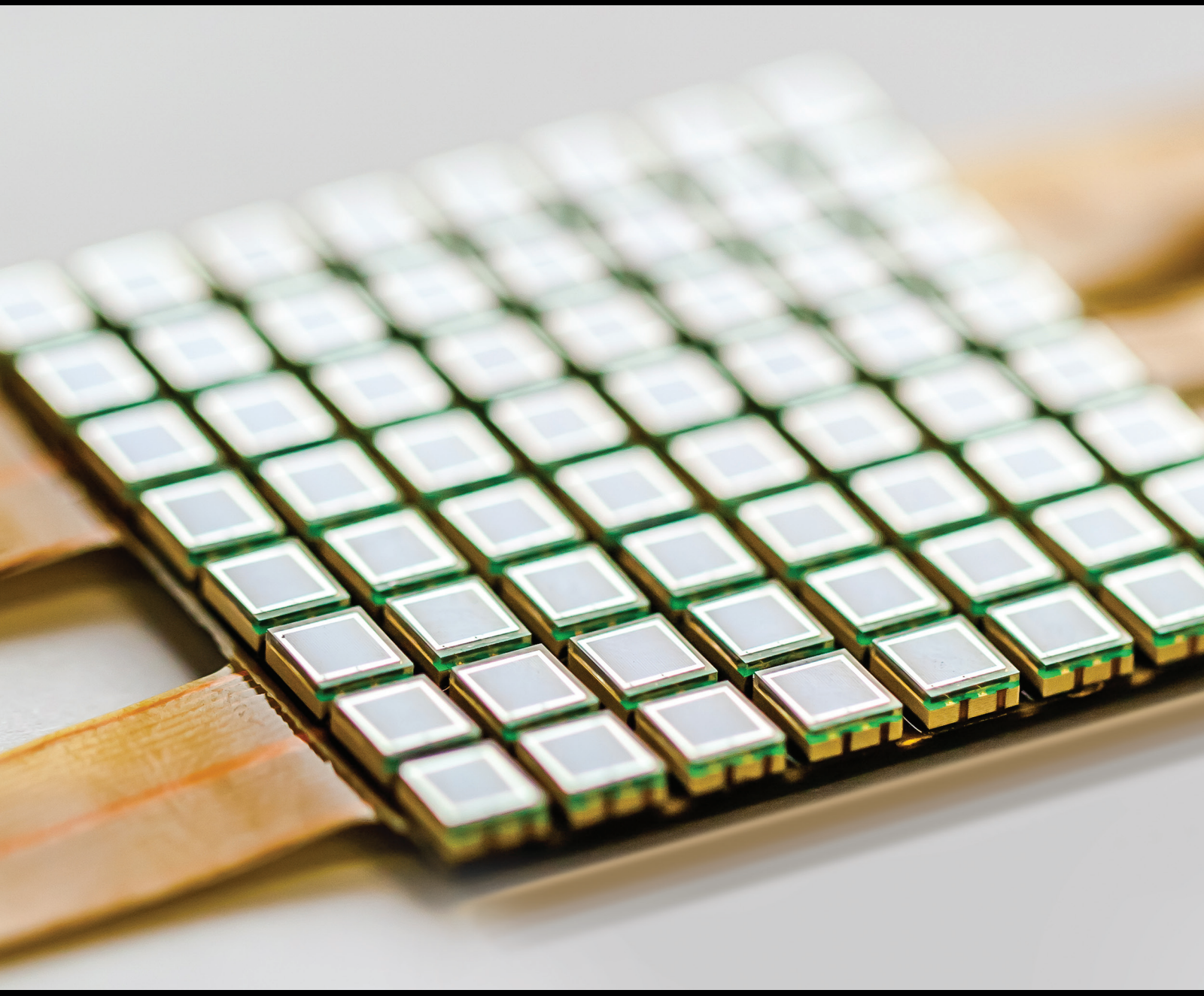
Journal of Sensors

Energy-Efficient Data Fusion Technique and Applications in Wireless Sensor Networks

Guest Editors: Yun Liu, Qing-An Zeng, and Ying-Hong Wang



Energy-Efficient Data Fusion Technique and Applications in Wireless Sensor Networks

Journal of Sensors

Energy-Efficient Data Fusion Technique and Applications in Wireless Sensor Networks

Guest Editors: Yun Liu, Qing-An Zeng, and Ying-Hong Wang

Editorial Board

Contents

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 903981, 2 pages
http://dx.doi.org/10.1155/2015/903981

Editorial

Energy-Efficient Data Fusion Technique and Applications in Wireless Sensor Networks

Yun Liu,[1] Qing-An Zeng,[2] and Ying-Hong Wang[3]

[1]*Department of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China*
[2]*Department of Computer Systems Technology, North Carolina A&T State University, Greensboro, NC 27411, USA*
[3]*Department of Computer Science & Information Engineering, Tamkang University, New Taipei City 25137, Taiwan*

Correspondence should be addressed to Yun Liu; liuyun@bjtu.edu.cn

Received 21 April 2015; Accepted 21 April 2015

Wireless Sensor Networks (WSNs) consist of a large number of source limited wireless sensor nodes for the purpose of data collection, processing, and transmission. Due to the limitations of sensor nodes' capabilities, especially the strictly limited energy, in-network data processing, such as data fusion which can significantly improve the energy-efficiency of the networks, is very important. This special issue mainly focuses on the latest research in the area of energy-efficient data fusion techniques and applications in WSNs. For this special issue, based on the review results we have selected seven papers that address the major issues of energy-efficient data fusion in WSNs and they are summarized as follows.

The paper "Minimum Cost Data Aggregation for Wireless Sensor Networks Computing Functions of Sensed Data" proposes a novel algorithm based on the key insight that there is a tradeoff in reducing costs between the local aggregation cost and finding a low cost path to the sink. The proposed algorithm can always empirically find the best tradeoff point. The authors also argue that the algorithm is applicable to many other similar types of problems. By simulation they show that significant cost savings can be achieved by the proposed algorithm.

The paper "Secure Data Fusion in Wireless Multimedia Sensor Networks via Compressed Sensing" proposes a novel secure data fusion strategy based on compressed image sensing and watermarking; namely, the algorithm exploits the sparsity in the image encryption. The approach relies on $l_1 - norm$ regularization, common in compressive sensing, to enhance the detection of sparsity over wireless multimedia sensor networks. The resulting algorithms endow sensor nodes with learning abilities and allow them to learn the sparse structure from the still image data and also utilize the watermarking approach to achieve authentication mechanism.

The paper "An Improved Ant Colony Routing Algorithm for WSNs" proposes the improved ant colony algorithm to balance the energy consumption of networks. The improved ant colony routing algorithm is inspired by the Dijkstra algorithm changing the wireless sensor network undirected graph to directed graph and also by energy equilibrium consumption ideas. Through simulation and comparison with basic ant colony algorithms, it is obvious that improved algorithm can effectively balance energy consumption and extend the lifetime of WSNs.

The paper "GTDM: A DTN Routing on Noncooperative Game Theory in a City Environment" proposes a fixed sink station routing algorithm based structure and a more proper routing algorithm named Game Theory Based Decision Making (GTDM). GTDM shows decision making process for neighborhood selection and packet delivering strategy which is based on the noncooperative game theory method and city environment characteristics. The performance of GTDM is evaluated using numerical simulations under Working Day Movement (WDM) model and the results suggested that GTDM outperforms other traditional DTNs routing approaches, such as Epidemic and Prophet algorithm.

The paper "Design and Implementation of Energy Efficiency in HVAC Systems Based on Robust PID Control for

Industrial Applications" develops the proportional-integral-derivative (PID) programming which can effectively handle the discrete, nonlinear, and highly constrained optimization problems. Energy efficiency process has been made by controlling of alternative current (AC) drivers for ventilation and exhaust fans, according to supplied air flow capacity and differential air pressure between supplied and exhaust air. Supervisory controller software is developed by using programmable controllers and human machine interface (HMI) units. The new designed HVAC control system would have a saving potential of about 40% as compared to the existing operational settings, without any extra cost.

The paper "A Cluster-Based Fuzzy Fusion Algorithm for Event Detection in Heterogeneous Wireless Sensor Networks" proposes a cluster-based data fusion algorithm for event detection. The authors use k-means algorithm to form the nodes into clusters, which can significantly reduce the energy consumption of intracluster communication. Distances between cluster heads and event and energy of clusters are fuzzified thus to use a fuzzy logic to select the clusters that will participate in data uploading and fusion. Fuzzy logic method is also used by cluster heads for local decision, and then the local decision results are sent to the base station. Decision-level fusion for final decision of event is performed by base station according to the uploaded local decisions and fusion support degree of clusters calculated by fuzzy logic method. The effectiveness of this algorithm is demonstrated by simulation results.

Yun Liu
Qing-An Zeng
Ying-Hong Wang

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 954159, 15 pages
http://dx.doi.org/10.1155/2015/954159

Research Article

Design and Implementation of Energy Efficiency in HVAC Systems Based on Robust PID Control for Industrial Applications

Muharrem Imal

Mechanical Engineering Department, Kahramanmaras Sutcu Imam University, 46100 Kahramanmaras, Turkey

Correspondence should be addressed to Muharrem Imal; muharremimal@ksu.edu.tr

Received 11 October 2014; Revised 16 March 2015; Accepted 23 March 2015

Academic Editor: Qing-An Zeng

Energy efficiency in heating, ventilating, and air-conditioning (HVAC) systems is a primary concern in process projects, since the energy consumption has the highest percentage in HVAC for all processes. Without sacrifice of thermal comfort, to reset the suitable operating parameters, such as the humidity and air temperature, would have energy saving with immediate effect. In this paper, the simulation-optimization approach described the effective energy efficiency for HVAC systems which are used in industrial process. Due to the complex relationship of the HVAC system parameters, it is necessary to suggest optimum settings for different operations in response to the dynamic cooling loads and changing weather conditions during a year. Proportional-integral-derivative (PID) programming was developed which can effectively handle the discrete, nonlinear and highly constrained optimization problems. Energy efficiency process has been made by controlling of alternative current (AC) drivers for ventilation and exhaust fans, according to supplied air flow capacity and differential air pressure between supplied and exhaust air. Supervisory controller software was developed by using programmable controllers and human machine interface (HMI) units. The new designed HVAC control system would have a saving potential of about 40% as compared to the existing operational settings, without any extra cost.

1. Introduction

Several studies have been carried out in the past few years for minimizing the energy consumption related with the control of HVAC systems. HVAC units used in industrial processes have high energy consumptions. It has been estimated that 40% of total energy is consumed in HVAC systems for fabric plants. So, the energy efficiency is the most important role played in HVAC systems. Therefore, a well-designed energy control system improves the energy efficiency [1–6].

In recent years, many researchers have studied dynamic models of HVAC systems and their components using theoretical or experimental approaches for buildings on the ground. However, ground buildings and semiopen built space have become major spaces of use for metropolitan residents [7, 8]. While many control strategies have been suggested in previous reports [9], to the best of our knowledge only a few studies have been conducted on the quantification and comparison of control strategies in process industry for industrial plants.

The HVAC systems are composed of a large number of subsystems; each of them may exhibit nonlinear characteristics. The parameters of the systems change with weather, load, and process occupancy. In many situations, the exact model of the system cannot be obtained, but the approximate model can be derived. These complexities can be eliminated by using a well-designed control technique, operating strategies, and optimum conditions of different HVAC equipment and subsystems.

Modern control methods such as proportional-integral-derivative (PID) controllers are widely used in the process industry; their effectiveness is often limited due to poor tuning; on the other hand they can be easily implemented due to low cost and being reliable in harsh field conditions. The PID

control techniques can be used on model-free and model-based control systems [5–8].

It is very useful to consider the simulation-optimization approach in response to the problem of effective energy efficiency. In the context of the plant services and HVAC systems, a variety of transfer functions have been developed according to the problems, like all year around energy consumption, lifecycle cost, thermal comfort, plant scheduling, design parameters, routing, and distribution. In addition, there are increasing applications of the plant simulation models for decision making purpose, and a multicriterion optimization approach would be adopted for a number of changeable system capacities instead of a single solution which would be provided for decision making.

This paper describes a novel simulation-optimization technique that was applied to devise a reset scheme of air flow for air temperatures and differential air pressure for supply and exhaust air of the HVAC control system in fabric manufacturing plants in industry. The designed control system has been used in a fabric group which is one of the biggest fabric companies in Turkey. The whole system was firstly constructed by LUWA Corporation. The mechanical system was not changed, but the complete control system was removed and the new designed control system was placed instead of old conventional control system. After the designed control system was operated, the energy consumption decreased approximately 40% in the four unit plants. The energy efficiency has been carried out by the controlling of ventilation and exhaust driver inverters depending on differential air pressure between supply and exhaust air and air flow measurement for air changes per second. The controller system consists of the four programmable controller modules, four human machine interface (HMI) units, and computer systems. Every HVAC system has been controlled depending on four parameters: humidifier pump controller for humidification, fresh and bypass damper controllers for temperature control, exhaust fan controllers for stability of differential air pressure, and ventilation fan controllers for air flow capacity. These parameters were controlled by four loop PID controllers implemented in programmable controllers. Supervisory control software was developed to analyze and to control the whole system from a HMI interface. In addition, energy and mass balance equations are applied to derive a dynamic model of single-zone HVAC system and a simulation example has been proposed to investigate different PID control algorithms on the derived mathematical model of the controlled zone. The results from the proposed simulation-optimization technique were used to evaluate the existing operational settings and suggest the possible saving potential from the newly optimized information.

2. Material and Method

2.1. System Structure. The system has such main components as an air conditioned room of fabric plant, the fresh, bypass, and exhaust air dampers, damper actuators, ventilation and exhaust air fans, humidifiers pumps, driver inverters for humidification, temperature, humidity, air flow, and differential air pressure sensors and controllers. Figure 1 shows the general schematic diagram of the single-zone HVAC controlled system for fabric plants. The ideal temperature and humidity levels are generally 25°C and 70%, respectively, in fabric air condition systems.

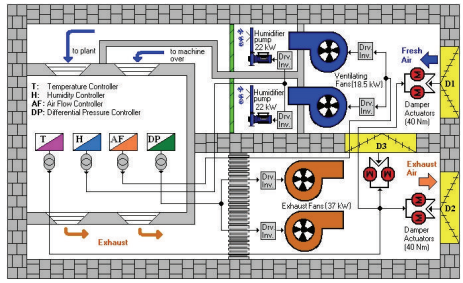


Figure 1: General block diagram of the controlled HVAC system for plants.

Cooling process is performed by utilizing the cold water used for humidification by mixing fresh air supplied from outdoor damper in the fabric air condition systems. Heating processes have been made by circulation of indoor air that generated heat from machines, lights, and people, using bypass dampers. According to desired temperature value, the temperature controller directs the damper actuators which are BELIMO trademarked proportional servomotors having the 40 Nm torque, 2 to 10 VDC control signal, and feedback signals. If the controlled zone temperature is to be raised, the controller commands to the bypass dampers in the direction of opening position, and conversely the fresh and exhaust dampers in direction of closing position. The rate of opening and closing values has to be the same values to equilibrate air pressure between indoor and outdoor.

Humidity controlling has been performed by using humidifier pumps and its driver inverters having the power of 22 kW, spray water flow rate of 78 m^3/h, and nozzle pressure of 2.1 bar. The task of the humidifier pumps is to provide necessary humidity by vaporizing water supplied from water pool. The vaporized water has been mixed by the supplied air from ventilating fans and then it has been carried to indoor space for humidification.

Air flow controller unit controls the desired volume flow rate of supplied air which is determined from the air changes per second for the need of controlled plant. The air flow capacity changes depending on the plant's size, the number of machines, and the type of the plant such as fabric or yarn. Air flow controller has been done by the ventilation fans with the power of 18.5 kW and the capability of 103.000 m^3/h air flow capacity. The desired air flow capacity is determined by the expert operators.

The air pressure difference between indoor and outdoor has to be equilibrated in the product type HVAC systems. The differential pressure controller compensates the air pressure based upon the differential pressure transmitter using the 37 kW exhaust fan drivers. Humidifier pumps, nozzles, fresh air dampers, exhaust air dampers, ventilating, and exhaust

Figure 2: General view of humidification and air supplier units of controlled system.

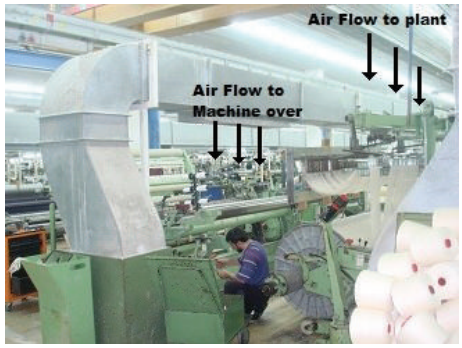


Figure 3: The machine over and the plant air flow channels of the controlled zone.

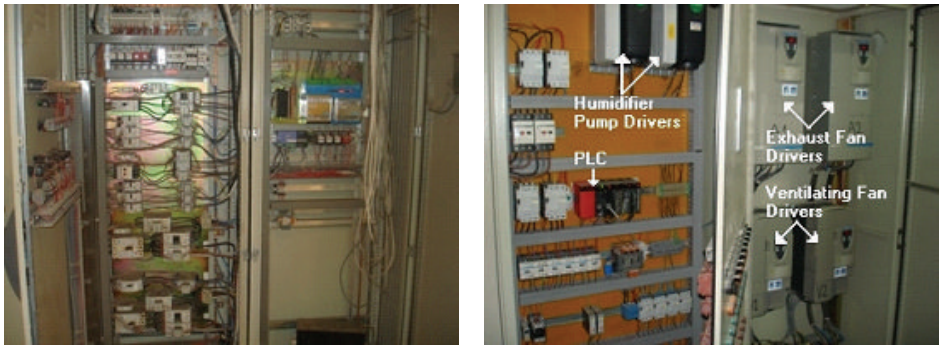
fans of controlled system have been shown in Figure 2. The supplied air has been directed to different air channels as shown in Figure 3. These channels are the machine over and the plant channels, respectively. These channels are controlled by the different controller units as shown in Figure 1.

The formerly designed controller by LUWA incorporation had only temperature and humidity controller. Ventilation and exhaust fans had operated at fixed rotation speed. Energy consumed by the ventilation and exhaust fans corresponds to 40% of total energy spent in a plant. To decrease energy consumption, the driver inverters have been used for speed control of ventilation and exhaust fans drivers according to desired air flow capacity and differential air pressure in the designed control system.

The old conventional and the new designed controller system have been shown in Figure 4 for a single zone of HVAC system.

Four independent controllers were designed for each HVAC unit as shown in Figure 5. These controllers were connected by the Ethernet communication units. The whole system has been controlled by the developed supervisory control software from a single control point as shown in Figure 6. Each unit has an operator panel and the set points can be entered from this panel or HMI unit. The developed supervisory control software is able to perform online data visualization, data storage, data trends, and animation graphs and full control of complete system.

2.2. Dynamic Control Model of HVAC System. Since the HVAC zones have the structure of complex thermal system structure and the actual air-conditioning system is more variable, it is very difficult to determine for its exact model, but the approximate overall dynamic model of a single-zone HVAC system consisting of all its component models can be derived. The whole plant model could reflect the HVAC system with complete dynamic operation of different equipment and subsystems for the cooling demand throughout the yearly operating hours. In the hourly control and operation of different equipment and subsystems, there were four sets of input data files: total and space loads, weather conditions, water temperature, and that specifically for the component model of water-cooled air conditioning systems. The total cooling load was used to determine the water temperature and water pump pressure in operation. In addition, enthalpy control



(a) (b)

Figure 4: (a) The old conventional control system for single zone, (b) the new designed control system for single zone.

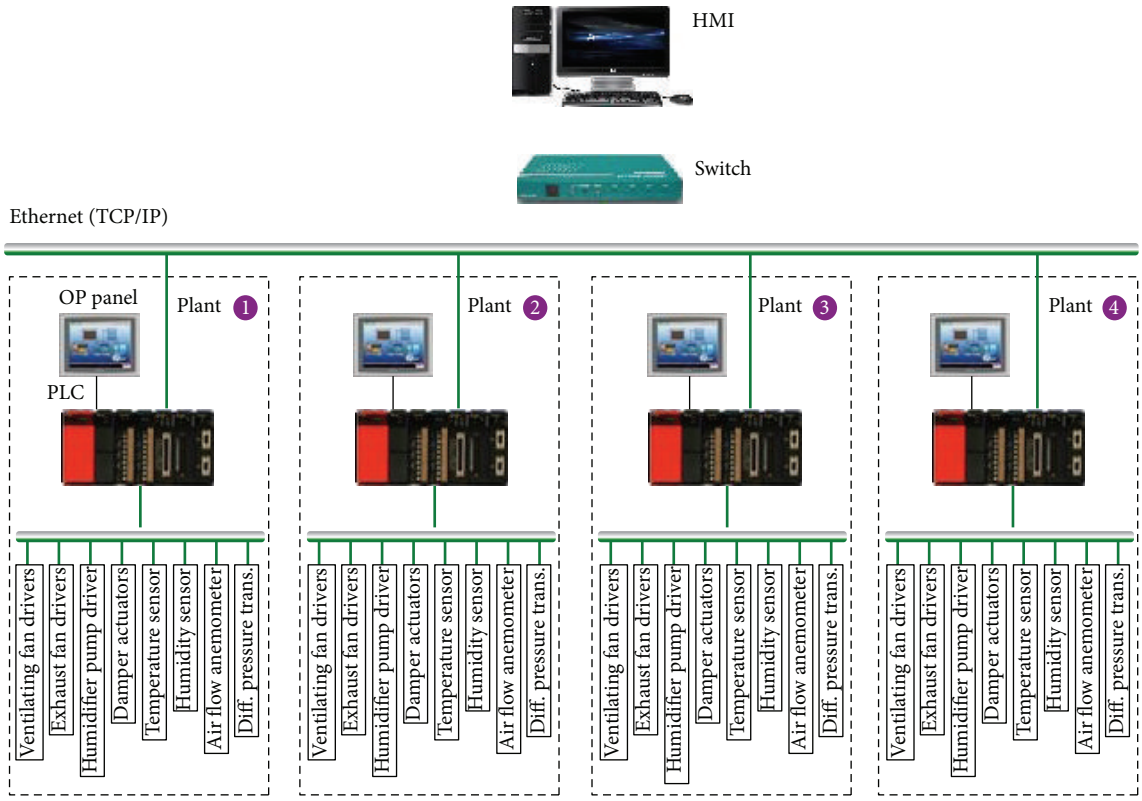


Figure 5: The structure of the designed control system.

was utilized to decide the implementation of free cooling mode, which would call for the free cooling fan in operation, with the same speed of the other operating fans. This dynamic control and operation algorithm was essential to provide the necessary system response to different loading and climatic conditions, although this would increase the degree of the nonlinear, constrained, and discrete nature of the system performance, in particular to the energy efficiency of both the water side and air side systems. The HVAC optimization problem hence requires an effective method that could handle the nonlinear, discrete, and highly constrained characteristics.

2.3. Zone Model. According to thermodynamic energy balance principles, a simple energy equation for a single-zone HVAC system can be written in (1)

$$Q_{\text{input}} + Q_{\text{generation}} = Q_{\text{out}} + Q_{\text{accumulation}} + Q_{\text{consumption}}, \quad (1)$$

where Q_{input} is heat from supply air input, $Q_{\text{generation}}$ is generated heat from people, woven machines, light, and so forth, Q_{out} is heat from exhaust air, $Q_{\text{accumulation}}$ is indoor space heat, and $Q_{\text{consumption}}$ is dissipated heat from indoor to outdoor.

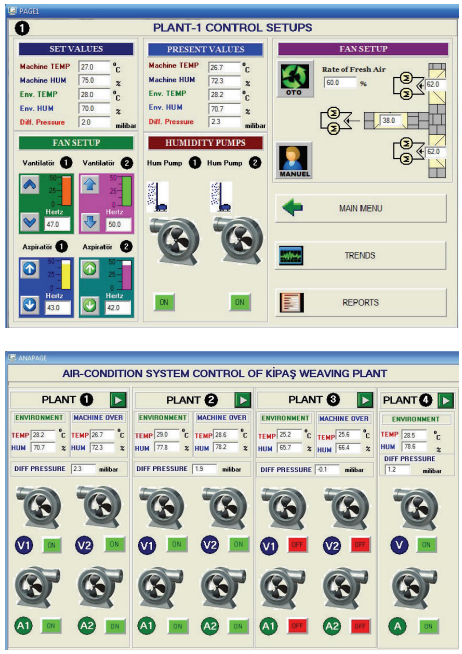


FIGURE 6: The developed program interface for control and analysis from one HMI unit.

The heat equation is derived from the first law of thermodynamics and conservation of energy. Any change in energy ΔQ is proportional to the change in temperature ΔT. That is,

$$\Delta Q = f_a c_p \rho_a \Delta T, \tag{2}$$

where f_a (m^3/s) is volume flow rate of air, c_p (kJ/kg·°C) is the constant pressure specific heat capacity of air, and ρ_a (kg/m^3) is the mass density of air. From the thermodynamic heat equations, the fresh and exhaust heat equations can be written as

$$\begin{aligned} Q_{\text{input}} &= f_{\text{sa}} c_{\text{pa}} \rho_a T_{\text{sa}}, \\ Q_{\text{out}} &= f_{\text{sa}} c_{\text{pa}} \rho_a T_i, \end{aligned} \tag{3}$$

where f_{sa} (m^3/s) is volume flow rate of supplied air:

$$\begin{aligned} Q_{\text{generation}} &= Q_{\text{room}}, \\ Q_{\text{accumulation}} &= \rho_a c_{\text{pa}} v_z \frac{dT_i}{dt}, \\ Q_{\text{consumption}} &= \frac{v_z c_{\text{pa}} \text{ACH}}{3600} (T_i - T_0), \end{aligned} \tag{4}$$

where Q_{room} (W) is heat gains from machines, light, people, and so forth and v_z (m^3) is volume of the zone. ACH is the air changes per hour. There are three modes of consumption heat from indoor to outdoor: conduction, convection, and radiation. The convection heat equation is defined as follows:

$$\begin{aligned} Q_{\text{convection}} &= h \Delta T, \\ Q_{\text{conduction}} &= k \Delta T, \\ Q_{\text{radiation}} &= \varepsilon \sigma (\Delta T)^4, \end{aligned} \tag{5}$$

where h is convection heat transfer coefficient ($W/m^2 \cdot °C$), k is conduction heat transfer coefficient (W/m·°C), ε is emissivity constant, and σ is Boltzmann constant ($W/m^2 \cdot °C$). The heat dissipates into the zone mostly through the conduction heat, so that the convection and radiation heats can be ignored. The temperature zone equation can be written as

$$\begin{aligned} v_z c_{\text{pa}} \rho_a \frac{dT_i}{dt} &= \frac{v_z c_{\text{pa}} \text{ACH}}{3600} (T_0 - T_i) \\ &\quad + f_{\text{sa}} c_{\text{pa}} \rho_a (T_{\text{sa}} - T_i) + Q_{\text{room}}. \end{aligned} \tag{6}$$

Temperature controlling has been performed by the three damper controller units in the fabric HVAC systems which are fresh damper $D1$, exhaust damper $D2$, and return air damper $D3$. Therefore, supply air volume flow rate is composed of mixing the returned air from inner (exhaust air) zone and outdoor air. This mixing equation can be written from mass balance equation

$$m_r c_{\text{pa}} T_i + m_o c_{\text{pa}} T_o = m_s c_{\text{pa}} T_{\text{sa}}, \tag{7}$$

where m_r, m_o, and m_s are mass flow rates of the return, outdoor, and supplied airs, respectively. This condition has to be fulfilled by the controlling of these three damper positions. If the position of $D1$ and $D2$ dampers is opened at the rate of 40%, the position of $D3$ damper must be opened at the rate of 60%. That is, $D1 = D2 = (100 - D3)$. Humidification is a mass transfer process of water vapor to atmospheric air, which results in an increase of water vapor in the mixture. Humidifier model is the same as temperature model and can be derived from mass and energy balance principles as given in [4]

$$C_h \frac{dT_h}{dt} = f_{\text{sa}} c_{\text{pa}} (T_{\text{sa}} - T_h) + \alpha_h (T_o - T_h), \tag{8}$$

where C_h is overall thermal capacitance of the humidifier, T_h is supply air temperature (in humidifier), and α_h is overall transmittance area factor of the humidifier (kJ/s·°C). Similarly, the rate change of moisture content in the zone is equal to the difference between the vapor added to and removed from the zone. Therefore, humidity zone equation can be written in

$$v_{zh} \frac{dW_h}{dt} = f_{\text{sa}} (W_{\text{sa}} - W_h) + \frac{h(t)}{\rho_a}, \tag{9}$$

where v_{zh} is the volume of humidifier, W_h is supply air humidity ratio (in humidifier) in kg/kg (dry air), W_{sa} is humidity ratio of supply air (to the humidifier) in kg/kg (dry air), and $h(t)$ is the rate of moisture air produced in air humidifier.

In order to find the transfer function of the controlled zone, the temperature zone (6)-(7) should be rearranged according the parameters of T_i and T_o:

$$\begin{aligned} v_z c_{\mathrm{pa}} \rho_a \frac{dT_i}{dt} + \left(\frac{v_z c_{\mathrm{pa}} \mathrm{ACH}}{3600} + f_{\mathrm{sa}} c_{\mathrm{pa}} \rho_a \right) T_i \\ = \frac{v_z c_{\mathrm{pa}} \mathrm{ACH}}{3600} T_0 + f_{\mathrm{sa}} c_{\mathrm{pa}} \rho_a T_{\mathrm{sa}} + \mathrm{Q}_{\mathrm{room}}, \end{aligned} \tag{10}$$

$$\begin{aligned} v_z c_{\mathrm{pa}} \rho_a \frac{dT_i}{dt} + \left(\frac{v_z c_{\mathrm{pa}} \mathrm{ACH}}{3600} + f_{\mathrm{sa}} c_{\mathrm{pa}} \rho_a \right) T_i \\ = \frac{v_z c_{\mathrm{pa}} \mathrm{ACH}}{3600} T_0 + f_{\mathrm{sa}} c_{\mathrm{pa}} \rho_a \left(\frac{m_r T_i + m_o T_0}{m_s} \right) \\ + \mathrm{Q}_{\mathrm{room}}. \end{aligned} \tag{11}$$

In order to simplify (11), τ and k can be defined as follows:

$$\begin{aligned} \tau &= \frac{3600 v_z \rho_a m_s}{m_s \left(v_z \mathrm{ACH} + 3600 f_{\mathrm{sa}} \rho_a \right) - 3600 f_{\mathrm{sa}} \rho_a m_r}, \\ k &= \frac{m_s v_z \mathrm{ACH} + 3600 f_{\mathrm{sa}} \rho_a m_o}{m_s \left(v_z \mathrm{ACH} + 3600 f_{\mathrm{sa}} \rho_a \right) - 3600 f_{\mathrm{sa}} \rho_a m_r}. \end{aligned} \tag{12}$$

Finally, temperature zone (6) has been simplified as

$$\tau \frac{dT_i}{dt} + T_i = k T_o + \mathrm{Q}_{\mathrm{room}}. \tag{13}$$

$\mathrm{Q}_{\mathrm{room}}$ is heat gains from machines, lights, people, and so forth and is a constant. The Laplace transform of (13) is expressed as

$$\tau s T_i(s) - T_i(0) + T_i(s) = k T_o(s). \tag{14}$$

Because of the distance between supply air and the measurement point of temperature, there is pure delay time (L) in the process. The plant heat transfer function is expressed as follows:

$$G_p(s) = \frac{T_i(s)}{T_o(s)} = \frac{k e^{-Ls}}{\tau s + 1}. \tag{15}$$

The system is considered as a first-order plus dead time (FOPDT) structure as shown in (15), where k (°C·s/kg) is the gain coefficient, τ is the time constant, and L is the time delay of the controlled HVAC system.

2.4. Proportional-Integral-Derivative (PID) Control. The proportional-integral-derivative (PID) method is used to control the air flow and differential pressure by manipulating the fan speed of the ventilation system. PID controllers are widely employed in feedback control loops in most industrial process due to their simple structure, robustness, easy implementation, and good performance [10–15]. As indicated by the name, PID controllers consist of a proportional part, an integral part, and a derivative part as follows [16].

Proportional terms are

$$U_p(t) = k_c \left(y_s(t) - y(t) \right). \tag{16}$$

Integral terms are

$$U_I(t) = \frac{k_c}{\tau_I} \int_o^t \left(y_s(\tau) - y(\tau) \right) dt. \tag{17}$$

Derivative terms are

$$U_D(t) = k_c \tau_d \frac{d\left(y_s(t) - y(t) \right)}{dt}, \tag{18}$$

where $y_s(t)$ is the set point of the process output, $y(t)$ is the process output, and $U_P(t)$, $U_I(t)$, and $U_D(t)$ are the controller outputs of the proportional, integral, and derivative parts, respectively. The constants k_c, τ_I, and τ_d in (18) are the proportional gain, integral time, and derivative time, respectively. The total output of the PID controller is a summation of the three individual outputs in (16)–(18); that is,

$$\begin{aligned} U(t) &= U_p(t) + U_I(t) + U_D(t) \\ &= k_c \left(y_s(t) - y(t) \right) \\ &\quad + \frac{k_c}{\tau_I} \int_o^t \left(y_s(\tau) - y(\tau) \right) dt \\ &\quad + k_c \tau_d \frac{d\left(y_s(t) - y(t) \right)}{dt}. \end{aligned} \tag{19}$$

The three constants are the proportional gain, integral time, and derivative time, respectively.

k_c, τ_I, and τ_d are usually called the parameters of the PID controller and should be determined under the dynamic condition of the process. So, the parameters for the improvement of the control performance have to be tuned.

2.5. Parameter Tuning. Up to now, numerous tuning methods have been developed, including the Ziegler-Nichols (Z-N) method, internal model control (IMC), and the integral of the time-weighted absolute value of the error tuning rule for a first-order plus time delay model (ITAE). In this paper, all three methods were implemented to tune the controller in the energy efficient HVAC control system. If the model of the system can be obtained, it is easy to design the PID controller for it.

The gain coefficients k_p and T_i can be achieved model based algorithms such as Ziegler-Nichols (Z-N) or variant of Z-N tuning rules (Chien-Hrones-Reswick (CHR), Cohen-Coon), Wang-Juang-Chan, Zhuang-Atherton optimum PID controller design, and internal model control (IMC) [7–9, 16–19].

The response of an FOPDT model is shown in Figure 7. A large number of systems can be approximately modeled by (15). If the step response of the system can be experimentally measured, the parameters of the k, L, T, and a can be determined as shown in Figure 7(a). In the same way, if the frequency response of the system can be experimentally observed, the crossover frequency ω_c and critical gain k_c can be obtained from the Nyquist plot as shown in Figure 7(b), where $a = KL/T$ and $T_c = 2\pi/\omega_c$.

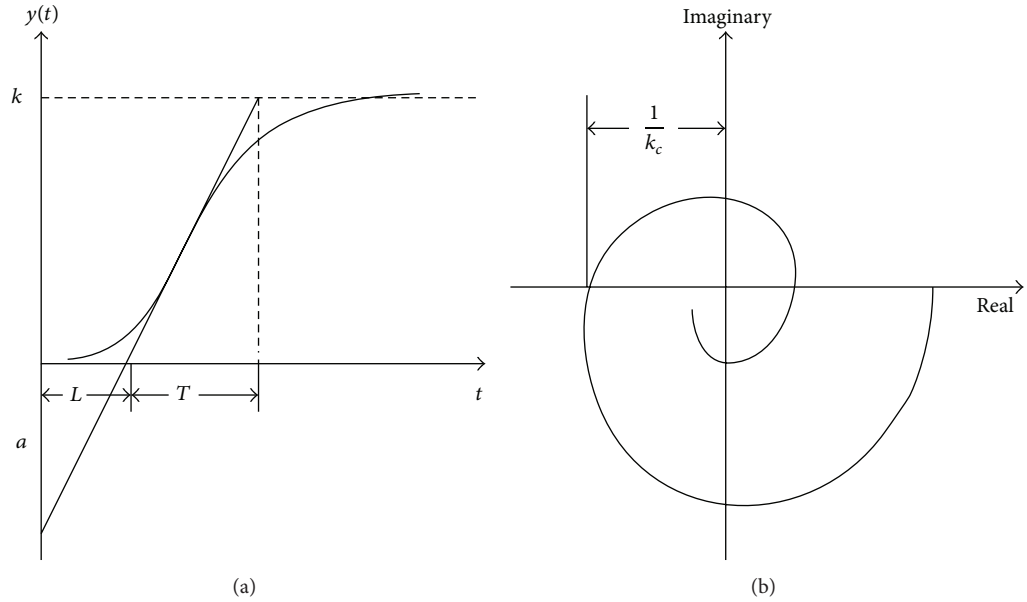


Figure 7: (a) The time response of FOPDT system, (b) the Nyquist plot of FOPDT system.

Table 1: Z-N tuning formulas.

