Solving Coverage Problem by Self-organizing Wireless Sensor Networks: (ϵ,h)-Learning Automata Collective Behavior Approach

Franciszek Seredyński[1], Miroslaw Szaban[1], Jaroslaw Skaruz[1], Piotr Świtalski[1], and Michal Seredyński[1]

University of Siedlce, Institute of Computer Science, Siedlce, Poland
franciszek.seredynski@uws.edu.pl miroslaw.szaban@uws.edu.pl
jaroslaw.skaruz@uws.edu.pl piotr.switalski@uws.edu.pl
michal.seredynski@uws.edu.pl

Abstract. We propose a novel multi-agent system approach to solve a coverage problem in Wireless Sensor Networks (WSN) based on the collective behavior of (ϵ,h)-Learning Automata (LAs). The coverage problem can be stated as a request to find a minimal number of sensors spending energy of their batteries to provide the requested level of coverage of the whole monitored area. We propose a distributed self-organizing algorithm based on the participation of LAs in an iterated Spatial Prisoner's Dilemma game. We show that agents achieve a solution corresponding to Nash equilibrium, which provides maximization of not known for agents a global criterion related to the requested level of the coverage with a minimal number of sensors which turn ON their batteries.

Keywords: Collective behavior, Learning automata, Network coverage problem, Self-organization, Sensor networks, Spatial Prisoner's Dilemma.

1 Introduction

Wireless Sensor Networks (WSN) is a fast-developing technology belonging to a broader group of information and communication technologies [8] applied today in the Internet of Things. They are composed of a large number of tiny computers- communication devices called sensors deployed in some areas- that can sense a local environment and send related information to a remote user who can make an appropriate decision. Designing such systems is a complex task demanding solving a number of issues on different levels related to a single sensor node, a network of nodes, applications, etc. (see, e.g., [7]).

This paper focuses on some issues related to designing fault-tolerant WSN-based mission-critical systems. We assume that monitoring is performed in a remote and difficult-to-access area, and sensors are equipped with single-use batteries that cannot be recharged. From the Quality of Service (QoS) point of view of such WSN, two closely related important issues exist: how to perform

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

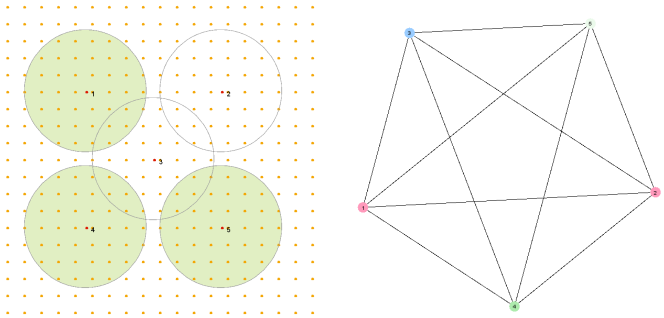
effective monitoring (coverage) of an area and how to maximize an operational lifetime. After deploying sensors, they should recognize their nearest neighbors to communicate and start making local decisions about turning ON or OFF their batteries to monitor events. These decisions will directly influence the level of area coverage, the amount of spending on sensors' battery energy, and the network's lifetime. One can notice that the lifetime maximization problem is closely related to the coverage problem. A group of sensors monitoring some area is usually redundant, i.e., more than one sensor covers the monitored targets, which forms some redundancy that can be exploited. Solving the coverage problem is crucial to solving the problem of maximization of the WSN's lifetime. In the paper, we will focus on the problem of the coverage.

This paper presents a novel approach to the problem of coverage, based on self-organization with the use of (ϵ,h)- Learning Automata (LAs) [14, 10]. Sensors do not have knowledge concerning a current level of WSN coverage or a total number of sensors. They have only information concerning their neighbors. Nevertheless, we expect that the system we will be able to self-optimize, i.e. to find, in a fully distributed way, a minimal number of sensors turned ON providing a necessary value of the level of coverage. Our approach is based on a multi-agent interpretation of the coverage problem and applies a recently proposed [11] methodology of game–theoretic interactions between players participating in the Spatial Prisoner's Dilemma (SPD) game. The paper extends the work [12], where a self-organizing system based on the application of a second-order Cellular Automata (CA) has been proposed. We show that the presented approach significantly outperforms the approach based on the application of CA.

Our approach contrasts with the others, currently used to solve the problem of the coverage. Because the problem is known to be NP-complete [2], centralized algorithms are oriented either on the delivery of exact solutions for specific cases (see, e.g. [1]) or applying metaheuristics to find approximate solutions (see, e.g. [5]). The main drawback of centralized algorithms is that they assume the availability of complete information about the problem and require an adequte computing power to find a schedule of sensors' activities. It means practically that it can be done only on a site of the remote user, and a solution must be delivered to WSN before starting the operation. As a result, distributed algorithms with only different forms of partial information about the problem have become more and more popular (see, e.g.[3]). Computational power independence of WSN from a remote user can provide scalability, and the entire operation of a WSN in real-time can be achieved only by self-optimizing systems. The need for such systems has been recognized in recent years in many industrial systems (see, e.g., [4]).

The structure of the paper is as follows. The next section states the problem of a coverage optimization in WSN. Section 3 presents a multi-agent approach to distributed online solving the considered problem. The concept of (ϵ,h)-LA used as players in the game is proposed in Section 4. Section 5 discusses relations between concepts of a global solution and Nash equilibria. Section 6 presents results of the experimental study, and the last section contains conclusions.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42



(a) (b)

Fig. 1: WSN 5: (a) an area monitored by WSN consisting of 5 sensors with $R_s = 18$ m, (b) a WSN graph for $R_s = 35$ m.

