Applied Mathematics & Information Sciences
An International Journal

A Genetic Algorithm for Data Mule Path Planning in Wireless Sensor Networks

Yung-Liang Lai[1], *and Jehn-Ruey Jiang*[2]*

[1]Department of Computer Science and Information Engineering, Taoyuan Innovation Institute of Technology, Jhongli, Taiwan
[2]Department of Computer Science and Information Engineering, National Central University, Jhongli, Taiwan

Received: 17 Jan. 2012; Revised 21 Jun. 2012; Accepted 23 Jul. 2012
Published online: 1 January 2013

Abstract: A data mule is a mobile device that can traverse a wireless sensor network field to move near stationary sensor nodes that are spatially dispersed for collecting data from them. Use of the data mule can significantly reduce energy consumption of sensor nodes compared to common multihop forwarding schemes. However, it also increases the latency of gathering data of all nodes. In this paper, under the assumptions that the data mule can sequentially move from a specific location to another specific location and that sensor nodes can adjust their radios to different power levels, we study the data mule path planning optimization (DMPPO) problem to achieve two goals under one constraint. The two goals are (1) to plan the path for the data mule to move near every sensor node to collect data so that the data mule traversal time (or latency) is minimized, and (2) to adjust the sensor nodes transmission ranges so that the total sensor node energy consumption is minimized. The constraint is that the data mule must move near each sensor node at least once for gathering data. The DMPPO problem is a multi-objective optimization problem; it is challenging since a sensor node can shrink its transmission range to reduce energy consumption but the range shrinking requires the data mule to move more for data gathering, which incurs longer latency. We propose a genetic algorithm using heuristics to find Pareto optimal solutions to this problem. We also simulate the proposed algorithm to show its effectiveness.

Keywords: Data mule, multi-objective optimization, Pareto optimal solution, wireless sensor network, genetic algorithm

1. Introduction

A wireless sensor network (WSN) consists of many small-sized sensor nodes equipped with microcontrollers, wireless radios, and analog/digital sensors. The sensor nodes can monitor physical environmental phenomena, such as electromagnetic magnitude, humidity, temperature, pressure, and so on, to provide a variety of services, such as military surveillance [1] or environmental monitor [2]. The sensed data should be delivered to a specific node, called the sink node, which is connected to the outside world. If all sensor nodes and the sink node are connected, then the data can be delivered to the sink node in a multihop manner (i.e., through the path of a chain of several intermediate nodes forwarding the data). If the sensor nodes are sparse and not connected, a data mule, which is a mobile device with abundant energy and storage space, can traverse the WSN to move to the locations near the sensor nodes for collecting data and dropping off the data to the sink node. There are some examples of data mules, such as the robot in underwater environmental monitoring [3] and the UAV (unmanned aerial vehicle) in structural health monitoring [4].

Using the data mule to collect data can significantly reduce energy consumption of sensor nodes, which are usually powered by batteries with limited energy. This is because sensor nodes only have to transmit its own sensed data over a short range. However, the use of the data mule also increases the latency of gathering data of all nodes. Thus, data mule based data collection systems have to consider both the data collection latency of the data mule and the energy consumption of sensor nodes.

The sensor node radio power level setting can influence the power consumption of sensor nodes. It can also influence the data mule path planning, which in turn affects the data collection latency. For example, in the scenario of Fig. 1, in which three sensor nodes S_1, S_2, and S_3 are deployed, if the three sensor nodes assume identi-

* Corresponding author: e-mail: jrjiang@csie.ncu.edu.tw

cal basic radio power level setting as shown in Fig. 1(a), the data mule has to visit locations A and B to collect data from the three nodes, and the data collection latency is the elapse time for the data mule to move from location A to location B. However, if the sensor nodes assume the conservative radio power setting shown in Fig. 1(b), the data mule has to visit locations A, B and C to gather data of all three nodes, and the data collection latency is the elapse time for the data mule to move from location A, to location B and then to location C.

