nodes, N_s, is 200, and the number of clusters, N_c, is 3. The total energy consumption metrics optimized by ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP are 10373, 10034, 9671 and 9221, respectively. Here, the total energy consumption metric is defined without the information of reserved energy, which is illustrated as follows

$$D^* = \sum_{j=1}^{m}\sum_{i=1}^{n_i}\left(\left(l_{ij}\right)^2/l_0^{\,2}\right) + \sum_{c=1}^{m}\left(\left(l_{h_c}\right)^2/l_0^{\,2}\right) \qquad (35)$$

where l_{h_c} is the Euclidean distance between cth cluster head and (c+1)th cluster head.

Figure 6. The routing and clustering results of (a) ACO-TSP, (b) SA-MTSP, (c) GA-MTSP, and (d) ACO-MTSP with the following parameters, $N = 200$, $m = 3$.

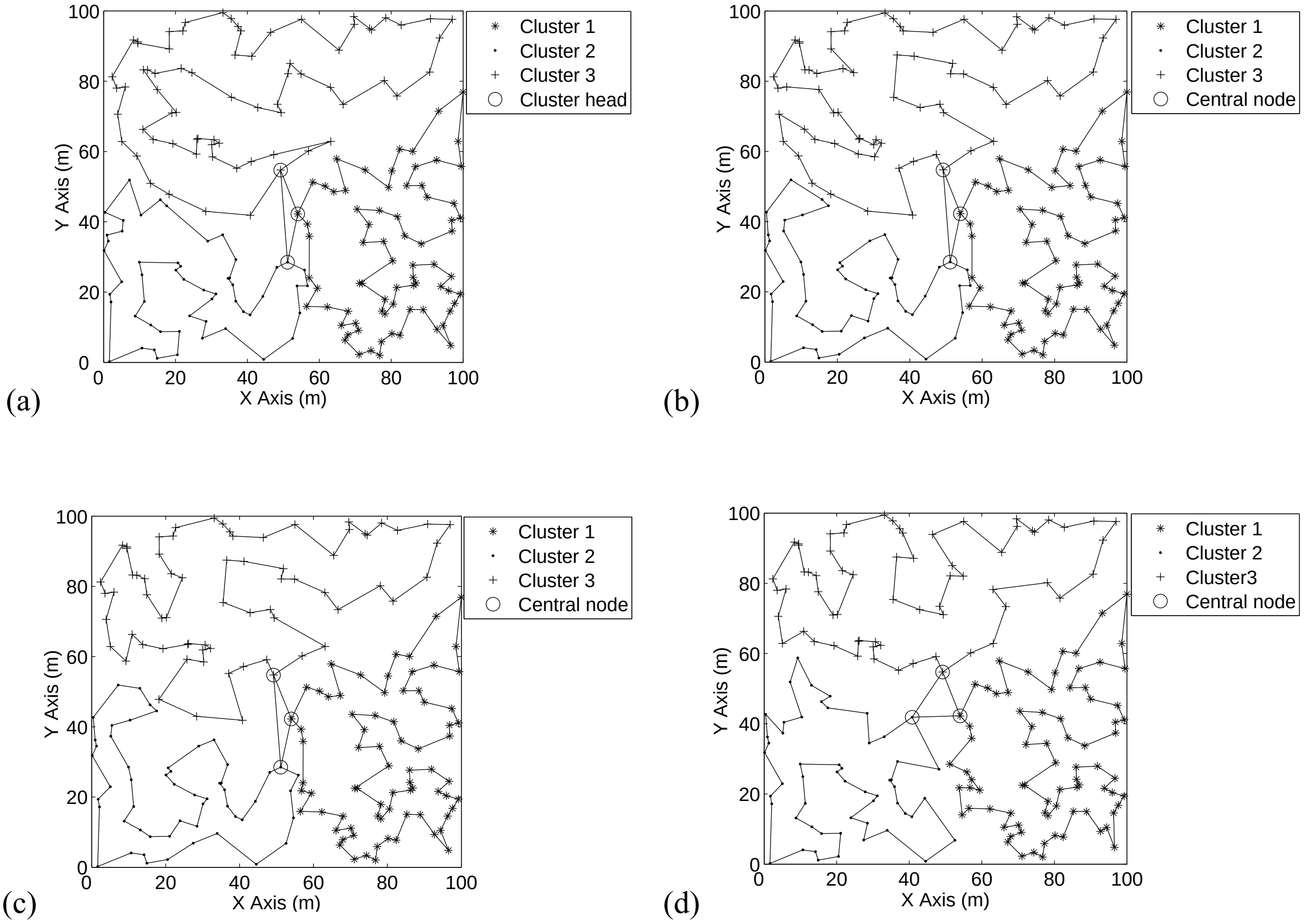
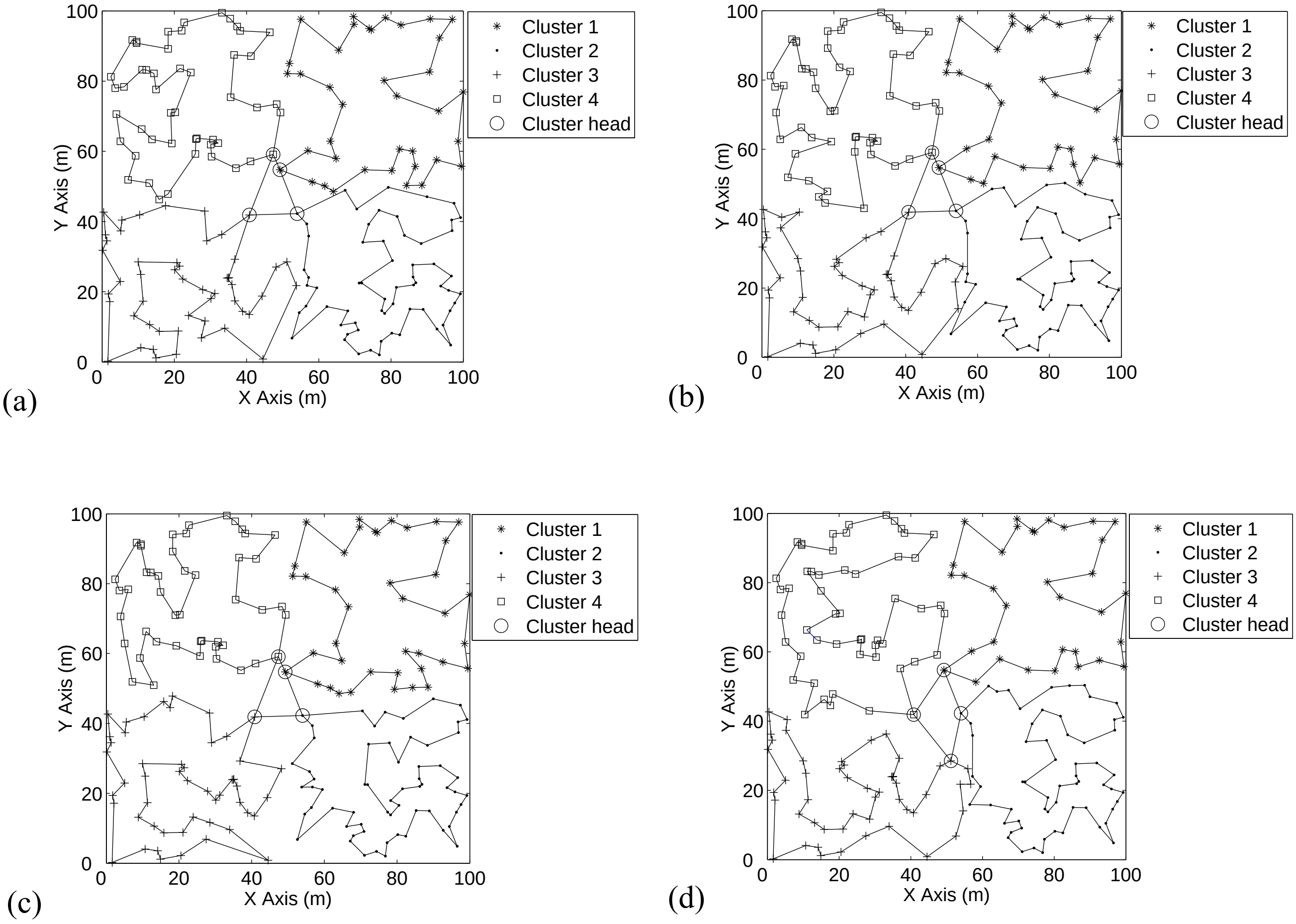