2 Coverage Problem in Wireless Sensor Networks

We assume that an area of the size of $L_1 \times L_2$ m^2 should be monitored by a WSN consisting of N sensors $(s_1, s_2, ..., s_i, ..., s_N)$ deployed over this area. More specifically, the area is represented by M Points of Interest (PoI) which regularly cover the area. Each sensor has a non–rechargeable battery and can monitor PoIs in a sensing range of R_s if its battery is turned ON. Fig.1a shows an example of such an area with $L_1 = L_2 = 100$ m with $M = 441$ PoI (in orange), where WSN 5 consisting of $N = 5$ sensors with $R_s = 18$ m was deployed. Some sensors of WSN are currently turned ON and monitor the corresponding area.

It assumed that a QoS measure exists evaluating the quality of a WSN performing monitoring. As such a measure, we accept the coverage value q defined as the ratio of the number of PoIs covered by active sensors to the whole number M of PoIs, i.e. $q = \frac{M_{obs}}{M}$. A desirable objective is to preserve the complete area coverage, but sometimes, it may be more practical to achieve a predefined c o v e r age rate that is just high enough. Therefore, we assume that this ratio should not be lower than some predefined requested value q_r $(0 < q_r \leq 1)$.

The coverage problem can be stated in the following way. Find a solution $s = (s_1, s_2, ..., s_i, ..., s_N)$, where $s_i = \{0, 1\}$ with a corresponding value of the coverage $q(s)$ and a number $n\_ON(s)$ of sensors turned ON such that it fulfills the following requirements: a) a number n_ON of sensors turned ON is minimal, b) $q(s) \geq q_r$, and c) $q(s)$ has the largest possible value.

We have to deal with a combinatorial optimization problem, which can be described by a proposed function that should be maximized:

$$f(q(s), n\_ON(s), q_r) = \begin{cases} N - n\_ON(s) + q(s), \ if \ q(s) \geq q_r \\ q(s), \ if \ q(s) < q_r. \end{cases} \quad (1)$$

This function assigns univocally values to solutions in such a way that a maximal value of it corresponds to a solution (or solutions) which provides a

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

maximal value of q fulfilling the requirement $q \geq q_r$ under the minimal value of n_ON of sensors turned ON.

We want to solve the coverage problem online in real-time using only a tiny amount of computational power and the communication possibilities of WSN sensors. A potential solution algorithm should work in a monitored area in real-time and react quickly to changes in values of parameters of sensors. Therefore, we will focus in this paper on working out a variant of a distributed algorithm to solve the coverage problem by self-optimization.

The first step in this direction is converting a WSN instance into a WSN interaction graph. Such a graph will be used as a core of a multi-agent system oriented on solving a coverage problem. The conversion is based on the principle saying [12] that two nodes of a WSN graph are connected if they have at least one common PoI within their sensing range R_s in a corresponding WSN.

Fig.1b shows an interaction graph for WSN 5 presented in Fig. 1a under an assumption that $R_s = 35$. The interaction graph contains five nodes, each corresponding to a sensor from the instance. The degree of each node corresponds to a number of neighbors which depends on the value of R_s, and in this case, each sensor has four neighbors.

3 Multi–agent System for Online Coverage Optimization

3.1 Agents and Their Actions

We assume that each node of a WSN interaction graph is controlled by an agent A_i of a multi-agent system consisting of N agents. Each agent has two alternative decisions (actions): $\alpha_i = 0$ (battery is turned OFF) and $\alpha_i = 1$ (battery is turned ON) and a decision unit is responsible for making a decision about turning ON and OFF the sensor. Different ways of coordination of agent decisions exist to reach a common goal. In our approach, we assume that the interaction of agents is based on a game-theoretic model, which is a variant of SPD game [11] related to the WSN coverage optimization problem [12]. In this game-theoretic model, we assume that all agents make discrete-time decisions regarding the activation of their batteries using specific rules/strategies.

We assume the following set of rules is available by agent-players:

- *all C*: always cooperate (C) what corresponds turning ON battery ($\alpha_i = 1$),
- *all D*: always defect (D) what corresponds to turning OFF battery ($\alpha_i = 0$),
- *k–D*: cooperate until not more than k neighbors defect, otherwise defect,
- *k–C*: cooperate until not more than k neighbors cooperate, otherwise defect,
- *k–DC*: defect until not more than k neighbors defect, otherwise cooperate.

One can notice that the first two rules do not consider player-neighbor decisions. The remaining three rules of a given agent-player take into account the decisions of player-neighbors from a previous round. Selecting a rule to turn ON/OFF a battery depends on an algorithm of a player decision unit. While in our previous study [12] we were using a second order CA, in this paper, we propose to use a new original reinforcement learning algorithm called (ϵ,h)-LA.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

3.2 Payoff Function of a Game

It is assumed that each agent-player knows the value of a requested coverage q_r and considers it as a local value $q_r^i = q_r$ which must be fulfilled. He participates in an iterated SPD-like game [12] consisted of some number of rounds (iterations) conducted in moments of time $t = 1, 2, ..., T$, where T is known only for an organizer (a remote user of WSN) of the iterated game. An agent decides whether to turn on his battery (i.e. cooperate (C)) or turn off it (i.e. defect (D)) and obtain a payoff that depends on his decisions (C/D) and of only his neighbors decisions. Neighbors of a given player can be considered as a virtual player-opponent. A value of a local coverage q_{curr}^i is a result of a game of the $i-th$ player with his virtual opponent. His payoff in a game depends on whether his current q_{curr}^i is below or above the requested q_r^i. The payoff function of a player is given in Tab. 1.

