Scalarized Q Multi-Objective Reinforcement Learning for Area Coverage Control and Light Control Implementation

Akkhachai Phuphanin[1], Non-member and **Wipawee Usaha**[2], Member

ABSTRACT

Coverage control is crucial for the deployment of wireless sensor networks (WSNs). However, most coverage control schemes are based on single objective optimization such as coverage area only, which do not consider other contradicting objectives such as energy consumption, the number of working nodes, wasteful overlapping areas. This paper proposes a Multi-Objective Optimization (MOO) coverage control called Scalarized Q Multi-Objective Reinforcement Learning (SQMORL). The two objectives are to maximize area coverage and to minimize the overlapping area to reduce energy consumption. Performance evaluation is conducted for both simulation and multi-agent lighting control testbed experiments. Simulation results show that SQMORL can obtain more efficient area coverage with fewer working nodes than other existing schemes. The hardware testbed results show that SQMORL algorithm can find the optimal policy with good accuracy from the repeated runs.

Keywords: Multi objective reinforcement learning, Area coverage control, Light control.

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of small computing devices with limited computational capabilities and energy supply. Various systems have developed and implemented such devices for a wide range of applications, such as automatic systems, environmental monitoring systems, elderly people monitoring systems and smart homes. These systems rely on the interaction and cooperation between the sensor nodes to carry out the operation. There are several key performance indicators to measure the performance of a WSN, such as coverage area, energy consumption, the number of working nodes, the coverage area per working node, or coverage area per unit energy consumed, etc. These metrics are vital in measuring the quality-of-service (QoS) of the network.

Manuscript received on April 3, 2018 ; revised on June 29, 2018.

The authors are with School of Telecommunication Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand, E-mail : a.phuphanin@gmail.com[1], wipawee@sut.ac.th[2]

Coverage control has gained much research interest in wireless sensor networks. Typically, coverage control aims to maintain or maximize coverage while preserving network lifetime and energy consumption of the WSN. Due to the multiple parameters which can affect coverage, coverage control problems are typically formulated as optimization problems with single or multiple objectives. Ref. [1] proposed a single objective coverage control, the optimal geographical density control (OGDC) scheme was proposed for guaranteed full coverage control. The scheme is based on grid redundancy check and sequential node activation. The grid redundancy requires that each sensor node maintains a list of the grid points it covers. Each active node sends out activation messages to neighboring nodes to reset their timers. Thus, the OGDC scheme is a deterministic optimization scheme which aims to maximize a single objective (i.e. maximize the coverage area).

Single objective coverage control has also been applied to enable energy efficient usage and convenience in smart homes or smart buildings. Ref. [2]-[6] apply coverage control for lighting control applications. Similar to [1], [2]-[6] are also based on a single objective which is to maintain coverage (light intensity) required by user. These works do not consider energy consumption which is an important parameter in WSNs.

Most aforementioned literature focus on applying wireless sensor networks to manage a single objective of maximizing coverage [1] or satisfying light intensity requirements [2]-[6]. However, such single objective optimization schemes may well conflict with other objectives such as minimizing energy consumption, or wasteful overlapping areas. In certain applications such as in visible light communication (VLC), overlapping coverage areas is undesirable as the identification data cannot be read in the light overlapping areas. On the other hand, the principles of multi-objective optimization (MOO) can support multiple objectives and be used to determine solutions [7]. Multiple objectives have been considered in many wireless sensor network applications. In [8], a multiple target tracking sensor management algorithm was investigated. The problem was to select subsets of sensors and assign frequency and minimize transmission power to working sensors in order to maximize the tracking performance of multiple targets. As for

coverage control application, MOO is generally used for optimizing contradicting objectives, for example, coverage maximization, minimization of working sensor nodes, minimization the unbalanced energy consumption and minimization of energy consumption to prolong the network lifetime [9], [10]. Ref. [11] presented an intelligent lighting control which considered two objectives of maintaining light intensity for users and minimizing energy expenditure. Each round of decision is dependent on light intensity information gathered from the immediate indoor and outdoor environment of the occupants. It can be seen that these aforementioned works in [7]-[11] rely on rule-based or threshold-based decisions. Such works may not adapt well in constantly changing surroundings such as lighting control with changing external lights. Furthermore, these algorithms have centralized operations which may be suitable for small scale coverage. However, such schemes may not be suitable for implementation in individual sensor nodes for distributed coverage control.

On the other hand, there are several researches in the existing literature which applied adaptive learning methods to the MOO framework. Such learning methods can predict the optimal policy from learning experience in presence of a constantly changing environment. Ref. [12] presents an approach in which Pareto dominance is incorporated into particle swarm optimization (PSO) in order to allow this heuristic to handle problems with several objective functions. Ref. [13] applied the multi-objective genetic algorithm (GA), called Energy-efficient Coverage Control Algorithm (ECCA), to the coverage control problem in WSNs. The objective of ECCA is to minimize the number of working sensor nodes while maximizing the coverage area. Another common algorithm to solve MOO problems is the Artificial Neural Network (ANN). Ref. [14] used ANN to find the optimal transmission path with the objective to minimize the global energy consumption. Ref. [15] also applied ANN to select randomly placed sensors to maximize a barrier coverage and minimize energy consumption. Although PSO, GA and ANN approaches can solve MOO problems, such algorithms are typically complex and slow in finding the optimal policy [10]. Another method used to solve MOO problems is reinforcement learning (RL). RL is a learning scheme which is based on the actual interaction between an agent and the environment. Upon each action decided by the agent when the environment is in a particular state, a reward is returned and the agent uses the reward to iteratively improve its action. One common RL tool is Q-learning which is an algorithm that an agent updates iteratively to improve its actions based the goodness of state-action pair function. The RL approach can be easily implemented in a distributed architecture like in WSNs. Ref. [16] proposed a service-wise protocol optimization technique for multi-objective, co-located and complex heterogeneous network. Their proposed solution efïňĄciently combines MOO with the reinforcement learning (RL) method by using linear function approximation to reduce the dimensionality of the problem. However, [16] combines individual rewards for each objective into a single reward, thereby transforming the MOO into a single objective optimization problem. If the objectives are conflicting, transforming the MOO into a single objective may not provide the best solution. Ref. [17] applied a multi-agent Q-learning scheme to an energy efficient coverage control problem in WSNs. Their results show that the multi-agent learning scheme is scalable and can outperform non-learning coverage control schemes despite its low complexity. However, their multi-agent RL scheme is based on a single cost function of multiple conflicting objectives. Such a single combined objective function may not attain the best policy particularly if each objective is contradicting.

