Adaptive auction protocol for task assignment in wireless sensor and actuator networks

Lu Wang, Ming Li, Jiaxun Kou, Kun Yang and Chunlan Jiang

Abstract
The development of wireless sensor and actuator network is leading to high complexity networks and subsequently, new challenges in task assignment for effective sensor–actuator coordination. This article proposes an adaptive auction protocol for task assignment in multi-hop wireless actuator networks, considering a scenario where all actuators are immobile and each of them can obtain the target information from sensors. Unlike existing methods that neglect the adaptive auction area required by dynamic networks, the proposed method uses an adaptive factor $k_0$ (reflects the adaptive auction area) that is deduced based on the relation between network characteristics and protocol performance. Simulation results show that the adaptive protocol can dynamically change the auction area during the network operation. In addition, comparison with existing methods, such as simple auction protocol, 1 hop simple auction aggregation protocol, and 1 hop simple auction aggregation protocol greedy extension, shows that the adaptive auction protocol can choose the optimal solution at a low communication cost with a high probability, irrespective of the changes in the network.

Keywords
Adaptive auction, task assignment, distributed protocol, wireless sensor and actuator networks

Date received: 13 August 2019; accepted: 26 April 2020

Handling Editor: Mohammad Hossein Anisi

Introduction
Wireless sensor networks (WSNs) consist of hundreds of low-cost, low-power, and multi-functional micro-sensor nodes with sensing, processing, and wireless communication capabilities. The applications of WSNs include a diverse spectrum, for example, surveillance, tracking of military units, prediction of natural disasters, home automation, health care, and traffic control. A network consisting of sensors and actuators, known as wireless sensor and actuator network (WSAN) is regarded as an extension of WSN. In WSANs, actuators can interact with the physical world, make autonomous decisions, perform the corresponding actions based on data gathered by sensors, and shared information. However, the sensors are simple devices limited by battery power, processing power, memory, and communication constraints, while the actuators are devices with higher capabilities.

Task assignment in WSNs can increase the system efficiency by allocating tasks according to the ability of nodes. In this article, a task assignment algorithm is proposed to find an optimal actuator to respond to a target, such as humans, moving vehicles, and fire. The optimal actuator has the highest reward and minimal cost to respond to the target among all the actuators.

Corresponding author:
Chunlan Jiang, School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China.
Email: jiangchunwh@bit.edu.cn
Most existing solutions for task assignment are centralized solutions with a direct or a multi-hop communication topology. The advantage of centralized solutions is that a central node collects and compares the whole network data, and then an optimal actuator can be found with certainty. However, the presence of a large number of actuators and cooperation between them cause significant delay and energy consumption and lead to faults if the topology is incomplete. On the contrary, a localized solution that uses locally available information can select an optimal actuator even with incomplete topology. The advantage of a localized solution is good scalability and fault tolerance. In addition, although the selected actuator may not be the optimal one in the whole network region, it is normally close to the optimal one.

Auction protocol is one of the localized solutions for solving task allocation problems in multi-actuator systems. This protocol is flooding based and each actuator in the auction retransmits auction request and responds to the auctioneer. Subsequently, the auctioneer selects the optimal actuator with the highest reward which means high efficiency and low-energy cost. The issue of cooperation between actuators has been studied in Mezei et al., Fatemi et al., Kim and Matson, Geyik et al., and Panchu et al. These literatures use appropriate mechanisms to find the optimal actuator in a particular application scenario with fixed network parameters. However, they ignore that the network parameter is time-varying in some scenarios. Through these references, it can be found that the auction area not only determines the energy consumption and time delay but also influences the probability of selecting the optimal actuator. Normally, a specific network needs a corresponding k (the number of hops of the auction), which represents the auction area needed to achieve the optimal protocol performance. In some military scenarios, the actuators are vulnerable and easily lose efficiency or the antenna power needs to change with the actuator density to maintain a balanced coverage. Therefore, it is necessary to adjust the value of k to compensate for these changes in the network.

In this article, our contributions can be outlined as follows:

- For the changeable network, we propose a localized adaptive solution called adaptive auction protocol (AAP), which can dynamically adjust the auction area, so that the protocol can select an actuator with the highest reward to take action on a target according to different network parameters.
- Through simulations, we summarize the relations between the factor k and each parameter affecting the performance of the AAP and integrate these parameters into an approximate formula to calculate the appropriate auction area.

- We analyze the AAP with three representative algorithms: simple auction protocol (SAP), 1 hop simple auction aggregation protocol (1-SAAP), and 1 hop simple auction aggregation protocol greedy extension (1-SAAPG). The simulation results show that the adaptive protocol is more flexible and stable, because it can change the auction area with the changing network and can always select the optimal actuator having relatively low communication cost (the number of communications) with a high probability.

This article is organized as follows. Section “Related works” describes related work and section “Network model” introduces the network model and the task sequence model. Section “AAP” presents the proposed algorithm and the three phases of AAP. In section “Performance evaluation,” a formula of the adaptive parameter k is deduced and the performance of AAP and its comparison with existing methods are discussed. Finally, the conclusion is presented in section “Conclusion.”