Table 1: Payoff function of SPD-like game for coverage problem

$i-$th agent's action	fulfilment of q_r^i	
turn ON battery (C)	$q_{curr}^{i-off} \geq q_r^i$	
	no	yes
	$payoff_i^{on+} = d$	$payoff_i^{on-} = c$
turn OFF battery (D)	$q_{curr}^i \geq q_r^i$	
	no	yes
	$payoff_i^{off-} = a$	$payoff_i^{off+} = b$

The payoff function assigns values to the i–th player in the following way:

- if he "turns OFF battery" then he calculates his local value of coverage q_{curr}^i; if this value $q_{curr}^i \geq q_r^i$ then he receives a payoff equal to b, otherwise a payoff equal to a,
- if he "turns ON battery" then he calculates what would be his value of q_{curr}^i (denoted as q_{curr}^{i-off}) if in fact he would have "turned OFF" his battery; if $q_{curr}^{i-off} < q_r^i$ then he receives a payoff equal to d, otherwise a payoff equal to c.

The proposed payoff function transforms the global optimization criterion (see Eq. 1) stated in Section 2 for the coverage problem into local optimization goals of players.

We assume that players are rational and act in such a way to maximize their payoff defined by the payoff function. However, we are not be interested in players' payoffs but in the evaluation of the level of collective behavior of the system. As a measure of the collective behavior of the system, we use an external criterion (not known for players) - the average total payoff (ATP) $\bar{u}()$:

$$\bar{u}(s_1, s_2, ..., s_i, ..., s_N) = \frac{1}{N}\sum_{i=1}^{N} u_i(A_i(s_i), A_i^{virtual}(s_{i_1}, s_{i_2}, ..., s_{i_r})), \qquad (2)$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

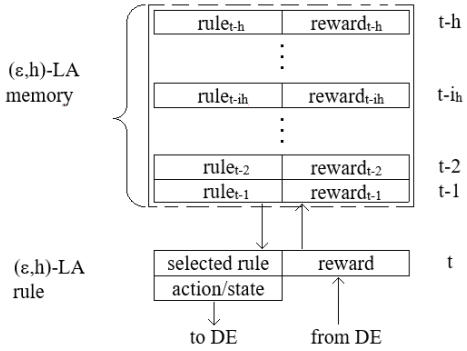


Fig. 2: Proposed (ϵ,h)-Learning Automaton.

where $u_i()$ is a payoff of an agent-player A_i in a game with a virtual player $A_i^{virtual}$ and i_r is the number of neighbors of a player A_i corresponding to his opponent, the player $A_i^{virtual}$.

Game theory predicts that players' behavior in noncooperating games is oriented towards achieving a Nash equilibrium (NE). We call a value of ATP corresponding to given NE the price of this NE. The game can have many NE points with different ATP. We call NE with the highest ATP the *maximal price point* (MPP). The question is whether we can expect of such a behavior of players that while they attempt to reach a NE at the same time ATP of the whole set of players is maximized, i.e. MPP is reached. Such a behavior depends on many factors, and one of them is a model of a player making decisions. In this paper, we examine the collective behavior of players modeled by (ϵ, h)-LA.

4 (ϵ, h)-Learning Automata

LAs are reinforcement learning algorithms proposed in 1969 by Tsetlin (see, [13]) and further extended and studied e.g. by [14, 6, 9] and many others. LAs distinctive feature is working in random environments and their ability to adapt in them. An idea of a Learning Automaton (LA) working in a deterministic environment and called ϵ-automaton was presented in [14] with a comment saying that such an idea seems to have a sense only if a deterministic environment can be randomized. However, it was shown in [10] that the concept of ϵ-LA is useful in game-theoretic models related to variants of PD game with deterministic environments. In this paper we extend this idea and propose a concept of (ϵ, h)-LA suitable for an SPD-like game model oriented on solving coverage problems in WSN.

Fig. 2 presents the proposed (ϵ, h)-LA construction. Each LA uses a subset of m rules $(0 < m \leq 5))$ from the whole set of available 5 rules (see, subsection 3.1). LA has a memory of a length h, where it stores pairs $(rule_t - i_h, reward_t - i_h)$ from last h moments of a discrete time $t-1, t-2, ..., t-i_h, ..., t-h$ of its

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

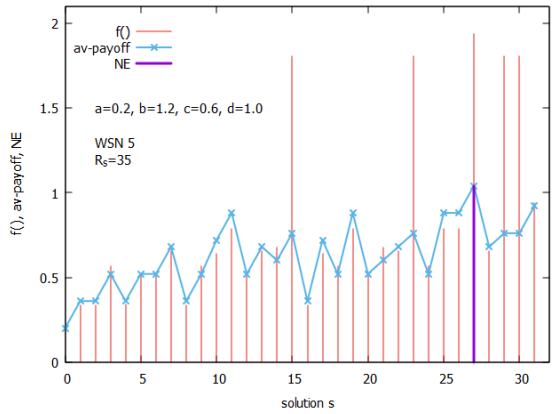


Fig. 3: Landscape of a global criterion function $f()$ and external criterion ATP for WSN 5.

operation. Each pair contains an information about a rule selected by LA at a discrete time $t - i_h$ and corresponding rewards for executing this rule.

At time t LA selects a new rule in the following way: with a probability $1 - \epsilon$ $(0 < \epsilon < 1)$ it selects from the memory a rule with the highest reward, or with a probability ϵ selects randomly a rule from a set of m rules. The selected rule is executed, i.e., the state of a corresponding sensor battery is changed by performing an action turn ON or turn OFF. Selected rule and a corresponding action (C or D) are sent to DE, which returns a reward for this action. Records of the memory are shifted in such a way that the oldest record is removed, and a new record with the latest strategy and corresponding reward is added to the memory.