Figure 7. The routing and clustering results of (a) ACO-TSP, (b) SA-MTSP, (c) GA-MTSP, and (d) ACO-MTSP with the following parameters, $N = 200$, $m = 4$.



The experimental results with the following parameters: $N_s = 200$, $N_c = 4$ are illustrated in Figure 7. The total energy consumption metrics optimized by ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP are 9011, 8963, 8747 and 8692, respectively. The experimental results in Figure 6 and Figure 7 verify that ACO-MTSP method has best performance for the routing and clustering of WMSN. Theoretically, with the optimization of ACO-MTSP, WMSN can implement the collaborative hybrid computing with the lowest energy consumption.

Figure 8 then shows the mean of total energy consumption metric optimized by ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP. For detailing, the experiments are carried out with different number of clusters, where the number of clusters changes from 3 to 10. Here, the number of sensor nodes keeps 200 for all experiments. And Figure 9 illustrates the corresponding time cost for optimization. For all experiments, the average results are acquired through 50 independent operations with random initialization.

Figure 8. The mean of total energy consumption metric optimized by ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP with different number of clusters, where the number of clusters changes from 3 to 10 and the number of sensor nodes keeps 200 for all experiments.

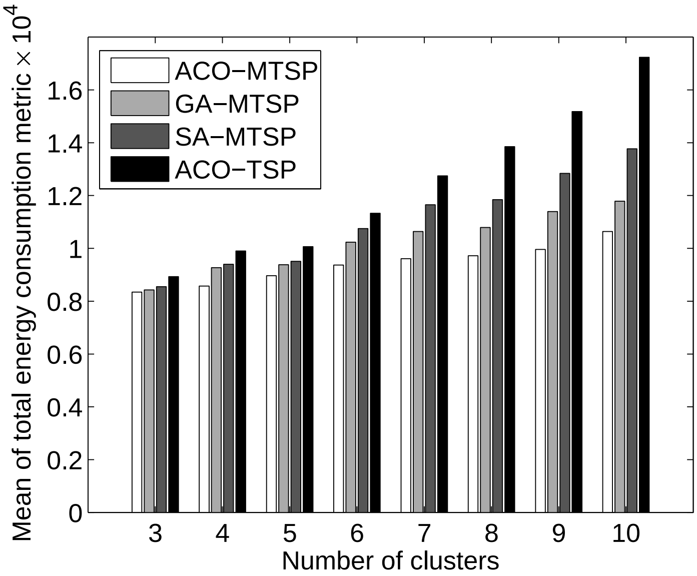
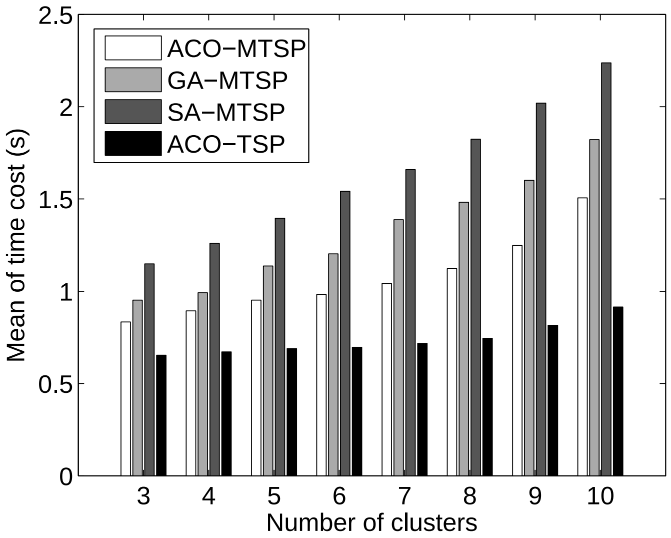