Related works

Multi-actuator task allocation

Various task assignment algorithms have been proposed for multi-robot systems, WSNs, unmanned aerial systems (UAS), and cooperation of multi-actuators. These algorithms can be classified into three types: centralized, decentralized, and hybrid. In the centralized algorithms, a central node is responsible for global data collection and processing. In the decentralized algorithms, each node is assigned tasks based on its neighbors’ data. In the hybrid algorithms, the network is divided into multiple clusters managed by a cluster header, but the cooperation between clusters is decentralized.

In Okhovvat et al., the authors consider both energy savings and time constraints of WSANs and propose two centralized algorithms for task allocation based on queuing networks. However, the algorithms only consider the load of the actuator, and its adaptability to perform a task is not taken into account. In Liwei et al., a centralized algorithm is proposed that considers cross-entropy (CE) and the required resources for task allocation for unmanned aerial vehicles (UAVs). The algorithm is extended for solving complicated combinatorial optimization problems. Sample selection and adaptive parameter settings can further
optimize this algorithm; however, a few issues remain unresolved.

Capitan et al.\textsuperscript{19} propose decentralized algorithms in multi-robot system. These algorithms use partially observable Markov decision processes (POMDPs) to lower the dependence between individual decisions and hence, lower the complexity. Two application examples show flexibility and robust coordination of the algorithms.

In Coltin and Veloso,\textsuperscript{20} three algorithms for assigning mobile robots to events in hybrid WSNs are discussed: a centralized algorithm, an auction-based algorithm, and a distributed approach that uses a spanning tree of the network. The results show that the tree-based algorithm has a low communication cost and a high expected network lifetime at the expense of assignment quality. However, when the scale of the sensors network increases, this algorithm suffers from significant delays.

Mezei et al.\textsuperscript{15} propose five different auction aggregation protocols for task assignment in multi-hop wireless robot networks. A robot collector leads an auction and initiates a tree network by transmitting the search message. Each robot decides whether to retransmit a search message up to k-hops away. Then, the robots aggregate the best bid responses to the robot collector. However, to reduce the size of the aggregation tree, the protocol has to provide positions, costs, and availabilities of all the neighbors of each robot, which requires additional communication resources.

In Feng et al.,\textsuperscript{21} a hybrid task algorithm based on score incentive mechanism (TASIM) for complex task execution in WSNs is proposed. Sensor nodes are ranked based on their residual resources and services capacities. Simulations show that the algorithm narrows the area of working nodes and balances the energy consumption. However, as the task allocation on the cluster head is centralized, it will quickly deplete its energy. Although the task migration in TASIM mitigates the problem of excessive energy consumption by the cluster head, it is still not completely solved.

**Auction-based algorithms**

Auction algorithms have been widely used for optimal task allocation in various fields such as multi-robot systems, sensor networks, and WSANs. In Lee et al.,\textsuperscript{22} a resource-oriented decentralized auction algorithm (RODAA) for multi-robot task allocation is proposed. The energy that a robot consumes while performing tasks is considered as a resource. An application for cleaning a solar panel based on the multi-robot system is proposed. Simulation results show that the algorithm has lower overall resource consumption compared with other auction-based task allocation algorithms. However, each auction needs to broadcast to the entire network. This kind of auction stil uses significant energy resources. A partial auction algorithm can be used to economize energy.

Lukic et al.\textsuperscript{23} describe two distributed dispatch algorithms for the task assignment problem in wireless sensor and robot networks. Auction is used in these algorithms to make better decisions. Simulations show that the proposed algorithms extend the system lifetime and decrease the communication cost.

Choi et al.\textsuperscript{24} present a consensus-based auction algorithm. It utilizes a decentralized auction method that has no auctioneer for conflict resolution. Each agent places a bid on each task and stores winning bids throughout the assignment process. Results show that this algorithm guarantees 50% optimal solutions of the time.

In Zheng et al.,\textsuperscript{1} an auction-based adaptive sensor activation algorithm for target tracking in the WSNs is proposed. The algorithm is used to choose cluster members in a predicted region. Simulation results show good performance in terms of quality of tracking, energy efficiency, and network lifetime.

In Mezei et al.,\textsuperscript{10} auctions are used to improve energy efficiency and the lifetime of the network. The localized auction aggregation protocol (k-SAAP) is used to improve the basic information mesh protocol. Simulation results show that the proposed algorithm guarantees 100% efficiency for finding the closest actuator in some scenarios.

In Mezei et al.\textsuperscript{16} a new algorithm is proposed that improves the auction aggregation protocols proposed in Mezei et al.\textsuperscript{15} Based on initial task assignment, the assigned actuator initiates another 1-hop auction to explore if any of its neighbors have a better reward. Simulation results show that the improved algorithm has acceptable additional communication cost, especially in networks with low values of k hop.