5 Nash Equilibria and Global Solutions

Our approach based on self-organization differs significantly from the classical one. We replace the problem of global optimization (see, Eq. 1) by the problem of reaching NE by the agent-players. The level of self-organization, which we call a level of collective behavior, is measured by the value of ATP at NE.

ATP is closely related to two global characteristics of WSN represented by parameters n_ON and q_r, which are also not known to the players. The payoff function presented in Table 1 was designed in such a way to express global parameters n_ON and q_r through local goals of the players. A correctly designed game should correctly link the global optimization criterion with a value of ATP at NE. We focus in this section on showing relations between global solutions offered by Eq. 1 and distributed solutions corresponding to ATP (Eq. 2) at NE.

Due to the exponential increase in the number of solutions, an analysis of relations between NE and a global optimization criterion can be performed only for small instances of the coverage problem. Therefore, in this Section, we will

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

use WSN 5 shown in Fig. 1a under the assumption that $R_s = 35$ m. WSN graph of this instance is shown in Fig. 1b. For the analysis, the payoff function with the settings: $a = 0.2, b = 1.2, c = 0.6, d = 1.0$ will be used, and $q_r = 0.8$.

Fig. 3 presents landscapes of both functions: the global function f() (see, Eq. 1) and ATP (see, Eq. 2) for WSN 5. The space of solutions s of the coverage problem consists of 32 solutions. Fig. 3 presents values of f() (in red) and ATP (in blue) for all solutions. One can notice that both functions indicate $s_{27} = (1, 1, 0, 1, 1)$ as an optimal solution with corresponding values $f(s_{27}) = 1.94$ and $\text{ATP}(s_{27}) = 1.04$, respectively. The solution shows that the number of sensors turned ON is equal to 4, with the corresponding value of $q = 0.94$.

It can be shown that s_{27} is NE, and at the same time it is *maximal price point* (MPP). It is a unique NE for WSN 5 and is marked in violet in Fig. 3.

6 Experimental Results

A number of simulation experiments have been conducted to learn the performance of the proposed methodology. The monitored area was of the size $L_1 = L_2 = 100\ m$ with $M = 441$ PoI. We use the game settings $a = 0.2, b = 1.2, c = 0.6, d = 1.0$, the request $q_r = 0.8$ and LAs with parameters $h = 8$ and $\epsilon = 0.005$. Batteries capacity is unlimited. Each experiment lasted 1000 iterations. When necessary, experimental results were averaged over 30-50 runs.

6.1 The Instance: WSN 5

The purpose of this set of experiments was to get some inside into the work of the self-organizing algorithm when a small instance of the problem WSN 5 is used. The experiments have been conducted under an assumption that $R_s = 35\ m$. The WSN interaction graph of the multi-agent system is presented in Fig. 1b and a theoretical analysis of the game was presented in Section 5. Experiments have been conducted under the assumption that the whole set of 5 rules is used.

Fig. 4 presents results of a single run of the algorithm. Fig. 4a and Fig. 4b show changes of global parameters q and n_{ON}, respectively. One can see that a suboptimal solution characterized by $q = 0.78$ and $n_{ON} = 3$ was reached very quickly at the iteration 7 and finally an optimal solution characterized by $q = 0.94$ and $n_{ON} = 4$ was found at the iteration 169. This solution was expected from the analysis provided in Section 5. It is stable till the end of the iterated game, because it corresponds to a unique NE in this game.

Fig. 4c shows moments of time when some agents use ϵ alternative of the LA algorithm to select rules. The first time this alternative is used at iter=161 (see, the line in violet) by the agent 3 but it does not result in any changes in the system. The second time it is used at iter=169 (see, line in blue) by the agent 1 and it causes shifting a solution to the optimal one. Indeed, the agent 1 changed his rule *all D* used until now into the rule *all C* what resulted in changing the state of the sensor 1 from 0 to 1.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