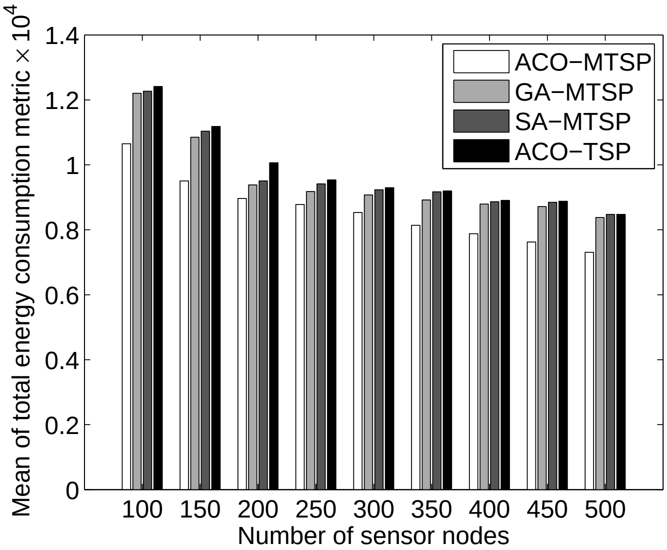
Figure 9. The mean of time cost for optimization in ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP with different number of clusters, where the number of clusters changes from 3 to 10 and the number of sensor nodes keeps 200 for all experiments.



When the number of clusters increases, the mean of total energy consumption metric optimized by four algorithms all increase. The reason is that the computation complexity accordingly increases as the number of clusters increases, which bring more difficulty for optimization. Furthermore, the more clusters will also bring more limitations, which may deteriorate the optimized results. Obviously, ACO-MTSP has the best performance in the optimization of routing and clustering, because the total energy consumption metric optimized by ACO-MTSP is the least in all experiments. Most importantly, the total energy consumption metric increases slowly as the number of clusters increases, which implies that ACO-MTSP is more scalable than other three algorithms. It must be noticed that the total energy consumption metric of ACO-TSP increases most rapidly. It means that the ACO-TSP is not competent for the routing and clustering of collaborative hybrid learning paradigm in WMSN, especially when the number of clusters is big. Moreover, the time cost of ACO-TSP is the lowest, because it just needs to implement routing for each cluster, respectively, which largely decreases the computation complexity. However, the verified poor performance in the routing and clustering of collaborative hybrid learning prevents its application. Compared to GA-MTSP and SA-MTSP algorithms, ACO-MTSP also has the lower time cost. Thus, ACO-MTSP can be considered as an efficient algorithm for the routing and clustering of collaborative hybrid learning in WMSN.

Furthermore, for investigating the impact of the number of sensor nodes, further research is carried out for WMSN with the number of sensor nodes changing from 100 to 500, where the number of clusters is fixedly set to 5. For each kind of WMSN, the average results are also acquired through 50 independent operations with random initialization. The mean of total energy consumption metric optimized by ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP are illustrated in Figure 10. The corresponding time cost results are illustrated in Figure 11.

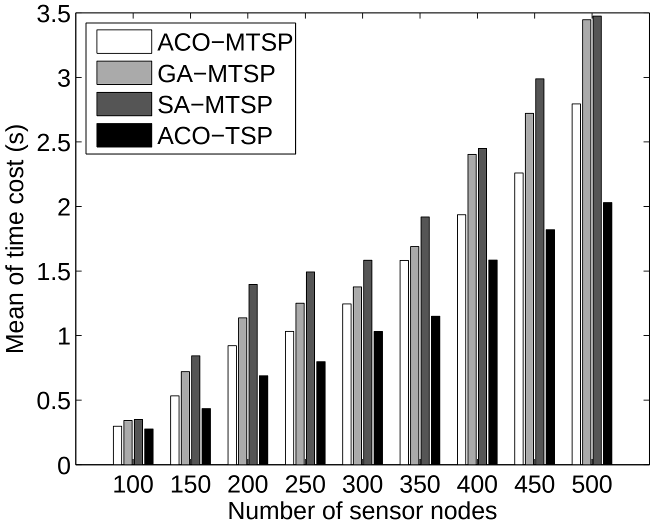
Figure 10. The mean of total energy consumption metric optimized by ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP with the number of sensor nodes changing from 100 to 500, where the number of clusters is fixedly set to 5.



The results in Figures 10 and 11 verify that the total energy consumption metric will decrease as the number of sensor nodes increases. It is because the energy consumption metric is proportional to the square of distance between sensor nodes. When the number of sensor nodes increases, the density of sensor nodes will also increase, which may decreases the distance between sensor nodes and accordingly decreases the energy consumption metric. However, the time cost for optimization remarkably increases as the number of sensor nodes increases, because the computation complexity of routing and clustering problem will exponentially increase. Similar to the first experiment, ACO-MTSP still has the best performance in the optimization of routing and clustering, and the computation time of ACO-MTSP is still lower than GA-MTSP and SA-MTSP. These results demonstrate that

ACO-MTSP is robust and efficient for routing and clustering of collaborative hybrid learning in WMSN. With the ACO-MTSP method, the performance of collaborative hybrid learning will be discussed in the following section.

Figure 11. The mean of time cost for optimization in ACO-TSP, SA-MTSP, GA-MTSP and ACO-MTSP with the number of sensor nodes changing from 100 to 500, where the number of clusters is fixedly set to 5.