**Network model**

In this article, we consider a military application to evaluate the performance of the proposed protocol. The model of a military system is shown in Figure 1. The system is composed of a large number of sensor nodes and several actuators. The sensors can transmit the perceived target information to their nearby actuators. The actuator is a type of weapon with a certain radius of attack, referred to as the execution radius in the figure. An actuator can know the target location accurately based on the sensor reports. The actuator acts if a reported target appears within its execution radius.

The graph model for the network is a unit disk graph (UDG) model. All actuators have the same communication radius $R_c$ and the same execution radius $R_e$. The
communication protocol is based on flooding. Actuators 2, 3, and 4 can directly communicate with actuator 1, and the remaining actuators need to communicate with actuator 1 through multiple hops. Moreover, we assume that each actuator can connect to all the actuators through multiple hops. Note that the sensors used to connect the actuators are not considered in this article. We also assume that the actuators are placed at fixed locations and can execute only once. This is a realistic assumption, because the ammunition in an automatic weapon cannot be replenished once it is empty. Similarly, the batteries of actuators are neither rechargeable nor replaceable. When an actuator runs out of the battery, it will lose its utility and new actuators will be deployed in the area as its replacements. In addition, all actuators can know their own locations, and each actuator can also obtain locations of other actuators through multi-hop connections.

Important notations used in this article are given in Table 1.

### Table 1. Notations.

| Terms | Description |
|-------|-------------|
| $R_c$ | The radius of communication |
| $R_e$ | The radius of execution |
| $k$  | The maximum hop count of the auction |
| $k_o$ | The optimal $k$ in the current state of the network |
| $k_{current}$ | The current hop of the packet |
| $D_{target}$ | The distance between actuator and target |
| $E$ | The residual energy of an actuator |
| $N$ | The number of actuators in the network |
| $N_{nei}$ | The number of neighboring actuators |
| $B_{value}$ | The value of bid |
| $\Delta$ | A prerequisite for participating in an auction |
| $\alpha$ | The coefficient in $k_o$ |
| $T_{hop}$ | The auction delay on each hop |
| $T_{max}$ | The specified maximum broadcast delay on each hop |
| $T_{spread}$ | The communication latency time per hop |
| $T_{process}$ | The processing time of an actuator |
| $P$ | The probability that the optimal actuator can be selected |

**Auction generation.** In auction generation, there are two main preconditions: First, a target appears in the network and actuators get its location via the sensors. Second, each actuator knows its own location before a target appears. Based on these two preconditions, each actuator knows the location of other actuators through multi-hop connection, and each actuator can judge whether it is closest to the target or not.

**Step 1.** The actuator that knows the target’s location and is closest to it starts an auction.

When a target appears, the actuator closest to it generates auction and transmits an auction packet to other actuators located within $k$-hops. An illustration is shown in Figure 3(a). Actuator A, being closest to the target, generates an auction to its neighboring actuators B, C, and D. Normally, the optimal actuator is close to the target in most cases. Therefore, by selecting the actuator closest to the target helps in finding the optimal actuator within a small area around the selected actuator. The generation process is shown in Figure 3(b).

An auction contains only one target. If more than one target appears, all targets will be dealt with one by one with the short duration auction.

**Step 2.** When an actuator becomes an auctioneer, it will generate an auction packet and broadcast it to other actuators.

The auction packet contains the following fields, as shown in Figure 4: $k$ refers to the maximum hop count...
in the auction, $k_{current}$ refers to the current hop in which the packet arrives, Route is used to record the routing path, and $B_{value}$ refers to the value of bid. When an auction is generated, $B_{value}$ refers to the bid of the auctioneer.

The selection of an optimal $k$, denoted as $k_0$, is described in section “Selection of $k_0$.” As existing references rarely give details about evaluation of the bid, we propose a method to calculate the bid value according to our algorithm requirements. The $B_{value}$ is calculated according to the following factors: the distance $D_{target}$ between the actuator and target, the execution radius $R_e$ of the actuator, the number of neighbors $N_{nei}$ of the actuator, and the percentage of remaining energy $E$ of the actuator.

As the actuator can only perform a task once, the actuator with lowest power is given priority to perform a task, if other parameters of different actuators are the same. Similarly, the actuator that is the closest to the target should be given priority, because such an actuator can execute a task better. The actuator that has a high number of neighboring actuators should be given a lower priority, that is, an actuator with no neighbors is the first one to execute a task, because it causes the least damage to the network connectivity. Based on these relations, we define $B_{value}$ as

$$B_{value} = \frac{1}{\Delta \times (N_{nei} + 1) \times E}$$

where

$$\Delta = \frac{D_{target}}{R_e}$$

The number of neighboring actuators is $N_{nei}$, $E$ is the residual energy of actuator, $D_{target}$ is the distance between actuator and target, and $R_e$ is the radius of execution. If $\Delta > 1$, we define $B_{value} = 0$, which means that the target is out of the actuator’s execution range, and hence the actuator cannot reach the target.