Therefore, this work is focused on the application of a MOO framework to an *online learning* scheme with *separate* objective functions for *coverage control* in WSNs. In particular, the Scalarized Q Multi-Objective Reinforcement Learning (SQ-MORL) method, which uses a MOO framework is applied to the coverage control problem in WSNs. Such online learning scheme is also adaptive to changes and perturbations. The algorithm has low complexity and is distributed. Therefore, it provides a promising implementation in sensor nodes which are resource constrained. Furthermore, this work also implements a hardware testbed to evaluate the performance of SQ-MORL in a multi-agent lighting control experiment.

The contribution of this work is therefore threefold: 1) The SQMORL coverage control and performance evaluation by means of simulation in uniform random and grid sensor layout; 2) Comparison of SQ-MORL with both learning and non-learning coverage control schemes in WSNs; 3) Development of a testbed and hardware performance evaluation of an automatic lighting control using SQMORL algorithm.

2. MULTI-OBJECTIVE REINFORCEMENT LEARNING FOR COVERAGE CONTROL

Coverage control is a critical issue in WSNs as it is one of the parameters which affects the QoS of the network. However, an increase in area coverage may have influence on other resources such as more energy consumption as well. Furthermore, if the network has a large number of sensor nodes placed in the network, excessive number of working sensor nodes in nearby areas may result in overlapping areas which is a waste of energy. Therefore, multi-objective reinforcement learning can be applied to find the optimum policy to select sensor nodes to maintain its coverage while simultaneously reducing overlapping area and hence

energy consumption.

2.1 Multi-Objective Optimization

In general, multi-objective optimization (MOO) problems include a number of objectives required for optimization simultaneously. Each objective may be related or conflicting. Therefore, the main function of MOO is to find the equilibrium points of different objectives. A multi-objective optimization problem (MOP) with n variables and m objectives ($m > 1$) can be formulated as [18]:

$$\min or \max g(x) = [g_1(x), g_2(x), ..., g_m(x)] \quad (1)$$

subject to $x \in \Omega$, where $\Omega \subset R^n$ is the decision space, $g \subset G$ and $G : \Omega \to R^m$ consist of m real valued objective functions, and R^m is the objective space.

2.2 Scalarized Q Multi Objective Reinforcement Learning for Area Coverage Control

Multi-objective reinforcement learning (MORL) problems differ from general RL problem in that there are multiple objectives to be achieved by the learning agent. Each objective has its own reward or penalty signals. This is a basic architecture which a single agent is simultaneously faced with a set of different objectives $o = 1, 2, ..., M$. Suppose an agent can be in a state $s \in S$ where $S = \{s_0, s_1, ..., s_N\}$ is a set of distinct states. The agent can take an action $a \in A = \{a_0, a_1, ..., a_K\}$ at a particular state. Upon taking an action, the agent changes into the next state $s' \in S$ with transition probability $P(s'|s, a)$, and a reward $r(s, a, o)$ associated to objective $o = 1, 2, ..., M$, is returned for taking that action at that state. Let $Q(s, a, o)$ be a Q-value (or action value), which is a numerical value for every state-action pair associated to objective o, that determines how good the action is at that particular state. Typically, $Q(s, a, o)$ satisfies the Bellman equation given by,

$$Q(s, a, o) = r(s, a, o) + \gamma \sum_{\forall s'} P(s'|s, a) \max_{a'} Q(s', a', o)$$

where $r(s, a, o)$ is the immediate reward associated to objective o for taking action $a \in A$ at state $s \in S$, the latter term is the expected future reward, and $\gamma \in [0, 1]$ is the discount factor which ensures that the reward for any state action pair decreases over time. The goal for RL is to find a policy π, which is the probability of selecting the optimal action at a given state, such that the action value is optimized. Let the vector MQ^π be defined by

$$MQ^\pi(s, a) = [Q_1^\pi Q_2^\pi ... Q_M^\pi]^T \quad (2)$$

where MQ^π is the vector of action value functions, which also satisfies the Bellman equation. The optimal action value function (i.e., the optimal objective function) is defined by

$$MQ^*(s, a) = \max_\pi MQ^\pi(s, a) \quad (3)$$

Thus, the optimal policy π^* (i.e. the optimal solution) can be obtained from

$$\pi^*(s) = \arg\max_a MQ^*(s, a) \quad (4)$$

In this basic architecture, the optimization problems of $\max_\pi MQ^\pi(s, a)$ and $\arg\max MQ^*(s, a)$ are both MOO problems.

In the design of the area coverage control Scalarized Q Multi-Objective Reinforcement Learning (SQ-MORL), a weighted-sum approach [19] is used. For the SQMORL algorithm, the state-action value function of each objective function o is defined by $Q(s, a, o)$ which is a function of state-action pair (s, a) and objective function o. By applying the weighted-sum approach, we obtain the weighted sum of the objective functions

$$SQ(s, a) = \sum_{o=1}^{M} w_o \cdot Q(s, a, o) \quad (5)$$

where by the weights are such that $w_o \in [0, 1]$ and $\sum_{o=1}^{M} w_o = 1$. In the SQMORL scheme, each agent i (i.e., sensor node i) in the SQMORL algorithm performs an update on its own action-value function. When an agent takes action a_t in state s_t for objective o, the action value function is updated. An action value function is a value that estimates how good an action selected by an agent is in a given state. The update in (6) is based on a reward r the agent received after selecting such action as well as the weighted action value function of neighboring agents in (7). Equation (7) takes account of the decisions of neighboring agents by taking the maximum of their action value function. The goal is to achieve the maximum action value function for each objective separately. The update rule at time step t for agent i is given by

$$Q_{t+1}^i(s_t^i, a_t^i, o) \leftarrow (1 - \alpha) Q_t^i(s_t^i, a_t^i, o) + \alpha \left(r_{t+1}^i(s_{t+1}^i, a_t^i, o) + \gamma \sum_{j \in Neigh(i)} f^i(j) V_t^i\left(S_{t+1}^j\right) \right), \; for \; o = 1, ..., M \quad (6)$$

$$V_{t+1}^j\left(s_t^j\right) = \max_{a \in A^j} Q_{t+1}^j\left(s_t^j, a, o\right) \quad (7)$$

where α is the learning rate where $\alpha \to 1$ refers to a rapid learning rate since the old estimate $(Q_t^i(s_t^i, a_t^i, o))$ is forgotten rapidly, $V_{t+1}^j(s_t^j)$ is the value function of neighboring agent j which is obtained from information exchange between agents

to learn about the good actions which neighboring agents have learned. Note that $f^i(j)$ is a factor that weighs the value function of neighbour j of agent i such that

$$f^i(j) = \begin{cases} \frac{1}{|Neigh(i)|}, if Neigh(i) \neq 0 \\ 1 \quad , otherwise \end{cases} \quad (8)$$

where $j \in Neigh(i)$ is in the set of neighbours of node i. Thus, the optimal policy π^* is a policy that satisfies

$$\pi^*(s) = \arg\max_a SQ^*(s,a) \quad (9)$$

In this paper, the SQMORL framework is applied to the energy-efficient coverage control problem. The following assumptions are used.