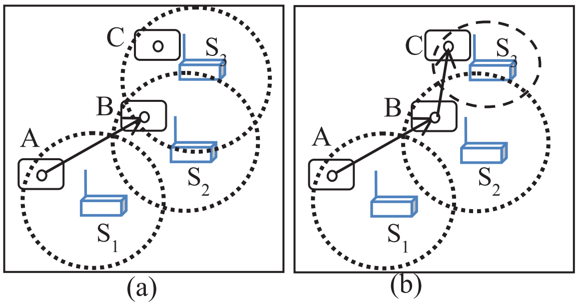

Figure 1 The illustration of the power level setting affecting the data mule path planning.

Many studies [6-11] investigated the data mule path planning for gathering data in WSNs. To the best of our knowledge, no study addresses data mule path planning under the consideration that the sensor node radio power level is adjustable to have a variety of transmission ranges. In this paper, under the assumptions that the data mule can sequentially and freely move from a specific location to another specific location and that the sensor nodes can adjust their transmission ranges to different levels, we study the data mule path planning optimization (DMPPO) problem to achieve two objectives: (1) to plan the path for the data mule to move near every sensor node to collect data so that the data mule traversal time (or latency) is minimized, and (2) to adjust the sensor nodes transmission rages so that the total sensor node energy consumption is minimized.

The DMPPO problem is a multi-objective optimization problem. It is challenging since a sensor node can shrink its transmission range to reduce energy consumption but the range shrinking requires the data mule to travel more for data gathering, which incurs longer latency. We propose a genetic algorithm to solve this problem with different priority settings of multiple optimization objectives. The NSGA-II (Nondominating Sorting GA-II) algorithm [5] is a famous heuristic algorithm in multi-objective optimization, which provides a general framework for finding the Pareto optimal solutions to a multi-objective problem. Based on the framework of the NSGA-II algorithm, we design a heuristic algorithm to find solutions to the DMPPO problem. We also simulate the proposed algorithm to demonstrate its effectiveness.

The rest of the paper is organized as follows. In section 2, we introduce the related work. In section 3, we formally define the DMPPO problem. Section 4 presents the Data Mule Planning Algorithm to solve the problem, and section 5 shows its simulation results. Finally, section 6 concludes this paper.

2. RELATED WORK

Using the data mule (or mobile data collector) is considered an efficient approach for data collection in WSNs. Shah et al. [6] proposed a three-tier architecture having mobile entities called Mobile Ubiquitous LAN Extensions (MULEs) for data collection in the WSN. Path selection of data mule is one of the most important topics in data mule based data collection. Zhao and Ammar [7] studied the problem about how to using data mule (called a message ferry in [7]) to mediate communications between sparsely deployed stationary nodes in the wireless networks. They formulated the planning problem based on the well-known Traversal Salesman Problems and proposed a planning algorithm to solve it. Somasundara et al. [8] studied the problem of choosing the path of a data mule that traverses through a sensor field with sensors generating data at a given rate. They designed a heuristic algorithm to plan the traversal path of data mule. Sugihara and Gupta [9] studied the speed control problem for the data mule to download data from the sensor nodes when the data mule is within sensors communication range. A heuristic algorithm is proposed to plan the schedule for downloading the data from the sensors.

Some studies investigated the case of multiple data mules for WSN data collection. Zhao et al. [10] extended their message ferry framework proposed in [7] to solve the problem of multiple data mules. Jea et al. [11] studied the case in which multiple data mules move on fixed paths. They proposed an algorithm to find the balanced assignment of sensors to different data mules so that the total tour time of mules is minimized.

3. DATA MULE PATH PLANNING OPTIMIZATION PROBLEM

We formulate the data mule path planning optimization (DMPPO) problem in this section. It is a multi-objective optimization problem with two goals (or objectives) and one constraint. The first goal is to minimize the latency of traversal path of data mule. The second goal is to minimize the total energy consumption of sensor nodes. The constraint is that the data mule must move near each sensor node at least once for gathering data.