Controller type	Step response			Frequency response		
	k_p	T_i	T_d	k_p	T_i	T_d
P	$1/a$			$0.5k_c$		
PI	$0.9/a$	$L/0.3$		$0.45k_c$	$0.83T_c$	
PID	$1.2/a$	$2L$	$L/2$	$0.6k_c$	$0.5T_c$	$0.125T_c$

Table 2: CHR method tuning formulas.

Controller type	Set point regulation			Disturbance rejection		
	k_p	T_i	T_d	k_p	T_i	T_d
P	$0.3/a$			$0.3/a$		
PI	$0.35/a$	$1.2T$		$0.6/a$	$4L$	
PID	$0.6/a$	T	$0.5L$	$0.95/a$	$2.4L$	$0.42L$

The critical gain k_c at the crossover frequency of ω_c is the first intersection of a Nyquist plot with the negative part of the real axis as shown in Figure 7(b):

$$\frac{k\left(\cos\omega_c L_i - \omega_c T \sin\omega_{cL}\right)}{1+\omega_c{}^2T^2} = -\frac{1}{k_c}. \tag{20}$$

The Z-N tuning method is a very useful empirical tuning formula. The PID controller parameters in Z-N tuning algorithms can be found from Table 1.

The Chien-Hrones-Reswick (CHR) method emphasizes the set point regulation or disturbance rejection. The CHR PID controller tuning formulas are summarized in Table 2 with the condition of no overshoot.

Another Z-N type tuning algorithm is the Cohen-Coon tuning formula. In this algorithm time constant τ is calculated as $\tau = L/(L+T)$ and PID parameters are tuned from Table 3.

Wang-Juang-Chan tuning method is based on an optimum ITAE (the integral of time multiplied by absolute error) criterion. If the k, L, and T parameters of the plant model are known, the controller parameters are given by

$$k_p = \frac{(0.7303 + (0.5307T/L))\,(T+0.5L)}{k\,(L+T)},$$

$$T_i = T + 0.5L,$$

Table 3: Cohen-Coon method tuning formulas.

Cont. type	Set point regulation		
	k_p	T_i	T_d
P	$\frac{1}{a}\left(1+\frac{0.35\tau}{1-\tau}\right)$		
PI	$\frac{0.9}{a}\left(1+\frac{0.92\tau}{1-\tau}\right)$	$\frac{3.3-3\tau}{1+1.2\tau}L$	
PID	$\frac{1.35}{a}\left(1+\frac{0.18\tau}{1-\tau}\right)$	$\frac{2.5-2\tau}{1-0.39\tau}L$	$\frac{0.37-0.37\tau}{1-0.81\tau}L$

Table 4: Zhuang-Atherton method optimum tuning formulas.

L/T		0.1–1			1.1–2	
Criterion	ISE	ISTE	IST^{2}E	ISE	ISTE	IST^{2}E
a_1	1.048	1.042	0.968	1.154	1.142	1.061
b_1	−0.897	−0.897	−0.904	−0.567	−0.579	−0.583
a_2	1.195	0.987	0.977	1.047	0.919	0.892
b_2	−0.368	−0.238	−0.253	−0.220	−0.172	−0.165
a_3	0.489	0.385	0.316	0.490	0.384	0.315
b_3	0.888	0.906	0.892	0.708	0.839	0.832

$$T_d = \frac{0.5LT}{T + 0.5L}. \tag{21}$$

Zhuang and Atherton proposed the optimum PID controller design for various criteria. This method uses the integrated square error (ISE) function

$$J_n(\theta) = \int_0^\infty \left[t^n e(\theta, t)\right]^2 dt, \tag{22}$$

where $e(\theta, t) \triangleq r(t) - y(t, \theta)$ is the error signal of the closed loop system and $\theta \triangleq [k_p, T_i, T_d]^T$ is a vector containing the controller parameters. For the optimum PID controller parameters can be represented as

$$k_p = \frac{a_1}{k}\left(\frac{L}{T}\right)^{b_1},$$
$$T_i = \frac{T}{a_2 + b_2(L/T)}, \tag{23}$$
$$T_d = a_3 T\left(\frac{L}{T}\right)^{b_3},$$

where (a, b) pairs are found from Table 4 according to the integral squared error (ISE) criterion, integral squared time-weighted error (ISTE) criterion, and the integral squared time-squared weighted error (IST^{2}E) criterion.

The internal model control (IMC) algorithm is based on the fact that an accurate model of the process can lead to the design of a robust controller both in terms of stability and performance.

The first step in using the IMC (lambda) tuning correlations is to compute the closed loop time constant τ_c. The closed loop time constant τ_c is the single adjustable IMC tuning parameter that allows the desired levels of performance and robust stability to be specified. IMC tuning PID parameters are chosen as the following equations:

$$k_p = \frac{L + 2T}{2k(\tau_c + L)},$$
$$T_i = T + \frac{L}{2}, \tag{24}$$
$$T_d = \frac{TL}{2T + L},$$

where T_c is a design parameter that affects the tradeoff between performance and robustness.

3. Simulation Method

The objectives of this study are to assess the energy efficient control of HVAC system and to determine an optimal control strategy and its tuning parameters in order to maintain an efficient temperature, humidity, and pressure condition in an industrial fabric space. The first step, the temperature, humidity, and pressure control system is implemented using Matlab/Simulink. In the second step, four temperature control structures of the system are designed Z-N based tuning algorithm model. The Z-N based tuning algorithm model establishes the kinetics of a process using a mathematical expression. One such expression is a transfer function that is the ratio of the Laplace transform of the process output and process input.

Totally four of the plants having the area of 22.000 m^3 have been controlled by using PI controller. The parameters of the biggest zone are as follows.

The volume of the zone is $v_z = 60 * 20.5 * 4.5 = 5535\ \text{m}^3$, the volume flow rate of supplied air is $f_{\text{sa}} = 18.05\ \text{m}^3/\text{s}$, the air density is $\rho_a = 1.25\ \text{kg/m}^3$, the air specific heat is $c_{\text{pa}} = 1.00\ \text{kJ/kg}\cdot^\circ\text{C}$, air change per hour is ACH = 9, mass flow rates of the return air is $m_r = 8.66\ \text{kg/s}$, mass flow rates of the outdoor air is $m_o = 5.78\ \text{kg/s}$, and mass flow rates of the supplied air is $m_s = 14.44\ \text{kg/s}$. Substituting these parameters to the calculation of τ and k constants, the transfer function of the plant is determined as follows:

$$G_p(s) = \frac{1e^{-20s}}{300s + 1}. \tag{25}$$

PI controller has been applied to the system having the feed forward transfer function of (25) as shown in Figure 8:

$$G(s) = G_p(s) \cdot G_c(s) = K_p\left(1 + \frac{1}{T_i s}\right) \cdot \frac{1e^{-20s}}{300s + 1}. \tag{26}$$

Figure 9 shows the simulation of four different PID tuning methods for the unit step responses according to PID parameters given in Table 5.

Z-N and Cohen-Coon method has fast response time, but approximately 20% overshoot has occurred as shown in Figure 9. Wang-Juang-Chan and IMC tuning method have not seen overshoot, but rise time is more slowly comparing to other methods. It is well known that, the rise time depends on the proportional constant but overshoot depends on the integral time constant in PID controller.

Table 5: Different types of PI controller parameters.

Tuning methods	k_p	T_i
Z-N	13.5	66.6
Cohen-Coon	14.3	57.9
Wang-Juang-Chan	8.4	310
IMC	7.5	310

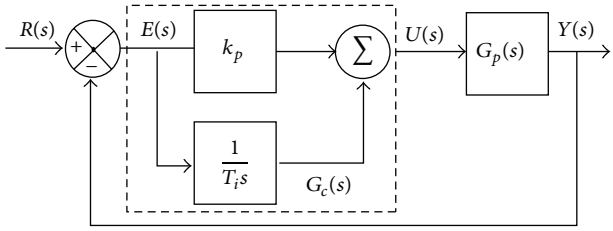


Figure 8: PI controller diagram for controlled parameters.

3.1. Experimental Results of the Real System Controller. Modern process control problems in the process industries are dominated by nonlinear, time-varying behavior, disturbances, and uncertainties. However, more than 90% of plants are controlled by the well-established PID controllers in industrial automation and process until today [8, 9]. The control block diagram of the single-zone HVAC system is shown in Figure 8. Four types of PI controllers were designed to control temperature, humidity, air flow, and differential air pressure. Response of the control system depends on the outdoor temperature and humidity. If the outdoor temperature is high, decreasing the temperature of the zone will be very slow and limited. If the outdoor temperature is low, decreasing the temperature is very quick. Humidity controller is altered by temperature controller. If the system operates by the returned air, the increase in humidity becomes quickly. If the system runs at the fresh air position the humidity increment becomes slowly.

Totally 22 PI controllers were developed in a single programmable controller. These controller parameters were adjusted by using Z-N tuning method. A software program was developed for quickly tuning the PID parameters of the controllers. The Z-N based tuning algorithm of the developed software is shown in Figure 10. The developed software is able to communicate with the programmable controller, selectable of control type (auto/manual control), and able to store control output data into database system and perform online analysis graphically as shown in Figure 11.

The complete HVAC system can be controlled by a HMI unit. An operator can enter the temperature, humidity, air flow, and differential air pressure set points values from the software. After the set points are determined by the operators, the controllers hold the system parameters at the desired set point values. The following experiments have been carried out to observe the performance of the controlled system.

As stated early, the responses of the controller depend on the outdoor climatic conditions. While the outdoor temperature was 21°C, the system temperature increased from 28°C to 31°C in 12 minutes at the heating position. At same condition, the temperature decreased from 31°C to 28°C in only 5 minutes at the cooling position. The heating process was performed by return air circulation, so the heating process happened more slowly. The cooling process was performed by the fresh air. Therefore, the system response depends on the outdoor temperature as shown in Figure 12.

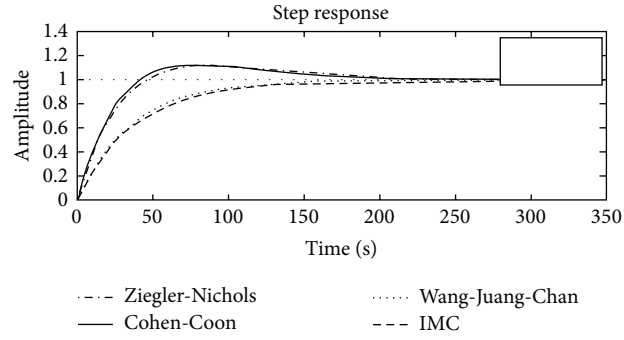


Figure 9: Comparison of four PID controller methods.

If the outdoor temperature is low, then the cooling process takes place as quickly; on the other side, if the outdoor temperature is high, then the cooling process is more slowly. If the set point of the temperature is determined as 32°C while the outdoor temperature is 34°C, then the cooling processes take place very slowly and heating processes are more rapid as shown in Figure 12.

The cooling process utilizes the temperature of sprayed water by mixing the supplied air from outdoor. Humidity controller has been affected from the outdoor humidity ratio. But indeed humidification performance depends generally on the capacity of pumps, the structure of control systems, and units. Technical specification of the humidifier pumps in designed system is: the flow rate is 78 m^3/h, pump head rate is 3 bar, spray water flow rate is 78 m^3/h, nozzle pressure is 2.1 bar, maximum rotating speed is 3000 rpm, and nozzle diameter of each unit is 6 mm.

Two experiments were done to test the performance of the humidity controller. The first experiment has been performed at the condition of 36% outdoor humidity ratios. The desired set point humidity has been obtained in 15 minutes. The humidity of the zone rose from 62% to 70% in this time interval. Decreasing the humidity again from 70% to 62% was carried out in 10 minutes as shown in Figure 13. The second experiment has been done at the 28% outdoor humidity ratio. The desired set point value has been reached in 18 minutes and the old humidity ratio has been reached in 8 minutes. Eventually, the humidity ratio is affected by outdoor humidity ratio as shown in Figure 13.

Differential pressure controller has been carried out by setting the speed of 37 kW exhaust fans according to measured value from differential air pressure transmitter. The differential pressure transmitter measures the difference between indoor and outdoor air pressure. Figure 14 shows the air pressure deviation measured from differential air pressure

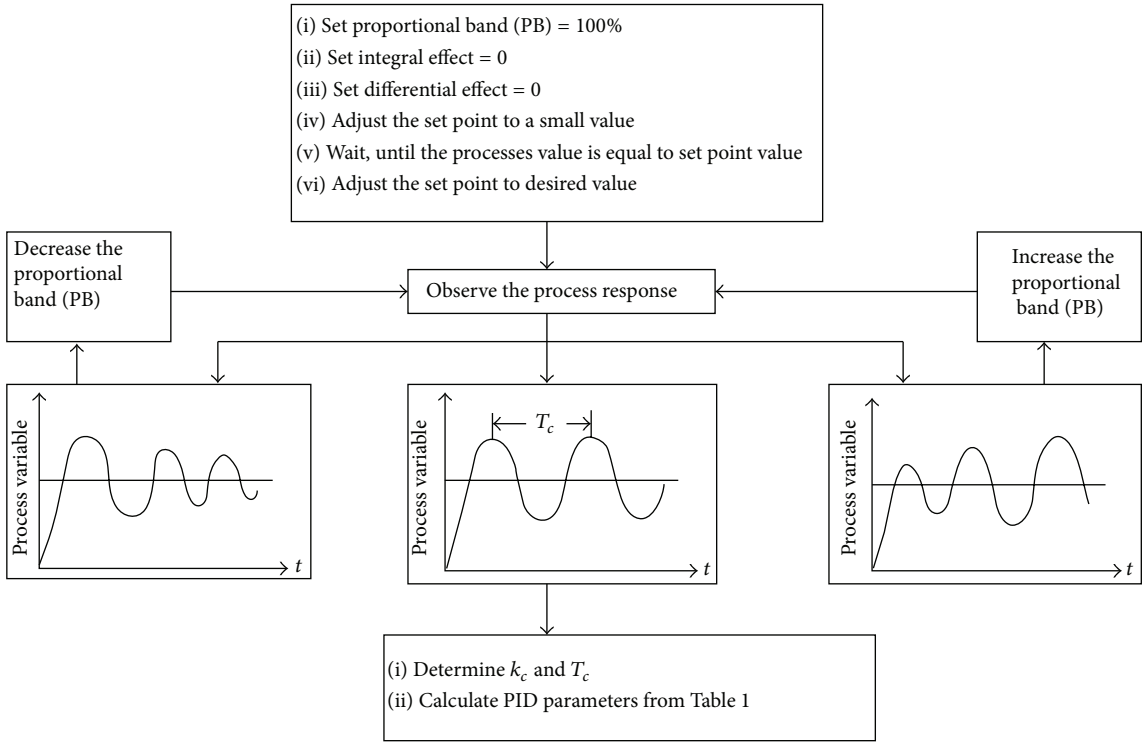


Figure 10: Z-N tuning based PID controller algorithm.

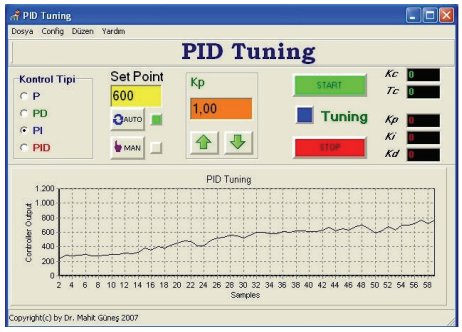


Figure 11: The developed Z-N tuning based PID tuning program.

transmitter during nine hours. The difference of the air pressure results from the event of clogging of exhaust filter and the difference air flow capacity between supply air from ventilation fans and exhaust air from exhaust fans. If the exhaust filter is clogged, the differential pressure transmitter warns operators and then the operator solves the filter problem and the filter clogging process does not affect differential pressure controller for a long time. That is, the differential pressure controller generally operates according to difference between air capacity of supply air and exhaust air.

Air flow capacity controller operates depending on the requirement of air changes of plant per second (m^3/s). The air supplies using ventilation fans by adjusting its rotational speeds. Air flow capacity controller has the fast responses as shown in Figure 15. The desired maximum air capacity has been reached during only 10 seconds.

3.2. Comparison of Real and Simulation Systems. The unit step graphical responses of the derived approximate transfer function of the system were shown in Figure 9 for temperature controller. The designed real system temperature controller responses were shown in Figure 12. The desired level of the unit exchange of temperature has been obtained approximately within 180 seconds depending on the outdoor temperature by the designed controller. Model-based controller reached the same unit temperature level within 150–200 seconds depending on the PID parameters as shown in Figure 9. So, the results of the derived model and the real application system are approximately corresponding.

4. Energy Saved in the Systems

In the old conventional control system, exhaust and ventilating fans had been driven at constant speed. The differential pressure sensors, anemometers, and driver inverters for exhaust and ventilating fans have been added to the new designed control system. In this manner, the exhaust and ventilating fans' speed have been controlled depending on the requirements of the plant conditions. The expert operator only enters the desired parameters by the operator panel on the supervisory control system. Before the design of control system, the overall energy consumption was measured by using energy analyzer to determine the rate of energy efficiency of the designed four HVAC systems through Tables 6 to 9 which shows the amount of energy efficiency data for each controlled plant. Table 6 shows the energy consumptions for plant 1, before and after the controller design. The

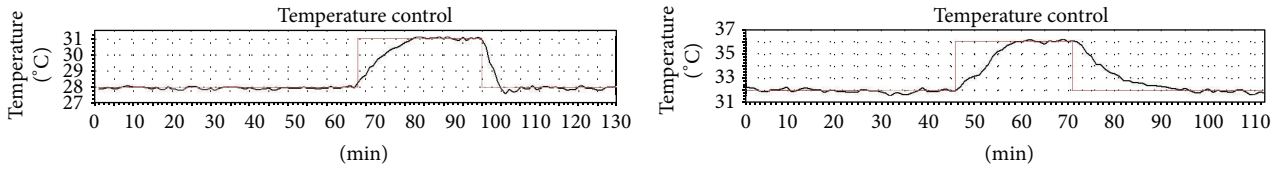


Figure 12: The temperature controller response of the outdoor condition at 21°C (a) and the temperature controller response of the outdoor condition at 34°C (b).

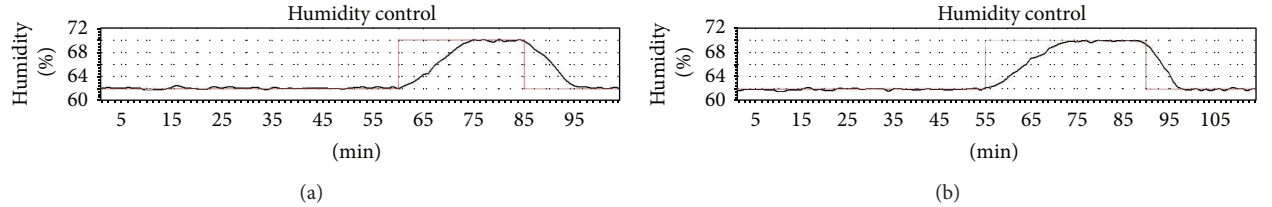


Figure 13: The humidity controller response of the outdoor condition at 36% humidity (a) and the humidity controller response of the outdoor condition at 28% humidity (b).

Table 6: Energy saving in plant 1.

Power consumption in old conventional system			Power consumption after the designed control system		
Date	Hour	kW	Date	Hour	kW
14.11.2013	10:30:00	108.21	05.01.2014	19:00:00	59.08
14.11.2013	11:00:00	109.53	05.01.2014	19:15:00	67.05
14.11.2013	11:30:00	108.87	05.01.2014	19:30:00	65.72
14.11.2013	12:00:00	109.2	05.01.2014	19:45:00	65.06
14.11.2013	12:30:00	108.87	05.01.2014	20:00:00	60.74
14.11.2013	13:00:00	108.21	05.01.2014	20:15:00	65.72
14.11.2013	13:30:00	108.87	05.01.2014	20:30:00	65.72
14.11.2013	14:00:00	108.54	05.01.2014	20:45:00	65.39
14.11.2013	14:30:00	108.87	05.01.2014	21:00:00	65.72
14.11.2013	15:00:00	108.54	05.01.2014	21:15:00	65.06
14.11.2013	15:30:00	108.54	05.01.2014	21:30:00	67.05
14.11.2013	16:00:00	109.2	05.01.2014	21:45:00	66.05
14.11.2013	16:30:00	108.87	05.01.2014	22:00:00	63.06
14.11.2013	17:00:00	109.53	05.01.2014	22:15:00	65.72
14.11.2013	17:30:00	108.87	05.01.2014	22:30:00	65.39
14.11.2013	18:00:00	109.53	05.01.2014	22:45:00	66.38
14.11.2013	18:30:00	109.2	05.01.2014	23:00:00	67.05
14.11.2013	19:00:00	105.55	05.01.2014	23:15:00	66.38
14.11.2013	19:30:00	105.88	05.01.2014	23:30:00	65.39
14.11.2013	20:00:00	106.88	05.01.2014	23:45:00	66.05
14.11.2013	20:30:00	109.2	05.01.2014	00:00:00	66.05
14.11.2013	21:00:00	108.87	05.01.2014	00:15:00	65.39
14.11.2013	21:30:00	108.87	05.01.2014	00:30:00	65.72
Average		**108.55**	Average		**65.26**

Average power consumption: old = 108.55, new = 65.26, and profit = 39.88%.

Table 7: Energy saving in plant 2.

Power consumption in old conventional system			Power consumption after the designed control system		
Date	Hour	kW	Date	Hour	kW
13.07.2013	10:00:00	107.7	06.01.2014	10:30:00	79.99
13.07.2013	11:00:00	106.7	06.01.2014	10:45:00	80.32
13.07.2013	12:00:00	104.2	06.01.2014	11:00:00	79.66
13.07.2013	13:00:00	107.5	06.01.2014	11:15:00	82.65
13.07.2013	14:00:00	107.4	06.01.2014	11:30:00	76.67
13.07.2013	15:00:00	107,7	06.01.2014	11:45:00	75.35
13.07.2013	16:00:00	108.3	06.01.2014	12:00:00	75.01
13.07.2013	17:00:00	108	06.01.2014	12:15:00	71.36
13.07.2013	18:00:00	108.4	06.01.2014	12:30:00	72.03
13.07.2013	19:00:00	108.2	06.01.2014	12:45:00	71.69
13.07.2013	20:00:00	107.9	06.01.2014	13:00:00	71.69
13.07.2013	21:00:00	107.9	06.01.2014	13:15:00	72.03
13.07.2013	22:00:00	108.4	06.01.2014	13:30:00	74.02
13.07.2013	23:00:00	107.8	06.01.2014	13:45:00	74.68
13.07.2013	00:00:00	107.9	06.01.2014	14:00:00	74.68
13.07.2013	01:00:00	107.2	06.01.2014	14:15:00	75.68
13.07.2013	02:00:00	106.7	06.01.2014	14:30:00	75.01
13.07.2013	03:00:00	106.3	06.01.2014	14:45:00	76.34
13.07.2013	04:00:00	106.5	06.01.2014	15:00:00	74.68
13.07.2013	05:00:00	106.8	06.01.2014	15:15:00	75.35
13.07.2013	06:00:00	106.8	06.01.2014	15:30:00	75.35
13.07.2013	07:00:00	106.2	06.01.2014	15:45:00	75.01
13.07.2013	08:00:00	106.2	06.01.2014	16:00:00	75.01
Average		**107.25**	Average		**75.4**

Average power consumption: old = 107.25, new = 75.40, and profit = 29.69%.

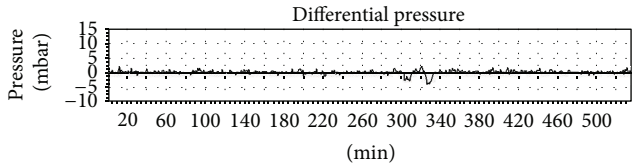


Figure 14: The differential air pressure of supply and exhaust air in the period of 9 hours.

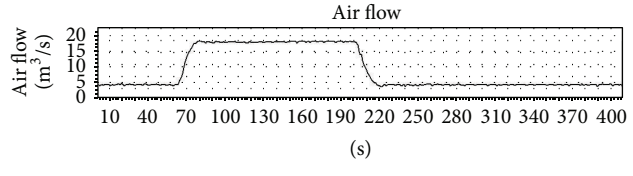


Figure 15: The air flow capacity controller response.

energy consumption in old conventional control system for plant 1 was measured as the average of 108 kW. For the energy consumption in the new designed system, this value was decreased to average of 65 kW. The energy efficiency has been obtained as 43 kW per hour for plant 1. The energy efficiency of 32 kW has been obtained for plant 2, 41 kW in plant 3, and 24 kW in plant 4, respectively. The money spent for plant 1, plant 2, and plant 3 was approximately 39450 USD and 8331 USD for plant 4. The total energy efficiency has been measured as 140 kW. The total cost of 47781 USD has been spent of the whole system.

5. Conclusion

This paper describes a novel simulation-optimization technique that was applied to devise a reset scheme of air flow for air temperatures and differential air pressure for supply and exhaust air of the HVAC control system in fabric manufacturing plants in industry. Due to the complex relationship of the HVAC system parameters, it is necessary to suggest optimum settings for different operations in response to the dynamic cooling loads and changing weather conditions during a year. Proportional-integral-derivative (PID) programming was developed which can effectively handle the discrete, nonlinear, and highly constrained optimization problems. To solve this problem, a HVAC control system has been successfully designed and the dynamic model of the controlled zone has been derived accurately for an integrated big plant in Turkey.

Table 8: Energy saving in plant 3.

Power consumption in old conventional system			Power consumption after the designed control system		
Date	Hour	kW	Date	Hour	kW
16.11.2013	10:30:00	85.64	06.01.2014	18:00:00	43.81
16.11.2013	11:00:00	85.3	06.01.2014	18:30:00	43.81
16.11.2013	11:30:00	85.64	06.01.2014	19:00:00	43.48
16.11.2013	12:00:00	84.97	06.01.2014	19:30:00	42.82
16.11.2013	12:30:00	84.97	06.01.2014	20:00:00	43.15
16.11.2013	13:00:00	84.97	06.01.2014	20:30:00	48.13
16.11.2013	13:30:00	84.97	06.01.2014	21:00:00	43.48
16.11.2013	14:00:00	83.64	06.01.2014	21:30:00	43.81
16.11.2013	14:30:00	84.31	06.01.2014	22:00:00	40.83
16.11.2013	15:00:00	83.98	06.01.2014	22:30:00	42.15
16.11.2013	15:30:00	83.64	06.01.2014	23:00:00	49.12
16.11.2013	16:00:00	83.64	06.01.2014	23:30:00	43.81
16.11.2013	16:30:00	83.64	06.01.2014	00:00:00	43.15
16.11.2013	17:00:00	83.64	06.01.2014	00:30:00	44.81
16.11.2013	17:30:00	83.64	06.01.2014	01:00:00	39.5
16.11.2013	18:00:00	83.64	06.01.2014	01:30:00	43.81
16.11.2013	18:30:00	83.31	06.01.2014	02:00:00	39.5
16.11.2013	19:00:00	83.98	06.01.2014	02:30:00	44.15
16.11.2013	19:30:00	83.64	06.01.2014	03:00:00	36.51
16.11.2013	20:00:00	83.64	06.01.2014	03:30:00	38.83
16.11.2013	20:30:00	83.98	06.01.2014	04:00:00	41.16
16.11.2013	21:00:00	84.31	06.01.2014	04:30:00	42.15
16.11.2009	21:30:00	83.98	06.01.2014	05:00:00	38.83
Average		**84.22**	Average		**42.64**

Average power consumption: old = 84.22, new = 42.64, and profit = 49.37%.

The simulation and the real controller responses have been observed, respectively. When the results have been compared, either of the system responses has been obtained similarly. Supervisory control software was developed to control overall system using a host computer and programmable controllers. All of the control units could communicate with the host computer by the MODBUS TCP/IP Ethernet protocol. Thus, the system parameters could be observed and controlled on the single central point by the host computer and an operator.

The approximately 40% energy efficiency was gained by the designed control system. This energy efficiency amount was clearly verified by measuring of energy consumption of each plant as shown in Table 6 through Table 9. Finally the following beneficial results have been obtained:

(i) total money spent to control system = 47781 \$;

(ii) total energy efficiency per hour = 140 kW;

(iii) 1 kW electricity cost = 0.087 \$/kWh;

(iv) energy efficiency per month = 140 kWh ∗ 24 hours ∗ 30 days ∗ 0.087 \$/kWh = 8770 \$;

(v) depreciation time of the system = 47781/8770 = 5.44 months.

Nomenclature

f_a: Volume flow rate of air (m^3/s)
c_p: Specific heat capacity of air (kJ/kg·°C)
ρ_a: Mass density of air (kg/m^3)
v_z: Volume of the zone (m^3)
ACH: Air changes per hour
h: Convection heat transfer coefficient, ($W/m^2 \cdot °C$)
h_w: Rate of moisture air produced in air humidifier
k_h: Conduction heat transfer coefficient ($W/m \cdot °C$)
k: Gain coefficient (°C·s/kg)
ε: Emissivity constant
σ: Boltzmann constant ($W/m^2 \cdot °C$)
m: Mass flow rates
C_h: Overall thermal capacitance of the humidifier
T_h: Supply air temperature (°C)
α_h: Overall transmittance area factor of the humidifier (kJ/s·°C)
v_{zh}: Volume of humidifier
w_h: Supply air humidity ratio (kg/kg)

Table 9: Energy saving in plant 4.

Power consumption in old conventional system			Power consumption after the designed control system		
Date	Hour	kW	Date	Hour	kW
17.11.2013	10:30:00	46.8	07.01.2014	11:00:00	22.24
17.11.2013	11:00:00	46.8	07.01.2014	11:15:00	22.24
17.11.2013	11:30:00	46.47	07.01.2014	11:30:00	22.24
17.11.2013	12:00:00	46.47	07.01.2014	11:45:00	22.24
17.11.2013	12:30:00	46.47	07.01.2014	12:00:00	20.25
17.11.2013	13:00:00	47.8	07.01.2014	12:15:00	19.92
17.11.2013	13:30:00	44.81	07.01.2014	12:30:00	19.92
17.11.2013	14:00:00	46.47	07.01.2014	12:45:00	22.24
17.11.2013	14:30:00	46.47	07.01.2014	13:00:00	22.24
17.11.2013	15:00:00	45.81	07.01.2014	13:15:00	22.24
17.11.2013	15:30:00	45.81	07.01.2014	13:30:00	22.24
17.11.2013	16:00:00	46.47	07.01.2014	13:45:00	22.24
17.11.2013	16:30:00	46.14	07.01.2014	14:00:00	22.24
17.11.2013	17:00:00	45.81	07.01.2014	14:15:00	22.24
17.11.2013	17:30:00	43.81	07.01.2014	14:30:00	23.9
17.11.2013	18:00:00	45.81	07.01.2014	14:45:00	22.24
17.11.2013	18:30:00	45.14	07.01.2014	15:00:00	22.24
17.11.2013	19:00:00	45.14	07.01.2014	15:15:00	22.24
17.11.2013	19:30:00	45.14	07.01.2014	15:30:00	22.24
17.11.2013	20:00:00	45.47	07.01.2014	15:45:00	19.92
17.11.2013	20:30:00	45.47	07.01.2014	16:00:00	20.25
17.11.2013	21:00:00	43.48	07.01.2014	16:15:00	21.78
17.11.2013	21:30:00	45.14	07.01.2014	16:30:00	21.78
Average		**45.79**	Average		**21.8**

Average power consumption: old = 45.79, new = 21.80, and profit = 52.39%.

w_{sa}: Humidity ratio of supply air (kg/kg)
τ: Time constant
L: Time delay.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] G. P. Henze, D. E. Kalz, S. Liu, and C. Felsmann, "Experimental analysis of model-based predictive optimal control for active and passive building thermal storage inventory," *HVAC&R Research*, vol. 11, no. 2, pp. 189–213, 2005.

[2] W. Z. Huang, M. Zaheeruddin, and S. H. Cho, "Dynamic simulation of energy management control functions for HVAC systems in buildings," *Energy Conversion and Management*, vol. 47, no. 7-8, pp. 926–943, 2006.

[3] S. Soyguder and H. Alli, "An expert system for the humidity and temperature control in HVAC systems using ANFIS and optimization with Fuzzy Modeling Approach," *Energy and Buildings*, vol. 41, no. 8, pp. 814–822, 2009.

[4] B. Tashtoush, M. Molhim, and M. Al-Rousan, "Dynamic model of an HVAC system for control analysis," *Energy*, vol. 30, no. 10, pp. 1729–1745, 2005.

[5] J. Teeter and M.-Y. Chow, "Application of functional link neural network to hvac thermal dynamic system identification," *IEEE Transactions on Industrial Electronics*, vol. 45, no. 1, pp. 170–176, 1998.

[6] J. Wang, C. Zhang, and Y. Jing, "Fuzzy immune self-tuning pid control of HVAC system," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '08)*, pp. 678–683, Takamatsu, Japan, August 2008.

[7] K. F. Fong, V. I. Hanby, and T. T. Chow, "HVAC system optimization for energy management by evolutionary programming," *Energy and Buildings*, vol. 38, no. 3, pp. 220–231, 2006.

[8] A. P. Wemhoff, "Application of optimization techniques on lumped HVAC models for energy conservation," *Energy and Buildings*, vol. 42, no. 12, pp. 2445–2451, 2010.

[9] S. Lee, M. J. Kim, S. H. Pyo, J. T. Kim, and C. K. Yoo, "Evaluation of an optimal ventilation IAQ control strategy using control performance assessment and energy demand," *Energy and Buildings*, 2014.

[10] R. C. Panda, "Synthesis of PID controller for unstable and integrating processes," *Chemical Engineering Science*, vol. 64, no. 12, pp. 2807–2816, 2009.

[11] S. Skogestad, "Simple analytic rules for model reduction and PID controller tuning," *Journal of Process Control*, vol. 13, no. 4, pp. 291–309, 2003.

[12] K. J. Åström and T. Hägglund, "The future of PID control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163–1175, 2001.

[13] K. J. Åström and T. Hägglund, "Revisiting the Ziegler-Nichols step response method for PID control," *Journal of Process Control*, vol. 14, no. 6, pp. 635–650, 2004.

[14] W. K. Ho, O. P. Gan, E. B. Tay, and E. L. Ang, "Performance and gain and phase margins of well-known PID tuning formulas," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 4, pp. 473–477, 1996.

[15] S. W. Sung, J. Lee, and I. B. Lee, *Process Identification and PID Control*, John Wiley & Sons, New York, NY, USA, 2009.

[16] L. Hongli, D. Peiyong, and J. Lei, "A novel fuzzy controller design based-on PID Gains for HVAC systems," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA '08)*, pp. 736–739, IEEE, Chongqing, China, June 2008.

[17] K. F. Fong, V. I. Hanby, and T. T. Chow, "System optimization for HVAC energy management using the robust evolutionary algorithm," *Applied Thermal Engineering*, vol. 29, no. 11-12, pp. 2327–2334, 2009.

[18] D. Xue, Y. Q. Chen, and D. P. Atherton, *Linear Feedback Control*, SIAM's Advances in Design and Control, SIAM, Philadelphia, Pa, USA, 2007.

[19] N. J. Killingsworth and M. Krstic, "PID tuning using extremum seeking: online, model-free performance optimization," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 70–79, 2006.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 641235, 11 pages
http://dx.doi.org/10.1155/2015/641235

Research Article

A Cluster-Based Fuzzy Fusion Algorithm for Event Detection in Heterogeneous Wireless Sensor Networks

ZiQi Hao,[1] ZhenJiang Zhang,[1] and Han-Chieh Chao[2]

[1]*Key Laboratory of Communication and Information Systems, School of Electronic and Information Engineering, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China*
[2]*Department of Electronic Engineering, National Ilan University, Yilan 26047, Taiwan*

Correspondence should be addressed to ZhenJiang Zhang; zhjzhang1@bjtu.edu.cn

Received 2 November 2014; Accepted 22 December 2014

Academic Editor: Qing-An Zeng

As limited energy is one of the tough challenges in wireless sensor networks (WSN), energy saving becomes important in increasing the lifecycle of the network. Data fusion enables combining information from several sources thus to provide a unified scenario, which can significantly save sensor energy and enhance sensing data accuracy. In this paper, we propose a cluster-based data fusion algorithm for event detection. We use k-means algorithm to form the nodes into clusters, which can significantly reduce the energy consumption of intracluster communication. Distances between cluster heads and event and energy of clusters are fuzzified, thus to use a fuzzy logic to select the clusters that will participate in data uploading and fusion. Fuzzy logic method is also used by cluster heads for local decision, and then the local decision results are sent to the base station. Decision-level fusion for final decision of event is performed by base station according to the uploaded local decisions and fusion support degree of clusters calculated by fuzzy logic method. The effectiveness of this algorithm is demonstrated by simulation results.

1. Introduction

Recent advances of low-cost wireless sensor networks have brought about many applications such as military, environmental monitoring, and intelligent transportation system. Event detection has been widely studied as a typical application of WSN.