Local agent state: Assume that each agent can sense the coverage area. Let i be the index of a sensor node and s_t^i be the local state of sensor node i (agent i) at time t. Its local state s_t^i is the state of each agent i which is based on its coverage area. In the simulation part in Section 3, the coverage area is measured in terms of the number of cells, whereas in the experimental part in Section 4, coverage is measured in terms of light intensity.

Local agent actions: Let A^i be the set of all possible actions for each agent i. Each agent i has the ability to take one of the following two actions in any state it is in, i.e., $a^i \in A^i$ where $A^i = Action\ 0$ (Turn off), $Action\ 1$ (Turn on). Thus, in the simulation and experiment, the local agent action is the action taken by the sensor node which is to turn off or on the light source.

Objective functions: There are two objective functions. The first objective function is to achieve maximized coverage area. In this paper, it is assumed that the coverage area is divided into squares of 1 sq.m. referred to as cells. The reward function of the objective function for coverage area is given by:

$$r^i\left(S_t^i, a^i, 1\right) = Area\_coverage\left(a^i\right) \times GAIN\_CELL\_BRIGHT \quad (10)$$

where,

$Area\_coverage\left(a^i\right) \times GAIN\_CELL\_BRIGHT$ is a function of the number of cells. Note that a high value of $GAIN\_CELL\_BRIGHT$ increases the reward of coverage area and thus promotes an increase in the number of active nodes. On the other hand, a small value of $GAIN\_CELL\_BRIGHT$ demotes the use of active nodes as the reward is reduced.

The second objective is to achieve minimized overlapping area that occurs between sensor nodes located nearby in order to reduce the energy consumption, i.e.,

$$r^i\left(S_t^i, a^i, 2\right) = C^i \quad (11)$$

where C^i is the overlapping area in terms of the number of cells in state S_t^i as a result from action

Scalarized Q Multi Objective Reinforcement Learning

BEGIN

1 Random topology

2 Initialize $Q_t^i(s_t^i, a_t^i, o) = 0$

$SQ_t^i(s_t^i, a_t^i) = 0, \text{for } t = 0, \forall i, s, a$

3 For time step $t = 1$: end_time_step

4 Each agent chooses action $a^i \in A^i$, reward r_1^i, r_2^i, and the next state

5 s^i = number of cell coverage,

$a^i = \arg\max_a SQ_t^i(s_t^i, a_t^i)$

6 Update separate Q tables

7 $Q_{t+1}^i(s_t^i, a_t^i, o) \leftarrow (1-\alpha) Q_t^i(s_t^i, a_t^i, o) + \alpha(r_{t+1}^i(s_t^i, a_t^i, o) + \gamma \sum_{j \in Neigh(i)} f^i(j) V_t^j(s_{t+1}^j))$, for $o = 1,2$

8 where $V_{t+1}^i(s_t^i) = \max_{a \in A^i} Q_{o,t+1}^i(s_t^i, a, o)$,

$f^i(j) = \begin{cases} \frac{1}{|Neigh(i)|} & \text{, if } Neigh(i) \neq 0 \\ 1 & \text{, otherwise} \end{cases}$

9 $SQ_{t+1}^i(s,a) \leftarrow \sum_{o=1}^{m} w_o \cdot Q_{t+1}^i(s,a,o)$, $m = 2$

10 endfor

END

Fig. 1: *Pseudo code of SQMORL for area coverage control.*

$a^i \in A^i$ taken by sensor node i. The overlapping area indicates redundant coverage and thus unnecessary energy consumption. The value of C^i is determined from counting the number of cells in the overlapping coverage area from turning on sensor node i and nearby sensor nodes. Thus, the objective function can be defined separately for each of the objective function as $g_1(s,a) = \max[areacoverage]$ and $g_2(s,a) = \min[area\_overlapping]$. Fig 1 depicts the pseudo code of SQMORL. The algorithm converges provided that all state-action pairs are visited infinitely often [19]. In practice, a reinforcement learning algorithm is considered to converge when the learning curve of the objective function (e.g. average reward) no longer increases significantly. A typical criterion for convergence is when the value of the objective function deviates smaller then a specified threshold.

3. PERFORMANCE EVALUTION: SIMULATION

This section evaluates the performance of the SQMORL algorithm for area coverage control in WSNs. We consider an area of 35 x 35 sq.m. space. Each sensor node has an area coverage of radius 5 m which covers an area of 81 cells per sensor node. Therefore, there are 82 possible states for each agent in the system. The sensors are laid out in 1) a uniform random placement with varying number of 25, 50, 75, 100 sensor nodes; and 2) a grid placement with varying number of 25, 81, 121,169 sensors nodes.

From Fig. 1, the learning rate α is 0.4, the discount factor γ is 0.7 and $GAIN\_CELL\_BRIGHT$ is 0.5. These are the values of the learning rate and

discount factor which allows the system to perform best and have been evaluated from the experiment. In the MOO framework, the weights in line 9 of Fig. 1 represent a trade-off which is parameterized by w_1, $w_2 = 0.5$. This represents a scenario which both objectives are equally important. However, a higher weight value for a particular objective promotes such objective function and demotes the other. Hence, the weight values can be adjusted in order to suit the coverage requirement by a user. The simulation results are averaged over 10 repeated runs.