Data Mule Path Planning Optimization (DMPPO) Problem

Given: LS: A location set of feasible stop locations of the data mule. **SS**: A sensor set of m sensor nodes. **PS**: A power-level set of radio power levels.

Decision variables: TPATH: Traversal Path, i.e., a list $[L_1, L_2, \ldots, L_n]$ of n stop locations of the data mule, where $[L_1, L_2, \ldots, L_n] \in$ LS, **PLASS**: Power Level Assignment, i.e., a list $[P_1, P_2, \ldots, P_m]$ of power levels assigned to each of the m sensors

Goal G1: Minimizing TD(TPATH)

$$TD(TPATH) = \sum_{i=1}^{n} D(L_i, L_{i+1}) \tag{1}$$

Goal G2: Minimizing TE(PLASS)

$$TE(PLASS) = \sum_{j=1}^{m} P(P_j) \tag{2}$$

Subject to Constraint C1:

$$\forall S_j \in SS, \exists L_i \in LS : D(S_j, L_i) \leq Range(P_j) \tag{3}$$

In the DMPPO problem, the sensor set (SS), the power level set (PS) and the locations set (LS) are given, where SS is the set of all sensor nodes, PS is the set of possible radio power levels assigned to a sensor node, and LS is the set of locations of feasible data mule stops in the sensor field. We want to find a data mule traversal path (TPATH) and a power level assignment (PLASS), a list of power levels assigned to each sensor node, to reach Goal G1 and Goal G2 under Constraint C1. Goal 1 is to minimize TD(TPATH), the total traversal distance of the data mule going along traversal path TPATH. The function TD is defined in Eq. (1), where $D(L_i, L_{i+1})$ returns the distance between locations Li and L_{i+1}. Goal 2 is to minimize TE(PLASS), the total energy consumption of all the sensors under the power level assignment PLASS. The function TE is defined in Eq. (2), where P(Pj) returns the power consumption of a sensor node using Pj radio power level. Constraint C1 is defined in Eq. (3), which states that for each sensor node Sj in SS, there exists one data mule stop with location Li such that $D(S_j, L_i) \leq Range(P_j)$, where $D(S_j, L_i)$ stands for the distance between the location of sensor S_j and the data mule stop location Li, and $Range(P_j)$ stands for the communication range of sensor S_j assigned to use radio power level P_j.

4. DATA MULE PATH PLANNING ALGORITHM

In this section, we present our data mule path planning algorithm. The algorithm is a heuristic algorithm based on the genetic algorithm. We first present the chromosome representation, crossover operation, and the fitness function. Then, we present the proposed algorithm.

4.1. Chromosome Representation

We use the chromosome of genes to represent one feasible solution, which contains both the traversal path (TPATH) and the power level assignment (PLASS), to the DMPPO problem.

4.1.1. Representing the traversal path (TPATH)

The traversal path TPATH is represented as a list (i.e., vector) of integers or a chromosome of genes, in which each integer or gene represents the identity of a data mule stop. The value range of the identity is 0, 1, 2, ..., ST_{MAX}, where STMAX is the maximum identity of the stops. It is notable that we use 0 to represent the stop near the sink node, where the data mule starts and ends the traversal path. Thus, a feasible chromosome must contain identity 0. For example, the following chromosome is feasible.

TPATH= [1, 3, 2, 0]

The above chromosome indicates that the traversal path of a data mule starts at the sink node, visits stops 1, 3, 2, and then returns to the sink node. It is notable that the identity 0 may appear before the last position in the vector. For example, for TPATH=[1, 3, 0, 2], it means the data mule does not visit stop 2 but directly returns to the sink node after visiting stop 3, which implies the traversal path is shorter.

4.1.2. Representing the power level assignment (PLASS)

The power level assignment PLASS is represented as a list (i.e., vector) of integers or a chromosome of genes, in which each integer or gene represents a sensor radio power level. The value range of the level is 1, 2, ..., PL_{MAX}, where PL_{MAX} is the largest power level. The following example of the PLASS chromosome shows the power level assignment of 5 nodes with 3 levels (1=normal, 2=enhanced, and 3=maximal).