**The spread of auction.** After an auction is generated, the auction packet will be spread between the actuators. There are two types of auction spread, forward and inverted. The spread process is shown in Figure 5 and an illustration is shown in Figure 6.
As shown in Figure 6, actuator A is the auctioneer, and actuators B, C, and D are its children located at \(k_{\text{current}} = 1\) hop. Based on flooding protocol, actuators E, F, G, H, and I located at \(k_{\text{current}} = 2\) hop will receive the auction packet from actuators B, C, and D. If \(k = 2\), in the forward spread, the auction packet will be generated at actuator A and transmitted first to actuators B, C, and D, and finally transmitted to actuators E, F, G, H, and I. The routes are shown by the dotted lines. In the inverted spread, the route of the auction packet is reversed.

**Step 3.** In the forward spread, the packet will be continuously transmitted to the next hop until \(k_{\text{current}}\) equals \(k\). Each actuator receives the same auction only once to avoid multiple reception.

The packet records the current hop \(k_{\text{current}}\) that denotes the hop number that it has reached. The \(k_{\text{current}}\) will be increased by one when the packet reaches a new actuator. The auction delay \(T_{\text{hop}}\) on each hop is defined as follows

\[
T_{\text{hop}} = T_{\text{spread}} + T_{\text{process}}
\]

where \(T_{\text{spread}}\) and \(T_{\text{process}}\) denote the spread latency time on each hop and the processing time of an actuator, respectively. Meanwhile, considering the delay caused by channel collision, the maximum delay between two actuators is defined as \(T_{\text{max}}\), where \(T_{\text{max}} \geq T_{\text{hop}}\).

**Step 4.** In the forward spread, the actuator saves the value of the bid \(B_{\text{value}}\) in the received packet called \(B_{\text{value}} - \text{parent}\). If the \(B_{\text{value}}\) generated by the actuator receiving the packet called \(B_{\text{value}} - \text{local}\) is larger than \(B_{\text{value}} - \text{parent}\), \(B_{\text{value}} - \text{parent}\) is replaced by \(B_{\text{value}} - \text{local}\). Subsequently, the actuator continues to spread the packet and starts to collect its children’s \(B_{\text{value}}\) called \(B_{\text{value}} - \text{child}\).

When \(k = k_{\text{current}}\), the actuator does not spread the receiving packet any further. It sends back its \(B_{\text{value}}\) to its parent, that is, the actuator from which it received the packet, only if its \(B_{\text{value}}\) is higher than the \(B_{\text{value}}\) from the received packet.

**Step 5.** In the inverted spread, each actuator sends a \(B_{\text{value}}\) to its parent. This value is selected as the largest \(B_{\text{value}}\) from among its own value (\(B_{\text{value}} - \text{local}\)) and the values from its children (\(B_{\text{value}} - \text{child}\)). The largest \(B_{\text{value}}\) is not sent if it is smaller than or equal to the \(B_{\text{value}}\), it is saved.

In this way, the communication cost can be minimized, and each actuator can select the largest \(B_{\text{value}}\) in a time of \(2 \times (k - k_{\text{current}}) \times T_{\text{max}}\). Moreover, the largest \(B_{\text{value}}\) can be selected by an auctioneer in a time of \(2 \times k \times T_{\text{max}}\).

**Authorization and execution**

**Step 6.** At the end of \(2 \times k \times T_{\text{max}}\) time period, the auctioneer obtains the largest \(B_{\text{value}}\). The auctioneer sends an authorization message to the winner, that
is, the actuator with the largest \( B_{\text{value}} \) and the auction ends.

The Route field in the message is recorded with the \( B_{\text{value}} \). When the winner receives the authorization message, it broadcasts the message notifying the whole network that it will perform the final task. The message is broadcasted by flooding, so all the actuators can update the information of surviving actuators and get ready for the next auction.

**Step 7.** The winner floods the whole network with the information that it has received the authorization message and then executes the attack task.

The authorization process is shown in Figure 7 and an illustration is shown in Figure 8. As shown in Figure 8, actuator A finds from the received bids that actuator H is the optimal actuator to perform the attack. Then, it sends the authorization message to the winner along the routes shown by the dotted lines.

**Selection of \( k_0 \)**

From section “Auction protocol process,” it can be observed that \( k \) represents the auction area and the duration of auction is primarily determined by \( k \). The larger the \( k \), the longer the duration. In addition, a larger \( k \) facilitates finding the optimal actuator. However, a smaller \( k \) leads to a quick auction, as there is communication with only a few actuators, which hinders finding the optimal actuator. In subsection “Impact of auction hop count \( k \),” we mainly discuss the impact of \( k \) on the auction process and show the necessity of selecting an adaptive \( k \). In subsections “Impact of density and \( k \),” “Impact of \( R_c \) and \( k \),” and “Relation between \( R_c \) and \( k \),” through simulations, we use a series of curves to show the relation between the characteristics of network and \( k \). Finally, a general formula of \( k_0 \), which denotes the optimal \( k \) is presented in subsection “Calculation of \( k_0 \).”

We use the OMNet++ simulation tool. In the simulation, we consider a real military system scenario, and all simulation parameters can be achieved by existing hardware. All actuators are uniformly distributed in an area and a target can appear anywhere in the coverage area. The residual power of the actuators is distributed uniformly. Other network attributes are listed in each subsection.