When each sensor node is placed in the area, each sensor node must learn to adjust its decision in order to discover the optimal action. The optimal action is one which satisfies the purpose of covering the maximum possible area and attaining the least number of active nodes. For performance comparison the following metrics are measured: the number of working sensor nodes selected, the percent of coverage area and the ratio of coverage area per working sensor node. The proposed multi-objective SQMORL algorithm is compared with 1) a non-learning scheme, a coverage control scheme called the Optimal Geographical Density Control (OGDC) which guarantees full coverage [1]; and 2) the single objective Distributed Value Function (DVF). Although both algorithms are both multi-agent RL algorithms, DVF considers only one objective. Thus, the Q- table is based on a single reward function, $r^i\left(s_t^i\right) = G^i\left(s_t^i\right) - C^i$ combined from (10) and (11). On the other hand, the SQMORL does not combine the individual reward functions, but rather maintains separate Q-tables for each reward (10) and cost (11). The Q-tables are updated separately as a result of each selected action. However, each selected action is based on a weighted linear combination of the Q-tables of each objective function, shown in the pseudocode in Fig.1 line 9. It is worth noting that for multi-objective RL problems, combining the rewards and costs straightforwardly into a single-objective RL scheme such as the DVF, may not always provide the best possible solution. Learning solutions separately from separate Q tables as in SQMORL may provide better solutions.

3.1 Simulation results: uniform random layout

In this scenario, the sensor nodes are placed in the area following a uniform random placement. We first consider the number of working sensor nodes which each algorithm decides to cover the area. Fig. 2 shows the number of working sensors nodes at node densities of 25 to 100 nodes per area. The DVF and SQMORL algorithms use 8 to 16 and 8 to 15 working nodes, respectively. As for the OGDC algorithm, it uses 18 to 26 working nodes to cover the area. From the figure, results indicate that SQMORL algorithm uses a comparable number of working nodes to the DVF algorithm while the OGDC algorithm uses the most

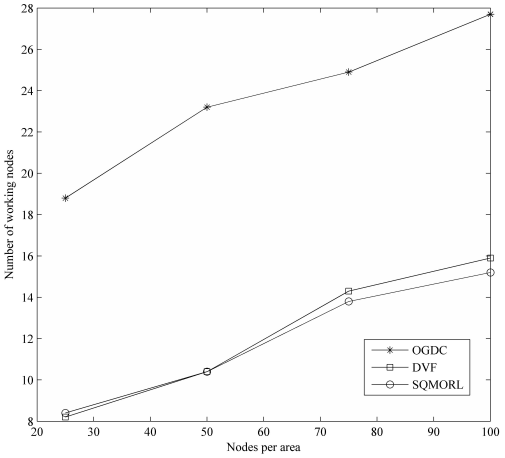

Fig.2: *Number of working nodes against number of nodes placed in the random topology network.*

number of working nodes.

Fig.3 depicts the percentage of coverage area obtained by each algorithm given the number of working nodes, for node densities of 25 to 100 nodes per area. Note that the DVF, SQMORL and OGDC algorithm can attain 75 to 95, 79 to 97 and 88 to 100% of coverage, respectively. At node density of 25 sensor nodes, no algorithm can attain full coverage due to the insufficient number of nodes. In terms of coverage area, the OGDC algorithm has more coverage than the other algorithms. At 100 nodes, SQMORL algorithm attains 97% coverage which is 2% more than DVF algorithm and 3% less than OGDC algorithm. However, SQMORL uses 1 and 11 fewer nodes than DVF and OGDC, respectively (see Fig.2).

To show the energy efficiency, in terms of the coverage area size (in cells) per working node, we compare the ratio of the number of cells in the coverage area over the number of working sensor nodes in Fig.4. The SQMORL algorithm can outperform DVF and OGDC algorithm in all cases. This is because SQMORL uses fewer number of working nodes, while the coverage area is comparable to the two other algorithms. Results show that though OGDC can attain the maximum coverage, it is at the expense of high number of working nodes.

3.2 Simulation results: grid layout

In order to evaluate SQMORL in a scenario similar to the light bulb placement as the automatic lighting control testbed, simulation was also conducted in a grid layout of sensors nodes. The number of sensor nodes is varied to 25, 81, 121 and 169 nodes. The sensors are placed in a regular grid spaced 5m apart in an area of 30×30, 50×50, 60×60 sq.m. space, respectively. The purpose of this experiment is to evaluate each algorithm in a setting of light bulbs

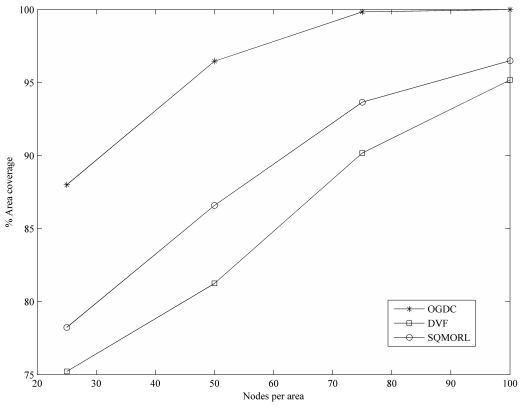

Fig.3: *Percentage of coverage area against number of nodes placed in the random topology network.*

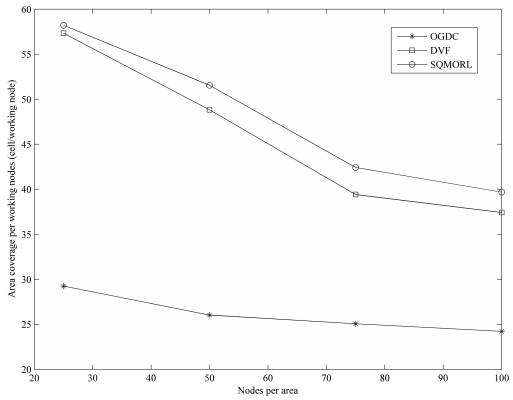

Fig.4: *Ratio between the number of cells in the coverage area and working nodes placed in the random topology network.*

placed indoors of a building. The SQMORL, DVF and OGDC algorithms have been compared.

Fig.5 shows the number of working nodes selected to cover the area. The SQMORL algorithm used 9 to 49 node working nodes, the DVF algorithm used 9 to 54 working nodes, whereas OGDC used the most working nodes with its selection of 22 to 166 nodes. This is because the SQMORL algorithm can select non-overlapping working nodes which cover the most area, whereas the DVF algorithm missed some positions. On the other hand, OGDC is only focused on uncovered area and thereby activated almost every node in the uncovered area, thus used the most number of working nodes.

Fig. 6 depicts the percentage of area coverage that each algorithm attained given their selected number of working nodes. It is found that the coverage remains relatively constant as the sensor nodes are placed at regular grid positions throughout the area. The DVF and SQMORL algorithm attained approximately 80% coverage. Note that the OGDC algorithm achieved 98% coverage area for all cases as it selects working nodes based on cells which are not yet covered by other nodes.