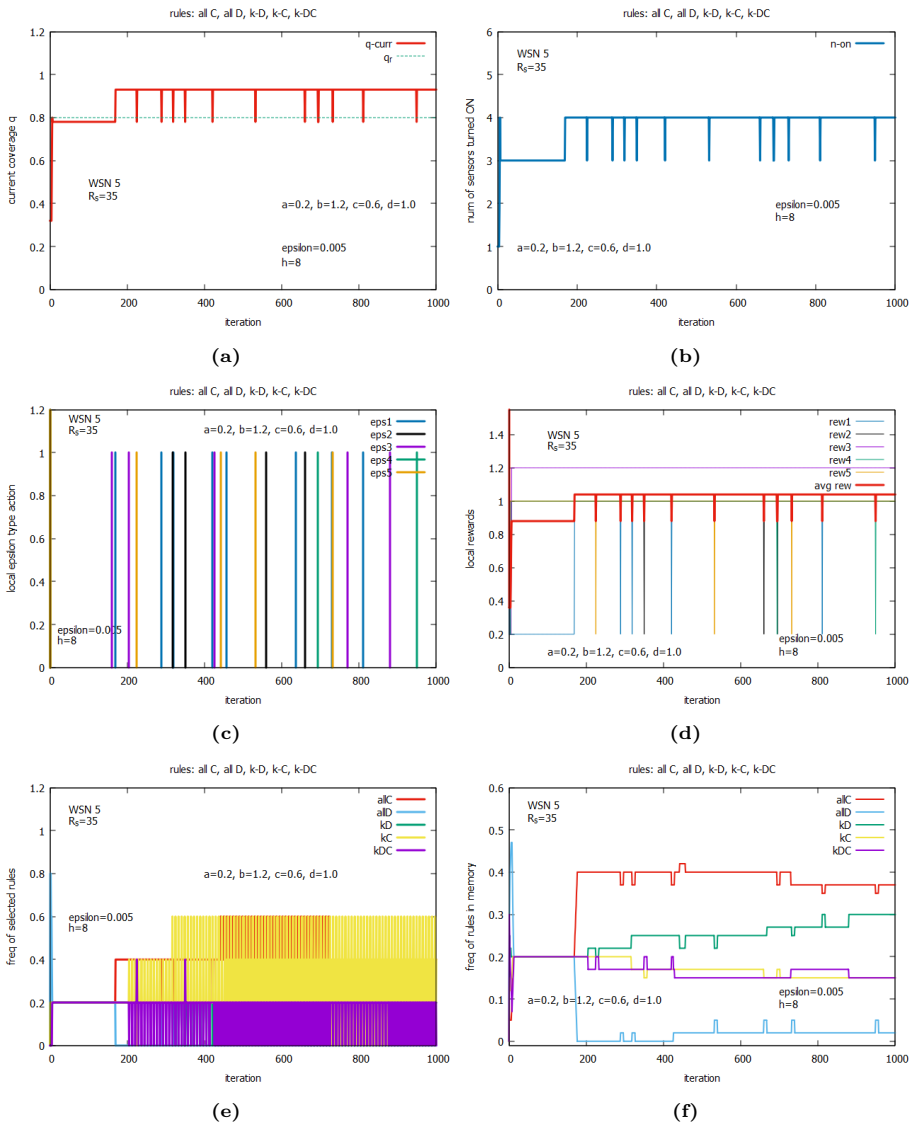
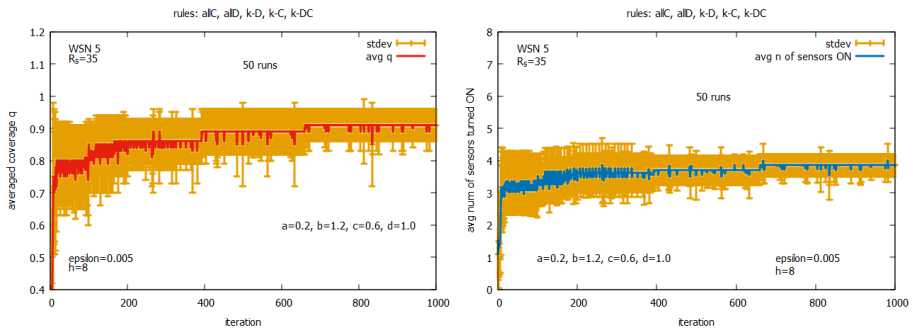


Fig. 4: Single run for WSN 5: (a) coverage q, (b) a number of sensors n_{ON}, (c) moments of taking actions caused by ϵ alternative of LA, (d) local rewards of agents, (e) frequencies of rules selected by agents, (f) structure of LA memories storing rules.

Fig. 4d shows how rewards of agents change in the game. One can see that before moving from the suboptimal solution to the optimal one the reward of the player 1 was equal to 0.2 (see, the line in blue), the reward of the player 3 was

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42



(a) **(b)**

Fig. 5: WSN 5: averaged value of (a) coverage q, (b) a number n_{ON} of sensors turned on.

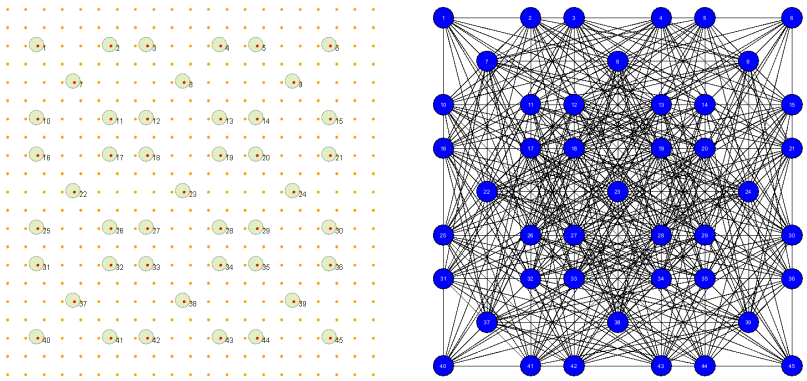
equal to 1.2 (see, the line in violet), and rewards of the remaining three players were equal to 1.0. Changing by the player 1 at iter=169 the state of his battery into 1, moved, as we already noticed, the suboptimal solution to the optimal one, and moved his personal reward to 1.0, while rewards of remaining players did not change. The optimal solution reached by the players is characterized by the highest average payoff (see, the line in red) of the game (ATP) equal to 1.04 corresponding to a specific NE called MPP. We can see that the solution is stable during the remaining rounds despite the attempts of changing the course of the game caused mostly by the ϵ alternative of LAs.

Figs. 4e,f give some insight into the process of managing rules of LAs which collectively influence on the global performance of the system. Fig. 4e shows how frequencies of rules selected by agents change in time. Till iter=169 all 5 rules are used with the same frequency equal to 0.2, but later we can observe a complex dynamics of different use of rules. Dominating rules are k–C (in orange) and *all* C (in red) which are used with frequencies from the range (0,0.6) and (0.2,0.6), respectively. The remaining three rules are used with the frequency from the range (0,0.2).

