Energy-efficient Low-delay TDMA Scheduling Algorithm for Industrial Wireless Mesh Networks

Yun Zuo[1,2] , Zhihao Ling[1,2] and Luming Liu[1,2]
[1] School of Information Science and Engineering, East China University of Science and Technology
Shanghai 200237, China
[2] Key Laboratory of Advanced Control and Optimization for Chemical Process, Ministry of Education
Shanghai 200237, China
[e-mail: happyxiaoyunzi@126.com, zhhling@ecust.edu.cn, llm361163@sohu.com]
*Corresponding author: Zhihao Ling

Received February 18, 2012; revised May 16, 2012; revised July 1, 2012; revised September 5, 2012; accepted September 28, 2012; published October 29, 2012

Abstract

Time division multiple access (TDMA) is a widely used media access control (MAC) technique that can provide collision-free and reliable communications, save energy and bound the delay of packets. In TDMA, energy saving is usually achieved by switching the nodes' radio off when such nodes are not engaged. However, the frequent switching of the radio's state not only wastes energy, but also increases end-to-end delay. To achieve high energy efficiency and low delay, as well as to further minimize the number of time slots, a multi-objective TDMA scheduling problem for industrial wireless mesh networks is presented. A hybrid algorithm that combines genetic algorithm (GA) and simulated annealing (SA) algorithm is then proposed to solve the TDMA scheduling problem effectively. A number of critical techniques are also adopted to reduce energy consumption and to shorten end-to-end delay further. Simulation results with different kinds of networks demonstrate that the proposed algorithm outperforms traditional scheduling algorithms in terms of addressing the problems of energy consumption and end-to-end delay, thus satisfying the demands of industrial wireless mesh networks.

Keywords: TDMA scheduling algorithm; industrial wireless mesh networks; genetic algorithm; simulated annealing algorithm

This work was supported by the National High Technology Research and Development Program of China (863 Program) under Grant 2011AA040103 and the Shanghai Leading Academic Discipline Project under Grant B504.

http://dx.doi.org/10.3837/tiis.2012.10.004

1. Introduction

In recent years, wireless technology has become a hot spot in industrial automation. However, because of several limitations, wireless technology is not entirely suitable for industrial applications. With the release of WirelessHART in 2007 [1], the process automation industry was given access to its first open wireless communication standard.

Industrial applications usually require real-time data collection. Meanwhile, the improvement of energy efficiency is an essential problem because nodes are battery-powered. Time division multiple access (TDMA) is a widely used media access control (MAC) technique that can provide collision-free communications, save energy by using the sleeping scheme, bound the delay of packets, and guarantee reliable transmission. Thus, TDMA is a suitable option for satisfying the requirements of industrial applications.

To better meet the needs of industrial applications, further studies are required on the TDMA scheduling scheme that can allocate time slots. A proper scheduling scheme should not only avoid communication conflict by silencing the interference in one time slot, but should also be capable of minimizing the number of time slots. In addition, high energy efficiency and low delay should be achieved.

1.1 Related Works

A number of scholars have recently conducted researches on TDMA scheduling algorithms to minimize the total number of time slots. Kalyan et al. [2] used particle swarm optimization (PSO) algorithm to minimize the overall transaction time. This method was modeled as a graph partitioning-based approach to group the sensors for parallel operation and scheduling. Gandham et al. [3] proposed a distributed edge coloring-based link scheduling algorithm to realize the feasible time slot allocation with a minimal number of time slots. Ergen et al. [4] adopted two heuristic centralized algorithms based on coloring method of graph theory to determine the shortest length of conflict-free assignment of time slots. However, these three studies did not consider the problems of energy conservation and reducing delay.

Energy conservation has also been studied along with slot allocation in TDMA scheduling algorithms. Pei et al. [5] combined passive clustering and TDMA to reduce the power consumption of large wireless sensor networks dramatically. In this study, the roles of cluster heads and gateways were rotated to conserve energy. The work of Jolly et al. [6] addressed energy conservation by minimizing the number of state transitions between active and sleep modes through the Tabu search-based technique. Shi et al. [7] built a nonlinear cross-layer optimization model to reduce overall energy consumption and end-to-end delay. The algorithms in [5][6][7] were based on the clustering scheme. Kulkarni et al. [8] discussed energy-efficient methods to save power by completely switching off the nodes' radio. However, the energy consumed by switching state needs to be further reduced. Ma et al. [9] proposed an energy efficient algorithm that only actives each node once to receive all data from its neighbors. Meanwhile, switching delay is reduced. However, this approach neglected the buffer size constraint of each node and the queuing delay of each packet.

Meanwhile, other researchers introduced hybrid intelligence algorithms to solve TDMA scheduling problem in many-to-one sensor networks for energy conservation. Mao et al. [10] presented a hybridized genetic algorithm (GA) and PSO algorithm to enhance searching capability. Minhazul et al. [11] used a modified discrete differential evolution algorithm with a good searching capability to determine the optimal slot allocation scheme. These two algorithms saved the energy consumed on state switching between the active and sleep states. However, these two studies did not consider how to reduce delay.

Several other studies showed how to reduce delay. Sridharan [12] presented a linear programming formulation and a corresponding distributed solution that improved fairness and shortened delay. Cui et al. [13] adopted relaxation methods to reduce delay by shortening queuing delay of each packet. Djukic et al. [14] formulated a min-max program to minimize the maximum delay along the longest path in the tree. However, the algorithms in [13][14] neglected switching delay. Nikolaos et al. [15] used the path-wakeup and wakeup message aggregation strategies to minimize the sleep-related delay, which resulted in control message overhead.

To overcome the aforementioned shortcomings, this paper considered the reduction in both energy consumption and delay attributed to the switching state.

1.2 Features of the Present Work

In this paper, the desired objectives are to minimize the total number of time slots, as well as to achieve high energy efficiency and low delay. To solve the multi-objective TDMA scheduling problem, a hybrid algorithm is proposed, which can provide energy-efficient and low-delay communications for industrial wireless mesh networks. This algorithm is based on two intelligence optimization techniques: GA and simulated annealing (SA) algorithm. A number of critical techniques are also adopted in the proposed algorithm. The main contributions of this study include the following:

(1) To solve the TDMA scheduling problem in industrial wireless mesh networks effectively, a hybrid intelligence algorithm combining GA and SA is proposed. This algorithm can determine the optimal slot allocation scheme, as well as strike a balance among the number of time slots, energy consumption, and end-to-end delay.

(2) To reduce energy consumption and shorten end-to-end delay further, a number of critical techniques are adopted in the hybrid algorithm.

The reminder of this paper is organized as follows. Section 2 introduces the industrial wireless mesh networks. Section 3 describes the scheduling model and defines the problem. Section 4 details the hybrid algorithm. Section 5 shows the simulation results. The conclusions are presented in Section 6.

2. Industrial Wireless Mesh Networks

The industrial control environment is harsh, thus there are stringent requirements for reliable and real-time communication. Missing or delaying process data may severely degrade control quality. The power limitation caused by battery usage worsens the problem.