Heterogeneous wireless sensor networks (HWSN) are networks composed of different kinds of sensors, which are different in some aspects such as energy, computing power, and storage space. In HWSN such as a cluster-based network, cluster heads are more powerful than cluster members in all the resources like power, storage, communication, and processing data; this heterogeneity alleviates the overhead of cluster members for the fact that all the expensive computations can be performed by cluster headers [1]. Therefore, the load-balance and lifetime of network can be significantly improved.

Data fusion is a technology that enables combining information from several sources in order to form a unified picture, and it is widely used in various areas such as sensor networks, robotics, and video and image processing [2]. As an efficient method for collaborative decision making of multiple sensors, data fusion has many advantages in WSN. Using multiple sensors to detect the same event can largely eliminate data ambiguity which may be caused by only one sensor, thus to enhance data reliability and ability of fault tolerance effectively. Moreover, as WSN is energy limited network, the sensors are generally battery powered and once deployed it is hard to be recharged; therefore energy saving becomes an important factor for expanding the lifetime of WSN. Concerning the fact that energy consumption caused by communication is considerably larger than that of data processing, when data fusion is performed, sensor data are fused and only the result is forwarded; thus the number of messages is reduced, which can significantly avoid collisions and save communication energy.

Clustering is a frequently used physical architecture of data fusion; it groups sensor nodes into several clusters in order to achieve the network scalability objective [3]. Every cluster has a cluster head (CH) for executing data fusion and serving as a relay; as a consequence CH consumes more

energy than ordinary sensor nodes; therefore a more powerful sensor is more likely to be selected as a cluster head.

Fuzzy reasoning is a theoretical reasoning scheme of data fusion; it introduces the novel notion of membership degree which enables dealing with imperfect data appropriately [4]. Generally, the sensing data of a single sensor may be vague and partial; thus it is difficult to obtain the final fusion decision result via precisely quantitative calculation of these imperfect data. Fuzzy logic uses the membership degree to fuzzify the partial data and then combines them with fuzzy rules thus to produce fuzzy output, which is an efficient solution to deal with the uncertainty of data.

Previous researches mainly focus on designing or improving the fusion or clustering algorithms independently; few works have combined the two technologies together to perform monitoring task. The authors of [5] have proposed a cluster based multisensor data fusion algorithm in WSN using fuzzy logic for event detection; the method adopted fuzzy logic approach to handle the uncertainty and vagueness present in the environment data in the local decision period of cluster head. However, the proposed fuzzy logic fusion method only considers data level fusion of intracluster, without designing the specific clustering algorithm and decision-level fusion method.

In this paper, we proposed a cluster-based fusion method for event detection. We consider a heterogeneous wireless sensor networks deployment environment, where two kinds of sensor nodes exist; one is ordinary sensors, which we assume to be same with nodes in [5], where each sensor node is equipped with diverse sensors (temperature, humidity light, and Carbon Monoxide); thus data of different attributes can be gained. The other kind of sensor is mobile nodes, which have the properties of mobility and high energy. Mobile nodes serve as CHs, which will move from random deployed initial position to the corresponding destinations calculated by k-means algorithm. The proposed k-means based clustering algorithm can achieve minimum energy consumption of intracluster communication. The function of CH is performing intracluster fusion and local decision, and it also serves as delay for sending the local decision results to the base station. In order to enhance the accuracy of event detection, we propose a *fusion support degree* for each cluster, which means the probability for a cluster to participate in the decision-level fusion for final decision of emergency event. The clusters of which the value of fusion support degree is less than a predefined threshold will be rejected from the data fusion, while the value serves as weight of local decisions when performing decision-level fusion. The fusion support degree is determined by distance between CH and event centre and remaining energy of CH and cluster members. Finally, the base station will perform decision-level fusion according to local decisions and *fusion support degrees* uploaded by CHs of clusters will join the data fusion to make final decision and make corresponding warning alarms.

The rest of paper is organized as follows: we present our related work in Section 2. Section 3 is some preliminaries before we introduce our cluster based fuzzy decision fusion algorithm (CFDF). Section 4 gives the CFDF overview and detailed descriptions. In Section 5, we evaluate the performance of the proposed algorithm. We conclude this paper in Section 6.

2. Related Work

Numerous researches have been done in the field of data fusion and clustering. In this section, we give some review of the related work.

Khaleghi et al. [2] give a critical review of data fusion state-of-the-art methodologies; a new data centric taxonomy of data fusion methodologies was introduced and the challenging aspects and existing algorithms in each category are discussed. The date-related data fusion algorithms can be divided into dealing with data imperfection, data correlation, data inconsistency, and data disparateness. Generally, researches and algorithms about data fusion mainly focus on three levels [6]. The lowest level fusion is data-level fusion, of which the processing is to merge the observed raw data. Raw sensor data can be directly combined if the sensor data are commensurate. Techniques for raw data fusion typically involve classic detection and estimation methods such as averaging method. However, if the data are uncorrelated, the fusion should be performed at a feature or a decision level. Feature-level fusion can be performed after data-level fusion or directly performed by a single sensor. In feature-level fusion, features are extracted from multiple sensor observations and combined into a single feature vector for decision-level fusion. Decision-level fusion is the highest level fusion, of which the main idea is fusion of local decisions of each sensor; the local decision is a preliminary determination of an entity's location, attributes, identity, and so on. The most commonly used decision-level fusion methods include voting techniques, Bayesian inference, and Dempster-Shafer's method [6].

Gók et al. in [7] proposed a decision-level fusion algorithm based on fuzzy logic used for single target classification. The research adopts kNN to make local decisions, which might cost a lot of time because it needs quite a few number of training samples to train classifiers. At each node, a fusion chance will be calculated according to the distance and energy of received information using a fuzzy method. The proposed decision-level fusion method achieves high efficiency on the occasion that the number of sensors is quite small. However, when a large number of sensor nodes are needed in a given sensing field, using every sensor to conduct local decision and uploading decision results will result in massive extra information processing and communication energy consumption. Grouping the sensors into several clusters could be a good way to solve the above problem, in that the majority of data processing and transmission could be done at CHs.

Clustering algorithms in the literature vary in their objectives; often the objective is related to application requirements [3]. For example, if the application is sensitive to data delay, the connectivity and length of data routing paths are usually considered as critical problems. Some other popular problems such as load balancing, fault-tolerance, minimal cluster count, and maximal network longevity are also concerned. In

the field of distributed clustering algorithms, each sensor executes the algorithm independently and their states are based on the cluster membership decisions on their own states. Low Energy Adaptive Clustering Hierarchy (LEACH) [8] is the most popular distributed clustering algorithm; a dynamic adaptive probability-based CH selecting mechanism was proposed. Unlike the LEACH of which the selection of CHs is random, hybrid energy-efficient distributed clustering (HEED) [9] proposes a CH selecting method considering a hybrid of energy and communication cost. DWEHC in [10] proposed by Ding et al. can achieve more aggressive goals than those of HEED, the weight of CH is based on its residual energy and distance to its neighbors, and the largest weight node in a neighborhood may become a CH. Simulation results illustrated that DWEHC can do better in load-balance and has $O(1)$ time complexity.

Manjunatha et al. in [5] suggested a data fusion method based on clustering for environment monitoring. In the proposed method, each sensor is equipped with diverse sensors, and the processing and fusion of these diverse sensor signals are carried out by CHs using proposed fuzzy rule based system. The reliability and accuracy of environmental detection are gained by multiple data fusion. However, this paper did not give a complete data fusion processing for it only considered fusion of intracluster and multiple sensing variables, but the decision-level fusion of intercluster was not mentioned.

3. Preliminaries

In this section, we will give a brief introduction to some mathematical models used in our proposed cluster-based fusion model and also give some assumptions under the monitoring environment of this paper.

3.1. A Brief Introduction to k-Means Algorithm. k-means clustering [11] is one of the simplest unsupervised learning algorithms that solve the clustering problem. The specific objective of k-means is to classify a given data set into a certain number of clusters (assume k clusters). The main idea is to define k centroids randomly or in a cunning way beforehand and then assign each data point to the closest centroid to be a member of the current cluster; the first round is completed after all the data points are assigned. By now k clusters are constructed and next the second round will be performed by recalculating k new centroids as barycenters of the clusters resulting from the previous round, and then a new binding has to be done between the same data set points and the nearest new centroids. And then, do this loop until centroids do not move any more.

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2, \quad (1)$$

where $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centres. The specific steps of k-means are described as follows [12]:

(1) Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.

(2) Assign each object to the group that has the closest centroid.

(3) When all objects have been assigned, recalculate the positions of the k centroids.

(4) Repeat Steps (2) and (3) until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

3.2. The Fuzzy Theory Introduction. The concept of fuzzy set was proposed by Zadeh in 1965 [4], and the introduction of fuzzy theory is to deal with problems involving knowledge expressed in vague or linguistic terms especially when the boundary of the set contained in the universe is ambiguous, vague, or fuzzy [5]. In general crisp set, the relation between an element and a given set is that the element is either belonging to the set or not. However, in a fuzzy set, the element is not definitely belonging to a set or not; instead some ambiguity or vagueness may exist; thus elements belong to a fuzzy set to a certain degree represented by a number in interval $[0, 1]$. The number is called membership degree; the larger the value of membership degree is, the greater the degree of the element belonging to the set is. When the values of all the elements are either 0 or 1, the fuzzy set will degenerate into a crisp set.

Considering a set X whose element is represented by x, the membership function $\mu_{\widetilde{A}}(x)$ is a function that associates with each x in X with a membership degree in $[0, 1]$. A fuzzy set A in X is a set of ordered pairs and is given as $\widetilde{A} = \{x_i, \mu_{\widetilde{A}}(x)\}$, $x_i \in X$, where x_i represents the element in a fuzzy set A.

Fuzzy logic inference is a process using the fuzzy input and fuzzy rules to obtain the fuzzy output, of which the substance is to map the input space into a given output space by fuzzy roles. The fuzzy roles normally use the IF/THEN statement, the format of which is IF x is A Then y is B, where x, y are the input linguistic variable and output linguistic variable, and A, B are the linguistic value of fuzzy set before reasoning and after reasoning, respectively. Mamdani fuzzy inference system proposed by Ebrabim Mamdani in 1975 is a widely used fuzzy inference method, which can implement the reasoning computation from input to output effectively through a serious of predefined fuzzy roles.

3.3. Assumptions. In this paper, we consider a cluster-based data fusion algorithm for event detection; our research is based on the following assumptions:

(1) To simplify the complexity of the problem, we assume the sensing field is a flat two-dimensional region; no barriers exist; thus the mobile nodes can smoothly move inside the sensing area. And also the nodes are

uniformly deployed in the sensing filed and cannot be recharged after deployment.

(2) In consideration of load-balance problem, we adopt a heterogeneous WSN composed of two kinds of sensor nodes, general static nodes and mobile nodes; the mobile nodes will act as CHs and the energy of which is preset higher than that of general nodes, because CHs will consume more energy on data transmission and processing.

(3) Each sensor node can achieve its location by a GPS module or some other techniques such as the method proposed in [13]. The locations will be forwarded to BS only once and stored by BS for the subsequent clustering and fusion calculation.

(4) Like LCA [14], a single-hop intracluster topology is established by our proposed k-means based clustering algorithm, and TDMA is used for intracluster communication. Intercluster uses multiple-hop routing to arrive at the sink node.

4. Cluster Based Fuzzy Fusion Algorithm (CBFFA)

In this section, we will introduce our proposed cluster-based fuzzy fusion algorithm (CBFFA). Considering the time complexity of clustering may decline the efficiency of event detection, we proposed a centralized none-event driven clustering method; namely, the clustering is performed before some event occurs. Algorithm 1 illustrated the pseudocode of CBFFA.

Our proposed CBFFA includes three phases, the initial clustering phase, event detection phase, and finally the event warning phase. In initial phase, the sensor nodes (including static nodes and mobile nodes) are randomly deployed in the two-dimensional sensing filed, and each sensor sends its position coordinate to the BS via multihop routing. Therewith the k-means based clustering algorithm (kBCA) will be conducted. We assume that the number of mobile nodes will be larger than that of clusters, and the remaining mobile nodes which are not selected to be the CH will serve as spare nodes. The detailed description of the proposed k-means based clustering algorithm (kBCA) is introduced in Section 4.1. After dividing the nodes in sensing filed into k clusters, the event detection will be performed. When an emergency arises, it will be detected by the sensors around immediately. In this paper, we assume the location of the event centre to be the position coordinates of the sensor which first detects the event.

It is obvious that a considerable number of sensors in each cluster can achieve the sensing data for event detection. In this paper, our objective is to select the clusters of which the sensing accuracy is relatively high to carry out data fusion, thus to improve accuracy of event detection. Meanwhile, for prolonging the lifetime of the whole WSN, the clusters having relatively low energy are not recommended to join in the data fusion. Instead, they will switch into sleeping mode.

The accuracy of the monitoring data may vary with the distance between sensor and the event centre; when the distance increases the accuracy will decline. In addition, the remaining energy of sensors also has an influence to the accuracy of the event detection, considering both sending and receiving information will consume energy of sensors, when a sensor has little energy and cannot be recharged during transmitting, the sensing data may fail to be sent; thus the transmission reliability is closely related to the remaining energy of sensors. We proposed a fuzzy theory based method to decide the chance (fusion support degree) of a cluster to participate in the data fusion; the fusion support degree is calculated in the BS according to the uploaded distance value between CH and event centre and average remaining energy value of a cluster. The specific calculation process of fusion support degree is described in Section 4.2. When the fusion support degree is less than a predefined threshold, which means the cluster is either quite far from the event centre or lower in energy, as explained before the sensing data accuracy of these clusters will be very low. Therefore, these clusters will be rejected from data fusion and enter into sleeping mode. The clusters of which fusion support degree is larger than threshold will conduct data transmission and fusion. Here, the fusion support degree is a value that explains the credibility of a cluster's local decision; the higher the value is, the more contribution the local decision will bring to the final decision-level fusion result. Finally, the final decision-level fusion will be conducted at the BS and afterwards the corresponding early warning information will be made.

4.1. k-Means Based Clustering. In this section, we will introduce the k-means based clustering algorithm (kBCA). The k-means clustering algorithm can be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centers. And for sensor nodes, the transmission energy consumption is closely dependent on transmission distance, which is proportional to squared Euclidean distance or quadruplicate Euclidean distance [15]. Therefore, kBCA will have good performance for decreasing intracluster energy consumption. However, the initialized centroids and the value of k are not specified in the algorithm. Usually, the initial centroids are determined by random way and k is a man-made value. In this paper, we adopt a cunning way to determine the value of k and the centroids' initial position; that is, we selected the least possible value of k to guarantee the network connectivity and initial positions of centroids are suggested to be uniformity distributed and far from each other. Algorithm 2 illustrated the pseudocode of kBCA.

First, we should determine the number of clusters according to size of a cluster. We adopt the mean of numbers of static nodes inside the circle centred by each of N_k mobile nodes to be the size of a cluster. In consideration of connectivity intercluster, the radius of the circle is suggested to be $R_c/4$. In order to increase the convergence speed of k-means, the initial centroids should be far from each other; thus we approximately divide the sensing field into k number of squares and let centres of them be initial centroids. After running k-means algorithm, the convergent centroids are achieved, and

Notations:
BS: the base station
k_i: the number of static nodes in cluster i
E_{CM}, E_{CH}: initial energy of cluster members and CH respectively
$\overline{E_{CM_i}}$: average remaining energy of members in cluster i
p_{CM}, p_{CH}: weighting factors of cluster member and CH, satisfying $p_{CM} + p_{CH} = 1$
$\overline{E_{C_i}}$: average remaining energy of cluster i
P_i: a predefined threshold to determine a cluster will join fusion or not
$k$meansBasedClustering(nodes): the clustering algorithm based on k-means proposed in this paper
FusionSupportDegree(CH_i): the chance of a cluster to participate in the final decision-level fusion
LocalDecision(CH_i): intra-cluster fusion and local decision performed by CH_i
DecisionLevelFusion(CHs): the process of decision-level fusion conducted by BS
Initialization clustering phase:
(1) Deploy all the nodes (both the static nodes and mobile nodes) randomly in the sensing filed.
(2) **For** each static node **do**
(3) Send its location (x_i, y_i) to BS via multi-hop routing
(4) Vector_staticNodes(i) $\leftarrow (x_i, y_i)$
(5) **End for**
(6) **For** each mobile node **do**
(7) Send its location (x_{ci}, y_{ci}) to BS via multi-hop routing
(8) Vector_mobileNodes(i) $\leftarrow (x_{ci}, y_{ci})$
(9) **End for**
(10) $k$meansBasedClustering(matrix_nodes);
Event detection phase:
(1) **If** an event occurs
(2) **Then** the first node that detect the event sent its location (x_{event}, y_{event}) to BS var multi-hop routing
(3) **For each cluster head** CH_i
(4) $d_{ci} \leftarrow \sqrt{(x_{ci} - x_{event})^2 + (y_{ci} - y_{event})^2}$
(5) **End for**
(6) **For** each CM_j in C_i
(7) Send its remaining energy E_j to C_i
(8) **End for**
(9) **For** each C_i
(10) $\overline{E_{CM_i}} \leftarrow \frac{1}{k_i}\sum_{j=0}^{k_i} E_j$, $\overline{E_{C_i}} \leftarrow p_{CM}\frac{\overline{E_{CM_i}}}{E_{CM}} + p_{CH}\frac{E_{CH_i}}{E_{CH}}$
(11) Send $\overline{E_{C_i}}$ to BS
(12) **End for**
(13) **For each cluster head** CH_i
(14) $P_i \leftarrow$ FusionSupportDegree(CH_i);
(15) **If** $P_i > P_s$
(16) **Then**
(17) $D_{C_i} \leftarrow$ LocalDecision(CH_i);
(18) D_{C_i} is sent to BS
(19) **Else**
(20) C_i turns on sleep mode
(21) **End If**
(22) **End For**
(23) $P_{final} \leftarrow$ DecisionLevelFusion(CHs);
(24) **End If**
Event alert phase:
(1) BS makes the corresponding early warning information according to decision-level fusion result.

Algorithm 1: Pseudocode of cluster based fuzzy fusion algorithm (CBFFA).

next the mobile nodes server as CHs should move to these centroids thus to ensure minimizing the energy consumption of intracluster communication. A variety of methods can be used to choose CHs and match them to the locations of centroids; the most common one is greedy method; for each of the k centroids, the nearest mobile node will be chosen as a CH and move to location of the current centroid. To achieve global optimum, Hungarian method can be used to obtain

Notations:
R_c: communication range of the CH
N_k: a selected number of mobile nodes to determine number of clusters k
Static_nodes: number of static nodes
Input: Coordinates of static nodes Vector_staticNodes
Determine value of k:
(1) **For** $i = 1 : N_k$
(2) **For** all static nodes (x_j, y_j) If
(3) $\sqrt{(x_j - x_{ci})^2 + (y_j - y_{ci})^2} \leq \frac{R_c}{2}$
(4) Static_Number(i)(++);
(5) **End for**
(6) **End for**
(7) cluster_size $\leftarrow \frac{1}{N_k} \sum_{i=1}^{N_k}$ Static_Number(i)
(8) $k \leftarrow \frac{\text{static\_nodes}}{\text{cluster\_size}}$
(9) Divide the sensing field into k squares, and order the centres of squares to be initial positions of centroids.
Determine converged centroids using k-means:
(1) Let initial initial positions of centroids to be $m_1, m_2, \ldots, m_k$
(2) **Until** there are no changes in any mean
(3) Use the estimated means to classify the samples into clusters
(4) **For** i from 1 to k
(5) Replace m_i with the mean of all of the samples for cluster i
(6) **End For**
(7) **End Until**
Motion period for CHs:
(1) Select k mobile nodes and dispatch them to locations of $m_1, m_2, \ldots, m_k$ using greedy method or Hungarian method.

Algorithm 2: Pseudocode of k-means based clustering algorithm (kBCA).

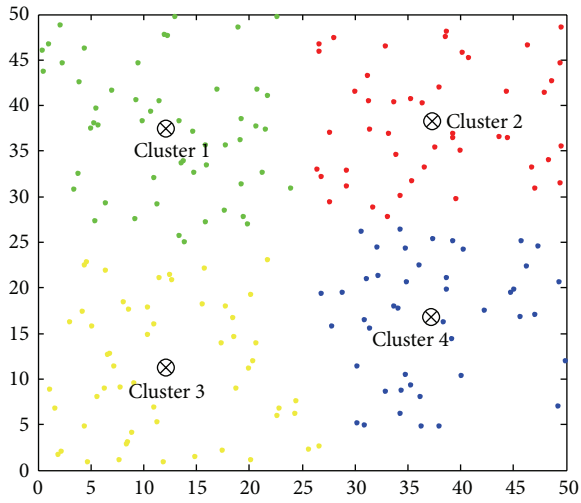


Figure 1: Clustering result of kBCA.

the optimal matching which can minimize the overall moving distance of mobile sensors. Figure 1 shows the clustering result of kBCA.

4.2. The Fusion Support Degree. The fusion support degree is a value that measures the reliability for the measured data of a cluster. As discussed before, the accuracy of event detection is closely related to distance between the sensor and event and the remaining energy of a sensor; thus this paper uses distance and remaining energy to decide the fusion support degree. Generally, we use *far* or *near* to describe a distance and use *high* or *low* to describe energy. However, the linguistic variables suchlike *far* or *high* may include some level of uncertainty and vagueness. Therefore, the final decision result of event may be imprecise if crisp sets of distance and energy are used to deal with data fusion. As fuzzy logic can deal with the knowledge of which the boundaries are not clear through membership function, for this reason we adopt fuzzy logic method to calculate fusion support degree. We adopt a similar method as proposed in [7], using distance and remaining energy as two fuzzy input variables, and fusion support degree as fuzzy output. Figure 2 illustrated the structure of fuzzy logic method, where the distance and energy are crisp input variables and fusion support degree is crisp output after defuzzification.

To simplify the analysis for problem, trapezoidal and triangular membership function is defined to describe fuzzy input variables *distance* and *energy*. The membership functions and corresponding linguistic states of distance and energy are shown in Figures 3(a) and 3(b).

Figure 4 shows the membership functions and corresponding linguistic states of fuzzy output variable, namely, fusion support degree. We introduce the same membership

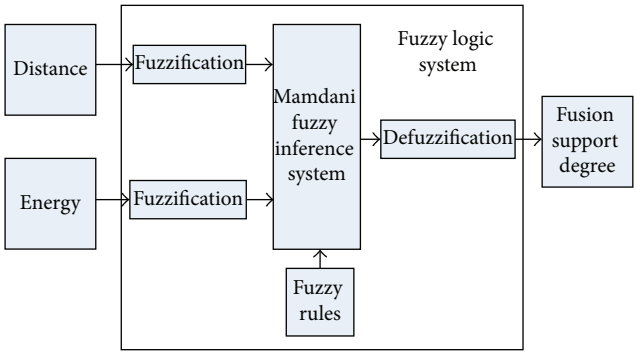


Figure 2: Structure of fuzzy logic method.

Table 1: Fuzzy rules for our fuzzy algorithm.

Rule number	Distance	Energy	Fusion support degree
1	Far	High	MM
2	Far	Medium	ML
3	Far	Low	LL
4	Medium	High	HL
5	Medium	Medium	MH
6	Medium	Low	LM
7	Near	High	HH
8	Near	Medium	HM
9	Near	Low	LH

functions used in [7]. The fusion support degree is divided into nine levels, namely, LL (Low in Low), LM (Medium in Low), LH (High in Low), ML (Low in Medium), MM (Medium in Medium), MH (Medium in High), HL (Low in High), HM (Medium in High), and HH (High in High), and the fusion support degree they represent is successively larger.

Table 1 illustrated fuzzy rules for our fuzzy algorithm. In research [7], the authors consider a sensing environment with obstacles; thus even the sensor is relatively near from the target; the sensing accuracy may be low due to obstacles. Therefore, they give more weight on energy fuzzy variable than distance when dealing with fuzzy output. In this paper, we assume no obstacles exist, so the accuracy is largely related to distance; consequently in most cases we give fuzzy input variable distance more weight. However, under the circumstances that remaining energy of cluster is quite low, to prolong the lifetime of networks, the cluster having low energy is not suggested to join the fusion; thus we give more weight on energy when the remaining energy is low.

After fuzzy variables and fuzzy rules are designed, a fuzzy inference method should be used to achieve fuzzy output, thus to get the final system outputs (FusionSupportDegree(CHi) in Algorithm 1) through defuzzification. In this paper, we use the most commonly used fuzzy inference system, namely, Mamdani method, to obtain fuzzy output.

4.3. Local Decision. We take fire detection as an example of event detection. In this scenario, fire centre can be modeled as a fixed coordinate point in the sensing filed. Each sensor has an integrated multiple sensors module which can collect environmental information such as temperature, humidity, light intensity, and the amount of Carbon Monoxide (CO) [5]; thus the sensing data structure of a sensor can be described as a multidimensional vector $(v_1, v_2, \ldots, v_n)$, where v_i $(1 \leq i \leq n)$ stands for the value of ith environmental information variable.

After fusion support degrees of each cluster are calculated by BS, the clusters whose fusion support degrees are less than the threshold will be eliminated from fusion processing, which means they will not conduct data uploading and data fusion, while the clusters whose fusion support degrees are larger than the threshold will perform local decision and upload decision results thus to join the final decision-level fusion. The local decision of a cluster includes two steps. Firstly, all sensors inside a cluster send their sensing data to CH; then intracluster fusion is performed by CH. Secondly, according to the fusion results, the CH will make the local decision, which reflects evaluation of emergency level for the current cluster to event.

When all sensors inside a cluster have sent their sensing data to the CH, intracluster data fusion is conducted. We use the mean of sensing values of each dimension to be intracluster fusion result. Considering that some sensors might have a possibility of being damaged during deploying, which will reduce the accuracy of sensing data, thus these singular sensors should be eliminated before data fusion.

Note that the multidimensional vector of cluster i after fusion is represented by $(\overline{v_{i1}}, \overline{v_{i2}}, \ldots, \overline{v_{in}})$. The mean of these n variables will be used to conduct local decision; however it is not easy to achieve a quantitative equation for local decision result; therefore we adopt the fuzzy logic method same as proposed in Section 4.2 to get local decision result by fuzzifying each dimension of $(\overline{v_{i1}}, \overline{v_{i2}}, \ldots, \overline{v_{in}})$ to be fuzzy input and defining decision result as fuzzy output variable. This paper did not design the specific fuzzy sets and rules for fuzzy input and output, so that the designing of fuzzy sets is closely dependant on the type of events; when event changes, fuzzy input and fuzzy rules will change as well.

4.4. Decision-Level Fusion. The result of local decision discussed in Section 4.3 is a percentage value calculated by a single cluster which illustrated the possibility an event will happen or the degree of emergency. The decision-level fusion means combining local decisions from several clusters to get a final evaluation of the event. The fusion support degree and local decision result introduced before are prepared for decision-level fusion, and (2) shows calculation of final decision-level fusion result:

$$P_{\text{final}} = \frac{\sum_{p_i > p_s} P_i D_{ci}}{\sum_{p_i > p_s} P_i}, \tag{2}$$

where P_i denotes fusion support degree of cluster i and D_{ci} denotes local decision result of cluster i. If P_{final} is larger than a threshold, BS will make the corresponding early warning information according to fusion result.

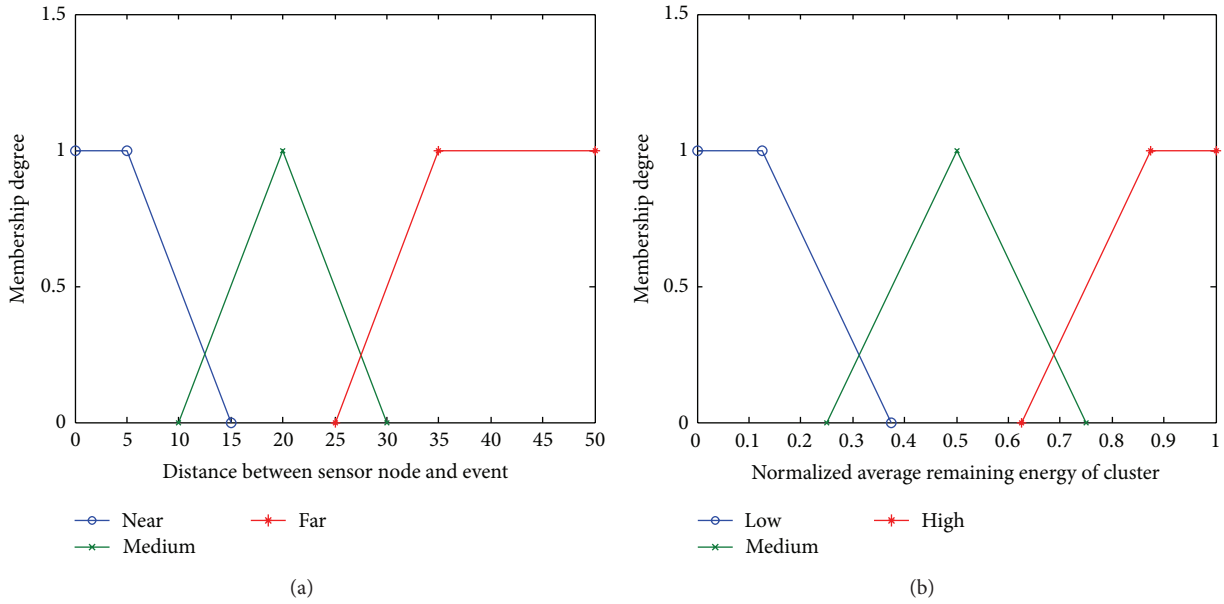


Figure 3: (a) Fuzzy sets for distance between sensor and event. (b) Fuzzy sets for remaining energy.

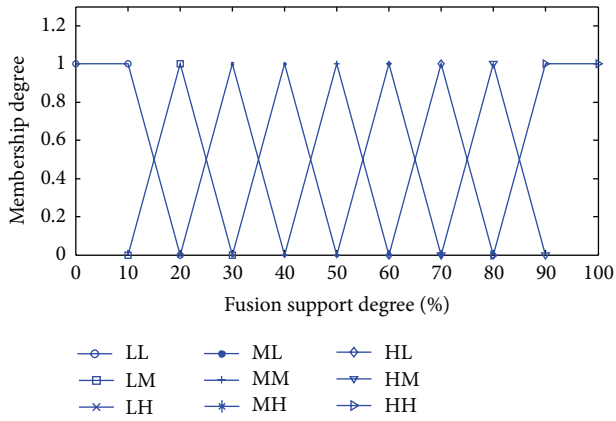


Figure 4: Fuzzy sets for fuzzy output variable fusion support degree.

5. Results and Discussion

In this section, we will evaluate the performance of our proposed cluster-based fuzzy decision fusion algorithm (CFDFA). The simulation is implemented in Matlab. We assume the sensing area to be a flat two-dimensional square region of which the size is 50 m × 50 m, 200 static sensor nodes and 15 mobile nodes are randomly deployed on the sensing area, and the communication radius (sensors can communicate with each other when their distance is less than communication radius) is assumed to be 30 m. We take fire detection as an example of event detection, and the sensing data structure of a sensor is defined as a multidimensional vector (temperature, light intensity, and CO density).

5.1. Analysis of k-Means Based Clustering Algorithm (kBCA). Firstly, we give a simulation of the initial clustering algorithm *k*BCA. According to Algorithm 2, we order $N_k = 5$; then the cluster size is estimated by (3), of which the value is 35.5. The value means the maximum size a cluster could be so as to meet the connectivity of the network. Consequently, the number of clusters should be more than the value of 200/35.5 = 5.6:

$$\text{cluster\_size} \longleftarrow \frac{1}{N_k}\sum_{i=1}^{N_k}\text{Static\_Number}\,(i)\,. \tag{3}$$

Figure 5 shows the clustering results when the number of cluster changes. It is obvious that, with the increasing of cluster number, the number of members in a cluster reduces consequently. Therefore, the average communication energy consumption of a cluster will reduce.

In order to analyze lifetime of the whole network, Figure 6 gives a quantitative analysis of overall communication distance and variance of cluster size when number of clusters changes. When the cluster number changes from 5 to 15, there is a declining trend of the overall distance, which illustrates that the larger the cluster number is, the more energy-efficient the network will be. Generally, the lifecycle of whole network in a clustering environment is not only related to the overall communication consumption, but also related to balance of cluster size. A balanced clustering means that clusters have approximate sizes, thus to guarantee balance of energy consumption between different clusters. As shown in Figure 6, the variance of cluster sizes changes irregularly when the cluster number is less than 10; however, when cluster number is larger than 10, the variance has a decline trend. Above analysis concludes that number of clusters will definitely influence the lifecycle of whole network. In some certain circumstance, the more the cluster number is, the more energy-efficient and load-balanced the network will be. However, increasing of cluster number will cause a corresponding increasing of

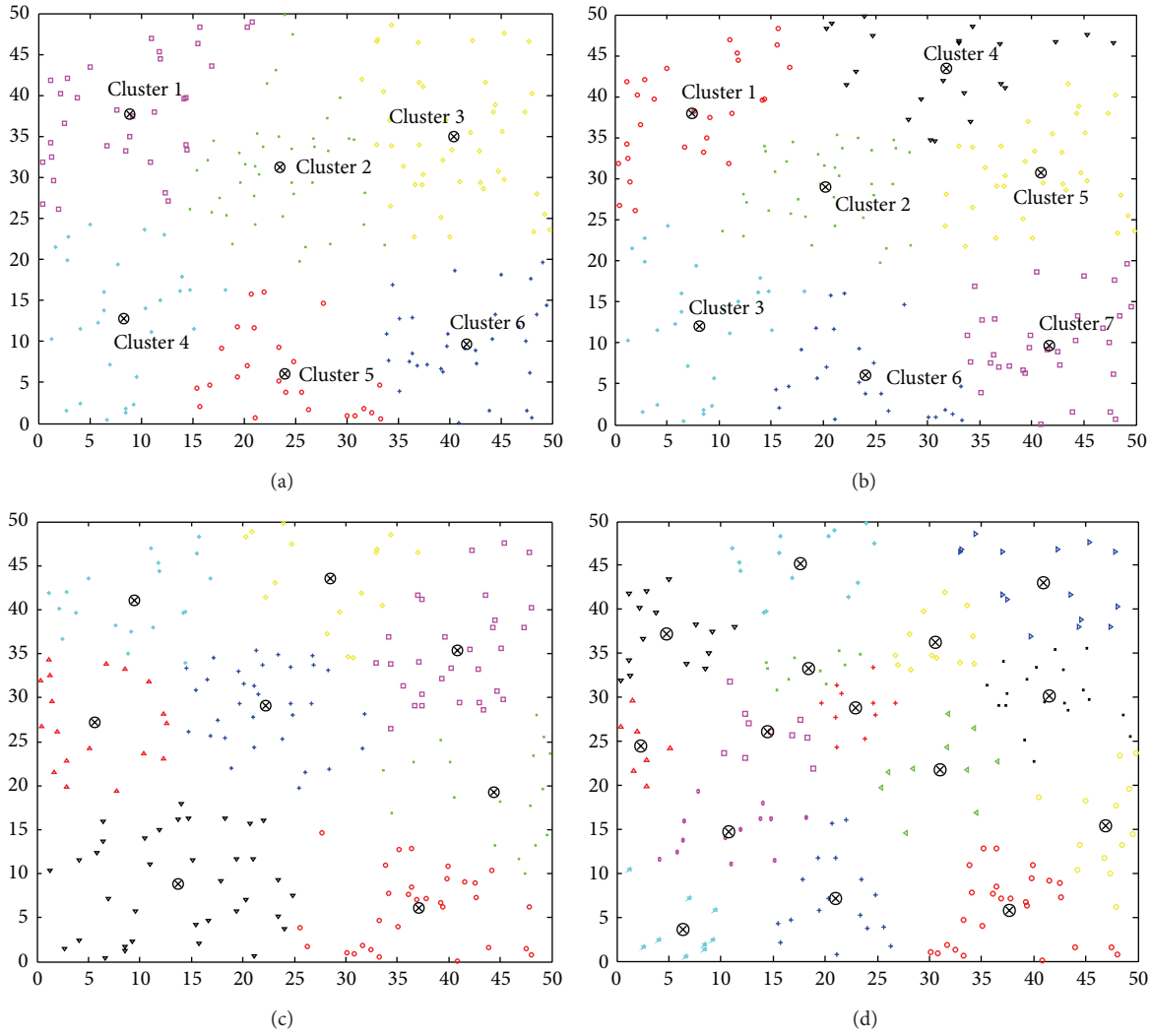


Figure 5: Clustering results of kBCA when cluster number k changes. (a) $k = 6$. (b) $k = 7$. (c) $k = 8$. (d) $k = 15$.