5.2 Real world target classification experiments

In this section, the performance of collaborative semi-supervised classifier learning algorithm for target classification with hybrid computing paradigm is investigated and compared to the centralized semi-supervised learning, distributed client/server semi-supervised learning and distributed mobile agent semi-supervised learning. Wireless multimedia sensor network which consists of 18 wireless multimedia sensor nodes is deployed in a room. According to the requirement of different learning paradigm, a processing node is deployed in the middle of the room, which is used as the processing center in C-C/S, D-C/S and MA learning paradigms, and it is also used as the central node in collaborative hybrid learning paradigm. The deployment scenarios of WMSN with different learning paradigms are illustrated in Figure 12.

The processing node and sensor nodes are all multimedia sensor nodes. Each one contains one image/pyroelectric-infrared sensor pair which has a 3.6mm camera lens with 60° visual angle and a 200MHz embedded processor (ARM9). In experiments, sensor node is working autonomously. When target enters the tracking area, the correlative multimedia sensor nodes are awakened by the pyroelectric-infrared sensor module. Then the image acquisition, target extraction, feature extraction are performed continuously until the target leaves. Target detection and feature extraction are processed at a frame rate of 10Hz. The images are down-sampled to 160×120 pixels. Then the binary images of detected targets are compressed and reconstructed to 16×16 pixels. In all learning paradigms, data transmission and semi-supervised learning is implemented every 2 seconds. The package amount in C-C/S learning and D-C/S learning is 800 bytes, which carries the raw data. The package amount in MA learning is 450 bytes, which carries the information of mobile agent and the

SVs. The package amount in collaborative hybrid learning is 160 bytes which only carries the SVs. During surveillance, each sensor node shares the information of reserved energy per minute. Wireless communication has data rate of 19.2 kbps, where the basic frequency is 900 MHz and the bandwidth is 7.2 MHz. CSMS/CA is used as MAC protocol.

Figure 12. The deployment scenario of WMSN with different learning paradigms: (a) C-C/S, (b) D-C/S, (c) MA and (d) collaborative hybrid learning.

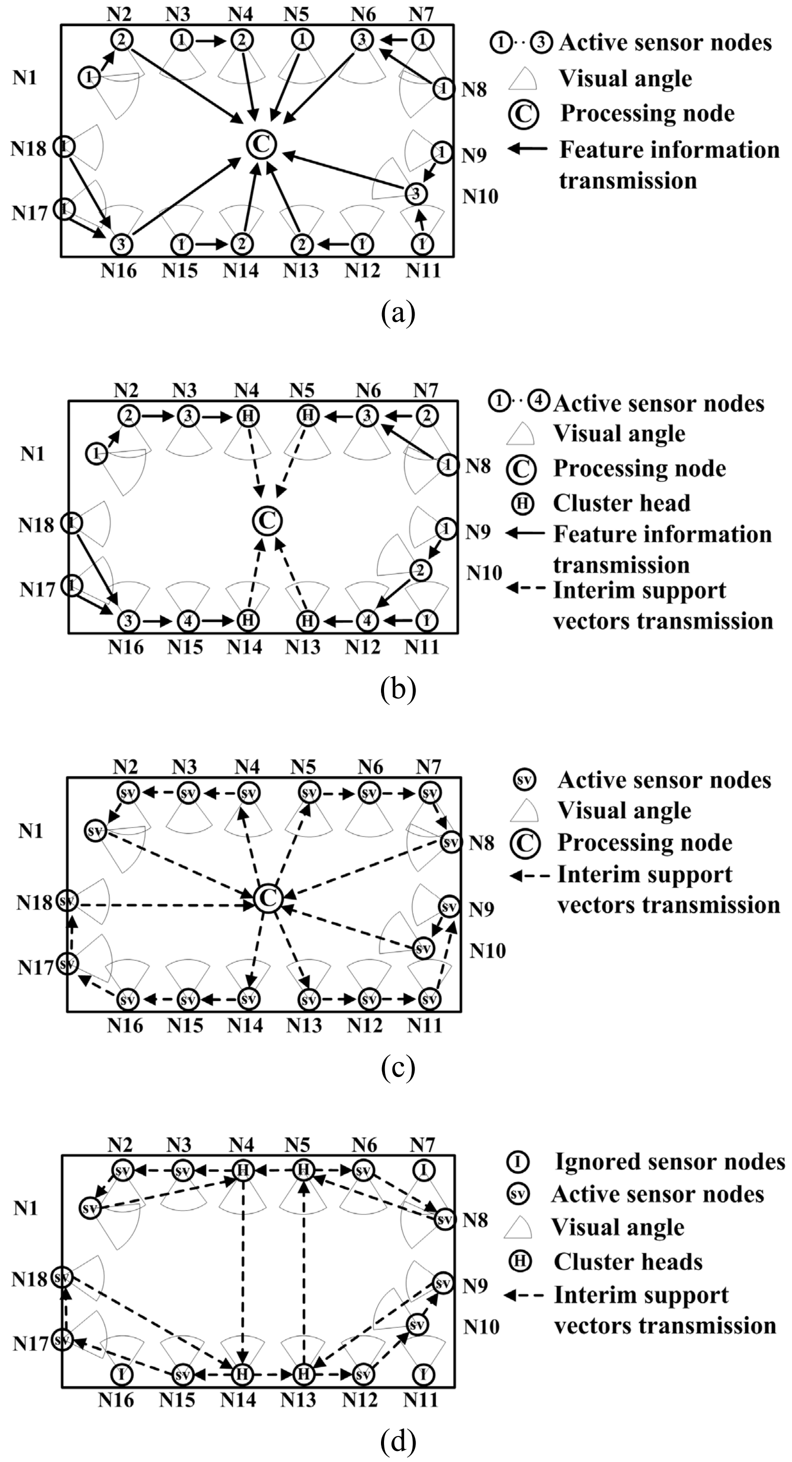
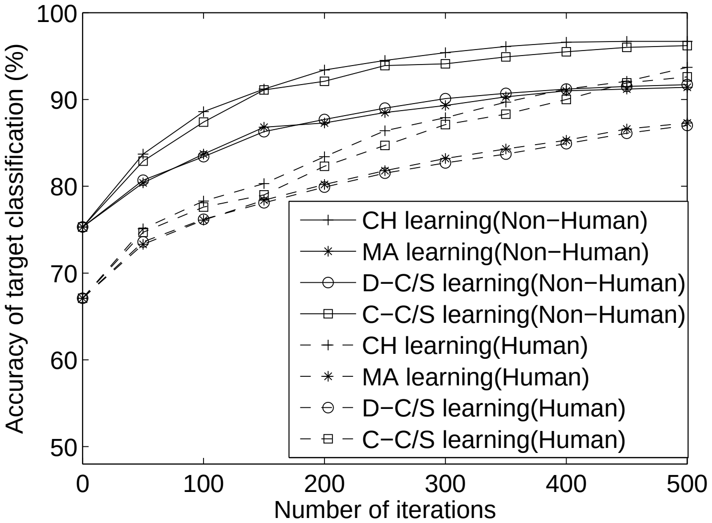