WirelessHART is the first global wireless communication standard specifically designed for process measurement and control applications. WirelessHART is based on the IEEE 802.15.4-2006 physical layer and defines its own time synchronized data link layer. The data link layer specifies the strictly 10 ms timeslot, network-wide time synchronization, channel hopping, channel blacklisting, and industry-standard AES-128 ciphers and keys. The network layer supports self-organizing and self-healing mesh networking techniques. Thus, messages can be routed around interferences, greatly improving network performance in harsh environments [16].

The structure of a WirelessHART network is illustrated in **Fig. 1.** A WirelessHART network contains three major elements:

(1) WirelessHART field devices that are connected to the process or plant equipment.

(2) WirelessHART gateway acting as a bridge that connects the network manager to field devices and is also used to convert one protocol to another.

(3) WirelessHART network manager that is responsible for configuring the network, scheduling communications between nodes, managing routing tables, and monitoring the health status of the network.

However, WirelessHART provides little guidance on how to meet the demands of industrial applications. This paper focuses on the TDMA scheduling scheme that can guarantee energy-efficient and low-delay communications in networks.

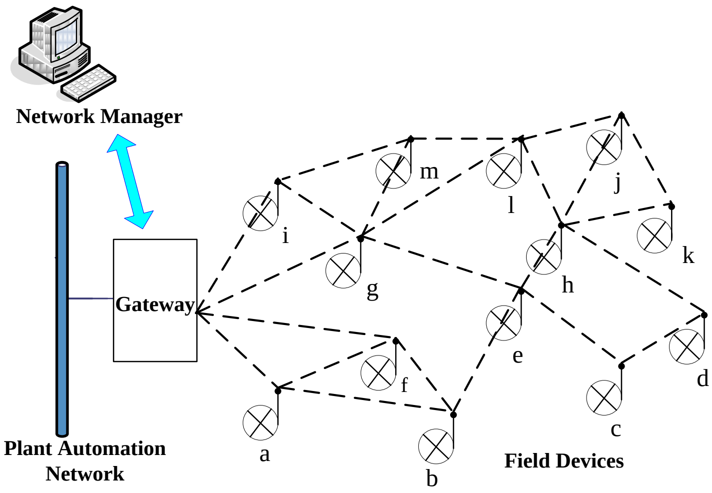


Fig. 1. The structure of a WirelessHART network

3. Problem Formulation

3.1 Network Model and Scheduling Model

An industrial wireless mesh network can be represented as a simple connected graph $G=(V,E)$, where V is the set of nodes, and $E \subset V \times V$ is the set of communication links between a pair of nodes. Given that one node can communicate with another node, a link must exist between them. Therefore, the distance $d(i, j)$ between nodes i and j is defined as the minimum number of links from one node to the other. If $d(i, j)$ =1, nodes i and j are neighbors. If $d(i, j)$ =2, i and j are two-hop neighbors. In the network, a node may interfere with its neighbors and two-hop neighbors. Thus, these nodes should not simultaneously transmit. The collision set of a node contains its neighbors and two-hop neighbors.

In industrial wireless mesh networks, all nodes maintain their collision set and routing table. The routing table indicates each next-hop node for the propagation of the packet toward its destination. According to the routing information, each packet from its source node flows to the gateway in several hops. Each hop in this process needs to occupy one slot. Thus, the aim of TDMA scheduling is to determine the set of slots such that no collisions will occur and some optimization criteria will be satisfied. Since the centralized management mode is adopted, the network manager masters the information of the network. Thus, the network manager conducts the slot allocation scheme and then sends information back to each node.

3.2 Optimization Objectives

In industrial wireless mesh networks, the scheduling scheme should meet the high demand of reliable and real-time communication, especially for industrial applications. In addition, data transmission should be energy-efficient. Thus, this work aims to minimize the number of time slots, to reduce energy consumption of the network, and to shorten end-to-end delay of the packets.

3.2.1 Total Number of Time Slots

The total number of time slots indicates the time required to finish all transmission tasks in a sampling cycle, which should be as small as possible to increase the sampling rate of the end nodes. In other words, the fewer slots are consumed, the more data can be collected in each sampling cycle.

3.2.2 Energy Consumption

In industrial wireless mesh networks, nodes can operate in four different modes: transmit, receive, idle, or sleep. Each mode is characterized by a different amount of power consumption [17]. Power consumption in the active mode (transmit or receive) comprises the transmission signal power and the circuit power consumption of all signal processing blocks. In the idle mode, the radio is not communicating but the radio circuitry remains turned on. Thus, power consumption in the idle mode is less than that in the active mode. The sleep mode, which switches the oscillator and voltage regulator off, provides the lowest current draw among all modes. However, the sleep mode involves the highest energy cost and the longest latency for switching the radio back to active mode [18]. Moreover, energy can be wasted by the node even when it operates in idle mode for a long time or when the buffer overflows at intermediate routing nodes in a multihop route [19]. According to Ref. [17-18], the energy consumption of the network is:

$$E_n = \sum_{i=1}^{n} [P_i^{tx}(t_i^{tx} + t_i^{st-tx}) + P_i^{rx}(t_i^{rx} + t_i^{st-rx}) + P_i^{idle} t_i^{idle}] \tag{1}$$

where n denotes the number of end nodes in the network; $P_i^{tx/rx}$ is the power consumption of the transmitter/receiver at node i; P_i^{idle} is the power consumption when node i stays in the idle mode; $t_i^{tx/rx}$ is actual data transmission/reception time at node i; $t_i^{st-tx/rx}$ is the total transition time consumed between the inactive and active modes; and t_i^{idle} is the time during which node i stays in the idle mode in a sampling cycle. For most sensors, power consumption in sleep mode is two orders of magnitude smaller than that in the other three modes. Thus, in this paper, the power consumption of the node in sleep mode is omitted.

3.2.3 End-to-end delay

End-to-end delay refers to the time taken by a packet to travel from its source to its destination. Thus, delay reduction can enhance real-time performance.

The delay over hop j of packet i is defined as the sum of three delay components. The first component is queuing delay, $D_{ij}^{queuing}$, that is the time during which packet i waits at the sending node of this hop, starting from its arrival until the outgoing link that carries it is scheduled to transmit. The second component is switching delay, D_{ij}^{switch}, which is the start-up time of the transceiver in the sending node, equal to the transition time from the idle or sleep mode to the transmit mode. The last component is transmission delay, D_{ij}^{trans}, which is the time from the transmission of packet *i* along the outgoing link until it arrives at the receiving node of this hop [20]. Combining these three aspects, the value of the delay over each hop is determined by the slot lengths assigned to each link as well as by the order of slots in which the links are scheduled.

The total end-to-end delay of all packets in the network is given by D_p, where *p* denotes the amount of packets transmitted from the end nodes to the gateway in the network, and h_i is the number of hops taken by packet *i*.

$$D_p = \sum_{i=1}^{p} \sum_{j=1}^{h_i} (D_{ij}^{queuing} + D_{ij}^{switch} + D_{ij}^{trans}) \tag{2}$$

To minimize the total number of time slots, as many non-conflicting transmissions as possible should be assigned to the same slot. To reduce energy consumption of the network, the most number of nodes should work in continuous slots. Thus, energy consumed by frequent state switching and long idle time is reduced. To shorten the total end-to-end delay of all packets in the network, as many packets as possible should be transmitted in continuous slots to minimize the queuing delay and switching delay.