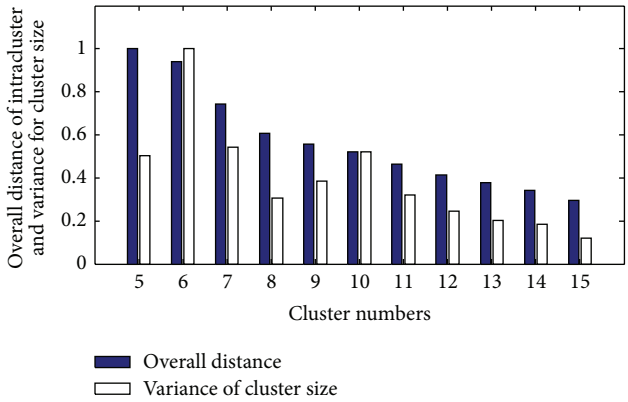


Figure 6: Overall distance of intracluster and variance of cluster size when cluster number k changes.

mobile nodes; generally mobile nodes are more expensive than ordinary nodes; thus with the increasing of cluster number, the cost of whole network will be increased either. So there is a tradeoff between lifetime and cost of the network; the number of cluster should be designed according to the actual situation.

5.2. Simulation for Event Detection. To make the compromise between the lifecycle and expense of network, the cluster number is proposed to be 12. Figure 7 shows the assumed event detection scenario, 12 clusters are formed by kBCA, and a random event occurs with a location of (28,18). Next, the fusion support degree of each cluster is calculated by fuzzy method introduced in Section 4.2. We use the fuzzy logic toolbox on Matlab to simulate the fuzzy output. Table 2 illustrates the fusion support degree results of 12 clusters, in which the distance means distance between fire centre and CH and energy means the average residual energy of a cluster. We define the threshold of fusion support degree to be 40%. Therefore, cluster 5, cluster 8, and cluster 10 will be rejected from data fusion and enter into sleeping mode; the data

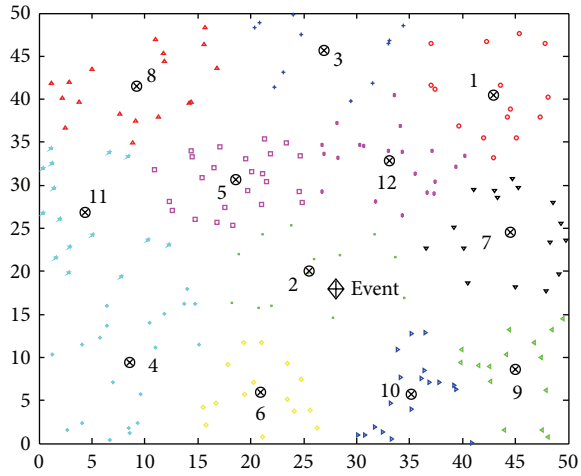


Figure 7: Event detection scenario when cluster number is 12.

Table 2: Fusion support degree of clusters.

Cluster number	Distance	Energy (%)	Fusion support degree (P_i) (%)
1	27.03	0.83	61.6
2	3.17	0.35	68.6
3	27.72	0.73	57.3
4	21.24	0.66	62.7
5	15.76	0.13	20
6	14.00	0.85	74.9
7	17.74	0.69	65.2
8	30.08	0.35	32.7
9	19.41	0.45	60
10	14.27	0.23	21.4
11	25.27	0.48	58.7
12	15.69	0.95	70

fusion and event detection processing are performed by the remaining 9 clusters.

In the simulation of local decision, we use the same fuzzy sets of inputs and output variables as proposed in [5]. $3^3 = 27$ rules are designed; some of the example rules are as follows:

> IF Temperature is low and light intensity is low and CO is low THEN fire probability is vary_low;
>
> IF Temperature is high and light intensity is high and CO is high THEN fire probability is vary_high;
>
> IF Temperature is medium and light intensity is high and CO is high THEN fire probability is high.

Table 3 is the local decisions for clusters, which means the fire probability evaluated by the current cluster.

Finally, the final decision-level fusion is calculated by formula (2); the result is $P_{\text{final}} = 60.5\%$, meaning that there is a probability of 60.5% that a fire disaster could happen.

Table 3: Local decisions for clusters.

Rule number	Fusion support degree	Intracluster fusion result	Fire probability (D_{ci}) (%)
1	61.6	(21, 322, 52)	31.5
2	68.6	(80, 875, 85)	92.2
3	57.3	(30, 300, 38)	44.5
4	62.7	(23, 225, 41)	32
6	74.9	(60, 810, 80)	91.6
7	65.2	(40, 355, 81)	75
9	60	(35, 450, 60)	62.5
11	58.7	(20, 268, 50)	25
12	70	(38, 303, 71)	75

The final result and corresponding early warning information will be sent to users for disaster prevention.

5.3. Algorithm Complexity Analysis. At last, we will give the computation complexity evaluation of the proposed CFDFA. It has been calculated that the time complexity of k-means is $O(tkdn)$, where t is number of iterations, k is number of clusters, n is number of data objects, and d is the dimension (equals 2 in this paper). The time complexity of event detection is $O(n + 4k + 2m + 1)$, where m is number of clusters taking part in data fusion. Therefore, the total time complexity is $O(n)$, which means the proposed algorithm is completed in linear time; thus it is efficient and enables scalability when dealing with large data sets. Moreover, as the clustering is performed once the sensors are deployed before event occurs, the time consumption of clustering can be ignored when analyzing the efficiency of event detection; thus the proposed algorithm is efficient enough for detecting and reporting sudden events.

6. Conclusions

In order to enhance the accuracy of event detection, this paper proposed a cluster-based data fusion algorithm using fuzzy logic method (CFDFA) for event detection in a heterogeneous wireless sensor network. In terms of the energy-efficient problem in WSN, we suggested a k-means based clustering algorithm (kBCA); the clustering algorithm can efficiently reduce energy consumption of intracluster, and simulation results illustrated that when a proper cluster number is chosen, well load-balance of cluster size can also be gained; thus the lifetime of network can be prolonged. A crisp form of possibility of event occurrence can be achieved by base station after CFDFA is performed; we take fire detection as an example to simulate our event detection algorithm and get crisp value of the fire probability.

In the future, we may consider more complicated scenarios where the event is no longer static; thus an event-driven dynamic clustering algorithm will be designed. Moreover, we may work on more details about the efficient intracluster data fusion algorithms suitable for our event detection algorithm, where singular point detection may be a key point to be considered.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by National Natural Science Foundation under Grant 61371071, Beijing Natural Science Foundation under Grant 4132057, and Academic Discipline and Postgraduate Education Project of Beijing Municipal Commission of Education.

References

[1] B. Premamayudu, V. Rao, and S. Varma, "A key management & establishment scheme in heterogeneous wireless sensor networks (HWSN)," *International Journal of Applied Engineering Research*, vol. 9, no. 20, pp. 6813–6821, 2014.

[2] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: a review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[3] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[4] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[5] P. Manjunatha, A. K. Verma, and A. Srividya, "Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method," in *Proceedings of the IEEE Region 10 and the 3rd International Conference on Industrial and Information Systems (ICIIS '08)*, pp. 1–6, IEEE, Kharagpur, India, December 2008.

[6] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[7] S. Gók, A. Yazici, A. Coşar, and R. George, "Fuzzy decision fusion for single target classification in wireless sensor networks," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ '10)*, pp. 1–8, Barcelona, Spain, July 2010.

[8] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[9] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[10] P. Ding, J. A. Holliday, and A. Celik, "Distributed energy-efficient hierarchical clustering for wireless sensor networks," in *Distributed Computing in Sensor Systems*, vol. 3560 of *Lecture Notes in Computer Science*, pp. 322–339, Springer, Berlin, Germany, 2005.

[11] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, University of California Press, Berkeley, Calif, USA, 1967.

[12] http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html.

[13] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM '03)*, vol. 3, pp. 1734–1743, 2003.

[14] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987.

[15] P. Ding, J. A. Holliday, and A. Celik, "Distributed energy-efficient hierarchical clustering for wireless sensor networks," in *Distributed Computing in Sensor Systems*, Lecture Notes in Computer Science, pp. 322–339, Springer, Berlin, Germany, 2005.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 506909, 17 pages
http://dx.doi.org/10.1155/2015/506909

Research Article

Minimum Cost Data Aggregation for Wireless Sensor Networks Computing Functions of Sensed Data

Chao Chen,[1] Kyogu Lee,[2] Joon-Sang Park,[3] and Seung Jun Baek[1]

[1]*Department of Computer and Communications, Korea University, Seoul 136-701, Republic of Korea*
[2]*Department of Digital Contents Convergence, Seoul National University, Seoul 151-742, Republic of Korea*
[3]*Department of Computer Engineering, Hongik University, Seoul 121-791, Republic of Korea*

Correspondence should be addressed to Seung Jun Baek; sjbaek@korea.ac.kr

Received 11 December 2014; Accepted 12 January 2015

Academic Editor: Yun Liu

We consider a problem of minimum cost (energy) data aggregation in wireless sensor networks computing certain functions of sensed data. We use in-network aggregation such that data can be combined at the intermediate nodes en route to the sink. We consider two types of functions: firstly the summation-type which includes *sum*, *mean*, and *weighted sum*, and secondly the extreme-type which includes *max* and *min*. However for both types of functions the problem turns out to be NP-hard. We first show that, for *sum* and *mean*, there exist algorithms which can approximate the optimal cost by a factor logarithmic in the number of sources. For *weighted sum* we obtain a similar result for Gaussian sources. Next we reveal that the problem for extreme-type functions is intrinsically different from that for summation-type functions. We then propose a novel algorithm based on the crucial tradeoff in reducing costs between local aggregation of flows and finding a low cost path to the sink: the algorithm is shown to empirically find the best tradeoff point. We argue that the algorithm is applicable to many other similar types of problems. Simulation results show that significant cost savings can be achieved by the proposed algorithm.

1. Introduction

Motivation. In this paper we consider the problem of minimum cost (energy) data aggregation in wireless sensor networks (WSN) where the aggregated data is to be reported to a single sink. A common objective of WSN is to retrieve certain *summary* of sensed data instead of the entire set of data. The relevant summary is defined as a certain function applied to a set of measured data [1]. Specifically we are given a function $g(\cdot)$ such that, for a set of measurement data $x_1, \ldots, x_n$, the goal of the sink is to retrieve $g(x_1, x_2, \ldots, x_n)$. Examples of $g(\cdot)$ are `mean`, `max`, `min`, and so forth. When `mean` function is used, $g(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i/n$. For applications such as "alarm" systems, one can use `max` as $g(\cdot)$, for example, $g(x_1, \ldots, x_n) = \max_{i=1,\ldots,n}[x_i]$ where x_i can be temperature values in forest-fire monitoring systems or the structural stress values measured in a building. We will refer to $g(\cdot)$ as a *summary function* throughout this paper. Certain types of $g(\cdot)$ allow us to combine data at the intermediate nodes en route to the sink. Such combining techniques are commonly referred to as *in-network aggregation* [2–4]. By using in-network aggregation one can potentially save communication costs by reducing the amount of traffic [5–7]. For instance, in the applications such as wireless multimedia sensor networks (WMSN) where the transmitted multimedia data has a far greater volume than that in typical WSNs, the in-network aggregation technique is crucial for the purpose of saving energy and extending network lifetime [8, 9]. While in-network aggregation offers many benefits, it poses significant challenge for network design, for example, designing routing algorithms so as to minimize costs such as energy expenditure and delay. In particular, we show that it is crucial to take into account how the summary function $g(\cdot)$ affects the statistical properties of sensed data.

Objectives. In this paper we study the minimum cost aggregation problem for several types of $g(\cdot)$. The performance of in-network aggregation relies heavily on the properties of

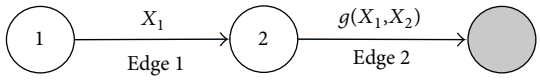


Figure 1: An example of computing and communicating a summary.

the function $g(\cdot)$. To be specific let us briefly look at the problem formulation. Consider the single-sink aggregation problem where we define the cost function as follows. Let E denote the set of links in the network. We would like to minimize

$$\sum_{e \in E} w_e \phi_e, \tag{1}$$

where w_e represents the weight associated with link e and ϕ_e represents the average number of bits transmitted over e. Note that the objective similar to (1) has been considered in [10–14] as well. The most relevant objective associated with (1) is the *energy consumption*. To see this, let us define weight $w_e := k_e d_e^{-\alpha}$ where d_e is the distance between nodes connected by Link e, α is the path loss exponent, and k_e is the related channel parameter. Hence (1) is proportional to the total transmitted energy consumed throughout the data aggregation. Note in [13, 14], the authors consider the same energy cost function. We refer to ϕ_e as the *aggregation cost function* (we will use notation ϕ to denote the cost function in general, whereas ϕ_e is used to denote the cost function specifically on Link e). Note that ϕ_e depends on the source measurements aggregated on e, and also on $g(\cdot)$ which is the summary function applied to the measurements. The work in [15] also studies an aggregation problem in sensor networks computing summary functions, assuming that all the packets generated in the network have the same size. However, the amount of information generated at intermediate nodes may vary, since a summary of data can be statistically different from the original data, which is our key observation.

Let us take an example. Consider the network in Figure 1 where Nodes 1 and 2 are the source nodes, and the node in shaded color represents the sink. The sink wants to receive a summary of information from Nodes 1 and 2. The sensor readings generated at Nodes 1 and 2 are represented by the random variables (RV) X_1 and X_2, respectively. Since Node 1 is a "leaf" node, Node 1 will simply transmit the raw reading X_1 to Node 2. Node 2 will combine X_1 with its own data, X_2, by computing the summary function $g(X_1, X_2)$ which is then transmitted to the sink. We define the aggregation cost function ϕ as follows. Suppose the sensor information to be transmitted on Edge e is random variable Y. The average number of bits to be transmitted on e, or ϕ_e, is defined as (We temporarily ignore communication overheads incurred in addition to the sensor information, e.g., the packet header size. We will however take such overheads into account later when we formally define ϕ.)

$$\phi_e = H(Y), \tag{2}$$

where $H(\cdot)$ denotes the entropy function. Note that the entropy function has been also adopted as cost function in [10, 12], and throughout this paper we will define ϕ in terms of $H(\cdot)$. The average numbers of bits transmitted on Edges 1 and 2 are, respectively, given by

$$\phi_1 = H(X_1), \qquad \phi_2 = H(g(X_1, X_2)). \tag{3}$$

Suppose $g(\cdot)$ is given by `sum`. Since $H(g(X_1, X_2)) = H(X_1 + X_2) \neq H(X_1)$, the costs incurred at Edges 1 and 2, that is, ϕ_1 and ϕ_2, are different. If we had used other types of $g(\cdot)$, such as `max`, we would have that $\phi_2 = H(\max(X_1, X_2))$ which would incur different cost from the case where $g(\cdot)$ was `sum`. In many cases we will assume symmetric sources; that is, ϕ depends only on the number of sensor readings to which $g(\cdot)$ is applied. In those cases we will treat ϕ as a function $\phi : \mathbb{Z}_+ \to \mathbb{R}_+$; that is, $\phi(m) = H(g(X_1, \ldots, X_m))$ (we will also examine the cases of asymmetric sources as well). We will show that $g(\cdot)$ determines the properties of $\phi(\cdot)$ such as convexity and monotonicity, and the structure of the aggregation problem heavily depends on those properties. Hence the aggregation scheme must be designed to capture key aspects of aggregation cost functions under the given summary function. The abovementioned links among summary functions, cost functions, and optimal aggregation strategies have not been previously well studied, as we will see in Section 2 through reviewing related works.

Contributions. In this paper we investigate the minimum energy aggregation problem for several widely used summary functions. We consider two types of $g(\cdot)$. The first type is called the *summation* type which involves sums of measurements: specifically `sum`, `mean`, and `weighted sum`. The second type is called the *extreme* type which is related to the extreme statistics of the data: specifically `max` and `min`. We will use the entropy function as the measure of information rate. We show that, when $g(\cdot)$ is `sum` or `mean`, and if the source data is i.i.d., ϕ is indeed concave and increasing, *irrespective of the distribution of the source data*. This implies that one can use well-known algorithms such as the Hierarchical Matching (HM) algorithm [16] in order to approximate the optimal cost. When $g(\cdot)$ is `weighted sum` however, it is unclear how we make association between the flow aggregation problem and the cost function. Nonetheless we prove that, if the source data is independent Gaussian random variables, there exists an efficient algorithm for the problem of aggregating `weighted sum` of data with arbitrary weights.

Next we consider extreme type summary functions such as `max`. We will show that for certain distributions of source data, ϕ can be *convex* and *decreasing* in the (nonzero) number of aggregated measurements. Note that the single-sink aggregation problems for concave/increasing [16–20] or convex/increasing cost functions [21, 22] have been widely studied, however convex and decreasing ϕ has not been well studied yet. We propose a novel algorithm which effectively captures such properties of ϕ. We begin by observing that there are two aspects in cost reduction as follows. Since ϕ is convex and decreasing, ϕ decreases faster when the number of aggregated data is smaller. The intuition is that it pays to locally aggregate data among nearby sources in the *early* stages of aggregation, that is, when the number

of measurements aggregated at sensors is small. This leads us to find a low-cost *local* clustering of sources, which is a "microscopic" aspect of cost reduction. Meanwhile we need to simultaneously find a low-cost route to the sink, which must take the *global* structure of the network into account and thus is a "macroscopic" aspect of cost reduction. These are conflicting aspects and a good tradeoff point between them should be sought. To that end we propose Hierarchical Cover and Steiner Tree (HCST) algorithm. The algorithm consists of multiple stages and is designed to empirically find the best tradeoff point over the stages. We show that, by simulation, the algorithm can significantly reduce cost compared to baseline schemes such as a greedy heuristic using shortest path routing, or the HM algorithm.

Our results show that the summary function $g(\cdot)$ can significantly impact the design of aggregation schemes. However there are many choices for $g(\cdot)$: suppose for example, we would like to compute L_p norm of the vector of measurement data. Note `sum` and `max` functions which we study in this paper are in fact related such that, if the measurement data is always positive, then `sum` function is simply L_1 norm and `max` is L_∞ norm of a data vector. One could ask: what are good aggregation strategies if we take $g(\cdot)$ as a different L_p norm, say L_2 norm? We leave such questions as future work.

Paper Organization. We briefly review related work in Section 2. Section 3 introduces the model and problem formulation. Sections 4 and 5 discuss the optimal routing problem for summation and extreme type summary functions, respectively. Simulation results are presented in Section 6. Section 7 concludes the paper.

2. Related Work

In general the single-sink aggregation problem to minimize (1) is NP-hard [23], and a substantial amount of research has been devoted to designing approximated algorithms depending on certain properties of ϕ. In our case it is important to note that such properties of ϕ are determined by the choice of $g(\cdot)$. Let us briefly review the related work on the single sink aggregation problem for two types of ϕ. Most research on the single sink aggregation problem has focused on the case where ϕ is *concave* and *increasing*. Due to the concavity of ϕ, the link costs associated with the amount of aggregated data exhibit *economies of scale*, that is, the marginal cost of adding a flow at a link is cheaper when the current number of aggregated flows is greater at the link. Buy-at-bulk network design [23, 24] is based on such property of ϕ. A number of approximation algorithms have been proposed, for example, [17–19]. When ϕ is known in advance, a constant factor approximation to the optimal cost is possible [20, 25]. Even when ϕ is unknown but is concave and increasing, Goel and Estrin [16] have proposed a simple and randomized algorithm called *Hierarchical Matching* (HM). The algorithm computes minimum weight matchings of the source nodes hierarchically over stages, and outputs a tree for aggregation. HM algorithm can approximate the optimal cost by a factor logarithmic in the number of sources [16]. Nonuniform variants of this problem such that ϕ differs among the links are also studied [26, 27] in which a polylogarithmic approximation to the optimal cost is shown to be achievable.

The case where ϕ is *convex* and *increasing* in the number of aggregated measurements has been studied in [21, 22]. Here ϕ exhibits (dis)economies of scale, that is, the marginal cost of routing a flow at a link is more expensive when a greater number of flows are aggregated at the link. Such phenomenon can be observed from many applications, such as speed scaling of microprocessors modeled by $\phi(m) = \beta m^a$ where m is the clock speed, $\beta \geq 0$ and $a \geq 1$ are constants, and $\phi(m)$ is the energy consumption at the processor. Notably the authors show that the problem can intrinsically differ from that for a concave and increasing ϕ. For example the authors show that constant-factor approximation algorithms do not exist for certain convex and increasing ϕ [21]. They nevertheless proposed a constant factor approximation algorithm for the case $\phi(m) = \beta m^a$. These results show that the single-sink aggregation problem crucially depends on certain properties of ϕ such as convexity. However, none of the above works deal with convex and decreasing ϕ which we will study in the sequel.

There have been many studies regarding the intermediate data combining in conjunction with routing in order for an efficient retrieval of the complete sensor readings. Scaling laws for achievable rates under joint source coding and routing are studied in [28]. The work [11] studies the problem of minimizing the flow costs under distributed source coding. They show that when $\phi(m)$ is linear in m, firstly applying Slepian-Wolf coding at the sources, and secondly routing coded information via shortest path tree from the sources to the sink is optimal. In [10] a single-input coding model was adopted in which the coding of information among the nodes can be done only in pairs, but joint coding of the source data from more than two nodes is not allowed. Assuming reduction in packet size is a linear function of correlation coefficient between each pair of nodes, they proposed a minimum-energy routing algorithm. The impact of spatial correlation on routing has been explored in [12]. They showed that, assuming the correlation decays over distance, it pays to form clusters of nearby nodes and aggregate data at the clusterheads. The aggregated information is then routed from clusterheads to the sink. The algorithm is shown to perform well for various correlation models. The tradeoff between integrity of aggregated information and energy consumption has been studied in [29]. Further works on in-network aggregation combined with routing include [30, 31] which propose efficient protocols for routing excessive values among sensed data. A scheme using spatially adaptive aggregation so as to mitigate traffic congestion was proposed in [32].

The above works aim at retrieving the *entire* set of data, instead of a summary, subject to certain degrees of data integrity. In our case, we design energy efficient aggregation schemes to compute the summary function $g(\cdot)$ of the sensor readings. Also in the above mentioned works, the in-network aggregation reduces cost mainly by removing correlation among the data set. In our work, by contrast, we will focus on losslessly retrieving a summary of *statistically independent* sensor readings. We assume the independence of sensor

readings because we would like to *decouple* the cost savings from removing correlation, and the savings from applying the summary function in association with aggregation strategies; we focus on the latter. Moreover, the assumption of the independence among the readings represents the "worst" case in terms of cost savings, since one cannot reduce the energy cost by removing correlations in sensor readings. In fact, the independence assumption can be valid in certain cases. For example, consider a large sensor network assuming that the sensed data is spatially correlated and the correlation decays quickly over distance. If the source nodes are sparsely deployed and thus tend to be far apart from one another, the correlation among their data can be very weak. Obviously such sparse node placement is motivated by cost efficiency: sparse placement of nodes enables us to reap as much information given a fixed number of sensor devices, assuming that the network senses a homogeneous field and the measure of information is given by the joint entropy function.

3. Model

3.1. Preliminaries. We are given an undirected graph $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ and $E \subseteq V \times V$ denote the set of vertices and edges, respectively. For $u, v \in V$, $(u, v) \in E$ denotes the (undirected) edge connecting nodes u and v. For each edge in E we associate a *weight* defined by $w : E \to \mathbb{R}_+$. A weight captures the cost of transmitting unit amount of data between two nodes, for example, expenditure of transmission energy in order to compensate path loss. The set $S \subset V$ denotes the set of source nodes, that is, the nodes which generate measurement data to be reported to the sink. Also define $\eta := |S|$ where $|\cdot|$ denotes the cardinality of a set. For a source node $u \in S$, its measured data is modeled by an RV denoted by X_u. We assume that X_u's are independent and identically distributed among the sources. The measured data is to be aggregated at the sink node denoted by $t \in V$. The nodes which are not source nodes act as relays in the aggregation process. For simplicity we will assume that any node in the network transmits data at most once during the aggregation process. Such an assumption has been made in other works such as [15]. Thus the routes for aggregation constitute a tree whose root is given by t. We refer to such tree as an *aggregation tree*. The aggregation process is performed as follows. The sources initiate transmissions. An intermediate node waits for all the data from the sources which are descendants of the node to arrive. Next the node computes the summary function of the aggregated data which is then relayed to the next hop.

In this paper a summary function is defined to be a nonnegative function denoted by $g(\cdot)$ which is a *divisible* function. Divisible functions are a class of summary functions which can be computed in a divide-and-conquer manner [1]. Divisible functions are defined as follows: given n data samples, consider a partition of the samples into sets of size k and $n - k$ denoted by $\{x_1, x_2, \ldots, x_k\}$ and $\{x_{k+1}, \ldots, x_n\}$, respectively. If $g(\cdot)$ is divisible, $g(g(x_1, \ldots, x_k), g(x_{k+1}, \ldots, x_n)) = g(x_1, x_2, \ldots, x_n)$ holds for any n and k. Examples of divisible functions are `sum`, `max` and `min`. Particularly when $g(\cdot)$ is divisible, the aggregation can be performed in a divide-and-conquer manner as follows. Suppose a set of data samples are aggregated at a node. If the node is a source, it applies $g(\cdot)$ to the collected samples and its own data. If the node is simply a relay, it applies $g(\cdot)$ to the aggregated data samples to obtain a summary of the samples where the summary of its aggregated data is transmitted to the next hop.

Abusing notation for the sake of simplicity, we let the function $g(\cdot)$ take a set, a vector, or their combination as its argument. For example if $g(\cdot)$ is `sum`, $g(x, y) = x + y$, and $g(\{x, y\}) = x + y$, and also $g(\{x, y\}, z) = x + y + z$. For some $U \subseteq V$, we define X_U as the set of RVs representing the measurements from the nodes in U; that is, $X_U := \{u \in U \mid X_u\}$. Thus $g(X_U)$ is the aggregation function applied to the set X_U, for example, if $g(\cdot)$ is `sum` then

$$g(X_U) = \sum_{u \in U} X_u. \tag{4}$$

3.2. Problem Formulation. We will define the problem of minimizing communication costs as follows. There exists a sink to which the data is to be aggregated. Our goal is to find a minimum-cost aggregation tree $\mathcal{T} = (V', E') \subseteq G$ rooted at the sink. We would like to solve the following aggregation problem:

$$(\mathbf{P}) \quad \underset{\mathcal{T}=(V',E') \subseteq G}{\text{Minimize}} \sum_{e \in E'} w_e \phi_e, \tag{5}$$

where ϕ_e represents the average number of bits communicated over Edge e. Note that the objective of (5) has been considered in the works [10, 12] as well. We call ϕ_e as *aggregation cost function* which we define as follows.

We will use the entropy function $H(\cdot)$ as our measure of information rate similar to works [13, 14]. We assume that the average number of bits to represent random sensor measurement X is given by $H(X)$. A precise definition of the entropy function $H(X)$ depends on the nature of X: if X is a discrete RV, $H(X)$ denotes the usual Shannon entropy. If X is a continuous RV, $H(X)$ is implicitly defined to be $H(\widehat{X})$ where $\widehat{X}$ is a discrete RV obtained by applying uniform scalar quantization to X with some quantization step size, say 2^{-b} for some integer $b > 0$. If the quantization precision is sufficiently high, it is known [33] that $H(\widehat{X}) \approx b + h(X)$ where $h(\cdot)$ denotes the differential entropy of continuous RVs. Note that a similar approximation has been made in defining the information rates for continuous RVs in [13, 14]. Hence in this paper, we will assume that continuous RV X incurs the cost of $b + h(X)$ bits where $b > 0$ is a sufficiently large parameter, and we denote such costs by $H(X) := b + h(X)$.

In addition, the measured data is transmitted as a packet in the network. Hence for each packet transmission, there is an overhead of metadata, for example, packet header. For any measurement Y, no matter how small $H(Y)$, there is always an overhead of transmitting such metadata in practice. We will assume the header length is fixed to $\alpha > 0$ bits throughout this paper. Hence the average number of bits required to send

measurement information Y per transmission over a link is given by

$$\alpha + H(Y). \tag{6}$$

For a given aggregation tree $\mathcal{T} = (V', E')$, let $p(s) \subset E'$ denote the path from a source $s \in S$ to the sink. For a given Edge $e \in E$, let $U(e) \subseteq S$ denote the set of source nodes whose aggregated measurements are transmitted over e, that is, $U(e) = \{s \in S \mid e \in p(s)\}$. The information to be communicated over Edge e is the function $g(\cdot)$ applied to the set of measurement value from $U(e)$, that is, $g(X_{U(e)})$. Hence we define the aggregation cost function as follows:

$$\phi_e = \begin{cases} 0 & U(e) = \emptyset \\ \alpha + H\left(g\left(X_{U(e)}\right)\right) & \text{otherwise.} \end{cases} \tag{7}$$

We would like to solve (**P**) using the definition of ϕ_e given by (7). In the following sections we investigate several widely used summary functions and the associated optimal aggregation problems.

4. Aggregation Schemes for Summation-Type Summary Functions

We consider the summary functions of `sum`, `mean`, and `weighted sum`.

4.1. `sum` *and* `mean`. We first discuss the case where $g(\cdot)$ is `sum`. We have that

$$H\left(g\left(X_{U(e)}\right)\right) = H\left(\sum_{u \in U(e)} X_u\right). \tag{8}$$

Clearly `sum` is a divisible function. Thus the aggregation process is as follows: a node simply applies `sum` function to the aggregated data, and relays the aggregated information to the next hop.

When the source data is i.i.d., we will show that there exists a randomized algorithm which finds an aggregation tree whose expected cost is within a factor of $(\log(\eta) + 1)$ of the optimal cost to (5).

Proposition 1. *Suppose X_i's are i.i.d. For any distribution of X_i, there exists an algorithm yielding the mean cost within a factor of $\{\log(\eta) + 1\}$ of the optimal cost of* (**P**).

Goel and Estrin [16] studied a single-sink data aggregation problem as follows. A source generates a unit flow which needs to be routed to a sink where the flows are aggregated though a tree. Their objective is to minimize the following cost function:

$$\sum_{e \in E} w_e \psi(m_e), \tag{9}$$

where w_e is the weight on Edge e, m_e is the number of flows on Edge e and $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ is a function that maps the total size of flow to its cost. They proposed an algorithm to minimize (9) when ψ is a canonical aggregation function defined as follows.

Definition 2 (see [16]). The function $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ is called a canonical aggregation function (CAF) if it has the following property:

(1) $\psi(0) = 0$.

(2) $\psi(\cdot)$ is increasing.

(3) $\psi(\cdot)$ is concave.

Their algorithm, called Hierarchical Matching (HM) [16], guarantees the mean cost to be within the factor of $\log(\eta) + 1$ of the optimal *irrespective of* ψ, provided that ψ is a CAF. As mentioned previously, since X_i's are i.i.d., ϕ_e depends only on $m_e := |U(e)|$. Specifically we will define ϕ as follows:

$$\phi_e = \phi(m_e) = \begin{cases} \alpha + H\left(\sum_{u \in U(e)} X_u\right), & m_e > 0, \\ 0, & m_e = 0. \end{cases} \tag{10}$$

We will show that $\phi(\cdot)$ is a CAF by showing that $\phi(\cdot)$ satisfies the three properties of Definition 2. Note this implies that HM algorithm can be used to approximately solve (**P**), since (9) and the objective of (**P**) are identical.

Proof of Proposition 1. For the first property, it trivially holds that $\phi(0) = 0$. For the second property, for any two independent RVs X_1 and X_2, it is known that $H(X_1 + X_2) \geq H(X_1)$ implying that $\phi(2) \geq \phi(1)$, that is, the sum of independent RVs always increases entropy [33], which implies that $\phi(m)$ is increasing in m. For the third property, consider the following. It is shown in [34] that the entropy of the sum of independent RVs is a submodular set function. That is, the following holds for independent RVs Y_1, Y_2 and Y_3 [34, Theorem I]:

$$H(Y_1 + Y_3) + H(Y_2 + Y_3) \geq H(Y_1 + Y_2 + Y_3) + H(Y_3). \tag{11}$$

Now consider $m + 2$ sensor measurements $X_1, \ldots, X_{m+2}$, and make substitutions $Y_1 := X_{m+1}$, $Y_2 := X_{m+2}$, and $Y_3 := \sum_{i=1}^{m} X_i$ in (11). We have that

$$\begin{aligned} &H\left(X_{m+1} + \sum_{i=1}^{m} X_i\right) + H\left(X_{m+2} + \sum_{i=1}^{m} X_i\right) \\ &\quad \geq H\left(\sum_{i=1}^{m+2} X_i\right) + H\left(\sum_{i=1}^{m} X_i\right). \end{aligned} \tag{12}$$

If we apply the definition of ϕ given by (10) to (12), the following holds due to symmetry:

$$\phi(m+1) + \phi(m+1) \geq \phi(m+2) + \phi(m). \tag{13}$$

Hence $\phi(m+2) - \phi(m+1) \leq \phi(m+1) - \phi(m)$ holds, or the slope is decreasing in m, which implies that $\phi(\cdot)$ is concave on the domain of integers. Thus $\phi(\cdot)$ satisfies all the properties of Definition 2, and is a CAF. This implies that, by using HM algorithm, one can achieve the expected cost which is within the factor of $1 + \log(\eta)$ of the optimal cost of (**P**). □

Next we consider mean as the summary function. Note that mean, as well as weighted sum considered in the next section, are not divisible functions in general. We will nevertheless show that the problem for those summary functions can be reduced to sum problem as follows. Suppose every source node is aware of the total number of the sources, that is, η. In our scheme every source simply scales its measurement by η^{-1} prior to transmission, that is, Source i transmits $\eta^{-1}X_i$, then such scaled measurements are aggregated in a similar way as the sum problem. The average number of bits transmitted over Edge e can be written as $\alpha + H\{\sum_{u\in U(e)}((1/\eta)X_u)\}$. Since $((1/\eta)X_u)$'s are i.i.d., for the minimum cost aggregation problem for mean we can use the same algorithm as that used for sum, for example, HM algorithm.

4.2. weighted sum. Next we consider the case where $g(\cdot)$ is weighted sum as follows. We assign arbitrary weights a_i, $i \in S$, to the source nodes. The goal of the sink is to compute $\sum_{i\in S} a_i X_i$. Our method of aggregation is similar to that for the case of mean, that is, Source i scales its measurement by a_i, then transmits $a_i X_i$ where the aggregation process is the same as that for sum. However the effective source data $a_i X_i$ seen by the network is no longer i.i.d., unless a_i's are identical for all $i \in S$. The aggregation cost function is given by

$$\phi_e = \alpha + H\left(\sum_{i\in U(e)} a_i X_i\right). \tag{14}$$

The difficulty lies in it is difficult to associate a "flow" with the source data $a_i X_i$ due to asymmetry, that is, the problem is no longer a flow optimization. Moreover, it is easily seen that (14) is not a CAF in general. Thus we restrict our attention to a specific distribution of X_i. We will show that, if X_i are independent Gaussian RVs, the problem for weighted sum is indeed a single-sink aggregation problem with concave costs, and there exist algorithms similar to HM algorithm which have good approximation ratio. Specifically we prove that our problem is equivalent to the single-sink aggregation/flow optimization problem with *nonuniform* source demands.

Proposition 3. *Suppose $X_i \sim N(\mu_i, \sigma_i^2)$, and $X_1, \ldots, X_\eta$ are independent. Let $g(\cdot)$ be* weighted sum *with arbitrary weights $a_1, \ldots, a_\eta$. For sufficiently large b, there exists an algorithm yielding the mean cost within a factor of $\{\log(\eta) + 1\}$ of the optimal cost of* (**P**).

Proof. Consider the information communicated over Edge e denoted by Y:

$$Y = \sum_{i\in U(e)} a_i X_i. \tag{15}$$

Since X_i's are independent Gaussian RVs, Y is also Gaussian with variance σ_e^2 where $\sigma_e^2 := \sum_{i\in U(e)} a_i^2\sigma_i^2$. Thus the differential entropy of Y is given by

$$h(Y) = \frac{1}{2}\log\left(2\pi e \sum_{i\in U(e)} a_i^2\sigma_i^2\right). \tag{16}$$

We observe that, from (16), we can treat $a_i^2\sigma_i^2$ as the "flow" generated by Source i, and the sum of flows at Edge e incurs the entropy cost as in (16). Specifically we will make the following definitions:

$$f_i := a_i^2\sigma_i^2, \quad i \in S, \tag{17}$$

$$f^* := \min_{i\in S} f_i, \tag{18}$$

$$\psi(x) := \alpha + b + \frac{1}{2}\log(2\pi e x), \quad x \geq 0, \tag{19}$$

$$\phi(x) := \begin{cases} \psi(x), & x \geq f^*, \\ \dfrac{\psi(f^*)}{f^*}x, & 0 \leq x < f^*. \end{cases} \tag{20}$$

Here f_i represents the (unsplittable) flow demand generated by Source i, and f^* denotes the minimum demand. Hence under a flow routing scheme, the total amount of flow at Link e is given by $\sum_{i\in U(e)} f_i$. Then from (16), the associated communication cost incurred at Link e is given by $\psi(\sum_{i\in U(e)} f_i)$ bits, that is, $\psi(\cdot)$ represents the information rate of a flow aggregated at Link e. Unlike the previously defined cost functions, ψ is no longer a function of the number of sources on a link, but instead the function of the amount of flow on that link. Finally we define the aggregation cost function ϕ in terms of ψ as in (20) in order to meet the concavity condition for ϕ as follows: ϕ is essentially identical to ψ, and if

$$b \geq \frac{1}{2}\log\left(\frac{1}{2\pi e f^*}\right), \tag{21}$$

one can show that $\phi(x)$ is concave and increasing for all $x \geq 0$. Hence under the condition (21), ϕ is an increasing concave function of the total flow aggregated on a link. In that case we can use the algorithm proposed by Meyerson et al. [19] which essentially extends the HM algorithm to the problems with nonuniform source flow demands, and can approximate the optimal cost by a factor of $\log(\eta) + 1$ on average. □

In summary, the key question was whether (**P**) can be cast as a flow aggregation problem, if $g(\cdot)$ is weighted sum. In general, it is difficult to make such association due to asymmetry; however, we revealed that such formulation is possible for independent Gaussian sources.