Fig. 4f gives some insight into the contents of LAs memories. One can see that while till the iter=169 all rules are equally distributed with the frequency equal to around 0.2, in further iterations we can observe the process of significant changes of LA memories structure. The rule *all* C becomes dominating occupying around 40% of the total LA memory, and *all* D occupies only around 5%. We can observe an increasing number of k–D rules reaching finally around 30% of the whole population of rules, and at the same time populations of k–C and k–DC are decreasing to around 15%.

Fig. 5 presents averaged over 50 runs results concerning of WSN 5. The requested coverage q_r is reached in the average after around 100 iterations (see, Fig. 5a) but the process of searching a solution is accompanied by noticeable

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42



(a) **(b)**

Fig. 6: WSN 45: (a) sensors localization, (b) WSN interaction graph for $R_s = 30$ m.

value of standard deviation (*std*). Improving the average values of q and n_{ON} (see, Fig. 5b)) can be observed in next iterations, with decreasing value of *std*.

6.2 The Instance: WSN 45

To verify a general performance of the proposed approach we use in this subsection a realistic instance called WSN 45 consisting of 45 sensors with $R_s = 30\ m$. Fig. 6a shows a deployment of sensors in the monitored area and Fig. 6b presents the corresponding WSN graph of the multi-agent system.

In the first set of experiments with WSN 45 we applied genetic algorithm (GA) (not shown here) to find an optimal or a suboptimal solution, and for this purpose the global function (see, Eq.(1)) was used. Conducted experiments indicated that there exists a solution providing the requested coverage $q \geq q_r$ with $n_{ON} = 4$, and it is used as a reference to results presented in this subsection.

Figs 7 and 8 show averaged over 30 runs values of q and n_{ON} for different subsets of rules. Figs 7a,b shows values of q and n_{ON} for the subset $\{all\ C, all\ D\}$ and Figs 7c,d for the whole set of 5 rules. For both versions the algorithm achieves the requested level of q_r. With the use of 2 rules goals of self-optimization are reached in the average at the 7-th iteration providing in the average $q = 0.83$ (see, Fig. 7a) with the average n_{ON} around 8.9 (see, Fig. 7b). The algorithm finds quickly a solution providing the average $q = 0.94$ which is maintained during the whole game close to this value. We can notice the decreasing number of sensors turned on, which reaches in the average $n_{ON} = 7.13$ in the final iterations.

The behavior of the system with the whole set of 5 rules is slightly different. The algorithm reaches at iter=14 the requested level of the coverage $q = 0.83$ (Fig. 7c) with low number of $n_{ON} = 6.16$ (Fig. 7d) and continues to improve the coverage until reaching $q = 0.94$ around iter=305 and maintains stable this value till the final iteration achieving n_{ON} equal around 7.0. The main difference

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

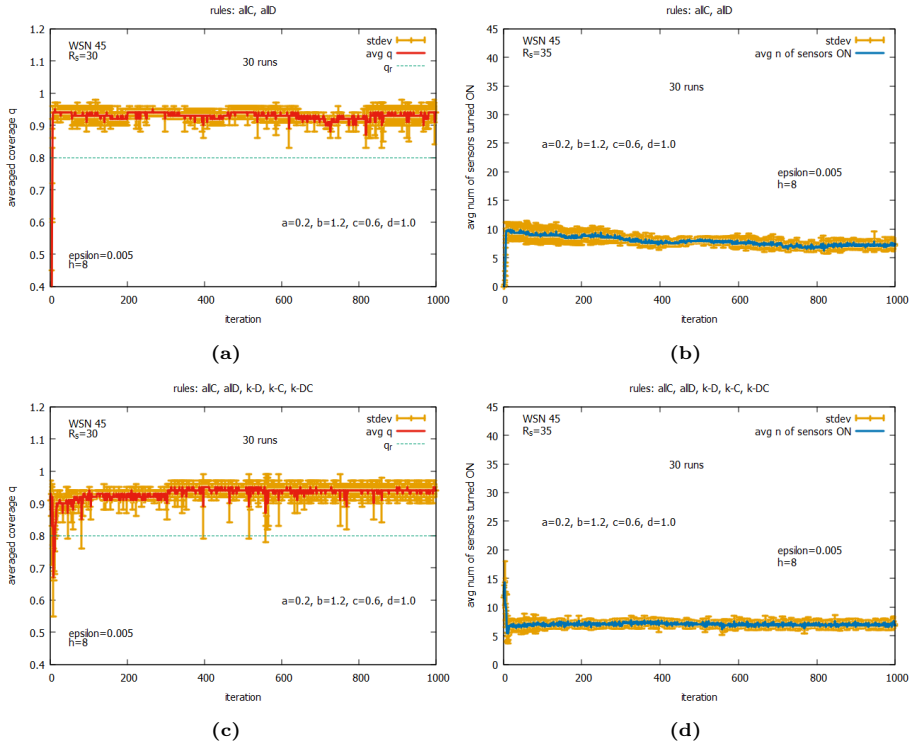


Fig. 7: WSN 45: averaged value of (a) coverage q for a subset of 2 rules, (b) a number of sensors n_{ON} for a subset of 2 rules, (c) q for the whole set of 5 rules, (d) n_{ON} for the whole set of 5 rules.

between both versions is that the algorithm with 2 rules requires at the beginning slightly more sensors turned on than the algorithm with the 5 rules, but finally both algorithms tend to reach a solution with the average n_{ON} close to around 7.00. Both versions are characterized by significantly reduced values of *std*.