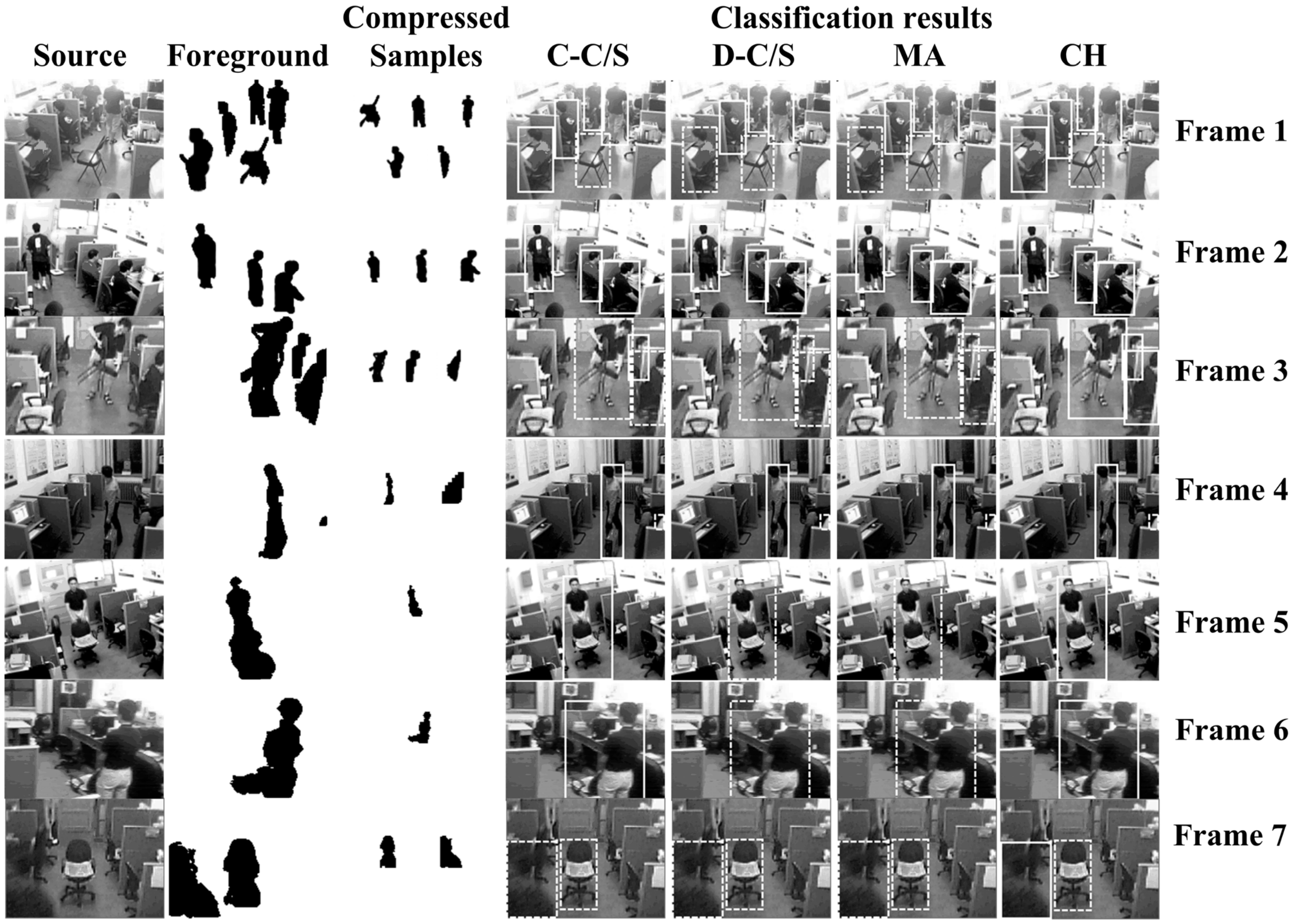
Figure 13. The classification accuracy of human target and non-human target with different number of iterations in four different learning paradigms.



In the experiment, the online learning process is iterated in four different learning paradigms, respectively. Target classification is simultaneously implemented with the classifiers which are acquired by different learning paradigms at different training stages. Figure 13 illustrates the accuracy of target classification with different number of iterations in four different learning paradigms. The results verify that the accuracy of target classification in four learning paradigms all increase as the number of iterations increases, which means that more samples will bring higher accuracy for all four learning paradigms. Furthermore, for each learning paradigm, the accuracy of non-human target classification is higher than the accuracy of human target classification. The reason is that human may make some poses, such as stoop, squat and grovel, which will lead to misclassification. Obviously, the accuracy of the classifiers acquired in collaborative hybrid learning paradigm is highest whether the target is human or non-human, although only SVs are used for the final learning in collaborative hybrid learning. It is because the sensor nodes selection can weaken the impact of inevitable missing detection and false detection, and the imprecise samples are mainly ignored in the learning process, which can accordingly improve the classification accuracy in online learning process.With the four different classifiers, some frames of online classification results acquired from different sensor nodes are illustrated in Figure 14.