4.3. Discussions. Note that some properties regarding X_i's such as the submodularity relation in (11), used to show that ϕ is a CAF rely heavily on the independence of X_i's. When X_i's are correlated, we can find examples of ϕ which are not CAF for the summary function of sum as follows. Let X_1 and X_2 be jointly Gaussian with the same marginal given by $N(0, 1)$ with $\mathbb{E}[X_1X_2] = \rho$. Then $X_1 + X_2$ is distributed according to $N(0, 2(1 + \rho))$, thus we have that, if $\rho < -0.5$, then

$$\begin{aligned} h(X_1 + X_2) &= \frac{1}{2}\log(2\pi e \cdot 2(1+\rho)) \\ &< \frac{1}{2}\log(2\pi e) = h(X_1). \end{aligned} \tag{22}$$

Thus the entropy function does not satisfy the second condition of Definition 2, that is, the increasing property, as a CAF. Hence for arbitrarily correlated sources, presumably few meaningful arguments can be made on optimal aggregation problems, even for simple summary functions such as `sum`.

The discussion so far enables us to deal with more general objective functions extended from (**P**). Consider a function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ which is concave and increasing. We now define communication overhead on an edge as the function γ of the average number of bits transmitted over the edge. Namely, we consider the following extension of (**P**):

$$\left(\mathbf{P}'\right) \quad \underset{\mathcal{T}=(V',E')\subseteq G}{\text{Minimize}} \sum_{e\in E'} w_e \gamma\left(\phi_e\right). \tag{23}$$

Consider ($\mathbf{P}'$) for the summary function `sum` for i.i.d. sources and `weighted sum` for independent Gaussian sources. Note that the composition of two concave and increasing functions is also concave and increasing [35]. Thus $\gamma(\phi_e)$ is a concave and increasing function of the amount of flows at an edge, and thus is a CAF. Hence HM algorithm can be used to approximate ($\mathbf{P}'$).

5. Aggregation Schemes for Extreme-Type Summary Functions

5.1. Case Study. In this section we consider summary functions regarding the extreme statistics of measurements, that is, `max` or `min`. We will first investigate the entropy of the extreme statistics of a set of RVs. Consider m measurements denoted by X_i, $i = 1, \ldots, m$. Since $(\max_{1\le i\le m} X_i) = -\{\min_{1\le i\le m}(-X_i)\}$, we will focus only on `max` without loss of generality. It is easily seen that `max` function is divisible, thus the aggregation process is similar to that for `sum`: a node simply applies `max` function to the aggregated data. For example, suppose a node receives data given by $X_1, \ldots, X_m$. The node simply computes $(\max_{i=1,\ldots,m} X_i)$ and forwards it to the next hop.

For extreme-type summary functions, we will show that ϕ is in general *not* a CAF. In particular we consider several cases of practical importance.

Case 1 (Gaussian RVs). We consider the problem of retrieving the maximum of i.i.d. Gaussian RVs. We assume that $X_i \sim N(0, 1)$ for $i \in S$ where we again assume that $\phi(m) = \alpha + b + h(\max_{i=1,\ldots,m}[X_i])$ for $m \ge 1$ and some constant b. We provide a numerical evaluation of $h(\max_{i=1,\ldots,m}[X_i])$ on the left of Figure 2. We observe that $\phi(m)$ is strictly *convex* and *decreasing* in m for $m \ge 1$, thus ϕ is not a CAF.

Case 2 (Extreme data retrieval problem). We consider the problem of *extreme data retrieval* defined as follows. Assume that a source node $i \in S$ measures some physical quantity which is distributed according to a continuous RV Y_i. We assume Y_i's are independent but not necessarily identically distributed. Suppose with some probability Y_i is equal to a large number, which indicates an "abnormal" event. An important application of sensor networks is to detect the maximum *abnormality* among the measurements. The abnormality is defined as how far a sensor's measurement has deviated from its usual statistics as follows. Let us denote the cumulative distribution function (CDF) of Y_i by $F_i(\cdot)$ or $\mathbb{P}(Y_i \le y) = F_i(y)$, $i \in S$. Consider realizations of $Y_1, \ldots, Y_\eta$ given by $y_1, \ldots, y_\eta$. We will quantify the abnormality at Source i in terms of how *unlikely* the measurement y_i is: specifically the goal of the sink is to retrieve $\min_{i\in S}[\mathbb{P}(Y_i > y_i)]$, or alternatively,

$$\max_{i\in S}\left[F_i\left(y_i\right)\right], \tag{24}$$

thus the abnormality of y_i is defined by $F_i(y_i)$. Let $X_i = F_i(Y_i)$. We will assume that the nodes transmit and aggregate X_i instead of Y_i, and the goal of the sink is to retrieve $\max_{i\in S}\{X_i\}$. Note since $X_i = F_i(Y_i)$ is the RV evaluated at its distribution function, one can show that X_i's are i.i.d. RVs uniformly distributed on $[0, 1]$. Thus the problem reduces to an optimal aggregation problem retrieving `max` of i.i.d. uniform RVs.

We will show that ϕ associated with the extreme data retrieval problem is *convex* and *decreasing* function when the number of aggregated measurements is greater than or equal to 2. Suppose the data aggregated at a node is given by $X_1, \ldots, X_m$ and define $Z_m := \max_{i=1,\ldots,m}[X_i]$. As previously we assume that the node requires on average $\phi(m) = \alpha + b + h(Z_m)$ bits to transmit Z_m.

Proposition 4. *Consider the extreme data retrieval problem. The aggregation cost function $\phi(m)$ is convex and decreasing for $m \ge 2$.*

Proof. Since Z_m is the maximum of m i.i.d. uniform RV's, the CDF of Z_m denoted by $F_{Z_m}(\cdot)$ is given by

$$F_{Z_m}(z) = z^m. \tag{25}$$

Thus the probability density function (pdf) of Z_m denoted by f_{Z_m} is given by mz^{m-1}. If we compute $h(Z_m)$,

$$\begin{aligned} h\left(Z_m\right) &= -\int_0^1 f_{Z_m}(z)\log\left\{f_{Z_m}(z)\right\}dz \\ &= -\int_0^1 mz^{m-1}\log\left(mz^{m-1}\right)dz = -\log(m) + \frac{m-1}{m}. \end{aligned} \tag{26}$$

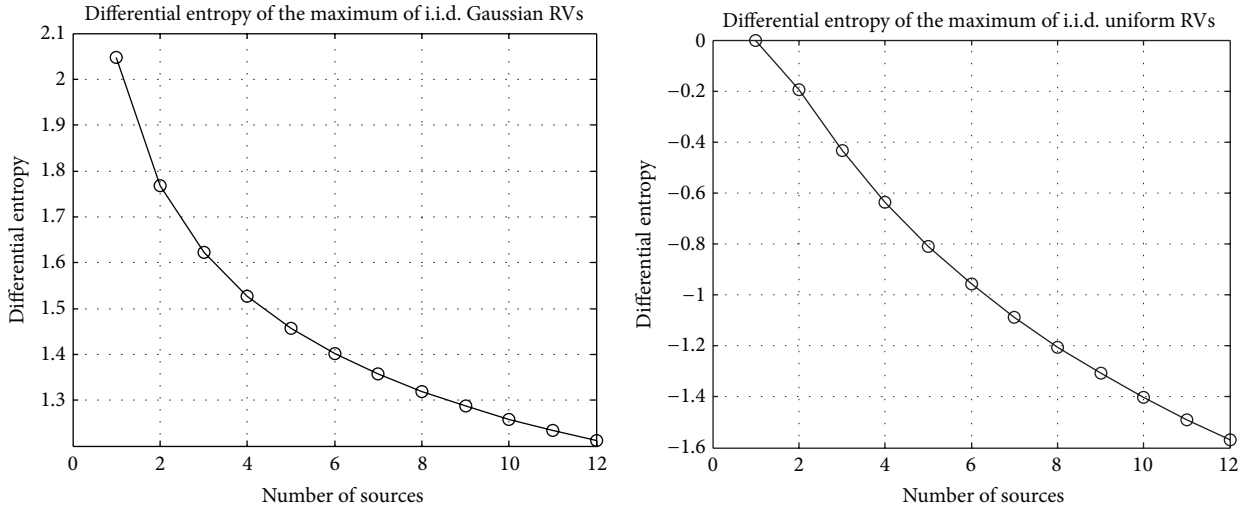
Thus

$$\phi(m) = \begin{cases} \alpha + b - \log(m) + 1 - m^{-1}, & m \ge 1, \\ 0, & m = 0. \end{cases} \tag{27}$$

By regarding m as a continuous variable, we have that, for $m \ge 1$,

$$\frac{d\phi}{dm} = -\frac{1}{m} + \frac{1}{m^2}, \qquad \frac{d^2\phi}{dm^2} = \frac{1}{m^2} - \frac{2}{m^3} = \frac{1}{m^2}\left(1 - \frac{2}{m}\right). \tag{28}$$

Clearly $\phi(m)$ is decreasing for $m \ge 1$, and since its second order derivative is nonnegative for $m \ge 2$, $\phi(m)$ is convex for $m \ge 2$. □



(a) (b)

Figure 2: The differential entropy of the maximum of a set of i.i.d. RVs distributed according to an RV X. On the left, $X \sim N(0, 1)$, and on the right, X is uniform on $[0, 1]$.

On the right of Figure 2 the plot of $h(Z_m)$ is shown. Note $h(Z_m)$ is strictly convex for $m \geq 2$, but overall appears to be approximately convex. Note that $h(Z_m)$ is nonpositive, thus one could select a sufficiently large b such that $\phi(\eta) = \alpha + b + h(Z_\eta) \geq 0$, so that $\phi(m) \geq 0$ for all $1 \leq m \leq \eta$.

In general, for a convex and decreasing ϕ, (**P**) is clearly NP-hard since the problem contains the Steiner tree problem as a special case. In the following section we present a novel algorithm which captures key properties of convex and decreasing ϕ. Later we show by simulation the algorithm effectively achieves low cost.

5.2. Algorithm for Convex and Decreasing Aggregation Cost Functions

5.2.1. Motivation. Before we describe our algorithm we present the motivation behind the algorithm. An important observation for the data aggregation problems was made in [25] for *concave* and *increasing* ϕ. They proposed a "hub-and-spoke" model for so-called *facility location problem*. The idea is that when ϕ is concave and increasing, one should first aggregate flows to some "hubs," then route the aggregated flow from the hubs to the sink at the minimum cost; this is done by building an approximately optimal Steiner tree where the hubs (facility locations) are the Steiner nodes. The rationale is that, once multiple flows are aggregated at hubs, the cost of routing them collectively to the sink is cheaper than routing the sources' flows separately, due to the concavity of ϕ. We observe two aspects in such hub-and-spoke schemes. Firstly by local aggregation of flows at hubs we aim at greedily reducing costs based on local information, which we view as the *microscopic* approach to reduce cost. Secondly by building an approximately optimal Steiner tree with respect to the hubs and the sink, we take the *global* network structure into account, which can thus be seen as the *macroscopic* aspect for cost reduction. Hence there exists a tradeoff between microscopic and macroscopic aspects of the cost reduction. A similar observation on such tradeoff was made in [12]. However our key question is that, how do we achieve an optimal tradeoff between those aspects for a *convex* and *decreasing* ϕ?

Consider the three examples of aggregation cost functions denoted by ϕ_1, ϕ_2, and ϕ_3 which are decreasing and convex for $m \geq 1$ as shown in Figure 3. In case of ϕ_1, we see that ϕ_1 is flat for $m \geq 1$, that is, the average number of bits communicated over a link is constant irrespective of the number of flows passed through it. Thus, the minimum cost routing problem reduces to a Steiner tree problem, in which case a completely "macroscopic" solution is optimal. In case of ϕ_2, we see that ϕ_2 decreases slowly in m. Thus, the more number of flows merges at a link, it takes the less number of bits to transmit the merged information. Suppose we use the hub-and-spoke scheme to aggregate flows in a local manner. The amount of aggregated flows at a hub is at least 2: note that however, ϕ_2 is approximately "flat" for $m \geq 2$. This implies that, once more than two flows are aggregated, the benefits from further local flow aggregation will be negligible. Hence the optimal routing problem from the hubs to the sink approximately reduces to the Steiner tree problem! Thus one could expect that local aggregation (microscopic approach) followed by an optimal Steiner tree construction (macroscopic approach) would yield a good solution. Now let us consider ϕ_3. The overall rate of decrease of ϕ_3 is higher than that of ϕ_2. It appears that when the number of aggregated flows is significantly high, for example, m is greater than 6, ϕ_3 becomes effectively "flat." This suggests that, one should keep aggregating flows until sufficient amount of flows, say 6, is aggregated, that is, the microscopic cost reduction should be applied for *multiple times in a hierarchical manner*, then build an optimal Steiner tree with respect to the aggregated sources, that is, applying macroscopic reduction.

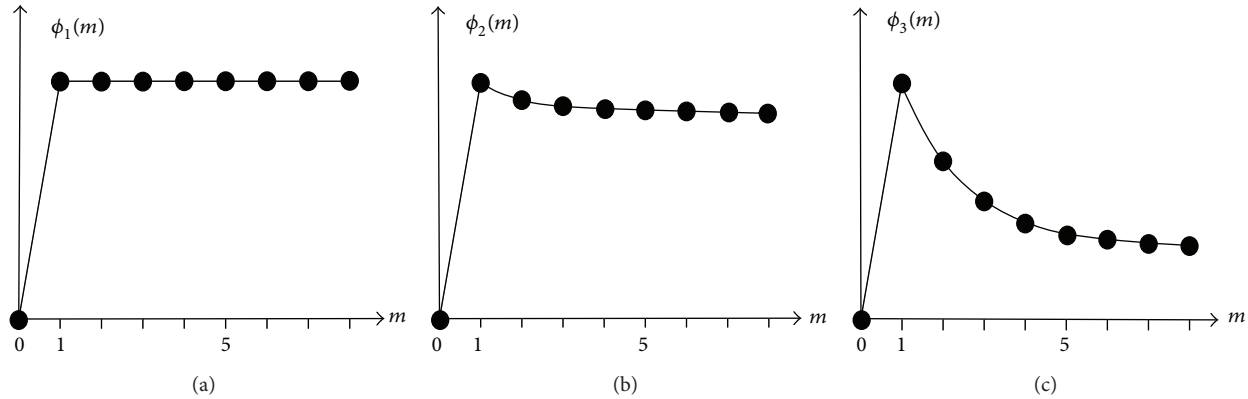


Figure 3: Aggregation cost functions which are convex and decreasing for $m \geq 1$.

The example provides us with some insights. Since $\phi(m)$ is convex decreasing, the marginal benefit of local aggregation is large for small m but decreases with increasing m. In other words, when m is small, that is, in the early stages of the overall aggregation process, one should focus on low-cost local aggregation in order to benefit from high rate of decrease of $\phi(m)$ for small m. Meanwhile, once a large number of flows are aggregated, it pays to perform macroscopic cost reduction from there on by building the optimal Steiner trees since ϕ becomes more "flat" with increasing m. This suggests that there exists a tradeoff point at which such microscopic and macroscopic reduction are optimally balanced. Unfortunately it is difficult to know such a tradeoff point in advance. The proposed algorithm not only exploits both the microscopic and macroscopic aspects of cost reduction for a convex and decreasing ϕ, but also empirically searches for the optimal tradeoff point. Details are presented in the following section.

5.2.2. Outline. An outline of the proposed algorithm is presented as follows. The algorithm consists of multiple stages. A hub-and-spoke problem (or facility location problem) is approximately solved at each stage. The flows from source nodes are merged at the hubs. The hubs at the present stage become the source nodes in the next stage, that is, the flows are merged hierarchically. Instead of solving complex facility location problem, we find a *minimum weight edge cover* (MWEC) on the source nodes at each stage as a simple approximation. The rationale is that we would like to cluster sources for local aggregation at low costs, and by definition the MWEC incurs low cost in doing that. MWEC consists of multiple connected components, each of which is a tree. For each connected component we select a source as a hub and call it a *center node* (details on the selection of center nodes are provided later). The flows in that component is aggregated at the center node.

At each stage, once the center nodes are determined, we build an approximately optimal Steiner tree with respect to the center nodes and the sink. We use algorithm in [36] for the Steiner tree construction. Their algorithm provides the best known ρ_S-approximation for Steiner tree problem where $\rho_S \approx 1.39$.

Each stage outputs an aggregation tree. The output tree at Stage i is the *union* of the paths from all the hierarchical aggregations found up to Stage i and the Steiner tree built at Stage i. Namely, the output tree at Stage i is a combination of i *consecutive* hierarchical aggregations (microscopic cost reduction) and a Steiner tree with respect to the sink and Stage i hubs (macroscopic cost reduction).

Hence, over the stages, the algorithm progressively changes the balance between microscopic and macroscopic aspects of cost reduction in the output trees. Roughly speaking, the output trees from later stages are more biased towards the microscopic aspect. After the stages are over, we pick the tree with the minimum cost among the output trees. As a result the algorithm empirically searches for the point of the "best" balance between the two aspects of cost reduction over the stages. Hence one could expect that our algorithm will work well for any convex and decreasing ϕ.

5.2.3. Algorithm Description. We present a formal description of the proposed algorithm followed by an explanation of further details. For given aggregation tree $\mathcal{T} \subseteq G$, let $c(\mathcal{T})$ denote the total energy cost associated with $\mathcal{T}$, as in the objective of (**P**).

Hierarchical Cover and Steiner Tree (HCST) Algorithm

Begin Algorithm

(1) (Metric completion of G) If G is not a complete graph, perform a metric completion of G to yield a complete graph. Namely, if there exist any pair of vertices without an edge, create an edge between the pair and assign the edge a weight which is the distance between the pair. The distance is measured in terms of the sum of the weights on the shortest path between the pair.

(2) (Initialization) $i \leftarrow 0$, $S_0 \leftarrow S$.

(3) (Initialize flows at sources) $f_u \leftarrow 1$, for all $u \in S$.

(4) (Initial output is a Steiner tree) Jump to Step 7.

(5) (Minimum weight edge cover) Let us denote the subgraph of G induced by S_{i-1} by G_s. Find a minimum-weight edge cover M_i in G_s. Let $C_i = (S_{i-1}, M_i)$ be the subgraph of G induced by the cover.

(6) (Node selection) Suppose C_i has ν connected components, and denote the jth connected component of C_i by $K_j = (V_j, E_j)$ for $1 \le j \le \nu$. For each K_j, select a node with the maximum degree (ties are arbitrarily broken), say u_j, which is called a *center node*. K_j is a tree, and u_j becomes the root of K_j. All the flows in K_j are aggregated at u_j such that every node transmits data to its parent node after the data from its child nodes has been aggregated at the node. The total flow at u_j is updated as follows:

$$f_{u_j} \longleftarrow \sum_{u \in V_j} f_u. \tag{29}$$

Remove all the noncenter nodes from S_{i-1}, and let S_i be the resulting set of source nodes.

(7) (Steiner tree construction) Build ρ_S-optimal Steiner tree T_i^s with respect to the source nodes in S_i and the sink, using the algorithm in [36].

(8) (Merging trees) If $i > 0$, merge all the MWECs found up to the present stage and the Steiner tree found in Step 7; that is, let

$$T_i \longleftarrow T_i^s \cup \left(\bigcup_{j=1}^{i} C_j \right). \tag{30}$$

If $i = 0$, $T_0 \leftarrow T_0^s$. We call T_i the output tree of Stage i.

(9) (Loop) If $|S_i| > 1$, $i \leftarrow i + 1$ and go back to Step 5. If $|S_i| = 1$, continue to Step 10.

(10) (Tree selection) The final output is the tree T_{j^*} such that

$$j^* = \underset{j=0,\ldots,i}{\operatorname{argmin}} \left[c\left(T_j\right) \right], \tag{31}$$

that is, the minimum cost tree among the output trees from all the stages.

End Algorithm

5.2.4. Comments. We explain the details of several steps in the algorithm. In Step 3 the flow variables denoted by f_u, $u \in S$, associated with the source nodes are initialized where we will track the amount of flows throughout the algorithm. In Step 6 it is natural to select a node with the maximum degree as the center node, since such node is literally a "hub." When solving the hub-and-spoke problem at each stage, we choose to solve the MWEC problem whereas in [25] the load-balanced facility location problem is solved. An advantage of solving MWEC problem is that it is considerably simpler than load-balanced facility location problems since an MWEC problem can be reduced to a minimum weight perfect matching problem [37]. Note that the algorithm in [25] solves the hub-and-spoke problem only once, that is, its output is analogous to the output tree from Stage 1 of our algorithm. Meanwhile HM algorithm solves minimum weight perfect matching at each stage in order to locally aggregate flows with low costs. HM algorithm solves the matching problem hierarchically until all the flows are aggregated to a single source, and the final output is the union of those matchings. Thus its final output is analogous to that from the final stage of our algorithm. In other words, the outputs of the abovementioned algorithms correspond to those from intermediate stages in our algorithm. The HIERARCHY algorithm proposed in [20] hierarchically constructs Steiner trees and solves load-balanced facility location problems, however in a way which heavily relies on the concave and increasing property of ϕ. Thus the algorithm may not be suitable for convex and decreasing ϕ.

5.3. Performance Analysis. In this section we analyze the performance of HCST algorithm. For set $\mathscr{E}$ of weighted edges, let $\|\mathscr{E}\|$ denote the sum of its edge weights, that is, $\sum_{e \in \mathscr{E}} w_e$. For given source set Σ, let $T_S(\Sigma)$ denote the edge set of the optimal Steiner tree associated with Σ.

Proposition 5. *For given network graph $G = (V, E)$, the cost achieved by HCST algorithm is higher than the optimal algorithm by a factor of at most B defined as*

$$B := \frac{1}{\phi(Z)} \cdot \min_{i=0,\ldots,\mathcal{I}} \left[\phi\left(2^i\right) \rho_S + \sum_{k=1}^{i} \phi\left(2^{k-1}\right) \rho_k \right], \tag{32}$$

where $\mathcal{I}$ ($\mathcal{I} \le \log_2 \eta$) denotes the stage at which HCST algorithm terminates. $\rho_S \approx 1.39$ denotes the approximation ratio for Steiner tree problem, and $\rho_i \in [0, 1]$ is the ratio of the sums of edge weights between MWEC M_i at Stage i of HCST algorithm and the Steiner tree associated with source set S, that is,

$$\rho_i = \frac{\|M_i\|}{\|T_S(S)\|}. \tag{33}$$

Also Z is defined as

$$Z := \frac{\sum_{i=1}^{\lfloor \eta/2 \rfloor} i w_{[i]} + \eta \sum_{i=0}^{n-\lfloor \eta/2 \rfloor - 1} w_{[|E|-i]}}{\sum_{i=1}^{\lfloor \eta/2 \rfloor} w_{[i]} + \sum_{i=0}^{n-\lfloor \eta/2 \rfloor - 1} w_{[|E|-i]}}, \tag{34}$$

where $n := |V|$, and $w_{[i]}$ denotes the ith smallest value of the edge weights of G. Note that the second summation term of (32) is defined to be 0 if $i = 0$.

Proof. Denote the optimal cost by OPT. We first find a lower bound for OPT. Let T^* denote the set of edges of the optimal aggregation tree. Let us sort the amount of edge flows of T^* in increasing order, and denote them by d_i, that is, $0 < d_1 \le d_2 \le \cdots \le d_l$ where T^* has l edges. There are at least η nonzero flows since there are η sources, hence $d_\eta > 0$ and $l \ge \eta$ hold. In addition l is at most n, since T^* is a tree. Also it is clear that

$d_i \le i$, for $i = 1, \ldots, \eta$, and d_i is at most η for $\eta < i \le l$. This implies that, since ϕ is decreasing, $\phi(d_i) \ge \phi(i)$, $i = 1, \ldots, \eta$. Let us denote the weight of the edge that carries flow d_i by v_i. For real numbers a and b, let $a \wedge b := \min(a, b)$. We have that

$$\mathrm{OPT} = \sum_{i=1}^{l} v_i \phi(d_i) \ge \sum_{i=1}^{l} v_i \phi(i \wedge \eta) \tag{35}$$

$$\ge \left(\sum_{i=1}^{l} v_i \right) \phi\left(\frac{\sum_{i=1}^{l} (i \wedge \eta) v_i}{\sum_{i=1}^{l} v_i} \right) \tag{36}$$

$$\ge \left\| T_S(S) \right\| \phi\left(\frac{\sum_{i=1}^{l} (i \wedge \eta) v_i}{\sum_{i=1}^{l} v_i} \right), \tag{37}$$

where (36) is by Jensen's inequality due to the convexity of ϕ, and (37) is from the definition of Steiner trees. Considering that ϕ is decreasing, we would like to make the argument of ϕ in (37) as large as possible in order to find a lower bound for OPT. Hence we would like to maximize $\lambda(v_1, \ldots, v_l)$ defined as

$$\lambda(v_1, \ldots, v_l) := \frac{\sum_{i=1}^{l} (i \wedge \eta) v_i}{\sum_{i=1}^{l} v_i}, \tag{38}$$

where v_i, $i = 1, \ldots, l$ are chosen from the edge weights of G. For the purpose of maximizing (38), we will assume $v_1 \le v_2 \le \cdots \le v_l$ WLOG, because over all possible permutations $\pi(1), \pi(2), \ldots, \pi(l)$ of $\{1, 2, \ldots, l\}$, $\sum_i (i \wedge \eta) v_{\pi(i)}$ is maximized when $v_{\pi(1)} \le v_{\pi(2)} \le \cdots \le v_{\pi(l)}$.

We first observe that $\lambda(\cdot)$ is decreasing in $v_1, \ldots, v_{\lfloor \eta/2 \rfloor}$, since if $k \le \lfloor \eta/2 \rfloor$, we have that

$$\begin{aligned} \frac{\partial \lambda}{\partial v_k} &= \frac{k \sum_i v_i - \sum_i (i \wedge \eta) v_i}{\left(\sum_i v_i \right)^2} \\ &= \frac{\sum_{j=1}^{k-1} (k - j)\left(v_j - v_{2k-j} \right) + \sum_{j=2k}^{l} \left(k - (j \wedge \eta) \right) v_j}{\left(\sum_i v_i \right)^2} \\ &\le 0. \end{aligned} \tag{39}$$

Hence $\lambda(\cdot)$ can be maximized over $v_1, \ldots, v_{\lfloor \eta/2 \rfloor}$ by choosing $\lfloor \eta/2 \rfloor$ smallest weights from the edge weights of G, that is, by letting $v_i = w_{[i]}$ for $i = 1, \ldots, \lfloor \eta/2 \rfloor$.

Next we would like to derive an upper bound for $\lambda(w_{[1]}, \ldots, w_{[\lfloor \eta/2 \rfloor]}, v_{\lfloor \eta/2 \rfloor + 1}, \ldots, v_l)$ as follows:

$$\lambda\left(w_{[1]}, \ldots, w_{[\lfloor \eta/2 \rfloor]}, v_{\lfloor \eta/2 \rfloor + 1}, \ldots, v_l \right) = \frac{\sum_{i=1}^{\lfloor \eta/2 \rfloor} i w_{[i]} + \sum_{i=\lfloor \eta/2 \rfloor + 1}^{l} (i \wedge \eta) v_i}{\sum_{i=1}^{\lfloor \eta/2 \rfloor} w_{[i]} + \sum_{i=\lfloor \eta/2 \rfloor + 1}^{l} v_i} \tag{40}$$

$$\le \frac{\sum_{i=1}^{\lfloor \eta/2 \rfloor} i w_{[i]} + \eta \sum_{i=\lfloor \eta/2 \rfloor + 1}^{l} v_i}{\sum_{i=1}^{\lfloor \eta/2 \rfloor} w_{[i]} + \sum_{i=\lfloor \eta/2 \rfloor + 1}^{l} v_i} \tag{41}$$

$$\le \frac{\sum_{i=1}^{\lfloor \eta/2 \rfloor} i w_{[i]} + \eta \sum_{i=0}^{n - \lfloor \eta/2 \rfloor - 1} w_{[|E| - i]}}{\sum_{i=1}^{\lfloor \eta/2 \rfloor} w_{[i]} + \sum_{i=0}^{n - \lfloor \eta/2 \rfloor - 1} w_{[|E| - i]}}. \tag{42}$$

For inequality (42), we used the fact that (41) is increasing in $\sum_{i=\lfloor \eta/2 \rfloor + 1}^{l} v_i$, hence we chose $l = n$ and the largest possible weights $w_{[|E|]}, w_{[|E|-1]}, \ldots$ for $v_{\lfloor \eta/2 \rfloor + 1}, v_{\lfloor \eta/2 \rfloor + 2}, \ldots$, in order to maximize $\sum_{i=\lfloor \eta/2 \rfloor + 1}^{l} v_i$. From (42), we obtain $\lambda(v_1, \ldots, v_l) \le Z$. Hence from (37), we obtain

$$\mathrm{OPT} \ge \left\| T_S(S) \right\| \phi(Z). \tag{43}$$

Now let us consider the cost of output tree at Stage i of HCST algorithm, or $c(T_i)$. Recall that in HCST algorithm, S_i denotes the source set at Stage i, and T_i denotes the output tree at Stage i. The cost of T_i is divided into (i) the cost incurred by hierarchical MWECs $M_1, \ldots, M_i$, and (ii) the cost of ρ_S-approximate Steiner tree T_i^s associated with S_i. Hence

$$c(T_i) = \sum_{k=1}^{i} \sum_{e \in M_k} w_e \phi(d(e)) + \sum_{e \in T_i^s} w_e \phi(d(e)), \tag{44}$$

where $d(e)$ denote the amount of flow at Edge e under HCST algorithm. Note that, the amount of flow in the network at Stage i is at least 2^{i-1}, since the flows are agglomerated through MWECs at every stage. Since $\phi(\cdot)$ is decreasing, the first summation of (44) is at most

$$\sum_{k=1}^{i} \phi\left(2^{k-1} \right) \sum_{e \in M_k} w_e = \sum_{k=1}^{i} \phi\left(2^{k-1} \right) \left\| M_k \right\|. \tag{45}$$

Note that the first summation of (44) is 0 for Stage 0. As for the second summation of (44),

$$\sum_{e \in T_i^s} w_e \phi(d(e)) \le \phi\left(2^i \right) \left\| T_i^s \right\| \tag{46}$$

$$\le \rho_S \phi\left(2^i \right) \left\| T_S(S_i) \right\| \tag{47}$$

$$\le \rho_S \phi\left(2^i \right) \left\| T_S(S) \right\|. \tag{48}$$

Inequality (48) is due to $S_i \subseteq S$; specifically, the Steiner tree for S is a tree that spans S_i, hence by definition, the sum of edge weights of $T_S(S_i)$ is no more than that of the Steiner tree associated with S.

In conclusion, we have that, from (43), (45) and (48),

$$\begin{aligned} c(T_i) &\le \rho_S \phi\left(2^i \right) \left\| T_S(S) \right\| + \sum_{k=1}^{i-1} \phi\left(2^{k-1} \right) \left\| M_k \right\| \\ &\le \frac{\mathrm{OPT}}{\phi(Z)} \left[\rho_S \phi\left(2^i \right) + \sum_{k=1}^{i} \phi\left(2^{k-1} \right) \frac{\left\| M_k \right\|}{\left\| T_S(S) \right\|} \right]. \end{aligned} \tag{49}$$

Since the cost of HCST algorithm is $\min_{i=0,\ldots,\mathcal{F}} c(T_i)$, the proposition is proved. □

An interpretation for ratio B in (32) is as follows: the first term in the bracket of B represents a bound on the macroscopic cost associated with the Steiner tree approximation. The second term in the bracket of B is a bound on the cost associated with the hierarchical aggregation of flows, that is, the microscopic cost reduction. Clearly we have

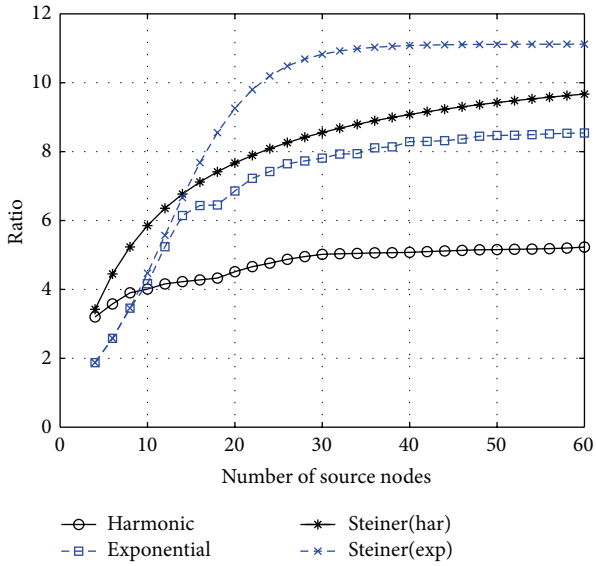


Figure 4: Performance bounds under varying number of sources.

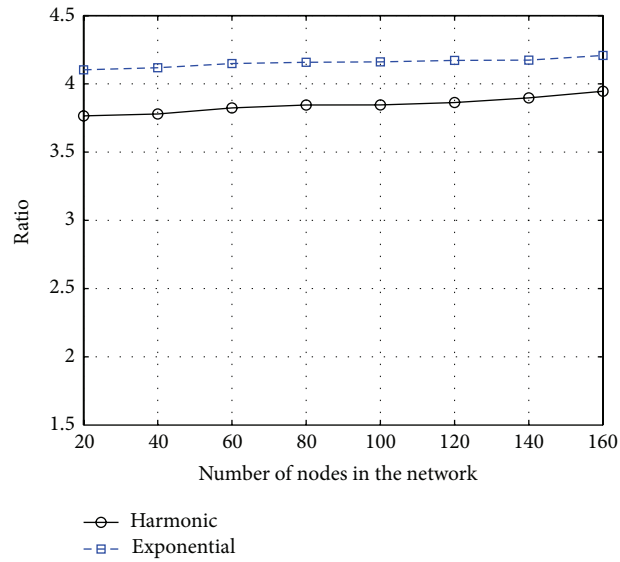


Figure 5: Performance bounds under varying network sizes.

that $\|M_1\| \geq \|M_2\| \geq \cdots$, due to $S_1 \supseteq S_2 \supseteq \cdots$, thus $\rho_1, \rho_2, \ldots$, is a decreasing sequence where $0 \leq \rho_i \leq 1$, for all i. The progressive cost reduction due to hierarchical flow aggregation is reflected in $\rho_1, \rho_2, \ldots$. As in (32), B is the minimum of $\mathcal{I} + 1$ numbers, each of which contains a weighted sum of $\phi(\cdot)$ in different combination of weights ρ_i. Hence B represents the empirical minimum of different degrees of tradeoff between microscopic and macroscopic cost reduction.

Next we discuss constant Z in (34). Firstly observe that $Z \leq \eta$; the first summation of the numerator of (34) is at most $\eta \sum_{i=1}^{\lfloor \eta/2 \rfloor} \nu_i$, in which case the first term of (34) is at most η. Note that a naive upper bound for $\lambda(\nu_1, \ldots, \nu_n)$ is simply η, yielding a lower bound OPT $\geq \|T_S(S)\|\phi(\eta)$; however we observe that our bound (43) improves such a bound since $\phi(Z) \geq \phi(\eta)$.

B can be numerically computed for a given graph, and in the next section we provide numerical examples of B. We also apply HCST algorithm to a specific graph as an example.

5.4. Illustrating Examples. In this section we consider a simple convex and decreasing ϕ. As previously the packet header length is α bits, and we assume that the maximum packet size is 10 times the header length, that is, 10α. We will accordingly consider $\phi(m)$ which is convex and decreasing for $m \geq 1$ of the following form:

$$\phi(m) = \begin{cases} 9\alpha m^{-1} + \alpha, & m \geq 1, \\ 0, & m = 0. \end{cases} \tag{50}$$

Clearly $\alpha < \phi(m) \leq 10\alpha$ holds for $m \geq 1$.