Two metrics are used to evaluate the performance with respect to \( k \): (1) number of communications that represents the communication cost in the auction and (2) optimal probability \( P \) which represents the probability of selection of the optimal actuator by the auction. The optimal probability \( P \) is calculated as the ratio \( N_{so}/N_s \), where \( N_s \) is the total number of simulations and \( N_{so} \) is the number of simulations in which the optimal actuator is obtained by auction.

**Impact of auction hop count \( k \).** In the simulations of this subsection, the simulation area is a two-dimensional (2D) space square with dimensions \( 500 \times 500 \) m\(^2\). The radii of actuator communication \( R_c \) and actuator execution capability \( R_e \) are set to 60 and 200 m, respectively. The number of actuators \( N \) in the area is 150. The target is randomly deployed in the area. The values of \( k \) are changed from one to seven. For each \( k \), simulations are generated 50 times, and optimal probabilities are calculated for all the simulations and averaged.

The results are shown in Figure 9. When \( k \leq 4 \), the optimal probability increases with \( k \) and then becomes constant at a value of 86%. However, the number of communications increases almost linearly with the increase of \( k \). We can note that there is an optimal \( k \) that is suitable for this network, that is, \( k = 4 \). With the optimal \( k \), the auction can reach the maximum optimal probability at a relatively low consumption cost.

**Impact of density and \( k \).** The relationship between the actuator density and \( k \) is explored in this subsection. The deployment area is again set to \( 500 \times 500 \) m\(^2\),
and \( R_c \) and \( R_e \) are equal to 100 and 200 m, respectively. The values of \( N \) are varied from 50 to 300, with a step-size of 50. The factor \( k \) is changed from one to four, and for each value of \( N \), simulations are generated 50 times. For each combination of \( k \) and \( N \), optimal probabilities are averaged over all the simulations.

The results in Figure 10 show that a small \( k \) can lead to a high optimal probability when the actuator density of the network is high. When the density is low, a large value of \( k \) is needed to attain a high optimal probability. In other words, \( k \) is inversely correlated with the density and can be adaptively selected based on this relation.

Impact of \( R_e \) and \( k \). In this subsection, the simulation area is set to 500 \( \times \) 500 m\(^2\), \( R_c \) is set to 80 m, \( N \) is equal to 150, and \( R_e \) is changed from 100 to 300 m in a step-size of 50. The value of \( k \) is varied from one to four, and simulations are generated 50 times for each \( R_e \). For each combination of \( k \) and \( R_e \), optimal probabilities are calculated and averaged over all the simulations.

Figure 11 shows that the optimal probability decreases slightly with the increase in \( R_e \). However, the optimal probability is significantly affected by the \( k \): A higher \( k \) leads to a higher optimal probability and vice versa. Therefore, selection of \( k \) can be carried out irrespective of \( R_e \).

Relation between \( R_c \) and \( k \). In the following simulations, the deployment area is set to 500 \( \times \) 500 m\(^2\), \( R_c \) is set to 200 m, \( N \) is 150, and \( R_c \) is varied from 60 to 200 m in a step-size of 20 m. The value of \( k \) is varied from one to five, and 50 simulations are generated for each value of \( R_c \). For each combination of \( k \) and \( R_c \), optimal probabilities are calculated and averaged over all the simulations.

Figure 12 shows that the optimal probability increases with the communication radius. For a small \( k \), a large communication radius is needed to achieve a high optimal probability, and vice versa. Therefore, it can be concluded that \( k \) is inversely correlated with the communication radius and the selection of adaptive \( k \) can be carried out considering this relation.

Calculation of \( k_0 \). \( k_0 \) facilitates effective auction performance. Its value is a constant that is calculated based on the state of network in each auction. A performance can be described as good if the auction has a high probability to select the optimal actuator with low communication cost. We define

\[
k_0 = \min\{k | P \in [0.95 \times P_{\text{max}}, P_{\text{max}}]\}.
\]

The above equation means that the optimal value \( k_0 \) is the minimum value of \( k \), for which the optimal probability is between 95% and 100% of the maximum optimal probability.

Based on the simulations with different values of \( k \), we can obtain values of \( k_0 \), which are the values of \( k \) corresponding to the shaded data in Table 2. However, in real scenarios, the auction needs a general formula of \( k_0 \) to achieve the best performance according to the changing parameters of network.

Based on the simulations described in the previous subsections, we can conclude that a changing \( R_e \) does not affect the choice of \( k \), and actuator density and \( R_c \) are inversely correlated with the selection of \( k \). As \( k_0 \) is an integer and the sample data are limited, equation (4) is an approximate solution. A normalization method is
adopted for the calculation of $k_\alpha$, as it is unitless and
the variables used in the calculation have different
units.