Fig.7 shows the energy efficiency of the working nodes selected. SQMORL algorithm achieved the highest number of cell coverage per working node at 80 cells/working node. This is because SQMORL algorithm uses fewer working nodes while attaining 80% coverage of the area (as seen in Fig.6). On the other hand, the DVF algorithm obtained 73 to 77 cells per working node and the OGDC algorithm obtained the least energy efficiency. Hence, even though OGDC can achieve the most percentage of coverage area (Fig.6), a trade-off exists as OGDC uses the highest number of working nodes (see Fig. 7). This is because OGDC considers uncovered areas and selects working nodes which overlap. Consequently, as the number of nodes placed in the area increases, the energy efficiency of OGDC algorithm decreases.

3.3 Discussion

In the simulation part, we present the MOO framework for area coverage control in WSNs. There are two objective functions, i.e. to maximize area coverage objective and to reduce the area overlapping that occurs between neighboring nodes. A multi-objective reinforcement learning method in conjunction with the weighted-sum approach, called Scalarized Q Multi Objective Reinforcement Learning (SQMORL), is then applied to find the optimal policy in area coverage control. To evaluate the performance of the SQMORL algorithm, the simulation is divided into two parts, i.e., uniform random and the grid sensor layout. The performance of SQMORL is compared with DVF algorithm and OGDC algorithm. In terms of the number of cells covered per working node ratio, the SQMORL algorithm outperforms DVF and OGDC algorithm in all cases as it requires the fewest number of working nodes. Similar results were obtained for both the uniform random and grid layout.

In the next section, SQMORL and DVF algorithms have been selected for performance evaluation in an implemented automatic lighting control testbed. These two algorithms have been selected because they both have self-learning characteristics, good efficient energy consumption with respect to area coverage and outperform OGDC.

4. PERFORMANCE EVALUATION: TESTBED

In this section, SQMORL and DVF are evaluated in an automatic lighting control testbed. In the testbed, each sensor is initialized to the initial default value setting i.e., Q-values for all state-action pairs of each agent equal to zero. There are two actions, i.e., "$Action0$" refers to turning off a light bulb, and "$Action1$" refers to turning on a light bulb.

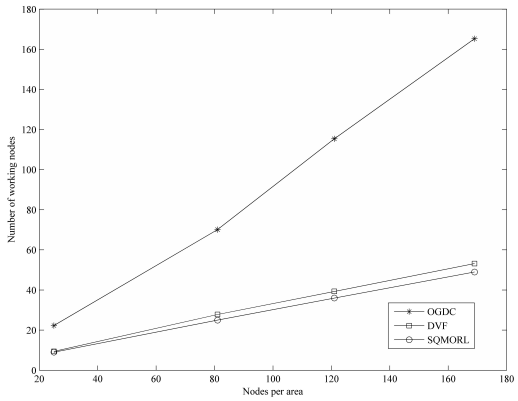

Fig.5: *Number of working nodes against number of nodes placed in the grid topology network.*

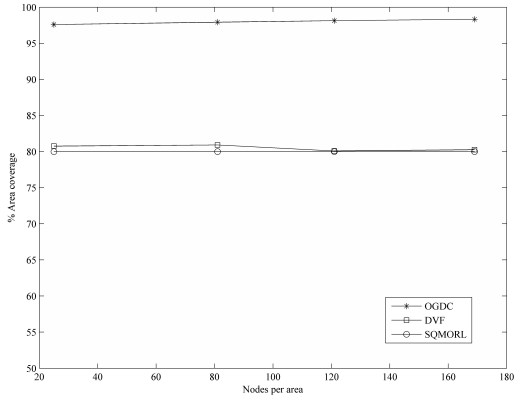

Fig.6: *Percentage of area coverage against number of nodes placed in the grid topology network.*

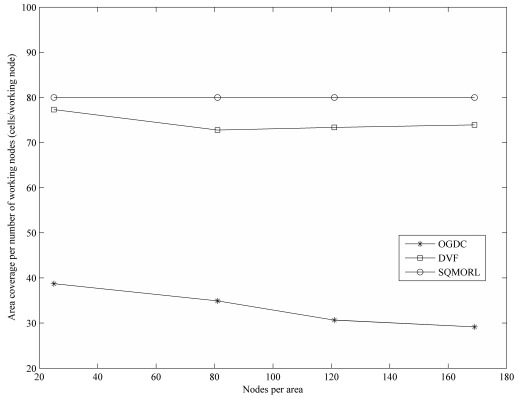

Fig.7: *Ratio between the number of cells in the coverage area and working nodes placed in the grid topology network.*

As the SQMORL convergence condition requires that every state-action pair be visited infinitely often, an "explore and exploit" scheme is implemented at each agent. In particular, each agent is set to randomly select (explore) actions for 100 time steps in the training phase. This enables each agent to explore all possible state-action pairs and update the action value functions. After the training phase, these values are then used to select (exploit) optimal action according to (4) with some probability $\varepsilon > 0$ and explore other actions randomly with the remaining probability. This is referred to as the Îţ-greedy action selection scheme. When an action is selected, every node waits for 5 seconds for the light intensity to become stable as a certain amount of delay is required in the hardware to turn on or off each light bulb. The nodes then measure the resulting light intensity and obtain the reward values from equations (10), (11). Each node then updates their state-action value functions according to equation (6). Then agent sends its own value function in (7) to its neighboring sensor nodes. The neighbors can then have the up-to-date value functions for their own state-action value updates. The process is repeated as shown in Fig.8 until convergence is achieved, i.e., the agent can find the optimal action. To ensure convergence in the testbed, the action value of each sensor node was first trained offline from simulation until near-optimal action values were achieved. The action values were then stored on the sensor nodes and further online training was performed in the testbed. The DVF algorithm was considered to converge when there is no further change in the actions taken in each state.

4.1 SQMORL automatic lighting control results

In order to evaluate the performance of the DVF and SQMORL algorithm, an automatic lighting control test bed was developed in the Wireless Communication Laboratory F4, Suranaree University of Technology (SUT), Thailand.