3.2.4 Objective Function

By using the weighted-sum approach, a multi-objective optimization problem can be converted into a mono-objective problem. Consequently, the weighted sum of the above three objectives was taken as the objective function, as shown by the following:

$$\min F = \alpha * N_{slots} + \beta * E_n + \gamma * D_p \quad (\alpha + \beta + \gamma = 1, \text{and } \alpha \geq 0, \beta \geq 0, \gamma \geq 0) \tag{3}$$

where N_{slots} is the total number of time slots under the schedule; E_n is the total energy consumption of all nodes in the network that can be calculated by Eq. (1); D_p is the sum of the end-to-end delay of all packets in the network corresponding to Eq. (2); and α, β and γ are the trade-off factors among these three objectives. The values of these objectives must be in the same order of magnitude when trade-off is expected.

For fast sampling rate, α must be adjusted to the highest possible value so that the most number of non-conflicting transmissions could be assigned in the same slot. A higher β value results in lower energy consumption because the energy consumed by state switching and long idle time is reduced. However, higher end-to-end delay results from simultaneous long-time queuing. For lower end-to-end delay, the value of γ is adjusted as high as possible so that numerous packets would be transmitted with shorter queuing delay and switching delay. In this case, however, energy consumption increases with the frequent state switching of each node. In addition, given numerous conflicts between the nodes in a network, strict requirements of delay or energy consumption can affect the sampling rate. Thus, the values of α, β and γ need to be adjusted to meet different requirements.

4. Energy-efficient and Low-delay TDMA Scheduling Algorithm

As described in the previous chapter, the TDMA scheduling problem is transformed into a multi-objective optimization problem. The intelligence algorithm has been successfully applied in numerous scheduling and multi-objective optimization problems, such as the flow shop and the traveling salesman problems, among others. Therefore, a hybrid algorithm combining GA with SA is proposed to solve the TDMA scheduling problem in industrial wireless mesh networks.

4.1 Operation-based Encoding and Decoding

To use the proposed hybrid algorithm, a TDMA scheduling solution should be expressed as a sequence of code. Several encoding schemes have been studied in the job-shop scheduling problem [21]. An operation-based encoding method is suitable for this algorithm.

In industrial wireless mesh networks, numerous data packets are transferred to the gateway in multiple hops. Each data collection that a packet flows to the gateway from its source node is called a task. Each hop in this process is a subtask, which is denoted as (TaskID, NodeID). TaskID indicates which task the subtask belongs to, and NodeID identifies which node executes the subtask. For the jth packet generated by node i, the TaskID is equal to $[n*(j-1)+i]$, where n is the number of end nodes in the network.

The random combination of TaskIDs of all subtasks forms an individual, where the ith appearance of one TaskID indicates the ith hop of the corresponding packet. In this way, a TDMA scheduling solution is encoded as an individual. The length of the individual denotes the number of subtasks. The array of the TaskIDs expresses the operating sequence of the subtasks. In this encoding method, the information of nodes and packets is not included in individuals, making it easier to execute crossover and mutation operations, as well as saving computation time [22].

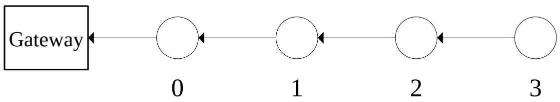


Fig. 2. One simple network model

To explain the given encoding approach, an example is shown in **Fig. 2**. In this simple network, each node generates two packets, such that it has two tasks. All the subtasks are shown below, and a possible individual can be 01122245566633337777.

Node 0: (0, 0); (4, 0);
Node 1: (1, 1), (1, 0); (5, 1), (5, 0);
Node 2: (2, 2), (2, 1), (2, 0); (6, 2), (6, 1), (6, 0);
Node 3: (3, 3), (3, 2), (3, 1), (3, 0); (7, 3), (7, 2), (7, 1), (7, 0).

In the proposed algorithm, an individual can be mapped to a TDMA scheduling solution by using the decoding method. Decoding an individual includes two steps. First, the individual is transformed into a sequence of subtasks. Second, slots are allocated for these subtasks by sequence. The rule is to assign as many non-conflicting subtasks as possible to one slot. For instance, if an individual from the above network is 3103133, the corresponding sequence of subtasks is (3, 3) (1, 1) (0, 0) (3, 2) (1, 0) (3, 1) (3, 0), and the slot allocation scheme is shown in **Table 1**.

Table 1. Slot allocation of the example

Slot	0	1	2	3	4	5
Subtask	(3,3), (0,0)	(1,1)	(3,2)	(1,0)	(3,1)	(3,0)

4.2 Priority Queuing

In GA, initial individuals are typically randomly generated. However, in this study, some rules are set to generate initial individuals to improve the efficiency inspired by multiprocessor scheduling [23].

(1) Tasks with a large number of subtasks are preferentially placed at the beginning of the individual to be scheduled in the front slots, thus reducing the total frame length. Total frame length will be at least as large as the total number of current time slots plus the number of subtasks of the added tasks. As shown in **Fig. 2**, packets generated by node 3 are given priority to be placed ahead because of their large number of hops.

(2) Subtasks that have numerous conflicts with others are placed before the subtasks with fewer conflicts as much as possible, because these subtasks have higher chances of being in a future collision set list. However, the nodes in **Fig. 2** have a similar number of neighbors, thus this rule is neglected.

4.3 Genetic Algorithm

GA is a stochastic optimization method developed by Holland [24]. It simulates natural evolution, in terms of survival of the fittest, by adopting biological operators to get a faster convergence rate. GA starts with a randomly initialized population of individuals. During each iteration step, a new population is formed by applying selection, crossover, and mutation operators to solutions in the current population based on their goodness.

The roulette-wheel selection approach is chosen as the selection operation in the proposed algorithm. A random number is generated, and the individual whose segment spans the random number is selected. The process is repeated until the desired number of individuals is obtained. The probability of the selection of an individual depends on the individual's relative fitness. Individuals with higher fitness have a greater probability of survival.

Crossover is used to recombine genetic materials in two parent individuals to produce two child individuals that share the characteristics of parents. In this paper, two-point crossover is applied at a rate of P_c, as shown in **Fig. 3**. Crossover rate refers to the number of chances in which the individuals in the population can undergo crossover. In this process, some illegal individuals have TaskIDs that are not equal to the setting value. Therefore, the number of these TaskIDs needs to be adjusted back to track.

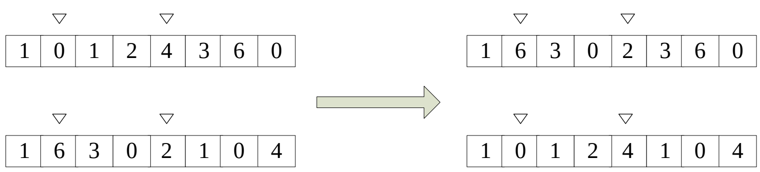


Fig. 3. Two-point crossover

To avoid illegal individuals, swap mutation at a rate of P_m is adopted in this work. With the increase in the length of individuals, the number of mutation positions in an individual increases. Swap mutation is shown in **Fig. 4**.

Fig. 4. Swap mutation

4.4 Simulated Annealing

SA was introduced by Kirkpatrick et al. [25]. This algorithm resembles the cooling process of molten metals through annealing. SA is a point-by-point method which avoids being restricted to a local minimum by accepting cost-increasing neighbors with some probability.

