Flying path optimization in UAV-assisted IoT sensor networks

Sang-Jo Yoo*, Jae-hyun Park, Su-hee Kim, Anish Shrestha

Department of Information and Communication Engineering, Inha University, Incheon, Republic of Korea

Received 28 July 2016; accepted 14 August 2016
Available online 24 August 2016

Abstract

In this paper, we present an optimal flying path for unmanned aerial vehicle-assisted internet of things sensor networks using a location aware multi-layer information map considering different utility functions based on the sensor density, energy consumption, flight time, and flying risk level. The overall weighted sum of multi-objective utility functions is maximized using the genetic algorithm. The simulation results verify that the optimum solution points can be obtained by adjusting the weights while satisfying the required constraints.

Keywords: Unmanned aerial vehicle; Optimum flying path; Utility function; Genetic algorithm

1. Introduction

A vision to connect almost everything has led to the concept of a new paradigm referred to as the internet of things (IoT), wherein entire surroundings like the environment, buildings, and landscapes can be closely monitored by sensors. IoT sensor networks can be applied in both civilian and military operations, such as search and destroy/rescue, intelligent transportation, wildfire monitoring, disaster management, border security, and many more. The IoT sensor network requires a large number of sensors deployed over a huge space, including remote areas and areas that are inaccessible to humans. Under such circumstances, collecting sensor data becomes a challenging task. Unmanned aerial vehicles (UAVs) have emerged as a pragmatic solution to such problems.

UAVs are a means to gather sensing data very quickly in a cost efficient way [1]. It complements existing techniques, fitting between satellite and terrestrial approaches. However, collecting data from each sensor in an IoT-based sensor network is difficult as it generates a tremendous volume of sensing data and requires UAVs to follow a long route, resulting in high energy consumption, longer delay, and exposure to hazardous environment. Therefore, we focus on the study of flying path optimization in UAV-assisted IoT sensor networks.

We use location dependent multi-layer information (MLI) map in this study and design multiple utility functions based on the sensor density, energy, flight time, and risk to find the optimum route of the UAV. The overall weighted sum of the multi-objective utility functions is maximized using the genetic algorithm (GA) under the constraints of predefined maximum delay and energy along with the prohibited area.

The remainder of this paper is organized as follows. In Section 2, we present a detailed explanation of the proposed scheme. The performance evaluation is presented in Section 3. Finally, the conclusions are drawn in Section 4.

2. Proposed scheme

2.1. System model

We consider $J$ types of sensors, each with different sensing values, deployed in a random or deterministic fashion within a region $(R)$, which is divided into $N$ small unit cells $(C_i)$ such that $i = 1, 2, \ldots, N$. The sensor density of each cell is also considered different. The cells are categorized as prohibited cells, flying cells, and sensing information gathering (SIG) cells. The UAV is not allowed to enter the prohibited cells $(C^p)$. It stays in the SIG cells $(C^s)$ to gather data and flies above the flying cells $(C^f)$ to follow the defined path.
A complete flying path consists of a set each of SIG cells and flying cells. The set of SIG cells is represented as
\[ S_{\text{sense}} = \{ C_1^s, C_2^s, \ldots, C_{N_s}^s \}, \] (1)
where \( C_i^s \) is the \( i \)th SIG cell and \( N_s \) is the number of SIG cells. Similarly, the set of flying cells comprises several subsets of flying cells and is represented as
\[ S_{fly} = \{ S_1^f, S_2^f, \ldots, S_{N_f}^f, S_{N_f+1}^f \}, \] (2)
where \( S_k^f \) is the set of flying cells that are positioned between the \((k-1)\)th SIG cell to the \( k \)th SIG cell, which can be expressed as
\[ S_k^f = \{ C_1^f(k), C_2^f(k), \ldots, C_{N_f(k)}^f(k) \}. \] (3)
The cardinality of set \( S_k^f \) is denoted by \( N_f(k) \) cells and \( C_1^f(k) = C_{k-1}^s, C_{N_f(k)}^f(k) = C_k^s, C_1^f(1) \) is the starting cell.