Based on multivariable non-linear curve fitting
method, we devise a formula of $k_\alpha$ based on fitting a
curve to the simulated curves in subsections “Impact of
density and $k$,” “Impact of $R_c$ and $k$,” and “Relation
between $R_c$ and $k$,” as follows

$$k_\alpha = \left\lfloor \frac{\alpha}{\sqrt{\rho \times R_c^2}} \right\rfloor \quad (5)$$

where $\alpha$ is a constant used to equalize the equation and \( \lfloor \cdot \rfloor \) is a function that rounds up the input value to the
nearest integer. The actuator density $\rho$ is calculated as follows

$$\rho = \frac{N}{S} \quad (6)$$

where $N$ is the total number of actuators and $S$ is the
area of actuator distribution.

For further determining the value of $\alpha$, part of the
simulation results showing the probabilities $P$ of opti-
mal task allocation versus different values of $k$ and
radius of communication are shown in Table 2. The
probabilities corresponding to the optimal $k$ are
shaded. The maximum probability is not 100% for
small values of radius, because the network connectiv-
ity is blocked.

As $\alpha/\sqrt{\rho \times R_c^2}$ is rounded up to the nearest integer,
when $k$ is 3, $\alpha/\sqrt{\rho \times R_c^2} \in [2, 3)$. Then, the range of $\alpha$
can be inferred, because $\rho$ and $R_c$ are known.
For example, for probability value of 85% shown in
Table 2, it can be calculated that $\alpha \in [2.94, 4.41)$. Similarly, based on the constraints given by the simula-
tion results, it can be figured out that $\alpha = 2.94$.

**Performance evaluation**

In this section, we evaluate the performance of the pro-
posed protocol in two scenarios: (1) We simulate the
scenario of a target appearing randomly and actuators
being consumed constantly. The AAP is compared with

**Table 2.** The probability of optimal task allocation for different values of $R_c$ (the area is $500 \times 500$ m², $R_e$ is 200 m, and $N$ is 150).

| P | $R_c = 40$ m | $R_c = 60$ m | $R_c = 80$ m | $R_c = 100$ m | $R_c = 120$ m | $R_c = 140$ m | $R_c = 160$ m | $R_c = 180$ m | $R_c = 200$ m |
|---|---|---|---|---|---|---|---|---|---|
| $k = 1$ (%) | 50 | 64 | 88 | 94 | 99 | 100 | 100 | 100 | 100 |
| $k = 2$ (%) | 54 | 79 | 96 | 96 | 100 | 100 | 100 | 100 | 100 |
| $k = 3$ (%) | 58 | 85 | 98 | 100 | 100 | 100 | 100 | 100 | 100 |
| $k = 4$ (%) | 58 | 88 | 99 | 100 | 100 | 100 | 100 | 100 | 100 |
| $k = 5$ (%) | 60 | 89 | 99 | 100 | 100 | 100 | 100 | 100 | 100 |
| $k = 6$ (%) | 61 | 89 | 99 | 100 | 100 | 100 | 100 | 100 | 100 |
| $k = 7$ (%) | 61 | 89 | 99 | 100 | 100 | 100 | 100 | 100 | 100 |
1-SAAPG, SAP, and 1-SAAP under this scenario. (2) As the antenna power is variable in some applications, we analyze the performance of AAP in networks with different communication radii and again compare it with SAP, 1-SAAP, and 1-SAAPG. We consider two scenarios.

**Fixed communication radius**

In the first scenario, the 2D space in the simulation is a square of dimensions $500 \times 500$ m$^2$. Different targets randomly appear in this space for a total of 340 times one by one. There are 350 actuators, execution radius is 200 m, and communication radius is 100 m. After an auction of a target, the selected actuator will attack the target and become unavailable. Simulations are generated 10 times and results are averaged over all the simulations. We measure the value of $k$, the number of non-optimal assignments, the total number of communication cost, and the average communication cost per actuator.

The number of non-optimal assignments represents the number of auctions that do not select the optimal actuator. The total number of communication cost represents the cumulative number of communications between actuators in 340 auctions after target appears 340 times. The average communication cost per actuator equals the number of communications cost per auction divided by the number of surviving actuators.

It can be seen in Figure 13 that $k$ changes during the auction. A target appearance means that one of the actuators is used, and hence the network density will decrease with the appearance of targets. The decrease of density will lead to a higher number of hops, as the actuators will be far from one another. The change of $k$ demonstrates that the AAP area can vary during the simulation. The value of $k$ keeps on increasing as long as the network remains in operation.

Figure 14 presents a comparison between the proposed and existing techniques, in terms of the cumulative number of non-optimal assignments versus the target number. The results show that the 1-SAAP has the most non-optimal assignments and the SAP has the least non-optimal assignments. The reason for this behavior is that the SAP attempts to use information of the whole network, whereas the 1-SAAP only uses the 1-hop neighbor information. In the 1-SAAPG, the 1-hop neighbor information is used and the search for a better assignment is carried out in the next 1-hop. Therefore, the number of non-optimal assignments in 1-SAAPG is larger than that in the SAP and smaller than that in the 1-SAAP. However, the AAP uses the values of $k$ from one to five and obtains results similar to the 1-SAAPG.