In SA, an initial solution is randomly generated. A neighbor of this solution is then generated by a suitable mechanism, and the change in the evaluation value of the neighbor ΔE is calculated. The law of thermodynamics is directly used in SA, which indicates that a reduction in the evaluation value is obtained, the current solution is replaced by the generated neighbor, whereas if the evaluation value of the neighbor increases, the worse solution is accepted only if $P(\Delta E) > \text{random}(0,1)$. The definition of $P(\Delta E)$ is given by:

$$P(\Delta E) = \exp(-\Delta E / T) \tag{4}$$

where T is a control parameter that corresponds to the temperature in the physical annealing process. A certain number of neighbors are conducted at each temperature, after which the temperature is decreased. This process is repeated until the system freezes into a steady state. This algorithm yields a near optimal solution. The search process of SA is controlled by a cooling schedule, which consists of the following parameters and principles:

(1) Starting temperature T_s: A sufficiently high T_s can make every possible solution searchable at equivalent possibilities. It has a significant influence on computational time as well. In industrial wireless mesh networks, T_s depends on the network conditions.

$$T_s = \Delta E_{\max} / (\ln P_a^{-1}) \tag{5}$$

where $\Delta E_{\max}$ is the maximum difference between the reciprocal of the fitness value of the best parent individual and that of each individual in one generation. In this paper, because $\Delta E_{\max}$ differs depending on the optimization objective, it is set according to specific circumstances. P_a denotes the accepted probability of a new worse solution, it is 0.3 in this study.

(2) Temperature decrement: The decrement of temperature to the stopping criterion is critical to the success of the algorithm. One method to decrement the temperature is by using a simple linear method as given below:

$$T_{i+1} = \mu T_i \qquad (0 < \mu < 1) \tag{6}$$

where i is the ith circle and μ represents the descending rate. The bigger μ is, the slower the search process will be, and vice versa. Ideally, μ should be between 0.8 and 0.99, with better results found at the higher end of the range.

(3) Final temperature T_f: Allowing the temperature to reach zero is unnecessary because as it approaches zero, the chance of accepting a worse solution becomes very small.

4.5 Critical Techniques

To better satisfy the requirements of industrial applications, a number of critical techniques are adopted.

4.5.1 Sleep and Idle Scheme

As previously mentioned, aside from the active mode, a node can either be idle or asleep. Although energy consumption is relatively low under these two modes, frequent state switching consumes large amounts of energy. By using the reference on CC2420 [26] given by Chipcon, the current consumption in each mode and the switching energy are shown in **Table 2** and **Table 3**.

Table 2. Current consumption of each mode for CC2420

Mode	Transmit (Tx)	Receive (Rx)	Idle	Sleep
Current consumption	17.4 mA	18.8 mA	426μA	20μA

Table 3. Switching time and energy for CC2420

Switching mode	Switching time (ms)	Switching energy (μJ)
Idle $\rightarrow$ Tx	0.03	0.916
Idle $\rightarrow$ Rx	0.03	0.992
Sleep $\rightarrow$ Tx	1.2	37.5
Sleep $\rightarrow$ Rx	1.2	40.6

Considering the energy consumption in each mode calculated by Eq. (1) as well as the switching energy, if the inactive state of a node lasts for less than six slots, this node should be set as idle. Otherwise, this node should go to sleep. In this way, the frequency of state transitions between the sleep and active states and the time when nodes stay in idle mode are both reduced. Thus, the energy consumed on these two aspects is reduced. This parameter can be changed according to actual requirements in future studies.

4.5.2 Time Synchronization

Accurate time synchronization between nodes is crucial because the basic time slot in industrial wireless mesh networks is only 10 ms. The following are the basic mechanisms for time synchronization:

(1) During initialization, the network manager divides the whole network into regions, and the number of regions depends on network scale and node density. In each region, the node with the highest number of neighbors is selected as the time synchronization source. These time synchronization sources are synchronized with the gateway. When a packet from a time synchronization source is received, the receiving node should adjust its network time.

(2) When a node receives a packet, this node calculates the difference between the arrival time of this packet and the ideal time at which this node believes the communication should have occurred. This difference must be communicated in every acknowledgment (ACK) reply packet sent to the sending node. Each communication measures the alignment of network time among the nodes. Nodes in industrial wireless mesh networks can tolerate small jitter.

(3) Time synchronization can be maintained by using the first two approaches if two nodes exchange data frequently. Otherwise, if the time since the last communication between the nodes is longer than the threshold, one node has to send "keep-alive" messages to the other to ensure time synchronization. The "keep-alive" message is mainly used for network management and for assisting in time synchronization. A "keep-alive" message should be sent no more than once per minute to avoid frequent messages. This method requires very little

overhead.

Time synchronization based on these three mechanisms depends on which node initiated the transmission. Time synchronization sources maintain a consistent local clock with the gateway and provide a time benchmark to their neighbors. In addition, nodes can adjust other nodes' time by using data exchange and the "keep-alive" message mechanisms. Time synchronization is achieved through the association between nodes.

4.5.3 Wakeup Scheme

To avoid large delays attributed to the transition from sleep mode to active mode, as shown in **Table 3**, a wakeup scheme is proposed. The work in [27] introduced the pipelined tone wakeup scheme, which wakes up all neighbors of transmitters ahead of the scheduled time. However, this scheme consumes a large amount of energy and involves control message overhead. In this study, each node knows when it should transmit or receive. Therefore, only the nodes that are involved in the upcoming communication will wake up. In this study, if a node is in sleep mode, it wakes up before the scheduled active slot and enters into idle mode. This node is then ready for transmission with relatively low switching delay at the beginning of the scheduled active slot.

This wakeup scheme synthetically considers three aspects. First, the time spent during each wakeup process of a node is different. Therefore, the period prior to the scheduled wakeup time should be sufficient so that the node can finish the wakeup process before the active slot. Second, the switching time from idle mode to active mode is significantly less than that from sleep mode to active mode. Therefore, the node should stay in idle mode after wake-up and then switch to the active mode at the beginning of the scheduled active slot. Third, the reduction of delay is at the cost of energy consumption, which is attributed to the early termination of sleep mode. Thus, the period between the actual wakeup time and the scheduled wakeup time should not be excessively long. Considering the above factors and by using CC2420 as a guideline, experiments are conducted to determine a balance between energy consumption and delay in the wakeup process. Results show that nodes should wake up 1 ms to 1.5 ms prior to the scheduled wakeup time. By applying this method, the delay attributed to state switching is minimized, and little energy is required.

4.5.4 Buffer Size Constraint

Each node in industrial wireless mesh networks has limited packet buffer capacity. Therefore, the buffer size of each node in the algorithm should be set according to practical application. The total number of packets in the buffer of each node should be less than the maximum buffer size. If the buffer of a node is congested, the node cannot process the packet promptly. Therefore, during scheduling, if the number of packets in one node is close to the maximum buffer size, this node should be preferentially scheduled to transmit. This rule can avert packet loss, reduce energy cost by retransmission, and shorten queuing delay.

4.5.5 Collision-Free Channel

Communicating nodes are assigned to a superframe, slot, and channel offset, thereby forming a communication link between nodes. As channel diversity is supported, each slot can be simultaneously used on multiple channels by different nodes. This process can be achieved by creating links with different channel offsets in the same slot.