### 2.2. Energy model

Energy is mainly spent on three activities, namely communication, flying, and sensing. We refer to the energy required for transmitting and receiving sensing information as the communication energy. The communication energy can differ depending on the type of sensor. For simplicity, we assume that all types of sensors consume the same energy, which is denoted by \( e^c \). On the other hand, the flying energy is location-dependent. We denote the flying energy at velocity \( v \) for unit time as \( e^f \), and the flying energy weight for cell \( i \) is denoted by \( e_i^f \). The flying energy weight for each cell can differ based on the terrain of the cell. Therefore, the flying energy from cells \( i \) to \( j \), which is denoted by \( e_i^f \), should be expressed based on their position. It can be expressed using Eq. (4) if the adjacent cells are located horizontally or vertically, and using Eq. (5) if the adjacent cells are located diagonally as given below
\[
e_i^f = \left( \frac{1}{2} w_i^f + \frac{1}{2} w_j^f \right) e^f \frac{d_{ij}}{v}, \] (4)
\[
e_i^{f,j} = \left( \frac{1}{2} w_i^f + \frac{1}{2} w_j^f \right) e^f \sqrt{2d_e}, \] (5)
where \( d_e \) is the distance between two cells. The UAV has to stay over the cells for a certain period in order to gather sensing information from several sensors within an SIG cell. The energy consumed in doing so is called the staying energy. The staying energy in cell \( i \) can be obtained as
\[
e_i^s = e^s t_s N_i^p \sum_{j=1}^{J} n_i^j, \] (6)
where \( e^s \) is the staying energy for a unit time in a cell, \( n_i^j \) is the number of type \( j \) sensor nodes in cell \( i \), \( t_s \) is the sensing packet exchange time, and \( N_i^p \) is the number of transmissions required to transfer a sensing packet in cell \( i \) successfully. Moreover, packet error probability also differs in each cell because each cell has different wireless channel conditions. Hence, the required number of packet transmissions for a successful packet transmission can be written as
\[ N_i^p = \frac{1}{1 - p_i^e}, \] (7)
where \( p_i^e \) is the packet error probability in cell \( i \). In addition, the flying risk during a given time unit for each cell \( i \) is defined as \( r_i \). We assume that all the required information is available in the form of a multi-layer information map.

### 2.3. Flying cell route

The flying cell route between consecutive SIG cells is determined in terms of the shortest path using “A star” algorithm. The heuristic function used to evaluate the shortest path is determined as
\[ f(i) = g(i) + h(i) \] (8)
where \( g(i) \) is the cost of the path from the starting cell to an arbitrary cell \( i \), and \( h(i) \) is the estimated cost from cell \( i \) to the destination. The A star algorithm consists of two arrays called the closed list and open list arrays. The starting cell is saved in the parent cell and its neighbors, except the prohibited ones, are saved in the open list. The heuristic function value of each cell in the open list is calculated using Eq. (8). The least value is saved in the closed list array. The selected cell is declared as the parent cell and its neighboring cells, except those in the closed list and the prohibited ones, are added to the open list. We can check which route is better and choose the best route if the open list already has any of the neighbors. The same parent cell is used if the value of \( g(i) \) of the previous cell is smaller; otherwise, the parent cell is changed. The process is continued until we arrive at the destination. Finally, the route is traced using the parent cell and the shortest path is selected.

### 2.4. Utility functions

In this section, we define several utility functions to formulate the weighted sum of the multi-objective utility function. We consider four main parameters to design the utility functions, namely sensing, energy, time, and risk, which are defined below.

1. **Sensing Utility**

The sensing utility in a cell \( i \) depends on the number of sensor nodes and its corresponding information value, which can be written as
\[ U_i^S = \sum_{j=1}^{J} n_i^j v^j, \] (9)
where \( n_i^j \) is the number of type \( j \) sensor nodes in cell \( i \) and \( v^j \) is the information value of sensor type \( j \). The overall sensing utility function can be expressed as
\[ U^S = \sum_{i=1}^{N_S} U_i^S, \] (10)
where \( N_S \) is the total number of SIG cells.
(2) Energy Utility

As explained in the previous section, energy is consumed in three different tasks. Therefore, we define the corresponding utility functions for each task as follows:

a. Flying energy utility

The flying energy utility from the \((k-1)\)th SIG cell to the \(k\)th SIG cell can be expressed as shown below.