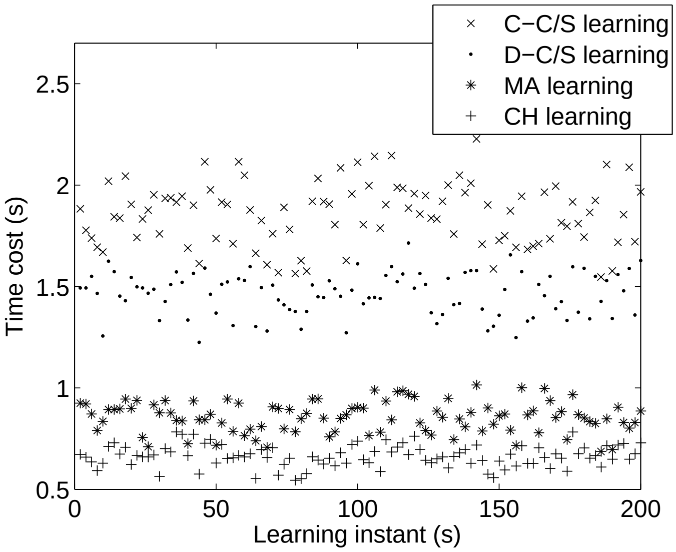
The results are represented by bounding boxes with different types of lines. Solid lines represent that the target is classified as a human target, while the dashed line represents the target classified as a non-human target. Obviously, the foreground targets can be properly acquired and compressed by background subtraction and ILWT algorithm. Then the target classification results demonstrate that the classifier acquired by collaborative hybrid learning paradigm performs best, because the corresponding classifier can successfully achieve target classification almost at all frames, unless target is almost wholly occluded. The failed examples are illustrated in Frame 4, where the human target sitting near the table is misclassified. Compared to CH learning paradigm, the classifier acquired by C-C/S learning paradigm cannot work accurately, when targets are heavily occluded. The examples of misclassification can be found in Frame 3, 4 and 7. However, the classifier acquired by C-C/S learning paradigm still performs better than D-C/S learning paradigm and MA learning paradigm. The classification results verify that the classifier acquired by D-C/S paradigm and MA paradigm may lead to misclassification if targets are occluded by obstacles or other targets. For examples, at Frames 1, 3, 4, 5, 6 and 7, the human targets are misclassified. With all the above experimental results, it is obvious that collaborative hybrid learning paradigm has best performance in semi-supervised classifier learning for WMSN.

Figure 14. Selected frames of target classification results in different sensor nodes.

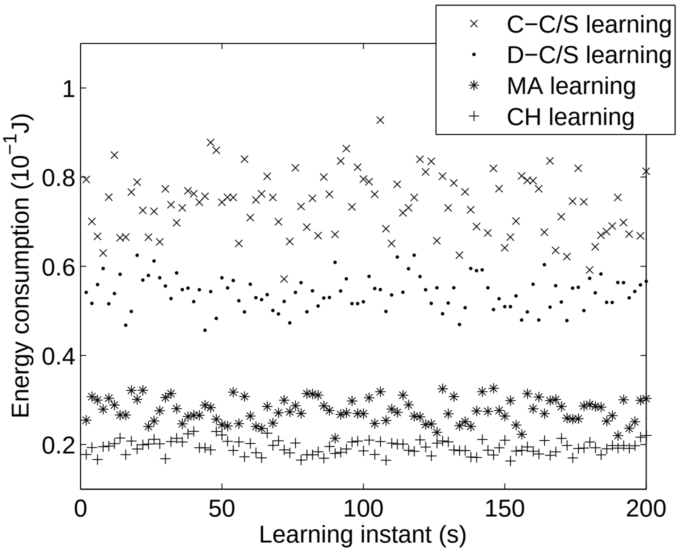


Furthermore, the energy consumption and time cost of each learning paradigm are also very important in WMSN, which has great impact on the lifetime and QoS of WMSN. Then 100 iterations of online classifier learning are carried out in each learning paradigm, and the instantaneous time cost and energy consumption at each learning step are compared for investigating the cost of each learning paradigm. The results are illustrated in Figure 15, where each point denotes the total time cost/energy consumption of full access among all selected sensor nodes. Here, the energy consumption is the sum of energy consumed for data transmission in all sensor nodes at current time instant. And the time cost is the sum of time cost in classifier learning and data transmission, which is measured by the time interval between the start and the end of each full learning access in different iterations.

Figure 15. The (a) time cost and (b) energy consumption in 100 iterations of online classifier learning with four different learning paradigms.



(a) Time cost



(b) Energy consumption

The time cost and energy consumption results of 100 learning steps illustrate that collaborative hybrid learning paradigm has the best performance in the terms of time cost and energy consumption. That is because hybrid learning paradigm implements progressive classifier learning by the collaboration of selected sensor nodes, which can largely decrease the congestion of data transmission. Moreover, the hybrid structure also brings more flexibility and balances the time cost and learning performance. Importantly, the ant optimization routing ensures the routing and clustering of data transmission is the optimal solution with the best energy efficiency. Compared to collaborative hybrid learning paradigm, MA learning paradigm spends more time and more energy in the classifier learning process, since the information of mobile agent will bring overhead cost. Furthermore, the access of imprecise sensor node will also cause unnecessary time cost and energy consumption. However, because MA learning paradigm implements classifier learning in progressive manner, the time cost and energy consumption of MA learning paradigm are both less than other two C/S learning paradigms. Obviously, in C-C/S and D-C/S learning paradigms, the raw data transmission largely deteriorates the QoS of WMSN and remarkably increases the energy consumption and time cost. Besides, the centralized classifier learning will also bring more time cost, especially when the number of samples is big. Moreover, the time cost and energy consumption of D-C/S learning paradigm is less than C-C/S learning paradigm, because D-C/S learning paradigm carries out distributed classifier learning in WMSN which will decrease the data amount in transmission.