In this study, a maximum of 15 channels can be simultaneously used. If more than 15 subtasks are arranged in one slot, collision occurs between the channels. Therefore, no more than 15 subtasks are assigned to one slot, especially in a large-scale network.

4.6 Hybrid Algorithm

The hybrid algorithm GA/SA presented in [28][29][30] is an integration of the GA and SA algorithms. This hybrid algorithm represents an application-independent approach and the resulting search process is highly adaptive. One of the problems of GA is its convergence behavior. Initially, the cost of the solutions improves rapidly, but further improvement becomes very difficult to achieve. Majority of the runtime is spent in the later phase of the process in which only small improvements are obtained after a long time. Although SA can generally obtain improvements in the later phase of the process, it does not converge as fast as GA in the initial phase. Thus, in this study, the hybrid algorithm that unifies GA with SA can combine the benefits of both algorithms and address their shortcomings.

4.6.1 Fitness Function

A fitness function measures the quality of individuals during the evolution process in the proposed algorithm. A fit individual suggests a better solution. In the evolution process, relatively fit individuals reproduce new individuals, and inferior individuals die. In this work, the individual’s fitness depends on the total number of time slots, energy consumption, and end-to-end delay. As a result, the fitness function can be presented as:

$$Fit(i) = \frac{1}{\alpha * N_{slots} + \beta * E_n + \gamma * D_p} \tag{7}$$

where i represents an individual of the hybrid algorithm.

4.6.2 Implementing the Algorithm

After network initialization, the network manager acquires information on network topology and the interference relationship. The operation-based encoding method and the priority queuing rule are used to generate the original population of individuals. In each iteration, individuals are evaluated by using the fitness function. Based on the fitness values of individuals, roulette selection, crossover, and mutation are separately performed. The probabilistic acceptance approach of SA is incorporated into GA to determine whether the new individual should be accepted to replace the worst individual in the population based on the value of ΔE. In this algorithm, ΔE is the difference between the reciprocal of the fitness value of a new individual and that of the best parent individual in one generation. The hybrid algorithm is repeated until the maximum generation or the final temperature is reached. Finally, the best individual can be obtained and then decoded to an optimal TDMA scheduling solution.

The flowchart in **Fig. 5** describes the main steps of the hybrid algorithm GSA.

5. Simulation Results

5.1 Parameter Settings

The performance of the proposed algorithm is evaluated by the simulations on a Pentium 4 desktop computer with 2 GB of RAM in the C++ environment. Industrial wireless mesh networks support the 10 ms time slot and 15 channels for data transmission. The data rate at the physical layer is 250 kbps, and the packet length is 1 kbits. For each packet, the transmission route is determined according to the information in the routing table that is calculated by using the routing algorithm. The current consumption in the sleep mode is ignored and that of the other three modes is set according to the values shown in **Table 2.** Swtiching time and energy between different modes are set according to the values shown in **Table 3.** Running parameters of the proposed algorithm are listed in **Table 4.**

In this paper, the hybrid algorithm GSA is compared in terms of its properties with existing algorithms, namely, Node Based Scheduling Algorithm (NBSA) proposed by Ergen and Varaiya [4], classical GA, Centralized Scheduling Algorithm with Spatial Reuse (CSASR) proposed by Ma et al. [9], and Hybridized GA and PSO Algorithm (HA) proposed by Mao et al. [10]. All the parameters of GA are the same as the GA part of GSA, and the parameters of HA are set as guidelines.

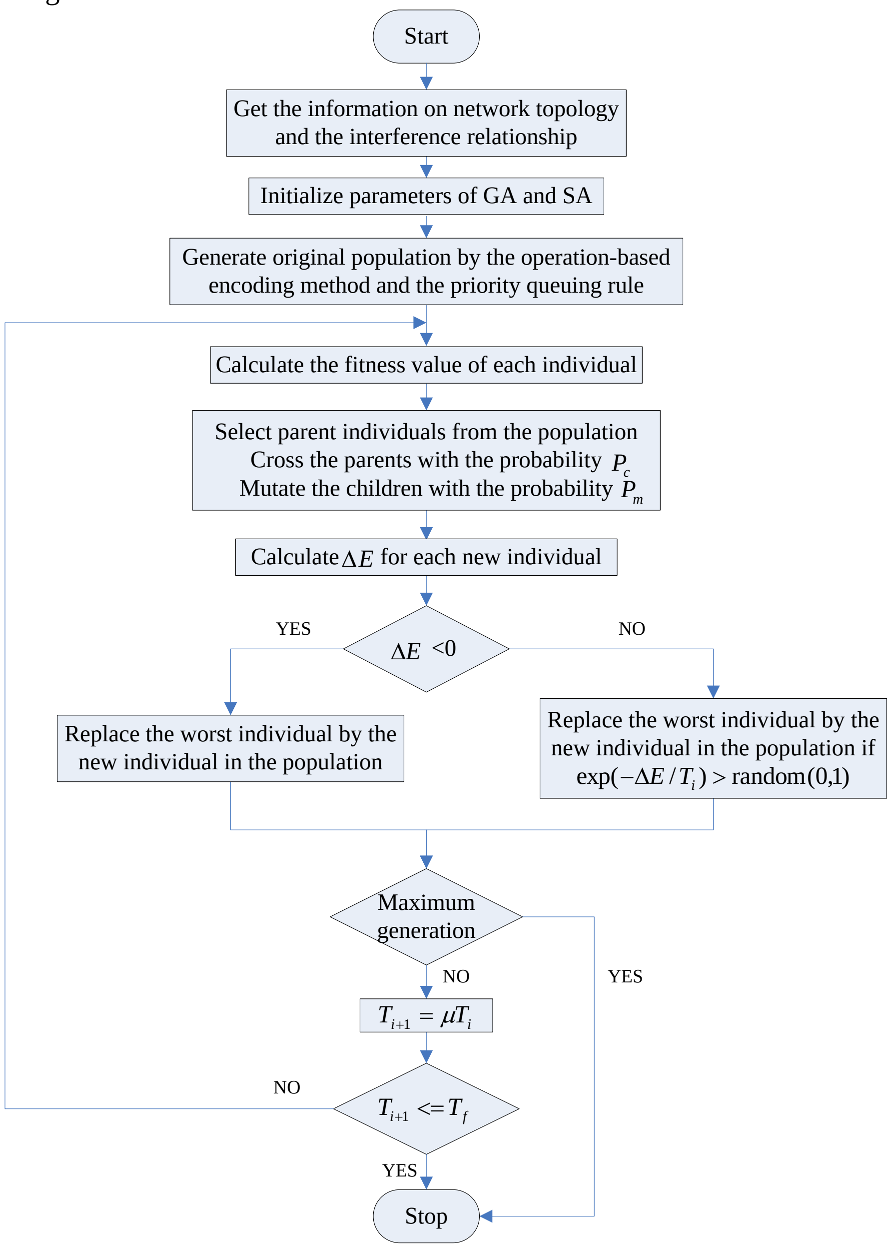


Fig. 5. Flowchart of the hybrid algorithm GSA

The performance metrics were considered similar to the optimization objectives previously mentioned. Metrics include the total number of slots, total energy consumption of the network, and average end-to-end delay of all packets.