\[
U_{k}^{FE} = \sum_{i=1, \forall C_{i}^{f}(k) \neq S_{k}}^{N_{f}(k)-1} e_{i,i+1}^{f}(k),
\]

where \(N_{f}(k)\) is the number of cells from the \((k-1)\)th SIG cell to the \(k\)th SIG cell, and \(e_{i,i+1}^{f}(k)\) is the flying energy from \(C_{i}^{f}(k)\) cell to \(C_{i+1}^{f}(k)\) cell of set \(S_{k}\). The overall flying utility of the route can be written as

\[
U^{FE} = -\sum_{k=1}^{N_{f}} U_{k}^{FE}.
\] (12)

b. Staying energy utility

The staying energy utility is the energy that the UA V requires to stay stable while gathering the sensing data at a given cell. It can be obtained as

\[
U^{SE} = -\sum_{i=1}^{N_{s}} U_{i}^{SE},
\] (13)

where \(U_{i}^{SE}\) is the staying energy utility function at the \(i\)th SIG cell, which is the same as (6).

c. Communication energy utility

The communication energy utility at the \(i\)th SIG cell is given by

\[
U_{i}^{CE} = e^{f} t_{c} N_{c}^{f} \sum_{j=1}^{J} n_{j}^{f}.
\] (14)

The overall communication energy utility can be obtained as

\[
U^{CE} = -\sum_{i=1}^{N_{s}} U_{i}^{CE}.
\] (15)

Finally, the total energy utility can be obtained by adding Eqs. (12), (13) and (15) as

\[
U^{E} = U^{FE} + U^{SE} + U^{CE}.
\] (16)

(3) Time Utility

The total time spent by the UAV depends on the flying and staying times. Accordingly, we design two separate utility functions, which are defined below.

a. Flying time utility

The flying time utility from one cell to another can be expressed as

\[
U_{k}^{FT} = \sum_{i=1}^{N_{f}(k)-1} t_{i,i+1}^{f}(k),
\] (17)

where \(t_{i,i+1}^{f}(k)\) is the flying time from \(C_{i}^{f}(k)\) to \(C_{i+1}^{f}(k)\) cell of set \(S_{k}\). It should be noted that the value of \(t_{i,i+1}^{f}(k)\) differs depending on the location of the cell \((i+1)\). Therefore, there will be two cases. We refer to the cell \((i+1)\) as case 1 if it is located horizontally or vertically, whereas if the cell \((i+1)\) is located diagonally, we refer to it as case 2. Then, the flying time can be obtained as

\[
t_{i,j}^{f}(k) = \begin{cases} \frac{d_{c}}{v}, & \text{case } 1 \\ \frac{\sqrt{2d_{c}}}{v}, & \text{case } 2. \end{cases}
\] (18)

The overall flying time utility can be obtained as

\[
U^{FT} = -\sum_{k=1}^{N_{f}+1} U_{k}^{FT}.
\] (19)

b. Staying time utility

The staying time utility in the \(i\)th SIG cell can be expressed as

\[
U^{ST} = t_{c} N_{c}^{f} \sum_{j=1}^{J} n_{j}^{f}.
\] (20)

The overall staying time utility can be expressed as

\[
U^{ST} = -\sum_{i=1}^{N_{s}} U_{i}^{ST}.
\] (21)

Finally, the overall time utility can be obtained by adding (19) and (21) as

\[
U^{T} = U^{FT} + U^{ST}.
\] (22)

(4) Risk Utility

Risk utility accounts for probable exposure to hazards and the uncertainties associated with a UAV in flight. The risk utility depends on the flying and staying cell utility as discussed below.

a. Flying cell risk utility

The flying cell risk utility from one cell to the next is given by

\[
U_{k}^{FR} = \sum_{i=1}^{N_{f}(k)-1} t_{i,i+1}^{f}(k) \left( \frac{1}{2} r_{i}^{f}(k) + \frac{1}{2} r_{i+1}^{f}(k) \right),
\] (23)

where \(r_{i}^{f}(k)\) is the risk of \(C_{i}^{f}(k)\) cell of set \(S_{k}\). The overall flying cell risk utility is given by

\[
U^{FR} = -\sum_{k=1}^{N_{f}+1} U_{k}^{FR}.
\] (24)

b. Staying cell risk utility

The staying cell risk utility in the \(i\)th SIG cell is given by

\[
U_{i}^{SR} = r_{i}^{f} t_{c} N_{c}^{f} \sum_{j=1}^{J} n_{j}^{f}.
\] (25)