The automatic lighting control system consists of sensor nodes, each of which is equipped with a wireless communication module with XBee Series 2, a microcontroller part with Arduino Uno R3 and an additional external memory unit for recording measurements for control purposes, a light dependent resistor (LDR) to measure the light intensity. Each sensor node has the ability to measure the intensity of light within its own area, exchange information between the neighboring node sensors and collect the data in its memory unit. The sensor nodeâĂŹs own coverage area is measured when all other light sources from other sensor nodes are switched off. When other sensor nodes switch on their light, the light intensity is measured. Each level of light intensity corresponds to a particular status of the light sources in the testbed. Thus by measuring the level of light intensity, a sen-

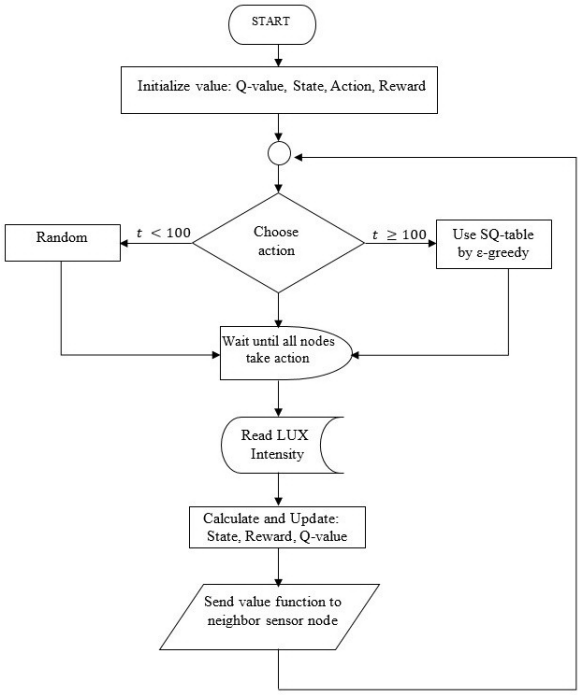

Fig.8: *Diagram for SQMORL for automatic lighting control testbed.*

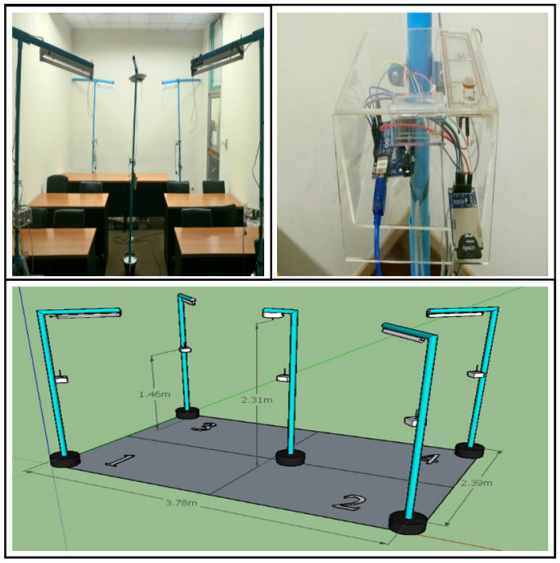

Fig.9: *Automatic lighting control testbed.*

sor node can determine the layout of the overlapped area. Five sensors nodes are placed at the positions as shown in Fig. 9, in the experiment room of which external light is blocked. The results are averaged over 20 repeated runs.

Fig. 10 shows the final policy obtained by SQ-MORL and DVF algorithms. Since there are 5 sensor nodes and each sensor node has 2 actions, there are 32 possible policies. Sensor node 0 is placed in the middle of the room (see Fig.9). If sensor node 0 turns on, its light coverage would overlap with that from all the other sensor nodes.

As the light intensity of the 4 sensor nodes in the corner provides sufficient coverage area, the sensor node placed at the center can be turned off to save energy. Thus, the optimal policy for this setting is where the 4 sensors in the corners of the room are turned on and sensor node in the middle of the room is turned off (i.e., the 16^{th} policy). Note that both algorithms were programed to select random actions for exploration during training in the first 100 time steps. Then the algorithms learned on their own according to the $\hat{I}\zeta$-greedy action selection scheme. However, it was initially found that the training phase of 100 time steps was insufficient as both algorithms could not find the optimal policy. This is shown by the Direct Learning DVF graph in Fig. 10, which the optimal policy (the 16^{th} policy) was not found at all in the 20 repeated runs. Therefore, to enhance the learning rate, we trained the sensor nodes off-line through simulation to obtain the state-action value tables prior to the testbed implementation. Then, we saved the trained Q-value tables in the memory unit of each sensor node. With this off-line training method, the sensor nodes are able to quickly adjust their actions to the testbed environment. Results are shown by the graphs labeled Trained DVF and Trained SQ-MORL which are the algorithms that learned policies from off-line training, i.e. with initialization from the trained Q-value tables. Fig. 10 shows that the Trained DVF and Trained SQMORL algorithms are able to find the optimal policy at 55% and 80% of the 20 repeated runs, respectively.

Fig.11 shows the measured light intensity of each sensor node running the DVF algorithm (sensor node 0 is placed at the center and the rest of the sensor nodes are placed in the corners). It should be noted that these measurements are made only after the DVF algorithm has learned the optimal policy. During training in the first 100 time steps, each node explores all actions by randomly selecting its actions. After training, each sensor node makes its decision based on exploitation of its trained value table. Sensor node 0 achieves the least light intensity of 116 lux as its own light bulb is turned off. Its light intensity obtained is from the 4 neighboring sensor nodes.

Fig. 12 shows the average reward of the DVF scheme which is computed from the reward obtained from each sensor node from the beginning until current the time step. Note that during the training phase with the random selection of actions, the average reward of all sensor nodes does not increase. However, once the training period is over at the 100^{th} time step, each sensor node can choose the optimal action at the 117^{th} time step, which is seen from the increase in average reward consistently.