PLASS = [1, 2, 3, 1, 3]

In the example, the sensor nodes 1, 2, 3, 4 and 5 set their radio power levels to be normal, enhanced, maximal, normal, and maximal, respectively.

4.2. Crossover and Mutation

We use crossover and mutation operations in the proposed algorithm. In our design, we have two types of chromosomes: Traversal Path (TPATH), and Power Level Assignment (PLASS) chromosomes. PLASS is of the simple chromosome type, so the typical crossover operator can handle it. In contrast, TPATH is of the permutation chromosome type, which has to keep the restrictions of permutation after the crossover operation is performed. In our design, we adopt the Order Crossover (OX) operator proposed by Davis [12]. The OX operator generates the new offspring from two parents while keeping the permutation type. The OX operator selects two cut points of the first parent chromosome, and then copies the genes between the two cut points to the new offspring. The remaining genes of the new offspring are ordered according to their order in the second parent. Table. 1 illustrates the OX operator. The cut points starts at the second position and ends at fourth position. Thus, the genes 3, 2, and 4 of the first parent are copied to the new offspring. The remaining genes, which are 1 and 5, of the new offspring are ordered according to their order in the second parent.

Table 1 Illustration of the Order Crossover (OX) operator

	Bits				
Parent 1	5	**3**	**2**	**4**	1
Parent 2	2	<u>1</u>	3	<u>5</u>	4
New Offspring	<u>1</u>	**3**	**2**	**4**	<u>5</u>

The mutation operator for the PLASS and the TPATH chromosomes are described as follows. The mutation operator of the PLASS chromosome is to randomly change the value of one gene to produce an offspring chromosome. The mutation operator of the TPATH chromosome is to randomly select two locations to exchange their values to produce an offspring chromosome.

4.3. Fitness Function

The fitness function is used to evaluate the goodness of individual offspring in the aspects of the specified objectives. In our design, we have two objectives, so we have two fitness functions. The first is for evaluating the total distance (TD) of the data mule traversal path. The second is for evaluating the total energy (TE) consumed by all the sensor nodes.

4.3.1. Fitness function of distance (FD)

The fitness function FD is used to evaluate the goodness of the chromosome in terms of the total distance of the data mule traversal path. If the traversal path violates the constraint C1 of the DMPPO problem, the fitness is set to be a negative infinity value λ to exclude this chromosome from the solutions set. If the traversal path does not violate the constraint C1, the fitness will be set as the total distance of the traversal path TPATH. The fitness function FD is given in Eq. (4), where the total distance (TD) of the traversal path TPATH is defined in Eq. (1).

$$FD(TPATH) = \begin{cases} TD(TPATH) & \text{if } Eq.\ (3)\ is\ Satisfied, \\ -\lambda & \text{if } Eq.\ (3)\ is\ not\ Satisfied \end{cases} \quad (4)$$

4.3.2. Fitness function of energy consumption (FE)

The fitness function FE is used to evaluate the goodness of the chromosome in terms of total energy consumptions of sensor nodes with the power level assignment PLASS. The fitness function FE is given in Eq. (5), where the total energy (TE) consumed by all sensor nodes with the power level assignment PLASS is defined in Eq. (2).

$$FE(PLASS) = TE(PLASS) \quad (5)$$

4.4. Data Mule Path Planning Algorithm

The proposed algorithm is an evolutionary genetic algorithm based on the (Nondominating Sorting GA) NSGA-II algorithm [5] for finding good path planning solutions to the DMPPO problem. The basic idea of the proposed algorithm is to find the Pareto Front, which is the set of nondominated feasible solutions that are not dominated by any others. It is noted that a solution x is said to dominate another solution y if and only if x is better than y in at least one evaluation of objectives and x is not worse than y in all evaluations of objectives.