In the simulation, three cases considered are shown below. The gateway is placed at the origin for all kinds of networks. The hybrid algorithm was run 10 times for each topology. The average results were then presented.

Case 1: A simple network with seven end nodes and one gateway.

Case 2: Four grid networks, including 5×5, 10×10, 15×15 and 20×20 nodes.

Case 3: Four random mesh networks with nodes numbering 100, 200, 300, and 400.

Table 4. Running parameters of the hybrid algorithm GSA

GA		SA	
Max generation	600	T_s	2~9
Population size	40	T_f	0.001
P_c	0.7	μ	0.96
P_m	0.1		

5.2 Simulation Results of Case 1

This simulation was performed to show slot allocation results in detail and to provide experimental basis for scheduling performance analysis. The network with seven end nodes and one gateway is illustrated in **Fig. 6**, where the ID of each node is enclosed in square bracket. In this case, each node generates one packet to transmit. The packet from the source node flows to the gateway along the shortest path in accordance with the routing table. Therefore, 17 subtasks have to be scheduled.

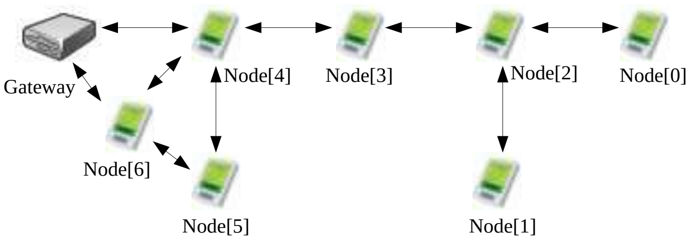


Fig. 6. A simple network

In **Table 5** and **Table 6**, HA shows two different values of *w*, which means that the objective is optimal time performance or optimal energy performance. Three extreme instances $(\alpha=1, \beta=1, \gamma=1)$ of GSA give optimal bound values of the three objectives, respectively. **Table 5** shows that GSA $(\alpha=1)$ and HA $(w=0)$ both gain the best performances on the number of time slots, whereas the performances of NBSA, GA and CSASR are slightly worse. **Table 6** shows the energy consumption and average end-to-end delay under the slot allocation schemes in **Table 5**. NBSA and GA do not consider high energy efficiency and low delay as the desired objectives. HA $(w=1)$ consumes more energy than GSA $(\beta=1)$ because not all nodes work in continuous slots under the slot allocation scheduled using HA. Therefore, achieving optimal results on energy and delay aspects is difficult when using these three algorithms. CSASR gets the same result with GSA ($\beta = 1$) on energy consumption. During scheduling through CSASR, the nodes can only start up twice in one scheduling period if the topology is a tree. Thus, CSASR can reduce energy consumption effectively in this case.

As shown in **Table 6**, GSA can flexibly adjust the sequence of subtasks to acquire the least time slots ($\alpha = 1$), the least energy consumption ($\beta = 1$), and the least average end-to-end delay ($\gamma = 1$). For a smaller number of time slots, as many non-conflict subtasks as possible are assigned to the same slot within the limitation of the channel offset. For lower energy consumption, nodes should work in more continuous slots. The results of GSA ($\beta = 1$) in **Table 5** show that each node works in continuous slots. Finally, for lower delay, the subtasks of one task should work in more continuous slots. **Table 7** shows that the subtasks of each task are operated in continuous slots. Therefore, taking high energy efficiency and low delay into account, GSA can achieve the best scheduling results.

Table 5. Slot allocation of the algorithms

Node / Slot / Algorithm	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
NBSA	2,6	0,4	1,5	3	2,6	4	3	3	2	4	3	4	4	—
GA	1,5	2,6	3	0,4	2	2,6	4	3	3	4	3	4	4	—
HA($w = 0$)	1,4	2,6	2,5	3	0,4	2,6	3	3	3	4	4	4	—	—
HA($w = 1$)	1,4	0,5	2,6	2,6	2	3	3	3	3	4	4	4	4	—
CSASR	0,5	1,6	2,6	2	2	3	3	3	3	4	4	4	4	4
GSA ($\alpha = 1$)	0,4	2,5	3	1,4	2,6	3	4	2,6	3	4	3	4	—	—
GSA ($\beta = 1$)	1,5	0,6	2,6	2	2	3	3	3	3	4	4	4	4	4
GSA ($\gamma = 1$)	1,5	2,6	3	0,4	2,6	3	4	2	3	4	4	3	4	—

Table 6. Performances of the algorithms

Algorithm	**Number of time slots**	**Energy Consumption** (mJ)	**Average end-to-end delay** (ms)
NBSA	13	7.434	41.14
GA	13	7.386	38.28
HA($w = 0$)	12	7.278	41.14
HA($w = 1$)	13	7.044	42.43
CSASR	14	6.966	46.85
GSA ($\alpha = 1$)	**12**	7.261	22.57
GSA ($\beta = 1$)	14	**6.966**	46.85
GSA ($\gamma = 1$)	13	7.244	**18.29**

Table 7. Slot allocation details of GSA ($\gamma = 1$)

Task	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Slot	4,5,6,7	1,2,3,4	8,9,10	12,13	11	1,2	5

5.3 Simulation Results of Case 2

In cases 2 and 3, the amount of packets generated by each node varies from 1 to 5. The fitness function in GA is modified to be as same as that in GSA, and the critical techniques are also adopted in GA. In order to have a clear view of the results, the results of HA with two extreme instances ($w=0, w=1$) and the results of GA with three extreme instances ($\alpha=1, \beta=1, \gamma=1$) were combined, respectively. And only the optimal results on each optimization objective were given.

In this case, four grid networks are simulated, where nodes are placed at integer coordinates in a two-dimensional grid. The performances of the algorithms are detailed in **Table 8.** GSA ($\alpha=1$) and HA achieve similar performances on the total number of time slots. Such performances are better than those of the other algorithms. In terms of energy consumption and average end-to-end delay, GSA ($\beta=1$) and GSA ($\gamma=1$) obtain optimal results, respectively, followed by the other algorithms. This advantage increases in value as the size of networks grows. With the abovementioned critical techniques, GSA can effectively search the solution space and obtain the best results.

Table 8. Performances of the algorithms in grid networks

	Number of time slots				**Energy Consumption** (mJ)				**Average end-to-end delay** (ms)			
	5×5	10×10	15×15	20×20	5×5	10×10	15×15	20×20	5×5	10×10	15×15	20×20
NBSA	143	791	1161	2018	176.4	2196.7	5299.1	10913.8	322.3	2099.8	4304.2	8086.8
GA	138	777	1138	1999	173.3	2113.7	5104.3	10392.9	309.8	2020.6	4198.6	7846.1
HA	136	772	1129	1979	171.0	2107.5	5153.5	10478.2	305.2	2044.9	4230.3	7938.2
CSASR	137	779	1145	2003	167.8	2073.4	5075.3	10292.9	338.6	2087.2	4259.1	8011.7
GSA ($\alpha=1$)	**136**	**772**	**1126**	**1976**	175.6	2111.2	5050.1	10494.9	335.6	2021.2	4187.2	7868.3
GSA ($\beta=1$)	137	780	1137	1995	**167.1**	**2065.6**	**5017.1**	**10196.6**	340.4	2055.9	4223.2	7975.6
GSA ($\gamma=1$)	138	777	1141	2006	173.2	2118.8	5092.3	10431.7	**288.6**	**1976.3**	**4078.5**	**7664.3**

5.4 Simulation Results of Case 3

In this case, 100, 200, 300, and 400 nodes are randomly distributed in a square area with side length of 10 units. Two nodes can communicate with each other if the distance between them is less than or equal to $\sqrt{2}$ units.