In summary, the classification accuracy, time cost and energy consumption of four learning paradigms verify that the proposed collaborative hybrid semi-supervised classifier learning algorithm has outstanding performance of online learning and classifying in WMSN with highly limited computing ability and energy resources.

6. Conclusions

In this paper, a collaborative hybrid semi-supervised classifier learning algorithm is proposed for achieving online SVM based classifier learning and target classification in strictly constrained hierarchical WMSN. For improving the energy efficiency and decreasing time cost, the proposed algorithm achieve classifier learning by a hybrid structure in hierarchical WMSNs, which carries out progressive distributed learning in each cluster and P2P learning between cluster heads. Besides, ACO algorithm is introduced to implement energy-efficient routing and clustering. A sensor nodes selection strategy is also introduced for the collaboration of sensor nodes, which can evaluate the contribution of each sensor node and select the effective sensor nodes for classifier learning. With the collaboration of sensor nodes, the impact of imprecise samples can be weakened, and the classification accuracy and energy efficiency can be accordingly improved. Then the performance of the proposed collaborative hybrid semi-supervised classifier learning algorithm is investigated and compared with other three learning algorithms, which are implemented in C-C/S, D-C/S and MA learning paradigms. The target classification results of complex indoor experiment demonstrate that the proposed collaborative hybrid semi-supervised classifier learning paradigm can effectively achieve online classifier learning and target classification in highly constrained WMSN. Compared to other three learning paradigms, the proposed algorithm can significantly reduce the time cost and energy consumption and improve the classification accuracy, which implies that the proposed algorithm is an effective learning algorithm for achieving target classification with outstanding accuracy and energy efficiency.

Appendix A

Let I_x^i be the intensity of location x in ith frame of N consecutive frames. σ_x and λ_x are the corresponding standard deviation and median value of intensities at location x in all images. The initial background model for a stationary pixel location x, $[m_x, n_x, d_x]$, is obtained as:

$$\begin{bmatrix} m_x \\ n_x \\ d_x \end{bmatrix} = \begin{bmatrix} \min_i \{I_x^i\} \\ \max_i \{I_x^i\} \\ \max_i \{|I_x^i - I_x^{i-1}|\} \end{bmatrix}, \text{where } |I_x^i - \lambda_x| < 2\sigma_x \tag{36}$$

After initialization, detection support map (gS) and motion support map (mS) are used to update the background model, which are defined as follows:

$$gS_x(t) = \begin{cases} gS_x(t-1)+1 & \text{if } x \text{ is background pixel} \\ gS_x(t-1) & \text{if } x \text{ is foreground pixel} \end{cases} \tag{37}$$

$$mS_x(t) = \begin{cases} mS_x(t-1)+1 & \text{if} \left(\left|I_x^t - I_x^{t+1}\right| > 2\sigma_x\right) \text{ and } \left(\left|I_x^{t-1} - I_x^t\right| > 2\sigma_x\right) \\ mS_x(t-1) & \text{otherwise} \end{cases} \tag{38}$$

During the process of background construction, the background model is separately computed for all pixels which are classified as foreground pixels $\left[m_x^f, n_x^f, d_x^f\right]$ and background pixels $\left[b_x^b, n_x^b, d_x^b\right]$. Let $\left[m_x^c, n_x^c, d_x^c\right]$ be the current background model, the new background model $\left[m_x, n_x, d_x\right]$ is updated as follows:

$$\left[m_x, n_x, d_x\right] = \begin{cases} \left[m_x^b, n_x^b, d_x^b\right] & \text{if} \left(gS_x > k_1 N\right) \\ \left[m_x^f, n_x^f, d_x^f\right] & \text{if} \left(gS_x < k_1 N \text{ and } mS_x < k_2 N\right) \\ \left[m_x^c, n_x^c, d_x^c\right] & \text{otherwise} \end{cases} \tag{39}$$

As presented in [9], k_1 and k_2 are the adjusting parameters which are used to adjust the effect of number N, and they are typically set to 0.8 and 0.1, respectively. Then each pixel is classified using the constructed background model. Given the background model parameters, the pixel x is a foreground pixel if:

$$B_x = \begin{cases} 0(\text{background}) & \left(I_x^t - m_x\right) < k_3 d_x \text{ or } \left(n_x - I_x^t\right) < k_3 d_x \\ 1(\text{foreground}) & \text{otherwise} \end{cases} \tag{40}$$

Here, k_3 is the adjusting parameter to refine the rule of pixel classification, which is set to 2 in this paper. Actually, the computational cost of background subtraction algorithm is low, and the detailed discussion can be found in [9]. It means that background subtraction is practically feasible in WMSNs.

Appendix B

Wavelet transform of a function $f(x)$ can be defined as:

$$WT\left\{f(x);\delta_a,\delta_b\right\} = \int f(x)\psi_{\delta_a,\delta_b}(x)dx \tag{41}$$

where $\psi_{\delta_a,\delta_b}(x)$ defines the family of wavelet functions:

$$\psi_{\delta_a,\delta_b}(x) = \left(1/\sqrt{|\delta_a|}\right)\psi\left(\left(x-\delta_b\right)/\delta_a\right) \tag{42}$$

where δ_a and δ_b are the scale parameters.

With the discrete translation of function $f(x)$ on a dyadic scale $\delta_a = 2^j$ and discrete translation $\delta_b = 2^j k$, the DWT can be presented as follows:

$$DWT\left\{f(x);2^j,2^j k\right\} = C_{j,k} \tag{43}$$

where $C_{j,k}$ is the wavelet coefficients of $f(x)$, j and k denotes the scale level. Thus, $f(x)$ may be obtained as follows:

$$f(x) = \sum_{j\in Z}\sum_{k\in Z} C_{j,k}\psi_{j,k}(x) \tag{44}$$

Acknowledgments

This paper is supported by the National Basic Research Program of China (973 Program) under Grant No. 2006CB303000 and National Natural Science Foundation of China under Grant #60673176, #60373014 and #50175056.