Figures 4 and 5 show the numerical examples of the performance bound B. B is computed and averaged over randomly generated graphs of uniformly distributed nodes in a square area. In Figure 4, network size n is fixed to 200, and B is plotted against the number of source nodes η. We consider two types of cost functions: the curve labelled "harmonic" represents the cost function (50) in which $\phi(\cdot)$ decreases as a harmonic sequence. The curve labelled "exp" corresponds to the case where the term m^{-1} in (50) is replaced by $\exp(-\delta(m-1))$ where the parameter $\delta > 0$ controls the decay rates of the cost function. We set $\delta = 0.2$ in this example. In addition, we compare B with a simple analytical bound; suppose we build a ρ_S-approximate Steiner tree based on S. The cost under that tree is at most $\rho_S\|T_S(S)\|\phi(1)$. By combining that cost with (43), we obtain a simple approximation ratio of $\rho_S\phi(1)/\phi(Z)$ for the approximately optimal Steiner tree. In Figure 4, the plots of such bounds based on ρ_S-approximate Steiner tree are added for both harmonic and exponential cost functions, and are labelled as "Steiner(har)" and "Steiner(exp)," respectively. We observe that B provides improved bounds as compared to those based on ρ-approximate Steiner tree. In Figure 5, B is plotted against varying n under the aforementioned harmonic and exponential cost function where we fixed δ to 10. In Figures 4 and 5, we observe that B eventually becomes nearly constant, or increases very slowly at most, even if the system size grows. Hence we conclude that B provides an approximation ratio which remains effectively constant irrespective of the system size.

Next we present an example of the application of the HCST algorithm to a specific graph. An example of G is given in Figure 6(a). G consists of $n = 10$ nodes where Node 1 is the sink, that is, $t = 1$. There are four source nodes: $S = \{2, 3, 4, 5\}$ where the sources are depicted in a shaded color. Each source generates 1 unit of data. We will again consider convex and decreasing $\phi(\cdot)$ given by (50), and assume $\alpha = 1$. Figure 6(b) shows the output of Stage 0 or T_0 which is an approximately optimal Steiner tree. Figure 7 shows the MWECs over the stages. Figure 7(a) shows the metric completion of the subgraph induced by S. Figure 7(b) shows the MWEC at Stage 1. Node 4 and 5 became the center nodes as emphasized in the figure. Figure 7(c) shows the MWEC and the center node at Stage 2.

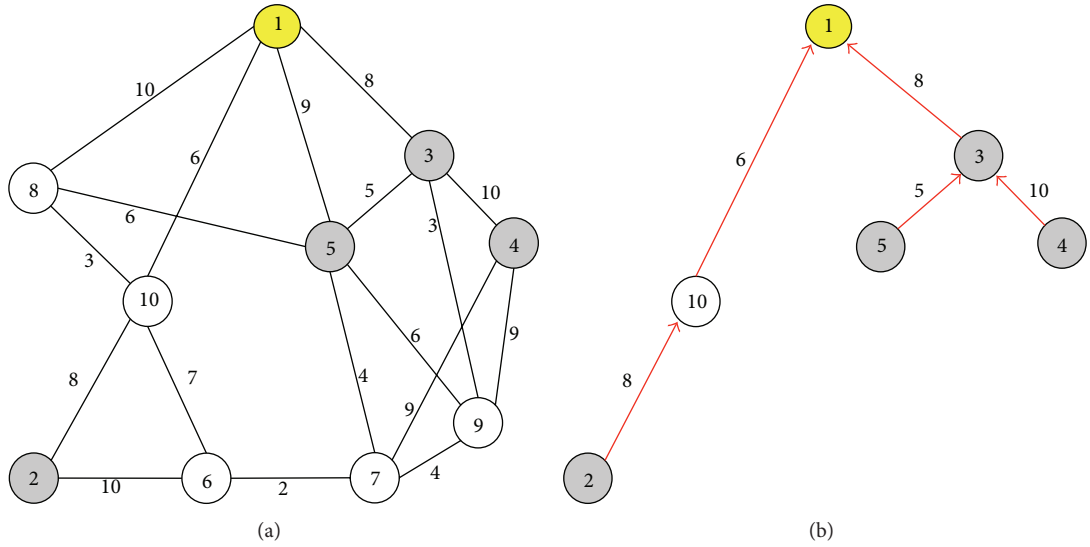


FIGURE 6: (a) G in the example. (b) The output T_0 from Stage 0.

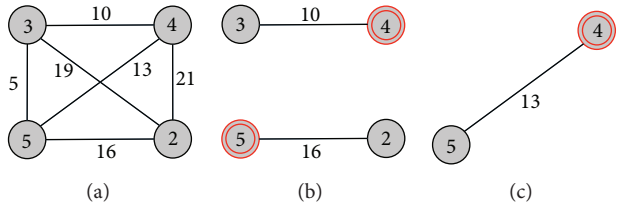


FIGURE 7: (a) A complete graph of the sources. (b) MWEC from Stage 1. (c) MWEC from Stage 2.

Figure 8(a) shows the full paths of the MWEC at Stage 1, that is, that in Figure 7(b), in G. By building an approximately optimal Steiner tree T_1^s associated with $\{1, 4, 5\}$ and taking the union of T_1^s and C_1 as in Step 8, we get T_1 as in Figure 8(b). Similarly Figure 9 demonstrates Stage 2 of the algorithm. The full paths for the MWEC from Figure 7(c) in G are shown in Figure 9(a). Note that Node 4 is selected as the center node, and the output from Stage 2 or T_2 is shown in Figure 9(b). Let us compare the energy costs from all the stages. For T_0, a total of three flows pass through the link between Node 1 and 3, while the flow on the other links is simply 1. Thus, the cost of T_0 from Stage 0 is given by

$$\begin{aligned} c(T_0) &= 8 \times \phi(3) + (8 + 6 + 5 + 10) \times \phi(1) \\ &= 8 \times 10 + 29 \times 4 = 322. \end{aligned} \tag{51}$$

Similarly, we have that

$$c(T_1) = 314.8, \qquad c(T_2) = 275.5. \tag{52}$$

Thus the final output of HCST is T_2 with the final cost of 275.5. Note that in this example, the Shortest Path Tree (SPT) heuristic incurs the energy cost of 374.

Next consider $\phi(m)$ such that

$$\phi(m) = \begin{cases} 1, & m \geq 1, \\ 0, & m = 0. \end{cases} \tag{53}$$

Assume that the algorithm has yielded the same T_0, T_1 and T_2 as the previous case. Since ϕ is constant for $m \geq 1$, the problem reduces to the Steiner tree problem, thus one would expect that T_0 would perform the best since T_0 is intended to be an approximately optimal Steiner tree. The energy costs are given by

$$c(T_0) = 37, \qquad c(T_1) = 44, \qquad c(T_2) = 43; \tag{54}$$

thus indeed the HCST algorithm will output T_0 as the best solution with cost 37, whereas the SPT heuristic will yield the energy cost of 41. This demonstrates that our algorithm can effectively deal with various types of convex and decreasing aggregation cost functions. In the following section we will evaluate the performance of the HCST algorithm by simulation.

6. Simulation

In our simulation we randomly generate G as follows. The node locations are generated independently and uniformly on a unit square. We define G as the Delaunay graph induced by the node locations. An example of G is depicted in Figure 10 for $n = 20$. As previously it is assumed that the average number of bits required to transmit the aggregated information $g(X_1, \ldots, X_m)$ is approximately $\alpha + b + h(g(X_1, \ldots, X_m))$ where we set header length a to 1 and the number of quantization bits b to 3. The edge weights are randomly selected from $\{1, \ldots, 10\}$ which represents the energy consumption per transmitted bit. In our simulation two types of sources are considered. The first type, called uniform type, is associated with the extreme data retrieval

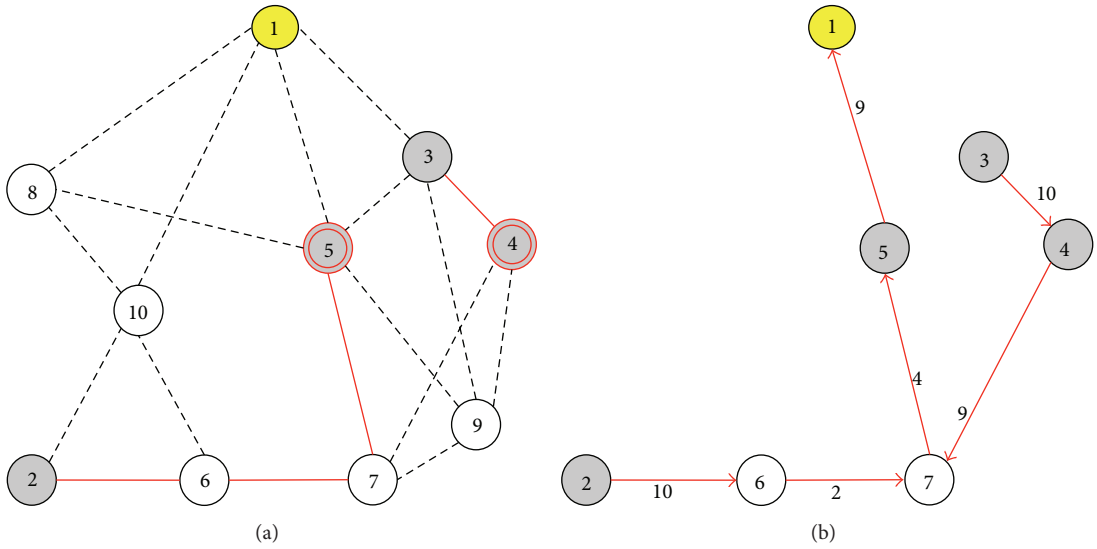


Figure 8: (a) MWEC in G from Stage 1. (b) The output T_1 from Stage 1.

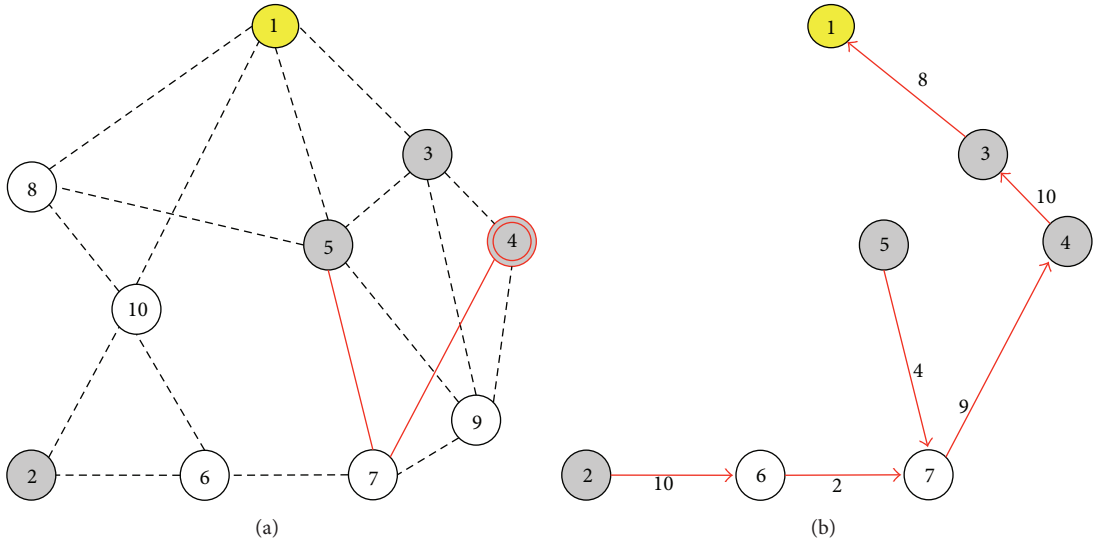


Figure 9: (a) MWEC in G from Stage 2. (b) The output T_2 from Stage 2.

problem, that is, X_i are i.i.d. uniformly on $[0, 1]$. The second type, called Gaussian type, is associated with retrieving the maximum of Gaussian source data where $X_i \sim N(0, 1)$. The summary function $g(\cdot)$ is given by `max` function.

We will compare the performance of the HCST algorithm with HM algorithm [16] and SPT heuristic. Figure 11 shows the average energy consumption of the algorithms when we fix the number of sources to 8 with varying n. The energy cost shown on the left (resp. right) of Figure 11 is associated with the sources of uniform (resp. Gaussian) type. We observe that the HCST algorithm achieves lower energy costs than the SPT heuristic in both types of the sources. The gain in the energy savings by the HCST algorithm ranges 35–38% for uniform type sources and 24-25% for Gaussian type sources. Compared to HM algorithm, our algorithm reduces the energy consumption by 20-21% and 14-15% for uniform and Gaussian type sources, respectively. HM algorithm focuses on microscopic cost reduction, which may be effective for concave and increasing cost functions, however not for convex and decreasing cost functions. Comparing SPT heuristic and HCST algorithm, we observe that the difference in the mean energy consumption of the algorithms slightly increases with n. This can be interpreted as follows: for larger networks, there is further room for improvement by HCST, for example, there are more choices for Steiner nodes and more ways to merge sources at low costs by MWEC. Thus the performance gain from the HCST algorithm relative to the SPT heuristic is expected to grow with n as shown in the simulation.

Figure 12 shows the mean energy costs with varying n where we scale the number of the sources proportional to n. Specifically in the simulation we let $\eta = n/5$, that is, one out of five nodes is a source node. In the figure we see that the HCST

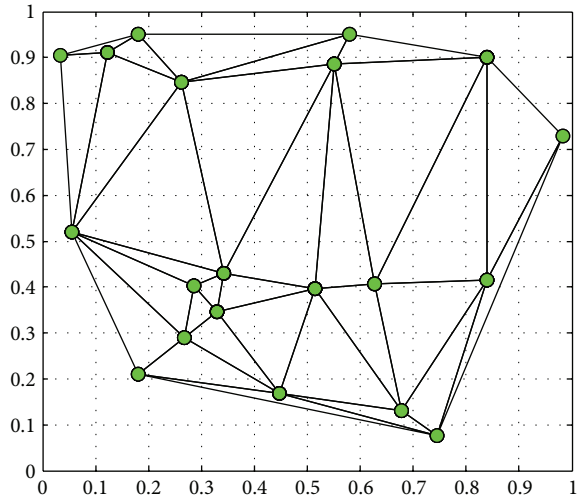


Figure 10: An example of randomly generated G for simulation.

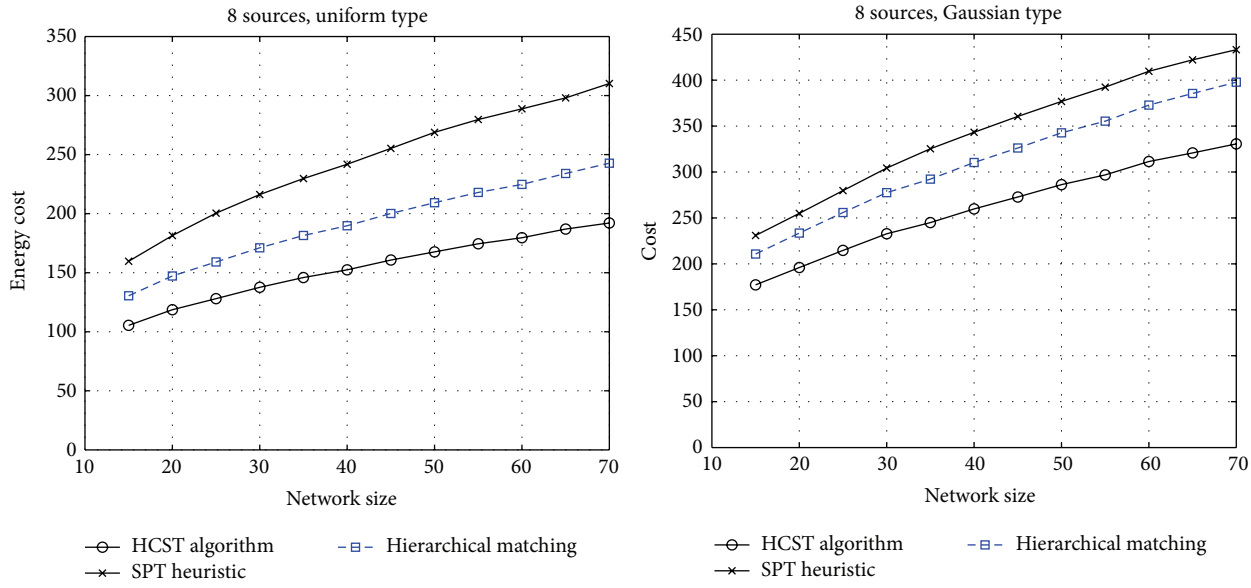


Figure 11: Energy cost associated with a fixed number of sources.

algorithm again outperforms the SPT heuristic. The relative savings in energy by HCST algorithm ranges 19–41% for uniform type sources and 14–27% for Gaussian type sources. Relative to HM algorithm, HCST algorithm saves energy costs by 20–23% and 14–17% for uniform and Gaussian type sources, respectively. The difference in the energy cost of the algorithms increases with n similar to the case of fixed number of sources, however, such a rate of increase is higher in the case of varying number of sources. This can be explained as follows. When we increase the network size, the number of sources also increases proportionally. When the network size grows, from the previous argument such that there is further room for improvement by HCST, its relative gain will increase with the network size. In addition to that, since the number of sources grows, the total number of stages at the end of the HCST algorithm will also increase. Since HCST chooses the best tree from the intermediate output trees collected over stages, a large number of stages implies that we can choose the final output tree from a large pool of trees having various degrees of tradeoff between microscopic and macroscopic aspects of the cost reduction. Thus the abundance of source nodes enables us to choose an aggregation tree with a "refined" tradeoff, which is crucial for a convex and decreasing ϕ. This explains the enhanced performance of HCST with increasing number of sources. Hence we conclude from the simulation that the HCST algorithm can improve performance for various proportions of source nodes among the network.

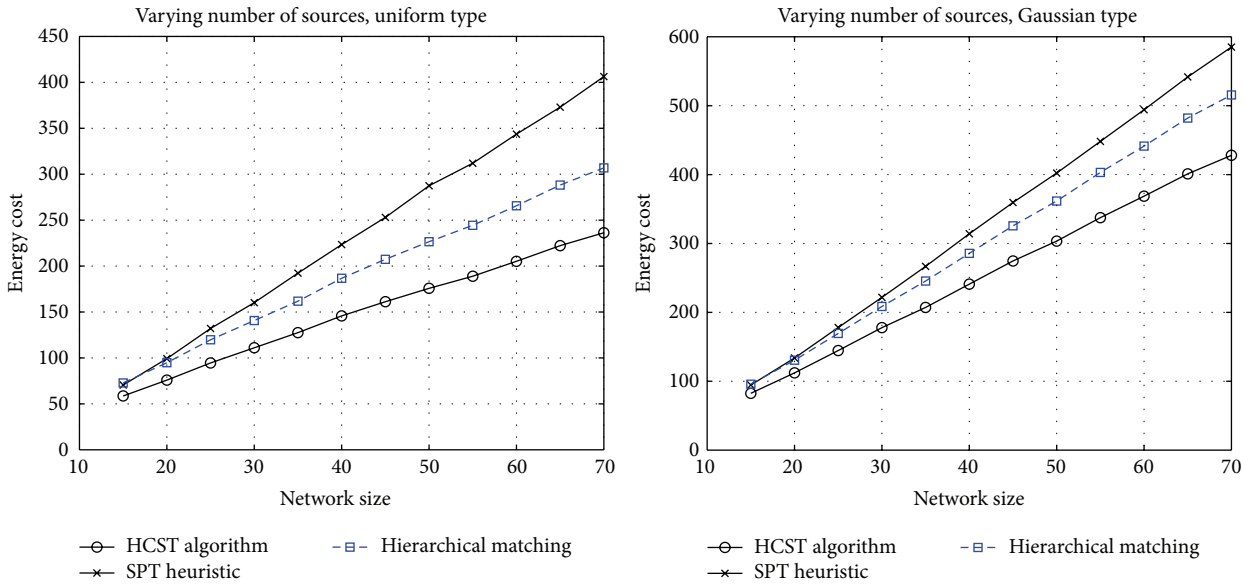


Figure 12: Energy cost associated with a varying number of sources.

7. Conclusion

In this paper we have studied a single-sink aggregation problem for wireless sensor networks computing several widely used summary functions. It is observed that the problem is characterized by the aggregation cost function ϕ which maps the amount of aggregated measurements to transmission costs at a link. We show that the properties of ϕ depend heavily on the chosen summary function $g(\cdot)$. When $g(\cdot)$ is given by `sum` or `mean`, we showed that ϕ is concave and increasing, implying that there exist algorithms such as the HM algorithm which can approximate the optimal algorithm by a factor logarithmic in the number of sources. A similar argument was made when $g(\cdot)$ is `weighted sum` for i.i.d. Gaussian sources. When $g(\cdot)$ is given by `max`, however, we have shown that ϕ is convex and decreasing for certain types of sources. For such ϕ we identify that there exists a tradeoff between the following two aspects of cost reduction: firstly local clustering of sources which is the microscopic aspect, and secondly a low-cost routing from the clustered sources to the sink which is the macroscopic aspect. We proposed the Hierarchical Cover and Steiner Tree algorithm which empirically finds the best tradeoff point between the aspects. Numerical examples and simulation results were presented to demonstrate that the HCST algorithm is versatile and improves performance for various types of convex and decreasing ϕ. A future direction would be investigating the optimal aggregation problems for a wider range of summary functions. In addition, the evaluation of the HCST algorithm in a real-world testbed environment is also part of our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by Basic Science Research Program through The National Research Foundation of Korea (NRF) funded by The Ministry of Science, ICT & Future Planning (NRF-2013R1A1A1062500), and in part by the ICT R&D program of MSIP/IITP, (10-911-05-006, High Speed Virtual Router that Supports Dynamic Circuit Network).

References

[1] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 755–764, 2005.

[2] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 146–159, 2001.

[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobilecomputing and networking*, pp. 56–67, ACM, August 2000.

[4] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, pp. 575–578, IEEE, 2002.

[5] M. Bagaa, Y. Challal, A. Ksentini, A. Derhab, and N. Badache, "Data aggregationscheduling algorithms in wireless sensor networks: solutionsand challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, pp. 1339–1368, 2014.

[6] L. A. Villas, A. Boukerche, H. S. Ramos, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, "DRINA: a lightweight and reliable routing approach for in-network aggregation in wireless sensor networks," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 676–689, 2013.

[7] J. Ma, W. Lou, and X.-Y. Li, "Contiguous link scheduling for data aggregation in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1691–1701, 2014.

[8] Z.-J. Zhang, C.-F. Lai, and H.-C. Chao, "A green data transmission mechanism for wireless multimedia sensor networks using information fusion," *IEEE Wireless Communications*, vol. 21, no. 4, pp. 14–19, 2014.

[9] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.

[10] P. Von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," in *Proceedings of the Joint Workshop on Foundations of Mobile Computing*, pp. 60–66, ACM, 2004.

[11] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked slepianwolf: theory, algorithms, and scaling laws," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4057–4073, 2005.

[12] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 28–35, Berkeley, Calif, USA, April 2004.

[13] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *Proceedings of the 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2571–2582, March 2004.

[14] J. Liu, M. Adler, D. Towsley, and C. Zhang, "On optimal communication cost for gathering correlated data through wireless sensor networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 310–321, ACM, September 2006.

[15] S. Hariharan and N. Shroff, "Maximizing aggregated revenue in sensor networks under deadline constraints," in *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 4846–4851, 2010.

[16] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk," in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, pp. 499–505, SIAM, 2003.

[17] M. Andrews and L. Zhang, "The access network design problem," in *Proceedings of the of IEEE Symposium on Foundations of Computer Science (FOCS '98)*, p. 40, IEEE Computer Society, 1998.

[18] Y. Bartal, "On approximating arbitrary metrics by tree metrics," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 161–168, ACM, May 1998.

[19] A. Meyerson, K. Munagala, and S. Plotkin, "Cost-distance: two metric network design," in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00)*, p. 624, IEEE Computer Society, 2000.

[20] S. Guha, A. Meyerson, and K. Munagala, "A constant factor approximation for the single sink edge installation problems," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pp. 383–388, ACM, 2001.

[21] M. Andrews, A. Anta, L. Zhang, and W. Zhao, "Routing for energy minimizationin the speed scaling model," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, IEEE, San Diego, Calif, USA, March 2010.

[22] M. Andrews, S. Antonakopoulos, and L. Zhang, "Minimum-cost network design with (Dis)economies of scale," in *Proceedings of the IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS '10)*, pp. 585–592, Las Vegas, Nev, USA, October 2010.

[23] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian, "Buy-at-bulk network design: approximating the single-sink edge installation problem," in *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 619–628, Society for Industrial and Applied Mathematics, January 1997.

[24] B. Awerbuch and Y. Azar, "Buy-at-bulk network design," in *Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science (FOCS '97)*, IEEE Computer Society, pp. 542–547, October 1997.

[25] D. R. Karger and M. Minkoff, "Building Steiner trees with incomplete global knowledge," in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00)*, pp. 613–623, IEEE Computer Society, November 2000.

[26] M. Charikar and A. Karagiozova, "On non-uniform multicommodity buy-at-bulk network design," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 176–182, ACM, 2005.

[27] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour, "Approximation algorithms for non-uniform buy-at-bulk network design," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS '06)*, pp. 677–686, Berkeley, Calif, USA, October 2006.

[28] A. Scaglione and S. D. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," in *Proceedings of the 8th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '02)*, pp. 140–147, Atlanta, Ga, USA, September 2002.

[29] L. Galluccio, S. Palazzo, and A. T. Campbell, "Efficient data aggregation in wireless sensor networks: an entropy-driven analysis," in *Proceedings of the IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '08)*, pp. 1–6, September 2008.

[30] D. Kandris, P. Tsioumas, A. Tzes, G. Nikolakopoulos, and D. D. Vergados, "Power conservation through energy efficient routing in wireless sensor networks," *Sensors*, vol. 9, no. 9, pp. 7320–7342, 2009.

[31] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, and S. Kotsopoulos, "Energy efficient and perceived QoS aware video routing over wireless multimedia sensor networks," *Ad Hoc Networks*, vol. 9, no. 4, pp. 591–607, 2011.

[32] L. Galluccio, A. T. Campbell, and S. Palazzo, "Concert: aggregation-based congestion control for sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pp. 274–275, San Diego, California, USA, November 2005.

[33] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.

[34] M. Madiman, "On the entropy of sums," in *Proceedings of the Information Theory Workshop (ITW '08)*, pp. 303–307, IEEE, Porto, Portugal, May 2008.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[36] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità, "An improved LP-based approximation for steiner tree," in *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC '10)*, pp. 583–592, June 2010.

[37] A. Schrijver, *Combinatorial Optimization*, Springer, New York, NY, USA, 2003.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 410298, 9 pages
http://dx.doi.org/10.1155/2015/410298

Research Article

GTDM: A DTN Routing on Noncooperative Game Theory in a City Environment

Wenzao Li,[1,2] Feng Lin,[3] Jiliu Zhou,[3] and Yan Wang[1]

[1]*College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China*
[2]*College of Communication Engineering, Chengdu University of Information Technology, Chengdu 610225, China*
[3]*School of Computer Science, Sichuan University, Chengdu 610065, China*

Correspondence should be addressed to Feng Lin; linfeng@scu.edu.cn

Received 16 October 2014; Revised 29 March 2015; Accepted 29 March 2015

Academic Editor: Qing-An Zeng

The performance of delay tolerant networks (DTNs) can be influenced by movement model in different application environments. The existing routing algorithms of DTNs do not meet the current city environments due to the large differences in node densities, social characteristics, and limited energy. The key indicators of DTNs such as success delivery ratio, average delivery latency, network lifetime, and network overhead ratio can influence the performances of civil DTNs applications. Aiming to improve the key indicators of DTNs in city environments, this paper presents a fixed sink station based structure and a more proper routing algorithm named Game Theory Based Decision Making (GTDM). GTDM shows decision-making process for neighborhood selection and packet delivering strategy which is based on the noncooperative game theory method and city environment characteristics. GTDM performance is evaluated using numerical simulations under Working Day Movement (WDM) model and the results suggested that GTDM outperforms other traditional DTNs routing approaches, such as Epidemic and Prophet algorithms.

1. Introduction

A delay tolerant network (DTN) is a network of thousands of resource-constrained mobile sensors for communication in some challenging environments. DTNs are characterized by their high latency, low delivery ratio, and long periods of disconnection due to the fact that network topology is changing continually. In recent years, the use of DTNs in metropolitan areas has attracted increasing attention [1–3]. In the future, "smart" cities may be created by essential public services using cutting-edge computing technologies [4], which are mainly realized on DTN platform. Unlike the use in other environments, such as battlefields, space, and oceans [5–7], there are more opportunities with the rapid development of portable smart devices. DTN routing problems are discussed based on the different requirements for various deployment environments and applications. Although there are several existing DTNs routing algorithms which can be applied to urban environments, the poor performance affects the efficiency of urban based DTN system. Thus, we focus on improving DTN performance in city scenarios; more efficient routing algorithm is designed for mobile nodes with constrained energy. In this paper, we propose a framework of sensor selection based on the game theory approach and the optimization of data queue management, which can find an energy consumption balance and have better network performance than Epidemic and Prophet routing algorithms. The main contributions of this paper are listed as follows.

(1) We describe the distributed fixed sink stations for data collection and the mobile node as the role of data source or relay node in a city environment. Then, a neighborhood node selection method is introduced.

(2) A packet delivering decision-making algorithm GTDM for DTNs is proposed based on game theory approach. It is for routing determination of DTNs in city environments. The GTDM can determine punishment or reward between a pair of nodes through a game, and the active node can be measured by node asset.

(3) The packet priority and packets transmission strategy between two nodes is present; in this way, a packet determined which node is the most proper packet carrier.

(4) The performance of the proposed algorithm is validated by a series of experiments under WDM movement model. The result shows that GTDM can prolong the network lifetime and has better routing performance than Epidemic and Prophet routing algorithms.

The remainder of the paper is organized as follows: Section 2 describes the related works of the city environment based routing algorithm and Section 3 describes the suitable scenario and GTDM algorithm process. In Section 4, this paper gives the forwarding decision based on game theory (GT). In Section 5, we present the simulation settings and the results, and the paper is concluded in Section 6.

2. Related Work

A DTN protocol should be cautious in how it saves the limited network resources. So, a packet is transmitted to improper relay node, which can cause the decreasing of network performance. Then, how to determine whether a node sends packets to another node? In some instances, GT can be applied to solve such problem [8, 9]. Generally, there are two kinds of GT: cooperative game theory and noncooperative game theory. Cooperative game theory pays more attention to the maximal profits of the group through cooperative efforts. A node will act selfishly by minimizing their individual utility in a distributed decision-making environment. Noncooperative game theory focuses on each node's individual utility, and less attention is given to the utility of the whole DTNs. The noncooperative method does not require nodes to have global knowledge. By contrast, under a cooperative approach, nodes with cooperative GT must agree on the premediated strategies and participants have global knowledge [10, 11]. Mobile nodes require a large amount of communication maintenance costs for global knowledge by cooperative approach; thus, DTNs with complete global knowledge are not realistic in reality. Considering the above, game formulations designed should adopt a noncooperative game approach which makes them more realistic.

Based on the same considerations, El-Azouzi et al. [12] proposed a noncooperative game approach, where source and destination nodes were enclosed in two partly overlapping regions. The authors used the Epidemic routing algorithm for a high number of nodes. The Epidemic forwarding algorithm in DTNs maximizes the probability of successful data delivery. However, it ignores the existence of relationships between two people in a city environment. For example, two colleagues will spend all day working, which causes the mobile nodes to have repeated contact [13]. In addition, we believe that there are active objects and inactive objects in city environments [14]. The former objects will have more encounter opportunity than inactive objects, and the active sensor carrier is suitable as relay node. Besides that, mobile nodes have movement regularity and cyclical contact in city environments; regularity of encounters can be expected by history. Unfortunately, these conditions were not put into account by common DTN routing algorithms. Therefore, most of the existing routing algorithms do not work at their highest efficiency in cities. Focusing on the forwarding decision problem of routing patterns, Prophet is a probabilistic routing protocol that uses the past-encounter history and aging methods for forwarding decision and is one of the most commonly used routing algorithms in DTNs. Obviously, the Prophet routing algorithm does not consider all the characteristics of social relationships and the routines of pedestrians in city environments. Likewise, Epidemic routing algorithm cannot release ability under the limited resource scenario. Thus, we propose a more practical and effective GTDM-based routing algorithm, it takes into account various aspects of city environments, and it is a more flexible and scalable DTN routing algorithm for city environments.

3. GTDM Routing Algorithm

In city environment, the delivery ability of each node is not the same because of the limitation of moving range and individual regularity. And GTDM is designed for high ability nodes selection which is judged by history game process. Thus, it is suitable for some scenarios with regular moving nodes. In game-based DTN, the mobile nodes are considered to be participants in the game and must abide by the rules of the designing mechanisms. The forwarding nodes are offered by using incentives, and the misbehaving nodes are punished by the mechanisms. Moreover, any transmission behavior should be cautious; the meaningless forwarding also consumed the nodes' energy. Obviously, the energy hole will decrease the performance of DTN; thus, the energy balance should be necessary to be considered in GTDM.

3.1. Mobile Nodes and Fixed Sink Stations in City Environments. When DTN is deployed in city environments, it can be used for data collection. Therefore, the mobile nodes should be data generator and relay carrier. The mobile nodes usually represent pedestrians, vehicles which are carrier sensors. In addition, some static stations as data collector are located in high node density regions. It is described as fixed sink station in this paper, and we have proposed it in the literature [15]. The mobile nodes and fixed stations are realistic schemes for application in city environments. For instance, Pham and Fdida [16] suggested that DTNs could aid in the distribution of news within cities through the use of fixed nodes. For another instance, the detection of malicious behavior through the use of a Misbehavior Detection System (MDS) was proposed to improve network performance, and it had been discussed in vehicular DTNs of metropolitan areas by Guo et al. [17]. There are some common traits between these described applications, such as node movement trajectory and regular moving. The trajectory can easily overlap the high-density path area with minimal costs and is matched with the given environment map. In addition, the Working Day Movement (WDM) model describes the dynamic behavior of people in cities [18]. So, we believe that the mobile nodes have many contact opportunities in higher path density regions, and fixed sink station is suitable

for distributed location in such regions because of this regularity. In this way, DTNs can flexibly adjust the number of fixed sink station and fixed locations to achieve the desired performance. Further, the nodes have certain movement regularities within a moving region and active nodes can be selected by the history record. From what we have mentioned above, the routing algorithm should be designed based on this scenario's features.

3.2. The GTDM Algorithm Process. The idea of GTDM is to build a series of determined strategies. Multiple indicators are established in each node; these indicators are changed constantly during the game process and delivering history. Then, several determinations are made in sequence.

The details of GTDM routing algorithm are described as follows.

(1) GTDM follows the normal form game of DTNs by a three-tuple $G = \langle N, S, F\rangle$, which represents the game. Here, $N = \{N_1, N_2, \ldots, N_i, \ldots, N_n\}$ is a finite set of n mobile nodes, which currently have effective connections. The strategy set space is described as $\{S_1, S_2, \ldots, S_i, \ldots, S_n\}$, where S_i represents strategy selection of the node i. The strategy combination is $S = \{s_i, \overline{s_i}\}$, where strategy s_i represents the choice of node i and $\overline{s_i}$ is the choice of the other $n-1$ mobile nodes. $F = \{F_1, F_2, \ldots, F_i, \ldots, F_n\}$ is the payoff function of the mobile node, which is defined as $F = \{\eta, \eta_i{}^*\}$. Here, η is the award function and η^* is the punish function.

(2) The mobile nodes in the DTNs are considered as participants of the game, and the nodes do not know the entire topology or the network parameters. Each mobile node establishes a metric list T, which contains the successful delivery of nodes list ω, coins count C_r^i, asset value v, the remaining buffer size L_r, and the energy level E^r. The list T of mobile node supports real-time updates. The value ω_i and energy level E_i^r of node i in list T are calculated in formulas (1) and (2) as follows:

$$\omega_i^{\varphi}(t) = \sum_{t=0}^{t} \varepsilon\left(\omega_i^{\varphi}, t\right), \tag{1}$$

where $\omega_i^{\varphi}(t)$ represents the accumulated packet delivery statistics at time t, φ is the destination address set, and ε is the event counting function

$$E_i^r = \frac{E_i^{\text{ini}} - \sum_0^t \varepsilon'\left(E_b, t\right)}{E_i^{\text{ini}}}, \tag{2}$$

where E_i^{ini} is the initial energy of node i. The energy consumption function is described by ε'.

(3) The mobile nodes are continuously scanning for neighbors. The routing process starts once a mobile node has packets to deliver. If node i finds neighbor nodes in the communication area, the node's neighbor set is built as a finite N_i. Then, the first stage of packet delivery is started, and it includes two steps.

Step 1. Node i traverses the packet in the sending buffer to determine whether the packet's destination belongs to the set N_i. If the destination of the packets belongs to a neighbor node j, node i will be delivered directly to j. In GTDM, each packet destination is a fixed sink station. On the other hand, GTDM will only conduct neighbor scanning when $N_i = \emptyset$.

Step 2. The destinations of the packets that belong to the set N_i are rearranged by increasing TTL values and are transmitted based on this order.