Figure 15 shows the average communication cost per actuator averaged over every ten auctions. As the SAP uses the whole network information, the actuators communicate twice per auction. After the appearance of several targets, many actuators are consumed. As a result, the actuator density decreases and the network cannot be fully connected, which results in some actuators being unable to participate in the auction. As the 1-SAAPG always uses one more hop information than the 1-SAAP, the average communication cost for the former 1-SAAPG is always larger than that for the latter. The AAP uses changeable $k$ and, therefore, the average communication cost is initially similar to that of the 1-SAAP. Subsequently, the cost becomes similar to the cost of the 1-SAAPG, and finally it increases and is in between the cost of the 1-SAAPG and SAP.

Figure 16 shows the total communication cost after the appearance of 340 targets. The SAP has the maximum cost and the 1-SAAP has the minimum cost. The communication cost in AAP is between those of the 1-SAAP and 1-SAAPG. This result is consistent with the average communication cost, as shown in Figure 15. Comparing Figures 14 and 16, we can observe that the SAP gives the least number of non-optimal assignments with very high communication cost. The 1-SAAP has the least communication cost and the highest number of non-optimal assignments, which is at least 14 averaged points larger than the other algorithms. Actually, optimal assignment and communication cost are both important in the auction. The 1-SAAP and SAP show that considering only one indicator leads to a sharp decline in other indicators’ performance. However, the communication cost in the AAP is three times larger than that in the 1-SAAP, and the AAP obtains 14 times decrease in the number of non-optimal assignments. In contrast, the SAP has ten times less non-optimal...
assignments, but the communication cost in SAP is over 27 times larger than that in the AAP. Although the AAP has a higher communication cost than the 1-SAAP, it is acceptable due to an increase in the number of optimal assignments. Moreover, the number of non-optimal assignments for both AAP and 1-SAAPG are similar, but the communication cost of the former is nearly half the latter’s communication cost. To sum up, thanks to the optimal factor $k_0$ that helps the network use the adaptive auction area, the AAP has a relatively high probability of obtaining the optimal assignment with acceptable communication cost.

Variable communication radius

In the second scenario, we evaluate AAP in the networks with different communication radii. The communication radius $R_c$ is varied from 40 to 240 m in a step of 20 m. The execution radius is 200 m and the area is a square of dimensions $500 \times 500$ m$^2$. The initial number of actuators is 200. A total of 100 targets appear randomly one by one. After all the targets have appeared, we measure the number of non-optimal assignments and the total communication cost. Simulations are generated 10 times for each $R_c$ and simulation results are averaged over all the simulations.

Figure 17 shows that the SAP has the lowest number of non-optimal assignments and the 1-SAAP has the highest number of non-optimal assignments for all values of $R_c$. When $R_c < 100$ m, the 1-SAAPG and AAP have similar number of non-optimal assignments. When $R_c > 100$ m, all the four algorithms perform similarly. When $R_c > 120$ m, the 1-SAAP is similar to the AAP, and the SAP is similar to the 1-SAAPG. Note that the number of non-optimal assignment is less than five for all the algorithms during this interval, which indicates a good performance.
However, taking the communication cost shown in Figure 18 into consideration, the AAP and 1-SAAP have similar performance as the SAP and 1-SAAPG in terms of number of non-optimal assignments, at a significantly lower communication cost. Therefore, we can conclude that the AAP and 1-SAAP are better choices when $R_c > 120$ m. When $R_c < 100$ m, we can find that the communication cost is directly proportional to the optimal assignment. The SAP has the highest communication cost with the highest number of optimal assignments. The 1-SAAP has the minimum communication cost with the lowest number of optimal assignments. If a high number of optimal assignments is required, the SAP is the better choice. If the communication cost is more important, the 1-SAAP is the better choice. If both optimal assignment and communication are equally important, for $R_c < 110$ m, the AAP and 1-SAAPG are better choices. For $R_c > 140$ m, the AAP and 1-SAAP are better choices. Therefore, for $R_c$ varying in a large range, the AAP can perform effectively when both optimal assignment and communication are to be optimized.

**Conclusion**

This article proposed a task assignment algorithm for a WSAN model in a military scenario. The AAP can be used to select an optimal actuator and assign the tasks in an adaptive auction area. We developed a formula for the adaptive factor used in the AAP. Through simulations, the relations between different characteristics of the WSAN, such as the actuator density, communication, and execution radii, execution range, and the protocol performance were presented. Furthermore, the performance of AAP was compared with three existing algorithms: 1-SAAP, 1-SAAPG, and SAP. Simulation results showed that thanks to the use of adaptive factor, the AAP can always maintain the optimal performance in the applications with the number of actuators being consumed or variable communication radius when both communication cost and number of optimal assignments need to be optimized.

**Acknowledgements**

The author would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the article.

**Author contributions**

L.W. was involved in conceptualization, methodology, validation, formal analysis, investigation, resources, and writing—original draft preparation; J.K. and M.L. were involved in writing—review and editing; K.Y. and M.L. were involved in supervision; and M.L. and C.J. contributed to funding acquisition.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) received no financial support for the research, authorship, and/or publication of this article.