Fig. 13 shows the measured light intensity of each sensor node under the SQMORL algorithm. The SQ-MORL algorithm can learn the optimal policy. This

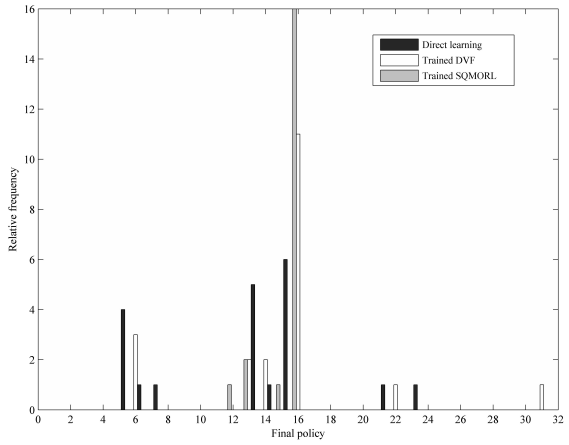

Fig.10: *Final policy distribution for each algorithm.*

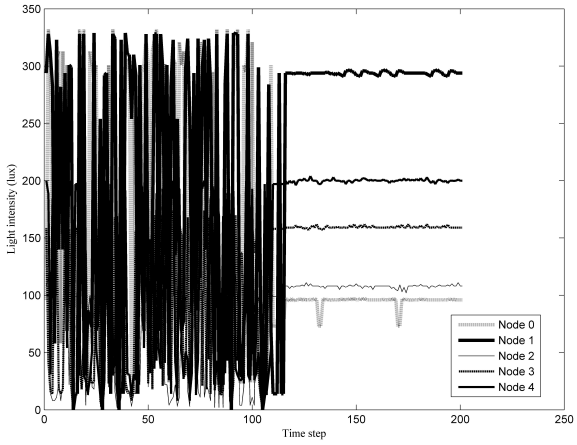

Fig.11: *Node light intensity from the DVF algorithm.*

result is similar to that of the DVF algorithm as the same the optimal policy (i.e., the 16^{th} policy) is expected.

Fig. 14 shows the average reward of all 5 sensor nodes obtained from the 1^{st} objective function which aims to maximize the light intensity. Results show that during training period in the first 100 time steps, the average reward does not increase due to the exploration from randomly selected actions. But after the training period, each sensor node can choose its optimal action consistently from time step 130 onwards, as seen from an increase of the average reward of the sensor nodes. Fig. 15 shows the average cost of all 5 sensor nodes obtained from the 2^{nd} objective function, which aims to reduce the overlapping areas. Results show that the consistently decreasing average cost at all sensor nodes occurs after time step 130, implies that the light intensity which occur from overlapping area decreases after this time step.

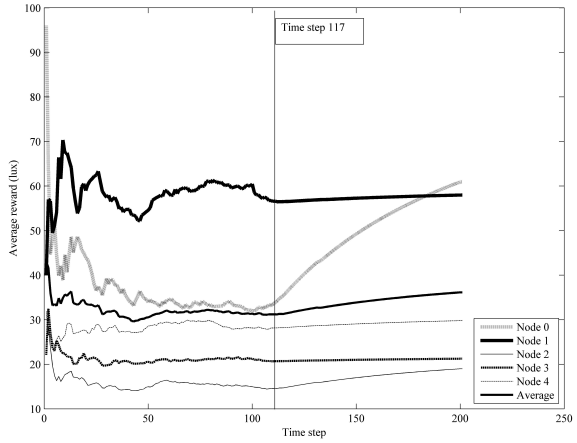

Fig.12: *Average reward of each sensor node from the DVF algorithm.*

4.2 Discussion

To test the performance of both the SQMORL and DVF algorithms, an automatic lighting control testbed has been implemented to evaluate the performance of SQMORL and DVF algorithms. From the experiment results, the DVF and SQMORL algorithms can obtain the optimal policy at 55% and 80% from the 20 repeated runs, respectively. In terms of the convergence speed, the DVF and SQMORL algorithm reached the optimal policy at time step 117 and 130, respectively. Such results suggest that the SQMORL algorithm can find the optimal policy more frequently than the DVF algorithm at a comparable convergence rate. Therefore, results suggest that MOO framework based on the SQMORL algorithm may be suitable for coverage control in WSNs, particularly for applications which require maximum coverage and minimum overlapping coverage.

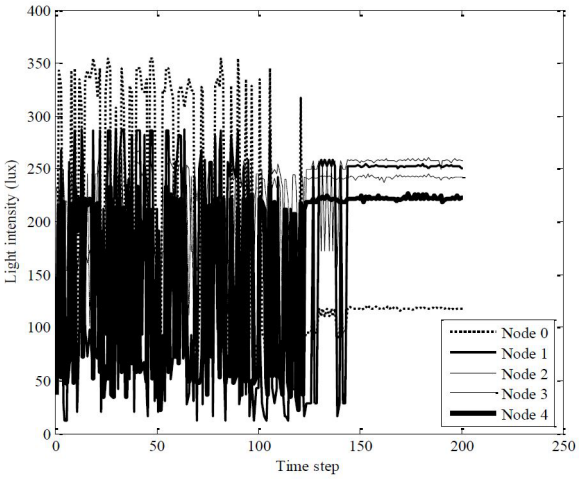

Fig.13: *Node light intensity from the SQMORL algorithm.*

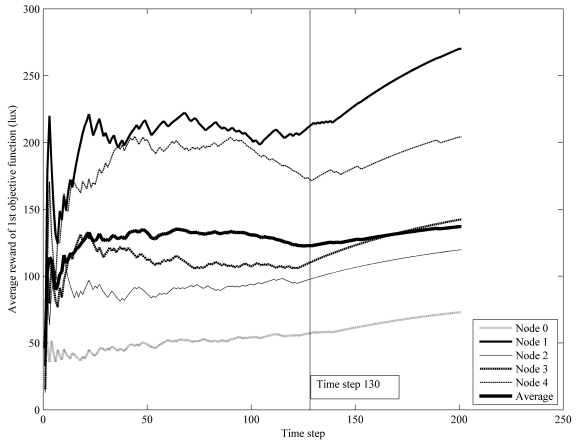

Fig.14: *Average reward of 1st objective function from the SQMORL algorithm.*

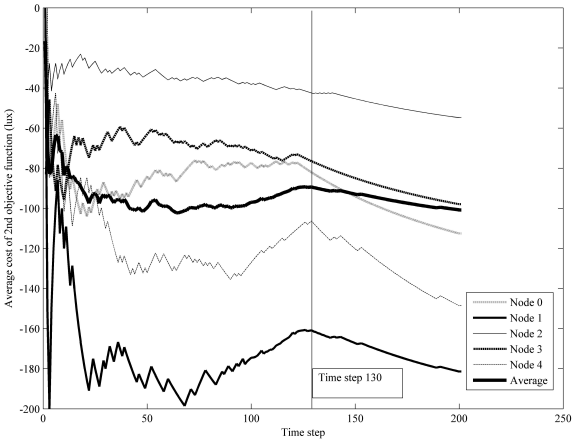

Fig.15: *Average cost of the 2^{nd} objective function from the SQMORL algorithm.*

5. CONCLUSION

The objective of this work is to extend coverage control using the single combined objective reinforcement learning algorithm to a multi-objective optimization (MOO) reinforcement learning framework. In particular, this paper proposes the Scalarized Q Multi-Objective Reinforcement Learning (SQMORL) which uses a MOO based on separated rewards for each objective function for the coverage control problem in WSNs. The algorithm has advantages of low complexity and scalability. The MOO framework also allows the optimal policy particularly for contradicting objectives to be found more effectively than the combined single objective function. This is evident from the simulation results in the uniform random node placement and grid node placements. In addition, this work has also developed a hardware testbed to evaluate the performance of SQMORL and the DVF in a multi-agent lighting control experiment. Results show that the SQMRL algorithm can efficiently find the optimal policy more accurately over the DVF algorithm.