The proposed genetic algorithm runs generation by generation. In each evolutionary generation, we want to find a front set $F = F_1, F_2, \ldots, F_r$ generated from the previous generation, where r is the maximum number of fronts. As shown in the example in Fig. 3, there are three fronts (F_1, F_2, and F_3) on the two dimensional space, where the two dimensions are two objective functions OF1 and OF2. The optimization goals in the DMPPO problem are to minimize the OF1 (i.e. total distance of path: TD) and OF2 (i.e. total energy consumption: TF), so a front closer to the origin point is with a higher rank. Each solution in the front F_i is not dominated by any solution in F_w, for all $w > i$. Since the populations are generated from the parents with the best finesses of the previous generation, the goodness of populations will be improved after some generations. In this way, when the algorithm is terminated, F_1 will be the set of optimized solutions in our problem. In the algorithm, we select Λ solutions to construct the front set F in each generation. In addition, we keep the diversity of solutions in the evolutionary by the crowding distance proposed in [5]. The crowding distance of a solution can be measured by the distance of the two closet neighbors of the solution, which is used to keep the solutions widely spread. Please refer to the paper [5] for the definition details. The pseudo code of the proposed algorithm is shown below. Initially, the generation counter t is 0 and the population P_t is randomly generated, where a member in Pt is an individual

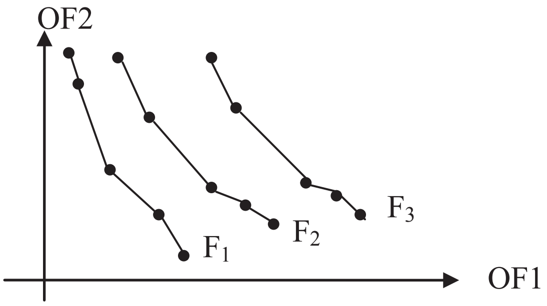

Figure 2 Illustration of a front set F, where $F = F_1, F_2, F_3$. Each point represents one feasible solution in one front in the 2-dimensional space. F_1 is the most efficient front, which is the result of the Data Mule Path Planning Algorithm. The ranks of the three fronts ordered from highest to lowest are F_1, F_2, and then F_3.

(or a solution) consisting of the traversal path chromosome and the power level assignment chromosome. An offspring population Q_t is set as empty initially.

Algorithm 1: Data Mule Path Planning Algorithm

Input: $StopSet = [ST_1, ST_2, \ldots, ST_{MAX}]$,
$PowerlevelSet = [PL_1, PL_2, \ldots, PL_{MAX}]$

Output: Pareto Front Set

Initialization:
$t = 0; P_t = randompopulation; Q_t = \phi;$

Main Loop:
A1: $R_t = P_t \bigcup Q_t$;

A2: F = Nondominated_Fronts_Sort(R_t, FD, FE), where $F = \{F_1, F_2, \ldots, F_r\}$ is the front set, FD is the fitness function for the first objective, and FE is the fitness function of the second objective

A3: $P_{t+1} = \phi$; i=1

A4: **while** $(| P_{t+1} | + | F_i | < \Lambda)$ **do**
$P_{t+1} = P_{t+1} \bigcup F_i; \quad i++;$
end

A5: Crowding_Distance_Sort(F_i)

A6: Insert the first ($\Lambda - | P_{t+1} |$) elements in the Sorted F_i into P_{t+1}

A7: Create offspring population Q_{t+1} from P_{t+1}

A8: **if** $t < MAX\_T$ **then**
$t = t + 1$; Goto A1;
end
else
Return F
end

As illustration in the Data Mule Path Planning Algorithm, in step A1, we set R_t to be $P_t \bigcup Q_t$. In step A2, the algorithm evokes the Nondominated_Fronts_Sort(R_t, FD, FE) function to sort solutions according to their nondomination ranks. The nondomination rank of a solution is the number of solutions dominated by the solution in terms of the fitness functions FD and FE, which are defined in Eq. (4) and Eq. (5), respectively. The result of the function is stored in Pareto Front F={ $F_1, F_2, \ldots, F_r$ }.