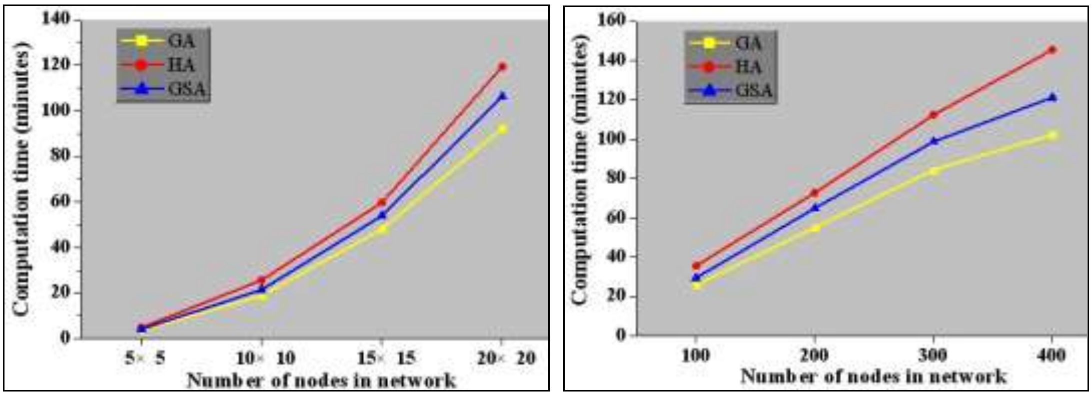
Table 9. Performances of algorithms in random networks

	Number of time slots				Energy Consumption (mJ)				Average end-to-end delay (ms)			
	100	200	300	400	100	200	300	400	100	200	300	400
NBSA	858	1447	2119	2069	2060.4	5537.7	9004.6	10802.1	2451.4	4786.5	7523.4	9468.2
GA	845	1430	2095	2040	1974.7	5446.5	8754.3	10389.3	2304.3	4537.7	7290.2	9176.6
HA	841	1420	2077	2022	1957.5	5402.3	8837.6	10463.3	2327.7	4611.9	7385.3	9295.4
CSASR	850	1432	2100	2062	1938.4	5351.9	8672.8	10236.5	2387.6	4721.8	7452.5	9453.1
GSA ($\alpha=1$)	**841**	**1418**	**2072**	**2018**	1953.4	5463.1	8664.1	10325.9	2393.7	4572.9	7063.9	8948.3
GSA ($\beta=1$)	845	1433	2082	2058	**1920.9**	**5297.8**	**8576.1**	**10090.2**	2300.6	4694.5	7429.2	9406.8
GSA ($\gamma=1$)	843	1425	2106	2046	1977.4	5472.2	8702.3	10464.6	**2210.4**	**4387.8**	**6946.9**	**8796.9**

The performances of the algorithms in random networks are detailed in **Table 9**. GSA ($\alpha=1$) and HA achieve similar performances on the total number of time slots, which is better than the other three algorithms. GSA ($\beta=1$) and GSA ($\gamma=1$) show the best performances in terms of energy consumption and average end-to-end delay, respectively. With the increase in the number of nodes, the advantage of GSA is more evident in random networks. GSA clearly exhibits a better optimization effect than GA because of SA's strong local-search capability. The search capability of GA is weaker than that of HA. However, the critical technologies mentioned above are adopted in GA. Therefore, these two algorithms achieve similar performances in terms of energy consumption and average end-to-end delay. The differences in energy consumption between CSASR and GSA ($\beta=1$) are both small in cases 2 and 3. However, without buffer size constraint, the number of retransmission packets in CSASR increases with the number of transmissions. Therefore, the energy consumption of CSASR will be significantly larger in practical applications. Switching delay is also reduced by using CSASR, while the queuing delay of each packet simultaneously increases.

The performances of GSA with three extreme instances ($\alpha=1, \beta=1, \gamma=1$) in grid networks and random networks are detailed in **Tables 8** and **9**, respectively. Both tables show that the total number of time slots, energy consumption, and average end-to-delay cannot be simultaneously optimized by using GSA. This finding suggests that a trade-off exists among these three objectives. For different optimal objectives, different criteria are used to evaluate individuals, as discussed in Section 3.2.4. In addition, the relationship among these three metrics varies according to network scale and node density. Therefore, the values of α, β and γ need to be adjusted according to actual demands and network conditions.

Fig. 7 illustrates the computation time of the three intelligence algorithms for different network scale. HA spends the largest amount of time because much time was spent in the PSO part. Although GSA has a longer computation time than GA, it achieves better performance. This finding suggests that the performance of GSA is improved at the expense of the computation time.



(a) Grid networks **(b)** Random networks

Fig. 7. Computation time (minutes)

In summary, GSA and HA achieve similar performances on the total number of time slots. Such performances are better than those of the other three algorithms. In terms of average end-to-end delay and energy consumption, GSA obtains optimal results for different network scale. GSA also appears to be more scalable because its performance improves with an increasing number of nodes. Moreover, the trade-off among the time, energy and delay objectives can be easily achieved by a proper weighted factor.

6. Conclusion

In this paper, a hybrid TDMA scheduling algorithm that combines GA and SA is proposed to solve the multi-objective scheduling problem for industrial wireless mesh networks. The desired objectives are to minimize the total number of time slots as well as to achieve high energy efficiency and low delay. In the proposed algorithm, the operation-based encoding method guarantees that the sequence of subtasks can be flexibly adjusted and the priority queuing rule enables the initial population to have better performance, thereby improving the efficiency of subsequent searching. In this manner, the hybrid intelligence algorithm which possesses a good searching capability can determine the optimal slot allocation scheme for each optimization objective. In addition, with critical techniques such as time synchronization and wakeup schemes, the proposed algorithm can provide energy-efficient and low-delay communications, thus satisfying the demands of industrial wireless mesh networks.

However, the search process of the proposed algorithm is time-consuming, especially as the network scale increases. We are presently working to introduce network segmentation into the algorithm. Further investigations on the scheduling solution in case of routing changes are still necessary.

References

[1] HART Communication Foundation, "TDMA Data Link Layer," http://WirelessHART.hartcomm.org/, Apr 2007.

[2] K. Veeramachaneni and L. A. Osadciw, "Optimal Scheduling in Sensor Networks Using Swarm Intelligence," in *Proc. of 38th Annual Conference on Information Sciences and Systems*, pp. 17-19, March 15-19, 2004.

[3] S. Gandham, M. Dawande and R. Prakash, "Link Scheduling in Wireless Sensor Networks: Distributed Edge-Coloring Revisited," *Journal of Parallel and Distributed Computing*, vol. 68, no.