(4) During the second stage, node i selects a neighbor as a relay node from set N_i to transmit the remaining packets. Node selection of the competition in the GTDM depends on asset v of the node in game theory. It satisfies the corresponding payoff function $f_i(s_i, \overline{s_i}) \geq f_i(s_i{}^*, \overline{s_i})$ for $s_i{}^* \in S_i$. In the design of GTDM, the asset v is defined as follows:

$$v = \begin{cases} v_0 & \text{initial value} \\ v_i & \text{otherwise,} \end{cases}$$
$$v_i\left(s_i\right) = \begin{cases} \gamma^{\eta(c_i)} & \text{if node wins in game} \\ \gamma^{\eta^*(c_i)} & \text{if node loses in game,} \end{cases} \tag{3}$$

where v_i obeys the exponential distribution in $[-\infty, +\infty]$. γ is the reward factor and $0 \leq \gamma \leq 1$, which represents the asset changes in intensity. $\eta(c_i)$ is the reward function, and $\eta^*(c_i)$ is the punish function. The node j with the most powerful asset is selected. Once the decision is taken, each node cannot find whether the behavior influences the whole efficiency, because each sensor does not know about the networks.

(5) After the neighbor node j has been selected as a proxy node, node i will selectively transmit the remaining packets with second traversal in the sending buffer. The node i then determines which packets will be transmitted to the proxy node j using the multidecision method. This process is described as follows.

Step 1. By first comparing the energy level of E^r, node i will stop delivering packets if the condition $E_i^r > E_j^r + \Delta E$ is satisfied. If this condition is not satisfied, node i will continue to Step 2.

Step 2. Each pending packet of node i is calculated with node j for comprehensive forwarding weight τ. Here, τ_i of node i is calculated in the following:

$$\tau_i = \alpha\rho\left(p_{ij}^{\sigma}\right) + \beta L\left(L_i^r\right), \tag{4}$$

where α, β are weight factors. Here, σ represents the destination address of the packet and $\sigma \in \varphi$. $\rho(p_{ij}^{\sigma})$ is the probability of delivering proportion. It represents which of the two nodes is more likely to successfully deliver packet and it is calculated in formula (5). $L(L_i^r)$ represents the free buffer ratio function and can be derived from formula (6):

$$\rho\left(p_{ij}^{\sigma}\right) = \begin{cases} \dfrac{\omega_i^{\sigma}(t)}{\omega_i^{\sigma}(t) + \omega_j^{\sigma}(t)} & t \neq 0,\ \omega_{ij}^{\sigma}(t) \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

$$L\left(L_i^r\right) = \frac{L_i^r}{L_i^{\text{total}}}, \tag{6}$$

where L_i^{total} represents the buffer size of node i. A higher value indicates a higher carrying capacity.

Step 3. At each stop in this second traversal, GTDM determines whether or not node i transmits the current packet based on its comprehensive forwarding weight τ. Node i will be allowed to transmit the packet if the condition $\tau_i < \tau_j$ is met.

(6) When node i transmits a packet to the proxy node j, it represents a failure of node i in the game, and node i is punished by the function $\eta^*(c_i)$. Likewise, node j wins the game and it is rewarded by the function $\eta(c_i)$. $\eta(c_i)$, $\eta^*(c_i)$ are calculated as follows:

$$\begin{aligned} \eta\left(c_i\right) &= c_r^i + 1, \\ \eta^*\left(c_i\right) &= c_r^i - 1. \end{aligned} \tag{7}$$

The asset value v of each node is changed by function (4) after this delivery decision. There are two traverses for neighborhood selection and packets selection in GTDM; the time complexity of GTDM is $O(N^2)$.

4. Forwarding Strategy Based on GT

In this section, we present the forwarding strategy used when there are two nodes based on GT. Game theory is a mathematical description that works to resolve conflict among self-decision-making players. Unlike the GTDM algorithm, the common GT considers both of the mobile nodes as individuals who can refuse the packets from the others. This behavior increases the complexity of the algorithm. After the forward decision is determined using the GT approach, node i transmits all the packets. The transmission range of a mobile node in a city environment is likely to cover several nodes due to population density characteristics. In GTDM, set N is defined as a finite covered node set. The Sender Node (SN) broadcasts the information to the Receiver Node (RN), which is selected from the set N_i by the competition approach. The assets function represents the efficient functioning of node i and consists of two parts. The reward function allows the nodes to receive a profit for winning the game, while the punish function costs the nodes when they lose.

In DTNs deployed in a city, pedestrian nodes move regularly with WDM in a relatively small area, and the destination addresses of each node packet may belong to different fixed stations. Under such circumstances, the node must deliver each packet to an appropriate relay. In other words, the packet would select a relay which is more likely to deliver a packet successfully or wait for a better opportunity.

Figure 1 suggests that the active pedestrian areas are relatively small and that the pedestrians have a regular routine. Based on the previous assumptions, if node i or node j has delivered the packets to the fix station successfully, the sink station will be nearby one or both of them in WDM. We calculate the delivery probability proportion by $\rho(p_{ij}^{\sigma})$ to solve the problem of whether or not node i will transmit a packet to node j. According to the above factors, this strategy determines whether to transmit the packet to j or to wait for a better competitor.

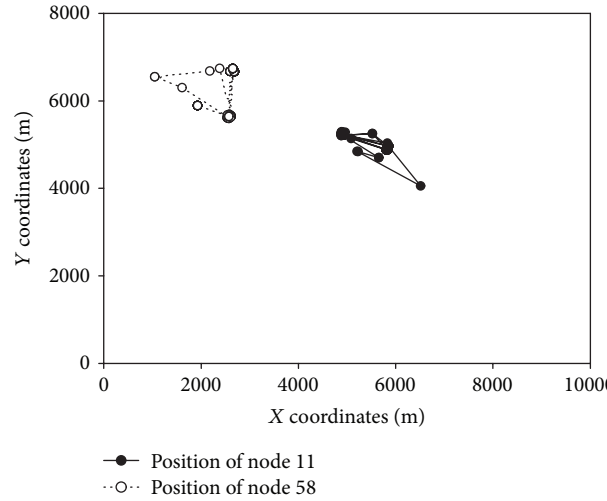


Figure 1: The positions of two pedestrian nodes in simulation map at sampled period for 3600 s.

Table 1: Game payment matrix of SN and RN: c_p represents the predicted cost value, and c_f represents the predicted profit value. c_r and c_s are the coin numbers of the RN and SN, respectively, before transmitting.

	RN	
	0	1
SN		
0	—	—
1	$c_s - c_p$, –	$c_s - c_p$, $c_r + c_f$

It is assumed that a value of 1 of SN represents the forwarding decision and a value of 0 represents the abandonment of forwarding. An RN value of 1 represents receiving and an RN value of 0 represents failure of receiving. Summing up the strategies and assets of SN and RN, the payment matrix in GTDM is shown in Table 1.

From this, we are able to gradually marginalize the inactive nodes. In this game, we use the Nash Equilibrium to make the RN transmission decision.

Selection Process

When RN selects 0: if SN selects 0, it will get the value 0. When SN selects 1, it will get $c_s - c_p$. Obviously, $(c_s - c_p) > 0$ and then SN should select 1.

When RN selects 1: if SN selects 0, it will get the value 0. When SN selects 1, it will get $c_r + c_f$. Obviously, $(c_r + c_f) > 0$, and then SN should select 1. The equilibrium is (1, 1), and it means that the SN and RN are encouraged to perform transmission and receiving. It is a Pure Strategy Nash Equilibrium.

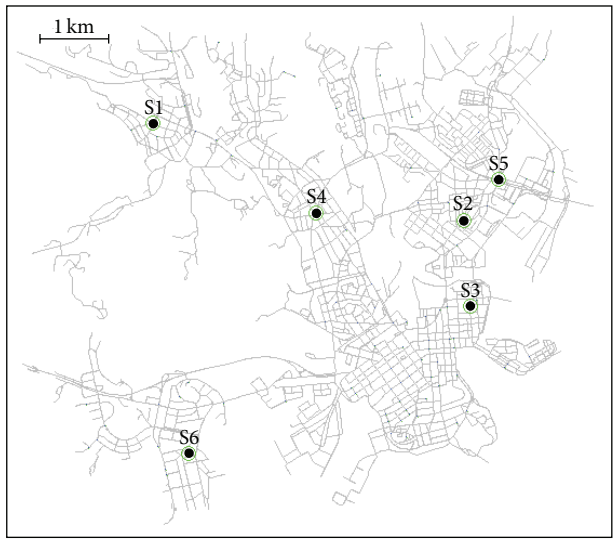


Figure 2: Fixed sink station in simulation scenario.

Table 2: Key simulation parameters.

Parameter name	Value
Word size	10000 ∗ 8000
Simulation time	5 days
City map	Helsinki
Buffer size	5 M
Message size	500 k–1 M
Message creation interval	10–15 seconds
Message TTL	1433 seconds
Transmission range for pedestrian	10 m
Transmission range for the others (taxi, station, and bus) nodes	100 m
Movement model for pedestrian	WDM
Movement model for taxi, bus	Shortest path of map based, bus movement model

Table 3: Key parameters set to GTDM algorithm.

Parameter symbol	Value
E_i^{ini}	300 K mAh
E_s, E_{sr}	1 mAh/s
E_{tr}	2 mAh/s
ΔE	0.1
α, β	0.5
γ	0.98
v_0	0.5

5. Simulation and Results

5.1. Simulation Design. The Opportunistic Network Environment (ONE) simulator [19] is chosen to realize the GTDM algorithm. Comparisons have been made between GTDM and the existing DTN routing algorithms, such as Epidemic and Prophet algorithms.

Several assumptions are associated with simulation. First, pedestrians, buses, fixed sink stations, and taxies are each represented by a different node type. The pedestrian nodes have limited energy and run on a regular routine. Using the energy model outlined in [19], the energy constrained mobile nodes are based on an energy budget approach and the energy consumption E_b of a node can be expressed as follows:

$$E_b = E_s + E_{\text{tr}} + E_{\text{sr}}, \tag{8}$$

where E_s represents the energy consumption associated with node scans. E_{tr} is the transmission energy consumption per second when a message is sent. E_{sr} is the energy consumption of each scanning response. The other types of nodes have no power constraints due to a constant power supply.

Secondly, the mobile nodes are the data source and relay nodes which forward the packets to the fixed sink stations. Six fixed station nodes are distributed at the relatively high path density associated with city environments (Figure 2).

Thirdly, for the purpose of mimicking "real world" environments as accurately as possible, each mobile node follows the WDM movement model. The WDM contains information about the locations of shops, houses, and offices. The residency time of a mobile node in any one location is controlled by the same mechanism. The map of downtown Helsinki is used as the deployment target. Additional city map details are provided by the ONE simulator [19].

5.2. Key Simulation Parameter Settings. In this simulation scenario, pedestrian nodes move regularly among shops, houses, and offices. These nodes are controlled by the WDM data set, which is provided by the ONE simulator. The key parameters of simulation scenario are shown in Table 2, and the assumed key parameters of GTDM simulation are shown in Table 3.

We should notice that we should fetch rational parameters. For instance, the mobile devices are commonly equipped with Bluetooth module, so the transmission range for pedestrian is 10 m.

5.3. Simulation Results. The DTN performance can be influenced by many factors such as the node density, packet size, and buffer size. Generally speaking, the more nodes lead us to encounter more opportunities, and then the game times in the high node density are more than the ones in the low node density. So, we observe the GTDM performance under different density.

The delivery ratio is important indicator for DTNs; the better algorithm should have relative higher delivery ratio. The delivery ratios of the three algorithms with equal parameters and settings are shown in Figure 3. We found that the delivery ratio of GTDM with the growth of node density is better than that of Epidemic or Prophet. At low node density, there is not a clear delivery ratio difference between Prophet and GTDM algorithms; it is more likely that the neighbor set N has only one node and it is not able to fully compete in the node selection. GTDM has better performance than

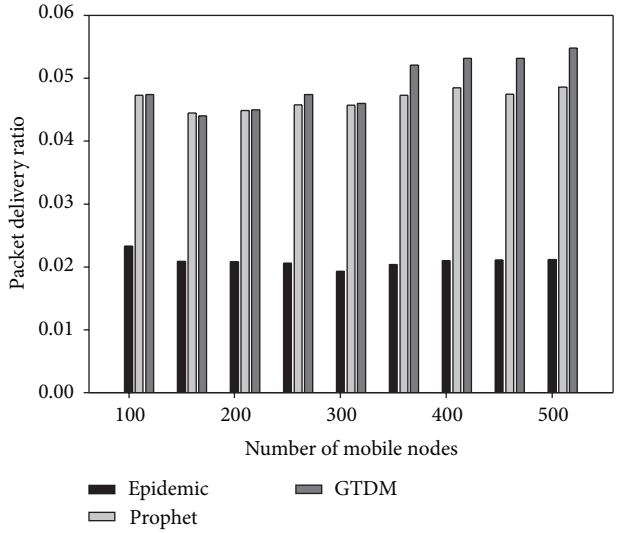


Figure 3: Packet delivery ratio comparison among different routing algorithms with mobile node density growth.

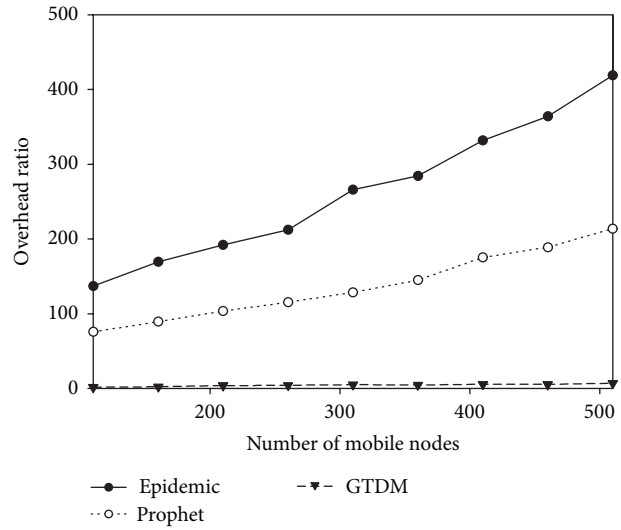


Figure 4: Overhead ratio comparison among different routing algorithms with mobile node density growth.

the other algorithms when DTNs are at high node density. The Epidemic algorithm has the worst performance due to its limited buffer size and unselected forwarding.

The overhead cost ratio of DTNs is another important performance indicator. One of the purposes for DTN routing algorithm design is to minimize the network overhead costs, which result from a high number of packet copies. The overhead ratio is described as repetition rate. A higher overhead ratio results in lower DTN delivery efficiency. The overhead ratio R_ξ is calculated in the following:

$$R_\xi = \frac{\sum_0^t \varpi_s - \sum_0^t \rho_s}{\sum_0^t \rho_s}, \tag{9}$$

where ϖ_s presents the successful transmitting times and ρ_s is the successful delivery times. If the R_ξ equals zero, it means that each transmitting behavior is a perfect delivery process.

The overhead ratio of GTDM is better than that of the other two algorithms (Figure 4). The overhead ratio of Epidemic and the Prophet algorithms becomes higher as the node density increases, meaning that the nodes cannot make efficient decisions as more opportunities become available.

The number of packets will increase with time lapse. The WDM captures the characteristic movement of real life scenarios, giving a greater number of contact opportunities at the peak of the crowd. The contact probability periodically changes over time (Figure 5). Prophet algorithm does not take into account this long time period. The proper transmission opportunity will be missed because of the predicted value decrease with time. Similarly, Epidemic algorithm forwarded too many packets during the low number of contacting periods. So, this behavior results in higher overhead ratio than others.

The delivery predictability P_{ij} of the Prophet algorithm impacts the decision-making and is influenced by an aging constant. For instance, even if a node has the ability to deliver

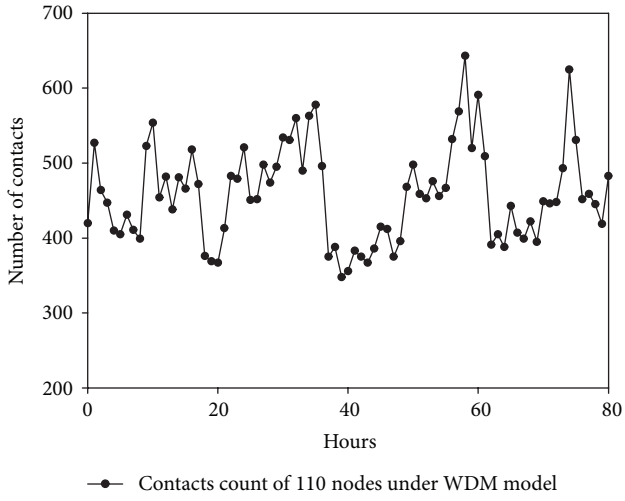


Figure 5: 110-node contact statistics under WDM.

a packet to the shop, its working time in the office may be much longer. In such case, P_{ij} decreases over time, which may result in losing a delivery opportunity. Therefore, Prophet is not a suitable algorithm for the city environment. Due to the characteristics of the WDM movement, the latency trends of the average curve fluctuate by the three algorithms (Figure 6). The result suggests that latency average of GTDM is less than that of Epidemic or Prophet all the time.

We have observed the 110-node survival status under the city scenario. Most of nodes are dead in Epidemic when time is 298800 seconds (Figure 7). The rest of nodes cannot maintain the normal performance of the system.

Likewise, most nodes exhaust the energy under the same condition with Prophet algorithm (Figure 8). The survived nodes are more in Prophet than in Epidemic routing.

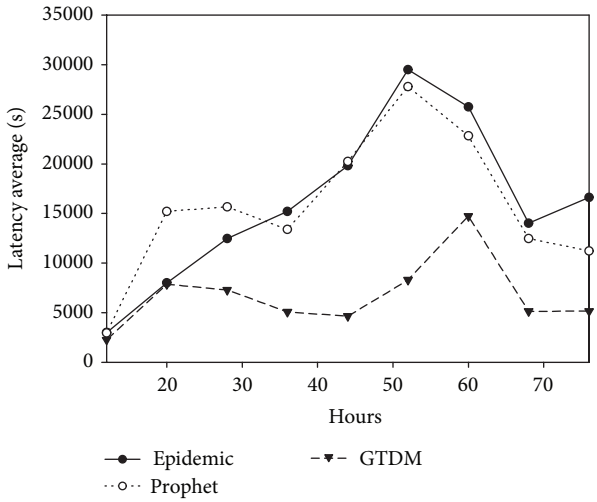


Figure 6: Latency average of Epidemic, Prophet, and GTDM with 110 nodes.

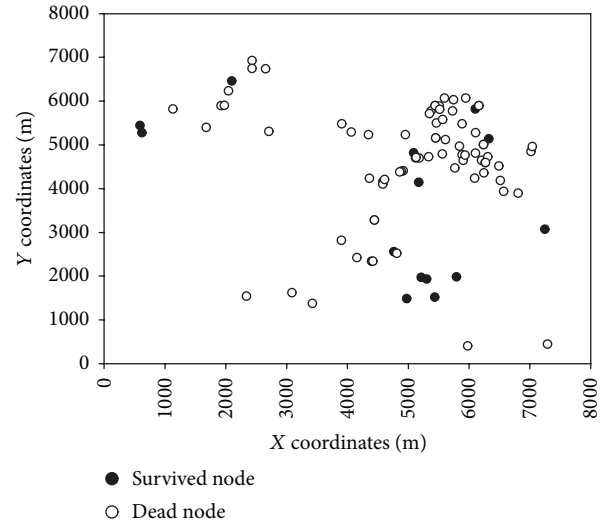


Figure 8: Survival status of mobile nodes in Prophet algorithm when time is 298800 s.

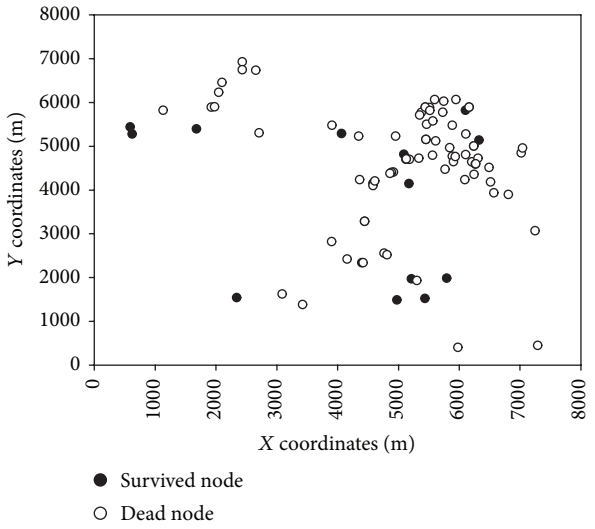


Figure 7: Survival status of mobile nodes in Epidemic algorithm when time is 298800 s.

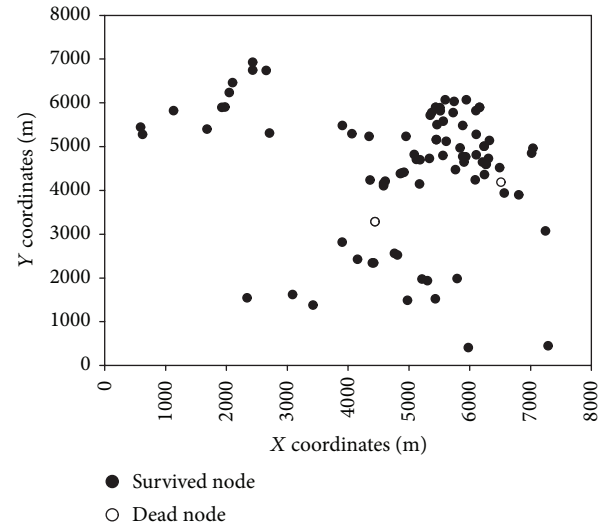


Figure 9: Survival status of mobile nodes in GTDM when time is 298800 s.

From Figure 9, it is shown that the survival nodes of GTDM algorithm are more than the other algorithms. It is clear that, in Epidemic routing, the node transmits packets to any node that it may come into contact with and the unlimited flooding algorithm will waste energy constantly. The Prophet algorithm transmits packets to a node with a relatively high delivery probability, but it is disturbed by the aging constant. Due to this, there are groups of node that have a similarly low delivery probability after a long period; thus, it is difficult to determine the relay node. Some nodes transmit the packets to others which have relatively high delivery probability under a period of time, and then the potential nodes will die out quickly. Obviously, the cascading effect will appear to shorten the DTNs lifetime.

In order to verify the energy balance by GTDM, the energy sample variance with time lapse is calculated by

$$S^2 = \frac{\sum_{i=1}^{n} \left(x_i - E\left(x\right)\right)^2}{n-1}, \tag{10}$$

where $E(x)$ represents the sample mean and x_i is the energy sample value. A higher S^2 value indicates higher energy gap in nodes, which could cause premature death of some nodes. From Figure 10, it is shown that GTDM algorithm works well in balancing energy consumption of mobile nodes by decreasing the quantity of failed nodes and prolonging the network lifetime.

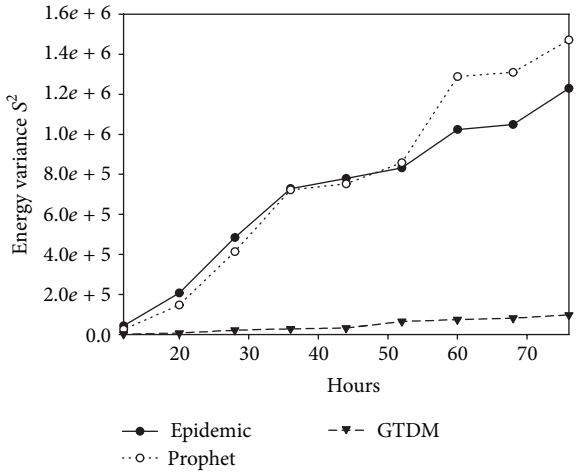


Figure 10: Energy sample variance of three algorithms.

6. Conclusion

Currently, the suitable DTN routing algorithm for city environments is one of the most important topics and it attracts numerous scholars' concern. We present GTDM algorithm, a routing algorithm of transmitting decision-making based on game theory. With the GTDM framework, several operations are implemented to adjust the transmitting queues and ensure that the proper node is chosen to transmit the proper packets. The overhead cost ratio, energy consumption, and average latency in GTDM are minimized through comparison with the algorithms of Epidemic and Prophet. Other simulation results by WDM, such as delivery ratio and network lifetime, support GTDM over Epidemic and Prophet algorithms in city environments. The provision, an algorithm with better energy balance, more efficient forwarding mode, and more flexible deployment, will help pave the way for future smart city networks and diverse applications of DTNs. Further studies are still necessary in the future; we should look for more efficient payoff functions for performance optimization and make further experimentation using real data set. In addition, there should be an efficient sleeping and awakening mechanism for energy-saving in city environments.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the persons who review and give some valuable comments to improve the paper quality. This work is also supported by Natural Science Foundation of China (nos. 61272448 and 71301182), Doctoral Fund of Ministry of Education of China (no. 20110181130007), Science and Technology Supporting Plan of Sichuan (nos. 2011RZ0004, 2012GZ0005, and 2015JY0047), and Soft Science Fund of Science and Technology Department of Sichuan (nos. 2014ZR0146 and 14YCG052).

References

[1] T. Watteyne and K. S. J. Pister, "Smarter cities through standards-based wireless sensor networks," *IBM Journal of Research and Development*, vol. 55, pp. 7:1–7:10, 2011.

[2] C. Jing, L. Ren, and D. Gu, "Geographical routing for WSN of street lighting monitoring and control system," in *Proceedings of the International Conference on Computer Design and Applications (ICCDA '10)*, pp. V3235–V3238, IEEE, Qinhuangdao, China, June 2010.

[3] G. Cardone, P. Bellavista, A. Corradi, and L. Foschini, "Effective collaborative monitoring in smart cities: converging MANET and WSN for fast data collection," in *Proceedings of the 4th ITU Kaleidoscope Academic Conference: The Fully Networked Human Innovations for Future Networks and Services (K '11)*, pp. 1–8, December 2011.

[4] H. Chourabi, T. Nam, S. Walker et al., "Understanding smart cities: an integrative framework," in *Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS '12)*, pp. 2289–2297, January 2012.

[5] D. Wang, F. Hong, B. Yang, Y. Zhang, and Z. Guo, "Analysis on communication capability of vessel-based ocean monitoring delay tolerant networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW '13)*, pp. 200–204, IEEE, April 2013.

[6] J. Wyatt, S. Burleigh, R. Jones, L. Torgerson, and S. Wissler, "Disruption tolerant networking flight validation experiment on NASA's EPOXI mission," in *Proceedings of the 1st International Conference on Advances in Satellite and Space Communications (SPACOMM '09)*, pp. 187–196, IEEE, July 2009.

[7] Z. Lu and J. Fan, "Delay/disruption tolerant network and its application in military communications," in *Proceedings of the International Conference on Computer Design and Applications (ICCDA '10)*, pp. V5-231–V5-234, Qinhuangdao, China, June 2010.

[8] X.-W. Zhou, Z.-M. Cheng, Y. Ding, J.-G. Lim, and Q. Liu, "Dynamic DTN routing strategies based on knowledge," *Wireless Personal Communications*, vol. 71, no. 3, pp. 1819–1836, 2013.

[9] L. Sundararaj and P. Vellaiyan, "Delay tolerant networking routing as a game theory problem—an overview," *International Journal of Computer Networks*, vol. 2, no. 3, pp. 152–172, 2010.

[10] H.-Y. Shi, W.-L. Wang, N.-M. Kwok, and S.-Y. Chen, "Game theory for wireless sensor networks: a survey," *Sensors*, vol. 12, no. 7, pp. 9055–9097, 2012.

[11] R. Machado and S. A. Tekinay, "A survey of game-theoretic approaches in wireless sensor networks," *Computer Networks*, vol. 52, no. 16, pp. 3047–3061, 2008.

[12] R. El-Azouzi, H. B. Sidi, J. Rojas-Mora, and A. P. Azad, "Delay tolerant networks in partially overlapped networks: a non-cooperative game approach," in *Bioinspired Models of Network, Information, and Computing Systems*, vol. 39 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 195–202, Springer, Berlin, Germany, 2010.

[13] M. R. Schurgot, C. Comaniciu, and K. Jaffrès-Runser, "Beyond traditional DTN routing: social networks for opportunistic communication," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 155–162, 2012.

[14] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *Internet and Network Economics*, vol. 4858 of *Lecture Notes in Computer Science*, pp. 306–311, Springer, Berlin, Germany, 2007.

[15] W. Z. Li, F. Lin, J. L. Zhou, and Y. Wang, "DTN routing with fixed stations based on the geographic grid approach in an urban environment," *Wireless Personal Communications*, 2015.

[16] T.-M. Pham and S. Fdida, "DTN support for news dissemination in an urban area," *Computer Networks*, vol. 56, no. 9, pp. 2276–2291, 2012.

[17] Y. Guo, S. Schildt, T. Pogel, and L. Wolf, "Detecting malicious behavior in a vehicular DTN for public transportation," in *Global Information Infrastructure Symposium (GIIS '13)*, pp. 1–8, October 2013.

[18] F. Ekman, A. Keränen, J. Karvo, and J. Ott, "Working day movement model," in *Proceedings of the 1st ACM SIGMOBILE Workshop on Mobility Models*, pp. 33–40, ACM, 2008.

[19] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, p. 55, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 636297, 7 pages
http://dx.doi.org/10.1155/2015/636297

Research Article

Secure Data Fusion in Wireless Multimedia Sensor Networks via Compressed Sensing

Rui Gao,[1,2] Yingyou Wen,[1,2] and Hong Zhao[1,2]

[1] *Key Laboratory of Medical Image Computing of Ministry of Education, Northeastern University, Shenyang 110819, China*
[2] *College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Yingyou Wen; yingyou_wen@163.com

Received 12 September 2014; Accepted 21 March 2015

Academic Editor: Qing-An Zeng

The paper proposes a novel secure data fusion strategy based on compressed image sensing and watermarking; namely, the algorithm exploits the sparsity in the image encryption. The approach relies on l_1-norm regularization, common in compressive sensing, to enhance the detection of sparsity over wireless multimedia sensor networks. The resulting algorithms endow sensor nodes with learning abilities and allow them to learn the sparse structure from the still image data, and also utilize the watermarking approach to achieve authentication mechanism. We provide the total transmission volume and the energy consumption performance analysis of each node, and summarize the peak signal to noise ratio values of the proposed method. We also show how to adaptively select the sampling parameter. Simulation results illustrate the advantage of the proposed strategy for secure data fusion.

1. Introduction

With the rapid development of wireless communication technologies, wireless multimedia sensor networks (WMSNs) are flourishing research fields due to the availability of low cost hardware such as CMOS and CCD cameras and the transmission of different kinds of data such as still images, multimedia streaming, and audio [1]. WMSNs not only enhance existing sensor network applications such as tracking, home automation, and environmental monitoring but also will enable several new applications like traffic avoidance, enforcement and control systems, advanced health care delivery, and so on [2]. Nowadays, when more and more sensitive information is transmitted over the WMSNs system, security has caught the attention of the research community with increasing multimedia applications of sensors [3].

Still images are important parts in WMSNs and usually used in the transmission of event triggered observations during short periods of time. Therefore, it is very important to protect still images from unauthorized access. Image encryption is one of the ways to ensure security and convert original image to another image that is hard to understand. Moreover, sensor nodes suffer from many constraints, including limited energy performances, low computation capability, less computational complexity, susceptibility to physical capture, tremendous transmission of multimedia data, and the use of insecure wireless communication channels [4]. Due to these limitations, it is essential to maximize the lifetime of sensors by reducing data transmission. Image data fusion is efficient in power consumption, so it plays a positive role in the computationally constrained and resource-constrained wireless multimedia sensor nodes.

In this paper, we propose a novel application of compressive sensing, namely, a lightweight still image encryption model for WMSNs security. We build our design on ideas from compressive sensing and watermarking. By analyzing the fact that image signal in WMSNs holds large amount of redundancy information, we project the captured image to the transform domain using only a few significant coefficients. Moreover, we utilize the watermarking approach to achieve authentication mechanism. These sparse coding coefficients are then deciphered using a novel reconstruction algorithm to recover the original image signal. We exploit the highly sparse data of the sensor nodes to obtain accurate

reconstruction with only a few measurements. The framework of the proposed algorithm is illustrated in Figure 1.

The key contributions of the paper are summarized as follows.

(i) The compressive sensing methods in signal domain and hierarchical fusion techniques are integrated for sensor data fusion, which is applied to wireless multimedia sensor networks.

(ii) An integrated image data fusion framework is proposed. The algorithm developed under this framework transmits effective volume of sensor data in different sensor nodes and extends the network lifetime.

This paper is organized as follows: Section 2 introduces the related work. In Section 3, we describe the redundancy analysis of still image and provide still image encryption and authentication model for WMSNs security. Then, the proposed scheme suggests data reconstruction scheme. Security analysis and performance evaluation are given in Section 4. Finally, Section 5 offers conclusions and future directions.

2. Related Work

Data fusion techniques for reducing the number of data transmissions by eliminating redundant information have been studied as a significant research problem. These studies have shown that data fusion in WMSNs may produce various trade-offs among some network related performance metrics such as energy, latency, accuracy, fault-tolerance, and security (Table 1). It can be classified into three categories according to the abstraction level of the sensor data: low-level fusion, medium-level fusion, and high-level fusion [5, 6].

Low-Level Fusion. It is also referred to as a signal value fusion. It combines several sources of raw data to produce new raw data. Wimmer et al. [7] suggested a combined analysis by descriptive statistics of image and video low-level descriptors for subsequent static SVM classification. This strategy also allowed for a combined feature-space optimization which would be discussed herein. Mieslinger et al. [8] proposed a new method for the adaption of satellite-derived solar radiation values to ground measured time-series that was developed. The method was tested at the two sites ESPSA and DZTAM and for the four satellite data providers DLR, EHF, GMS, and HC3. Wu et al. [9] presented a simplified analytical model for a distributed sensor network and formulated the route computation problem in terms of maximizing an objective function, which was directly proportional to the received signal strength and inversely proportional to the path loss and energy consumption. Low-level fusion can keep all knowledge for the final decision process and the ability of a combined value-space optimization. Gao et al. [10] proposed a similarity model and power model. The proposal scheme divided multimedia data into multiple different pieces and transmitted the effective pieces to the selected sensor nodes.

Medium-Level Fusion. It is also described as attribute level fusion. Attributes or features of an entity (e.g., shape, texture, and position) are fused to obtain a feature map that may be used for other tasks (e.g., segmentation or detection of an object). Wang and Yin [11] proposed a perceptual hashing-based robust image authentication scheme, which applied the distributed processing strategy for perceptual image hashes. The scheme protected multimedia data from unauthorized access but increased the energy consumption simultaneously. Fucai et al. [12] presented a novel partial dynamic reconfiguration image sensor node prototype for wireless multimedia sensor networks to transmit sensitive images. The method proposed in the paper enabled the partial dynamic reconfiguration to decrease the volume of data. Medium-level fusion

Table 1: The performance is measured by PSNR (dB).

Number of samples M	Peppers	Lena	Boats
2000	19.03	18.56	18.32
3000	22.34	21.23	21.16
4000	25.12	25.01	24.93
5000	28.20	28.02	28.09
6000	30.16	29.67	29.73

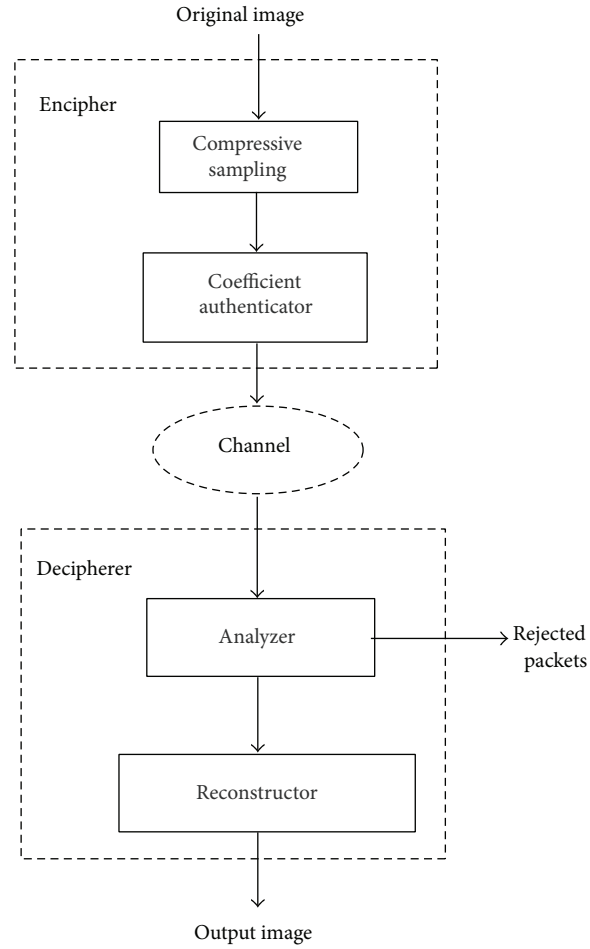


Figure 1: The framework of the algorithm.

reduces some information for the transmission process and achieves the combined attribute-space optimization.