**ORCID iD**

Lu Wang https://orcid.org/0000-0001-5161-0392

**References**

1. Zheng J, Bhuiyan MZA, Shaohua L, et al. Auction-based adaptive sensor activation algorithm for target tracking in wireless sensor networks. *Fut Gen Comput Syst* 2014; 39: 88–99.
2. Ojha T, Misra S and Raghuwanshi NS. Wireless sensor networks for agriculture: the state-of-the-art in practice and future challenges. *Comput Elec Agri* 2015; 118: 66–84.
3. Rashid B and Rehmani MH. Applications of wireless sensor networks for urban areas: a survey. *J Net Comput Appl* 2016; 60: 192–219.
4. Rawat P, Singh KD, Chaouchi H, et al. Wireless sensor networks: a survey on recent developments and potential synergies. *J Supercomput* 2014; 68: 1–48.
5. Rault T, Bouabdallah A and Challal Y. Energy efficiency in wireless sensor networks: a top-down survey. *Comput Net* 2014; 67: 104–122.
6. Ahmed AT, Mahmoud SM, Ahmed A, et al. Energy harvesting internet of things health-based paradigm: towards outage probability reduction through inter-wireless body
area network cooperation. *Int J Dist Sensor Net.* Epub ahead of print 1 October 2019. DOI: 10.1177/1550147719879870.

7. Hammood DA, Rahim HA, Ahmad RB, et al. Enhancement of the duty cycle cooperative medium access control for wireless body area networks. *IEEE Access* 2019; 7: 3348–3359.

8. Ahmed A, Ahmed AT, Fahad AA, et al. WBSN in IoT health-based application: toward delay and energy consumption minimization. *J Sens.* Epub ahead of print 8 May 2019. DOI: 10.1155/2019/2508452.

9. Coates M. Evaluating causal relationships in wireless sensor/actuator networks. *IEEE Int Conf Acoust Speech Sig Process* 2005; 5:937-940.

10. Mezei I, Lukic M, Malbasa V, et al. Auctions and iMesh based task assignment in wireless sensor and actuator networks. *Comput Communicat* 2013; 36: 979–987.

11. Fatemi A, Zamanifar K and Nematbakhsh N. Capability-based task allocation in emergency-response environments: a coalition formation approach. *Turk J Electr Eng Comput Sci* 2013; 21: 1166–1181.

12. Kim Y and Matson ET. A realistic decision making for task allocation in heterogeneous multi-agent systems. *Proc Comput Sci* 2016; 94: 386–391.

13. Geyik SC, Shah SY, Szymanski BK, et al. Market mechanisms for resource allocation in pervasive sensor applications. *Pervas Mob Comput* 2012; 8: 346-357.

14. Panchu KP, Rajmohan M, Sundar R, et al. Multi-objective optimisation of multi-robot task allocation with precedence constraints. *Def Sci J* 2018; 68: 175–182.

15. Mezei I, Malbasa V and Stojmenovic I. Robot to robot: communication aspects of coordination in robot wireless networks. *IEEE Robot Automat Magaz* 2010; 17: 63–69.

16. Mezei I, Malbasa V and Stojmenovic I. Greedy extension of localized auction based protocols for wireless actuator task assignment. *Ad Hoc Sensor Wirel Net* 2012; 17: 73–85.

17. Okhovvat M, Sharifi M and Momeni H. Task allocation to actors in wireless sensor actor networks: an energy and time aware technique. *Proc Comput Sci* 2011; 3: 484–490.

18. Liwei H, Qu H and Zuo L. Multi-type UAVs cooperative task allocation under resource constraints. *IEEE Access* 2018; 6: 17841–17850.

19. Capitan J, Spaan MTJ, Merino L, et al. Decentralized multi-robot cooperation with auctioned POMDPs. *Int J Robot Res* 2013; 32: 650–671.

20. Coltin B and Veloso M. Mobile robot task allocation in hybrid wireless sensor networks. In: *Proceedings of the 2010 IEEE/RSJ international conference on intelligent robots and systems.* Taipei, China, 18–22 October 2010, pp. 2932–2937. New York: IEEE.

21. Feng W, Guangjie H, Jinfang J, et al. A task allocation algorithm based on score incentive mechanism for wireless sensor networks. *Int J Distri Sensor Net.* Epub ahead of print 16 August 2015. DOI: 10.1155/2015/286589.

22. Lee DH, Zaheer SA and Kim JH. A resource-oriented, decentralized auction algorithm for multirobot task allocation. *IEEE T Automat Sci Eng* 2015; 12: 1469–1481.

23. Lukic M, Barnawi A and Stojmenovic I. Robot coordination for energy-balanced matching and sequence dispatch of robots to events. *IEEE Tran Comput* 2015; 64: 1416–1428.

24. Choi HL, Brunet L and Jonathan PH. Consensus-based decentralized auctions for robust task allocation. *IEEE Tran Robot* 2009; 25: 912–926.