The step A3 is to set the population P_{t+1} to be empty and set the counter i to be 1 before the algorithm enters the loop in step A4. The step A4 is to insert the nondominated solutions into P_{t+1}. The step A5 is to generate a sorted F_i by the crowding distance in the descending order. The step A6 is to insert the most widely spread $(\Lambda - | P_{t+1} |)$ solutions using the crowding distance values in the sorted F_i into the P_{t+1}. Thus, the solution included in the population P_{t+1}. For the example shown in Fig. 3, F_3 is the solutions with higher crowding distances from F_3, where the size of F_3 is $(\Lambda - | P_{t+1} | = \Lambda - | F_1 \bigcup F_2 |)$.

The step A7 is to create new offspring population Q_{t+1} from P_{t+1} by mutation and crossover operations, where the size of Q_{t+1} is Λ. In step A8, the algorithm checks whether the maximum generation is reached. If the generation counter t is less than the maximum value (MAX_T), then t is increased by 1 and then algorithm goes to step A1; otherwise, the algorithm terminates.

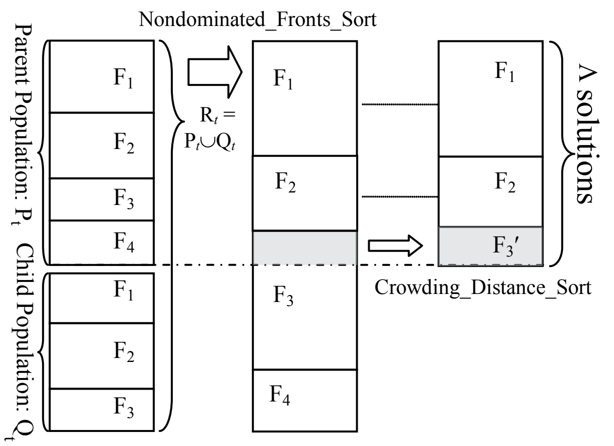

Figure 3 Procedures of generating new population P_{t+1} from P_t and Q_t

5. SIMULATIONS

In this section, we evaluate our proposed heuristic planning algorithm by implementing our algorithm on the Matlab platform [13]. The test case has six data mule stop locations and six sensor nodes with two power levels (i.e., the normal power level, and the enhanced power level). The locations are randomly assigned in the 7 units x 7 units square box.

The simulation setting is as follows. In energy consumption, the normal power level costs 1 Joule unit, and the enhanced power level costs 2 Joule units. The transmission radius of the normal power level is one unit, and the transmission radius of the enhanced level is 2 units. As to the latency, we assume the data mule spends one second to move one unit of distance. The value of Λ is set to be 100 units and the simulation is terminated after 1000 generations.

In Fig. 4, we plot the simulation results of the Data Mule Path Planning Algorithm, where each plotted point represents one solution in terms of the objectives of the latency and the energy consumption. As shown in Fig. 4, the plotted points form the Pareto Front. Moreover, we observe the latency is decreased when the energy consumptions of wireless sensors is increased.

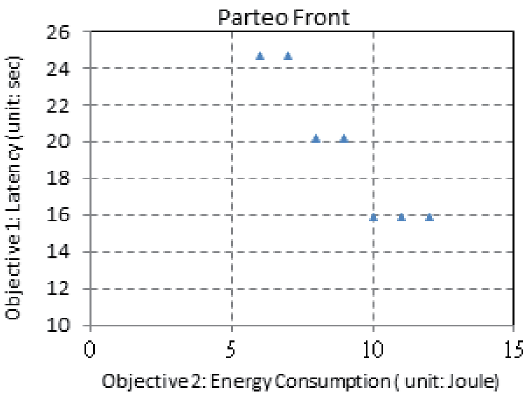

Figure 4 Simulation results of Data Mule Path Planning Algorithm

6. Conclusion

It is promising to use the data mule for gathering data in wireless sensor networks. To reduce the energy consumption of sensor nodes and reduce the latency of data gathering, we have formulated a data mule path planning optimization (DMPPO) problem for data mule based data gathering. The problem is a multi-objective optimization problem for planning the data mule path to achieve the shortest traversal time, and for adjusting the radio power level to achieve the lowest total sensor node energy consumption. We have proposed a heuristic genetic algorithm for solving the DMPPO problem. We have also performed simulations for the proposed algorithm, and the simulation results demonstrate that the proposed algorithm is effective.