High-Level Fusion. It is also known as decision level fusion. It takes decisions or symbolic representations as input and combines them to obtain a global decision. Blasch et al. [13] sought to describe MOEs for high-level fusion based on developments in Quality of Service and Quality of Information that supported the user and the machine, respectively. They defined HLF MOE based on information quality, robustness, and information gain. Costa et al. [14] proposed to address this issue with the use of highly expressive Bayesian models, which provided a tighter link between information coming from low-level sources and the high-level information fusion systems, and allowed for greater automation of the overall process. They illustrated the ideas with a naval HLIF system. High-level fusion needs higher artificial intelligence algorithm.

In short, secure data fusion has been incorporated into a wide range of existing sensor data. The crucial question is how to prolong the resource-constrained WMSNs lifetime to the longest time in secured mode. Thus, we analyze the advantages from different fields to build a novel image data fusion framework.

3. The Proposed Scheme

In this section, we explain the proposed scheme. We illustrate two stages in the proposed framework: still image encryption and decryption stages. The still image encryption stage gets several sources of image data and then projects the captured image data to the transform domain using only a few significant coefficients after analyzing the fact that image data in WMSNs holds large amount of redundancy information. In the decryption stage, we analyze the coherent measurements of image data from different sensor nodes. These measurements are then decoded using a novel reconstruction algorithm to recover the original image data.

3.1. Still Image Encryption. As everyone knows, WMSNs circumstance requires still images to minimize the energy consumption of data transmission as much as possible with security mode. The main target is how to securely and economically transmit the volume of image signal. Unlike common data, image signal in WMSNs usually holds large amount of redundancy information including spatial redundancy and surrounding redundancy [15], so the paper takes advantage of the characteristic to specify still image encryption model.

Spatial redundancy exploits the fact that an image very often contains strongly correlated pixels and the same brightness pattern repeated with statistic similarity [16]. The redundancy is explored by predicting a pixel value based on the values of its neighboring pixels. In addition, the availability of the same image block is represented by other blocks, due to the fact that light intensity and color in such areas are highly correlated. Thus, image represented by all pixel values suffers from a high level of spatial redundancy. The surrounding redundancy describes strongly correlated pixels of similar surroundings in different image signals. Natural images consist of separate areas indicating the object and its sceneries. We pay attention to the trends of the object in WMSNs and capture image signal containing the same surroundings. Thus, a high compression ratio can be achieved by eliminating these redundancies.

Compressive sensing has recently become very popular due to its interesting theoretical nature and wide area of applications [17]. Here, we consider compressive sensing in sensor data aggregation.

Suppose that x_i denotes image data of sensor node number i. Image data from sensor nodes is encoded by compressive sensing and is formally formulated as follows:

$$x = (x_1, x_2, \ldots, x_n). \tag{1}$$

Suppose that the natural image $x_i \in R^N$ is K sparse on some basis Ψ and x_i can be represented by a linear combination of K vectors of Ψ with $K \ll N$. We have $x_i = \Psi\theta$, where Ψ is sparse matrix and θ is $N \times 1$ vector with only K nonzero entries. The measurement is described as $y_i = \Phi \cdot \widehat{\Psi} x_i$, where Φ is $M \times N$ matrix with $M \ll N$ and y_i represents the M measurements [17]. In WMSNs applications, the noise is present; we define y as

$$y_i = \Phi \cdot \widehat{\Psi} x_i, \quad i \in (1, n). \tag{2}$$

According to the principle of discrete wavelet transform (DWT) sampling, the DWT coefficients to be measured should also cluster over low frequency bands. Typical measurements are chosen uniformly random. But, by placing the samples selectively but still in a random manner, we achieve better quality of image reconstruction. In this paper, we present a new sampling matrix designed in the frequency transform domain, which is applicable to the DWT. We define Φ as

$$\Phi = Q_M F W_N, \tag{3}$$

where F is the $N \times N$ diagonal matrix that is written as

$$F = \begin{bmatrix} F_f & & & \\ & F_f & & \\ & & \ddots & \\ & & & F_f \end{bmatrix}, \tag{4}$$

where F_f represents the $f \times f$ Fourier matrix. Q_M picks up columns of F randomly and W_N is a scrambling operator that randomly takes the N rows of $Q_M F$. Simply speaking, (2) takes the partial Fourier sampling over different frequency bands randomly. Obviously, we decrease the total amount of information to transmit and guarantee at the same time the expected security level. After compressive sampling, we utilize the authentication mechanisms to improve security and transmission efficiency. In the following, we discuss how to make the coefficient authenticator by watermarking approach.

Getting local time, we take the minute value m and second value s from WMSNs. The algorithm is shown in Figure 2.

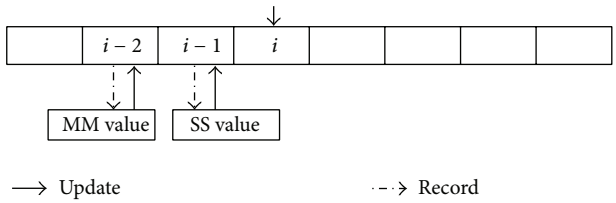


Figure 2: Authentication mechanism by watermarking approach.

Finding the ith sparse element in the y matrix, set the fact that $i = (m-1)\%M$, the $(i-1)$th element value is s, and the $(i-2)$th element value is m. In addition, record $(i-1)$th and $(i-2)$th element values. When the node receives the delivered package, it chooses the ith sparse element as a mark and decides the data integrity. Check $(i-1)$th and $(i-2)$th element values with the MM value and SS value of local time on WMSNs. If the values match, namely, $(i-1)$th element value is equal to the MM value and $(i-2)$th element value is equal to the SS value, update $(i-1)$th and $(i-2)$th element values. If $(i-1)$th element value is not equal to the MM value or $(i-2)$th element value is not equal to the SS value, the sensor node will reject the package.

3.2. Decryption Algorithm. In order to decrypt the image signal x from incoherent measurements y, we utilize l_1-optimization and min-TV solver. Obviously, we exploit the correlation of neighboring pixels and minimize the l_1 norm [18].

Considering the surrounding redundancy, we utilize the neighboring captured image signal $x_1, x_2, x_i, \ldots, x_n$ of x to reconstruct the signal x. The objective function is $\min \|x\|_1$ and set equality constraint $y = \Phi \cdot \widehat{\Psi} x = \Theta \cdot x$. The Θ is shown as

$$\Theta^* = \arg\min \sum_i \left\| y_i - \sum_j \Theta_{ji}(x_i) \right\|_1, \tag{5}$$

where $y_1, y_2, y_i, \ldots, y_n$ are the measurement vectors of $x_1, x_2, x_i, \ldots, x_n$. Calculate the gradient x_k^* in every iteration $(x - x_k)^T (y - \Theta \cdot x_k)$, and x_{k+1} is shown as

$$x_{k+1} = x_k + \partial \left(x_k^* - x_k \right), \tag{6}$$

where ∂ is an adjustable parameter. According to the redundant constraints image signal, minimize the TV of the reconstructed signal to reconstruct image signal x again. The object function is as follows:

$$\mathrm{TV}(x) = \arg\min \left(\mathrm{TV}_1(x) + \mathrm{TV}_2(x) \right), \tag{7}$$

where TV represents the TV between the pixels of the natural image x and $\mathrm{TV}_2(x)$ represents the TV of the surrounding vector. Firstly, find the matching pixels of each pixel at different surrounding vector, calculate the partial derivative of current location, and modify the corresponding parameter. Change the parameter value that is less than the given threshold, until the minimal value.

In short, the algorithm for still image decryption is summarized in Algorithm 1.

4. Simulation Results

In this section, extensive simulations are provided to illustrate the effectiveness of the proposed methods. We first compare the total transmission volume of conventional encryption algorithm with the proposed scheme and then describe the energy consumption of every sensor node on single hop networks. Considering an encryption scheme, we present malicious node detection rate (DR) or false alarm rate (FAR) with the variation of the proportion of malicious nodes. Finally, we analyze the performance of the decrypted signal. We have developed a simulator based on MATLAB implementation to evaluate our proposed scheme. Assume that 50 sensors are deployed uniformly in 100×100 (m) network field and the radius of communication is 25 m. All the images are assumed sparse in the Daubechies-4 wavelet domain and have a spatial resolution of 256×256. For the still image encryption, we present results with $f = 16$.

Figure 3 shows the total transmission volume in different numbers of sensor nodes in WMSNs. Obviously, the proposed scheme provides less transmission volume than the exiting scheme [19]. The paper provides an energy-efficient transmission of still image signal. From simulation analysis, the amount of transmission volume is reduced by about 18.3% than the schemes without processing.

Due to still image encryption scheme, different sensor node makes different energy consumption. We analyze the security performance of still image encryption and authentication. For comparison purposes, suppose that 50 sensors deploy on signal hop network. Inspired by [19], Figure 4 shows the energy consumption of every sensor node. The 1st node consumes 1950 mA energy and the 2nd node takes up 2050 mA, while the average consumption of other nodes is 500 mA. The 1st and 2nd nodes consume significant energy, and then they accelerate the death time. But our scheme reduces the total energy consumption of the whole wireless multimedia sensor networks. Figure 5 describes the transmission volume of every sensor node. Our scheme provides less transmission volume in most of the sensor nodes except for the 1st and 2nd nodes. But our scheme reduces the total transmission volume.

Considering the security scheme, the paper analyzes the security performance of multimedia data transmission. Figure 6 describes the malicious node detection rate (DR) with the variation of the malicious nodes proportion. Our work supposes that attack probabilities are 0.02, 0.25, and 0.5. With the increase in the proportion of malicious nodes, the DR shows a slowly downward trend. When the proportion of malicious nodes is in the range $[0, 0.25]$, we can obtain a higher malicious node detection rate. Figure 7 describes the false alarm rate (FAR) with the variation of the malicious nodes proportion. The paper supposes that attack probabilities are 0.02, 0.25, and 0.5. With the increase in the proportion of malicious nodes, the FAR continues with the upward trend correspondingly. When the proportion of malicious nodes is in the range $[0, 0.25]$, the result achieves a low false alarm rate.

Considering the complexity of image data, we group scenarios into three different types: simple, general, complex.

Input: Encrypted images $x_1, x_2, x_i, \ldots, x_n$
Output: Decrypted image signal y
Procedure:
(1) **Initialization**
$x^* = \arg\min |\nabla x|_1$
(2) **Repeat** following steps
Update the matching surrounding Θ^*
$$\Theta^* = \arg\min \sum_i \left\| y_i - \sum_j \Theta_{ji}(x_i) \right\|_1$$
Update the gradient of x_k^*
$x_k^* = \arg\min (x - x_k)^T (y - \Theta x_k)$
Update x by following optimization
$x^* = \arg\min |\nabla x|_1 + \sum_j \Theta_{ji}(x_i)$
Until there is no much changes.
(3) **Minimize** TV(x)
$\mathrm{TV}(x) = \arg\min (\mathrm{TV}_1(x) + \mathrm{TV}_2(x))$

Algorithm 1: Still image decryption.

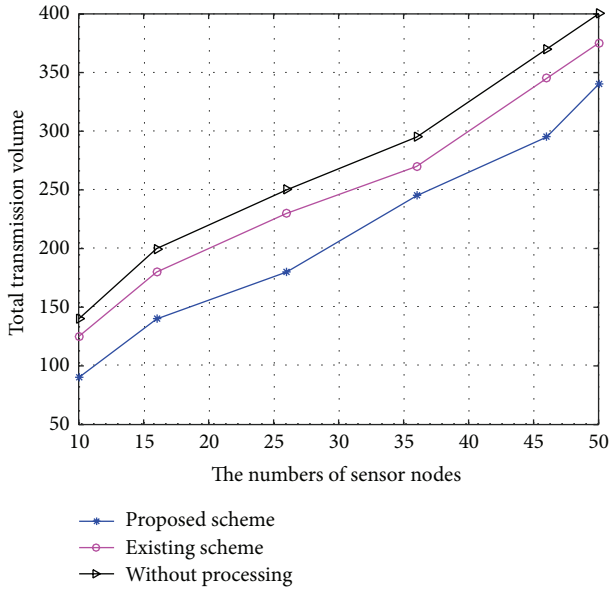


Figure 3: The total transmission volume of sensors nodes.

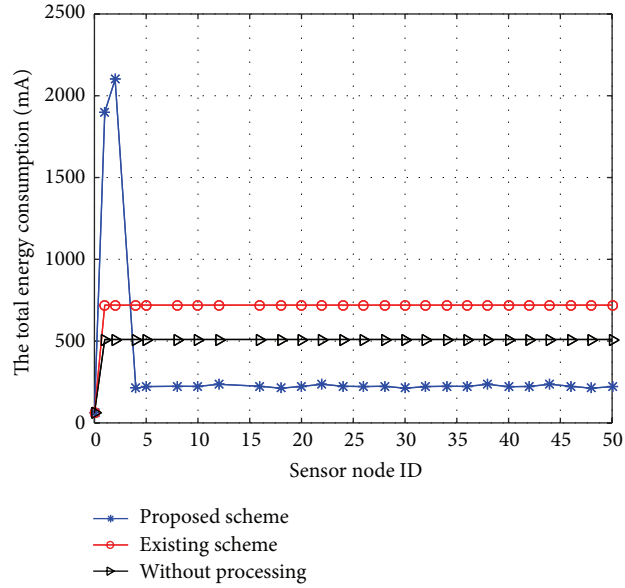


Figure 4: The energy consumption of every sensor node.

Figure 8 shows the MSE of the whole reconstructed image data from various scenarios. It can be observed that the image qualities of our proposed scheme, exiting scheme and the scheme without processing have errors in the process of compression and reconstruction during the data fusion. Along with the scene complexity increases, the proposed scheme makes more obvious immediate advantage.

Finally, to demonstrate the decryption practicality of the image signal, we apply it for three 256 × 256 images: Peppers, Lena, and Boats, and the data are collected from an average of ten runs. Table 1 summarizes the PSNR values when the number of measurements ranges from 2000 to 6000 (sampling ratio from 12.2% to 36.7%). Considering the limitation of bandwidth and power consumption, results of the measurements 6000 are even slightly better and the percentage of elements in our scheme is only 36.7%.

The simulations again show that the proposed still image encryption and authentication scheme consistently have better performance than other existing schemes. But some nodes consume significant energy and have a short life cycle.

5. Conclusion and Future Directions

This paper has proposed a secure conventional still image encryption algorithm. We develop an encryption algorithm

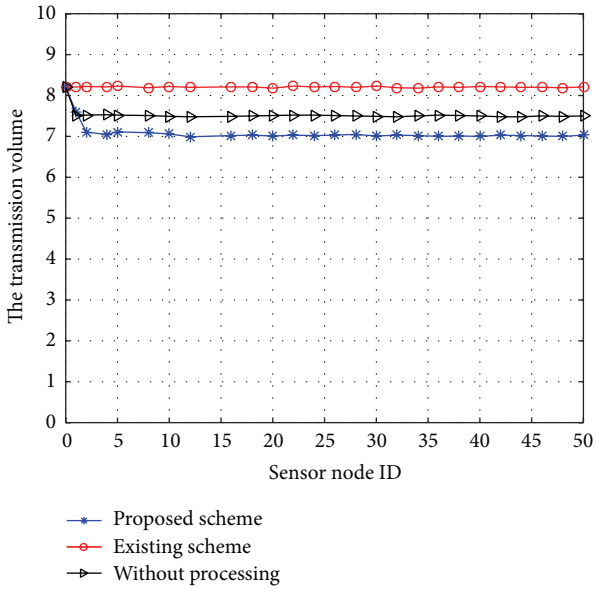


Figure 5: The transmission volume of every sensor node.

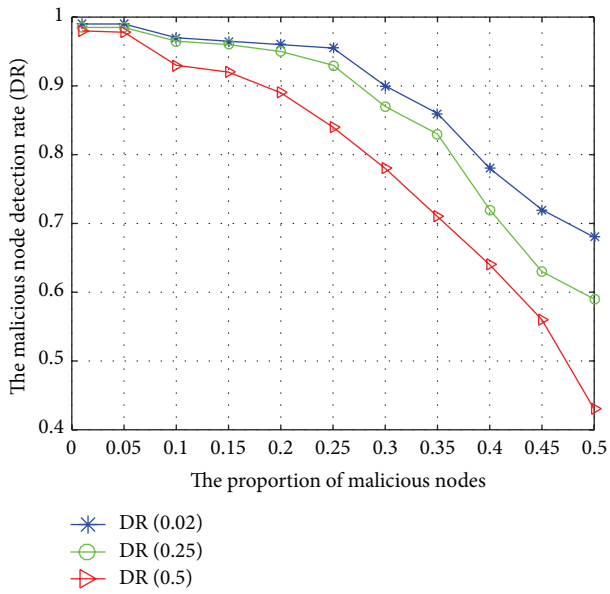


Figure 6: DR versus proportion of malicious nodes.

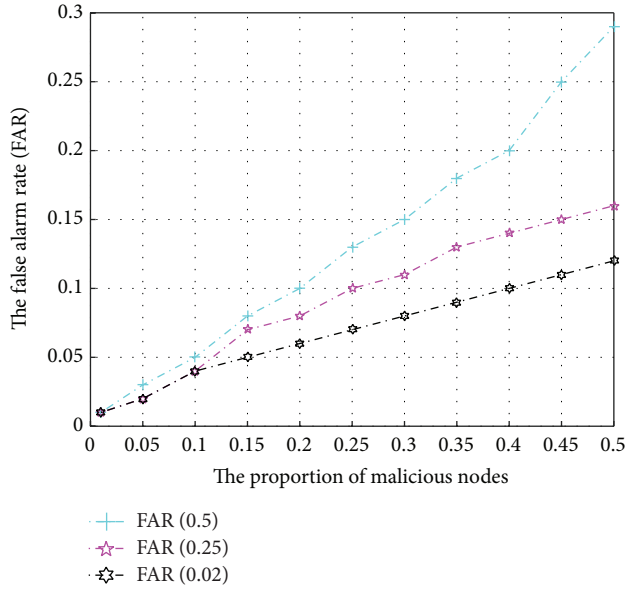


Figure 7: FAR versus proportion of malicious nodes.

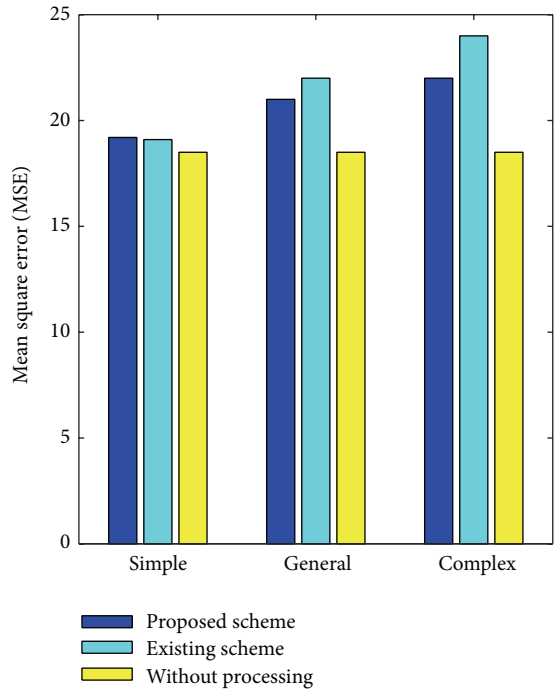


Figure 8: Mean square error of various scenarios.

based on compressed image sensing, which exploits the high degree of inherent sparsity in the natural images captured by sensor nodes and presents the fact that the image can be reconstructed using only a few acquisitions. We decrease the total amount of information to transmit and guarantee the expected security level. Moreover, the proposed design provides data authentication scheme that does not suffer from the influence of many existing malicious nodes. The proposed scheme has shown that the design transmits image signal securely and economically. In future work, we plan to extend our work to further improve compressive performance and conserve nodes energy.

Conflict of Interests

The authors declare that they do not have any commercial or associative interests that represent a conflict of interests in connection with the work submitted.

Acknowledgments

This work was supported by the Special Fund from the Central Collegiate Basic Scientific Research Bursary of China (Grant nos. 110818001 and 100218001) and in part by Grants from the National Natural Science Foundation of China (Grant nos. 60903159, 61173153, and 61300196).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.

[3] G. Z. Manel, Z. Ruken, M. José, and B. Kemal, "The future of security in wireless multimedia sensor networks," *Telecommunication Systems*, vol. 45, no. 1, pp. 77–91, 2010.

[4] C. S. Deshmukh and S. V. Dhopte, "Network environment using compressive sensing technique of computer science and applications," *International Journal of Survey on Video Coding in Wireless Multimedia Sensor*, vol. 6, no. 2, pp. 13–17, 2013.

[5] E. F. Nakamura and A. A. F. Loureiro, "Information fusion in wireless sensor networks," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1365–1371, June 2008.

[6] M. Guerrero-Zapata, R. Zilan, J. M. Barceló-Ordinas, K. Bicakci, and B. Tavli, "The future of security in wireless multimedia sensor networks," *Telecommunication Systems*, vol. 45, no. 1, pp. 77–91, 2010.

[7] M. Wimmer, B. Schuller, and D. Arsic, "Low-level fusion of image, video feature for multi-modal emotion recognition," in *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP '08)*, pp. 145–151, Madeira, Portugal, January 2008.

[8] T. Mieslinger, F. Ament, K. Chhatbar, and R. Meyer, "A new method for fusion of measured and model-derived solar radiation time-series," *Energy Procedia*, vol. 48, pp. 1617–1626, 2014.

[9] Q. Wu, N. S. V. Rao, J. Barhen et al., "On computing mobile agent routes for data fusion in distributed sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 740–753, 2004.

[10] R. Gao, Y. Wen, H. Zhao, and Y. Meng, "Secure data aggregation in wireless multimedia sensor networks based on similarity matching," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 494853, 6 pages, 2014.

[11] H. Wang and B. Yin, "Perceptual hashing-based robust image authentication scheme for wireless multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 791814, 9 pages, 2013.

[12] L. Fucai, J. Zhiping, and L. Yibin, "A novel partial dynamic reconfiguration image sensor node for wireless multimedia sensor networks," in *Proceedings of the IEEE 9th International Conference on Embedded Software and Systems*, pp. 1368–1374, 2012.

[13] E. Blasch, P. Valin, and E. Bosse, "Measures of effectiveness for high-level fusion," in *Proceedings of the 13th Conference on Information Fusion (FUSION '10)*, pp. 1–8, July 2010.

[14] P. C. G. Costa, K. B. Laskey, K. C. Chang, W. Sun, C. Y. Park, and S. Matsumoto, "High-level information fusion with bayesian semantics," in *Proceedings of the 9th Bayesian Modelling Applications Workshop*, 2012.

[15] C. Tang and C. S. Raghavendra, "Compression techniques for wireless sensor networks," in *Wireless Sensor Networks*, pp. 207–231, Springer, New York, NY, USA, 2004.

[16] S. K. Kil and J. S. Lee, "Lossless medical image compression using redundancy analysi," *International Journal of Computer Science and Network Security*, vol. 6, no. 1, pp. 50–56, 2006.

[17] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[18] l1-magic, http://www.l1-magic.org.

[19] E. J. Duarte-Melo and M. Liu, "Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 21–25, Taipei, Taiwan, November 2002.

Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 438290, 4 pages
http://dx.doi.org/10.1155/2015/438290

Research Article

An Improved Ant Colony Routing Algorithm for WSNs

Tan Zhi and Zhang Hui

Beijing University of Civil Engineering, Beijing 10044, China

Correspondence should be addressed to Tan Zhi; zhtan@bjtu.edu.cn

Received 12 December 2014; Accepted 4 March 2015

Academic Editor: Qing-An Zeng

Ant colony algorithm is a classical routing algorithm. And it are used in a variety of application because it is economic and self-organized. However, the routing algorithm will expend huge amounts of energy at the beginning. In the paper, based on the idea of Dijkstra algorithm, the improved ant colony algorithm was proposed to balance the energy consumption of networks. Through simulation and comparison with basic ant colony algorithms, it is obvious that improved algorithm can effectively balance energy consumption and extend the lifetime of WSNs.

1. Introduction

Ant colony algorithms that emerged by M. Dorigo et al. [1, 2] observe the behaviors of ants foraging for food. It has been quite popular for a wide variety of discrete optimization problems such as the traveling salesman problem, quadratic assignment problem, and job-shop scheduling [3]. Due to the characteristics of self-organization, autooptimizing, and dynamic topology, an ant colony algorithm is applied to the Ad Hoc network, wireless sensor network, and so on. Ant colony based routing algorithm is improved continuously. Kassabalidis et al. [4] proposed Ant-Net algorithm which realizes routing optimization through the forward ants and return ants (forward ants collect node information; return ants use this information update routing table). And ABC algorithm [5] (ant based control algorithm) is based on probability of mode selection and updates the path. This algorithm is only one kind of ants released from source nodes and these ants arrived at destination node after death. The node's routing table will be updated, when the ant arrives at the destination node.

In the paper, we propose improved ant colony algorithm to acquire the optimal path by routing optimization. We try to save the energy consumption and prolong network life under the condition that the path meets the needs.

2. Dijkstra Algorithm

2.1. Dijkstra Algorithm. Dijkstra algorithm is proposed by Dutch computer scientist Edsger Wybe Dijkstra to solve problem about the shortest path from an original point to other points in the directed graph.

Every point updates the shortest path information from the original point. It is usually defined as follows.

Suppose V and T are two sets that represent original point and other points.

At the initial moment, O is subsets and coverage of V and $t_1, t_2, \ldots, t_m$ are subsets and coverage of T.

Step 1. The distance $D(t_i)$ is the shortest distance from original point O to other points except to t_i. Calculating $D(t_i)$, we could get point x which is the closest point to point O in the set T.

Step 2. Stop if the set T is null; otherwise return to Step 1, where $S = S \cup \{x\}$, $T = T - \{x\}$.

3. Improved Ant Colony Routing Algorithm

In the ant colony algorithm model, wireless sensor network could be described as an undirected graph. At the beginning, the lack of initial pheromone leads to low solving speed

and high consumption, which has affected the overall performance of ant colony algorithm [6, 7]. In order to solve the problem, the improved ant colony algorithm based on Genetic-Ant Colony algorithm [8] is presented. However, it might cause an increase of the break in data transmission after crossover and mutation. In the paper, the idea have been proposed about Dijstra algorithm to improved ant colony algorithm distributing the nodes into the directed graph. It could increase the initial efficiency of algorithm and ensures the stability.

3.1. Pheromone Initial Optimization. Node number m and communication radius R are determined. Let $V_i = \Phi$, $i = 1, 2, \ldots, m$, be the set recording the previous node number and $P_i = 0$ the node coefficient. Let d_{ij} be the traffic demand between the node i and the node j.

Step 1. End point communicates with adjacent nodes. The adjacent nodes P ($k = 1, 2..., m$) record the number of termination point and update node coefficient.

If d_{o_2k} is less than the R, the P_k is set as one and O_2 is put into the set V_k.

Step 2. Get the number S of nodes which could direct communication with start point. The initial value of S is zero, and the value plus one when d_{o_1k} is less than the R.

Step 3. Node j communicates with adjacent node k. When $P_j < P_k$ or $P_k = 0$, the node k updates p and records number of node j until start point gets number of its adjacent nodes ($j = 1, 2..., m$).

Through the operation, the node gets the information about routes from the node to the end point. The routes have no back haul.

3.2. Path Choices. Simulated with the search food process of ant colony, the model for basic ant colony algorithm is as follows.

Suppose S is a set and $s_1, s_2, \ldots, s_m$ are subsets and coverage of S and, at the initial moment, they are selected; ant will be randomly placed on the m-subsets, assuming that the initial information of each subset $\tau_{ij}(0) = C$. The probability of ant k transfer from the subset i to subset j is [9]

$$p_{ij}^k = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{t\in \text{allowed}_t}\left[\tau_{it}\right]^{\alpha}\left[\eta_{it}\right]^{\beta}}, & j \in \text{allowed}_t \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Among them, t is the iteration number, k is ID ($k = 1, 2, \ldots, m$) for ants, and t is the iteration number; allowed t is the next subset selected from ant k; $\tau_{ij}(t)$ is the pheromone strength from subset i to subset j; η_{ij} is the inspired degree of ant k shifted from subset i to the subset j. These two parameters α and β are accumulation of information and inspired information in the process of ant's sports, reflecting the relative importance of ants to choose the next subset.

In order to balance the energy consumption of nodes, improved ant colony algorithm entered the energy factor based on the basic ant colony routing algorithm to find shorter and high energy path. Then, the improved probability from node i to node j is defined as follows:

$$p_{ij}^k = \begin{cases} \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}\left[\xi_{ij}\right]^{\chi}}{\sum_{t\in \text{allowed}_t}\left[\tau_{it}\right]^{\alpha}\left[\eta_{it}\right]^{\beta}\left[\xi_{it}\right]^{\chi}}, & j \in \text{allowed}_t \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Using the relative factor represents normalization of method, where ξ is relative energy factor which equals residual energy of node j divided by the initial node energy and multiplied by 1.2. The parameters χ are accumulation of information and inspired information in the process of ant's sports, it reflects the relative energy consumption of ants to choose the next subset.

3.3. Pheromone Update. Subset p is determined, and ant will stop when elements that selected subset contained target node; this will mark the end of the cycle. After all the ants have completed a cycle, the pheromone of subsets is adjusted according to the following equations:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \sum_{k=1}^{m}\Delta\tau_{ij}^k,$$
$$\Delta\tau_{ij}^k = \begin{cases} \dfrac{Q}{L_k}, & \text{if } j \in \text{solution of ant } k \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Among them, $(1 - \rho)$ is the attenuation coefficient of the pheromone; usually $\rho \in [0, 1]$ to avoid unlimited accumulation of informational on subset; L_k is the number of subsets and k selected in this cycle; Q is the pheromone strength; it affected the convergence speed of algorithm to a certain extent.

4. Simulation Result

Computation on simulation examples and comparison with basic ant colony routing algorithm show that improved ant colony routing algorithm is effective. As shown in Figure 1, we simulate a stationary network with 48 sensor nodes deployed in the target region of 80 m × 80 m square region. We also initialize that battery energy of each sensor is 1 J and the sensing radius R of each node is 30 m. The default value of ant colony algorithm parameters is set to $\alpha = 1$, $\beta = 2$, $\chi = 2$, $\rho = 0.7$, $Q = 100$, and $\tau_{\text{int}} = 0$.

Comparing the life cycle of two algorithms, we suppose energy consumed is 0.1 J by receiving and sending information. As is shown in Figure 2, the solid lines signify that improved ant colony routing algorithm and the dotted line represent the basic ant colony routing algorithm. The total energy of the algorithms is the same at the beginning. As time moved on, the improved ant colony routing algorithm has longer running times. After many time tests, as is shown in Figure 3, improved ant colony routing algorithm has longer life cycle, and it is a good way to save resources.

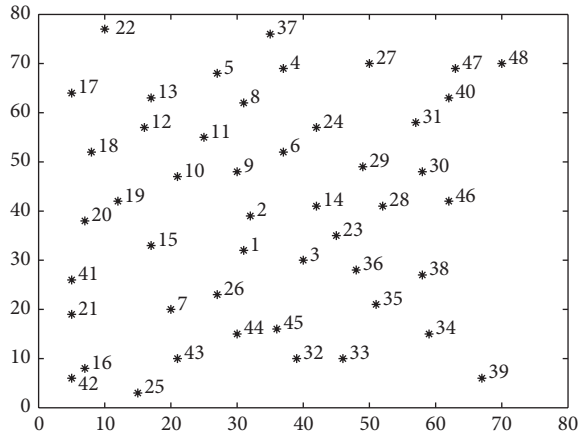


Figure 1: Node distribution.

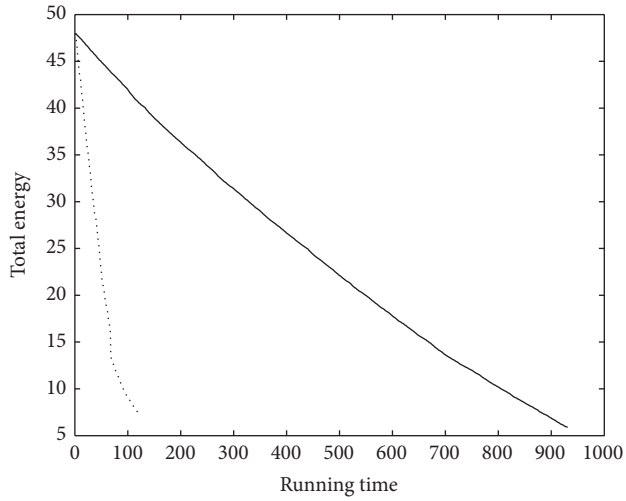


Figure 2: Comparison of the duration of the energy consumption in two algorithms.

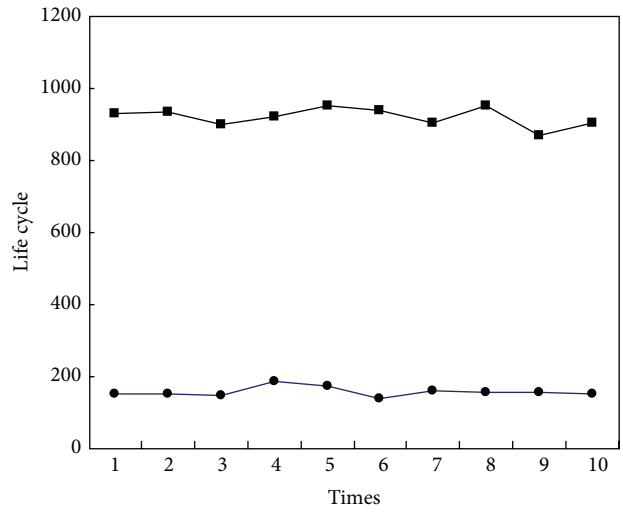


Figure 3: Comparison of life cycle in two algorithms.

Table 1: Average routing distance comparison.

Algorithm	Improved ant colony routing algorithm	Basic ant colony routing algorithm
1	114.354	304.503
2	114.518	269.581
3	114.700	267.072
4	114.261	319.672
5	114.559	339.222
6	113.953	266.705
7	113.710	301.923

On the other hand, Table 1 shows the comparison with the average routing path of the two algorithms. The average routing path of improved ant colony algorithm has reduced to about a third of basic ant colony algorithm. The improvement measures make the routing reduce. At the same time, the improved algorithm also reduces the energy consumption. We can see that the improved ant colony routing algorithm has more optimum routing path to guarantee the message fluency.

5. Conclusions

The wireless sensor network has distinguishing characteristics such as weak node calculation ability and energy limited node, so we should make it efficiency and save sources in the design of wireless sensor network routing algorithm. Ant colony algorithm is a new heuristic searching algorithm, which has many advantages in route optimization but wastes some time and energy due to pheromone deficiency. Therefore, this paper puts forward improved ant colony routing algorithm, which is inspired by the Dijkstra algorithm changing the wireless sensor network undirected graph to directed graph and energy equilibrium consumption ideas to improve the ant colony algorithm.

Compared with basic ant colony routing algorithm, improved ant colony routing algorithm is an algorithm with low energy and has high performance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by Beijing Education Science and Technology Development Program (KM201110016015).

References

[1] M. Gunes and O. Spaniol, "Routing algorithms for mobile multi-hop ad-hoc networks," in *Proceedings of the International Workshop on Next Generation Network Technologies*, pp. 20–27, 2002.

[2] Z. Modi, S. Jardosh, and P. Ranjan, "Optimized rumor routing algorithm for wireless sensor networks," in *Proceedings of the 5th International Conference on Wireless Communication and Sensor Networks (WCSN '09)*, pp. 1–6, December 2009.

[3] J. Kahn, R. Katz, and K. Pister, "Next century challenges: mobile networking for 'Smart Dust'," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 271–278, ACM Press, 1999.

[4] I. Kassabalidis, M. A. El-Sharkawi, R. J. Marks, P. Arabshahi, and A. A. Gray, "Swarm intelligence for routing in communication networks," *Global Telecommunications*, vol. 6, no. 6, pp. 3613–3617, 2001.

[5] B. McBride, C. Scoglio, and S. Das, "Distributed biobjective ant colony algorithm for low cost overlay network routing," in *Proceedings of the International Conference on Artificial Intelligence (ICAI '06)*, vol. 2, pp. 518–521, June 2006.

[6] S. Begum, N. Tara, and S. Sultana, "Energy-efficient target coverage in wireless sensor networks based on modified ant colony algorithm," *International Journal of Ad hoc, Sensor & Ubiquitous Computing*, vol. 1, no. 4, pp. 29–36, 2010.

[7] F. Zhu and H. Wang, "A modified ACO algorithm for multicast state scalability problem based on multicast tree similarity," in *Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT '11)*, pp. 972–976, Seoul, Republic of Korea, February 2011.

[8] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, and R. Rastogi, "Load balancing of nodes in cloud using ant colony optimization," in *Proceedings of the 14th International Conference on Computer Modelling and Simulation (UKSim '12)*, pp. 3–8, March 2012.

[9] P. Purkayastha and J. S. Baras, "Convergence results for ant routing algorithms via stochastic approximation and optimization," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 340–345, December 2007.