References

[1] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Int. J. Ad Hoc Sensor Wireless Networks*, Vol. 1, No. 1-2, pp.89-124, Mar. 2005.

[2] A. A. Kumaar, G. Kiran and T. S. B. Sudarshan, "Intelligent Lighting System using Wireless Sensor Networks," *Int. J. Ad hoc, Sensor Ubiquitous Comput.*, Vol. 1, No. 4, pp. 17-27, Dec. 2010.

[3] T. P. Huynh, Y. K. Tan and K. J. Tseng, Energy-Aware, "Wireless Sensor Network with Ambient Intelligencefor Smart LED Lighting System Control," *Proc. Annu. Conference IEEE Ind. Electron. Soc.*, 2011.

[4] R. Mohamaddoust, A. T. Haghighat, M. J. M. Sharif and N. Capanni, "A Novel Design of an Automatic Lighting Control System for a Wireless Sensor Network with Increased Sensor Lifetime and Reduced Sensor Numbers," *J. Sensors*, Vol. 11, pp. 8933-8952, Sep. 2011

[5] P. Meng-Shiuan, Y. Lun-Wu, C. Yen-Ann, L. Yu-Hsuan and T. Yu-Chee, "A WSN-based Intelligent Light Control System Considering User Activities and Profiles ," *IEEE Sensor J.*, Vol. 8, No.10, pp. 1710-1721, Sep. 2008,

[6] M. Okada, H. Aida, H. Ichikawa and M. Miki, "Design and Implementation of an Energy-EfĭňĄcient Lighting System Driven by Wireless Sensor Networks," *Proc. Int. Conference Mobile Comput. Ubiquitous Networking*, Mar. 2015.

[7] M. Iqbal, M. Naeem, A. Anpalagan, N. N. Qadri and M. Imran, "Multi-objective Optimization in Sensor Networks: Optimization Classification, Applications and Solution Approaches," *J. Comput. Networks*, Vol. 99, pp. 134-161, Apr. 2016.

[8] R. Tharmarasa, T. Kirubarajan, J. Peng and T. Lang, "Optimization-Based Dynamic Sensor Management for Distributed Multitarget Tracking," *IEEE Trans. Syst., Man, Cybernetics Part C*, Vol. 39, No. 5, pp. 534-546, Sep. 2009.

[9] M. Iqbal, M. Naeem, A. Anpalagan, A. Ahmed and M. Azam, "Wireless Sensor Network Optimization: Multi-Objective Paradigm," *J. Sensors*, Vol. 15, No. 7, pp. 17572-17620, Jul. 2015.

[10] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen and L. Hanzo, "A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms, and Open Problems," *IEEE Commun. Surveys Tutorials*, Vol. 19, No.1, pp. 550-586, Sep. 2016.

[11] V. Singhiv, A. Krause, C. Guestrin, J. H. Garrett and H. Matthews, "Intelligent Light Control using Sensor Networks," *Proc. 3rd Int. Conference*

Embedded Networked Sensor Syst., pp. 218-229, Nov. 2005.

[12] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, "Handling Multiple Objectives with Particle Swarm Optimization," *IEEE Trans. Evol. Comput.*, Vol. 8, No.3, pp. 256-279, Jun. 2014.

[13] J. Jia, J. Chen, G. Chang and Z. Tan, "Energy efficient Coverage Control in Wireless Sensor Networks based on Multi-Objective Genetic Algorithm," *J. Comput. Math. Applicat.*, Vol. 57, No.11-12, pp. 1756-1766, Jun. 2009.

[14] J. Barbancho, C. Leon, F. J. Molina and A. Barbancho, "Using Artificial Intelligence in Routing Schemes for Wireless Networks," *J. Comput. Commun.*, Vol. 30, No.14-15, pp. 2802-2811, Oct. 2007.

[15] Z. Tafa, "Artificial Neural Networks in WSNs Design: Mobility Prediction for Barrier Coverage," *Proc. IEEE Int. Symp. Signal Proc. Inform. Technology*, Dec. 2016.

[16] M. Rovcanin, E. D. Poorter, D. Akker, I. Moerman, P. Demeester and C. Blondia, "Experimental Validation of a Reinforcement Learning based Approach for a Service-wise Optimisation of Heterogeneous Wireless Sensor Networks," *J. Wireless Networks*, Vol. 21, No.3, pp. 931-948, Apr. 2015.

[17] A. Phuphanin and W. Usaha, "A Multi-Agent Scheme for Energy-Efficient Coverage Control in Wireless Sensor Networks," *Proc. Int. Conference Inform. Technology Sci.*, Jun. 2016.

[18] S. M. Jameii, K. Faez and M. Dehghan, "Multi-Objective Optimization for Topology and Coverage Control in Wireless Sensor Networks," *Int. J. Distributed Sensor Networks*, Vol.11, No. 2, pp.1-11, Feb. 2015.

[19] K. V. Moffaert, M. M. Drugan and A. Nowe, "Scalarized Multi-Objective Reinforcement Learning: Novel Design Techniques," *Proc. IEEE Symp. Adaptive Dynamic Programming Reinforcement Learning*, Apr. 2013.

Akkachai Phuphanin received his Bachelor's Degree in Telecommunication Engineering from Suranaree University of Technology in 2009. For his postgraduate study, he continued his Master's degree in the School of Telecommunication Engineering, Institute of Engineering, Suranaree University of Technology. He is currently pursuing his Ph.D in the same school. His current research interests include the design and simulation of network area coverage control in wireless sensor networks and smart home applications.

Wipawee Usaha received her Bachelor's Degree (Hons) in Electrical Engineering from Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani, Thailand, and Master's and PhD Degree in Communications and Signal Processing from Imperial College London, London, U.K.. She is currently an assistant professor at the School of Telecommunication Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand. Her current research interests include resource allocation and event detection in wireless sensor networks and their applications.