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] J. Yick, B. Mukherjee, D. Ghosal, Analysis of a Prediction-based Mobility Adaptive Tracking Algorithm, Proceedings of the IEEE Second International Conference on Broadband Networks (BROADNETS). 753-760 (2012).

[2] M., Mascarenas, D., Flynn, E., Rosing, T., Lee, B., Musiani, D., Dasgupta, S., G. Wener-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, M. Walsh, Deploying a wireless sensor network on an active volcano, IEEE Internet Computing. **10**,18-25 (2006).

[3] I.Vasilescu, , K. Kotay, D.Rus, M.Dunbabin, , and Corke, P. I, Data collection storage and retrieval with an underwater sensor network, Proceedings of the 3rd international conference on Embedded networked sensor systems, 154-165. (2005).

[4] S. Kpotufe, D. Hsu, R. Gupta, G., Park, T. Overly, M. Nothnagel, and C. Farrar, A different approach to sensor networking for SHM: Remote powering and interrogation with unmanned aerial vehicles, Proceedings of the 6th International workshop on Structural Health Monitoring. (2007).

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation. **6**, 182-197. (2002).

[6] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette, Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks, Ad Hoc Networks. 215-233. (2003).

[7] W. Zhao, and M. Ammar,Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networksm, Proceedings of the IEEE Workshop on Future Trends in Distributed Computing Systems. 398-314. (2003).

[8] A.Somasundara, A. Ramamoorthy, A., and Srivastava, M. B, Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines, Proceedings of the 25th IEEE international Real-Time Systems Symposium. 296-305 (2004).

[9] R. Sugihara, and R. Gupta, Optimal speed control of mobile node for data collection in sensor networks, IEEE Transactions on Mobile Computing. **9**, 127-139. (2010).

[10] W. Zhao, M.Ammar, and E.Zegura, Controlling the mobility of multiple data transport ferries in a delay-tolerant network, Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. 1407-1418. (2005).

[11] D. Jea, A.Somasundara, and M. Srivastava, Multiple controlled mobile elements (data mules) for data collection in sensor networks, Distributed Computing in Sensor Systems. 244-257. (2005).

[12] L. Davis, Applying adaptive algorithms to epistatic domains, Proceedings of the 9th International Joint Conference on Artificial Intelligence, 162-164 (1985).

[13] Matlab (2012), http:///www.mathworks.com/.

Yung-Liang Lai received his BS Degree in Applied Mathematics from Chinese-Culture University, Taiwan, in 1996, and his MS Degree in Information Management from National Yun-Lin Science Technology University in 1998. He received the PhD Degree in the Department of Computer Science and Information Engineering, National Central University in 2011. His current research interests include mobile computing, wireless networks, wireless sensor networks, and optimization theory.

Jehn-Ruey Jiang received his Ph. D. degree in Computer Science in 1995 from National Tsing-Hua University, Taiwan, R.O.C. He joined Chung-Yuan Christian University as an Associate Professor in 1995. He joined Hsuan-Chuang University in 1998 and became a full Professor in 2004. He is currently with the Department of Computer Science and Information Engineering, National Central University, and co-leads the Adaptive Computing and Networking (ACN) Laboratory, which aims at developing adaptive mechanisms for collaborative computing entities to make proper adjustments according to their current understandings about the computing environments or resources, in order to efficiently perform given tasks. He was Guest Editors of International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC) and Journal of Information Science and Engineering (JISE). His research interests include distributed computing, pervasive computing, peer-to-peer computing, mobile ad hoc networks, and wireless sensor networks.