Fig. 8 presents behavior of the system with other rules from the whole set of rules. Figs 8a,b show changes of the average values of q and n_{ON}, respectively for the rule *k–D*. One can see that this rule alone is able to provide a distributed search of a solution. However, it is accompanied by a relatively large *std* of q. At the iter=452 the average q becomes equal to 0.8 and it is accompanied by $n_{ON} = 4.33$, what means that current solutions are very close to one offered by GA. However, the solutions are not stable and they are lost. Only from the iter=663 the values of q fulfil the requirement of q_r with corresponding values of $n_{ON} = 6.93$. During the last 100 iteration a solution with low *std* of both q and n_{ON} is maintained with values $q = 0.92$ and $n_{ON} = 6.10$.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

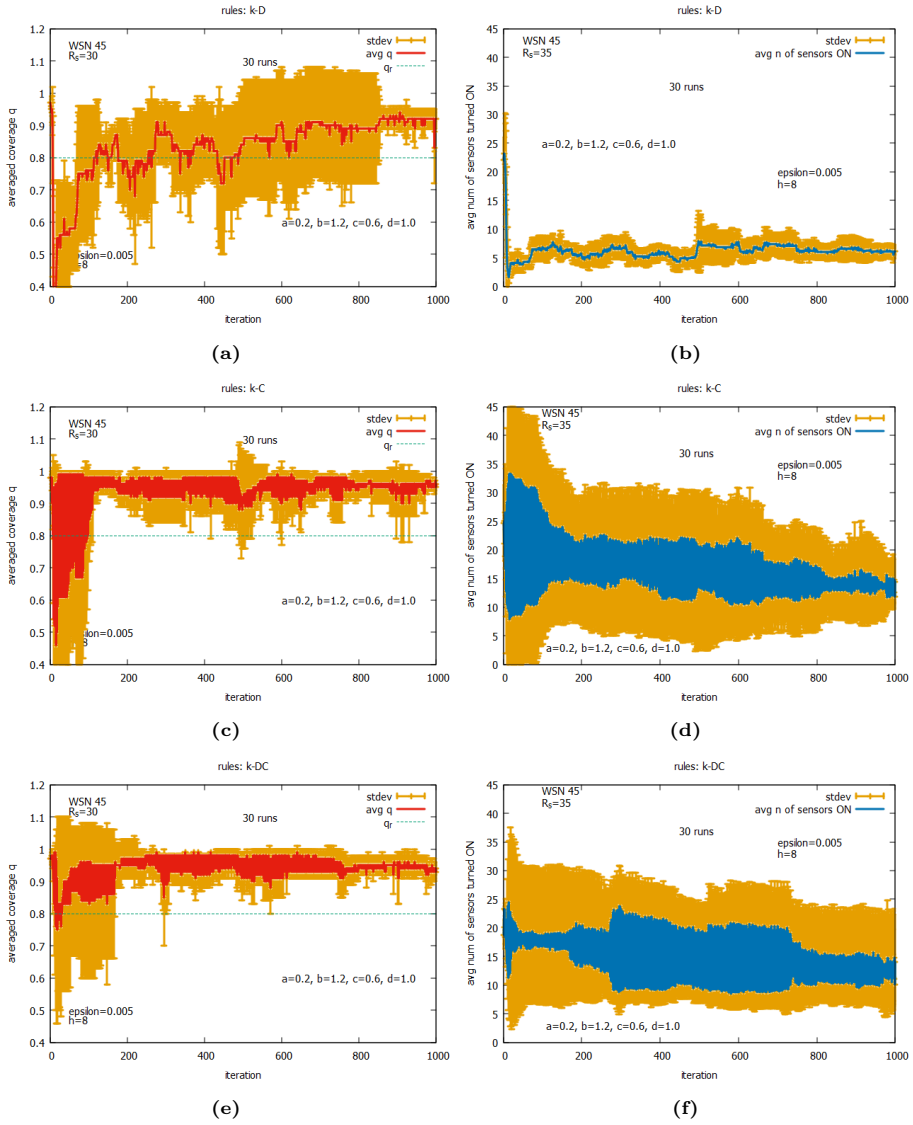


Fig. 8: WSN 45: averaged value of (a) q for rule k–D, (b) n_{ON} for rule k–D, (c) q for k–C, (d) n_{ON} for k–C, (c) q for k–DC, (d) n_{ON} for k–DC.

Figs 8c,d and Figs 8e,f show the behavior of the algorithm with rules k–C and k–DC, respectively. Both versions of the algorithm have self-organizing features. However, it is observed a relatively large number of sensors which are turned on in the beginning of a searching process. The convergence of both