8, pp. 1122–1134, Aug 2008. Article (CrossRef Link)
[4] S. C. Ergen and P. Varaiya, "TDMA Scheduling Algorithms for Wireless Sensor Networks," *Wireless Networks*, vol. 16, no. 4, pp. 985-997, May 2010. Article (CrossRef Link)
[5] G. Y. Pei and C. Chien, "Low Power TDMA in Large Wireless Sensor Networks," in *Proc. of IEEE Military Communication Conference,* vol. 1, pp. 347-351, Oct 2001. Article (CrossRef Link)
[6] G. Jolly and M. Younis, "An Energy-Efficient, Scalable and Collision-Free MAC Layer Protocol for Wireless Sensor Networks," *Wireless Communications and Mobile Computing,* vol. 5, no. 3, pp. 285-304, May 2005. Article (CrossRef Link)
[7] L. Q. Shi and A. O. Fapojuwo, "TDMA Scheduling with Optimized Energy Efficiency and Minimum Delay in Clustered Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 927-940, Jul 2010. Article (CrossRef Link)
[8] S. S. Kulkarni and M. Arumugam, "SS-TDMA: A Self-Stabilizing Medium Access Control (MAC) for Sensor Networks," In *Sensor Network Operations,* S. Phoha, T. F. La Porta and C. Griffin (Eds.), Wiley- IEEE Press, pp. 186-215, Apr 2006. Article (CrossRef Link)
[9] J. C. Ma, W. Lou, Y. W. Wu, X. Y. Li and G. H. Chen, "Energy Efficient TDMA Sleep Scheduling in Wireless Sensor Networks," in *Proc. of 28th IEEE International Conference on Computer Communications,* pp. 630-638, Apr 19-25, 2009. Article (CrossRef Link)
[10] J. L. Mao, Z. M. Wu and X. Wu, "A TDMA Scheduling Scheme for Many-to-One Communications in Wireless Sensor Networks," *Computer Communications*, vol. 30, no. 4, pp. 863-872, Feb 2007. Article (CrossRef Link)
[11] S. M. Islam, S. Ghosh, S. Das, A. Abraham and S. Roy, "A Modified Discrete Differential Evolution based TDMA Scheduling Scheme for Many to One Communications in Wireless Sensor Networks," in *Proc. of 2011 IEEE Congress on Evolutionary Computation*, pp. 1950-1957, Jun 2011. Article (CrossRef Link)
[12] A. Sridharan and B. Krishnamachari, "Max-Min Fair Collision-Free Scheduling for Wireless Sensor Networks," in *Proc. of 2004 IEEE International Conference on Performance, Computing and Communications*, pp. 585-590, Apr 2004. Article (CrossRef Link)
[13] S. G. Cui, R. Madan, A. Goldsmith and S. Lall, "Energy-Delay Tradeoffs for Data Collection in TDMA-based Sensor Networks," in *Proc. of 2005 IEEE International Conference on Communications*, vol. 5, pp. 3278-3284, May 2005. Article (CrossRef Link)
[14] P. Djukic and S. Valaee, "Delay Aware Link Scheduling for Multi-Hop TDMA Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 870-883, Jun 2009. Article (CrossRef Link)
[15] N. A. Pantazis, D. J. Vergados, D. D. Vergados and C. Douligeris, "Energy Efficiency in Wireless Sensor Networks Using Sleep Mode TDMA Scheduling," *Ad Hoc Networks*, vol. 7 no. 2, pp. 322-343, Mar 2009. Article (CrossRef Link)
[16] HART Communication Foundation, "Network Management Specification," http://WirelessHART.hartcomm.org/, Apr 2007.
[17] E. Shih, S. H. Cho and N. Ickes et al., "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," in *Proc. of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 272-287, Jul 2001. Article (CrossRef Link)
[18] R. Jurdak, A. G. Ruzzelli and G. M. P. O'Hare, "Radio Sleep Mode Optimization in Wireless Sensor Networks, " *IEEE Transactions on Mobile Computing,* vol. 9, no. 7, pp. 955-968, Jul 2010. Article (CrossRef Link)
[19] A. N. Kim, F. Hekland, S. Petersen and P. Doyle, "When HART Goes Wireless: Understanding and Implementing the WirelessHART Standard," in *Proc. of IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 899-907, Sep 2008. Article (CrossRef Link)
[20] J. Kim, S. Kim and J. Lee, "Adaptive Duty Cycling MAC Protocols Using Closed-Loop Control for Wireless Sensor Networks," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 1, pp. 105-122, Jan 2011. Article (CrossRef Link)
[21] L. Wang and D. Z. Zheng, "An Effective Hybrid Optimization Strategy for Job-Shop Scheduling Problems," *Computers and Operations Research*, vol. 28, no. 6, pp. 585–596, May 2001. Article

(CrossRef Link)
[22] X. Z. Cao and Z. H. Yang, "An Improved Genetic Algorithm for Dual-Resource Constrained Flexible Job Shop Scheduling," in *Proc. of 2011 International Conference on Intelligent Computation Technology and Automation*, pp. 42-45, Mar 2011. Article (CrossRef Link)
[23] G. Ritchie, "Static Multi-processor Scheduling with Ant Colony Optimization and Local Search," Master of Science thesis, University of Edinburgh, 2003.
[24] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, 1975.
[25] S. Kirkpatrick, C. D. Gelatt Jr. and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4958, pp. 671-680, May 1983. Article (CrossRef Link)
[26] Texas Instruments CC2420 Radio Transceiver, http://focus.ti.com/docs/prod/folders/print/cc2420.html, 2007.
[27] X. Yang and N. H. Vaidya, "A Wakeup Scheme for Sensor Networks: Achieving Balance between Energy Saving and End-to-End Delay," in *Proc. of 10th IEEE Real-Time and Embedded Technology and Applications Symposium,* pp. 19-26, May 2004. Article (CrossRef Link)
[28] S. G. Ponnambalam and M. Mohan Reddy, "A GA-SA Multiobjective Hybrid Search Algorithm for Integrating Lot Sizing and Sequencing in Flow-Line Scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 21, no. 2, pp. 126-137, Jan 2003. Article (CrossRef Link)
[29] K. P. Dahal and N. Chakpitak, "Generator Maintenance Scheduling in Power Systems Using Metaheuristic-based Hybrid Approaches," *Electric Power Systems Research*, vol. 77, no. 7, pp. 771-779, May 2007. Article (CrossRef Link)
[30] A. Norozi, M. K. A. Ariffin, N. Ismail and F. Mustapha, "An Optimization Technique Using Hybrid GA-SA Algorithm for Multi-Objective Scheduling Problem," *Scientific Research and Essays*, Vol. 6, no. 8, pp. 1720-1731, Apr 2011. Article (CrossRef Link)

Yun Zuo is currently working towards her Ph.D. at the School of Information Science and Engineering, East China University of Science and Technology (ECUST), China. Her research interests include the routing and scheduling algorithms in industrial wireless mesh networks.

Zhihao Ling is a professor at the East China University of Science and Technology (ECUST), China. He received his Ph.D. in Control Science and Engineering from ECUST in 2005. He is an executive director of the Process Control and Instrumentation Technical Committee of the China Instrument and Control Society and a director of the Shanghai Association of Automation. His research interests are networked measurement and control system in industrial automation.

Luming Liu is currently working towards his Ph.D. at the School of Information Science and Engineering, East China University of Science and Technology (ECUST), China. His research interests are in the areas of wireless networks in industrial automation.