References and Notes

1. Akyildiz, I. F.; Melodia, T.; Chowdhury, K. R.: A survey on wireless multimedia sensor networks. *Computer Networks*, **2007**, *51*, 921-960.
2. Duarte, M. F.; Hu, Y. H.: Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, **2004**, *64*, 826-838.
3. Flouri, K.; Beferull-Lozano, B.; Tsakalides, P.: Training a SVM-based classifier in distributed sensor networks. *Proc. 14nd European Signal Processing Conference*, **2006**, 1-5.
4. Wang, X.; Wang, S.; Bi, D.; Ma, J.: Distributed peer-to-peer target tracking in wireless sensor networks. *Sensors*, **2007**, *7*, 1001-1027.
5. Shilton, A.; Palaniswami, M.; Ralph, D.; Tsoi, A.: Incremental training of support vector machines. *IEEE Trans. Neural Networks*, **2005**, *16*, 114-131.
6. Rüping, S.: Incremental learning with support vector machines. *Proc. IEEE Int. Conf. on Data Mining*, **2001**, 641-642.
7. Joachims, T.: Transductive inference for text classification using support vector machines. *International Conference on Machine Learning*, **1999**, 200-209.
8. Wang, X.; Wang, S.: Collaborative signal processing for target tracking in distributed wireless sensor networks. *Journal of Parallel and Distributed Computing*, **2007**, *67*, 501-515.
9. Haritaoglu, I.; Harwood, D.; Davis, L. S.: W4: real-time surveillance of people and their activities. *IEEE Trans. Pattern Recognition and Machine Intelligence*, **2000**, *22*, 809-830.
10. Sweldens, W.: The lifting scheme: a custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, **1996**, *3*, 186-200.
11. Martina, M.; Masera, G.; Piccinini, G.; Vacca, F.; Zamboni, M.: Embedded IWT evaluation in reconfigurable wireless sensor network. *The 9th International Conference on Electronics, Circuits and Systems*, **2002**, pp. 855-858.
12. Chen, Y.; Wang, G.; Dong, S.: Learning with progressive transductive support vector machine. *Pattern Recognition Letters*, **2003**, *24*, pp. 1845-1855.
13. Liao, D.; Jiang, B.; Wei, X.: Fast learning algorithm with progressive transductive support vector machine. *System Engineering and Electronics*, **2007**, *29*, pp. 88-91.
14. Wang, X.; Wang, S.; Ma, J.: An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment. *Sensors*, **2007**, 7, pp. 354-370.
15. Xu, Y.; Qi. H.: Distributed computing paradigms for collaborative signal and information processing in sensor networks. *Journal of Parallel and Distributed Computing*, **2004**, *64*, pp. 945-959.
16. Rahman, Md. A.; Miah, Md. S.; Gueaieb, W.; Saddik, A.: SENORA: a P2P service-oriented framework for collaborative multirobot sensor networks. *IEEE Sensor Journal*, **2007**, *7*, pp. 658-666.

17. Patwari, N.; Ash, J. N.; Kyperountas, S.; Hero, A. O.; Moses, R. L.; Correal, N. S.: Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, **2005**, *22*, pp. 54-69.
18. Wang, X.; Wang, S.: Peer-to-Peer collaborative signal processing for target tracking in wireless sensor networks. *Fifth International Conference on Grid and Cooperative Computing*, **2006**, pp. 160-163.
19. Kant, K.; Iyer, R.; Tewari, V.: A framework for classifying peer-to-peer technologies. *The 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, **2002**, 1-8.
20. Wu, Q.; Rao, N. S. V.; Barhen, J.; Iyengar, S. S.; Vaishnavi, V. K.; Qi, H.; Chakrabarty, K.: On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE Trans. Knowledge and Data Engineering*, **2004**, *16*, pp. 740-753.
21. Wang, X.; Ma, J.; Wang, S.; Bi, D.: Cluster-based dynamic energy management for collaborative target tracking in wireless sensor networks. *Sensors*, **2007**, *7*, pp. 1193-1215.
22. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, **2006**, *34*, pp. 209-219.
23. Somhom, S.; Modares, A.; Enkawa, T.: Competition-based neural network for the multiple travelling salesmen problem with minmax objective. *Computers & Operations Research*. **1999**, *26*, pp. 395-407.
24. Song, C.: Lee, K.: Lee, W. D.: Extended simulated annealing for augmented TSP and multi-salesmen TSP. *Proceedings of the international joint conference on neural networks*, **2003**, *3*, pp. 2340-2343.
25. Sofge, D.; Schultz, A.; De Jong, K.: Evolutionary computational approaches to solving the multiple traveling salesman problem using a neighborhood attractor schema. *Lecture notes in computer science*, **2002**, 2279, pp. 151-160.
26. Dorigo, M.; Gambardella, L. M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evolutionary Computation*, **1997**, *1*, pp. 53-66.
27. Pan, J.; Wang, D.: An ant colony optimization algorithm for multiple travelling salesman problem. *First International Conference on Innovative Computing, Information and Control*, **2006**, *1*, pp. 210-213.
28. Rosenkrantz, D. J.; Stearns, R. E.; Lewis, P. M.: An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, **1977**, *6*, pp. 563-581.
29. Sim K. M.; Sun W. H.; Ant colony optimization for routing and load-balancing: survey and new directions. *IEEE Trans. on Systems, Man and Cybernetics, Part A*, **2003**, *33*, pp. 560-572.
30. Iyengar, S. S.; Wu, H.-C.; Balakrishnan, N.; Chang, S. Y.: Biologically inspired cooperative routing for wireless mobile sensor networks. *IEEE Systems Journal*, **2007**, *1*, pp. 29-37.

31 Lee S.; Crawford, M. M.: Unsupervised classification for multi-sensor data in remote sensing using Markov random field and maximum entropy method. *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, **1999**, *2*, pp. 1200-1202.

32. Wang, L.; Ji, H.: Gao X.: Clustering based on possibilistic entropy. *Proceedings of 7th International Conference on Signal Processing*, **2004**, *2*, pp. 1467-1470.