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

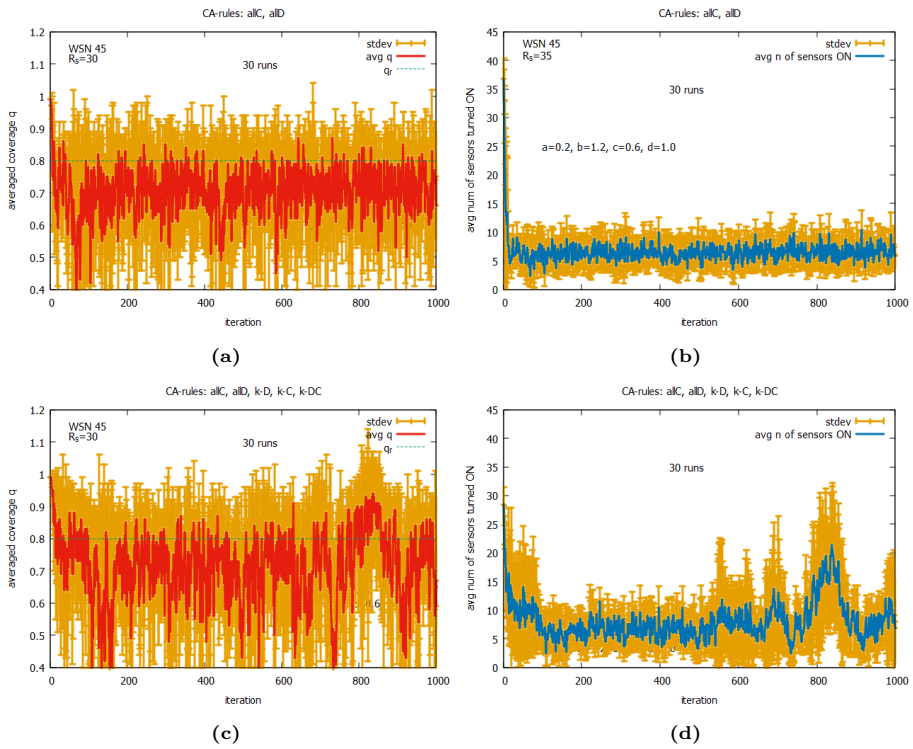


Fig. 9: WSN 45 (CA approach): averaged value of (a) q for 2 rules, (b) n_{ON} for 2 rules, (c) q for 5 rules, (d) n_{ON} 5 rules.

versions in terms of a number of active sensors with batteries turned on is slow. The algorithms are characterized by large values of *std*, and a phenomenon of oscillation of values q and n_{ON} can be observed. We can see it in Figs 8d,f, where values n_{ON} oscillate what results in large average values of n_{ON} (in blue).

6.3 The Instance: WSN 45 - Cellular Automata Approach

In [12] we proposed an approach to solve the coverage problem with the use of second-order CA. Fig. 9 presents simulation results for WSN 45 with the use of CA for two options: with the use of two rules (see, Figs 9a,b) and with the use of the whole set of 5 rules (see, Figs 9c,d).

Results show that for both options we observe a relatively large *std* of q. We can see also that the average values of q are mostly below the requested q_r. The option with the 2 rules is slightly better and more promising than the option with the 5 rules. It provides much more stable values of both q and n_{ON}. The comparison of both approaches based either on the use of LA-based agents or CA-based agents shows unambiguously that the LA-based approach significantly outperforms the CA-based approach.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42

7 Conclusions

We have proposed a novel self-organizing algorithm which is able to solve the problem of coverage in WSN in a fully distributed way by self-optimization. The approach uses three components: (a) a graph model of WSN, (b) a multi-agent system with agents corresponding to nodes of a WSN graph, and (c) agents implemented as (ϵ,h)-LA participating in the spatial PD game.

The system is able to self-optimize, i.e. agents act in such a way to maximize their own rewards but at the same time they reach unknown for them global solution specified by two objectives: a coverage and the corresponding number of sensors that are turned ON. The proposed LA-based approach significantly outperforms the approach based on using of CA-based agents.

Our future work will be oriented toward a more detailed study of the proposed approach with use of larger set of WSN instances to consider issues of scalability and the influence of specific model parameters on the algorithm's performance.

References

1. P. Berman *et al.*, Power efficient monitoring management in sensor networks. *2004 IEEE Wireless Comm. and Networking Conf.* (2004) 2329–2334
2. M. Cardei, D-Z. Du, Improving Wireless Sensor Network Lifetime through Power Aware Organization, *Wireless Networks* 11(3) (2005) 333–340
3. Manju, S. Chand, B. Kumar, Target coverage heuristic based on learning automata in wireless sensor networks, *IET Wireless Sensor Systems* 8 (2018) 109-115
4. J. Jaschke, Y. Cao, V. Kariwala, Self-optimizaing control - A survey, In *Annual Reviews in Control* 43 (2017) 199–223
5. Manju, S. Chand, B. Kumar, Genetic algorithm–based meta–heuristic for target coverage problem, *IET Wireless Sensor Systems* 8(4) (2018) 170–175
6. K.S. Narendra, M.A.L. Thathachar. *Learning Automata Theory. An Introduction*, Prentice Hall: Englewood Clifs, New Jersey (1989)
7. F. Ojeda *et al.*, On Wireless Sensor Network Models: A Cross-Layer Systematic Review, *Journal of Sensor and Actuator Networks* 12, 50 (2023)
8. P. Östberg, J. Byrne *et al.*. Reliable capacity provisioning for distributed cloud/edge/fog computing applications, In *European Conference on Networks and Communications (EuCNC 2017)* (2017) 1–6
9. B.J. Oommen, R.O. Omslandseter, L. Jiao. Learning automata-based partitioning algorithm for stochasting grouping problems with non-equal partition sizes, *Pattern Anal. App.* 26(2), (2023) 751–772
10. F. Seredyński. Competetive coevolutionary multi-agent systems: The application to mapping and scheduling problems, *J. of Paral. and Distr. Comp.* 47(1) (1997) 39–57
11. F. Seredyński, T. Kulpa, and R. Hoffmann, Evolutionary Self-optimization of Large CA-based Multi-Agent Systems, *J. of Computational Science* 68 (2023) 101994
12. F. Seredyński, T. Kulpa, R. Hoffmann, and D. Désérable, Coverage and Lifetime Optimization by Self-optimizing Sensor Networks, *Sensors* 23(8): 3930 (2023)
13. M.L. Tsetlin. *Automata Theory and Modeling of Biological Systems*, Elsevier: Amsterdam, The Netherlands (1973)
14. W.I. Warschawski. Collective Behavior of Automata, Nauka, Moscow (1973) (in Russian)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_42