The overall staying cell risk utility is given by

\[
U^{SR} = -\sum_{i=1}^{N_{s}} U_{i}^{SR}.
\] (26)

The overall risk utility can be obtained by adding (21) and (26) as

\[
U^{R} = U^{FR} + U^{SR}.
\] (27)

(5) Total Utility

The total utility function can be finally obtained by the weighted sum of each utility function as

\[
F = \omega_{s} U^{s} + (1 - \omega_{s})(\omega_{E} U^{E} + \omega_{T} U^{T} + \omega_{R} U^{R}),
\] (28)

where \(\omega_{s}, \omega_{E}, \omega_{T},\) and \(\omega_{R}\) are the utility weights for sensing, energy, time, and risk, respectively, such that, \(0 < \omega_{s} \leq 1, \omega_{E} + \omega_{T} + \omega_{R} = 1.\)
2.5. Genetic algorithm

The flowchart in Fig. 1 depicts the implementation of the GA to maximize the multi-objective function formulated in Eq. (28) for an optimal UAV flying path [2–4]. We consider three major constraints. First, the total end-to-end flight time should not exceed the threshold time limit $T_{\text{limit}}$. Second, the total energy consumption of the UAV should not be greater than the maximum energy limit $E_{\text{limit}}$. Finally, the UAV should not fly above the prohibited cells.

Initially, a population of candidate solutions in the form of chromosomes is initialized. Each chromosome consists of a combination of sensing cells and flying cells as shown in Fig. 2. Flying cells are selected in such a way that they connect two corresponding SIG cells with the shortest path. The fitness of each chromosome in the population is evaluated using Eq. (28) based on the selected flying and SIG cells in the chromosome. Some of the best chromosomes are selected as an intermediate population using roulette wheel selection scheme. ‘Order 1 crossover’ is applied to ‘cell id’ with probability $P_c$ for each consecutive pair. The ‘swap mutation’ operation is performed by flipping the SIG indication value of each offspring with probability $P_m$. A fraction of the previous population with lower fitness values are replaced by new random solutions. The process is repeated until the stopping criterion is not satisfied.

3. Performance evaluations

We evaluate the set of utility values using the fitness function of the GA. When it satisfies the stop conditions, we select the route path that has the maximum utility value. The scheme is implemented using MATLAB. The variables used in the simulation are given in Table 1. Fig. 3 presents the total utility for different values of $\omega_s$ with $\omega_e = 0.3$, $\omega_r = 0.4$, and $\omega_t = 0.3$. We can observe that the total utility increases with the number of iterations and saturates after a certain number of iterations. Increasing the weight of the sensing utility increases the total utility but the UAV has to visit more SIG cells, which requires longer time and more energy.

Fig. 4 shows the individual utility for $\omega_s = 0.5$ and 0.7 and $\omega_e = 0.3$, $\omega_r = 0.4$, and $\omega_t = 0.3$. It is observed that the sensing utility improves with increase in $\omega_s$ as expected. However, an increase in $\omega_s$ reduces the overall weight i.e., $(1 - \omega_s)$, assigned to the rest of the utility functions. An increase in $\omega_e$ reduces the energy, time, and risk utility, which increases the cost. It is possible to improve a particular individual utility by adjusting its corresponding weights.

4. Conclusion

We proposed a novel scheme to optimize the flying path using GA in UAV-assisted IoT sensor networks. In our proposed method, each cell of the sensor field is characterized according to its environmental characteristics such as the prohibited area,
channel condition, sensor deployment statistics, and risk level. A multi-objective utility function was formulated to derive the optimum path in terms of flying and SIG cells. Simulation results showed that our proposed method could derive the desired optimum path while satisfying the constraints.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (NRF-2014R1A2A2A01002013). This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-H8501-16-1019) supervised by the IITP (Institute for Information & communications Technology Promotion).

References